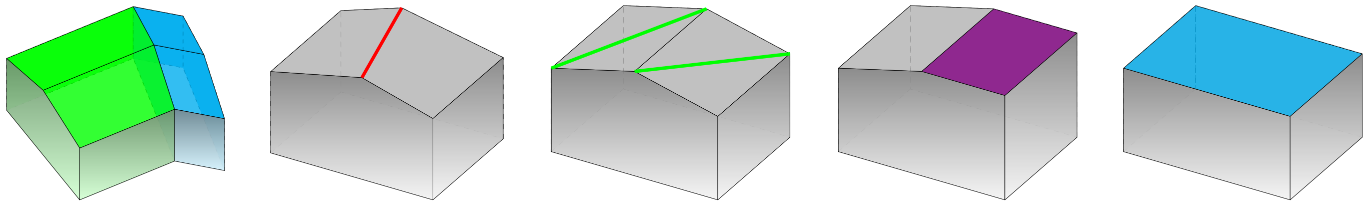
Semantic aware quality evaluation of 3D building models

Oussama Ennafii

Quality evaluation of 3D building models

A scalable approach

Oussama Ennafii



Jury

Franz Rottensteiner	IPI, Leibniz University Hannover, Germany	Reviewer
Gilles Gesquière	LIRIS, University of Lyon, France	Reviewer
George Vosselman	ITC, University of Twente, Netherlands	Examiner
Pooran Memari	LIX, École Polytechnique, France	Examiner
Renaud Marlet	IMAGINE, École des Ponts ParisTech, France	President
Arnaud Le Bris	LaSTIG, IGN-ENSG/University of Paris Est, France	Supervisor
Florent Lafarge	Titane, Inria, France	Co-advisor
Clément Mallet	LaSTIG, IGN-ENSG/University of Paris Est, France	Advisor

LaSTIG
Institut National de l'Information Géographique et Forestière
(IGN)
73 avenue de Paris, 94165 Saint-Mandé cedex, France

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy delivered by:

Université Paris-Est – MSTIC (ED532)

Spetialty: Geographical Information Sciences and technologies

Quality evaluation of 3D building models

A scalable approach

Oussama Ennafii

Defended *January 10, 2020*
at *ISC-PIF, Paris, France*

Jury

Franz Rottensteiner	IPI, Leibniz University Hannover, Germany	Reviewer
Gilles Gesquière	LIRIS, University of Lyon, France	Reviewer
George Vosselman	ITC, University of Twente, Netherlands	Examiner
Pooran Memari	LIX, École Polytechnique, France	Examiner
Renaud Marlet	IMAGINE, École des Ponts ParisTech, France	President
Arnaud Le Bris	LaSTIG, IGN-ENSG/University of Paris Est, France	Supervisor
Florent Lafarge	Titane, Inria, France	Co-advisor
Clément Mallet	LaSTIG, IGN-ENSG/University of Paris Est, France	Advisor

LaSTIG
Institut National de l'Information Géographique et Forestière
(IGN)
73 avenue de Paris, 94165 Saint-Mandé cedex, France

Colophon

This disseration was typeset using the LaTeX typesetting system originally developed by Leslie Lamport, based on TeX created by Donald Knuth.

The thesis took place at the StruDEL (Structures spatio-temporelles pour l'analyse des territoires) team at LaSTIG (Laboratoire en Sciences et Technologies de l'Information Géographique) and The Titane team at Inria Sophia Antipolis.

The bibtex entry to cite this thesis is:

```
@phdthesis{ennafii2020,
    author = {{Ennafii, Oussama}},
    title = {{Quality evaluation of 3D building models:
            a scalable approach}},
    school = {{Université Paris-Est}},
    year = {2014},
    month = {January}
}
```

This thesis is dedicated to my parents and family.

Abstract

The automatic generation of 3D building models from geospatial data is now a standard procedure. An abundant literature covers the last two decades and several softwares are now available. However, urban areas are very complex environments. Inevitably, practitioners still have to visually assess, at city-scale, the correctness of these models and detect frequent reconstruction errors. Such a process relies on experts, and is highly time-consuming with approximately 2 h/km^2/expert. This work proposes an approach for automatically evaluating the quality of 3D building models. Potential errors are compiled in a novel hierarchical and modular taxonomy. This allows, for the first time, to disentangle fidelity and modeling errors, whatever the level of details of the modeled buildings. The quality of models is predicted using the geometric properties of buildings and, when available, Very High Resolution images and Digital Surface Models. A baseline of handcrafted, yet generic, features is fed into a Random Forest or Support Vector Machine classifiers. Advanced features, relying on graph kernels as well as Scattering Networks, were proposed to better take into consideration structure. Both multi-class and multi-label cases are studied: due to the interdependence between classes of errors, it is possible to retrieve all errors at the same time while simply predicting correct and erroneous buildings. The proposed framework was tested on three distinct urban areas in France with more than 3000 buildings. 80 to 99 % F-score values are attained for the most frequent errors. For scalability purposes, the impact of the urban area composition on the error prediction was also studied, in terms of transferability, generalization, and representativeness of the classifiers. It shows the necessity of multi-modal remote sensing data and mixing training samples from various cities to ensure a stability of the detection ratios, even with very limited training set sizes.

Keywords. 3D urban modeling, Buildings, Dataset, Quality assessment, Error taxonomy, Error detection, Aerial imagery, Very High Resolution, Digital Surface Model, Geometry, Statistical learning, Multi-label classification.

ملخص

اصّناعة ديال انّمادج تولاتية لبعاد ديال لبنايات بطريقة تلقاءيّة ولّات معمول بيها هاد ليّام. كاين فهاد عشرين عام انّحرّة عدد كافي من لمراجع لّي اتّطرقات لهاد لموضوع، زيد عليها عدد ديال لبارامج. لموشكيلة أنّاهو لمجالات لحضريّة منوّعة بزّاف، اشّي الّي تيخلّي أنّا، بطريقة ولّا اخرة، أين موقاربة تقدر تغلط. هادشي تيدفع فلواقع أنّانا نتئكّدو باشّوف من اصّحة ديال انّمادج، لقاضيّة الّي تاتاخد عاداتان 2 سوايع فكيلومطرمربع لكلّا بناية لكلّا خبير. على ود هادشي تانقتارحو طريقة باش نقيّمو انّمادج ديال لبنايات تلقاءيّان. أول حاجة هيّا أنّانا صنّفنا لأغلاط الّي تقدر تخسّر اّنمادج بلوحدات أو بشكل هرميّ. هاد اتّفراز تيفرّق لأول مرّة مابين لمشاكيل ديال اشّكل من لأغلاط فادّقة. باش نقشعو هاد لأغلاط الّي عرّفنا، تانعوّلو على لخاصّيات لهندسية ديال انّموذج الّي تانتحقّقومنّو، بازيّادة على خاصيّات برّانيّة منّي تانقارنو انّموذج مع تصويرة عموديّة ولّا تصويرة ديال لعمق. فهاد لخدمة الّي درنا، قتارحنا أولا خاصيّات بصيطة أوليّة باش نتئكّدو من لمقاربة ديالنا. هاد لخاصيّات تاندوزوهوم لموصنّيفات بحال لمفرّز بأوسع هامش أولّا لغابة ديال اتّفراق لعلّا وي. فلمحالة اتّالية، قتارحنا خاصيّات آخرة باش نتحقّقو من كولّا نمودج. هاد لخاصيّات تيعوّلو على نفس لمعطايات فادّخلة، والاكن تاتعتمد علا طريقات مقدّمة كتر. توكدنا من هاد اتّسلسل باتّجربة على تلاتة ديال لمدينات ففرانسا: **ايلونكور** (2009 بناية)، **نانط** (748 بناية) أو لحيّ 13 ديال باريز (**باريز-13**: 478 بناية). انّتيجة ديال هاد اتّجاريب عطات معدّل ديال ادّقة أو لكموليّة مابين 80 % أو 99 %. تأكدنا تاني فهاد اتّجاريب أنّا اتّفراز الّي تانتعلموه فمنطقة ايقدر ايتطبق لمنطقات أوخرى بسيفتو قابل إيتحوّل أو إيتعمّم أو إيتدرّج.

الكلمات الرئيسية. انّمدجة تولاتيّة لبعاد ديال لمناطق لحضريّة، مجموعة دلموعطايات، اتّحقّق ديال لجودة، اتّصنيف ديال لأغلاط، لقشيع دلأغلاط، اتّصاور اسّماويّة، اتّصاور ارّقّمية ديال اصّطح، لهندسة، اتّعلّوم بلإيحصاء، اتّصنيف متعدّد.

Résumé

La génération automatique de modèles de construction 3D à partir de données géospatiales est maintenant une procédure standard. Une littérature abondante couvre les deux dernières décennies et plusieurs solutions logicielles sont maintenant disponibles. Cependant, les zones urbaines sont des environnements très complexes. Inévitablement, les producteurs de données doivent encore évaluer visuellement, à l'échelle de villes, l'exactitude de ces modèles et détecter les erreurs fréquentes de reconstruction. Un tel processus fait appel à des experts et prend beaucoup de temps, soit environ 2 h/km^2/expert. Cette thèse propose une approche d'évaluation automatique de la qualité des modèles de bâtiments 3D. Les erreurs potentielles sont compilées dans une nouvelle taxonomie hiérarchique et modulaire. Cela permet, pour la première fois, de séparer erreurs de fidélité et de modélisation, quelque soit le niveau de détail des bâtiments modélisés. La qualité des modèles est estimée à l'aide des propriétés géométriques des bâtiments et, lorsqu'elles sont disponibles, d'images géospatiales à très haute résolution et des modèles numériques de surface. Une base de référence de caractéristiques *ad hoc* génériques est utilisée en entrée d'un classificateur par Random Forests ou par Séparateurs à Vaste Marge. Des attributs plus riches, s'appuyant sur des noyaux de graphes ainsi que sur des réseaux de type Scattering ont été proposés pour mieux prendre en compte la structure dans la donnée 3D. Les cas multi-classes et multi-étiquettes sont étudiés séparément : de par l'interdépendance entre les classes d'erreurs, il est possible de détecter toutes les erreurs en même temps tout en prédisant au niveau sémantique le plus simple des bâtiments corrects et erronés. Le cadre proposé dans cette thèse a été testé sur trois zones urbaines distinctes en France avec plus de 3000 bâtiments étiquetés manuellement. Des valeurs de F-score élevées sont atteintes pour les erreurs les plus fréquentes (80 – 99 %). Pour une problématique de passage à l'échelle, l'impact de la composition de la zone urbaine sur la prédiction des erreurs a également été étudié, en termes de (i) transférabilité, de (ii) généralisation et de (iii) représentativité des classificateurs. Cette étude montre la nécessité de disposer de données de télédétection multimodale et de mélanger des échantillons d'entraînement provenant de différentes villes pour assurer une stabilité des taux de détection, même avec des tailles d'ensembles d'entraînement très limitées.

Mots-Clés. Modélisation 3D de ville, Bâtiments, Jeu de données, Qualification, Taxonomie d'erreurs, détection d'erreurs, Imagerie aérienne, Très Haute Résolution, Modèle Numérique de Surface, Géométrie, Apprentissage statistique, Classification multilabel.

Contents

List of Figures

List of Tables

Acknowlegements

I would like to thank Clément who not only was my advisor but also supervised my work from the start. I learned a lot with him in different aspects. He always pushed me to do better and be as rigorous as humanly possible. In fact, despite being very knowledgeable, he is always ready to learn new things.

I have to acknowledge equally Arnaud (a.k.a. *Chuck*) for always being available. He seems to always have answers to every question, be it scientific, technical, administrative or historical of nature: hence the nickname. He was also available, to his great sorrow, to suffer a great deal trying to decipher the first iterations of my writings.

Florent deserves also a special recognition. Despite being far from us, he provided us with sound advice at each turn. I would like to thank him and Pierre Alliez for inviting me for a stay at the Titane team. I thank Pierre also for being part of the thesis committee.

I would like to address my recognition to Renaud Marlet, who was interested from the beginning in our work. Being a member of the thesis committee, he also accepted to preside over the defense jury. I thank also Franz Rottensteiner and Gilles Gesquière for their reviews and thorough comments. I have George Vosselman and Pooran Memari to thank for their relevant remarks.

I would like to acknowledge Mohammed El Rhabi who was of great help as my academic supervisor at École des Ponts Paristech.

I thank also Sébastien who was, for the first half of my stay at LaSTIG, my office mate. He had to suffer my (terrible) juggling skills and my numerous `QGIS` questions. After suffering for one year and a half from the scorching heat of the south wing, I moved north to find myself with Teng, Raphaël and Stéphane for the rest of my stay. I have to thank them for the time we spent together, mainly playing football in the office (sorry Teng) and learning each other's languages. Stéphane deserves a special mention, as he was of great help especially during the last months of my thesis.

Je tiens à remercier en particulier David Correia qui nous a énormément aidé, surtout quand on commettait des anneries (`<>:/usr/lib/foo$ sudo rm -rf ../`). Je ne peux pas aussi oublier Laurent Schneider qui m'a été d'une grande aide pour déchiffrer les erreurs de compilation de `CGAL` et qui a joué beaucoup, malgré lui, le rôle de rubber duck. Marie-Claude ainsi que Alain qui, à eux seuls, maintiennent le fonctionnement du laboratoire, méritent la reconnaissance de tout le personnel du LaSTIG.

During my three year stay at LaSTIG, I had the opportunity to meet some endearing people out of which I can mention only a few: Nathan, Bastien, عمّي علي, Quy Thy, Yilin (always making jokes), محمّد, أمين (Diabetes guy), Imran (SJW), Lâman (Tea guy), Anatol (Némar fan boy), Laurence (standing tall), Mattia (the literary guy who uses Haskell), Ewelina (globe trotter), Laurent (l'élite), إيمان, Jean-Pierre, Quoc, Neelanjan, Vivien (για χαρά), David (le Nantais), Bahman, Clément "Junior", Yann (le faux breton), Alexandre (a.k.a. Waldo), Paul (Coffee dictator), Maxime (le banlieusard de Lyon), Evelyne, Qasem (a.k.a. Alex 2), Yanis (l'autre banlieusard de Lyon), Bruno, Mathieu, Guillaume, Julien (a.k.a. Jupé), Marc (spécialiste de l'apéro), Bénedicte and others.

Je tiens aussi à remercier Marc Poupée, et l'ensemble du staff de l'ENSG, avec qui j'ai collaboré pour dispenser les cours au cycle ingénieur.

I would also like to thank members of the Titane team during my stay at Inria Sophia Antipolis: namely Cédric, Muxingzi, Gaétan, Nicolas, Jean-Philippe and Flora.

ضاروري نشكر أهم ناس فحياتي. تانهضر بلخصوص على واليديّا، الّي ضحّاو بكترمن 25 عام من شبابهوم على ودّ موستقبالي. بغيت نشكر تاني هدى، خطيبتي، الّي صبرات معايا هاد تلت سنين أو تحمّلات جزء من اضّغط ديال خدمتي. مروان ”ضارك فزّ"، محمّد ”اصّايمو"، يوسف ”الّي عارف راسو" أو سامي ”يا زلمي" تيستاحقو حتّى هوما اشّكر أو لعتيراف ديالي: وقفو معيا فواحد من أصعب لمواقيف فحياتي.

Résumé étendu

Sommaire

Dans ce chapitre, un résumé étendu du mémoire est présenté. On commence par mettre le sujet dans le contexte et de le motiver dans l'introduction. Après quoi, notre méthode de qualification de modèles 3D de bâtiments est présentée. En troisième lieu, nous présentons les différents résultats expérimentaux. Ce résumé étendu est achevé par une conclusion et une liste de perspectives possibles.

Introduction

Modélisation 3D de bâtiments

On entend par modèle 3D de bâtiment un produit cartographique qui représente la surface du bâtiment en question. À l'image d'une carte 2D, ce modèle est une généralisation de la réalité dont le but n'est pas de représenter minutieusement tous les détails mais les principales structures (au sens de spécifications). Cependant, le modèle doit rester le plus fidèle, en terme géométrique, du bâtiment qui nous intéresse. Ainsi, la fidélité géométrique est pondérée par rapport à la généralisation et la compacité. Le bon compromis est choisi en fonction des besoins de l'utilisateur final.

La géométrie de la surface du modèle n'est donc pas suffisante pour décrire les objets urbains (Biljecki et al., 2016b), et en particulier les bâtiments. Le problème de modélisation rejoint donc la notion de sémantique. Les modèles de ville peuvent enregistrer d'autres informations comme la fonction de chaque élément architectural. On nomme ce type d'information la sémantique *explicite*. Cette dernière a un effet significatif sur la géométrie du modèle. En effet, les éléments architecturaux correspondent généralement à une ou plusieurs formes géométriques simples, le plus souvent planes (Kolbe et al., 2005). En conséquence, une information géométrique dense (c'est-à-dire un maillage 3D dense) n'est pas nécessairement la représentation la plus précise ou, du moins, la meilleure. Cet effet est désigné ici par le terme de sémantique *implicite*, car pas réellement exprimée. Connaître la fonction d'un objet permet donc de représenter sa géométrie de manière efficace. Ainsi, la sémantique implique une compacité dans la représentation des bâtiments. C'est pourquoi le dernier critère a été utilisé, par exemple, en plus de l' Erreur Quadratique Moyenne (EQM), comme mesure d'évaluation dans (Lafarge et al., 2012). Ainsi, distingue-t-on désormais modèles 3D et maillages 3D de bâtiment. Si ces derniers ne prennent en compte que la précision géométrique, les autres véhiculent également des propriétés sémantiques.

Ces modèles 3D urbains sont importants pour plusieurs applications dans différents domaines. Une étude plus complète de ces applications a été présentée dans (Biljecki et al., 2015b). L'objectif est de persuader le lecteur de la pertinence de la modélisation 3D de bâtiments et de l'importance de l'impact qu'elle peut avoir sur tout le monde. En effet, les modèles urbains en 3D répondent à des besoins divers : administratifs, environnementaux, scientifiques et sociétaux. Nous présentons ici les applications liées à préservation de l'environnement qui seront d'une importance capitale pour les années à venir. Les agglomérations urbaines sont l'un des plus gros consommateurs d'énergie. Une utilisation plus efficace de l'énergie est nécessaire pour soutenir leur croissance effrénée. C'est pourquoi il est nécessaire de quantifier la consommation d'énergie des établissements urbains (Wate et al., 2015) ou les coûts de modernisation (Previtali et al., 2014). Biljecki et al. (2015a) utilisent également des modèles 3D de bâtiments afin de prévoir l'irradiation solaire. En effet, l'estimation du potentiel solaire peut être utile pour évaluer la pertinence de projets de panneaux solaires coûteux. Ce type d'études peut également être appliqué à l'urbanisme, car les simulations pourraient être calculées pour les futurs développements urbains.

Le sujet de la modélisation 3D des bâtiments a été largement étudié depuis plus de vingt ans. Cependant, il existe encore quelques problèmes non résolus dans ce domaine (Musialski et al., 2013 ; Lafarge, 2015). Le premier relève de la donnée acquise qui peut souffrir de divers défauts (bruit d'acquisition, recalage de données, données manquantes ...) qui sont répercutés sur le modèle final. La deuxième problématique touche à l'automatisation qui reste encore inatteignable. On peut aussi citer un troisième verrou à lever : l'évaluation qui est encore généralement réalisée de façon manuelle.

Évaluation de modèles de bâtiments

L'évaluation de modèles 3D de bâtiments peut porter sur la vérification de leur cohérence topologique. Un travail considérable a été accompli afin de parvenir à une représentation standardisée des modèles 3D des villes. Cela a notamment abouti à la norme CityGML de l'Open Geospatial Consortium (OGC) (Gröger et al., 2012b). Cependant, dans la pratique, elle n'est pas toujours respectée, comme le montre (Biljecki et al., 2016a), où jusqu'à 89 % des modèles se sont avérés non valides du point de vue topologique. Cela peut expliquer pourquoi le sujet de l'inspection automatique de la cohérence topologique des modèles de ville a attiré une forte attention dans la communauté des SIG. On peut noter en particulier les travaux présentés dans (Ledoux, 2013) qui est le premier a explorer pleinement les possibilités topologiques offertes par la norme CityGML.

L'évaluation de modèles 3D de bâtiments peut relever aussi de la comparaison entre la géométrie du modèle 3D et la géométrie réelle du bâtiment. Deux moyens peuvent être utilisés pour juger de la qualité de la représentation géométrique d'un bâtiment. L'évaluation manuelle repose sur l'interaction humaine pour déterminer dans quelle mesure le modèle est proche de la réalité. L'approche automatique s'appuie uniquement sur le modèle et d'autres données[1] sans impliquer un opérateur humain dans la boucle. La dernière approche est la plus intéressante, mais aussi la plus difficile. En effet, même si l'évaluation de la géométrie peut s'apprêter à être automatisé, le point névralgique reste l'aspect sémantique de l'évaluation. Bien que cette évaluation sémantique soit facile à réaliser manuellement, elle résiste encore à l'automatisation, mais a aussi été relativement peu étudiée.

Contributions

Fondée sur la discussion précédente, nous avons adopté une orientation de recherche qui a rarement été prise jusqu'à présent, pour autant que nous le sachions. Notre objectif est d'évaluer la qualité des Modèles 3D de bâtiments de manière **automatique** et à **grande échelle**. Ceci implique de mettre en place :

1. Une taxonomie hiérarchique et adaptative des erreurs est proposée.
2. Une évaluation sans Modèles 3D de bâtiments de référence.
3. Des attributs afin de caractériser les Modèles 3D de bâtiments ainsi que de les comparer à des images ou Modèle Numérique de Surface (MNS).
4. Une étude de passage à l'échelle pour la chaîne de traitement établie.
5. Un ensemble d'outils pour traiter la géométrie des modèles 3D de bâtiment.

Qualification de modèle 3D de bâtiments

État de l'art

Différentes méthodes de qualification de modèles 3D urbains ont été proposées. Elles peuvent être classées selon les critères sur lesquelles elles s'appuient : **indices géométriques de précision** ou **erreurs sémantiques** (topologiques ou géométriques). Les indices géométriques permettent de quantifier la précision d'une modélisation à partir de

[1] e.g. Modèle de référence, données 3D de télédétection...

la précision de points particuliers (sommets, points d'intersection …), des surfaces ou des volumes des modèles 3D, en les comparant à des données de références de plus grande précision (Zeng et al., 2014). Ces indices ne permettent cependant pas de bien décrire tous les défauts d'une reconstruction et sont, la plupart du temps, trop locaux. Une taxonomie d'erreurs sémantiques est donc préférable. Elle peut reposer sur le paradigme des feux de circulation (Boudet et al., 2006) (Correct, Acceptable, Généralisé et Faux), mais nécessite de définir le niveau de généralisation acceptable pour une reconstruction. La taxonomie peut également adopter le point de vue des méthodes de reconstruction : i.e. connaissant les sources d'erreur affectant cette méthode. Les erreurs peuvent, alors, être discriminées en erreurs d'emprise de bâtiments (contour erroné, bâtiment inexistant, cours intérieure manquante et emprise imprécise), en erreurs de reconstruction intinsèques (sous-segmentation, sur-segmentation, toit inexact, translation en Z) à la méthode et en erreur due à l'occlusion végétale comme dans (Michelin et al., 2013). Dans les deux cas précédants, l'évaluation d'un modèle urbain est donc faite grâce une classification supervisée qui prend comme étiquettes les erreurs ainsi définies. Pour caractériser ces modèles, des attributs peuvent être calculés à partir d'images aériennes ou de MNS à très haute résolution spatiale (20à25 cm), en comparant des segments 3D ou des indices de corrélation de texture, comme par exemple (Boudet et al., 2006 ; Michelin et al., 2013). La plupart du temps, la difficulté réside dans le choix de la taxonomie. Il faut éviter qu'elle soit trop générale pour ne pas être surajustée par rapport à une scène ou une méthode de reconstruction donnée. C'est le type d'approche qui a ici été retenu.

Une taxonomie d'erreurs de modélisation

Pour définir une nouvelle taxonomie générique mais flexible, deux critères dont pris en compte : le Niveau de Détails (LoD) et la `finesse` de l'erreur. La `finesse` représente le niveau de spécificité des erreurs. Une erreur est dite de `finesse` maximale si elle correspond à une action unitaire de la part d'un opérateur au moment de sa correction. On définit ainsi ce que l'on appellera une erreur `atomique`.

Du point de vue opérationnel, les bâtiments ne sont pas tous qualifiables. En effet, quelques bâtiments peuvent être occultés par la végétation ou se trouver au bord de la région traitée. Dans ces cas pathologiques, nous estimons que la qualification n'est pas un problème bien défini. Nous discriminons, ainsi, bâtiments `qualifiables` et bâtiments `non qualifiables`. Cette classification est considérée de `finesse` = 0. Au niveau de `finesse` suivant, les bâtiments sont classés selon qu'ils sont `Valides` ou `Erronés`. Ces derniers sont ensuite divisés selon le Niveau de Détail LoD en familles d'erreurs de `finesse` = 2. En effet, une première famille d'erreurs, nommée `Erreurs de Bâtiment`, est consacrée aux défauts qui affectent le bâtiment dans son intégralité (niveau LoD-0 ∪ LoD-1). À l'inverse, la famille `Erreurs de Facette` contient les erreurs qui concernent les facettes — façades ou toit — des bâtiments (niveau LoD-2 ∪ LoD-3). Ces familles contiennent chacune des erreurs `atomiques` de `finesse` maximale égale à 3. Ces erreurs sont présentée dans la Figure 1.

Cette catégorisation est indépendante de la méthode de reconstruction ou de la scène à modéliser. L'étiquetage est non redondant : les erreurs `atomiques` relevées sont indépendantes entre elles et ne représentent que des défauts particuliers, topologiques ou géométriques. Les erreurs topologiques relèvent les erreurs de structure du modèle reconstruit. Les erreurs géométriques mettent en évidence l'imprécision de la reconstruction. Chaque erreur `atomique` se voit attribuée une note, au moment de l'annotation par l'opérateur, sur une échelle de 0 to 10, et représente le degré de confiance en la présence du défaut.

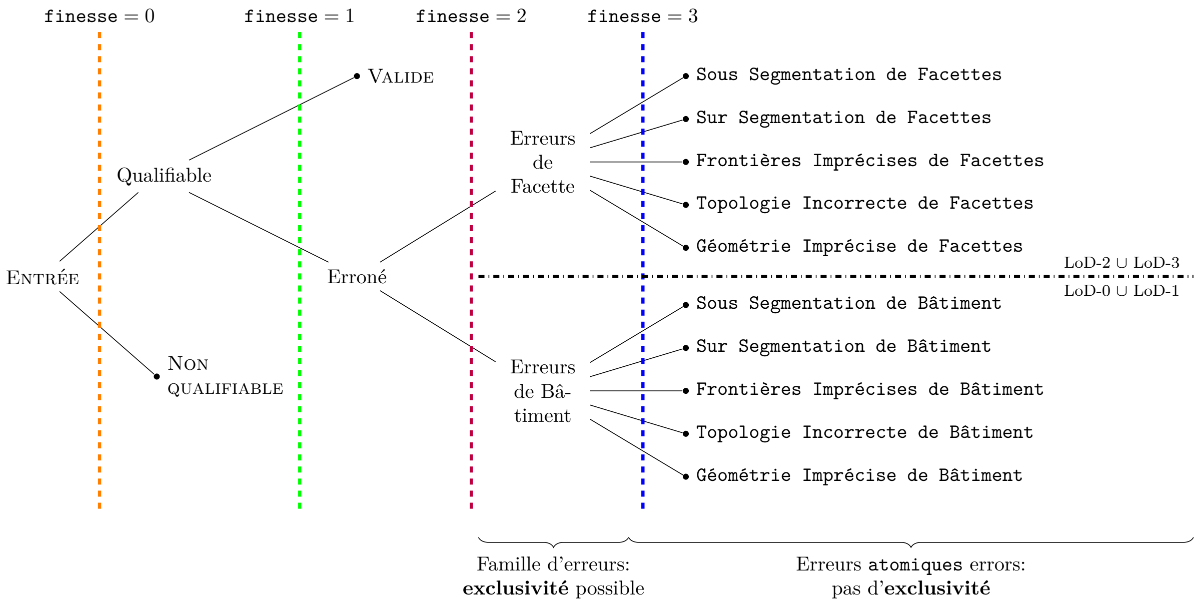

finesse = 0
finesse = 1
finesse = 2
finesse = 3
Entrée
Qualifiable
Non qualifiable
Valide
Erroné
Erreurs de Facette
Erreurs de Bâtiment
Sous Segmentation de Facettes
Sur Segmentation de Facettes
Frontières Imprécises de Facettes
Topologie Incorrecte de Facettes
Géométrie Imprécise de Facettes
LoD-2 ∪ LoD-3
LoD-0 ∪ LoD-1
Sous Segmentation de Bâtiment
Sur Segmentation de Bâtiment
Frontières Imprécises de Bâtiment
Topologie Incorrecte de Bâtiment
Géométrie Imprécise de Bâtiment
Famille d'erreurs: exclusivité possible
Erreurs atomiques errors: pas d'exclusivité

Fig. 1 : La taxonomie d'erreurs proposée. Dans le cas de modèlisation à partir d'image aérienne Très Haute Résolution spatiale, deux familles d'erreur sont présentées. Au niveau 2 de `finesse`, l'hierarchisation entre ces familles est possible : le paramètre d'**exclusivité** peut intervenir. Cependant, au niveau 3 de `finesse`, les erreurs `atomiques`, étant indépendants entre eux, sont indifférents à l'**exclusivité**.

Cela revient à une discrétisation de la probabilité d'existence de l'erreur. Les erreurs de finesse inférieure héritent des erreurs de leurs filles (i.e. de `finesse` plus grande). En effet, elles sont aussi sûres que les erreurs qu'elles contiennent. Leur note attribuée est donc le maximum des notes des erreurs filles.

Au moment de la qualification, trois paramètres entrent en jeu : un niveau de détail d'évaluation (**eLoD**), un niveau de `finesse` d'évaluation (**eFin**) et l'**exclusivité**. En précisant un **eLoD** donné, les erreurs de plus grand Niveau de Détail sont ignorées. En fixant une **eFin** donnée, on ne discrimine que selon les erreurs du même ordre de `finesse`. Le dernier paramètre est l'**exclusivité** des erreurs. Dans la cas exclusif, nous ne relevons que la famille d'erreurs représentant le plus petit Niveau de Détail : c'est un problème de classification Multi-Classes. Dans le cas contraire, nous rapportons toutes les erreurs (i.e. un objet peu être affecté par plusieurs erreurs) : c'est un problème Multi-Étiquettes.

On propose ici les familles d'erreurs suivantes pour le cas de modélisation de bâtiments à partir de données aériennes ou satellitaires :

1. `Erreurs de Bâtiment` :
   - Sous segmentation (`BOS`) : deux bâtiments, ou plus, représentés comme un seul ;
   - Sur segmentation (`BUS`) : un bâtiment est modélisé en deux ou plusieurs bâtiments ;
   - Frontières imprécises (`BIB`) : les frontières de l'emprise du bâtiment sont inexactes ;
   - Topologie incorrecte (`BIT`) : la topologie de l'emprise du bâtiment est inexacte ;
   - Géometrie imprécise (`BIG`) : la géometrie 3D du bâtiment est mal estimée ;
2. `Erreurs de Facette` :

- Sous segmentation (`FOS`) : deux facettes, ou plus, représentées comme une seule ;
- Sur segmentation (`FUS`) : une facette est modélisée en deux ou plusieurs facettes ;
- Frontières imprécises (`FIB`) : les frontières de la facette sont inexactes ;
- Topologie incorrecte (`FIT`) : imprécise : la topologie de la facette est inexacte ;
- Géometrie imprécise (`FIG`) : la géometrie 3D de la facette est imprécise.

L'apprentissage au service de la qualification

Afin de satisfaire les contraintes de **passage à l'échelle** et d'**automatisation**, nous proposons de formuler le problème comme un problème d'apprentissage supervisé. Les erreurs sont considérées comme des étiquettes à prédire. Des attributs sont calculés de manière à décrire les bâtiments observés. En réalité, en première approche, l'existence de toutes les erreurs est prédite au niveau du bâtiment, même pour les étiquettes d'`Erreurs de Facette`.

Nous proposons ainsi des attributs de base pour les modèles de bâtiments qui restent les plus simples possibles. Ils sont de trois types :

Les attributs géométriques : ils sont calculés à partir de statistiques (histogramme ou liste contenant le maximum, le minimum, la moyenne, le médian et/ou l'écart type) de quelques propriétés géométriques des facettes du bâtiments : nombre de sommets, aire de chaque facette, circonférence de chaque surface, angles entre les normales de facettes adjacentes, distance entre les centroïdes des facettes adjacentes et/ou toutes les facettes.

Les attributs basés sur la hauteur : ils sont issus d'un histogramme de la différence entre le modèle 3D et un MNS externe à très haute résolution spatiale.

Les attributs basés sur l'image : ils sont issus d'un histogramme de la similarité entre les arêtes du modèle 3D projeté dans la direction nadir et des arêtes réelles dans l'orthoimage correspondante.

La résolution du MNS et de l'orthoimage doit être inférieure à l'ordre de grandeur des bâtiments mais aussi plus grande que l'erreur planimétrique du modèle 3D afin de garantir une robustesse au bruit. Différents attributs (géométrie(4×5)+hauteur(20)+image$(20) = 60$) obtenus sont en suite concaténés dans un seul vecteur.

Expérimentations

Données

Nous avons sélectionné des modèles 3D tirés de trois villes française différentes afin d'évaluer la performance de notre protocole : **Élancourt**, **Nantes**, et le XIIIe arrondissement de Paris (**Paris-13**). La petite ville d'**Élancourt** contient divers types de bâtiments : des zones résidentielles avec des bâtiments à toit en croupe ainsi que des quartiers avec de grands bâtiments industriels à toit plat. **Nantes** représente un cadre urbain plus dense mais avec une diversité de bâtiments plus faible. **Paris-13** se compose principalement de tours à toit plat qui coexistent avec des bâtiments de style haussmannien dont les toits sont généralement très fragmentés. La scène d'**Élancourt** (*resp.* **Nantes** et **Paris-13**) contient 2009 (*resp.* 748 et 478) modèles de bâtiments annotés. Afin de traiter ces modèles, `proj.city`, une bibliothèque `C++`, a été développée en s'appuyant sur la bibliothèque `CGAL` (Fabri et al., 2000). Grâce à cet outil, nous pouvons projeter des modèles de bâtiments dans la direction du nadir et produire les cartes de hauteur correspondantes. La résolution spatiale du MNS et de l'image orthorectifiée est de 0,06 m pour la première zone, alors qu'elle est de 0,1 m pour les autres.

Les modèles 3D ont été générés à partir d'une base de données d'emprises cadastrales par la méthode de (Durupt et al., 2006). L'extrapolation de la topologie du toit se fait en simulant les formes de toits plausibles, à partir des images orientées, avant de les confronter à un MNS à 0,06 m (*resp.* 0,1 m) de résolution sur **Élancourt** (*resp.* **Nantes** et **Paris-13**). Les façades du modèle relient la ligne de goutière au sol. On obtient une modélisation 2.5D de la scène à un LoD-2.

Expériences Socles de base

Nous avons utilisé un classifieur de type Forêt aléatoire sur deux types d'attributs : un simple EQM ainsi que les attributs de bases. Les Forêts Aléatoires — qui peuvent gérer des descripteurs multimodaux et nombreux — sont choisis avec 1000 arbres de décisions de profondeur maximale égale à 4. La profondeur est ainsi limitée dans le but d'éviter le surapprentissage, contrairement au nombre d'arbres qui est très grand de façon à couvrir tout l'espace d'attributs. Ce classifieur a été adapté, au cas de la classification Multi-Étiquettes, en utlisant la stratégie *Un contre Tous*.

D'après les résultats expérimentaux, au niveau **eFin** $= 3$, nous avons appris que l'EQM ne permet pas de prévoir complètement les erreurs définies dans notre taxonomie. En outre, les `Erreurs de Bâtiment` sont mieux détectées sur **Élancourt** que sur les autres scènes. Concernant l'erreur `FOS`, elle est très bien détectée avec plus de 95 % de F-score, sur toutes les zones d'intérêt. À l'exception de cette dernière, plus la zone urbaine est dense, mieux les `Erreurs de Facette` sont détectées. Comme prévu, les erreurs rares telles que `BIT` et `FIT` sont mal détectées sur toutes les zones urbaines. Les attributs géométriques seules ont prouvé être suffisants pour la prédiction d'erreur. Par conséquent, le type de bâtiment est donc un bon indicateur pour la détection d'erreurs dans une même zone. Ceci est toujours vrai à une exception près : `BUS` pour laquelle les autres modalités sont significativement meilleures. Paradoxalement, bien que, dans la plupart des cas, elles n'améliorent pas la prédiction des erreurs, les modalités extrinsèques sont aussi importantes que les caractéristiques géométriques intrinsèques, d'après les importances d'attributs fournis par les forêts aléatoires.

Aux niveaux 2 et 1 de l'**eFin**, nous avons aussi prouvé expérimentalement que les `Erreurs de Facette` sont détectées avec un F-score d'environ 90 % non affecté par la scène d'intérêt. Comme nous l'avons vu précédemment au niveau 3 de l'**eFin**, les `Erreurs de Bâtiment` sont difficiles à entraîner sur **Nantes** et **Paris-13**, avec un F-score d'environ 75 %. En contre partie, comme prévu, la distinction entre `Erroné` et `Valide` s'est avérée difficile à prédire.

Expériences de passage à l'échelle

L'objectif ici est de prouver l'extensibilité de l'approche proposée. Nous avons gardé les mêmes configurations expérimentales que dans les expériences Socles. Pour y parvenir, nous avons testé la capacité des classifieurs entraînés sur un ensemble de sources à obtenir des bons scores sur un ensemble cible différent. En fonction de la manière dont l'ensemble source a été choisi, trois types d'expériences ont été examinés : la transférabilité, la généralisation et la représentativité. Elles ont toutes donné des résultats cohérents qui ont contribué à prouver entre autre que **Nantes** et **Paris-13** se ressemblent comparés à **Élancourt**. Nous avons aussi vu comment la prédiction de `FOS` et des `FIG` est hautement transférable, avec un F-score de plus de 95 % et que les `Erreurs de Facette`, en général, sont plus faciles à passer à l'échelle que celle des `Erreurs de Bâtiment`. D'autre part, ces dernières ainsi que `FIT` sont mieux détectées à **Élancourt** que dans les autres villes. En effet, à l'exception de `FIT`, les `Erreurs de Facette` sont mieux détectées sur **Nantes** ou **Paris-13**. Les erreurs rares, telles que les `BIT` et les `FIT`, sont toujours mal détectées. Concernant la représentativité, 20 % des échantillons prouvaient être suffisants pour l'apprentissage sur l'ensemble de données mixtes. D'un autre point de vue, les attributs géométriques ne sont pas aussi bon que les modalités extrinsèques pour le passage à l'échelle. En effet, ces dernières (en particulier celles fondées sur les images) jouent un rôle plus crucial dans la transférabilité et la généralisation.

Attributs avancés

Dans cette section, nous avons examiné expérimentalement la valeur ajoutée d'attributs plus élaborés que nous avons proposé. Nous avons pu voir comment dans tous les cas, ScatNet (Mallat, 2012 ; Sifre et al., 2013 ; Oyallon et al., 2015) s'est avéré plus utile que les attributs de base pour la prédiction d'erreurs. Ainsi, contrairement à la version de base, les attributs basés sur la hauteur avec ScatNet ont-ils prouvé être crucial pour détecter les étiquettes d'erreur. Bien qu'ils soient utiles, d'un point de vue d'importance d'attributs, ces attributs sont, de loin, les plus faibles. La fusion retardée a prouvé être la meilleure option lorsqu'elle était utilisée avec de Forêts aléatoires, tandis que la fusion précoce fonctionnait mieux avec le classifieur SVM. En utilisant les attributs basés sur ScatNet, le SVM s'est avéré globalement meilleur que les Forêts aléatoires, en particulier sur **Élancourt**. Concernant les noyaux pour graphes, comparés aux attributs de base, ils permettent d'améliorer les résultats de prédiction des étiquettes, à condition que celles-ci soient suffisamment fréquent. **Élancourt** était la zone la plus facile pour l'apprentissage avec un F-score minimum de 73 % sur tous les cas, ou encore, 85 % pour 7 des 9 étiquettes d'erreur. Quant à **Na-P13** (fusion de **Nantes** et **Paris-13**), les `Erreurs de Bâtiment` se sont encore avérées être les plus difficiles à prédire avec des F-scores qui peuvent descendre jusqu'à 45 %.

Conclusion et perspectives

Nous avons proposé une nouvelle méthode de qualification de modèles 3D de bâtiments. Il repose sur une taxonomie hiérarchisée d'erreurs sémantiques indépendantes des modèles[2] à qualifier. Ce nouveau cadre a été appliqué au cas de la modélisation urbaine aérienne, où les caractéristiques sont extraites d'images aériennes THR et d'un MNS. Un jeu de données entièrement annoté contenant 3 235 modèles de bâtiments reconstruits par voie aérienne avec une grande diversité et provenant de trois zones distinctes a été utilisé pour évaluer notre méthode. Les données de télédétection externes permettent d'extraire des caractéristiques optiques RVB et MNS multimodales. Bien qu'ils soient atténués par rapport aux erreurs sous-représentées, les résultats sont satisfaisants dans les cas bien équilibrés. En outre, avec le bon choix du classifieur ainsi que la configuration

[2] Indifférence à leur scéne d'origine et la méthode de modélisation.

de caractéristiques avancés, nous pouvons obtenir de meilleurs résultats que ceux obtenus avec les caractéristiques de base. Plus important encore, nous avons prouvé que la composition de la scène urbaine affecte grandement la détection des erreurs. En fait, les scores de certaines prédictions ne sont pas seulement stables, lorsqu'ils se forment sur une scène urbaine différente, ils sont même plus performants lorsqu'ils apprennent sur la même scène. Nous avons aussi remarqué comment, pour un ensemble de données de composition hétérogènes, la taille de l'ensemble de formation n'a pratiquement aucun effet puisque les résultats des tests restent stables pour toutes les erreurs. Cela démontre que le cadre proposé peut être facilement adapté avec un bon choix d'échantillons pour l'entraînement et avec peu de données générées manuellement. Cela répond exactement à la question soulevée, contrairement à la littérature actuelle. Nous pensons que notre approche est suffisamment robuste pour évaluer les zones invisibles. Il représente également une base solide pour une correction interactive ou automatique ultérieure du modèle de construction en 3D.

1 Introduction

Contents

The goal of this introduction is to acquaint the reader with concepts that are manipulated in this study. In Section 1.1, we illustrate what is meant by urban 3D models, and specifically, 3D building models. The role of this chapter is also to motivate the need for a semantic evaluation of such models. Indeed, in Section 1.2, we address different aspects of building 3D model inspection and the issues it raises. We conclude by stating our contributions (cf. Section 1.3) and announcing the structure of the thesis (cf. Section 1.4).

1.1 Urban 3D reconstruction

Object 3D reconstruction is a wide research field that interests Photogrammetry, Computer Vision and Computer Graphics communities. Starting from some input sensor data[3] (Light Detection And Ranging (LiDAR) point clouds or multiview stereo images), the goal is to model the 3D surface that bounds an object of interest. In Geographic Information Science (GIS), 3D data is instrumental to model the Earth surface at different spatial scales. In particular, this field takes a special interest in urban areas. Objects present in such scenes usually obey some specific rules which constrain their shape. In this section, we discuss applications of urban 3D models (cf. Section 1.1.1). We afterwards tackle the subject of 3D building modeling and the issues it raises (cf. Section 1.1.2). This section ends in Section 1.1.3 with a list of some of the important technical challenges that are still to be overcome in this domain.

1.1.1 Applications of urban 3D models

In the following, we present a brief survey of urban 3D models applications. A more comprehensive study was presented in (Biljecki et al., 2015b). The goal is to persuade the reader of the relevance of urban 3D modeling and how much affected everybody can be. Indeed, urban 3D models answer to various needs: administrative, environmental, scientific and societal.

Administration related challenges.

Administrative bodies need to document land usage in an effort to efficiently manage territories. Cadastre is a tool that helps track ownership and usage of a each land parcel. Cadastral records are produced and maintained to define fiscal policies. Cadastral data, since conception, came in the form of two dimensional (2D) maps (Billen et al., 2003). Although many urban planning norms were always formulated taking into account height or depth information (Brasebin et al., 2018), the need for a 3D cadastral management was made relevant with the advent of complex architectural features (Biljecki et al., 2015b). To illustrate this case, we show, in Figure 1.1, how 2D cadastral maps are insufficient when it comes to modeling overhangs.

Urban planning related challenges.

Urbanization raises some serious issues that need to be adressed. Consequently, decision makers and all stakeholders in general need to have adequate tools at their disposal. 3D models are, in this sense, suitable as shown hereafter.

While 3D cadastre describes an urban scene at a certain time, urban planners need to access the same kind of information but in a later time. In fact, they need to know in advance the impact of proposed urban norms on the evolution of their zone of interest. One of the issues they study is urban sprawl (Ludlow, 2006). The goal is to limit, as far as possible, land occupation (Tannier et al., 2012) and predict the shape of the urban scene (Brasebin et al., 2018) by tuning local urban plans. This can be achieved, for instance, through simulation, at different scales, based on a formal description of urban planning norms as shown by the work of Colomb et al. (2017). In reality, the 2D city footprint is not sufficient to assess its capacity to contain people. One has to compute the number of habitable units, which depends on the height of buildings. This motivates the

[3] which usually are unstructured.

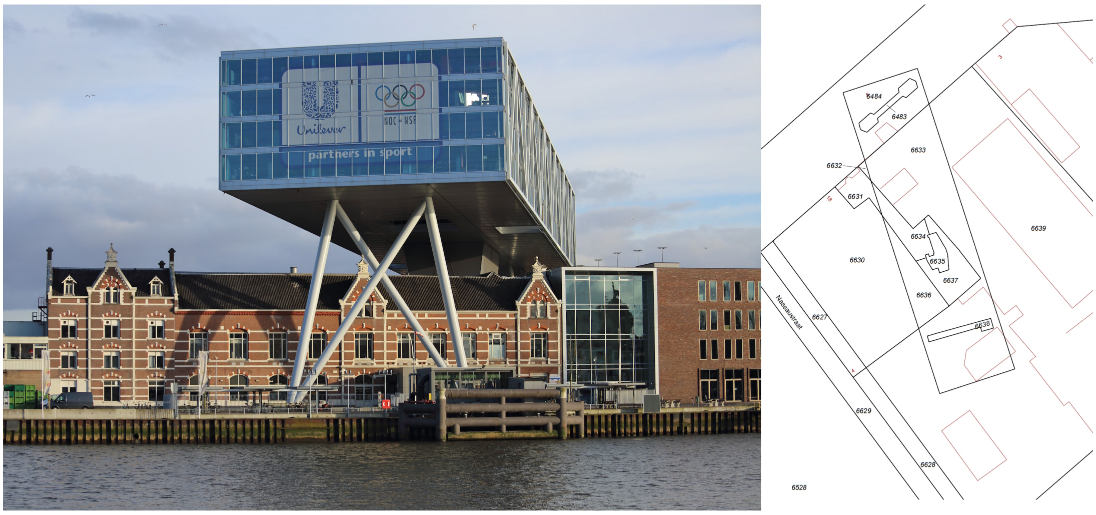


Figure 1.1: Example of a problematic situation where 3D cadastre would be of help and 2D maps fail. The two building footprints overlap and are too complex to be represented in 2D. Image taken from (Biljecki et al., 2015b).

need of 3D data as shown in Figure 1.2. The vizualization (cf. Figure 1.3) and simulation of urban zones can be equally helpful for public consultation (Wu et al., 2010).

Related to urban planning, 3D models could be used as a reference for other planners. These models could be used to describe the flow of vehicles and pedestrians in an urban environment as illustrated in (Vanhoey et al., 2017). In addition, it could be used in physical simulations for urban applications. For instance, communication companies need to have a 3D model of urban scenes to predict signal propagation in the goal of optimal network planning (Yun et al., 2007). It can be also useful for flood simulation. Predicting the height of overflowing water inherently requires 3D information. This is the usecase of Varduhn et al. (2015), where 3D models are used to assess the flood risk. Such information could be instrumental for evacuation planning or for insurance managers. In the same direction, one can simulate fire propagation (Dimitropoulos et al., 2010) or estimate noise propagation in urban scenes (Stoter et al., 2008) (cf. Figure 1.4).

All these simulation derived information could be supplied to decision makers in order to better plan the cities of tomorrow (Huck et al., 2019).

Environmental challenges.

Environment preservation is a of utter importance in the forecoming years. Urban settlement are one of the largest energy consumers. A more efficient energy utilization is necessary to sustain the frantic growth of urban areas. This motivates the need to quantify the energy consumption of urban settlements (Wate et al., 2015) or retrofitting costs (Previtali et al., 2014). Biljecki et al. (2015a) also use 3D models of buildings in order to predict solar irradiation. In fact, solar potential estimation (cf. Figure 1.5) can be useful for assessing the benefits of expensive solar panels projects. This kind of studies can also be applied in urban planning as the simulations could be undergone for future urban developments.

Another critical environmental issue that affects cities is air pollution. Indeed, it has serious implications on human health as demonstrated in (Pascal et al., 2013) and (Chen et al., 2013). In order to understand its dynamics, researchers simulate the local air flow (i.e., the city microclimate) using computational fluid dynamics. This requires a detailed

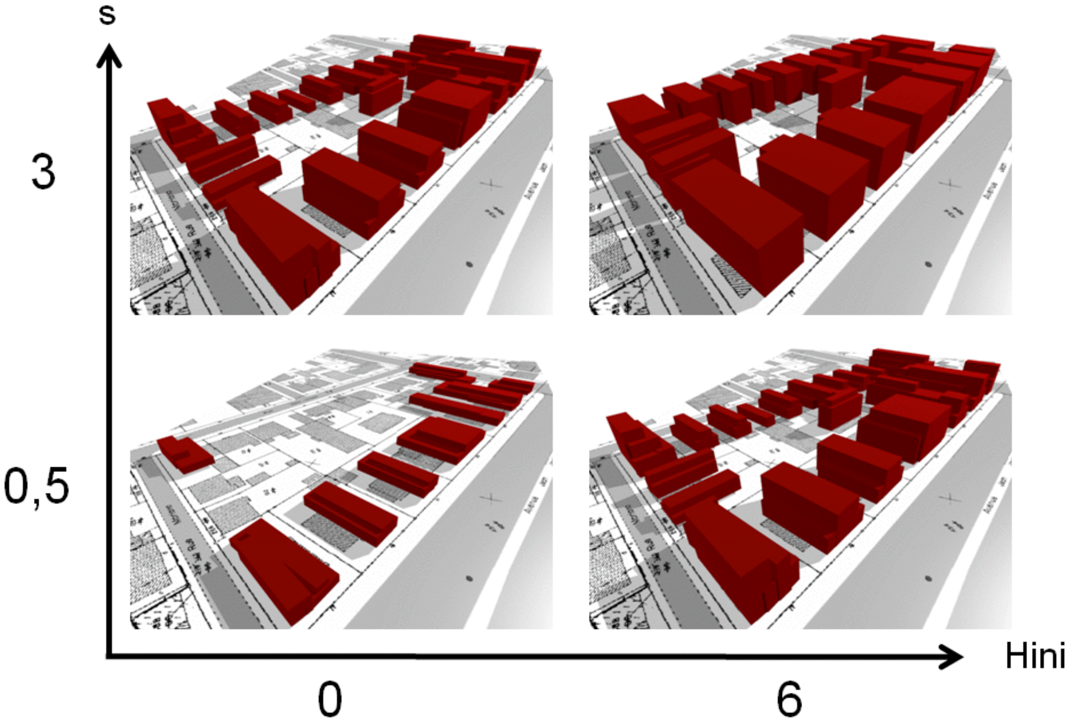


Figure 1.2: Example of 3D urban scene simulated based on known urban norms. Results obtained using the SimPLU3D tool developed by Brasebin et al. (2017). Different parameter values (building surface s and initial height $Hini$) result in different urban scenery. Urban planners can assess the projected urban density of future districts for better decision making. Image taken from the SimPLU3D website (Brasebin et al., 2014–2019).

knowledge of the scene layout. One way to acquire this information is through 3D models of urban settlements (Ujang et al., 2013).

Autonomous navigation related challenges.

Autonomous navigation has seen a great technological leap in recent years. Localization is an important step in navigation (Bonin-Font et al., 2008). 3D models play an important role in visual localization (Biljecki et al., 2015b) and (Piasco et al., 2018).
The basic idea is to match an image to a known 3D model of the city which can be textured or not (Cham et al., 2010; Ardeshir et al., 2014; Arth et al., 2015; Christie et al., 2016) as shown in Figure 1.6.

Once the image matched, one can retrieve an absolute 6-Degree of Freedom (DoF) pose estimation. This is especially helpful in urban canyons as shown in (Piasco et al., 2018).
The same ideas can be applied also for indoor environments, such as social cues aware robots navigating alongside humans or industrial grade robots (Gupta et al., 2018).

Entertainement related challenges.

3D models also appeals to various agents in the entertainement industry. One of the first examples that comes to mind is the video games community (Watson et al., 2008). In their search for realism, in order to engage as many customers as possible, studios reproduce entire cities as a virtual playing ground for the game story. We can cite the ”Spider-Man” virtual New-York city (Gilbert, 2018; Plante, 2013) or the realistic facsimile of Seattle in ”Infamous Second Son” (McWhertor, 2013) as instances of such use.

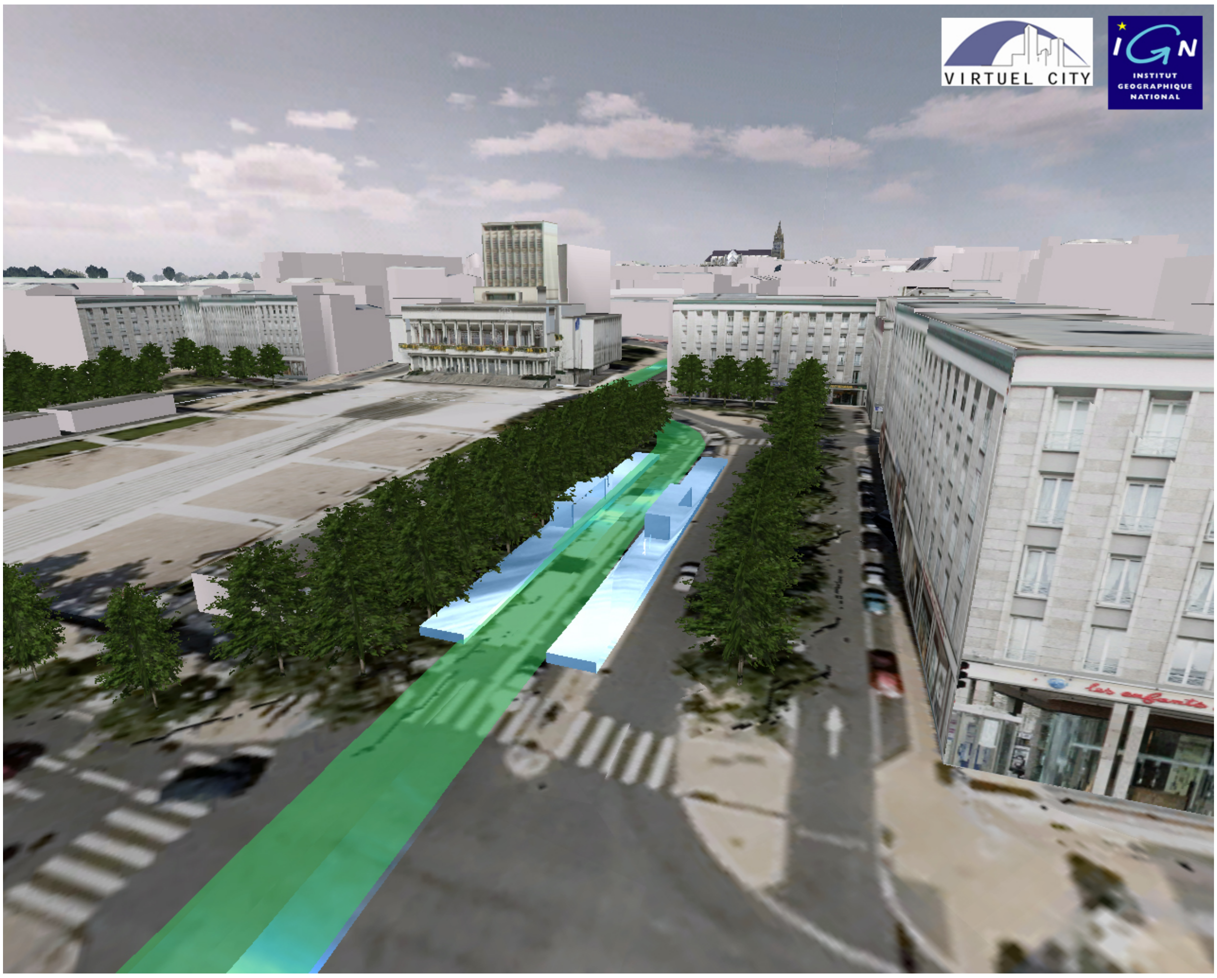


Figure 1.3: Example of the use of city 3D models in public consultation. This image was produced using the Institut National de l'Information Géographique et Forestière (IGN) Bati3D® models of Brest, France. The goal is to simulate the impact of the tramway line to be built on the urban landscape.

Tourism can also benefit from such models. In fact, virtual touring has become more attainable with works like (Koutsoudis et al., 2007). For instance, tourists can use such tools to prepare their trip by familiarizing themselves with the city they intend to visit. This can be possible through mixed or augmented reality as shown by Devaux et al. (2018) (cf. Figure 1.7a). It can also be employed in the service of art. Actually, Russell et al. (2011) and Aubry et al. (2014) illustrated how it is possible to align paintings or photographs with 3D scenes (cf. Figure 1.7b).

This last work could be also used to help marketers sell living units. Indeed, customers can, for example, visit a digitally reconstructed apartment and virtually furnish it (Kim et al., 2019). Another application is living unit pricing. In fact, one would not have to travel to the asset location in order to assess it. Markerters can do so using its 3D model. For instance, one of the determining factors in estimating buildings is the façade visibility. The latter can be simply measured using building models, as shown by Albrecht et al. (2013).

Security related challenges.

Security and emergency fields are not the exception, when it comes to the utilization of city 3D models (Kwan et al., 2005; Rüppel et al., 2011). For instance, Chen et al.

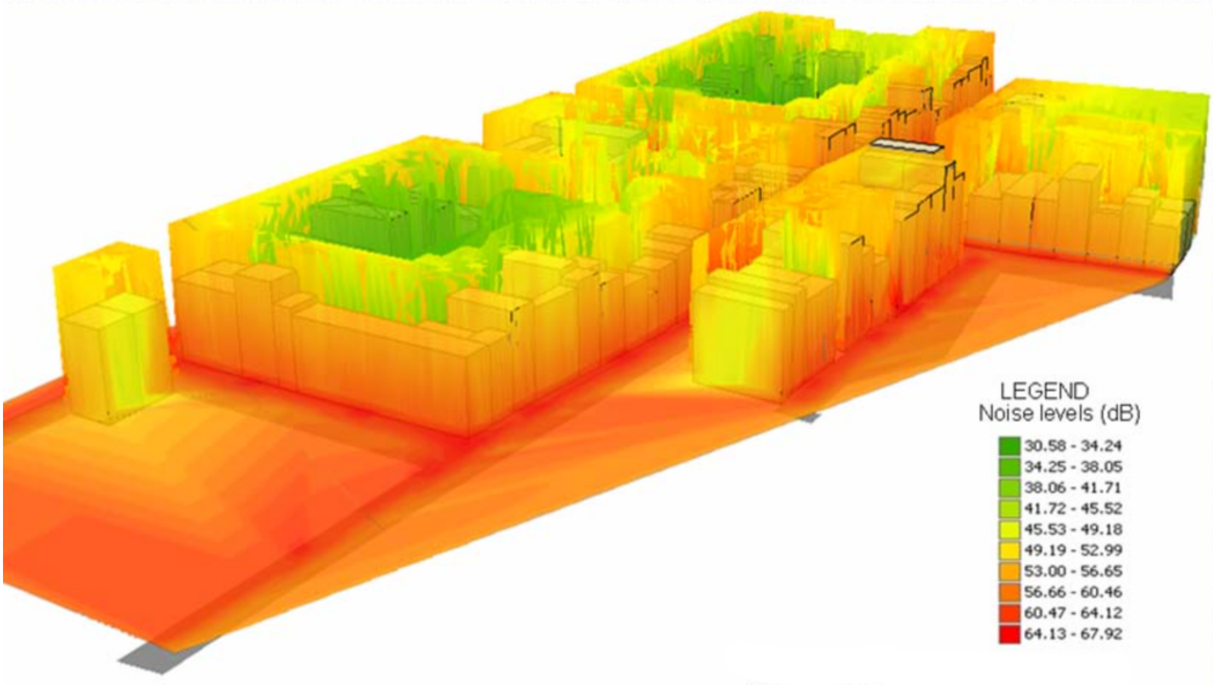


Figure 1.4: Example of noise propagation simulation study using 3D city models. Image taken from (Kurakula et al., 2007).

(2014) shows how these models are used for ladder trucks optimal deployement planning by firefighters. 3D city models can also be used to determine safe margins in the case of bomb disposal operations. This can be possible through explosion simulation in the urban environment of interest (Willenborg, 2015).

Security forces can equally benefit from city 3D modeling. 3D models can be used to help analysing crime scenes (Wolff et al., 2009), as well as it can also be helpful for crime prevention as proved by Wolff et al. (2008). These models could be instrumental in military applications (Zlatanova et al., 2002; Budroni et al., 2010). Military forces could, indeed, use building 3D model based augmented reality to train for intervention scenarii, such as hostage rescue operations.

1.1.2 3D building modeling

We have seen, previously, how city 3D models can be instrumental. They have a large range of applications in entertainement, industry, security, urbanization and sustainable development. In this work, the focus is put on outdoor, rather than indoor, modeling.

Precisely, we will raise, herein, the issue of **large-scale** outdoor city modeling. In this manuscript, by large scale, we mean that the number of urban objects that are studied is large[6]. We will see how building reconstruction has a prominent role in the field. Afterwhat, we will discuss different building model acquisition techniques and how these influence the quality of the resulting models. We end with an examination of semantics in building models.

Large-scale outdoor city modeling.

Not all urban items have the same importance. This is particularly true when modeling city scapes. For this purpose, all different urban elements are analysed based on tem-

[6] Usually, a mid-size city contains around 10,000 buildings.

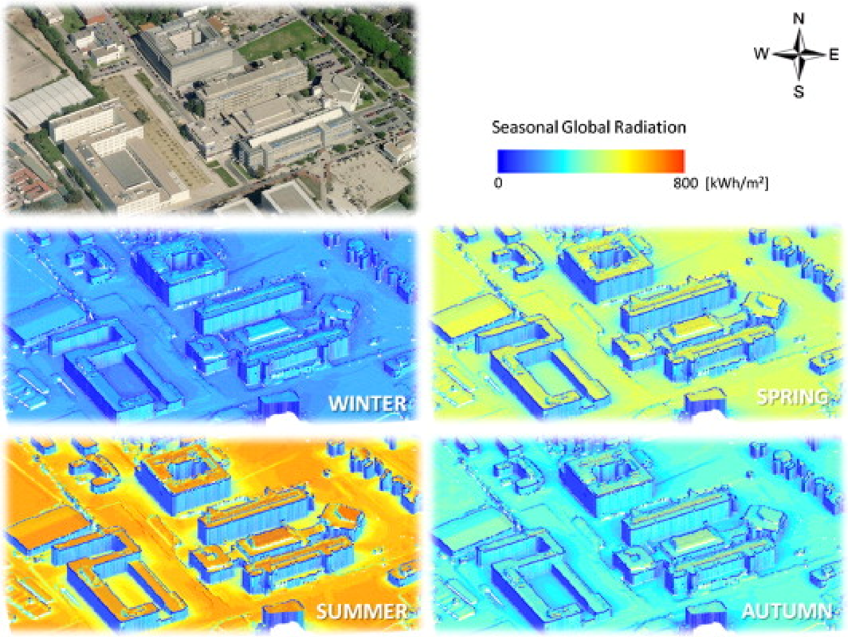


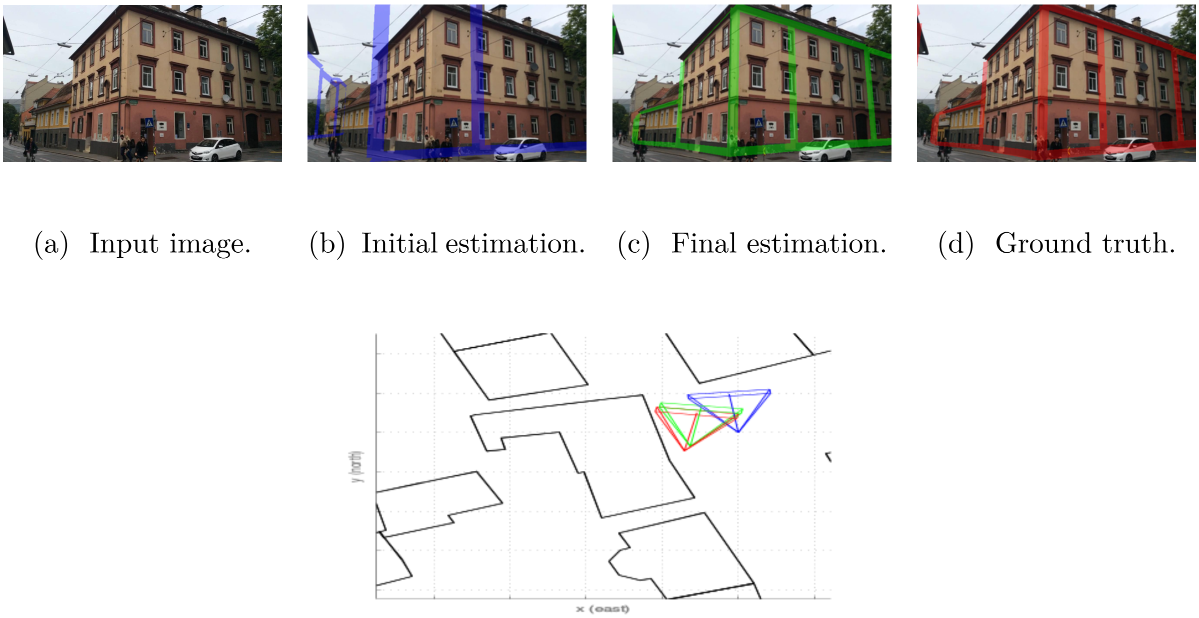
Figure 1.5: Solar potential estimation using 3D building models. The slope, orientation and dimensions of roofs are determinant factors in computing solar irradiation. Image taken from (Redweik et al., 2013).

poral and spatial dimensions. First, urban elements are clustered into different groups depending on how fast they undergo change. 3D modeling is discussed depending on this mentioned categorization. Afterwhat, the spatial differentiation will be used to explain how buildings concentrate most of the interest in urban 3D modeling.

Urban environments are temporally dynamic in nature (Vanhoey et al., 2017). However, constituent items do not evolve with uniform speed. Therefore, urban objects are distinguished depending on their change rate. Pedestrians — as well as all living animals in general — and transportation vehicles are in perpetual movement. In urban scenes, water bodies and vegetation evolve with an annual or seasonal period. Last comes city furniture, roads, bridges, buildings and terrain, which have a much lower change frequency. In this section, we proceed to study how each of these three groups are modeled in 3D.

Besides technical difficulties, there are ethical and legal issues when reconstructing 3D models of humans and vehicles. Indeed, accurate reconstruction involves person indentification. This has proven to be an intricate subject, as shown by Thornton (2010) and Tavani (2011). Adding to the previous discussion about the high temporal frequency of such objects, seeking the most faithfull models proves to be superfluous. In fact, one can populate city models by generic Computer aided design (CAD) models of these humans (Shao et al., 2007) and vehicles. Even more so, a lot of aspects of human/vehicle and city interactions do not require 3D modeling. For instance, Løvås (1994) can simulate pedestrian traffic flow, which is inherently a 2D problem[7], without requiring 3D human models.
There is little or no work, as far as we know, that discusses 3D modeling water body situated in urban areas. Vegetation can be modeled accurately using LiDAR acquisition (Omasa et al., 2006). It is, however, too demanding in ressources and unnecessary in a large-scale context. Trees are usually modeled by template matching, such as the ellipsoidal form model in (Lafarge et al., 2012), or by generic, species dependent, CAD

[7] We can safely assume that human do not fly in 2019.



(a) Input image. (b) Initial estimation. (c) Final estimation. (d) Ground truth.

(e) Different poses.

Figure 1.6: Position estimation that relies on urban 2.5D models. The initial pose is derived from a sensor measurement that is not reliable. In Figures 1.6b (*resp.* 1.6c and 1.6d), the 2.5D model is projected on the image using the initial pose (*resp.* the refined pose and the ground truth one). The 2.5D model helps refine the camera pose. It is shown in the last graph (cf. Figure 1.6e), where the final estimation (green) is closer to the ground truth (red) than the initial one (blue). Images taken from (Armagan et al., 2017).

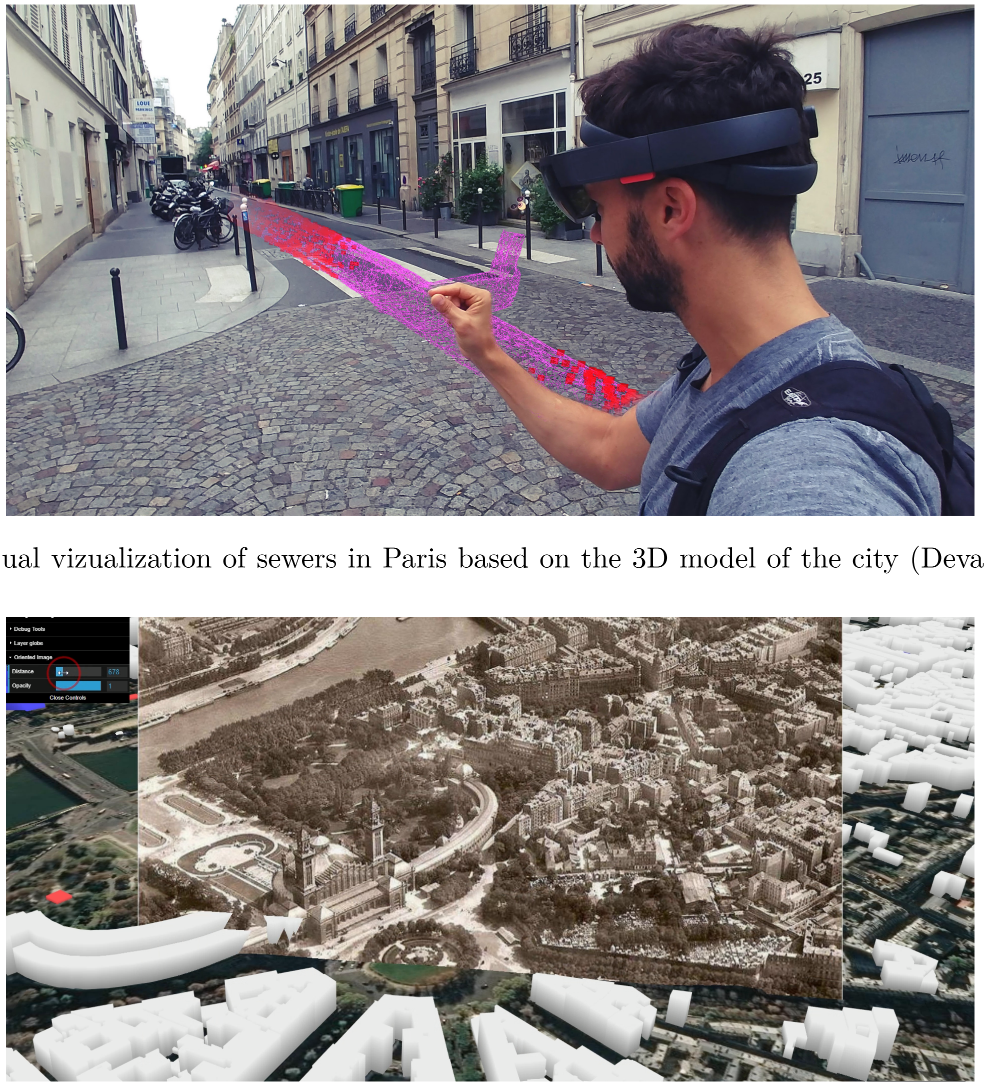
models like in (Iovan et al., 2008) (cf. Figure 1.9).

Regarding the first and second groups, we have shown the lack of motivation for precise 3D reconstruction, in a large-scale setting. Exhibiting less temporal volatility, precise modeling of items from the third group seems to be easier. In fact, terrain relief can be modeled simply from a Digital Surface Model (DSM) or a Digital Terrain Model (DTM). Although not being so easy to detect (Mnih et al., 2010), roads could be naturally modeled using simple planar structures. On the other hand, city furniture, bridges and buildings are more complex. While detailed accurate models are needed for buildings and bridges, they are not necessary for city furniture. Indeed, it is, for instance, pointless to model each single road sign. One would only need to detect its class and reconstruct it, accordingly, using a generic CAD model portaying the same meaning (Soheilian et al., 2013) (cf. Figure 1.10).

Based on the previous temporal frequency differentiation analysis, we narrowed down the urban elements that need special consideration in modeling to two: buildings and bridges. Actually, we can rule out the latter, this time, owing to a spatial frequency study. In terms of land cover, the most present objects in an urban environment are roads, vegetation and buildings (OECD.stat, 2020). Hence, modeling these three object types becomes crucial in order to obtain a viable urban 3D model. We have seen previously how roads and vegetation could be satisfactorily reconstructed using relatively simple models. This is, however, not the case of buildings. That is why, out of all urban features, buildings seem to attract the most attention in urban 3D modeling.

[6] iTowns: `http://www.itowns-project.org`.

[7] Alexandre Devaux website: `https://umrlastig.github.io/adevaux_homepage/`.



(a) Virtual vizualization of sewers in Paris based on the 3D model of the city (Devaux et al., 2018).

(b) Pose estimation of a historical aerial image of Trocadero square (Paris, France) that was registered (Harrach et al., 2019) to a city 3D model visualized with iTowns[4] (Devaux et al., 2012).

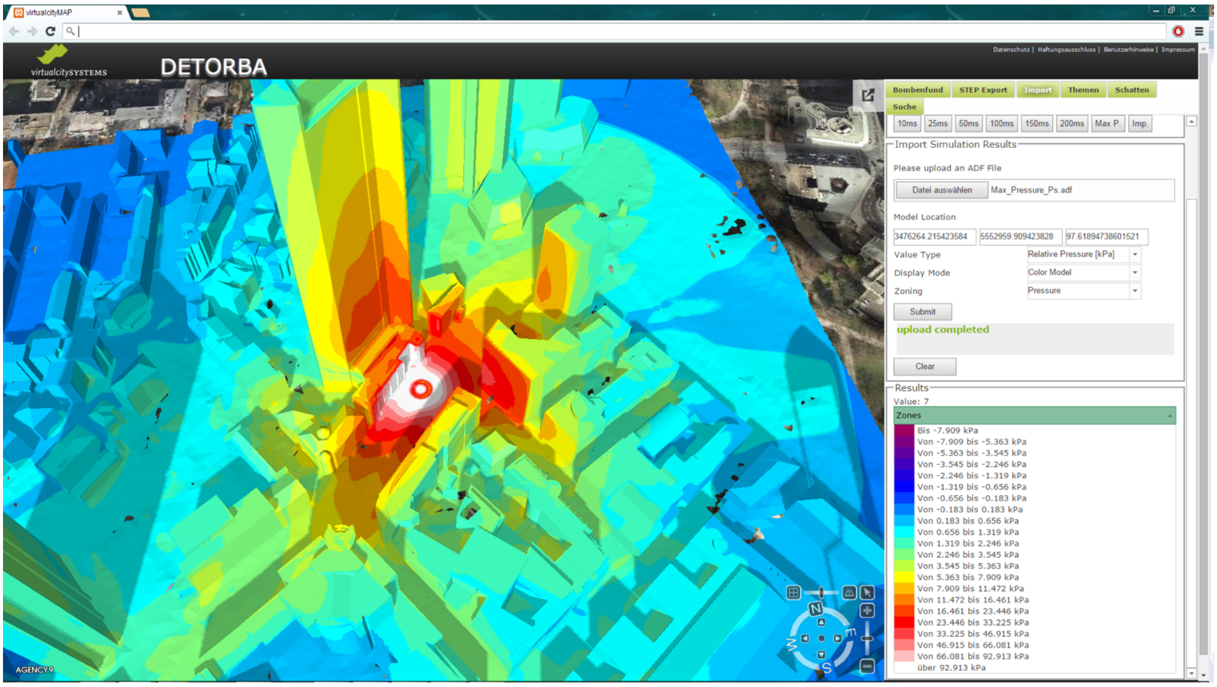
Figure 1.7: Examples of 3D model applications in the entertainement field. Images are courtesy of Alexandre Devaux[5].

3D building modeling.

Before discussing further on the different types of building models, a definition is provided first. A building 3D model is a cartographic product which represents the surface of the building in question. The latter is a generalization of the reality whose goal is not to represent all the details meticulously. However, the model should not part from the real geometry of the building of interest. Thus, geometric fidelity is weighted against generalization. The right compromise is chosen based the final user needs. This issue will intervene often in this work.

Discussed herein are types of building models. Scale of reconstruction is a main factor in dividing the latter into two classes (cf. Figure 1.11). At a small scale, Building information model (BIM) or CAD models are easy to use. On the contrary, Geographic Information Science (GIS) models are more suitable at a large-scale.

BIM models are volumetric in nature: each element is represented by a volumetric primitive. These models are bottom-up: every information about the building, from the



Import Simulation Results

Please upload an ADF File

Datei auswählen Max_Pressure_Ps.adf

Model Location

3476264.215423584 5552959.909423828 97.61894738601521

Value Type: Relative Pressure [kPa]

Display Mode: Color Model

Zoning: Pressure

upload completed

Results

Value: 7

Zones

- Bis -7.909 kPa
- Von -7.909 bis -5.363 kPa
- Von -5.363 bis -3.545 kPa
- Von -3.545 bis -2.246 kPa
- Von -2.246 bis -1.319 kPa
- Von -1.319 bis -0.656 kPa
- Von -0.656 bis -0.183 kPa
- Von -0.183 bis 0.183 kPa
- Von 0.183 bis 0.656 kPa
- Von 0.656 bis 1.319 kPa
- Von 1.319 bis 2.246 kPa
- Von 2.246 bis 3.545 kPa
- Von 3.545 bis 5.363 kPa
- Von 5.363 bis 7.909 kPa
- Von 7.909 bis 11.472 kPa
- Von 11.472 bis 16.461 kPa
- Von 16.461 bis 23.446 kPa
- Von 23.446 bis 33.225 kPa
- Von 33.225 bis 46.915 kPa
- Von 46.915 bis 66.081 kPa
- Von 66.081 bis 92.913 kPa
- über 92.913 kPa

Figure 1.8: Explosion simulation in urban environments using 3D building models. Image taken from (Biljecki et al., 2015b).

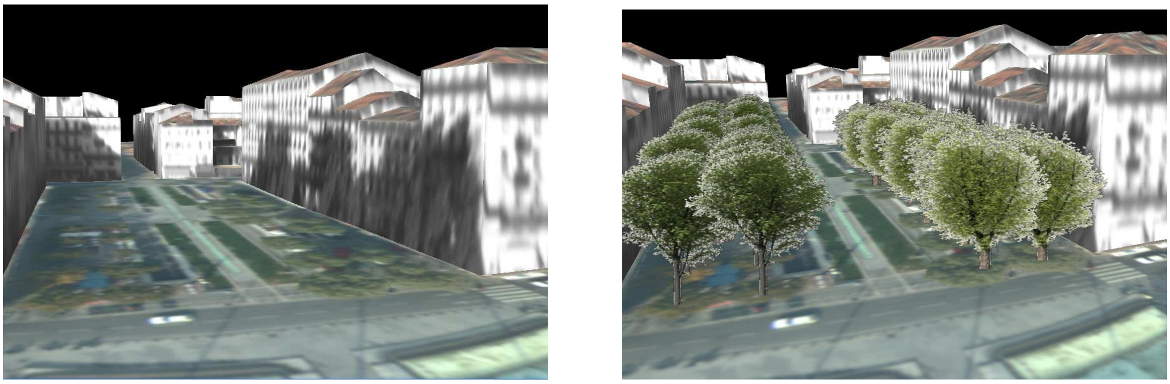


Figure 1.9: The city model is populated with generic models of trees depending on their class. Images taken from (Iovan et al., 2008).

finest details to the most general ones, are aggregated to form the building model. A BIM model is manually constructed as a blueprint of a building before being constructed. It has then to follow the buildings evolution in time until its destruction. This is the most detailed available virtual representation of a building.
However, it does not come without its own issues. First, we can see that it rules out all buildings that predate this convention, especially, historical buildings. Secondly, since this type of models require a high interaction with experts, in order to follow the state of the real buildings, it would be almost impossible to avoid that the model diverges from the reality (Pătrăucean et al., 2015). Third and final, geometric issues, like self-intersections and non 2-manifoldness, affect these models as most BIM tools do not perform geometric sanity checks.

In contrast, GIS models represent the surface of buildings. The goal is to describe the building geometry, as well as all urban items, at a large-scale. Furthermore, this model type carries semantics related to all urban objects in the scene as well as their relations. These models could be acquired manually, as with BIM, like the example of *Ref3DNat®:*

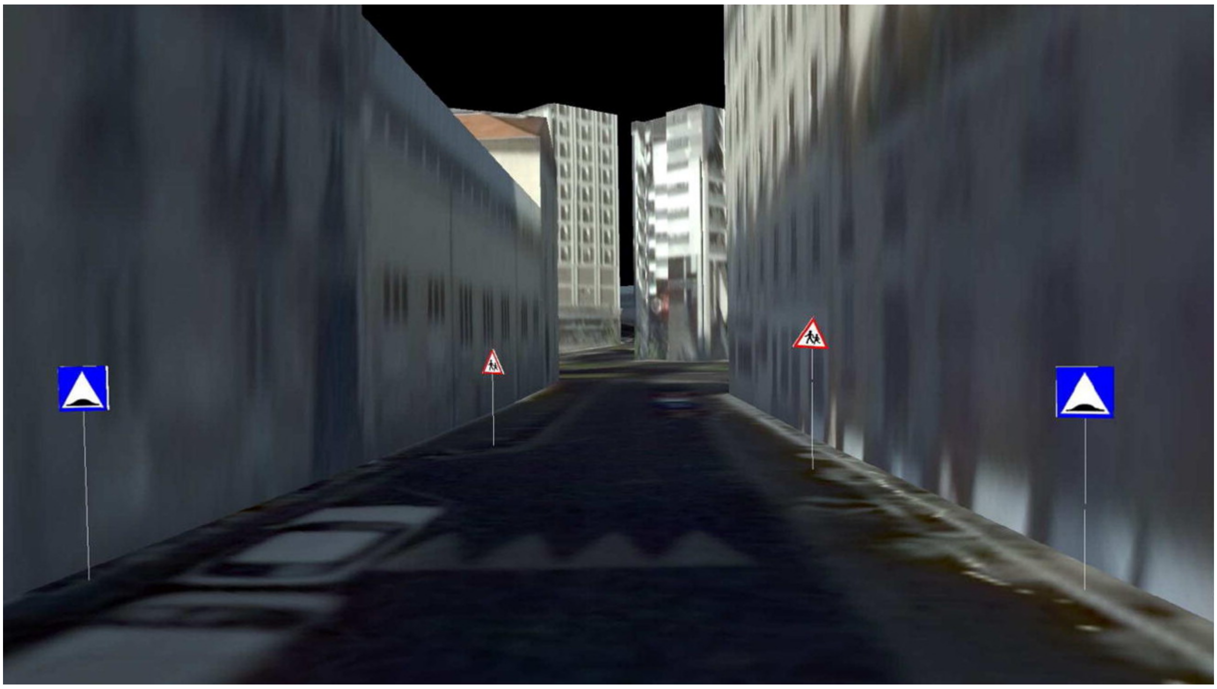


Figure 1.10: A city model populated with detected road signs using CAD models. Image taken from (Soheilian et al., 2013).

Spécifications du noyau du référentiel (2017). Another way to proceed consists in acquiring the geometry, automatically or interactively, using sensor data (Musialski et al., 2013). Crowdsourcing could be equally used for large-scale reconstruction of buildings as depicted in (Uden et al., 2013).

Some works tackled the issue of bridging the two model types (Deng et al., 2016). This field is far from being fully mature as proved by Stoter et al. (2018). In addition to the geometric inconsistancies that emanate from BIM and GIS model mapping, Stoter et al. (2018) show how semantic ambiguities thwart the automatic conversion from one format to the other.

In this thesis, the scalability[8] of 3D modeling is a main concern. As a consequence, a special focus is given to automatic and, in a lesser degree, interactive 3D building modeling techniques.

Building 3D models meets semantics.

The surface geometry is not sufficient to describe urban objects (Biljecki et al., 2016b), and buildings in particular. Semantics are integral part of their representation. City models record, for instance, the function of each architectural feature. Other pieces of information can be attached to each feature depending on its use.
This kind of information is what is referred to as *explicit* semantics in this manuscript. The latter has a significant effect on the geometry of the model. In fact, architectural elements correspond usually to one or a composite of simple geometric shapes, that are mostly planar (Kolbe et al., 2005). As a consequence, dense geometric information (i.e., a dense 3D mesh) is not necessarily the most accurate representation. This effect is denoted herein as *implicit* semantics. As a consequence, knowing the function of an object helps representing its geometry more efficiently. For instance, modeling a column using a cylin-

[8] The capacity of the modeling method to scale at district and even ciy level.

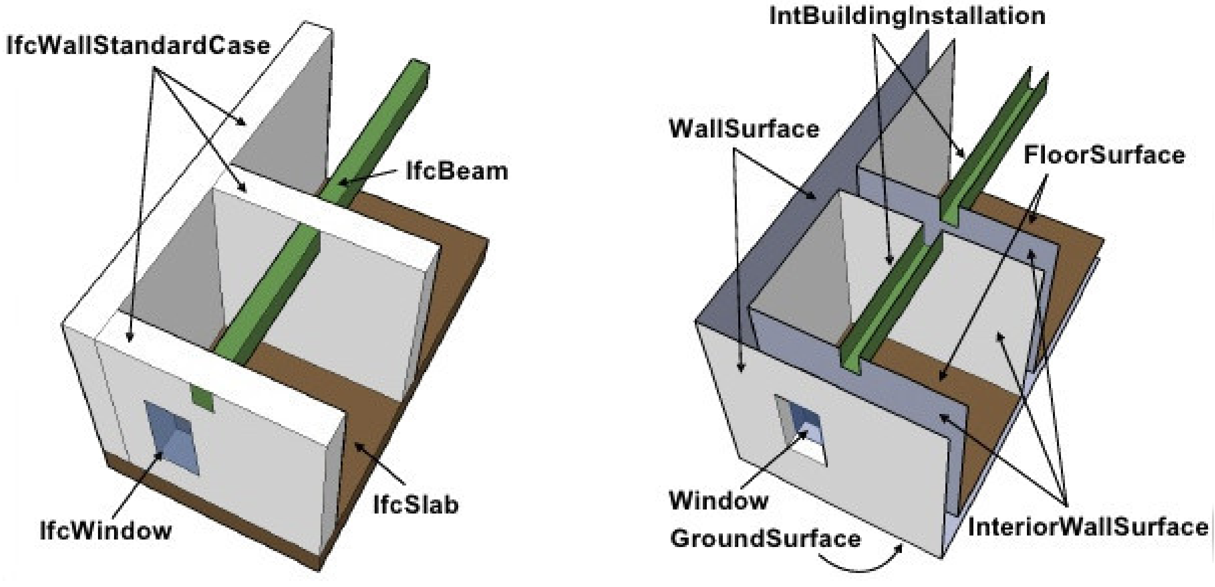


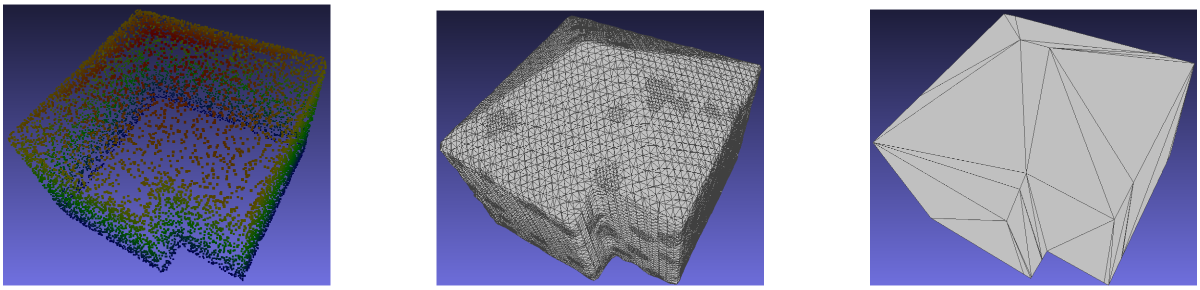
Figure 1.11: Industry Foundation Classes (IFC) and CityGML representations of the same object (building storey). BIM (left hand side) is an inherently volumetric model while GIS (right hand side) represent the surface of the building. Image taken from (Nagel et al., 2009).

der and estimating its radius and height is more accurate than using a non-parametric representation of the same object at a given resolution, while using much less information. Hence, semantics implies compaction in the geometric representation of buildings. That is why the last criterion was used, for example, in addition to the RMSE, as an evaluation metric in (Lafarge et al., 2012).
As a consequence, we distinguish, from now on, between a *3D model* and a *3D mesh* of a building. While the latter accounts for the geometric precision, the other conveys, in addition, semantic properties. This is illustrated in Figure 1.12.

As argued previously, compaction, which results from semantics, is a primordial characteristic of 3D building models. It implies a discretization in generalization levels. Indeed, Gröger et al. (2012b) use this property to formalize a discrete and intuitive Level of Detail (LoD) scale. Even though the original LoD specification was far from being mature it is widely used in the GIS and Computer Vision communities (Rau et al., 2006; Biljecki et al., 2014). A detailed study of the issue was conducted in Biljecki et al. (2014) and Biljecki et al. (2016b). This work will content itself with the simple intuitive definition of LoDs.
A LoD-0 model corresponds to the 2.5D footprint of the building. Next, the LoD-1 consists of the LoD-0 footprint extruded up to a uniform height. LoD-2 enhances the previous model scale with more geometrically accurate roof structures. The LoD-3 reveals even more details, as it models small superstructures, as well as openings. Last comes LoD-4 which conveys indoor details that are ignored in this work. Figure 1.13 depicts these definitions.

1.1.3 Challenges in 3D building modeling

The subject of 3D building modeling has been widely studied in more than twenty years. Still, there are some unsolved issues in the field (Musialski et al., 2013; Lafarge, 2015).



(a) 3D point cloud sampled from the surface of a building (5000 points).

(b) 3D mesh of a building surface (28,876 triangles).

(c) 3D model of a building (50 faces).

Figure 1.12: Difference between a 3D point cloud, mesh and model of a building. The point cloud (with 5000 points) does not provide any topological informations about the surface of the building. A 3D model is a special case of 3D meshes which describes the surface of the object in a compact way. The 3D mesh is poor in *implicit* semantics compared to the 3D model, for which each architectural feature corresponds to a single geometric primitive.

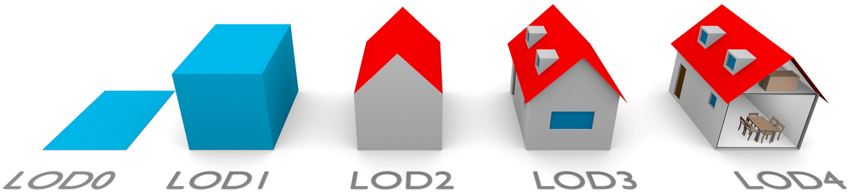


Figure 1.13: LoD categorization used in this work. LoD-4 is ignored herein. Image taken from (Biljecki et al., 2016b).

Here is presented the same classification as (Lafarge, 2015).

Data acquisition.

Related to sensor data acquisition are some serious issues, in signal processing in general, just as in 3D modeling in particular. In fact, noise is an integral part of physical measurement processes. One should take good care in avoiding error propagation through the processing pipelines. For instance, outliers in point-of-interest detection can render a photogrammetrically constructed mesh accumulate sizable geometric errors.

Missing data is also an important issue in 3D, as some background objects in the scene could easily be occluded by objects in the foreground. One way of dealing with this type of obstacles is to multiply the data acquisition settings. This can, actually, help mitigate not only occlusion problems, but also noise interference.

However, too much data heterogeneity can also hinder the 3D modeling process. In fact, with more accessible data acquired using different sensors (optical cameras, Radio Detection And Ranging (RaDAR) and LiDAR, for instance), in various settings (aerial, satellite or terrestrial) and conditions (rainy, sunny, foggy, night-time, day-time,...), other hurdles need to be overcome. More data does not always mean more knowledge, as demonstrated in (Brachmann et al., 2018). In fact, one should take good care in choosing how to fuse their input data (Kedzierski et al., 2014). For instance, the latter needs be coregistered in the same referential (Monnier et al., 2013; Mezian et al., 2016) (cf. Figure 1.14). There can also be variabilities in radiometry that needs to be taken into account (Yan et al., 2012).

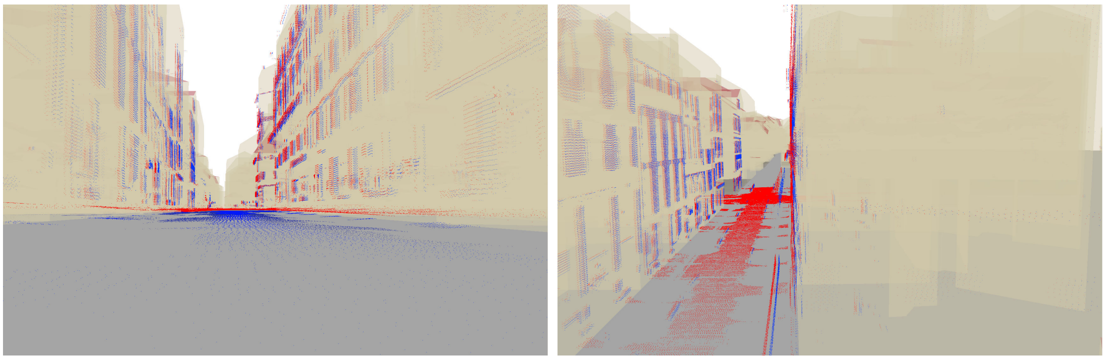


Figure 1.14: Terrestrial LiDAR point clouds are registred against 3D building models. The registred point clouds can be used to enrich the 3D models with detailed reconstructions of façades. The raw point cloud is represented in red while the registred one is illustrated in blue. Image taken from (Monnier, 2014).

Modeling automation.

The most accurate approach to produce semantically aware and geometrically accurate representations of a building is a manual one. It can be either based on high accuracy *in situ* geodetic measurements (errors are of the order of 0.05 m (Kaartinen et al., 2005)). Although, it is the most accurate representation possible, it is an arduous and expensive process. Stereo-plotting consists on using couples of overlapping images in order to manually determine the 3D geometry of lines. It can be used in building modeling to manually plot building edges in 3D and hence extract its surface. This reduces the complexity of the manual modeling process. It produces, however, lower precision models as errors are of the order of 0.5 m (Jamet et al., 1995). This is still a slow process that is highly demanding in terms of operator expertise (Rüther et al., 2002).
The manual labour can be alleviated, partly, by automating some parts of the 3D surface aquisition pipeline (Musialski et al., 2013). Some interactive approaches have been proposed to model building. The operator is still needed for highly complex semantic tasks (Mayunga et al., 2005; Castellazzi et al., 2015). Such methods suffer from imprecision compared to the fully manual ones: for instance, Mayunga et al. (2005) produce models with a geometric error standard deviation in the order of 1 m.
Full automation is still unattainable, especially when it comes to semantics. Different strategies are used, where each one targets a specific resolution, balancing between compaction and geometric accuracy. Ordered from the most to the least compact, these strategies are listed herein. In a Manhattan-world setting, one assumes that buildings are collections of boxes (Vanegas et al., 2010; Li et al., 2016). Lafarge et al. (2012) and Nan et al. (2017) rely on the hypothesis that building are made of piecewise-planar primitives. Rich grammars can give rise to less compact but more accurate models, as is the case of Demir et al. (2015) and Zeng et al. (2018). Mesh simplification strategies comes last in terms of compaction (Zhou et al., 2010; Verdie et al., 2015). A sketch of this comparison is presented in Figure 1.15.

The compromise, between compaction and geometric precision, relies on an *a priori* knowledge of the modeled urban scene. Such hypotheses do not naturally hold at large-scales. Multiple factors play a role in architectural styles of buildings that do not always go hand in hand: geographic proximity does not imply always a temporal or architectural closeness for buildings as shown later in Figure 5.5. At a regional or continental scale, the differences become even more overt. As a result, some authors choose to alternate

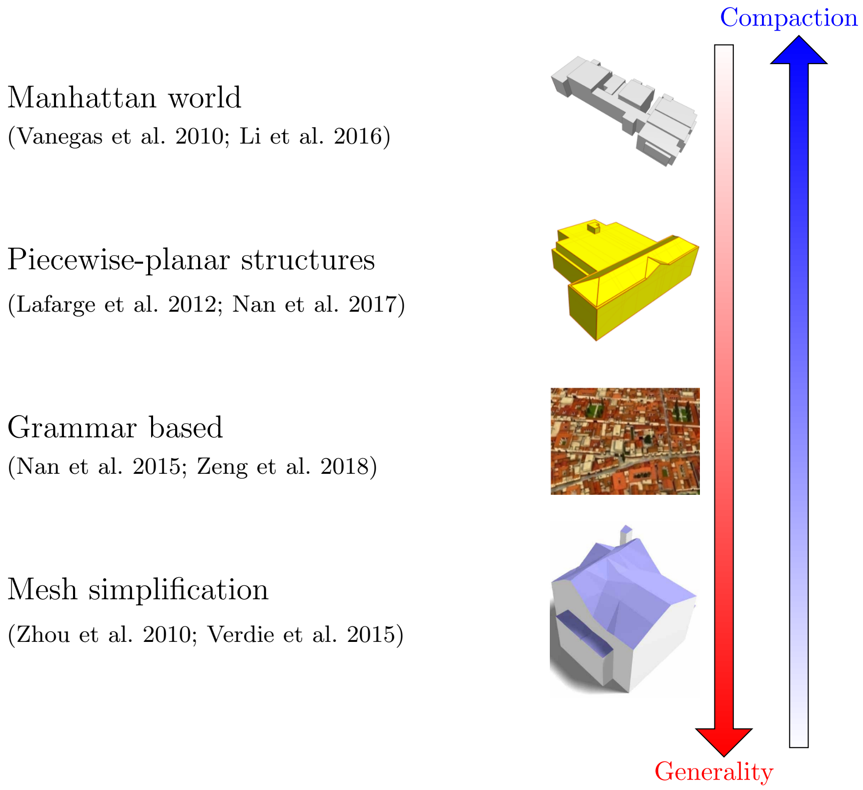

Compaction
Manhattan world
(Vanegas et al. 2010; Li et al. 2016)
Piecewise-planar structures
(Lafarge et al. 2012; Nan et al. 2017)
Grammar based
(Nan et al. 2015; Zeng et al. 2018)
Mesh simplification
(Zhou et al. 2010; Verdie et al. 2015)
Generality

Figure 1.15: Modeling strategies and the targeted compaction and geometric accuracy. Depending on the final use of the model, a compromise is chosen between the model compaction and its geometric accuracy.

automatic and interactive methods (Musialski et al., 2013): an automatic reconstruction followed by a interactive correction step.

Quality evaluation.

As discussed, semantics play a prominent role in 3D building models. Naturally, it should also be taken into account in the evaluation process. It is easier said than done. Usually, it is indirectly checked using the compaction criterion (Lafarge et al., 2012).
Up to now most studies try to evaluate the quality of modeling algorithms by assessing the accuracy of the geometry of a handfull of buildings. These evaluation approaches rely on purely geometric metrics (usually the RMSE) to compare the reconstructed building to a reference models. In a large-scale setting, this becomes prohibitive. In fact, it means that one should reconstruct, at least, a whole district to be able to judge a city model. This is expensive in ressources and time. Alternatively, many works favor visual inspection of models. This requires, in average, $2\,\mathrm{h}/\mathrm{km}^2$/expert, and in some cases, is time-consuming as stereo-plotting the building in the first place.
In the next section, we will discuss, in details, the various ways of evaluating a building model.

1.2 Evaluating building models

We have just seen how building model quality evaluation is one the main challenges in the field. In this section, we present the different types of quality assessement a building model can undergo. First, in Section 1.2.1, it is described how models could be checked for their topological consistency. On the other hand, the 3D model geometry has also to be checked against the real building geometry. This issue is visited in Section 1.2.2. Manual geometric inspection is discussed before considering automatic evaluation and its challenges.

1.2.1 Topological consistency inspection

Extensive work has been accomplished in order to achieve a standardized representation of city 3D models. This has resulted in the Open Geospatial Consortium (OGC) CityGML standard (Gröger et al., 2012a). However, in practice, it is not always respected, as shown in (Biljecki et al., 2016a), where up to 89 % of models were found to be topologically invalid.

In consequence, the subject of automatic inspection of the topological consistency of city models has drawn a strong attention in the GIS community. The basic common principle is the two-manifoldness of surface representations (Gröger et al., 2011), which aims at excluding self-intersections. It is however not sufficient for complex buildings. An urban object, in the international standards is, in reality, represented as an agregation of two-manifold surfaces, as shown in (Gröger et al., 2011) and (Ledoux, 2013). This kind of structures can be simply modeled as a 3D Linear Cell Complex (LCC) (Damiand et al., 2014)[9], as demonstrated in Diakité et al. (2014). This leads to the use of 3D LCCs to check the consistency of city models (Gorszczyk et al., 2016). The idea relies, however, on the presence of reference data to compare to. While being very useful for format conversion, sanity checks, as intended in the original paper, are pointless when inspecting 3D models in the wild.
Not all efforts did fully explore the topological possiblities that are provided by the standard (Ledoux, 2013; Biljecki et al., 2016a). In fact, most authors assume that polygons and solids in the representation are not allowed to have holes, like the work of Gröger et al. (2011) and Alam et al. (2014). This was, however, the case of Ledoux (2013). Their work built on the axioms of Gröger et al. (2011): it does not only deal with polygons and surfaces, but also takes care of solids and allows holes of different dimensions (polygons with holes, surfaces with boundaries[10] and volumes with cavity). The errors are organized in increasing order of the geometric dimension of the object they affect. The process stops at the dimension of the first detected inconsistancies and ignores the higher ones. An open source library and a web-application are publicly available (Ledoux, 2018), in order to further the sanity of exchanged 3D models of urban scenes.

We should also mention the efforts made to fix the invalid models. These methods could be divided into local and global ones.

Local approaches: they rely on local topological operators to solve detected issues in the model. One such method is CityDoctor (Alam et al., 2014). On the downside, these methods have the tendency to introduce more errors in the hope of solving present ones. In fact, the problem is often ill-posed, and multiple solutions can be suggested to alleviate the same problem.

Global approaches: the idea is to represent models as volumes constrained by the defectuous surfaces. The goal here, as in urban reconstruction, is to infer topological information from the observed geometric properties, taking into account the unreliability of the data as well as *a priori* information on the model. This is the case of (Zhao et al., 2013), which relies on a tetrahydralization of the input model volume

[9] An nD LCC is an embedding of combinatorial maps in $\mathbb{R}^n$.

[10] This is actually taken into account by the 2.8D models in (Gröger et al., 2011).

and a heuristic carving of unnecessary 3-simplices. This approach, in fact, resembles the surface reconstruction procedures from unstructured point clouds using Delaunay triangulation (Cazals et al., 2006; Berger et al., 2014).

1.2.2 Geometric inspection

Guarantying topological consistency is a necessary criterion for the quality of city 3D models. It is, however, not sufficient. One would want to assess how close is the building geometry to reality. This is what we denote by geometric quality evaluation.

This issue has attracted a lot of attention in 2D (Mooney et al., 2010), but is not as popular in 3D. This is may be due to the lesser attention that is given to the latter. This can be explained by the different levels of difficulty when producing these data. It is indeed more difficult to reconstruct a 3D model of a building than detecting their footprints. For example, one can take a look in the number of submission for each task of the proposed International Society for Photogrammetry and Remote Sensing (ISPRS) benchmark (Rottensteiner et al., 2012; Rottensteiner et al., 2014). The 3D reconstruction task received less submissions than the other one.

Two ways can be used to judge the quality of the geometric representation of a building. Manual evaluation relies on human interaction to determine how close the model is to reality. The automatic approach relies only on the model and other external data to do so without involving a human in the loop. Both are discussed herein.

Manual evalutation.

Manual inspection involves a human operator checking the validity of the reconstructed geometry compared to a reference data. The latter could be classified into two types:

Reference 3D building models: These are high quality models that were manually produced. One approach is to acquire these on the field by topographic survey. This is, nevertheless, very expensive. Another alternative is to use stereoplotted models using oriented images. This is a more affordable and scalable but a less accurate solution.

Reference sensor data: They are more available in constrast with the earlier alternative. Oriented images, Digital Surface Model (DSM) and point clouds are instances of such data. This setting is, with the same or less positional accuracy as the previous case, the easiest to adopt in a large-scale.

Manual inspection is, actually, ideal in the sense that humans are naturally most suited to detect semantic flaws in building representation. The latter are, however, not so adapted for quantitative comparisons. One way to alleviate such difficulties is to provide software tools to help measuring inaccuracies in geometry.

In contrast, the most pressing issue is scalability. Manual inspection is indeed a laborious task that requires some kind of expertise in the field, for a reasonable efficiency. Humans are also not so infallible when assigned precise repetitive jobs. Human involvement in the reconstruction effort is especially one of the reasons behind topological inconsistancies in 3D models.

Automatic evalutation.

In order to achieve scalability, the get-go solution is automation, or failing that, semi-automation. As usual, it is easier said than done.

Methods have been proposed to automatically assess building model geometry. They rely, however, on ratios that describe the global quality of the model compared to a reference data. The flaws are twofold in this case. First, global ratios do not capture the finer details in the building. Secondly, reference 3D models are not cheap to acquire, as discussed earlier. While the first issue is a manageable issue, as proven in (Rottensteiner et al., 2012), the second is not so easy to mitigate.

The most crucial problem is, in constrast, the semantic aspect of the quality assessement. As discussed, the models in question possess high semantic properties that have a significant effect on their geometry. Assessing one without the other is fundamentally unsound. While this is easy to achieve manually, it has yet to be incorporated in automatic settings.

1.3 Contributions

Based on all the previous discussions, we adopted a research direction that has rarely been taken up to now, as far as we know. Our aim is to evaluate the quality of 3D building models automatically and at large scales.

Herein is explained the path that led to this subject. To do so, the context of this work is explained in detail in Section 1.3.1. Further on, promising and conceivable utilizations are stated in Section 1.3.2. This is before ending with a summary of the main contributions of this work (cf. Section 1.3.3).

1.3.1 Positioning

As seen in Section 1.2, geometric evaluation of 3D building models remains a field open for investigation. Topologic consistency inspection, in contrast, as discussed in Section 1.2.1, has received ample attention from the GIS community. In consequence, it will not be of the focus in this work.

Building modeling has been shown, earlier (cf. Section 1.1.2), to be a bottleneck for urban outdoor modeling at large-scales. As a consequence, evaluating 3D building models is essential for entire scene model inspection. Semantics play a prominent role for building models (cf. Section 1.1.2). Geometric fidelity metrics are sufficient in order to faithfully capture the structural properties induced by semantics. As a consequence, the proposed approach should factor semantics in the evaluation as well. As a result, like with LoDs, errors are expected to be categorized owing to the semantic character of the evaluation.

The evaluation is, herein, studied under two constraints: **large-scale** and **automation**. Below, consequences of each one are discussed in detail.

Large-scale.

The implications of this constraints are two-pronged. Scalability is at odds with both the heterogeneity of building models and the reference data used for evaluation. Both aspects are discussed herein.

Due to the **large-scale** requirement, the proposed categorization must be stable no matter how the treated urban scene change. The same robustness is pursued when considering the different 3D modeling approaches and the heterogenuous sensors behind the evaluated building models. As a consequence, the latter come in in all possible LoDs and are vulnerable to particular defects, depending on the used modeling method and the input data type. Having each urban zone or building model type with its own specific error definitions would hinder scaling the defect categorization. In fact, this would involve frequent human interactions in order to alleviate possible issues when merging possibly conflicting error classifications. Such interventions are always expensive and scale poorly at district, city or country level, let alone the global stage.

Quality evaluation involves the use of reference data against which reconstructed building models are compared. In Section 2.2.2, are discussed the different forms these data can take. A high quality reference usually requires manual labour that is evidently not scalable. Actually, the less costly the reference data is, the more adaptive the evaluation method will be at large scales. In fact, the best alternative would be, if possible, to avoid using reference data altogether.

Automation.

Semantically aware inspection is a natural task for human operators. It is, however, challenging to automate. In fact, as seen in Computer Vision, the more semantic tasks are, the more difficult they are to automate. It is even more arduous taking into account the **large-scale** constraint which implies a robustness to the natural heterogeneity in urban scenes as well as the variance of modeling approches.

1.3.2 Application cases

Ahead was discussed the context of the proposed work. Hereafter is discussed the potential use of a semantically aware automatic geometry evaluation of 3D building models.

Change detection.

This is possible in two ways. First, one can consider the same approach that is used for error detection as a method for change detection in 3D building models. This is equivalent to considering change as an semantic error that impacts the geometry of the building. Second is the fact that change can be implicitly detected from geometric errors. In fact, the change that can be observed from sensor data will render the outdated model invalid geometrically. Figure 1.16 depicts how 3D change detection in urban areas could be achieved.

Building model correction.

This is the most obvious usecase for this work. In practice, errors are detected by operators in order to correct them. The first task, if automated, can be time saving in the correction post-processing step. It is even possible to automate the whole correction process using the advances achieved in interactive reconstruction (Kowdle et al., 2011) (cf. Figure 1.17).

Reconstruction method selection.

Evaluation of the building models can help selecting the most adapted reconstruction methods for a certain urban scene. Indeed, depending on the needs of the potential final user, some errors are to be watched more than others. As an example, an insurance agent who is only interested in flood simulations will not be interested in the geometric accuracy

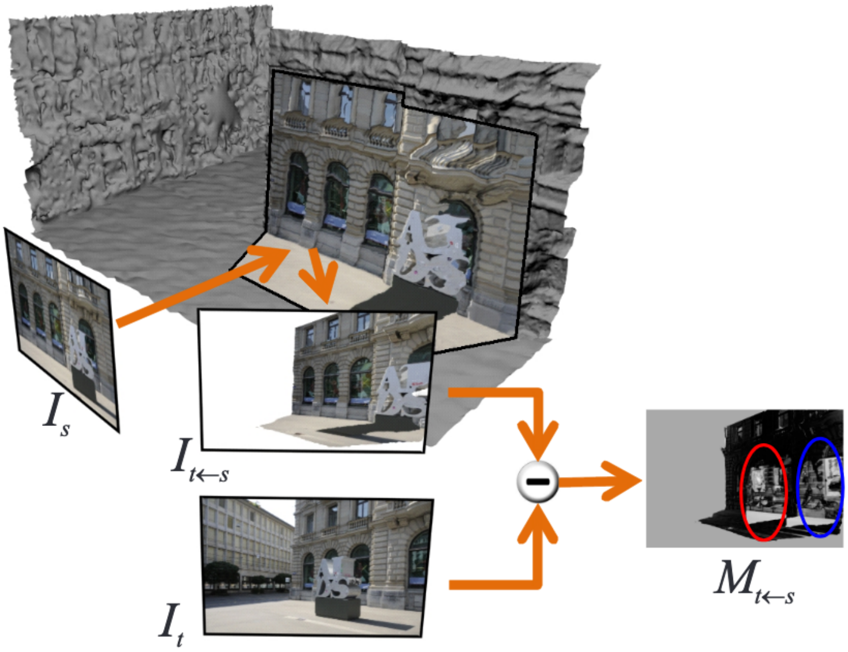


Figure 1.16: Illustration of change detection in urban areas. Image taken from (Taneja et al., 2013).

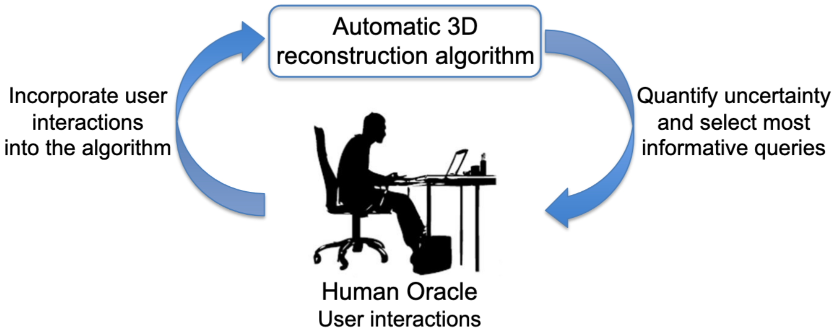

Automatic 3D reconstruction algorithm
Quantify uncertainty and select most informative queries
Incorporate user interactions into the algorithm
Human Oracle
User interactions

Figure 1.17: Example of an interactive pipeline that needs human operators to provide correct initial reconstructions. Image taken from (Kowdle et al., 2011).

of LoD-2 features and will focus principally at the LoD-1 representation errors. Modeling algorithms are naturally biased towards a given setting that depends on the hypotheses chosen by their creators. They will produce less errors when those conjectures are met and fail otherwise. These assumptions are fixed, for instance, in terms of building types (Haussmann style, Manhattan-world ...), geometric criterea (planar surfaces, symmetry ...). Hence, choosing a modeling approach can determine how good are the models and vice versa. In Figure 1.18, one can see how different methods could results in different building models.

Crowdsourcing evaluation.

Crowdsourcing (Kovashka et al., 2016) can be seen as a building modeling method. It has become easier with the help of some online tools like SketchUp[11] (cf. Figure 1.19), where anyone can model, for instance, their home and share it publicly. The quality of the models depends on their authors. An automatic evaluation method can help to check that the uploaded models respect the specifications. Another issue is the presence of

[11] SketchUp: https://www.sketchup.com.

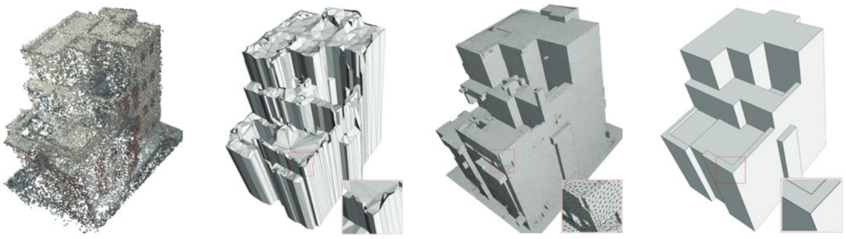
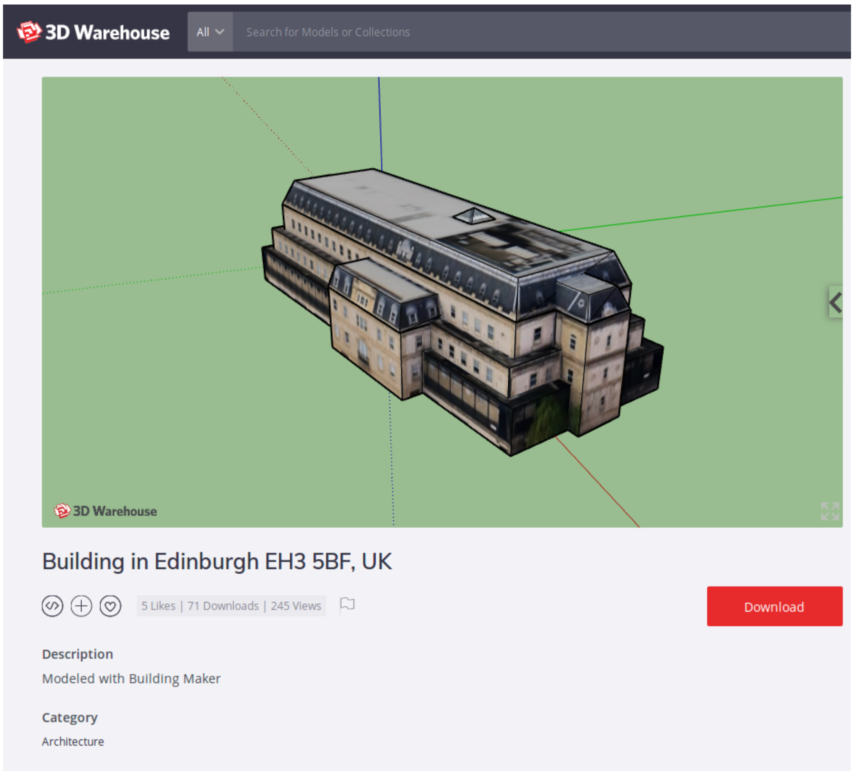


Figure 1.18: Depiction of different models for the same building. Image taken from (Li et al., 2016).

vandalism in open data (Neis et al., 2012). The presented approach can be used to help understanding user behavior and detect vandalisers.



3D Warehouse
All
Search for Models or Collections
3D Warehouse
5 Likes | 71 Downloads | 245 Views
Download

Building in Edinburgh EH3 5BF, UK

Description
Modeled with Building Maker

Category
Architecture

Figure 1.19: A sample of a building model made by a SketchUp user.

1.3.3 Main contributions

Based on the detailed discussion hereinabove, we present our contributions in the field of geometric evaluation of 3D building models. They are four-fold. A graphical abstract is presented in Figure 1.20 giving an overview of our contributions.

Error taxonomy.

A hierarchical and adaptive taxonomy of errors is proposed. It is designed to be independent from the urban scene or the modeling approach. In order to achieve this, errors are chosen meticulously as a compromise between generalization and expressivity. The first ensures that errors from the taxonomy can fit any scene, while the second implies that errors are not equivocal. It is also an adaptive categorization, since, depending on the final user needs, a set of errors can be extracted after specifying some parameters.

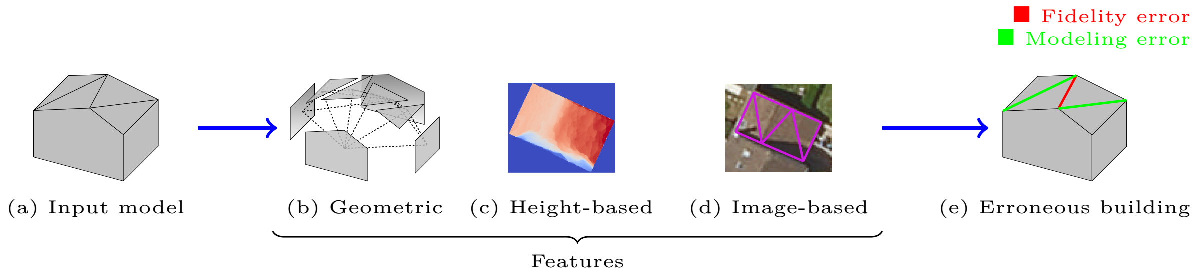


Figure 1.20: Our semantic evaluation framework for 3D building models (a). Semantic errors affecting the building are predicted using a supervised classifier and handcrafted features. In addition to the input model topological structure (b), features are extracted from Very High resolution overhead data. It can be based on a comparison with the DSM (c). Optical images can also be used through, for instance, local gradient extraction (d). Several errors can be detected at the same time, in a hierarchical manner (e). Fidelity errors correspond to geometrical imprecision as shown in red. On the other hand, modeling errors denote morphological inconsistancies with the real object.

3D model reference free evaluation.

Acquiring 3D reference models is very expensive in ressources and particularly not scalable at large-scale. As a consequence, the problem is formulated as a supervised learning one. Based on the errors that are extracted from the taxonomy, a classifier is trained in order to predict defects for unseen models.

Custom features.

Since the formulated problem has not been massively studied, there was no feature extraction baseline to compare to. Hence, a baseline of features is presented in this work. In addition, we also presented some more advanced ones relying on ScatNets and graph kernels, with the aim of improving on the detection rates of the baseline approach. They are computed based on the geometric attributes of the building model and the comparison to external remote sensing data: Very High Resolution (VHR) optical images and height maps.

Scalability analysis.

One of the goals of this work is to achieve scalability in large-scale. This implies the transferability of the learned predictors to unseen urban settings or untested modeling techniques. It involves also a representativeness and generalization study of the training sets. An analysis is conducted accordingly. We experimentaly prove the stability of classifiers, under some settings.

3D model handling.

We also developed a set of tools to handle the geometry of 3D building models. They were necessary in order to manage poorly stored models. It was also useful in computing exact projections of polyhedral buildings and extracting other useful pieces of information that were later used for feature extraction.

1.4 Structure of the Thesis

The remainder of the thesis is organized in five main chapters.

- Chapter 2 provides a global overview on state-of-the-art techniques in fields related to our main subject. In fact, in order to have a general idea on the 3D building modeling domain, we provide a summary and a broad categorization of techniques that can be found in the litterature. We also provide a detailed account of all quality evaluation methods that are used by the community.
- In Chapter 3, we introduce the first leg of our approach. It consists in defining a general layout of a **large-scale** adaptive error taxonomy. Applied to the overhead setting, and based on observations of automatically modeled buildings, we implement this layout by defining semantic errors to populate the taxonomy.
- **Automation**, which is the second and remaining constraint defined in Section 1.3.1, is handled in Chapter 4. Actually, aiming to automate a semantic task, we introduce a learning based approach where errors are treated as labels which are predicted using a trained supervised classifier. Features are extracted based on the intrinsic properties of building models as well as extrinsic attributes making use of external optical or depth based data.
- This proposed approach is then tested on multiple urban scenes as shown in Chapter 5. The latter provides in depth details about the origin of the evaluated building models as well as the used algorithms. The different baseline features are tested on all urban areas yielding mitigated results which depend on the combination of defined errors and urban scenes. To satisfy the already stated **large-scale** need (cf. Section 1.3.1), a scalability analysis is drawn based on various experiments involving the presented urban zones. Next, the framework is tested at a lower specificity level (cf. Section 3.1).
- Chapter 6 presents a new set of feature extrators for a better evaluation of 3D building models. This involves using graph kernels for intrinsic features and ScatNet for extrinsic ones. We explain, in this chapter, how they are integrated in the proposed workflow.
- In Chapter 7, the previously mentionned 3D building model representation is assessed experimentally. First, the choice of classifiers is also discussed based on exhaustive experiments. Afterwhat, results of a thorough testing of the proposed advanced features, which take into account the structural information that was missed by the baseline, are reported and profusely analysed.
- The final Chapter 8 concludes the thesis by presenting a summary of the proposed approach as well as experimental results to support it. Final thoughts on underdiscussed perspectives of research are also adressed.

2 State-of-the-art

Contents

In this Chapter, a tour of the different notions necessary for the understanding of the present manuscript is presented. It is divided into three sections. First, we categorize, in Section 2.1, the 3D building modeling techniques. A global idea of the subject is required to have a good idea of which defects to expect in evaluated building models. Secondly, we present a complete state-of-the-art survey on quality evaluation for 3D building models (cf. Section 2.2).

2.1 Automatic 3D building modeling

In this work, we study the quality evaluation of **large-scale** urban modeling. Building models at city scale cannot be obtained manually. Automatically acquired building models are, actually, the first target of our study. As a consequence, there is a need for a overview of automatic 3D building modeling. The aim of this section is not to give a thorough survey of all 3D modeling techniques, but rather explain the main approaches and how they affect the quality of the final product. For a comprehensive study of urban reconstruction methods, we refer the reader to the work of Musialski et al. (2013).

Automatic 3D building modeling consists mainly on two steps. Section 2.1.1 presents the different approaches used in the first step: building delineation. Afterwhat, the second step consists in modeling the surface of the building. Section 2.1.2 presents a general categorization of modeling strategies for buildings.

2.1.1 Building extraction

In automatic building modeling, in order to model the building surface out of the input sensor data, the building location must first be determined. There are actually two ways to approach this problem. The first relies on retrieving building footprints out of a GIS database, such as cadastral ones (Taillandier et al., 2004; Durupt et al., 2006; Horna et al., 2007; Ledoux et al., 2011; Biljecki et al., 2017; Biljecki et al., 2019). In constrast, the second approach type extracts building outlines directly from the input sensor data (Poullis et al., 2009; Lafarge et al., 2012; Nguatem et al., 2017; Zhu et al., 2018). Both approaches suffer from some shortcomings.

GIS databases. The first type of approaches relies on the fact that the GIS database is acquired at the same time as the sensor data. Otherwise, there will be no guaranty that the modeled scene has not changed in the meantime. For instance, a building can be removed — as a whole or partially — between the two data acquisitions. This would result in having terrain or vegetation reconstructed as a building part.

Even if the database is guaranteed to be produced at the same time as the input data, there is another issue that must be dealt with. In some datasets, building outlines are actually not stored in their original form: they are instead deformed to fit their specifications. For instance, buildings smaller than a certain area can be ommitted. The specified generalization level can also play a role as small features could be smoothed away. A part from this type of issues, the provided building outlines could be erroneous simply due to a faulty, wether manual or automatic, process.

Classification. The second family of building extraction techniques is based on the classification of input sensor data. As a consequence, the final building delineation quality depends mostly on the input data. Contrarily to the first case, there is no redundancy that can help filter out noise in the data. Low density LiDAR point clouds (with a density less than 10 pts/m^2) are, for example, too sparse and unstructured for instance to detect exact boundaries between objects (Michelin et al., 2012). Conversly, images offer an alternative but, as a passive sensor, can suffer from issues related to shadows (of buildings or other urban objects) that are detrimental to building delineation (Adeline et al., 2013).

The classification process is not flawless either. Just as with GIS databases, classification relies on some *a priori* information that is generally encoded in regularization terms (Lafarge et al., 2008; Zhu et al., 2018; Zeng et al., 2018). These are needed to retrieve outlines of buildings efficiently. However, they can also fail to represent the ground truth faithfully and results in wrong building footprint extraction.

2.1.2 Modeling strategies

Once the building footprint is extracted, the goal is to generalize its 3D surface out of the input data. Several strategies are conducted. They can be classified in three categories: Model-driven, Data-driven and mixed approaches.

Model-driven approaches. This type of methods, also called top-down methods, rely on strong assumptions on the type of buildings to be modeled. For example, the Manhattan-world class assumes that buildings are an aggregation of boxes (Vanegas et al., 2010; Ledoux et al., 2011; Arroyo Ohori et al., 2015; Li et al., 2016). This is actually an instance of grammars utilized in procedural modeling. In the latter, a set of predetermined rules are used to determine the best shapes, out of a library, in order to better fit the input data (Lafarge et al., 2008; Koutsourakis et al., 2009; Zhou et al., 2010; Simon et al., 2011; Mathias et al., 2011; Martinovic et al., 2013; Nan et al., 2015; Demir et al., 2015; Zeng et al., 2018). Top-down approaches are usually used when the input data resolution is not sufficiently accurate to directly retrieve detailed architectural features of the building (Lafarge et al., 2008).

Data-driven approaches. They can also be called bottom-up methods. These modeling techniques need very high resolution sensor data in order to retrieve constituing geometric features of the building. This is usually achieved using primitive extraction. For instance, building surfaces could be viewed as piecewise-planar structures (Taillandier et al., 2004; Chauve et al., 2010; Lafarge et al., 2012; Nan et al., 2017). As a consequence, plane arrangements are drawn from the original data. Edges are also examples of features that can be extracted from the data to guide building modeling (Baillard et al., 1999). A second step consists in aggregating the extracted geometric features into a single surface. This is usually implemented either through a greedy approach (Taillandier et al., 2004) or a global optimization method (Poullis, 2013; Verdie et al., 2015; Zhu et al., 2018; Holzmann et al., 2018).

Mixed approaches. The first type of methods fails to scale for multiple urban scenes as it is contigent on having a comprehensive enough grammar, which is almost never the case. On the other hand, the second kind of methods knows other drawbacks as it mainly tends to oversegment facets in order to stay close to the measured data. One way to alleviate these issues is to mix the two approaches. Werner et al. (2002), for instance, fits predetermined shapes (wedges and rectangles) to already extracted edges and planes. However, these usually rely on greedy approaches that do not scale well.

Limitations of current modeling methods. There are still some limitations that these automatic approaches did not adress. Higher resolution sensors are yielding more and more data with high frequencies that are still not taken into account as most produced models are at most LoD-2 ones. These can pose serious problems in modeling: Brédif et al. (2007) proved that undetected superstructures can lead to bad quality models and their reconstruction can help in that regard. High buildings can cast shadows on, or occlude, neighbooring ones which results in issues with their surface retrieval. This has not been widely studied (Lafarge et al., 2012; Bao et al., 2013), as buildings are moslty modeled one by one and and often not in challenging[12] areas. All these issues induce defects that should be characterized and classified for a better understanding of the field and, subsequently, objectively understand which methods are the most adapted for which areas.

2.2 Quality evaluation of 3D building models

We have seen previously various methods used to automatically model buildings. The goal of this section is to describe the available approaches that evaluate the quality of such models. These could be distinguished based on two criteria.

In Section 2.2.1, are presented the quality evaluation methods based on their output. An alternative perspective to characterize quality evaluation methods relies upon the type of reference data (cf. Section 2.2.2). Based on this survey, we state in details, in Section 2.2.3, how the approach presented in this thesis is different from what was already proposed in the literature.

2.2.1 Output types

We distinguish herein quality evaluation methods depending on the kind of output they produce. Fidelity metrics constitues a first instance of output types. The second is semantic labels. In what follows, we explain, in details, the differences between the two types of method.

Fidelity metrics based methods.

One way to characterize the quality of a building model is to compute indices or metrics reporting its geometric accuracy.

Most metrics provide information on the geometric precision of the model. They are computed for different geometric objects. Below, depending on the geometric dimension of the latter, we categorize the geometric fidelity metrics.

We start with zero dimensional objects: i.e., points. In this case, metrics are based on their coordinates. The goal is to detect positional inaccuraccies (Kaartinen et al., 2005). In constrast, the choice of points to be inspected is not simple. Corner points resulting from the intersection of edges in the model is one choice. In fact, Zeng et al. (2014) registers corner points from the evaluated model and the corresponding reference. Based on this registration, a comparison is drawn using Root Mean Square Error (RMSE), just as in (You et al., 2011; Landes et al., 2012). The same points are used as a proxy for manual quality inspection by Elberink et al. (2011). Another alternative is to sample points from lines or surfaces to be compared. These could be predetermined manually as in (Kaartinen et al., 2005) or sampled regularly as demonstrated by Vögtle et al. (2003) and Tran et al. (2019). Imprecisions are not computed only relying the RMSE, but can also be separated into planimetric (LoD-0) and height inaccuraccies (LoD-1) (Vögtle et al., 2003; Jaynes et al., 2003; Kaartinen et al., 2005).

[12] Like buildings delimited by narrow streets or ones with complex architectures.

Second comes edges and all one dimensional objects in general. These convey structural, in addition to positional, information. Kaartinen et al. (2005) compare lengths as well as slopes of edges formed by reference points. Edge related metrics are also used as an intermediary as shown by Elberink et al. (2011) and Michelin et al. (2013). They are both interested in edges resulting from facet intersections. While the first relies on RMSE, the second computes more complex metrics that compares model edges to ones extracted out of sensor data.

Next are compared surfaces (i.e., two dimensional objects), bounded (for example, polygons) or not (like planes). These hold more structural information than the first ones and hence are widely used for evaluation. Rottensteiner et al. (2014) used height discrepancy of roof planes so as to evaluate building models. This metric is ideal for Manhattan-world model evaluation, as in the case of Zebedin et al. (2008). In addition to height discrepancy, normal displacement is computed using always the same RMSE metric by Henricsson et al. (1997). Conversely, Zeng et al. (2014) use also a RMSE for comparison, but not in the Euclidean space. In fact, after mapping the evaluated and reference models to a sphere, they compare their spherical harmonic (Brechbühler et al., 1995) representations. Just as with edges, planes can be evaluated using angular measurements, as was proposed by Henricsson et al. (1997) and Landes et al. (2012). Another alternative is to compare reconstructed and reference models based on surface area comparisons. These are mostly based on ratios like completeness and correctness[13] (Henricsson et al., 1997; Schuster et al., 2003; Landes et al., 2012; Rottensteiner et al., 2014).

Last, are three dimensional (i.e., volume) evaluation. The same detection ratios that were computed for surfaces are again calculated to evaluate volumes this times, as shown by Jaynes et al. (2003), Mohamed (2013), Zeng et al. (2014), and Nguatem et al. (2017). These are the only metrics used for volumes that we are aware of.

Regarding *implicit* semantics, as far as we are aware, only one metric is widely used to evaluate its impact. As discussed previously in Section 1.1.2, compaction is one byproduct of semantics. As a consequence, it was used as an index to evaluate reconstructions[14]: the more a model was compact the better it was. This is reflected, for instance, in the work of Lafarge et al. (2012), Duan et al. (2016), Zhang et al. (2017), Zeng et al. (2018), and Zhu et al. (2018), where this metric is never used alone but always alongside geometric ones.

Semantic-based methods.

In a drive to provide a quantitative assessment of building models, the previously defined metrics fail to convey specific and localized information about a predetermined building model. These indices are usually used to give a general idea of the quality of models produced by some modeling method. Moreover, they do not usually carry semantics, which is critical for further processing steps such as manual correction (Elberink et al., 2011). Some evaluation methods, in an effort to alleviate these issues, yield semantic labels that describe the errors of an evaluated model. Hereafter, are described the different types of labels that were proposed in the literature.

(Boudet et al., 2006) was the first ever work, that we are aware of, which tackles semantic labels as outputs of the evaluation. This approach has four classes which indicate

[13] In other words, recall and precision, respectively.
[14] It is sometimes called by its antonym: complexity.

how valid the model is: “acceptable” and “correct” portray valid buildings while “false” and “generalized” are refused. It can be seen as a four grade based score system expressing the confidence in a building model. The main limitation of this method is the fact that the proposed labels do not specify what defects a model presents if it is not valid. It is, therefore, hard to use for model correction. Besides, the acceptable defect definitions depend also on each use case. For instance, a communications company would be more adament on the accuracy of roof parts, which would affect wave propagation, rather than an insurance company interested in flood damage estimation. This means that each use case implies a relabeling that could be potentially different from other one. Durupt et al. (2006) used also the same labels to evaluate their reconstruction.

The first hints of a fine grained semantic labeling of building model errors lay in the work of Rottensteiner et al. (2014). This work was the first to report segmentation issues, at facet level, in labels instead of a global ratio. They distinguish between oversegmentation cases, undersegmentation cases and cases where both co-occur.

On the other hand, these mentioned errors are not comprehensive: they do not cover all possible building model errors. Michelin et al. (2013) came up with a richer taxonomy of errors that are categorized into three big families:

Footprint errors: these portray errors relative to the building footprint, which is used by many modeling algorithms as input. Errors contained in this family are: “erroneous outline”, “unexisting building”, “missing inner court” and “inaccurate footprint”.

Reconstruction errors: these are caused by the modeling approach. These defects can be the result of the incompatibility of some *a priori* hypotheses about the scene, for instance. Such errors are: “under-segmentation”, “over-segmentation”, “inaccurate roof” and “Z translation”.

Vegetation errors: this corresponds to a special case when modeled building are occluded, completely or partially, by higher vegetation. It becomes impossible to evaluate properly these models.

Although the last taxonomy is rich, it is not exhaustive enough as it misses cases that are not present in the urban zone that was studied in the paper. For instance, the fact of modeling a cone-shaped roof with a piecewise-planar structure is not taken into account in this taxonomy. Moreover, it adopts the point of view of the modeling method that was used to provide their dataset. Knowing the specific weaknesses of the latter guided the choice of error family classification and the error definitions. This is particularly clear when looking at their error categorization. In fact, the first error class relates to the fact that the used building modeling approach (Durupt et al., 2006) relies on footprints given as input. The second category corresponds to the actual step of building roof structure inference.

2.2.2 Types of reference data

In order to evaluate models, reference data are utilized for comparison. Based on the latter, another way to discriminate among modeling methods is possible. The first class of reference data are high quality ground truth 3D building models. Remote sensing data is the second alternative that is used for reference. Neither of these two choices do influence the type of output the evaluation method produces. Hereafter, is explained the difference between the two categories.

High resolution ground truth.

Ground truth building models are mainly acquired manually. Herein, are listed the different ground truth measurement techniques, in descending order of accuracy.

The most obvious case consists in field measurements of the modeled building. Dick et al. (2004), for instance, evaluated their buildings based on manual tape measurements taken on specific architectural features, like windows and columns, with an accuracy of 0.01 m for the first and 0.1 m for the second. Complete and scalable measurements of buildings are possible using topographic surveys. Using the latter, building models could be reconstructed with precision up to ±0.1 m (Henricsson et al., 1997) or ±0.05 m (Vögtle et al., 2003).
There is also the possibility of manually modeling the building using overhead images, or stereoplotting. Zebedin et al. (2008) use such a method to produce their reference data from aerial images with an accuracy up to the order of ±0.15 m. The same procedure was used also by Jaynes et al. (2003) producing models with inaccuraccies bounded by ±1/3 m.

Remote sensing data.

Reference ground truth models are not readily available at large quantities (Schuster et al., 2003). The previous choice is hard to scale up to district or city levels. In fact, they are more suitable to evaluate few building models: e.g., in order to validate a reconstruction method. Conversely, remote sensing data are more accessible and could be used, instead, as reference. In fact, since these are usually fed as input in modeling methods, it is only reasonable to evaluate the produced models in comparison to the input. Listed here, are the different types of remote sensing data and how they could be used for building model evaluation.

Aerial images. These could be original images or preprocessed orthoimages. The first class of images are the most precise but the other one is more available. Images were used by Michelin et al. (2013) to extract 3D segments based on plane sweeping techniques. Reference edges are filtered out and are used to evaluate the 3D model (intersection) edges. Based on these segments, Boudet et al. (2006) not only evaluate edges but also corners (i.e., edge intersections) and planes. Facets were also evaluated, in (Boudet et al., 2006), based on correlation functions computed from multiple overlapping overhead images. Orthoimages were used also to, for instance, compute Normalized Difference Vegetation Index (NDVI) scores for vegetation occluded building model discrimination.

LiDAR point clouds. This type of data have the inherent advantage, compared to images, of directly providing depth information. Kaartinen et al. (2005) used data that was acquired multiple times with a guaranteed precision up to 0.083 m in planimetry and 0.035 m in height. Out of the latter were chosen reference points that were compared to equivalents in the evaluated building models (check the previous Section). All the points could also be used for comparison by computing metrics such as RMSE (Lafarge et al., 2012; Zhu et al., 2018). Original input data is not always accessible. One other issue arises when using data from multiple sources where it has to be coregistred in the same reference system. This is taken into account by Akca et al. (2010), before addressing the completeness of building models.

Height maps. These are not sensor data as they are a byproduct of earlier data types. Still, they are considered here since they are used as input by some building modeling pipelines. Just like with LiDAR point clouds, DSM is used for building model comparison based on RMSE (Zeng et al., 2018). Michelin et al. (2013), however, use the same data but differently. In fact, DSM, being a result of overhead images, can be used as proxy, to help extracting 3D geometric features instead of recomputing correlation scores like in (Boudet et al., 2006). It can also be valuable for missing court detection by calculating sky viewshed angle scores (Michelin et al., 2013).

2.2.3 Novelty of the proposed method

Quality evaluation approaches presented hereinabove are mostly unsuitable for the stated objectives in Section 1.3.1.

In fact, the semantic character of the evaluation overrules approaches relying only on geometry based metrics. This represents most of the methods discussed earlier (cf. Section 2.2.1). Actually, these metrics could be used once a semantic error is identified to help quantify the defect. Conversely, the need for reasonably available reference data, which is dictated by the **large-scale** constraint, implies the reliance on remote sensing data based methods (cf. Section 2.2.2).

There are only two approaches that are both semantic and rely only on more readily available remote sensing data: (Boudet et al., 2006; Michelin et al., 2013). Both define semantic errors that are predicted with the help of overhead images and DSMs. A classifier learns statistical properties of the 3D building models using features that are derived from these sensor data. The learning process further helps scaling the evaluation pipeline to unseen data by predicting errors using the same attributes.

Our work relies on the same idea, but goes further by allowing the possibility to evaluate building models without using any reference data[15] and relying on their intrinsic attributes instead. It also offers a new taxonomy of errors that is intended to be exhaustive and generalizable. These latter notions are developed in details in Chapter 3, while Chapter 4 presents how features are chosen to represent building models and the learning process.

In fact, we propose a novel hierarchical and modular taxonomy that assembles all possible semantic and geometric errors that can affect the building model. Based on this taxonomy, depending on the end user needs, error labels are extracted for the classification. We also developed a baseline of features, intrinsic as well as extrinsic[16], which describe the evaluated models. Using these features so as to train some regular off-the-shelf classifiers, we proved the feasibility of our approach. Finally, we made use of some advanced feature extractors with the aim of improving the prediction scores of the proposed pipeline.

[15] With the exception of training data.

[16] If available.

3 Semantic evaluation of 3D models

Contents

In this chapter, the first building bricks of our approach are presented: the error definitions. We proceed by reprising the main consequences of the constraints that were imposed for our evaluation approach in subsection 1.3.1. These will determine the desired properties that are seeked in this work.

We establish, in Section 3.1 a general structure of the error taxonomy. Section 3.2 details an implementation of this general stucture for the overhead automatic modeling case. After discussing some properties of the chosen error defnitions, we explain, in the final Section 3.3, how the final labels are extracted from the error taxonomy based on the quality evaluation requirements.

3.1 The general framework

As stated previously in Section 1.3.1, a semantic evaluation implies a categorization of errors affecting building models. In the same subsection, we discussed the desired properties in order to achieve a **large-scale** and **automatic** semantic evaluation. Before delving into details, we first examine the implications of such properties on error definitions in Section 3.1.1. Next, Section 3.1.2 presents a general layout of the proposed error taxonomy.

3.1.1 Hierarchization and modularity

The goal of this subsection is to state the desired characteristics in the categorization of building model defects based on the discussion in Section 1.3.1. We introduce first two notions: generalizability and exhaustivity. These are directly linked to the **large-scale** aspect of the evaluation approach. These are implemented in the error categorization based on two principles: hierarchization and modularity.

The generalizability *vs.* exhaustivity compromise.

In Section 1.3.1, we have seen how the **large-scale** constraint on the quality evaluation approach entails the method's robustness to changes in the urban scene as well as to the pipeline behind the evaluated model. The first condition implies the proposed error categorization capacity to be generalizable: the ability to describe defects of building models unhindered by the specificities of one scene or another. The second conveys the exhaustivity of the evaluation: the power to take into account all possible errors that a building model can be affected by, free of any consideration of its origin.

The two notions are condradictory. At one hand, every possible defect should be accounted for at all levels. This may be possible through a meticulous analysis of models from a specific area. In fact, similar to the procedural modeling approaches, building model errors could be portayed relying on expert knowledge. For instance, Haussmann style building modeling defects could be described exhaustively. Eventhough these errors would be sufficient for a small disctrict in downtown Paris, they are clearly not comprehensive enough to encorporate cases from other types of buildings like in La Défense, just 3 km away of Paris, let alone ones from Timbuktu, Mali.

On the other, the categorization has to stay always relevant no matter the origin of those models. This can be, for example, achieved based on a list of errors that are common across all possible building types. However, this has the disadvantage of not covering all the instances specific one input method or another. A compromise has to be reached in the definition of errors between generalizability and exhaustivity.

General structure.

We introduce a hierarchical structure to categorize errors in order to mitigate the last point. The higher in the ladder the error is, the more generalizability (and less exhaustivity) is achieved. At the same hierarchical level, to avoid having to exhaustively list all possible errors, the identified defects are described modularly based on some predefined independent errors. This helps cover a wide range of possible defects while the basic errors are chosen to be as generalizable as possible. Hierarchization and modularity are the main ingredients in our proposed flexible framework.

To implement these properties for the error taxonomy, we rely on two criteria for error compilation: the input building model LoD and the error semantic precision level, named henceforth `finesse` (cf. Figure 3.1). Different degrees of `finesse` describe, from coarse to fine, the specificity of defects. `Finesse` degrees corresponds to error hierarchy levels. The LoD is used, on the other hand, to differenciate between errors in the same specificity (or `finesse`) level. Multiple errors at the same `finesse` can indeed affect the same building. For instance, topological defects almost always induce (and hence co-occur) with geometrical ones.

Errors with maximal `finesse` are called `atomic` errors. `Atomic` errors are to be intuitively correlated to independent actions needed by an operator or an algorithm so as to correct the model.

3.1.2 A general classification of errors

Herein, based on the previous discussion, a general layout is detailed for building model evaluation.

At a first level, model qualifiability is studied. In fact, aside from formatting issues or geometric inconsistencies (Ledoux, 2018), other reasons make building models unqualifiable. For instance, buildings can be occluded by vegetation and thus cannot be assessed from most of the remote sensing data sources. Generally speaking, input models can be impaired by some pathological cases that are outside our evaluation framework. In consequence, `Qualifiable` models are distinguished here from `Unqualifiable` buildings. This first level corresponds to a `finesse` equal to 0.

At the `finesse` level 1, we predict the correctness of all qualifiable buildings. It is the lowest semantization level at which the evaluation of a model is expressed. Then, a model is either `Valid` or `Erroneous`. Most state-of-the-art evaluation methods address errors up to this level.

Model errors are grouped into two families depending on the underlying LoD. The first family of errors `Building Errors` affects the building in its entirety. It corresponds to an accuracy evaluation at LoD-0 (footprint errors) ∪ LoD-1 (height/geometric error).

At the next LoD-2, the family `Facet Errors` gathers defects that can alter the facet accuracy of façades or roofs (LoD-2) as well as superstructures and openings (LoD-3).

Each family contains `atomic` errors of maximal `finesse` equal to 3. Although they can co-occur in the same building model and across different families, these errors are semantically independent[17]. They represent specific topological or geometric defects. Topological errors translate inaccurate structural modeling, while geometric defects raise positioning infidelity.

The general structure is not fixed and can evolve to adapt to more cases. In fact, instead of grouping LoD-2 and LoD-3 errors, the latter can be made into a different family that can be called `Superstructure Errors`. Due to the lack of sufficient observations, we did not make this choice in order to guaranty the generalizability of the taxonomy. Another alternative consists, for instance, in gathering error families by resolution: this will produce a continuum of errors families going from the coarsest level that would correspond to `Building Errors` to the finest possible one. This last option, although offering an exhaustive and potentially generalizable taxonomy, was ruled out since it does not provide a truly semantic description of the errors. Regarding `finesse`, it is also possible to have additional levels. The maximal level of 3 was chosen in order to preserve the generalizability of the taxonomy, since the more specific the error categorization is, the more observations are needed to define the corresponding errors.

[17] As mentioned before, they relate, instinctively, to independent corrective tasks.

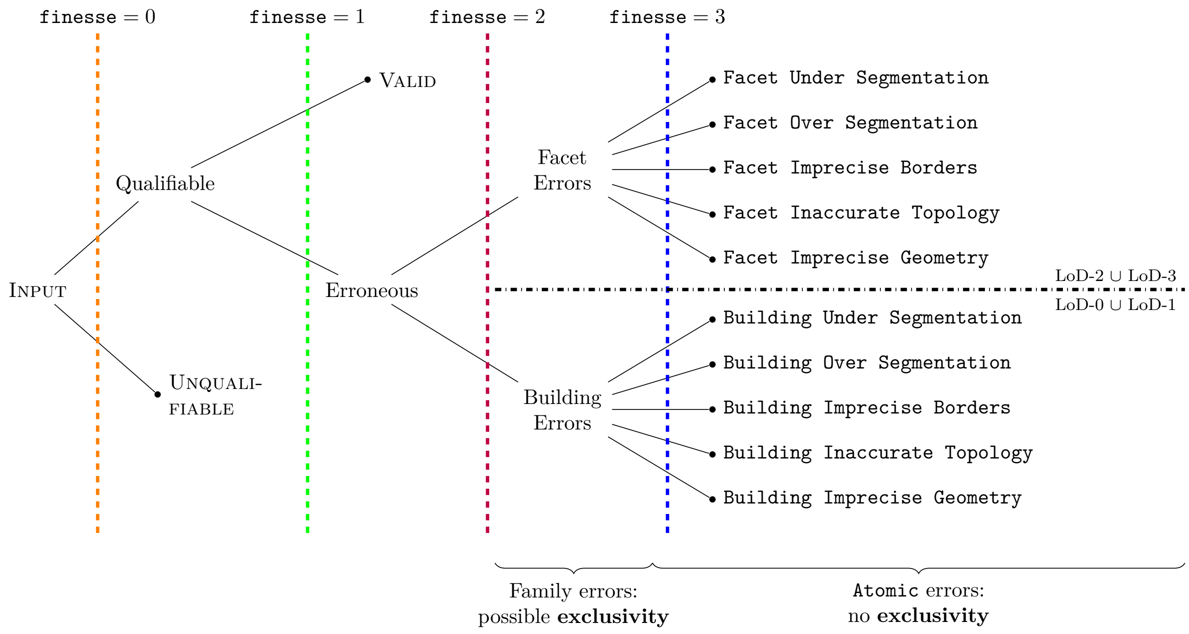


Figure 3.1: The proposed taxonomy structure. In our case of VHR overhead image modeling, only two family errors are depicted. At `finesse` level 2, hierarchization is possible: an **`exclusivity`** parameter can thus act. However, it is not the case at the `atomic` errors level since they are independent.

3.2 Application to the overhead case

Our observations were based on large datasets of 3D models of buildings reconstructed automatically out of VHR images or, if available, LiDAR point clouds. The framework introduced in the previous subsection was applied to our special case. To do so, let us define the `atomic` errors before discussing their properties.

3.2.1 Atomic error definitions

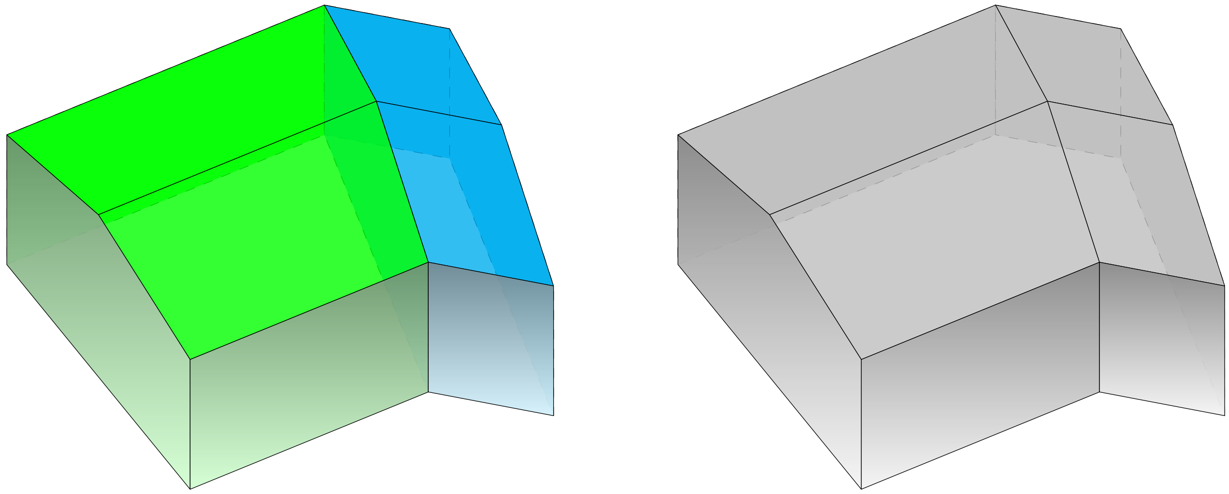
In the template structure presented in Section 3.1.2, were left out the `atomic` error definitions. Indeed, since they represent the most specific level, their choice is critical to guaranty both the desired exhaustivity and generalizability. We conducted a thorough inspection of all defects that we detected in our datasets and came up with the following definitions (cf. Figures 3.1). Eventhough these errors were defined based on models of buildings computed out of overhead acquired data at large scales, we think they are exhaustive enough to describe the quality of models in other settings (façade modeling, manually plotted 3D models).

`Building Errors` family.

Herein are presented the `atomic` errors regarding the LoD-0 and LoD-1 aspects.

`Building Under Segmentation`. BUS corresponds to the case where two or more buildings are modeled as one. In Figure 3.2, two distinct buildings were identified as one building, eventhough they can be visually distinguished.

This is a very common error which results from a faulty footprint of the building. The latter is either retrieved automatically during the modeling (Lafarge et al., 2012),



(a) Ground truth 3D models. The different buildings are in different colors: blue and green.

(b) 3D models of the buildings fused into one model.

Figure 3.2: Illustration of a BUS error.

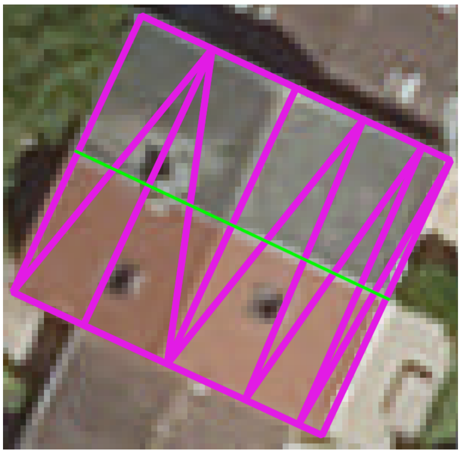


Figure 3.3: Nadir projection of an erroneous building superposed on the corresponding orthoimage showing a BUS error. We can recognize, based on the color differences of roof tiles, the existance of two buildings instead of one.

or is provided as input (Durupt et al., 2006). The first case is the most error inducing one as it relies on extrinsic large-scale remote sensing data that are devoid of semantics. The second one is expected to be more close to the reality, but can be unsuitable if the footprints are outdated or too generalized, as discussed in Section 2.1.1.

Building Over Segmentation. Building Over Segmentation (BOS) corresponds to the case where one building is subdivided into two or more. This is the opposite of the previous situation. Figure 3.4 shows a single building that, when modeled, was subdivided into three parts.

This is also a very common error. It is the consequence of the same reasons as the under segmentation that was discussed earlier. Both these errors are highly semantic and ,thus, creates confusion between both classes. Depending on the chosen semantics, a building part (in the sense of CityGML) can be also considered as a single building in other cases. These defects depend, actually, on the human perception of buildings and

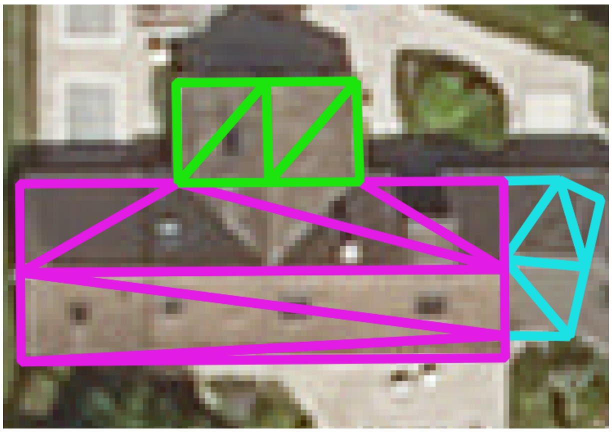
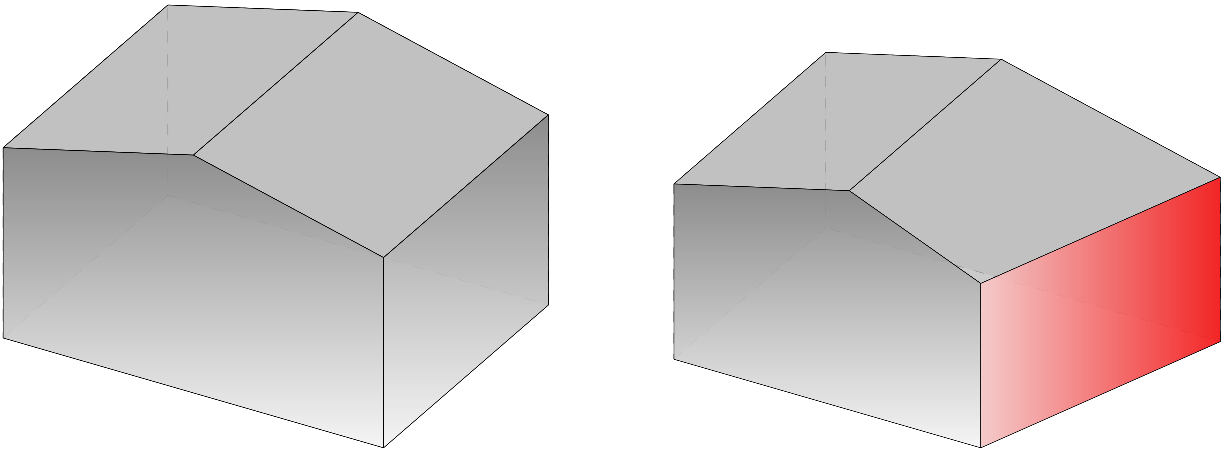


Figure 3.4: Nadir projection of an erroneous building superposed on the corresponding orthoimage showing a BOS error. We can see how a single building that was modeled into three different ones depicted here in three different colors.

are more ambiguous by nature, unless *explicit* semantic information is provided along the geometry fo the model. In fact, there is no combination of geometric characteristic that can help separate buildings, such as convexity or compactness. This issue is expected to weight negatively on the predictive capacity of the proposed evaluation approach as will be further studied in Section 5.3.

Building Imprecise Borders. BIB corresponds to the case where at least one building footprint border is incorrectly located. A sample is shown in Figure 3.5.



(a) Ground truth 3D model.

(b) 3D models with an imprecise border (in red).

Figure 3.5: Illustration of a BIB error.

This is a purely geometric error that is caused by an imprecise footprint. Actually, semantics play a role as BIB is mainly linked to the end user preferences: one can ignore errors up to a certain threshold. As with previous BUS and BOS errors, the footprint border precision is understandably susceptible on the quality of the input data used for modeling. It is also expected that the error detection precision will hinge on the resolution of the used reference data and its registration accuracy. Regarding automatic modeling methods, border imprecision can be attributed to the quality of the used edge detection algorithms (Baillard et al., 1999; Werner et al., 2002; Nan et al., 2015) or inaccurate

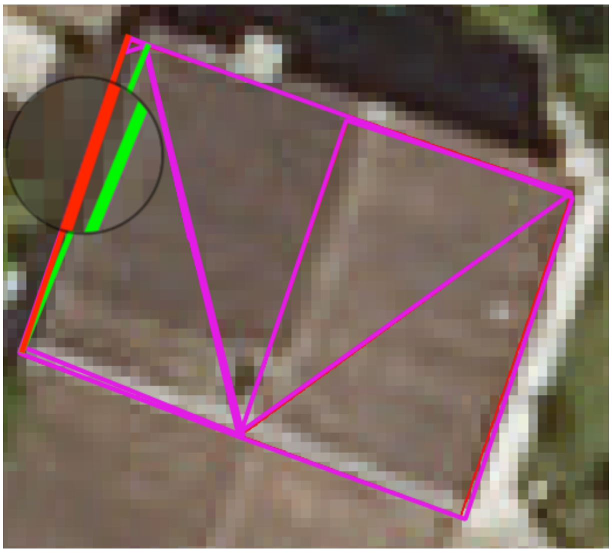


Figure 3.6: Nadir projection of an erroneous building superposed on the corresponding orthoimage showing a `BIB` error. In red is the reconstructed model border that is misestimated, as can be checked in the orthoimage. We can distinguish in green the actual edge using a Nadir projection.

surface estimation[18] (Durupt et al., 2006; Xiong et al., 2014). Outside the scope this study, one can try also to estimate the imprecision so as to correct the reconstruction.

`Building Inaccurate Topology.` `BIT` corresponds to the case where the building footprint suffers from topological defects. Modeled as a 2D flat surface, the cases that fall into this label are:

Missing inner court: It corresponds to a missing hole (cf. Figure 3.20a);

Inaccurate border shape: It is due to a wrong primitive fitting: the shape of the footprint can be better described by a different geometrical shape. Figure 3.8 gives an example where the polygon has a wrong number of sides. Another case is where a circular footprint can be approximated by a polygon.

This error, as the earlier one, is a result of a defective edge estimation. The main difference between both of them states in the fact that `BIB` is geometric in nature while `BIT` is topological. Both errors are independent and can overlap.

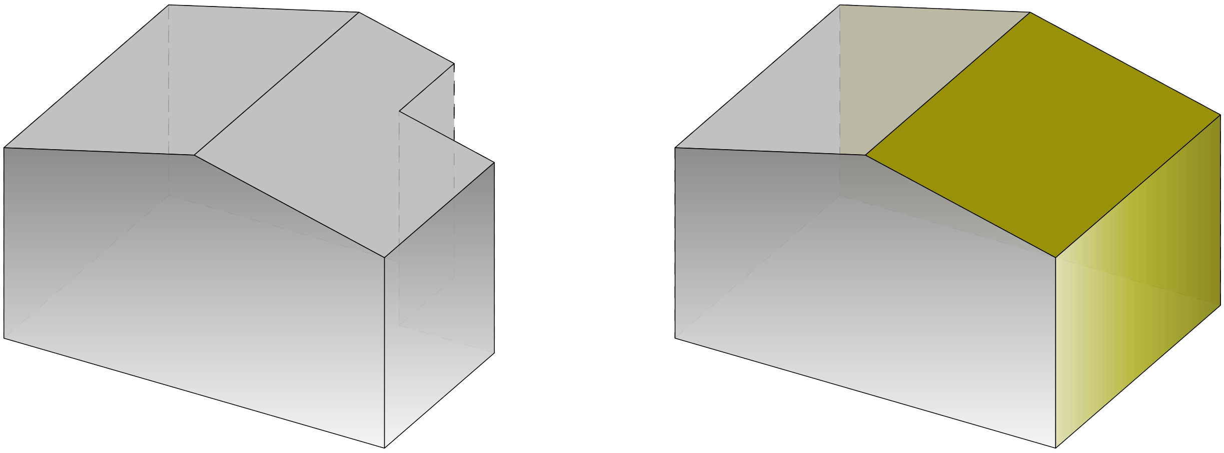
`Building Imprecise Geometry.` `BIG` corresponds to the case of inaccurate building geometric estimation.

Up to LoD-1, it coincides with height imprecision, as depicted in Figure 3.9. This is yet again a geometric error. Semantics play a role in the definition of the height of a building. It can be defined as the height at the highest point, the mediane height or any other valid alternative. In case of evaluating at higher than LoD-2, this error is not reported as it becomes redundant with errors delineated below. In fact, if a geometric error is detected at the facet level then it will naturally impact negatively on the geometry of the model as a whole.

`Facet Errors` family.

In this Section, LoD-2 and LoD-3 corresponding `atomic` errors are presented.

[18] The border is computed as intersection of the detected surfaces.



(a) Ground truth 3D model.

(b) 3D models with an inaccurate topology. The concerned borders are in yellow.

Figure 3.7: Illustration of a BIT error.

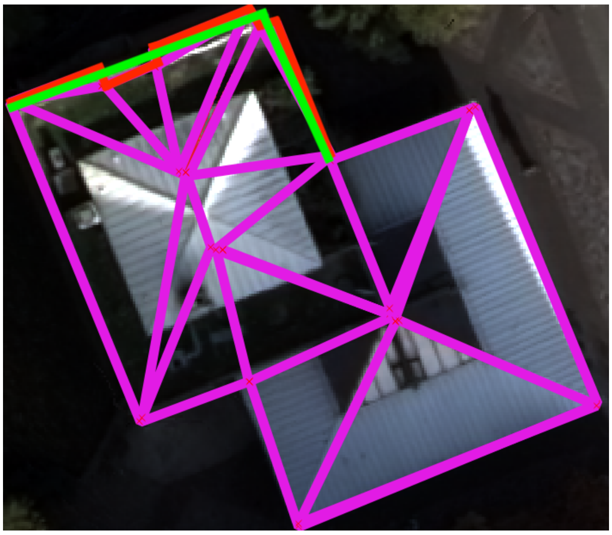
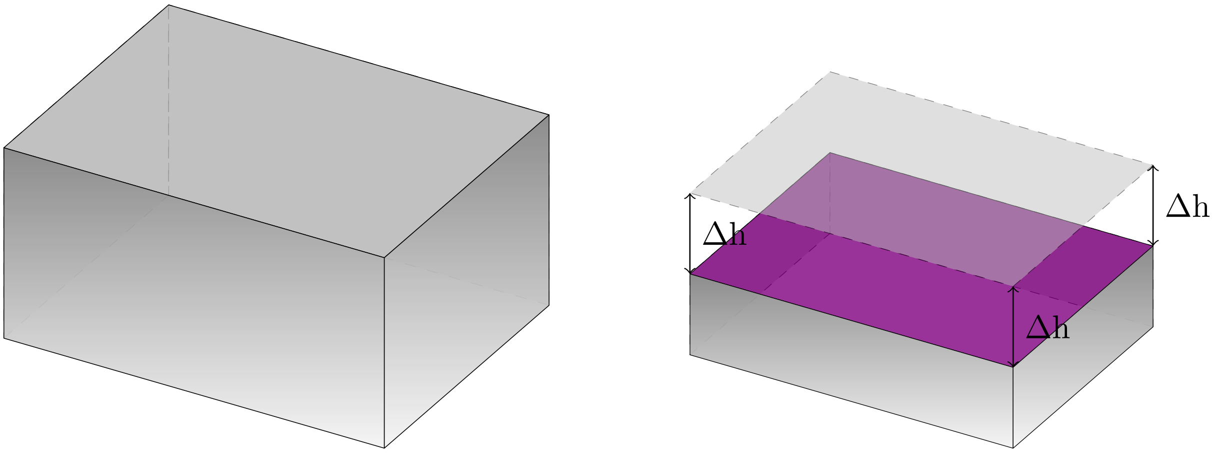


Figure 3.8: Nadir projection of an erroneous building superposed on the corresponding orthoimage showing a BIT error. We can identify (in green) the correct footprint shape compared to the one that was reconstructed (in red).

Facet Under Segmentation. FUS corresponds to the case where one facet is subdivided into two or more facets. This is the same kind of error as BUS but at the facet level.

Usually automatic reconstruction methods rely on an initial surface (usually plane) extraction step that generates proposals for further refinement. Noise from stereo pairing or missing data in point clouds result in imprecisions in elementary surface retrieval which then lead to surfaces being confused. The accuracy drops even further in some cases. For instance, superstructures like dormer windows can be big enough to be confused with the roof facets. Another setting where surfaces are hard to extract occurs when a building part is shadowed by an other that is higher. This is depicted in Figure 3.11. Methods relying only on plane extraction (Taillandier et al., 2004; Durupt et al., 2006; Nan et al., 2017) are particularly vulnerable to this error type.

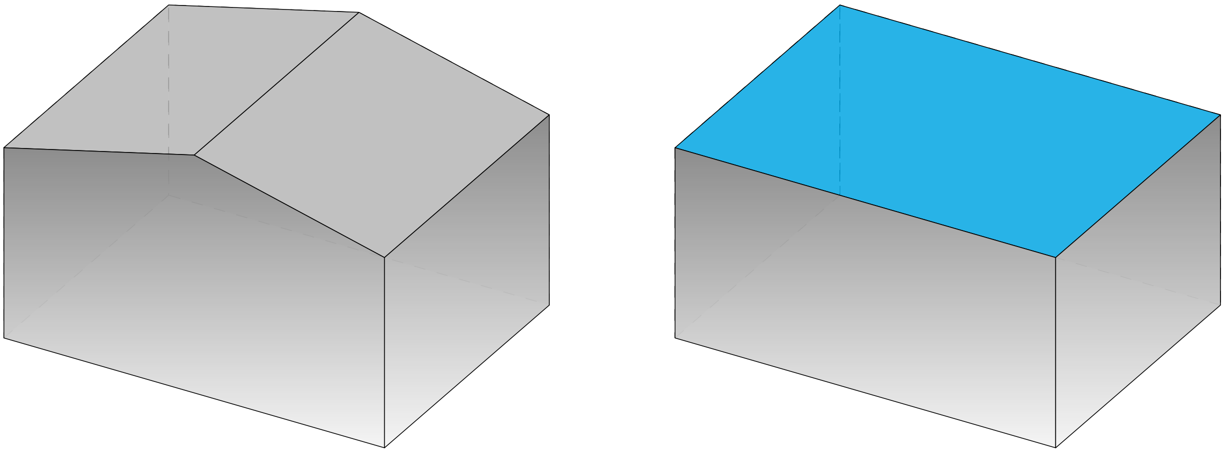
This defect can be mitigated through the use of 3D lines as cues to guide the plane extraction (Zebedin et al., 2008; Sinha et al., 2009). One can also discard plane extraction



(a) Ground truth 3D model.

(b) 3D models with an imprecise height estimation.

Figure 3.9: Illustration of a BIG error.



(a) Ground truth 3D model.

(b) 3D model with a FUS error. The erroneous facet is colored in blue.

Figure 3.10: Illustration of a FUS error.

alltogether and try to reconstruct the building surface based only on 3D lines (in other words, a wireframe building model) (Hofer et al., 2017; Langlois et al., 2019). Using grammars of possible stuctures is another alternative, provided it is adequate to the modeled building. Lafarge et al. (2008) fits the best type of roof models to alleviate issues caused by high levels of noise like when working with Satellite based DSMs. In some cases, there may be no remedy for the issue, as the used grammar is not exhaustive enough, a 3D line goes undetected or even a human operator intervention is unable to dispel the ambiguity.

Facet Over Segmentation. FOS corresponds to the case where two or more facets are modeled as one, as illustrated in Figure 3.12. This is to BOS what FUS is to BUS.

As seen previously, lines are used to help find correct planes and avoid under segmentation. However, an overdetection of lines can lead to an oversegmentation of the model. This is not rare due to problems that can be encountered with illumination conditions: for instance, a roof structure can cast its shadow on a neighbooring one and cause a gradient in image signal that will be translated to a virtual edge. Superstructures play also a

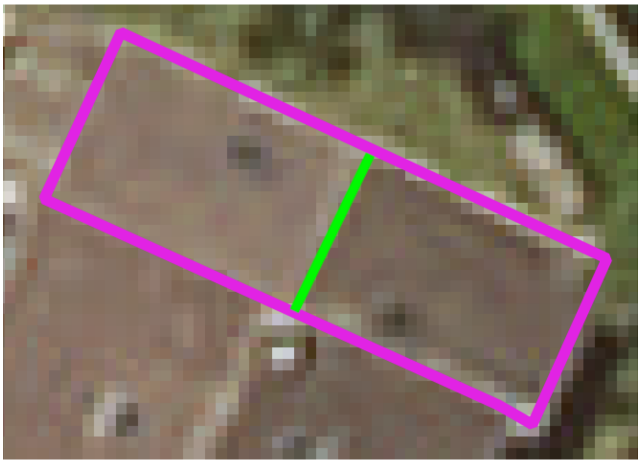
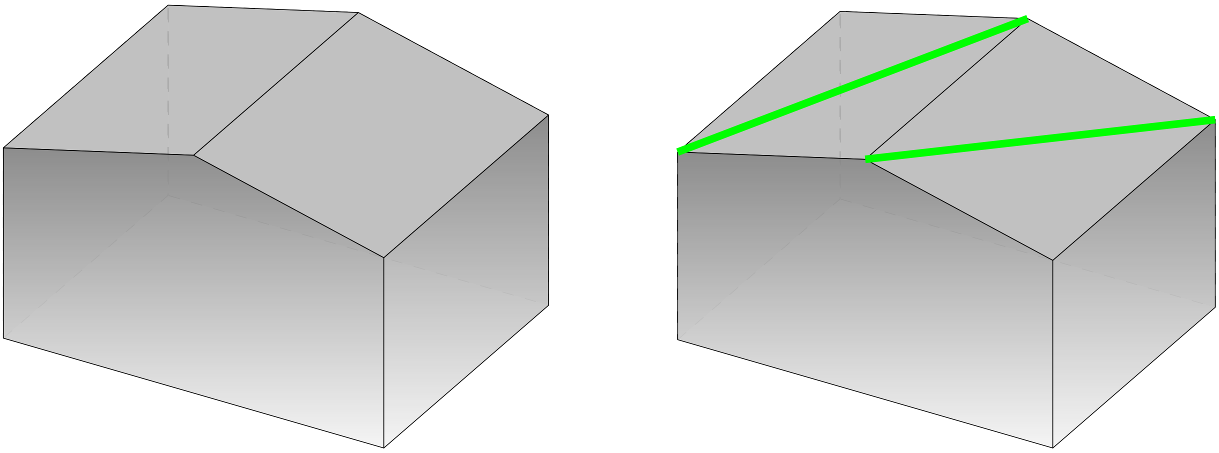


Figure 3.11: Nadir projection of an erroneous building superposed on the corresponding orthoimage showing a FUS error. It shows a case where a higher neighboring building part can drive a misestimation of both facet planes which end up confused in one flat roof. The line segment highlighted in green gives away the fact that the roof was undersegmented.



(a) Ground truth 3D model.

(b) 3D model with a FOS error. In green are colored the erroneous edges.

Figure 3.12: Illustration of a FOS error.

negative role just as explained previously. This time it is the ones that are small in planar size compared to the noise order of magnitude that are not detected but add bumps that pollute the signal and result in an misestimation of planes. High neighbooring buildings are also to blame due to the same reasons as with FUS, but this time they result in bumps like with superstructures.

To solve this kind of issues, mesh simplification techniques can be helpful. In fact, Verdie et al. (2015) uses this approach to smooth away these problems and produce a good generalization of the underlying buildings. Another way is to filter the extracted lines relying on redundancy as shown in Michelin et al. (2013). Grammar based methods can equally come to rescue. As an example, Brédif et al. (2007) uses a set parameteric models in order to model superstructures and better fit LoD-2 roof facets.

Facet Imprecise Borders. FIB corresponds to the case where at least one facet border is incorrectly located. As an example, Figure 3.15 shows that the central edge that

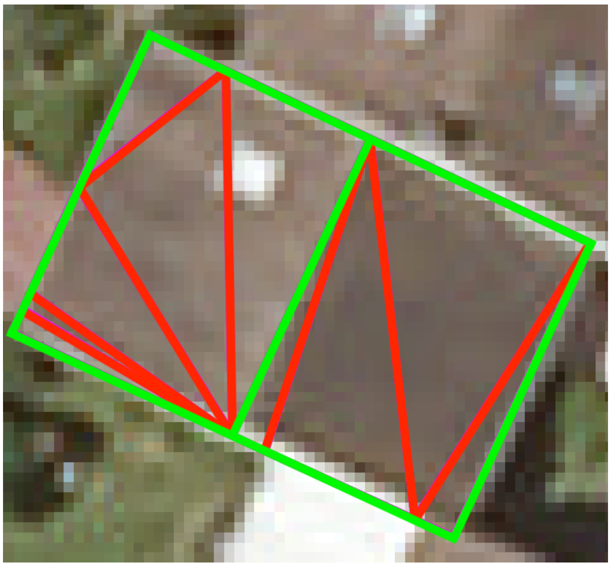
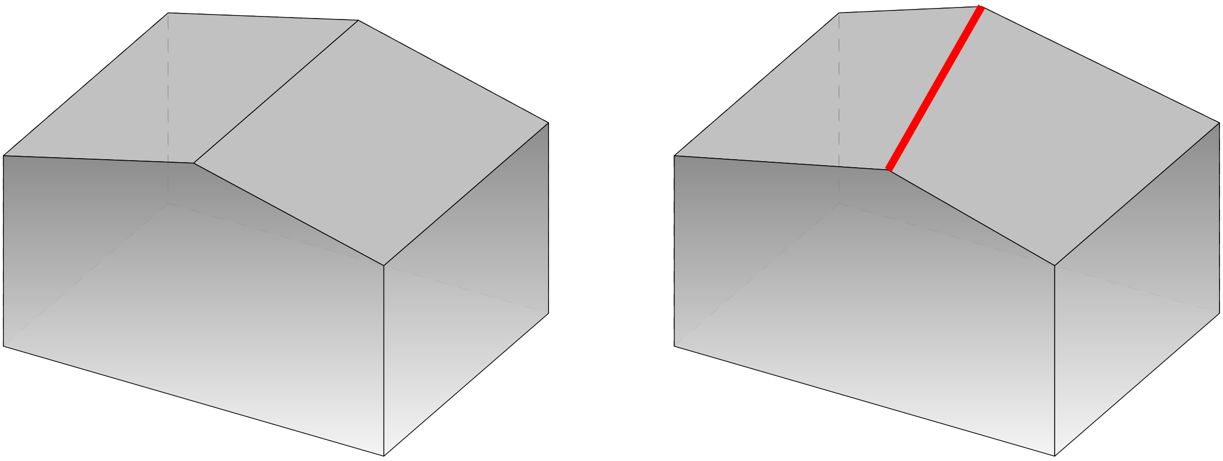


Figure 3.13: Nadir projection of an erroneous building superposed on the corresponding orthoimage showing a FOS error. A slim chimney in the below corner of the ridge results in a defect ladden DSM which translates into an oversegmentation. The erroneous edges are colored in red. One can check using the orthoimage that these are not real.

links the two main roof sides does not correspond to the one on the image position. This is a purely geometrical error similarly to BIB.



(a) Ground truth 3D model.

(b) 3D model with a FIB error. The misplaced edge is colored in red.

Figure 3.14: Illustration of a FIB error.

Line extraction is usually very faithfull to the data and depends mostly on the resolution and quality of the input data used for modeling. The most likely reason usually behind this kind of errors is usually imprecise fitting of primitives that leads to a shifted intersecting edge such as shown in Figure 3.15.

Just as with oversegmentation, one way to make line retrieval more accurate is to rely on redundancy by extracting them from different modalities. An alternative is to rely on symmetries (Verma et al., 2006) to automatically correct surface intersections.

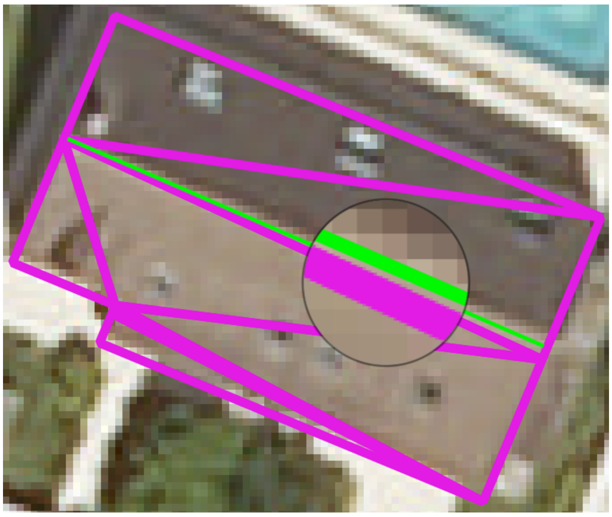
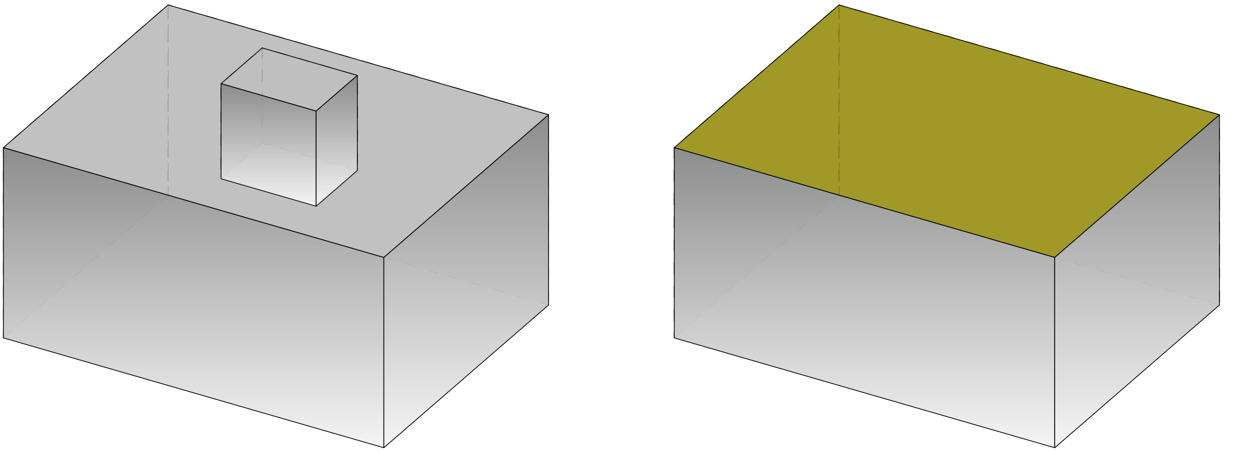


Figure 3.15: Nadir projection of an erroneous building superposed on the corresponding orthoimage showing a `FIB` error. The real location of the edge is shown in green.

`Facet Inaccurate Topology.` `FIT` corresponds to the case where the facet suffers from topological defects such as wrong primitive fitting (for example, a dome approximated by planar polygons). In Figure 3.16, we can observe how two cylindrical towers were reconstructed as a rectangular parallelepiped.



(a) Ground truth 3D model.

(b) 3D model with a `FIT` error. The facet in yellow has a missing hole.

Figure 3.16: Illustration of a `FIT` error.

This can be due to various reasons. Most methods rely on the assumption that buildings are piecewise linear (Nan et al., 2017) or Manhattan-world like (Li et al., 2016). This is evidently not always the case (cf. Figure 3.17). Even with the right assumptions, this specific case cannot have been well modeled. If so it would have at least approximated the circular cylindrical structures with a regular polygon cylinder. This is due to the fact that the quality of the data was poor and was unreliable as explained in the `FUS` case in Figure 3.11. The same effect can cause a missing hole being undetected.

Solving this issue is hard besides the obvious change of primitive assumptions. It depends highly on the quality of the data. One can try to overcome this issue once again, like with the undersegmentation problem, thanks to line detections to reveal convoluted

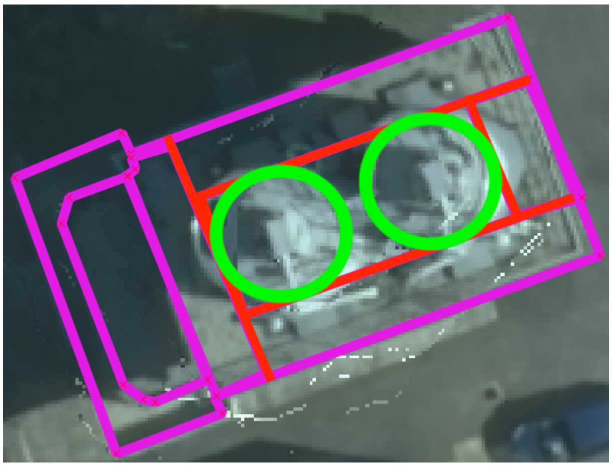
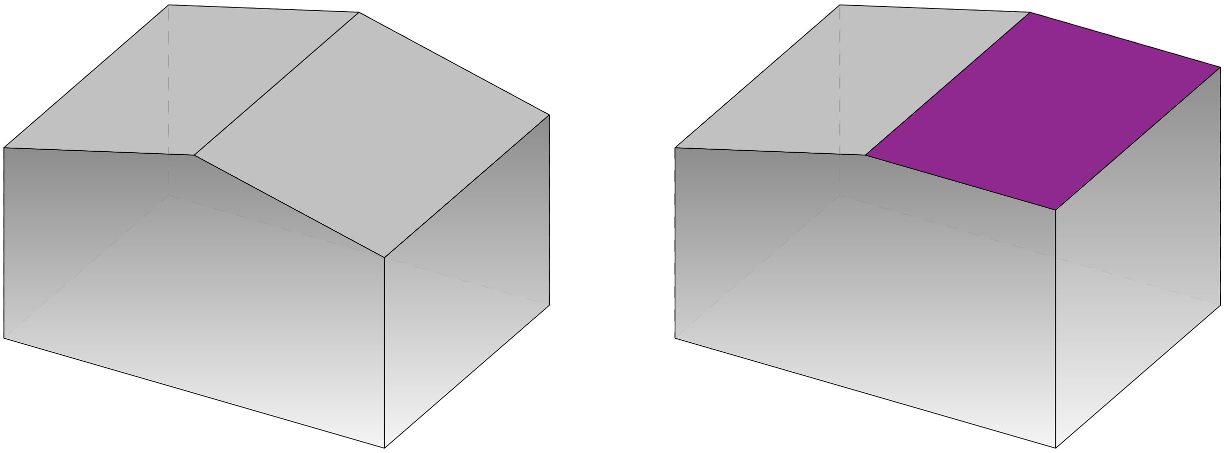


Figure 3.17: Nadir projection of an erroneous building superposed on the corresponding orthoimage showing a `FIT` error. The true form (in green) of the towers that is completly misrepresented (in red).

structures when relying on plane extraction only. Usually, this problem can be efficiently solved relying mainly on human operators.

`Facet Imprecise Geometry.` `FIG` corresponds to the case of inaccurate facet geometric estimation. In mathematical terms, this means that the surface primitive parameters are misestimated. In Figure 3.18b, the planar surface slope was miscalculated as flat while it was of *ca.* 25°.



(a) Ground truth 3D model.

(b) 3D model with a `FIG` error. The facet in purple has a wrong slope: It is flat instead of being sloped.

Figure 3.18: Illustration of a `FIG` error.

This is linked in particular to the input sensor data quality. The case of Figure 3.19 illustrates how neighbooring building parts influence the data as it blunts away the slope of the roof. Semantics play again a role in detecting a purely geometric error. For correction, as the corruption in the data resulting from the modeling limits could be filtered out using semantics, one can reestimate the parameters of the primitives. Failing that, this correction step is usually conducted by human operators.

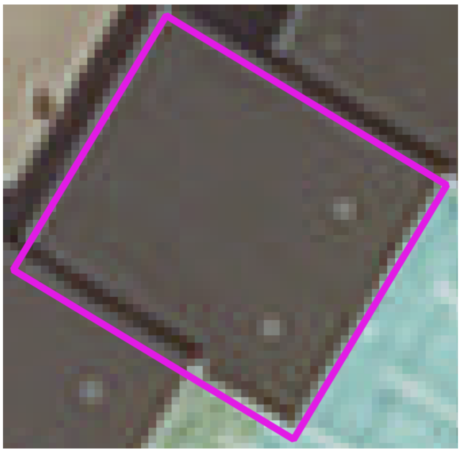


Figure 3.19: Nadir projection of an erroneous building superposed on the corresponding orthoimage showing a `FIG` error. This kind of error is impossible is vizualize on the orthoimage. Projected on the DSM, we can reveal the sloped character of the face.

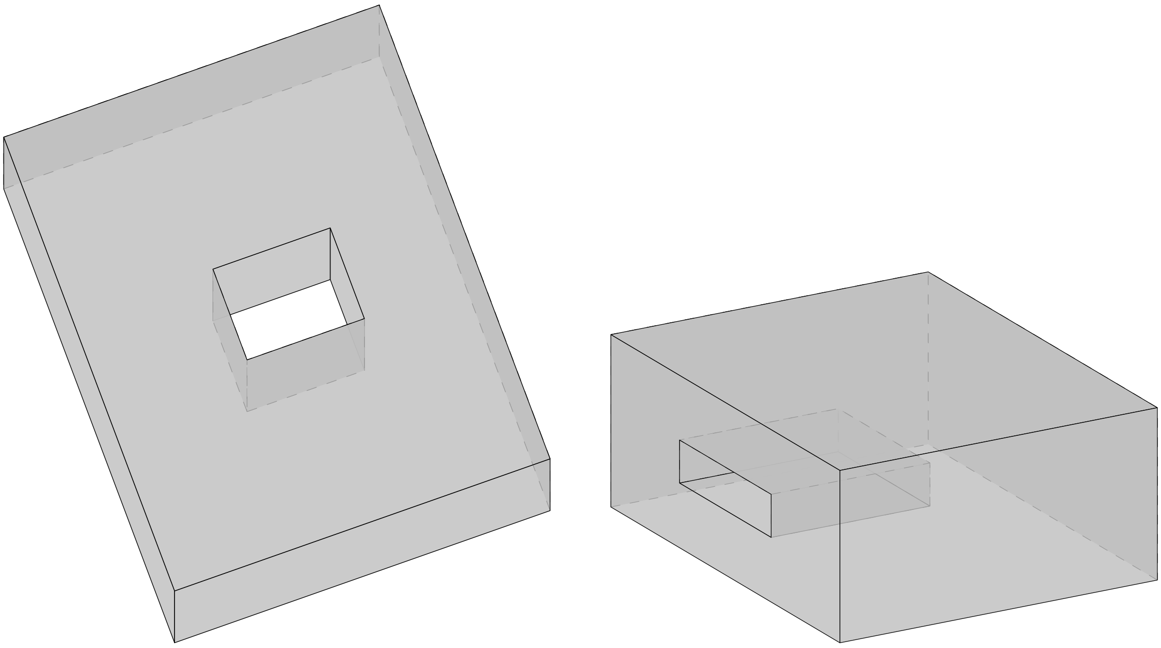
3.2.2 Discussion

In the previous subsection, we defined `atomic` errors based on our observations related to automatic overhead modeling of urban scenes. We also discussed how each type of defects can occur and how to proceed in order to correct them. Herein, we explore the properties of the ensuing taxonomy of errors. In addition, we also discuss the relation between this taxonomy and the ones existing in the litterature.

Error taxonomy properties.

The similarity of the `Building Errors` and `Facet Errors` is striking. Indeed, this is what was intended. The idea came after observing that segmentation issues can occur both at building and facet errors. It was ruled that both should be separated as they occur at different LoDs so as to have a fine representation of those defects as discussed in Section 3.1.2. If `atomic` errors are defined on particular defects observed only at a certain level, one can expect that these definitions are going to be too specific and not exhaustive enough. This was the case of an earlier categorization of errors based only on a smaller subset of our datasets (Ennafii et al., 2018a). For instance, at building level, in our dataset, missing courts are fairly present. However, there are no cases where facets have holes like illustrated in Figure 3.20b. Both can be grouped in one single label "missing hole". This example, illustrated in Figure 3.20, makes the case for using similarity across LoDs being instrumental for generalizability.
This will also be an interesting option to use even if the error families were different as was proposed in Section 3.1.2.

Defining errors in 3D is not an easy task. The *implicit* semantics, discussed in Section 1.1.2, implies the fact that each facet of the model is special and has a meaning. Each facet contained in the model is, hence, supposed to be unique and persistent: it cannot be replaced by an approximating set of different facets as in a mesh. This meant that evaluating a whole building model amounts to evaluating each facet individually. This simplifies greatly the issue as one can think about the problem moslty as a 2D evaluation one. In fact, the first four `atomic` errors, at each level, involve only 2D sufaces, which in the case of planar surfaces amount to polygon evaluation. The last error is the only one that takes into account the 3D ascpect of the facets. This explains why border impreci-



(a) A hole at building (viewed from the top) level (LoD-1) corresponds to an internal court.

(b) Example of a facet (LoD-2) with a rectangular hole corresponding to a balcony.

Figure 3.20: Holes observed at different LoDs.

sions (which are geometric issues) were separated from the other geometric inaccuracies. These latter errors are, in fact, somewhat overlapping. These discussed properties does not only help the generalizability and exhaustivity in the overhead case, but can also be applied to terrestrial based façade modeling. In fact, the same `atomic` errors can describe perfectly the possible errors that can affect facets on a façade, owing always to the effect of *implicit* semantics on building models.

Related taxonomies.

This Section discusses the semantic error defnitions in the state-of-the-art. As mentioned in Section 2.2.1, Rottensteiner et al. (2012) proposed two semantic errors "over segmentation" and "under segmentation". These, however, relate only to facets. Xiong et al. (2014), although not aiming at evaluating the quality of a model in its entirety, provides some error labels relative to their reconstruction method. For instance, "Missing Node" (*resp.* "False Node", "Missing Edge" and "False Edge") correspond to, or are included in, the topological `atomic` errors from the `Facet Errors` family: `FUS` (*resp.* `FOS`, `FIT`, and `FIT`).

On the other hand, the taxonomy developed by Michelin et al. (2013) is the closest to ours. In fact, footprint errors could be reshuffled into `Building Errors` as `BIB` ("erroneous outline" and "imprecise footprint") and `BIT` ("missing inner court"). Intrinsic reconstruction errors are divided into both levels. In fact, "over-segmentation", "under segmentation" could be part of both `Building Errors` as well as `Facet Errors` families. "inaccurate roof" is a general error that can include `FIB` and possibly `FIT` also. "Z translation" is the last label in "Reconstruction errors" and is either a `BIG` error when working at LoD-0 ∪ LoD-1 level, or included into `FIG` if dealing with flat roof facets. Finally, "vegetation occlusion" and " non existing" are gathered into the `Unqualifiable` label at `finesse` level 0.

Last, Boudet et al. (2006) studied rather the acceptability of a model as a whole. Confidence in a building model is a subjective assessment of building models that depend on the end user needs. Consequently, such a taxonomy cannot directly fit with our labels. The acceptability dimension can be incorporated into our framework by attributing a confidence score to each error: for example, a prediction probability.

3.3 Parametric label extraction

When evaluating building models, not all errors are taken into account depending on the end user needs. In fact, labels are extracted according to the taxonomy based on the evaluation settings. We will see in this subsection how this is possible and what are the consequences of this choice.

3.3.1 Evaluation parameters

At evaluation time, three parameters play a role in determining which error labels to consider. We determine hereafter these parameters and their role.

The first parameter is the **eLoD**. Every reconstruction method targets a certain set of LoDs. In consequence, when assessing a reconstruction, a LoD must be specified. At a given eLoD, all error families involving higher orders will be ignored.
Depending on the target of the qualification process, a `finesse` level might be preferred. This corresponds to the second parameter called **eFin**. It specifies the appropriate semantic level at which errors will be reported.
The last one is error **exclusivity**. It derives from the established family error hierarchy. If **exclusivity** is set ON errors of a given LoD-l family are prioritized over ones with higher LoDs: i.e., LoD $>$ LoD-l. The latter are simply ignored. This would useful in the case where the quality evaluation is used by a correction module, either manually or automatically operated. In this case, the latter should prioritize solving low LoD errors rather than trying to solve more detailed ones. This stems from the fact that dealing whith low LoD errors would probably impact the shape of higher LoD features. As a consequence, detecting and correcting the latter rather than prioritizing the low LoD ones is going to be virtually wasteful in ressources.

3.3.2 Evaluation labels

Depending on the previously defined parameters, the considered labels that will be used for the evaluation would differ. We will visit herein all the possible cases based on the defined taxonomy for the overhead reconstruction case (cf. Figure 3.1). For this purpose we must first define the function children that gives the children of a non-leaf node in the taxonomy tree and outputs the same node if it is a leaf:

$$\begin{aligned} \text{children} : V &\to V \\ n &\mapsto \begin{cases} \{n\} & n \in L \cdot \\ \{m \in V : (n,m) \in E\} & n \notin L \end{cases} \end{aligned} \tag{3.1}$$

where:

V : is the set of all vertices of the taxonomy tree;
L : is the set of leaf nodes in the tree;
E : is the set of edges (directed) in the tree.

There are eight cases corresponding to one for the case **eFin** = 1, three for the error families level and four for the last case of `atomic` error level.

i) **eFin** = 1. The model can either be `Valid` or `Erroneous`. As discussed previously, most quality evaluation methods reason at this level.

ii) **eFin** = 2 and **eLoD** = LoD-1. The model can either be `Valid` or have a `Building Error`.

iii) **eFin** = 2, **eLoD** = LoD-2 and the **exclusivity** is ON. The model can either be `Valid`, have a `Building Error` or a `Facet Error`.

iv) **eFin** = 2, **eLoD** = LoD-2 and the **exclusivity** is OFF. The model can either have a `Building Error` or not. Independently, it can have also a `Facet Error` or not.

v) **eFin** = 3, **eLoD** = LoD-1 and the **exclusivity** is ON. The model can either be `Valid` or have a `Building Error`. If the latter is the case, then it is decided if the building model has a LoD-1 `atomic` error or not. All the latter errors are considered independently from each other.

vi) **eFin** = 3, **eLoD** = LoD-1 and the **exclusivity** is OFF. The model can have each LoD-1 `atomic` error or not independently from the others.

vii) **eFin** = 3, **eLoD** = LoD-2 and the **exclusivity** is ON. The model can either be `Valid`, have a `Building Error` or a `Facet Error`. If it is affected with a `Building Error`, then only its corresponding `atomic` errors are considered being present or not independently. The same decision is applied if `Facet Error` was detected, but this time with LoD-2 `atomic` errors.

viii) **eFin** = 3, **eLoD** = LoD-2 and the **exclusivity** is OFF. The model can have each `atomic` error (LoD-1 and LoD-2) or not independently from the others.

This will influence the evaluation pipeline that will be described later in the next chapter. Indeed, the latter will provide more insight about how this takes place.

4 A learning approach for quality evaluation

Contents

In the previous chapter (Chapter 3), we developed a hierarchical and modular taxonomy of errors for the overhead imagery modeling case. Based on this taxonomy, and depending on the particular needs, specific error labels are extracted and considered during the quality evaluation step.

In this chapter, we present the second part of our proposed approach, as we cast the problem as a supervised classification one. Issues related to the latter are detailed in Section 4.1. Next, Section 4.2 presents in details the baseline of features extracted out of 3D building models. Third and last, implementation details regarding the baseline features are discussed in Section 4.3.

4.1 Quality evaluation as a classification task

In order to satisfy the **large-scale** condition imposed at Section 1.3.1, we propose to formulate the problem as a supervised learning one. Errors are considered as labels while features are computed so as to describe the observed buildings. Actually, as a first approach, the existence of all errors is predicted at the **building level**, even for `Facet Errors` labels[19]. Determining which facet is affected by an error is a much more challenging problem than the previous one. That is why the facet level prediction of errors is not explored in this work. The goal is, instead, to test the feasibility of the learning approach.

Errors are predicted based on learned statistical characteristics of the evaluated models. The learned approach is usually used to take care of highly semantic tasks, such as ours, that are otherwise hard to process using engineered metrics.
Provided an initial manual annotation effort, the prediction phase is fully automatic, as will be proved by experimental results in Chapter 5. In order for this approach to be scalable, and hence verify the second constraint in Section 1.3.1 "**automation**", prediction results should be stable enough independently of the urban scene. This is fully studied in Chapter 5.

Two aspects should be discussed to in order to apply this approach to the building 3D model quality evaluation. First, as seen in Section 3.3, the parametric nature of the taxonomy leads to a varying set of labels. For this purpose, we describe in Section 4.1.1 the different classification problems depending on the evaluation parameters. Secondly, the feature extraction procedure should also be compliant with the **large-scale** objective set beforehand. This aspect and more are detailed in Section 4.1.2. Third, the classifier should be able to handle the heterogeneity of the feature vector and must adapt to different input vectors types and sizes. The choice of classifiers is, hence, discussed in Section 4.1.3.

4.1.1 Different classification problems

The nature of the different classification problems are presented in Table 4.1 depending on the three evaluation parameters defined in Section 3.3. As a reminder, these are: the **eFin**, the **eLoD** and the **exclusivity**.

eFin	**eLoD**	**exclusivity**	**Classification output**
1	—	—	*binary*(`Valid`, `Erroneous`)
2	LoD-1	—	*binary*(`Valid`, `Building Errors`)
2	LoD-2	ON	*multi_class*(`Valid`, `Building Errors`, `Facet Errors`)
2	LoD-2	OFF	*multi_label*(`Building Errors`, `Facet Errors`)
3	LoD-1	ON	*multi_stage*(`Valid`, `Building Errors`)
3	LoD-2	ON	*multi_stage*(`Valid`, `Building Errors`, `Facet Errors`)
3	LoD-1	OFF	*multi_label*(children(`Building Errors`))
3	LoD-2	OFF	*multi_label*(children(`Building Errors`) ∪ children(`Facet Errors`))

Table 4.1: All possible classification problem types depending of the evaluation parameters: **eFin**, **eLoD** and **exclusivity**.

In Table 4.1, $multi\_class(l_1, l_2, \ldots, l_c)$ (*resp.* $multi\_label(l_1, l_2, \ldots, l_c)$) corresponds

[19] These errors are, in fact, by definition more local.

to the multi-class (*resp.* multi-label) setting. We note that:

$$multi\_label(\texttt{Valid}, l_1, l_2, \ldots, l_c) \equiv multi\_label(l_1, l_2, \ldots, l_c).$$

binary refers to the special case of *multi_class* where $c = 2$: i.e.,

$$binary(l_1, l_2) \equiv multi\_class(l_1, l_2)$$

Two consecutive classification problems can be concatenated in a hierarchical multi-stage classification: depending on the class that is predicted in the first stage multi-class classifier, a second classification problem predicts the existence of some corresponding labels. This denoted by:

$$multi\_stage(l_1, l_2, \ldots, l_3) \equiv multi\_label(\text{children}(multi\_class(l_1, l_2, \ldots, l_3))).$$

eFin = 1 level corresponds to the standard binary classification problem: `Valid` or `Erroneous`. At **eFin** = 2, the **eLoD** can then take two values in the aerial reconstruction case: LoD-1 or LoD-2. If set at LoD-1, it is a binary classification problem: `Valid` or `Building Errors`. For LoD-2, if the **exclusivity** is on, it will be a multi-class problem: `Valid`, `Building Errors` or `Facet Errors`. If set off, it becomes a multi-label one with the same labels. At **eFin** = 3 level, if the **exclusivity** is on, it is a 2-stage classification problem. In the first stage, a multi-class task[20] predicts the error family, after which a second multi-label problem decides between the predicted error family children. If the **exclusivity** is off, it turns into one stage multi-label problem that predicts the existence of each atomic error corresponding to the chosen eLoD.

4.1.2 Feature extraction

The proposed quality evaluation approach, being formulated as a supervised classification problem, requires extracting feature vectors describing characteristics of the evaluated model. This is possible through the use of the intrinsic properties of the building model, as well as comparisons with external data.

Intrinsic feature extraction consists in make use of the geometric structure of the 3D model. Equally, semantics, as well as building model meta-data, could be utilized for the purpose of intrinsically evaluating a model. This case corresponds to the minimal amount of information one can use for building model evaluation. In this case, we talk about self-evaluation of building models. Since we are considering all possible cases, especially automatically reconstructed building models, only the model geometry is guaranteed to be always available.

Extrinsic feature extraction relies on comparing the model to an available external data. Obviously, high quality reference models are the best type of data to compare the evaluated model with. However, taking into consideration the **large-scale** objective that was fixed earlier (Section 1.3.2), this is not a viable solution. We rely then on more basic reference data such as remote sensing acquired data that are the basis of large-scale modeling of urban scenes.

For instance, raw depth information can be used in quality evaluation. It can prove helpful in detecting geometric defects that are intrinsically of 3D nature. This was illustrated in Figure 3.18, as comparing the projected model to the orthoimage did not yield anything out of place, contrarily to the DSM comparison. Depth data can take multiple forms:

[20] It is binary in the special case eLoD = LoD-1, problem, like in the previous semantic degree.

unstructured point clouds, originating for instance from LiDAR sensors, or dense depth maps such as DSM for the overhead case.
Optical images can also be employed in this framework. These provide complementary information such as edges (high frequencies, in general) and textures which are suited for inner defect detection as an example (cf. Figure 3.11). This type of data comes usually in two different shapes: overhead images or orthoimages.

4.1.3 Classifier choice

The choice of classifiers shoud take into consideration the highly modular nature of the framework with multimodal features involving many parameters. Two classifiers where chosen in this study: RF and SVM. Both were discussed in details in Chapter B. Hereafter, we explain how each one is used in this setting.

Random Forest.

RF classifiers is a natural choice in our case. As seen in Section B.3, this type of classifiers can manage a large number of features with different dynamics and coming from multiple modalities. In fact, the computed features could be geometric, image based or height based. Each one of these modalities can also be heterogeneous in terms of extracted value types, as will be discussed later in Section 4.2. Relying on their bagging property, a high number of trees is required to cover most of the feature space, while a limited tree depth is needed to avoid overfitting during training. While the multi-class case is natively taken into account by the RF classifier, the multi-label one requires adopting a one-vs-all approach so as to address each label separately.

SVM.

SVMs do not manage well enough heterogeneity in feature vectors. Moreover, only binary classification is inherently handled. However, it offers other advantages that are not met by RF. In fact, SVMs can be useful when labels are not equally distributed in the training set. This is actually the case of some errors that are rare in our dataset: specifically the inaccurate topology ones (cf. Section 5.1.2). It also manages to learn efficiently on sets with limited sets, as will prove to be the situation (cf. Section 5.1.1). This type of classifiers naturally encorporates kernels such as the ones presented later in Section 6.2.1. Finally, it is also preferred when dealing with high dimensional feature vectors like those produced by ScatNets (cf. Section 6.2.2).

4.2 Feature baseline

Since there is no comparable work that studied the learned detection of errors defined in Chapter 3, we propose a baseline for each one of the three modalities: geometric, height based and image based features. Attributes are kept simple so as to be used in most situations relying on generally available data. We avoid computing and comparing 3D lines (Michelin et al., 2013), correlation scores (Boudet et al., 2006) or any Structure-from-Motion (SfM) based metric (Kowdle et al., 2011). In addition of being very costly, these features are methodologically redundant with the 3D modeling techniques. They are, hence, vulnerable to the same defects. Conversely, evaluation metrics used in the 3D building reconstruction literature (e.g., RMSE) are too weak for such a complex task. This will be proven later on in Section 5.2.

In Section 4.2.1, we describe the used baseline for geometric features. Next, in Section 4.2.2, height based features extraction is explained. We end with image based features in Section 4.2.3.

4.2.1 Geometric features

Given a building model M, the facet set is denoted by $\mathsf{F_M}$. $\forall(f, g) \in \mathsf{F_M} \times \mathsf{F_M}$ $f \sim g$ correspond to facets f and g being adjacent: i.e., they share a common edge. As the roof topology graph in (Verma et al., 2006), the input building model M can be seen as a facet (dual) graph:

$$\mathsf{M} \triangleq \left(\mathsf{F_M}, \mathsf{E_M} \triangleq \{\{f, g\} \subset \mathsf{F_M} : f \sim g\}\right). \tag{4.1}$$

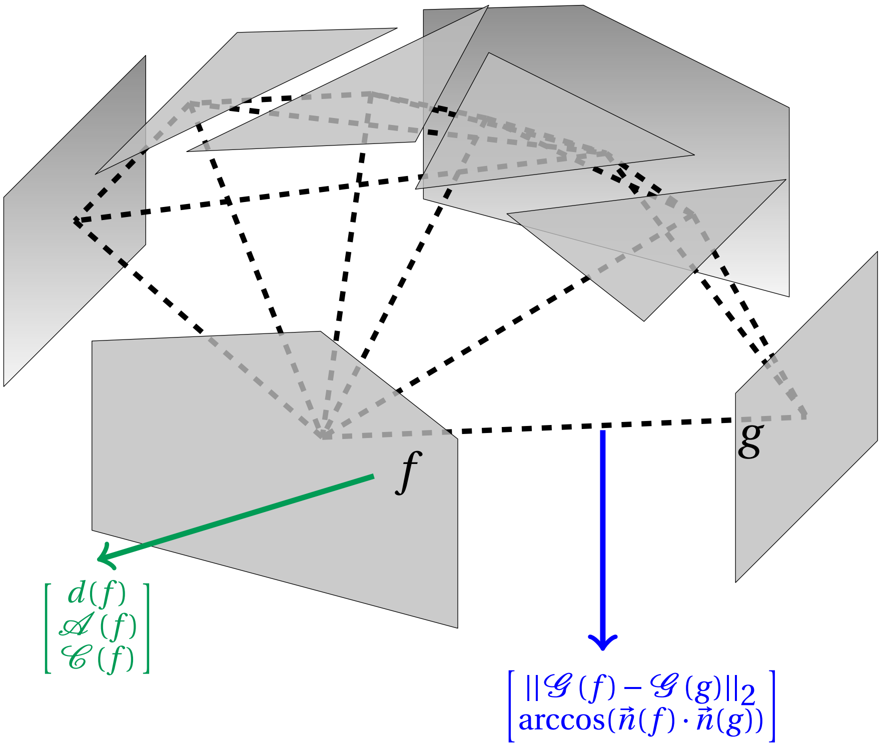

f
g
$\begin{bmatrix} d(f) \\ \mathscr{A}(f) \\ \mathscr{C}(f) \end{bmatrix}$
$\begin{bmatrix} \|\mathscr{G}(f) - \mathscr{G}(g)\|_2 \\ \arccos(\vec{n}(f) \cdot \vec{n}(g)) \end{bmatrix}$

Figure 4.1: Computed geometric attributes represented on the dual graph, for facets f and g. The green vector groups the node (facet) attributes while the blue one shows the edge features.

The dual graph is illustrated in Figure 4.1. For each facet $f \in \mathsf{F_M}$, we compute its degree (i.e., number of vertices; $f \mapsto d(f) \triangleq |\{v : v \text{ is a vertex of } f\}|$), its area $f \mapsto \mathscr{A}(f)$ and its circumference $f \mapsto \mathscr{C}(f)$. These are all geometric invariants with respect to $\mathbb{R}^3$ isometries, contrarily to facet centroids $\mathscr{G}(f)$ and normals $\vec{n}(f)$. This is countersteped by looking, for each graph edge $e = \{f, g\} \in \mathsf{E_M}$, for the distance between facet centroids $\{f, g\} \mapsto \|\mathscr{G}(f) - \mathscr{G}(g)\|$ and the angle formed by their normals $\{f, g\} \mapsto \arccos(\vec{n}(f) \cdot \vec{n}(g))$. Statistical characteristics are then computed over building model facets using specific functions χ, like a histogram:

$$\chi = \chi^{p}_{\text{histogram}} : l \mapsto \text{histogram}(l, p), \tag{4.2}$$

with p standing for histogram parameters. A simpler option could be:

$$\chi = \chi_{\text{max,min,mean,med}} : l \mapsto \begin{bmatrix} \max(l) \\ \min(l) \\ \text{mean}(l) \\ \text{median}(l) \end{bmatrix}. \tag{4.3}$$

These features are designed for general topological errors. For instance, over-segmentation may result in small facet areas and small angles between their normals. Conversely, an undersegmented facet would have a large area. Later on, the importance of these features will be discussed in details based on experimental results.

Each building M can consequently be characterized by a geometric feature vector that accounts for its geometric characteristics:

$$v_{\text{geometry}}(\mathsf{M}) = \begin{bmatrix} \chi\left(\left(d\left(f\right)\right)_{f\in\mathsf{F}_\mathsf{M}}\right) \\ \chi\left(\left(\mathscr{A}\left(f\right)\right)_{f\in\mathsf{F}_\mathsf{M}}\right) \\ \chi\left(\left(\mathscr{C}\left(f\right)\right)_{f\in\mathsf{F}_\mathsf{M}}\right) \\ \chi\left(\left(\|\mathscr{G}\left(f\right) - \mathscr{G}\left(g\right)\|\right)_{\{f,g\}\in\mathsf{E}_\mathsf{M}}\right) \\ \chi\left(\left(\arccos\left(\vec{n}\left(f\right)\cdot\vec{n}\left(g\right)\right)\right)_{\{f,g\}\in\mathsf{E}_\mathsf{M}}\right) \end{bmatrix}. \tag{4.4}$$

Additionally to individual facet statistics, regularity is taken into account by looking into adjacent graph nodes as in (Zhou et al., 2010). Such features express a limited part of structural information. Dealing with this type of information implies graph comparisons which are not a genuinely simple task to achieve. Since our objective is to build a baseline, this approach has not yet been considered.

4.2.2 Height based features

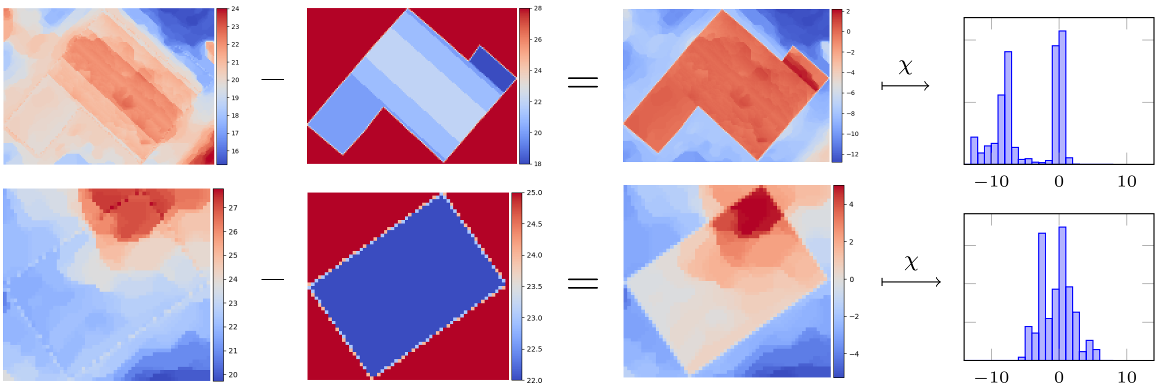
Regarding this modality, raw depth information is provided, for a building model M, by a DSM as a 2D height grid that is cropped to fit the building footprint: $dsm \in \mathbb{R}^{h_\mathsf{M}\times w_\mathsf{M}}$[21]. This type of reference data must date back to the same time where the building models where produced. Otherwise a lot of defects will result simply from change in the scenery.

The DSM is compared to the model height (Brédif et al., 2007; Zebedin et al., 2008). The latter is inferred from its facets plane equations. It is rasterized into an grid structure $alt_\mathsf{M} \in \mathbb{R}^{h_\mathsf{M}\times w_\mathsf{M}}$ using the same spatial resolution as dsm_M. The difference between the two height grids provides a discrepancy map as shown in Figure 4.2c). A baseline approach is herein proposed relying on the statistics of pixel values computed using the χ functions (cf. Figure 4.2).

$$v_{\text{height}}\left(\mathsf{M}\right) = \chi\left(dsm_\mathsf{M} - alt_\mathsf{M}\right). \tag{4.5}$$

The histogram could actually be computed for the building alone without taking into account the terrain arround it. However, since reference data is unavailable, cropping out the terrain height values implies that the building model footrpint is flawless, which is not the case. As a consequence, the heigth discrepancies around the building model are also computed in order to provide some information on the footprint shape, and hence detect `BIT` or `BIB` errors.

[21] w_M (*resp.* h_M) is the grid width (*resp.* height) and is determined by the size of the building and the resolution of the DSM.



(a) DSMs

(b) Model heights

(c) Residuals

(d) Histogram

Figure 4.2: Illustration of the baseline for height based features. First, residuals (cf. Figure 4.2c) are computed by substracting the model height maps (cf. Figure 4.2b) from the DSMs (cf. Figure 4.2a). Histograms are then computed out of these residuals and taken as feature vectors as shown in Figure 4.2d.

Equation 4.5 summarizes how building height based features are computed. Different from a RMSE (Lafarge et al., 2012; Poullis, 2013), the histogram captures the discrepancy distribution, which is particularly helpful in detecting undersegmentation defects or geometric imprecision. However, as for the previous geometric attributes, the grid structure information coming from the model is lost. Errors cannot be spatialized and linked to a specific facet.

4.2.3 Image based features

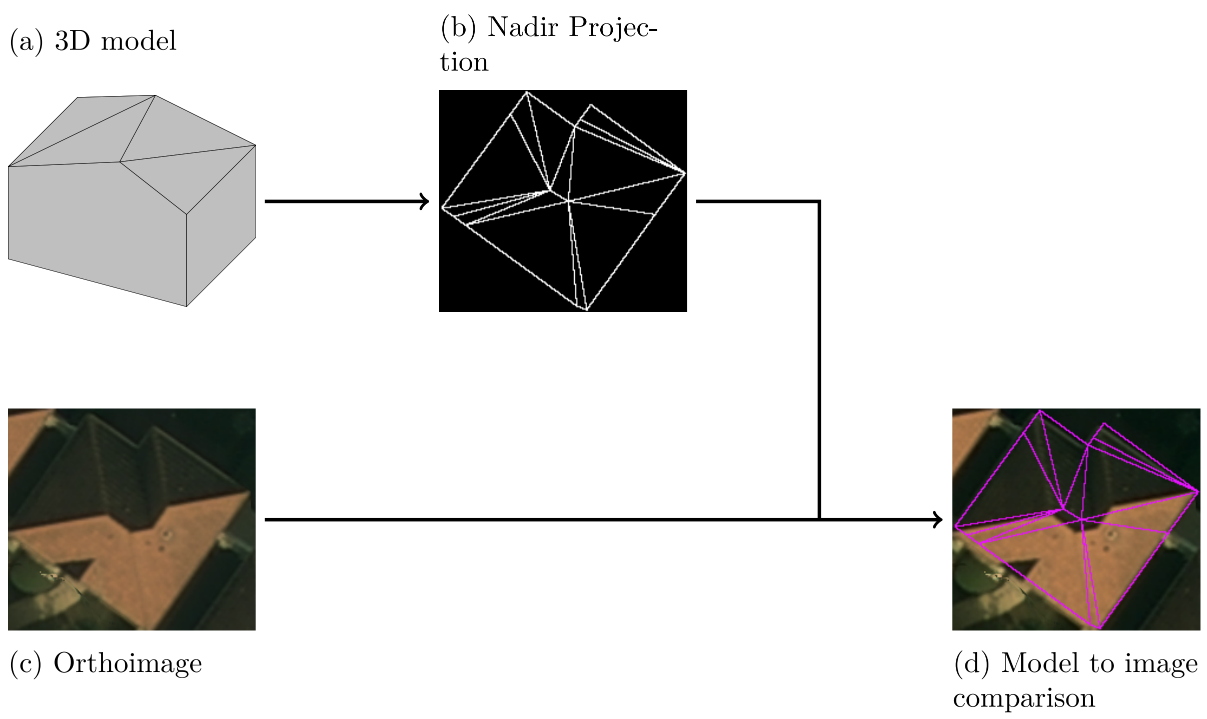
The framework is general enough to encompass both orthorectified images and overhead ones. For now, we rely only on more accessible orthorectified images, eventhough they can be riddled with artifacts. In an ideal scenario, using oriented images is better for edge verification (as already shown in (Michelin et al., 2013)) as orthoimages are a byproduct of earlier ones. However, in practice, overlapping overhead imagery would give rise to other issues, especially, registration problems.

We aim to benefit from the high frequencies in VHR optical images. Building edges, like any image edge, correspond to sharp discontinuities in images (Ortner et al., 2007). The latter are detected using gradient filters on images. Gradient based features are advantageous compared to any radiometry based ones. This is due to the fact that the latter are much less invariant to changes in the than the first ones. Indeed, classic Computer Vision techniques rely on gradient based features: (Lowe, 2004; Dalal et al., 2005). This is adequate to our case where models and images are part of large heterogeneous datasets.

We apply this to our context by comparing building model edges to local image gradients. We start by projecting building models in the nadir direction, as shown in Figure 4.3. For each facet $f \in \mathsf{F}_\mathsf{M}$, this operation takes into account occlusions and results in:

A 2D polygon: If the facet is not vertical (i.e., $\vec{n}(f) \cdot \vec{z} \neq 0$) and not completly occluded by other facets;

A segment: If the facet is vertical (i.e., $\vec{n}(f) \cdot \vec{z} = 0$);


(a) 3D model

(b) Nadir Projection

(c) Orthoimage

(d) Model to image comparison

Figure 4.3: Nadir projection of 3D models to image comparison. The input model is projected in the nadir direction. It is then superposed on the orthoimage and compared to it.

An empty polygon: If the facet is completly occluded by other facets.

The last two cases are filtered out and the final projection consists in a set of polygons forming a partition of the building footprint denoted:

$$\mathsf{F_M}^q \triangleq \{q(f) : f \in \mathsf{F_M}\} \tag{4.6}$$

where:

q is the function that yields the projection of a facet if it is a polygon or an empty polygon othewise.

This projection is compared with a corresponding orthorectified image $I_\mathsf{M} \in \mathbb{R}^{h_\mathsf{M} \times w_\mathsf{M} \times 3}$. For each facet projection, we isolate each edge s (Figure 4.4a). In an ideal setting, gradients computed at pixels $g \in I_\mathsf{M}$ that intersect s need to be collinear with its normal $\vec{n}(s)$. In consequence, applying a statistics functions χ[22], we compute a distribution of the cosine similarity between the local gradient and the normal all along each edge s (cf. Figure 4.4b):

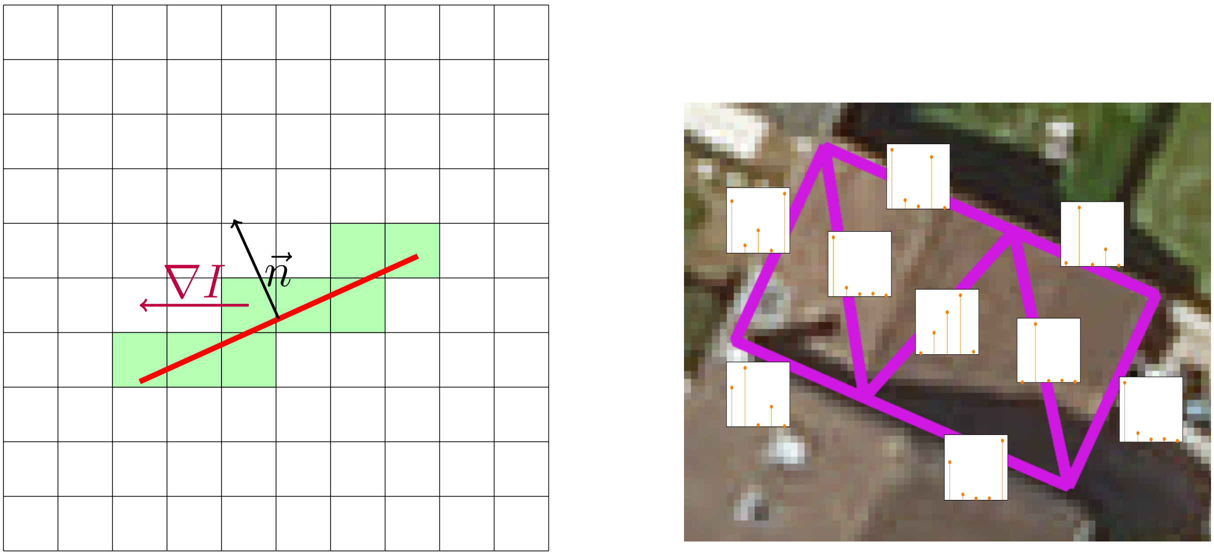
$$\mathsf{D}_\chi(s, I_\mathsf{M}) \triangleq \chi\left(\left(\frac{\nabla I_\mathsf{M}(g) \cdot \vec{n}(s)}{\|\nabla I_\mathsf{M}(g)\|}\right)_{\substack{g \in I_\mathsf{M} \\ g \cap s \neq \emptyset}}\right). \tag{4.7}$$

For each polygon $f^q \in \mathsf{F_M}^q$, the distributions over all its edges[23] $s \in f^q$[24] are stacked to yield a distribution over the whole projected facet (cf. Figure 4.5a). In the case of histograms $\chi^p_{\text{histogram}}$ with the same parameters p (and thus the same bins), it is equivalent to summing out the previous vectors $\mathsf{D}_{\chi^p_{\text{histogram}}}(s, I_\mathsf{M})$. In order to take into account the

[22] For instance, the functions defined in Equations 4.3 or 4.2.

[23] The empty polygons are ignored as they have no edges.

[24] Abuse of notation.



(a) Local gradients (in purple), on intersecting pixels (in green), are compared to the edge (in red) normal (in black).

(b) Histograms describing the similarity between edges and the orthoimage.

Figure 4.4: Illustration of how edges from the projected model are compared to the orthoimage.

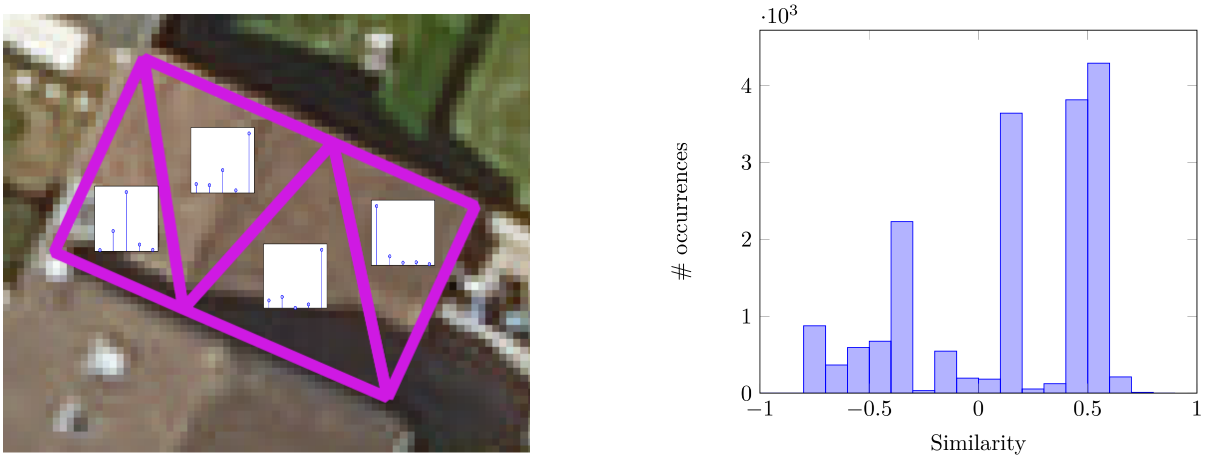
variability of segment dimensions, this sum is weighted by segment lengths.

$$D_{\chi^p_{\text{histogram}}}\left(f^q, I_{\mathsf{M}}\right) \triangleq \sum_{s \in q(f)} \|s\| \cdot \mathsf{D}_{\chi^p_{\text{histogram}}}(s, I_{\mathsf{M}}). \tag{4.8}$$

The same can be done over all facets of a building M (cf. Figure 4.5b). The weights are added in order to take into account the geometry heterogeneity. The gradient to normal comparison is similar to the 3D data fitting term formulated in (Li et al., 2016). Once again, the model structure is partially lost when simply summing (weighted by the projected facet area) histograms over all segments.

$$v_{\text{image}}\left(\mathsf{M}\right) = D_{\chi^p_{\text{histogram}}}\left(\mathsf{M}, I_{\mathsf{M}}\right) \triangleq \sum_{f^q \in \mathsf{F}_{\mathsf{M}}{}^q} \mathscr{A}\left(f^q\right) \cdot \mathsf{D}_{\chi^p_{\text{histogram}}}(f^q, I_{\mathsf{M}}). \tag{4.9}$$

These image based attributes (cf. Figure 4.5) are helpful for precision error detection. As example, facet imprecise borders can be detected as local gradients direction will be expected to differ greatly from the inaccurate edge. It can also be instrumental in under-segmentation detection as colors can change considerably from one facet or one building to another inducing an gradient orthogonal to edge normals.



(a) Depiction of histograms describing the similarity of each of the model's projected facet to the orthoimage.

(b) Illustration of the final histogram describing the similarity between the model edges and the orthoimage.

Figure 4.5: Illustration of the baseline for image based features. The comparison between a model and the orthoimage is conducted by aggregating the edge similarity histograms (cf. Figure 4.4b): First at facet level, which results in a histogram per each of the projected facets (cf. Figure 4.5a), and then at the whole model level, with a final output describing the similarity between the input 3D model and the orthoimage as shown in Figure 4.5b.

4.3 Implementation details

In this section, we give a detailed account of how every ingredient of our pipeline is parameterized. We first start, in Section 4.3.1, by specifying the different feature configurations that are used in the experimental study and how there were implemented. Next, in Section 4.3.2, we discuss in detail how the classification process was conducted.

4.3.1 Feature configurations

We present herein the features that were used in experiments. The main objective herein is to prove the efficiency of the proposed learning framework. Hence, we test different configurations using the baseline features.

The geometric features are intrinsic and are always available. As a consequence, we tested four feature configurations: "geometric features" (**Geom.**) only, "geometric and height features"(**Geom. ⊕ Hei.**), "geometric and image features"(**Geom. ⊕ Im.**) as well as "geometric, height and image features"(**All**). In order to have a better understanding of how important each modality is, their feature vectors are by design drawn to be of the same size: 20.

Actually, using the function $\chi_{\text{max,min,mean,med}}$ defined in Equation 4.3, the geometric feature vector in Equation 4.4 is of dimension

$$\underbrace{4}_{\substack{\text{the output dimension}\\ \text{of the function } \chi_{\text{max,min,mean,med}}}} \times \underbrace{5}_{\substack{\text{the number of the facet}\\ \text{graph attributes}}} = 20.$$

Regarding height based feature vectors, their dimension depends on the histogram parameters. Since we compute differences between observed and model height at terrain

level also (cf. Section 4.2.2), and because these are virtually unbounded and can differ from one scene to another, the maximum possible discrepancy is limited to ±50 m for all scenes. In order to have the same feature vector dimension for this modality as for the geometric one, the number of bins is fixed manually to 20.
For image based features, the cosine similarity between normal vectors, used in Equation 4.8, is by definition bounded in the interval [-1, 1]. This interval is consequently divided evenly into 20 bins for the histogram computation. The DSMs and orthorectified images used to derive height and image features have the same spatial resolution as the reconstruction input data.

These features implementation is conducted in `Python` and is not optimized. Geometric features are computed in average in 0.05 s/building, height based ones take in average 1.4 s/building and finally image based ones need more than 69 s/building. In order to reduce the runtime of each experiment, the last two types of features are cached once computed for a building model and retrieved latter for tests.

4.3.2 Classification settings

Feature extractors being dicussed above, we give now extensive details how the considered classifiers (cf. Section 4.1.3) are applied. We also provide the metrics that will help measure the accuracy of predictions.

Considered labels.

All **eFin** levels were tested. The input models are generalized to LoD-2. As a consequence, we chose **eLoD** = LoD-2. If the **exclusivity** is ON, at **eFin** level 3, the second stage classification results depend on the first one. After a brief experimental study, the first stage did not yield good enough results in order to test this configuration. In fact, we remind the reader that, at this stage, our goal is, first and foremost, to prove the feasibility of the proposed approach. That is why we limited the experiments to the case where the **exclusivity** is set to be OFF.

Classifiers.

As discussed in Section 4.1.3, two classifier types were considered in the experimental study. As already explained beforehand in Section 4.1.3, a one-vs-all approach is used to adapt RFs to multi-label settings. For now, we use only a RF classifier with baseline features to conduct the first sets of experiments since it was the easiest to parameterize. We relied upon `scikit-learn` the already available and ubequitous implementation in `Python` (Pedregosa et al., 2011).

A standard grid search involving a smaller set of building models and only baseline geometric features (Ennafii et al., 2018b) yielded comparable results for the number of trees set in the range 850 to 1000 and a maximum tree depth from 3 to 5. Given the already immense parameter search space involving all possible feature configurations, feature extraction parameters and label possibilities, the RF parameters are set to 1000 for the tree number and 4 for the maximum tree depth for all other experiments without performing any grid search.

Metrics for quantitative assessment.

The overall accuracy is not interesting due to the highly unbalanced label distribution. We prefer reporting recall $\boldsymbol{Rec}$ and precision $\boldsymbol{Prec}$ ratios. As a reminder these metrics are defined as follows:

$$\boldsymbol{Rec} \triangleq \frac{tp}{tp + fp} \tag{4.10}$$

$$\boldsymbol{Prec} \triangleq \frac{tp}{tp + fn}, \tag{4.11}$$

where:

tp : the number of instances predicted positive that are positive in reality;
fp : the number of instances predicted positive that are negative in reality;
fn : the number of instances predicted negative that are positive in reality.

Recall expresses, from a number of samples of a given class, the proportion that was rightfully detected as such. Precision indicates how much samples, amongst the detected ones, were, in truth, part of the studied class (Powers, 2011). We also summarize these two ratios with their harmonic mean, the F-score:

$$\boldsymbol{F_{score}} \triangleq \frac{2}{\frac{1}{\boldsymbol{Rec}} + \frac{1}{\boldsymbol{Prec}}}. \tag{4.12}$$

Unless said otherwise, all experiments were conducted performing a 10-fold cross validation to avoid overfitting or underfitting issues. Only test results are reported.

5 Assessing the learned approach

Contents

In Chapters 3 and 4, we proposed a semantic quality evaluation of 3D building models employing a learning formulation. Herein, we put our pipeline to the test. First, in Section 5.1, we delineate how the building models that are used for the study are obtained. Second, Section 5.2 provides a first experimental outlook of the capacity of our approach to detect errors affecting 3D building models. Third, in Section 5.3, we study the scalability of the proposed approach under different scenarii. Finally, in Section 5.4, we analyse results of the experiments at low **eFin** levels.

5.1 Datasets

In this section, we present the studied building models. Section 5.1.1 describes the selected urban scenes as well as the modeling technique which helped produce the studied models. Next, in Section 5.1.2, we analyse the distribution of the previously defined errors (cf. Chapter 3) over the three urban areas of interest.

5.1.1 Urban scenes and modeling techniques

3D models from three different cities of France are selected in order to assess the performance of our framework: **Elancourt**, **Nantes**, and the XIII$^{\text{th}}$ district of Paris (**Paris-13**) (cf. Figure 5.1). Figure 5.2 depicts the small city of **Elancourt** which contains diverse types of buildings: residential areas with gable and hip roof buildings and districs with large industrial flat roof buildings (cf. Figure 5.5a). **Nantes**, as shown in Figure 5.3, represents a denser urban setting but with a lower building diversity (cf. Figure 5.5b). In Figure 5.4, one can see how **Paris-13** consists of mostly flat roof high towers which coexist with Haussmann style buildings that typically exhibit highly fragmented roofs (cf. Figure 5.5c). The **Elancourt** (*resp.* **Nantes** and **Paris-13**) scene contains 2009 (*resp.* 748 and 478) annotated building models. In order to handle these models, `proj.city`[25], a `C++` library, was developed making use of the `CGAL` library (Fabri et al., 2000). Thanks to this tool, we can project building models in the nadir direction as well as produce the corresponding height maps. The DSM and the orthorectified image spatial resolution is 0.06 m for the first area while it is 0.1 m for the other ones.

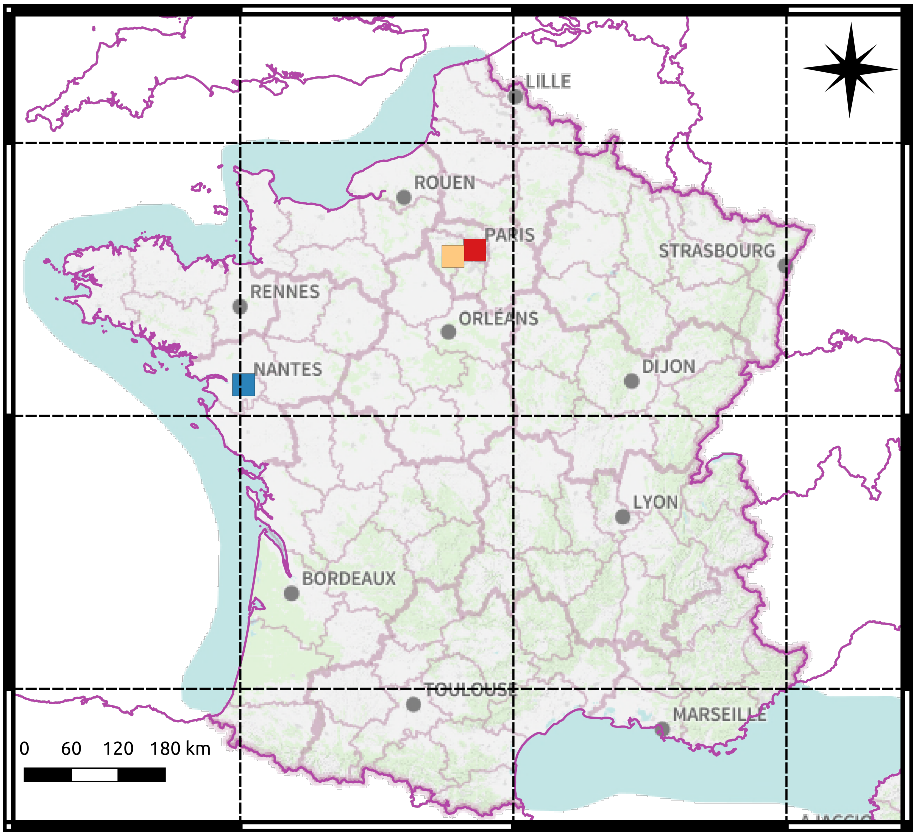


Figure 5.1: Map of France showing the studied urban scenes. Each square corresponds to a city: 🟥 **Paris-13**, 🟧 **Elancourt**, 🟦 **Nantes**.

[25] `proj.city`: `https://github.com/ethiy/proj.city`

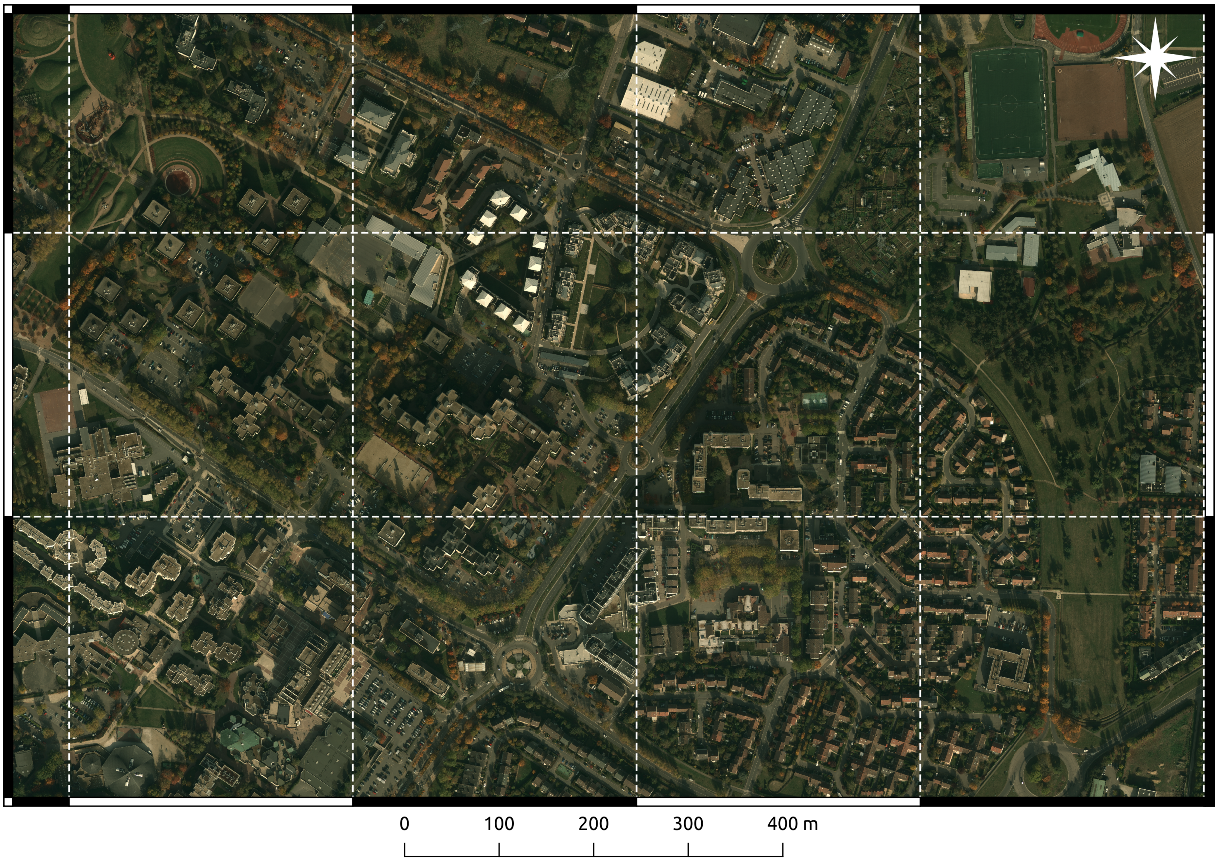


Figure 5.2: Orthoimage showing the diversity of buildings in Elancourt.

Bati3D® models were used in these experiments (cf. Figure 5.6). This product is based on the algorithm described in (Durupt et al., 2006). The latter generates models out of existing building footprints and aerial VHR multi-view DSMs. The modeling algorithm simulates possible roof structures with facets satisfying some geometric constraints. The best configuration is selected using a scoring system on the extrapolated roofs. Finally, vertical building façades connect the optimal roof to the building footprint. These models have a LoD-2 level. This method has been adapted to roof types of low complexity and favors symmetrical models that are common in residential areas. It has been selected to ensure a varying error rate for the three areas of interest, especially since models were generated with partly erroneous cadastral maps. Consequently, the modeling will fail, allowing to tackle the evaluation of the ensuing models. 3235 buildings in total are considered. They were annotated according to the atomic errors list provided by our taxonomy.

5.1.2 Error statistics

Each one of these scenes contains more than 10,000 buildings each. Only a small fraction of these building models were annotated in the aim of building a training dataset. To annotate a building model, the manual operator compares the nadir projection of the model to the corresponding orthoimage and DSM. This was possible thanks to the work[26] of Clémence Chupin, who was a Master student at Ecole Nationale des Sciences Géographique (ENSG). She helped develop a `pyQT` interface in order to ease the task. This

[26] `sGrISner`: `https://github.com/sGrISner/sGrISner`

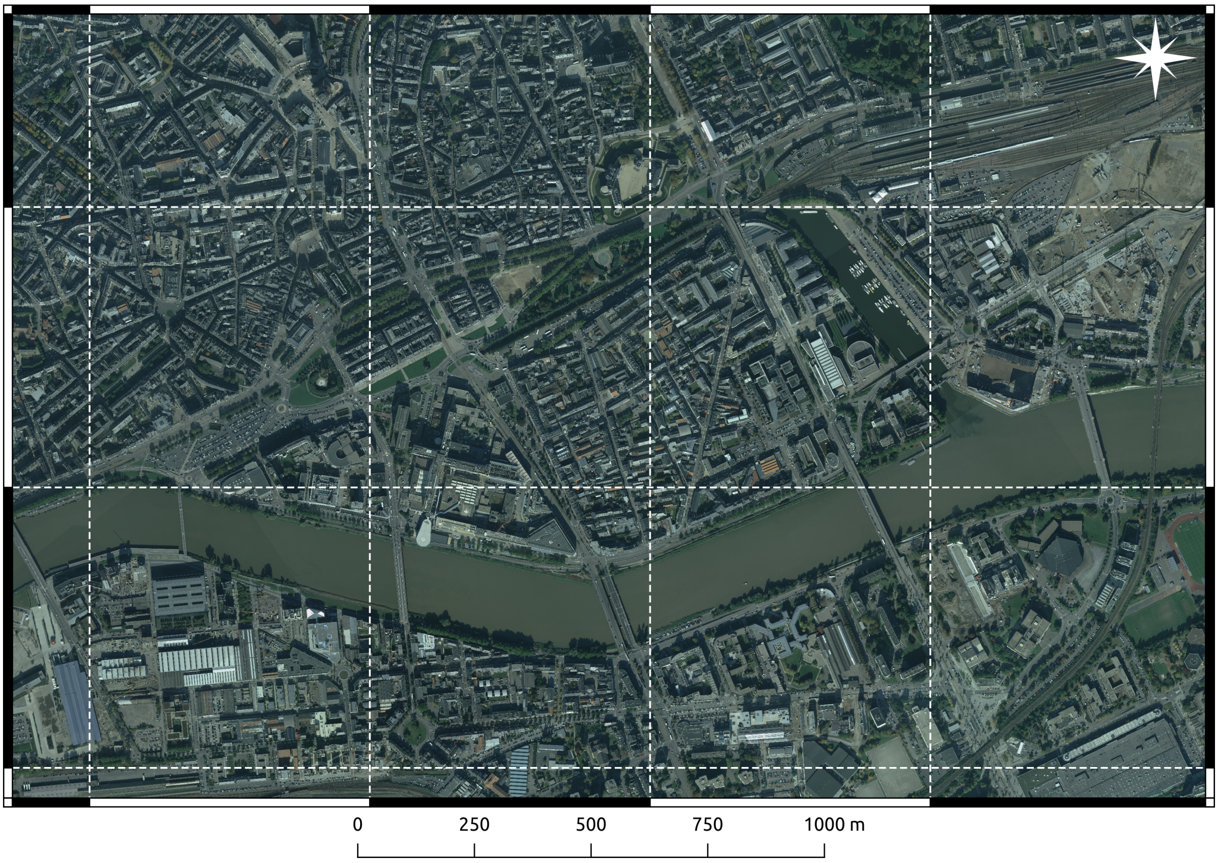


Figure 5.3: Orthoimage depicting the dense city center of Nantes.

allows to switch between data sources and better assess the errors[27] affecting each building. This tool can also be used in the case of active learning where the operator validates or corrects the predictions from a classifier which is connected in the backend. Figure 5.7 shows how the `sGrISner` interface enables the operator to compare the building model to the orthoimage in order to assign the correct labels.

Based on this annotation step modeling errors statistics were computed and compiled. These are depicted in Figure 5.8.

These statistics are first analysed depending on the family errors at the `finesse` $= 2$ level (cf. Figure 5.8a). Due to the fact that geometrically inconsistent 3D models were filtered out in a preprocessing (nadir projection) step, `Unqualifiable` buildings represent a small fraction of the dataset (less than 7.5 %). Actually, the latter corresponds to the, partially or completly, occluded buildings that could not be qualified. Moreover, only a small fraction of buildings are `Valid`: 57 (2.84 %) in **Elancourt**, 55 (7.35 %) for **Nantes** and 21 (4.39 %) in **Paris-13**. Most buildings are affected by the `Building Errors` family (over 58.16 %) and the `Facet Errors` one (over 75.94 %).

At the **eFin** level 3, two axes of analysis are possible. First, we group errors that are very frequent in the dataset. Over-segmentation errors (`FOS` and `BOS`), in both error families, are well represented ranging from 38.9 to 66.8 %. The same is true for `FIG` with a frequency from 59.8 to 80 %. `FIT`, on the other hand, are very rare in all the areas

[27] There can be multiple errors per building.

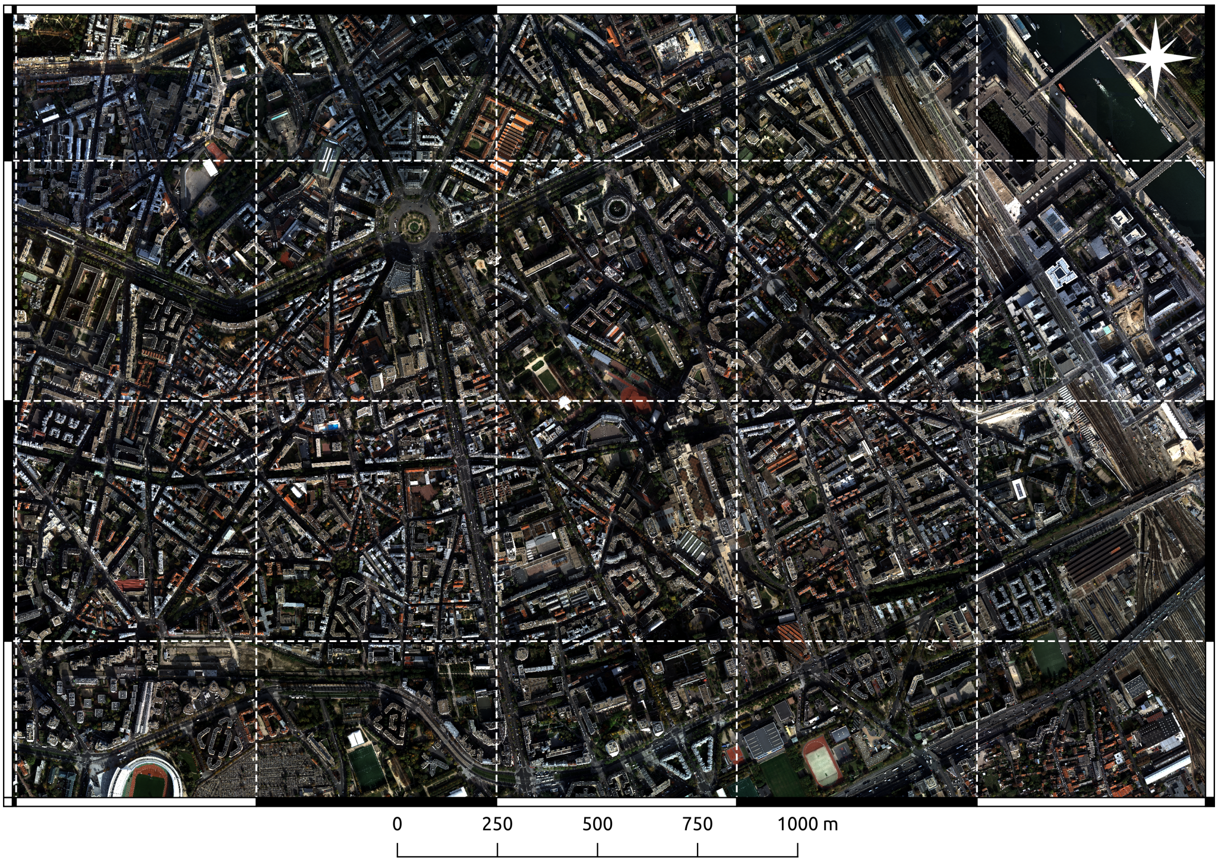
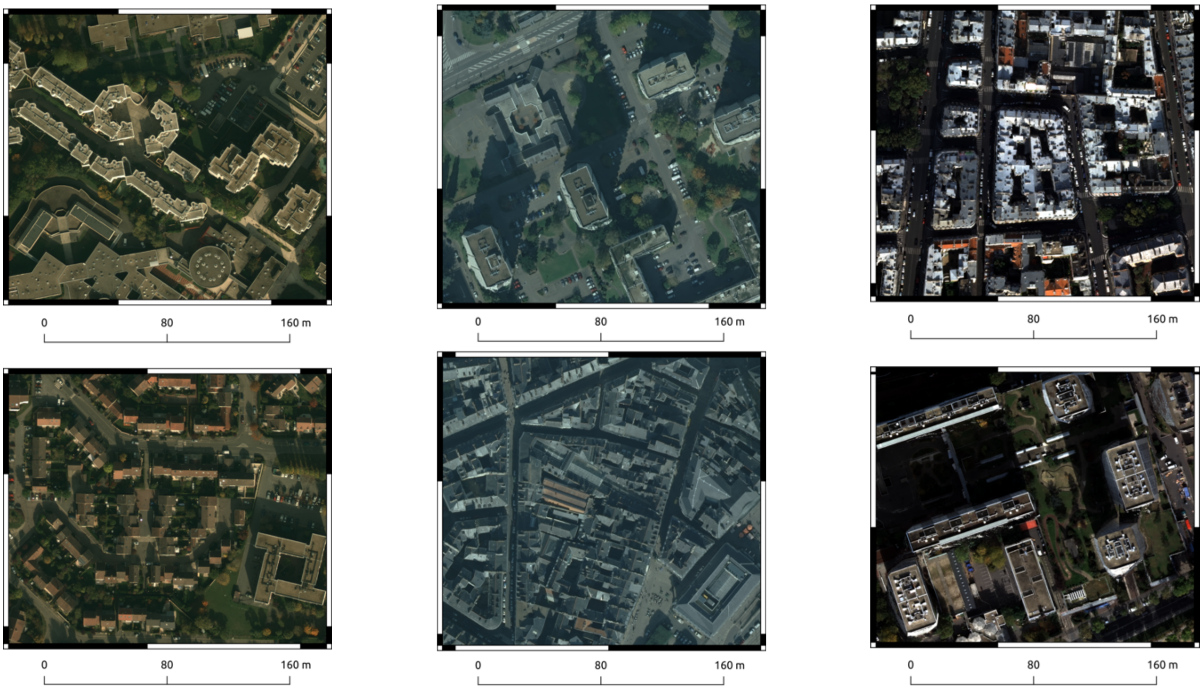


Figure 5.4: Orthoimage showing the heterogeneity of the XIII$^{\text{th}}$ district of Paris.

with ratios a little less than 1.5 %. The rest have a presence ratio within the percentage interval of [10, 30]. The errors that are rare are understandably going to have a negative impact on the learning process.

Secondly, we can compare error frequency discrepancies depending on the studied scene. **Elancourt** is different compared to the relatively close sets **Nantes** and **Paris-13**, with regards to BOS, BUS and BIT. In fact, the last two areas are more densely urbanized than the first one exhibitting the same properties at LoD-0. BIB, on the other hand, are equally distributed over the different datasets as this error type depends mostly on the input sensor data resolution independently of building types.
At the facet level, FIT is also equally occuring across all the scenes different from the rest of Facet errors. In fact, FOS occurence ratio is related to the size of facets in urban scenes. Actually, the less complex roof structures are, the more big are facets and the more chance they have to be over segmented. Indeed, **Elancourt**, **Nantes** and **Paris-13** scenes are ordered in an ascending manner of their roof structure complexity. Conversely, in line with the previous analysis, FUS are less present in **Elancourt** than in **Nantes** which, in turn, contains less of the same error than **Paris-13**. FIB is distributed in the same manner as FUS. This is mainly due to the fact that the more a roof structure is complex the more precision errors are possible. FIG does not keep the same dynamic as its frequency keeps stable from **Elancourt** and **Nantes** but jumps considerably in **Paris-13**. This may be explained by the fact that the gap in FUS error ratios between **Paris-13** and **Nantes** is more important than that of FOS errors, contrarily to the same gaps for the same errors between **Nantes** and **Elancourt** which compensate each other.

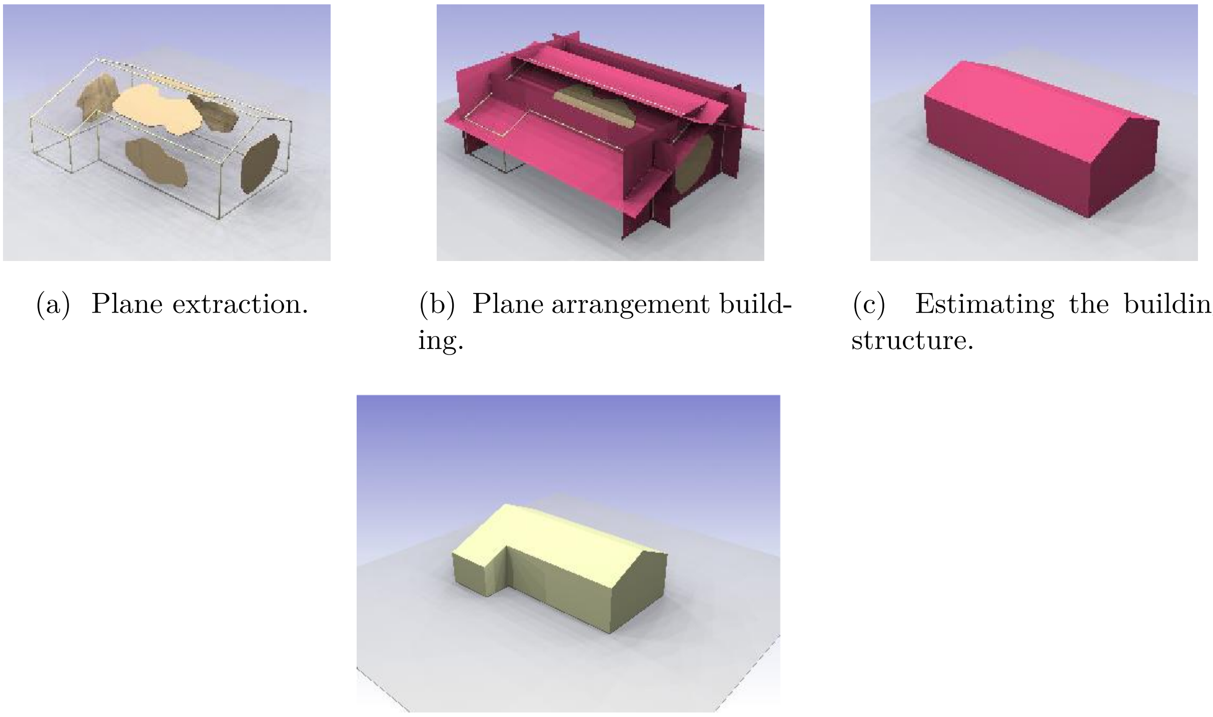


(a) Elancourt contains flat roof buildings (top) as well as gable roof ones (bottom).

(b) Nantes exhibits high rising towers (top) along densely packed fragmented roof buildings (bottom).

(c) The XIIIth district of Paris is made of Haussmannian buildings (top) and high rising towers bottom).

Figure 5.5: Samples of building types per area of interest.



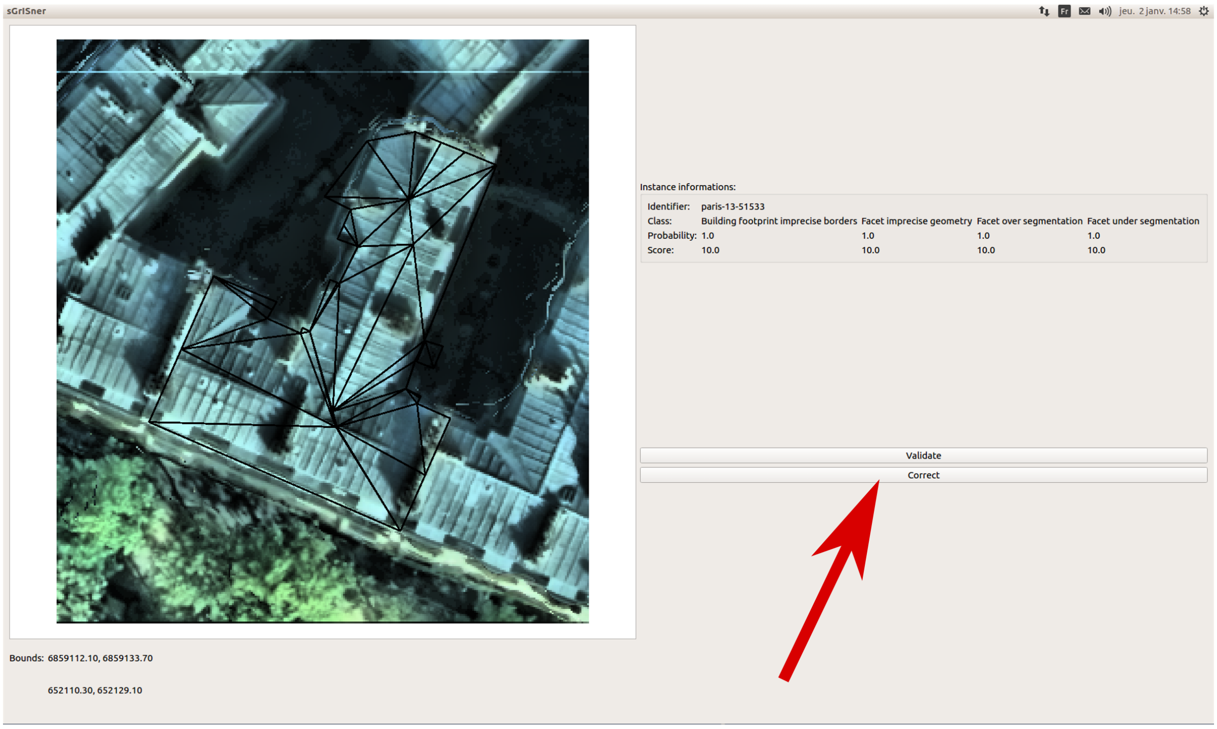
(a) Plane extraction.

(b) Plane arrangement building.

(c) Estimating the building structure.

(d) The ground truth model.

Figure 5.6: Illustration of how the modeling steps of the approach presented in (Durupt et al., 2006). A first step consists of plane extraction (cf. Figure 5.6a). Out the resulting plane arrangements (cf. Figure 5.6b), the most plausible building structure is chosen (cf. Figure 5.6c). Figure 5.6d shows the ground truth model. Images taken from (Brédif, 2010).

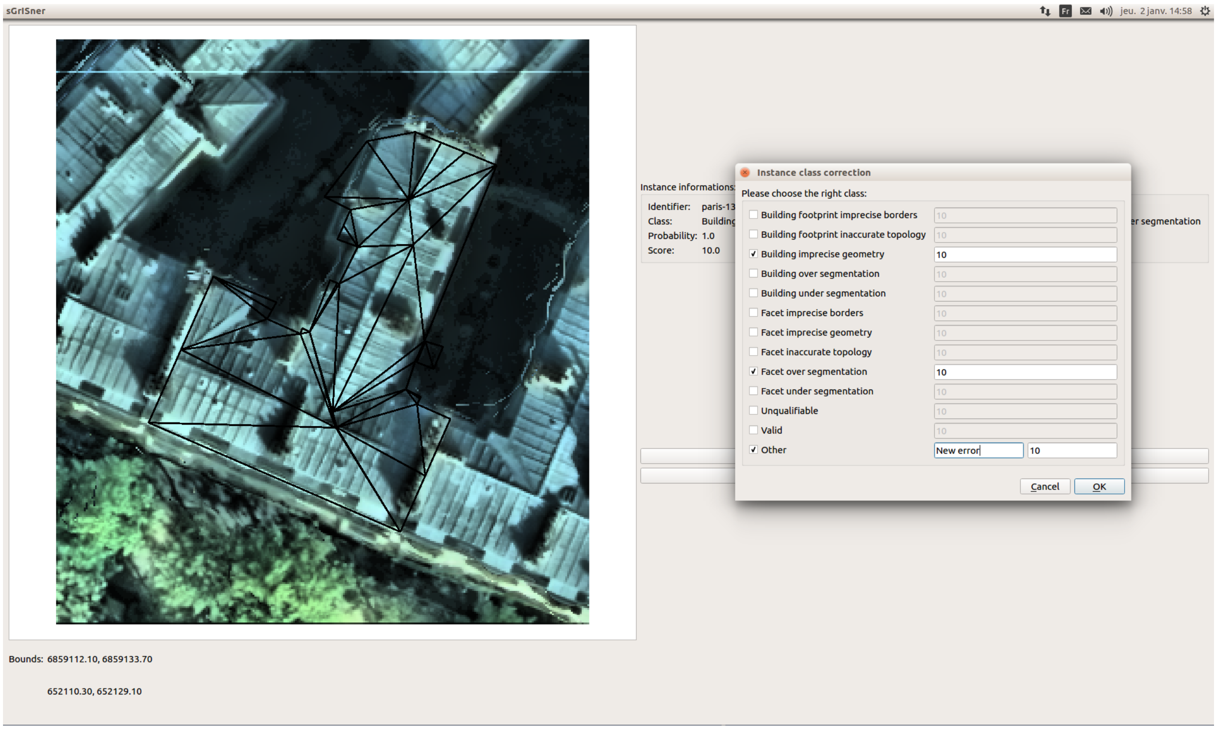


Instance informations:

Identifier:	paris-13-51533			
Class:	Building footprint imprecise borders	Facet imprecise geometry	Facet over segmentation	Facet under segmentation
Probability:	1.0	1.0	1.0	1.0
Score:	10.0	10.0	10.0	10.0

Bounds: 6859112.10, 6859133.70
652110.30, 652129.10

(a) The model is nadir projected and superposed to the orthoimage. The operator either validates the error list or ammends it (red arrow).



Instance class correction

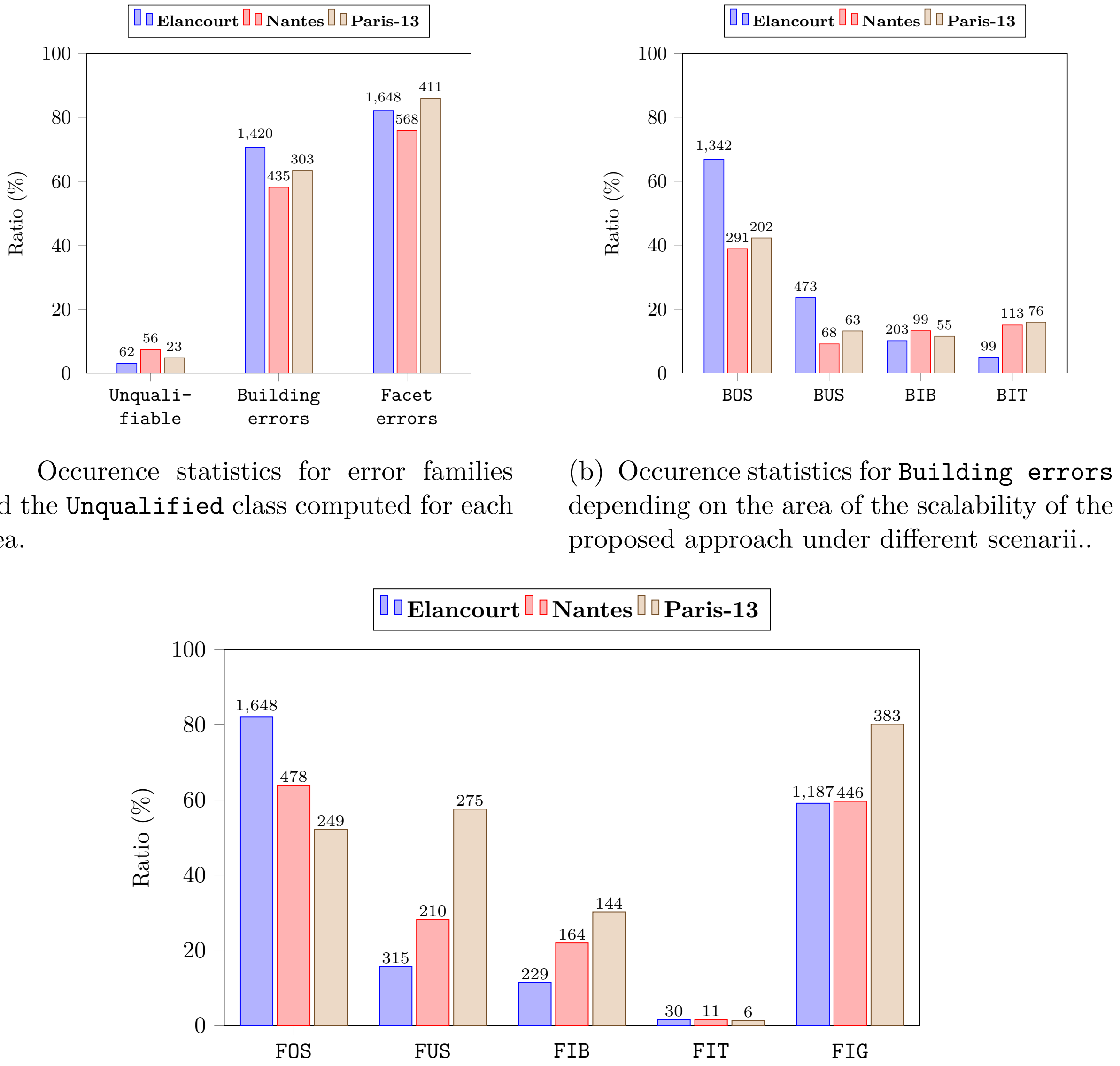
Please choose the right class:

- ☐ Building footprint imprecise borders 10
- ☐ Building footprint inaccurate topology 10
- ☑ Building imprecise geometry 10
- ☐ Building over segmentation 10
- ☐ Building under segmentation 10
- ☐ Facet imprecise borders 10
- ☐ Facet imprecise geometry 10
- ☐ Facet inaccurate topology 10
- ☑ Facet over segmentation 10
- ☐ Facet under segmentation 10
- ☐ Unqualifiable 10
- ☐ Valid 10
- ☑ Other New error 10

Bounds: 6859112.10, 6859133.70
652110.30, 652129.10

(b) The operator can ammend the error list from an established taxonomy or add a new error label if need be. They can also assign confidence scores to each label.

Figure 5.7: Screenshots from the used annotation tool.



(a) Occurence statistics for error families and the `Unqualified` class computed for each area.

(b) Occurence statistics for `Building errors` depending on the area of the scalability of the proposed approach under different scenarii..

(c) Occurence statistics for `Facet errors` depending on the area of interest.

Figure 5.8: Detailed error statistics depending on the urban scenes. The height of bars indicates the frequency of each errors while the number of occurences is displayed above the bars.

5.2 Baseline feature analysis

In the previous sections, we have shown how our dataset of building models was assembled and how the urban scene composition influences modeling error statistics. We also specified in detail how each block of the pipeline was set up. At this stage, our aim is to prove the feasablity of the approach proposed in Chapter 4. As a consequence, for now, we limit ourselves to using the baseline features.

First, in Section 5.2.1, the RMSE is proven to be inadequate for detecting errors in our taxonomy. Secondly, prediction results from all possible configurations of baseline features are compared in Section 5.2.2. Third and last, Section 5.2.3 concludes the analysis by studying the feature importance, for all training zones.

5.2.1 Root Mean Square Error predictive capacity

The Root Mean Square Error (RMSE) is the standard measure in most of 3D modeling methods. As a consequence, we use it herein as a reference that our baseline is to be compared to. We train the classifier on Elancourt with the one dimensional feature vector containing the RMSE. This scene was sufficient enough for our analysis. Mean test results are shown in Table 5.1.

	`BOS`	`BUS`	`BIB`	`BIT`	`FOS`	`FUS`	`FIB`	`FIT`	`FIG`
Rec	99.55	0.21	0	0	98.68	0.63	0	0	98.15
Prec	68.78	33.33	—	0	66.60	0.25	—	0	61.15
$\boldsymbol{F_{score}}$	81.35	0.42	0	0	79.52	1.24	0	0	75.36
Acc	68.46	75.65	89.57	94.66	66.36	83.62	88.24	98.36	60.86

Table 5.1: **eFin** 3 error prediction results using the RMSE on **Elancourt**. ***Acc*** expresses the overall accuracy ratio.

We can distinguish two groups of errors:

- `BOS`, `FOS` and `FIG`: these have a high recall and a low precision and overall accuracy;
- `BUS`, `BIB`, `BIT`, `FUS`, `FIB` and `FIT`: these have low recall and precision ratios.

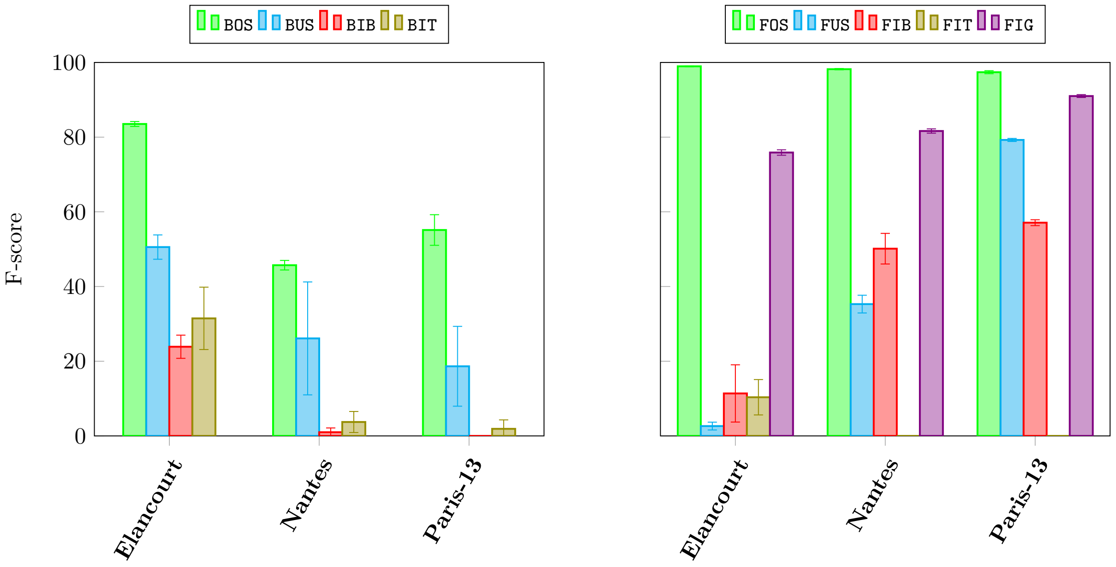
The first (*resp.* second) group coincides exactly with errors that affect more (*resp.* less) than half of the buildings. For this kind of errors, the classifier assigns to almost all samples the positive class. In fact, we end up with a high ratio of false positives (false alarms) and hence a high recall ratio that is coupled with a weak precision and overall accuracy. Exactly the inverse happens with the rest of the errors as we obtain a high percentage of false negative. We can safely conclude that the RMSE is not able to detect errors defined in our taxonomy.

5.2.2 Vanilla experiments

We tested the different feature configurations, at **eFin** level 3 and in all urban zones[28]. Mean precision and recall test results are reported in Table 5.2.

[28] We train and test on different folds of the same scene.

In Table C.2, we can see how FOS and FIG are always well detected. This is mainly due to the fact that they are very frequent in all datasets (cf. Figure 5.8). The same reason explains why BOS is well detected in **Elancourt**, as well as FUS and FIB are in **Paris-13**. Understandably, rare errors are difficult to detect as shown by FIT on the three zones for instance.



(a) Building errors.

(b) Facet errors.

Figure 5.9: Visualizing mean F-score and standard deviation for the feature ablation study. Details are reported in Table C.3.

F-scores are averaged across all feature configurations and represented in Figure 5.9. The first thing we can observe is the fact that geometric features alone give comparable results to ones where extrinsic modalities are added. This was the case for most errors as confirmed by the low variance in F-score. However, in some exceptional cases, adding more modalities impacts greatly the results as shown in Table C.2. The first case regards BUS. In fact, on **Elancourt**, height and image based features both contribute to an increase of around 6 % in F-score. This can be explained by the fact that a lot of under-segmented building models have different heights which reflects easily on height based features. Similar behaviour occurs for **Nantes** and **Paris-13** with image based features driving at least a 20 % jump in F-score. In fact, buildings could be identified by their roof texture or color, as depicted in Figure 3.3. As a consequence, image based features are instrumental in detecting the discrepancy between two under-segmented buildings, especially in dense uniform settings. The second and less important example is FIB on **Nantes** where image based features adds around 10 % in F-score as designed in Section 4.2.3. Last comes the case of BIT, on **Elancourt**, which performs at least 6 % worse when adding more modalities.

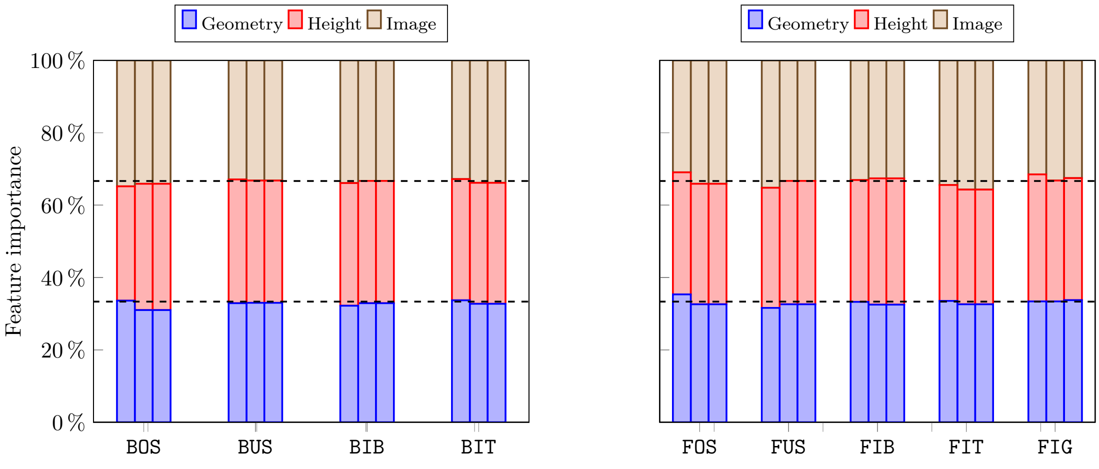
This can be actually interpreted by the fact that geometric features, take into account intrinsic attributes of a building model. These features could, in turn, be intuitively related to the type of building they describe. Consequently, the big role that this kind of features plays in detecting errors implies that the existence of a defect is highly correlated to the type of model it affects.
This is, in fact, expected for errors of topological nature as the geometric structure of the model is assessed. As an example, we can see how BOS is better detected on **Elancourt** and **Nantes** based on geometric features only. Extrinsic features, on the other hand, act more as clues in predicting these types of errors as discussed with BUS in the previous

paragraph.

Conversely, the image and height based features are expected to yield at least as good results as the intrinsic features alone for fidelity defects. In fact, in Table 5.2, `BIB` as well as `FIB` and `FIG` are better detected, in terms of F-score, when adding height or image based modalities. The exception is `FIB` and `FIG` on **Elancourt** where only geometric features yielded better results. This may be explained by the fact that the baseline features are not rich enough and the shear number of training instances in **Elancourt** was sufficient to link building model types to the occurence of such errors.

Figure 5.9 shows that all `Building errors` labels are better detected on **Elancourt**. Moreover, we can even establish that the more the urban scene is dense the more difficult LoD-0 ∪ LoD-1 errors detection becomes. There is one exception where `BOS` existence is slightly better predicted on **Paris-13** than on **Nantes**. This may be attributed to the low number of training samples on both these zones, as they are not sufficiently representative of the actual scenes. On the other hand, `Facet errors` F-scores are proportional to the density of the studied urban scene, minding two exceptions. `FOS` and `FIT` F-scores are actually stable across the different urban areas suffering only from slight decreases that can be attributed to noise. This can be due to the fact that `FOS` is highly frequent in all urban scenes and so easily predictable that F-scores cannot get any better. Conversely, `FIT` are so rare (cf. Figure 5.8c) their existance is barely predictable.

5.2.3 Feature importance



(a) `Building errors.` (b) `Facet errors.`

Figure 5.10: Modality importance computed by stacking single feature importances retrieved from the RF classifier. The first (*resp.* second and third) column represents **Elancourt** (*resp.* **Nantes** and **Paris-13**).

RF classifiers can easily infer feature importances at training time. These were here computed and aggregated by modality in all urban scenes (cf. Figure 5.10).

At first, we observe how important individual attributes are before being gathered. For geometric features, all attributes are equally important. However, concerning image and height based features, only a few are relevant (i.e., they have a higher feature importance ratio). Indeed, these few attributes correspond to the highest and lowest values of the histograms. As described earlier, image and height features consist of a histogram of distances between the model and the real measured signals: vector cosine similarity, for the first, and the L_2 norm for the last. It is clear that the presence of errors would result in saturating the high values in the histogram, while an absence of defects would imply a big number of low values. This is one intuitive explaination of this observed phenomenon.

In a second time, we notice that no modality is more important than the others, contrarily to what was observed in Table 5.2. In fact, as shown in Section 5.2.2, for most `atomic` errors, test results using geometric features are comparable to those obtained with more modalities. However, during training, all modalities are relevant with importance ratios approximating 1/3 as shown in Figure 5.10. As a consequence, for subsequent experimentations all configurations are taken into consideration.

5.2.4 Summary

In this section, we have used the RF classifier on two types of features: a simple RMSE and the custom build baseline features. From the experimental results, we have learned that:

- The RMSE fails completly to predict errors defined in our taxonomy;
- `Building errors` are better detected on **Elancourt** than on the other scenes;
- `FOS` is very well detected with over 95 % in F-score, over all areas of interest;
- Except `FOS`, the more the urban zone is dense the better `Facet erros` are detected;
- Rare errors such as `BIT` and `FIT` (on all urban areas) are poorly detected;
- Geometric features alone are, with a small margin, as good as the other extrinsic based configurations for error detection;
- Consequently, the building type is a good indicator for error detection in the same area;
- `BUS` is the only error that the other modalities are better at with a large gap;
- Although, in most cases, not improving error prediction, external data based modalities are as important as geometric features.

Elancourt								
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	**93.96**	**76.15**	91.43	77.76	91.51	76.08	90.83	76.14
BUS	32.98	76.47	**41.86**	**75.57**	40.38	71.00	39.32	71.81
BIB	12.32	67.57	12.81	68.42	**16.26**	**67.35**	16.75	68.0
BIT	**25.25**	**92.59**	20.20	90.91	20.20	95.24	11.11	91.67
FOS	98.91	99.07	**98.91**	**99.30**	98.99	98.84	98.91	98.84
FUS	**1.90**	**54.55**	0.63	66.67	1.61	50	1.27	66.67
FIB	**9.17**	**87.5**	0	—	8.30	82.61	7.42	100
FIT	6.67	100	**8.73**	**95.24**	3.33	100	3.33	100
FIG	**80.54**	**73.14**	80.45	72.62	78.69	72.12	79.02	71.82

Nantes								
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	**38.14**	**61.67**	36.43	60.23	36.77	62.21	34.71	60.48
BUS	7.35	62.5	7.35	55.56	**29.41**	**66.67**	26.47	64.29
BIB	0	—	0	—	**1.01**	**50.0**	**1.01**	**50.0**
BIT	1.77	22.22	**3.54**	**44.44**	0	0	2.65	50.0
FOS	**98.54**	**98.13**	**98.54**	**98.13**	98.33	97.92	98.12	97.91
FUS	27.62	55.24	**27.62**	**59.18**	24.76	54.74	23.33	53.85
FIB	37.80	62.0	36.59	63.16	**49.39**	**60.90**	46.39	60.90
FIT	0	—	0	—	0	—	0	—
FIG	86.32	78.09	**86.77**	**78.02**	84.53	78.71	83.86	78.08

Paris-13								
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	45.54	65.25	46.53	68.61	**50.0**	**68.24**	46.53	70.15
BUS	6.35	66.67	7.94	71.43	**22.22**	**77.78**	7.94	62.5
BIB	0	—	0	—	0	0	0	—
BIT	**2.63**	**50.0**	0	—	1.32	50.0	0	0
FOS	97.19	97.19	97.19	97.19	**97.59**	**98.38**	97.19	97.19
FUS	**85.09**	**75.0**	84.36	74.12	85.09	74.52	84.36	74.12
FIB	53.47	62.10	51.39	61.67	**53.47**	**63.11**	52.78	61.79
FIT	0	—	0	—	0	—	0	—
FIG	97.65	84.62	**98.96**	**84.79**	97.65	84.62	**98.96**	**84.79**

Table 5.2: Baseline feature ablation study results preformed on the three areas at **eFin** level 3. Test results are expressed in percentage. All `atomic` errors are considered over all possible configurations. Results in bold indicate the feature configuration with the highest F-score.

5.3 Scalability analysis

In the previous section, error detection was proven to depend on the scene composition. This fact motivates studying training the classifier and testing prediction on different scenes. The goal is to prove the resilience of error detection to unseen urban scenes. As the annotation process requires a lot of effort, this trait is crucial to guarantee the scalability of this method under the **large-scale** constraint.

Different configurations are possible, as depicted in Figure 5.11. In the first type of experiments, we train on one urban scene and test on another one. The goal is to examine the `transferability` of the classifier model. Experimental results are reported and analyzed in Section 5.3.1. In a second configuration, the classifier is trained on two scenes and tested on the last one: the objective is to investigate the classifier `generalization`. The results of such experiments are shown in Section 5.3.2. The last experiment class, whose results are presented in Section 5.3.3, targets the `representativeness` of a single 3-area dataset by trying multiple train-test split sizes.

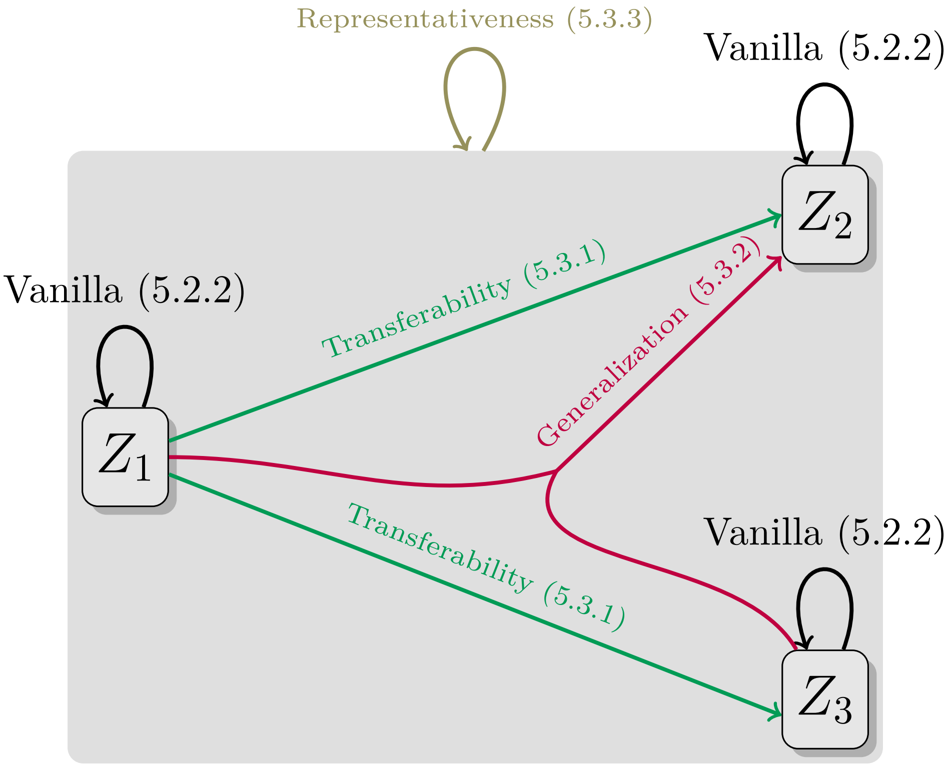


Figure 5.11: A graph representing possible experiments: arrow origins represent training scenes while test ones are depicted as targets. $Z_i, i = 1, 2, 3$ represent the urban zones. All these nodes are assembled in one, meaning that all urban scenes were aggregated in on train/test node. The numbers indicate in which section each experiment is analyzed.

5.3.1 Transferability study

In this configuration, we test how transferable the learned classifiers are from one urban scene to another. We train on a zone Z_i and test on another one Z_j. We will denote each transferability experiment by the couple (Z_i, Z_j) or by $Z_i \to Z_j$. Six transferability couples are possible and are tested with all possible feature configurations. Mean and standard deviation of F-scores are plotted, for each label and experiment, in Figure 5.12. For more details, refer to Tables C.4 and C.5, which compiles all precision, recall and F-score test results.

`Coherence` study. First, a `coherence` analysis is performed. We compare the results of the transferability experiments to the ablation results with the same training scene. This is achieved by looking, for a given area Z_i in all couples $(Z_i, Z_j)_{\forall j \neq i}$, at the differences between Tables C.5 and C.2. The goal is to see to which extent the test and training zone are similar.

All these comparisons are provided in Table 5.3. A color scheme was devised to encode the amplitude of change and is illustrated in Figure 5.13. The resemblance between **Paris-13** and **Nantes** are striking. In fact, when training on one of the two zones and testing on **Elancourt**, out of the nine labels only two are different. `BUS` has a gain between 5 and 15 % when trained on **Paris-13** while its F-score records a drop in the range -15 to -5 % if trained on **Nantes**. `FUS`, in contrast, looses between -45 and -35 % when trained on **Paris-13** and -35 and -25 % for **Nantes**.
Conversely when training on **Elancourt** and testing on the other two areas, the comparisons are not as identical as before. However, from a general standpoint, we can see how the `building errors` are always affected by losses when testing on both **Paris-13** and **Nantes**. Except for `FIT`, the inverse situation is obseved for `Facet errors` where labels are either better detected or stable.

`Projectivity` analysis. Secondly, we investigate how the urban scene composition helps predicting defects in an unseen one. This is called the `projectivity` comparison. For a given test scene Z_j in couples $(Z_i, Z_j)_{\forall i \neq j}$, we compare again results from Tables C.5 and C.2. A gain in F-score for a label can be interpreted by the transferability power of its learning process.

Comparisons are detailed in Table 5.3 with the same color scheme as for `coherence` comparisons. Error family wise, we can summarize these comparisons as follows:

`Building errors:` Out of 20 possible `projectivity` comparisons, 13 yield negative results. This proves how hard it is to transfer learning for this error family.

`Facet errors:` Conversely, only 7 out of 27 `projectivity` scores are worse compared to training and testing on the same test area. One notable pattern is the stability of `FOS` and `FIB` errors no matter the chosen transferability couple.

When looking at the transferability power of `Facet errors`, we can see how training on **Nantes** and testing on **Paris-13** yields the same scores as when the opposite experiment is conducted. We can also see how **Elancourt** is best for learning `FIT` as it transfers well to **Nantes** while having no effect on **Paris-13**. In contrast, both comparisons are negative when training of the last two areas. With the exception of this label, training on the latter dense zones proves to be advantageous than learning on **Elancourt**.
In terms of `Building errors`, clear discernible patterns are hard to find. This agrees with the previous conclusion that this error family is not easy to learn. The only exceptions were `BIT` and `BUS`. For these two labels, training on **Elancourt** proved to be slightly better than on the other scenes.

		BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Coherence	**El. → Na.**	🟥	🟧 **All**	🟥	🟧 **All**	▨	🟩 **Im.**	🟩 **Im.**	▨ **Im.**	🟩
	El. → P13	🟧	🟧 **Im.**	🟧	🟥	▨	🟩 **Im.**	🟩 **Im.**	🟧	🟩
	Na. → P13	🟧	🟥 **Im.**	⬜	🟩 **Geom.**	▨	🟩 **Geom.**	🟩	⬜	🟩 **Hei.**
	Na. → El.	🟩	🟨 **Im.**	🟩 **Im.**	🟩 **Geom.**	▨	🟥 **Im.**	▨ **Im.**	🟩	🟨
	P13 → Na.	🟧	🟧 **Im.**	⬜	🟩 **Geom.**	▨	🟥	▨ **Im.**	⬜	🟧
	P13 → El.	🟩	🟩 **All**	🟩 **Im.**	🟩 **Geom.**	▨	🟫	▨ **All**	🟩	🟨
Projectivity	**El. → Na.**	🟧	🟧 **All**	⬜	🟨 **All**	▨	🟨 **Im.**	🟨 **Im.**	🟩 **Im.**	▨
	El. → P13	▨	▨ **Im.**	⬜	▨	▨	🟥 **Im.**	🟨 **Im.**	⬜	▨
	Na. → P13	🟥	🟧 **Im.**	⬜	🟩 **Geom.**	▨	🟧 **Geom.**	▨	⬜	▨ **Hei.**
	Na. → El.	🟨	🟧 **Im.**	▨ **Im.**	🟧 **Geom.**	▨	🟩 **Im.**	🟩 **Im.**	🟨	▨
	P13 → Na.	🟨	🟥 **Im.**	⬜	🟩 **Geom.**	▨	🟧	▨ **Im.**	⬜	▨
	P13 → El.	🟨	🟧 **All**	🟩 **Im.**	🟧 **Geom.**	▨	🟩	🟩 **All**	🟨	▨

Table 5.3: Evolution of the F-score value, for each error, between transferability experiments and the vanilla experiments (cf. Section 5.2.2). **El.** (*resp.* **Na.** and **P13.**) stand for **Elancourt** (*resp.* **Nantes** and **Paris-13**). Feature sets having a significant impact on the classification results are mentioned in the corresponding cell (cf. Table C.5). The used color scheme is presented in figure 5.13.

Impact of external modalities. As we can suspect from looking at the large standard deviations illustrated in Figure 5.12, modalities play an instrumental role in insuring the transferability experiments. All modalities that stood out, in terms of the F-score, were mentioned in the corresponding cell in Table 5.3.
As intended (cf. Section 4.2.3), image based features are instrumental in transferability for FUS, BUS, FIB and BIB.
Geometric features, on the other hand, play a lesser role helping mostly with BIT transferability when training on dense urban scenes as well as FUS for the **Nantes** → **Paris-13** experiment. Actually, BIT is a purely topological error that can be learned with the help of these structural features. On the contrary, adding more modalities would only confuse the classifier as also seen in the ablation study in Table 5.2.
Height based errors alone are not crucial for transferability, intervening only once in the case of FIB in the **Nantes** → **Paris-13** experiment. However, it has a bigger impact when added to image based ones (cf. cells with **All** in Table 5.3).
All these previous findings further justify why we did not leave out any modality, as they are more frequently critical for transferability than in the ablation study (cf. Table C.2).

5.3.2 Generalization study

We try to find out how omitting one urban zone from the training dataset affects the test results on that same area. Another way to look at it is, from an operational point of view, to find out how much learning on a union of many urban scenes is helpful when testing on an unseen one. Experiments that merge all zones except Z_i ($\bigcup\limits_{\forall j \neq i} Z_j$) for training and test on Z_i are denoted by the couple ($\bigcup\limits_{\forall j \neq i} Z_j, Z_i$) or by $\bigcup\limits_{\forall j \neq i} Z_j \rightarrow Z_i$. There are three possibilities: **Elancourt** ∪ **Nantes** → **Paris-13**, **Paris-13** ∪ **Nantes** → **Elancourt** and **Paris-13** ∪ **Elancourt** → **Nantes**. The F-score evolution, per experiment and label, is depicted in Figure 5.14.

	`BOS`	`BUS`	`BIB`	`BIT`	`FOS`	`FUS`	`FIB`	`FIT`	`FIG`
Elancourt	🟥 **Im.**	🟥 **Im.**	🟥	🟧 **Hei.**	▨	🟩	🟩 **Im.**	🟧	🟩
Nantes	▨ **Im.**	🟧 **Im.**	⬜	🟩	▨	▨	🟨 **Im.**	⬜	▨
Paris-13	🟨	▨ **Im.**	🟩 **Im.**	🟩 **Geom.**	▨	🟥 **Im.**	▨ **Im.**	🟩	🟨

Table 5.4: Evolution of the F-score value, for each error and test area, between generalization experiments and the vanilla experiments (cf. Section 5.2.2). Feature sets having a significant impact on the classification results are mentioned in the corresponding cell (cf. Table C.8). The used color scheme is presented in figure 5.13.

We compare the F-score ratios of these experiments, shown in Table C.8, with ones from the ablation study for each test area. These comparisons are compiled in Table 5.4.

We start by summarizing comparisons with respect to error families:

`Building errors`: Out of the 11 possibilities, 6 yield worse results. This is approximatly 10 % less than the transferability study.

`Facet errors`: 4 out of 14 comparisons exhibit the same trend. This is sensibly the same ratio as the transferability comparisons with only a 2 % jump.

This confirms again how difficult `Building errors` are to learn compared to `Facet errors`. However, this time combining urban scenes yielded better results. We can suspect that this is the consequence of the fact that these studied zones are complementary for learning.

Similarly to the previous study, image and height modalities play a major role in error detection. Image based features are crucial for `FIB`, `FUS`, `BOS`, `BIB` and `BUS` detection (Table 5.4). Height based attributes has always a minor role contributing only once for `BIT` as well as geometric features. Image based errors proves again how instrumental they are in learning, even for topological errors.

Based on the generalization study we can deduce the best zones to train each error label on. In fact, if a zone yields the worst scores it means that it contains the hardest instances that are beneficial for training. Conversely, if taken into account in training, it will help the classifier yield better results when testing on other scenes.

For `Building errors` a clear pattern can be observed compared to the transferability study. **Elancourt** when left out suffers the most. In fact, as **Nantes** and **Paris-13** are very dense, buildings are homogeneous in terms of shape with moslty Haussman style or parallelepiped shaped buildings. On the other hand, **Elancourt** has more heterogeneous buildings allowing a better learning. However, the differences are not that big between the latter and **Nantes** as was proved in the `projectivity` comparisons. **Nantes** has some interesting observations to learn on, especially for `BIB`.

Regarding `Facet errors`, the differences are sharper if we set aside `FOS` as it is always well learned over all sets. For `FUS` it is **Paris-13** that suffers the most when left out. This is understandable as Haussman style buildings are usually under-segmented and hence allow for a better learning than other sets, especially **Elancourt**. **Paris-13** is also the better alternative for `FIG`. Next, we can observe how `FIB` is better learned on **Nantes** as it also presents compact buildings with a lot of small facets that can be easily misestimated. However, `FIT` is the exception as **Elancourt** offers the better alternative, as it comprises various building types that cannot be easily modeled.

5.3.3 Representativeness study

The objective is to find out, after merging all training samples from all datasets, what is the minimal amount of data that can guaranty stable predictions. Figure 5.15 depicts the F-score as a function of training ratios (between 20 and 70 %) and `atomic` errors.

We note the high stability of the F-score. This indicates that having a small heterogeneous dataset is not detrimental to the learning capacity and can sometimes be the most suitable solution. `BOS`, `FOS`, and `FIG` have a standard deviation under 2 %, as opposed to `FIB`, `BIT` and `FIT`. Indeed, the latter vary greatly with respect to the training size, and even a larger standard deviation according to the feature configurations.

However, when looking at the best feature configurations, the variability of the F-scores towards the training size dwindles. Extrinsic modalities play again a big role in these experiments as shown in Table C.11. The issue, in this setting, is the lack of consistency when it comes to which modality is the most important. This may be explained by the fact that training samples do not always contain the same instances.

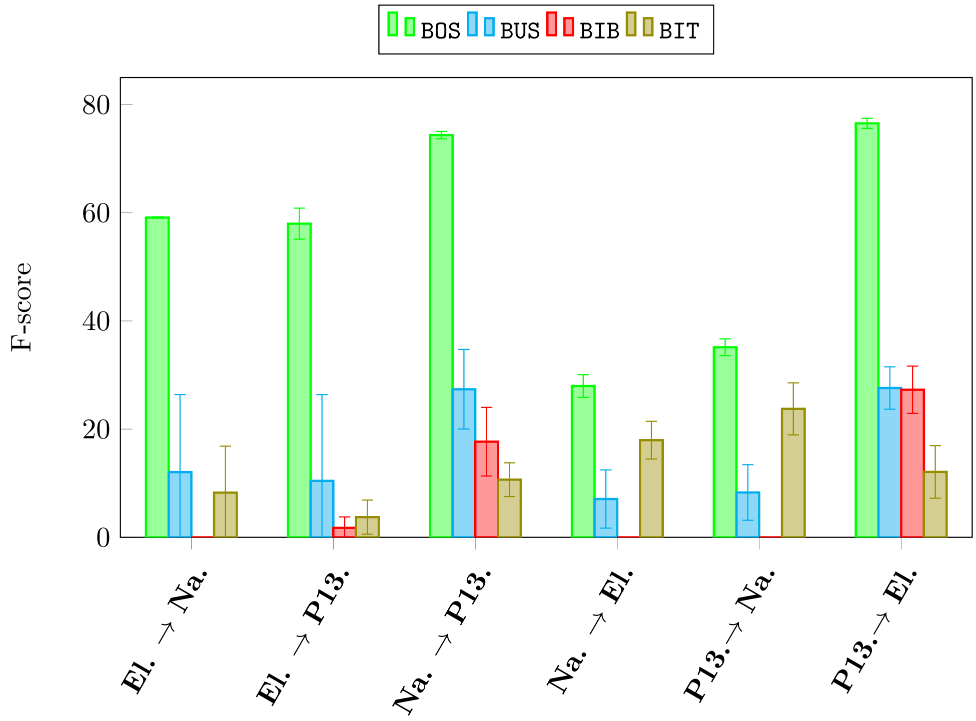
For each error we can then determine the minimum training size needed to achieve stability. As seen `BOS`, `FOS`, and `FIG` are very stable and a 20 % training set is sufficient to retieve similar performance to the initial ablation study. For `BUS` (*resp.* `FUS`), using the best performing features, 20 % is also enough to achieve stability in the range 48 to 52 % (*resp.* 47 to 53 %). `FIB` can also be stabilized between 42 and 46 % but requires a 30 % of instances to train on. However, `BIT`, `FIT` and `BIB` prove to be harder to stabilize with mean F-scores around 16 %, 9 % and 19 % respectively. We can hence observe how, understandably, mixing the training sets produces scores that averages out the ones from the ablation study. On the other hand, more sophisticated features are still required to help predicting the less frequent and more semantic labels.

5.3.4 Summary

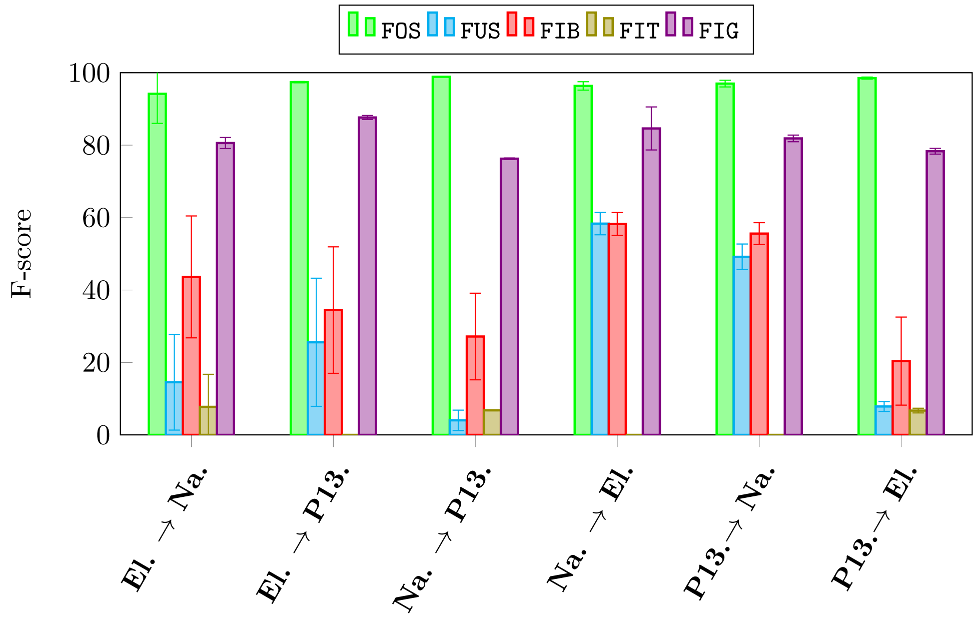
The goal of this section was to prove the scalability of the proposed approach. To achieve this, we tested how well classifiers trained on a source set can score on a different target one. Depending on how the source set was chosen, three types of experiments were examined: transferability, generalization and representativeness. They yielded consistent results which helped prove:

- **Nantes** and **Paris-13** are similar compared to **Elancourt**;
- `FOS` and `FIG` detection is highly transferable with over 95 % in F-score;

- `Facet errors` detection is easier to scale than `Building errors`;
- `Building errors` and `FIT` are better detected on **Elancourt** than on the other scenes;
- Except `FIT`, `Facet errors` are better detected on either **Nantes** or **Paris-13**;
- Rare errors such as `BIT` and `FIT` (on all urban areas) are poorly detected;
- Geometric features are not as good as extrinsic based configurations for error detection scalability;
- Extrinsic features play a more crucial role in transferability and generalization, especially image based ones;
- Representativeness wise, 20 % is sufficient in most error labels for learning.



(a) `Building errors.`



(b) `Facet errors.`

Figure 5.12: Mean F-score and standard deviation for the transferability study. **El.** (*resp.* **Na.** and **P13.**) stand for **Elancourt** (*resp.* **Nantes** and **Paris-13**). This is a vizualization of results reported in Table C.6.

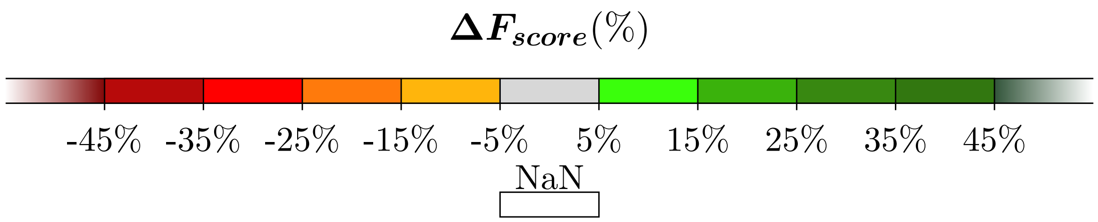
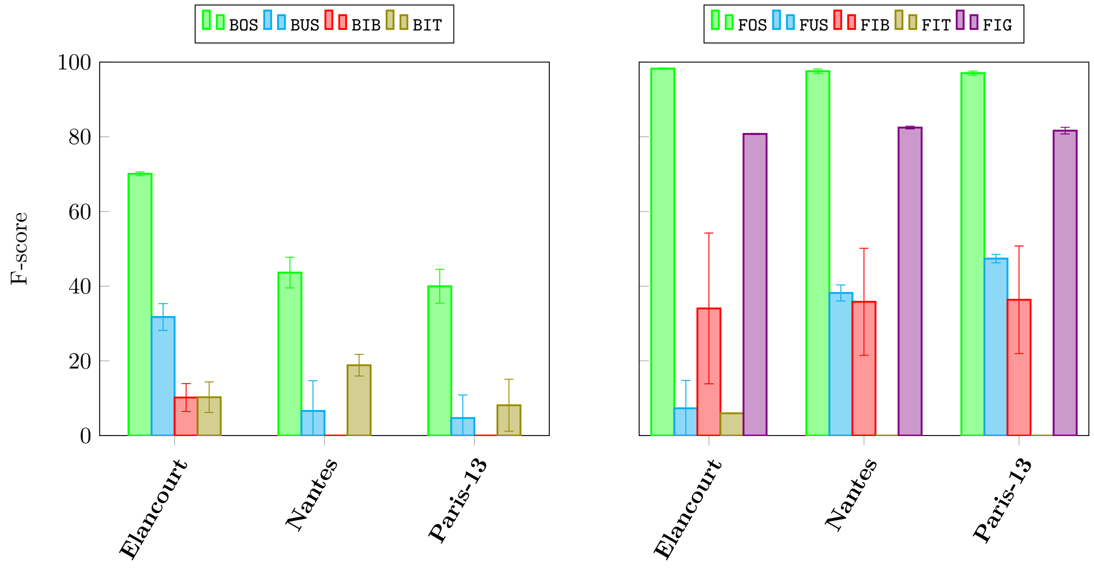


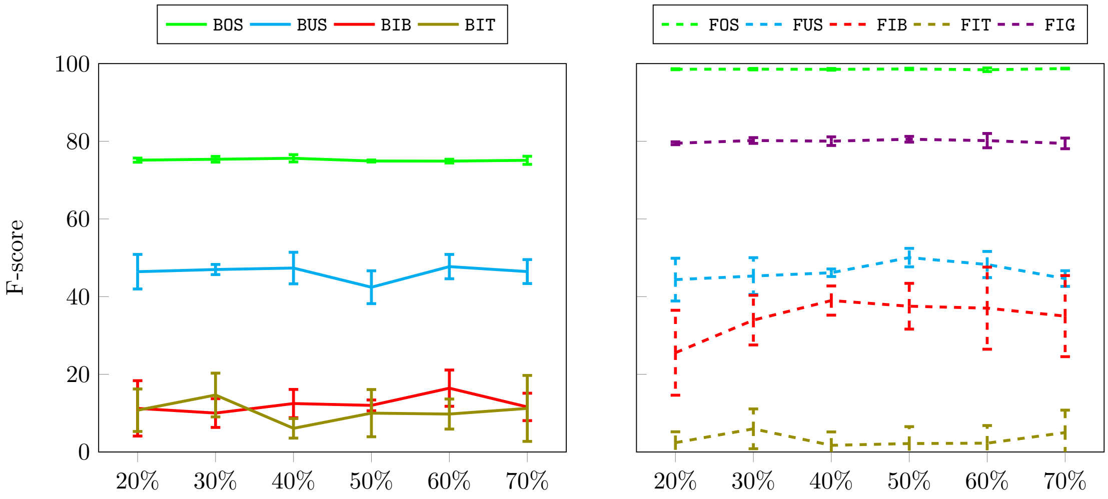
Figure 5.13: Color scheme used for F-score comparisons in this study. When two null F-scores are compared, the cell is colored in white.



(a) Building errors.

(b) Facet errors.

Figure 5.14: Mean F-score and standard deviation for the generalization study per test zone. These are also provided in Table C.9.



(a) Building errors.

(b) Facet errors.

Figure 5.15: Mean F-score and standard deviation for the representativeness experiments depending on the training set size.

5.4 The impact of eFin

In this section, we reproduce the experimental settings described in Section 5.2 and Section 5.3. First, the **eFin** is set to be 2. The goal is to find out how good, transferable and stable are the predictions of modeling error families: `Building errors` and `Facet errors`. Afterwhat, **eFin** is fixed at level 1. The idea is to observe the predictability of defectuous model compared to `Valid` ones.

5.4.1 Error family detection

	Elancourt							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`Building errors`	**99.76**	**85.96**	99.82	85.88	99.88	85.57	**100**	85.55
`Facet errors`	91.79	89.79	92.65	89.40	**93.21**	**89.45**	93.46	89.16

	Nantes							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`Building errors`	85.98	67.27	87.59	67.79	85.75	68.32	**86.90**	**69.23**
`Facet errors`	91.20	94.01	91.37	94.36	91.20	94.35	**91.73**	**94.21**

	Paris-13							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`Building errors`	97.36	68.76	97.36	68.76	97.36	68.76	97.36	68.76
`Facet errors`	99.03	91.26	99.03	91.26	99.03	91.26	99.03	91.26

Table 5.5: Baseline feature ablation study on the three datasets for the **eFin** level 2. Results are expressed in percentage. All four modality configurations are compared across both family errors.

We start by the ablation study. Table 5.5 reveals that adding more remote sensing modalities do not dramatically change the prediction results. This is perfectly illustrated, in Figure 5.16, by the low variance of F-scores for the three areas of interest. We can explain this by referring to the analysis at the **eFin** level 3 in Section 5.2.2. Two main reasons could be noticed:

- Out of all `atomic` errors, only `BUS`, on all datasets, and both, `BIT` and `FIB`, on **Elancourt**, have been greatly impacted by a change in feature configurations. Both occur under 25 %, 5 % and 12 % respectively.
- On the other hand, all the other errors are unaffected, especially, `FOS` and `FIG` which are prevalent on all datasets.

The last observation added to the fact that these labels are, in a large capacity, easily detected individually[29] helps understanding why the F-score reaches at least 90 % for the `Facet errors` family (Figure 5.16a) in constrast with `Building errors`. Moreover, we can see a smaller discrepancy (below 5 %) between F-scores on different scenes for `Facet errors` compared to `Building errors` with 15 %.

The transferability study (Figure 5.16b) compares the F-scores with the ablation study provided in Figure 5.16a. Out of all 12 possible `projectivity` comparisons, only 2 exhibit a decrease in error discrimination. Both happen when training `Building errors`

[29] `FOS` (*resp.* `FIG`) achieves more than 90 % (*resp.* 80 %) in F-score, as depicted in Figure 5.9b.

	Elancourt							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	Valid	***Rec***	Valid	***Rec***	Valid	***Rec***	Valid
Erroneous	99.95	$\frac{1}{57}$	99.95	$\frac{1}{57}$	99.95	$\frac{0}{57}$	99.95	$\frac{1}{57}$

	Nantes							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	Valid	***Rec***	Valid	***Rec***	Valid	***Rec***	Valid
Erroneous	99.84	$\frac{0}{55}$	99.84	$\frac{0}{55}$	100	$\frac{0}{55}$	**100**	$\frac{0}{55}$

	Paris-13							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	Valid	***Rec***	Valid	***Rec***	Valid	***Rec***	Valid
Erroneous	99.77	$\frac{3}{21}$	99.77	$\frac{3}{21}$	99.77	$\frac{3}{21}$	99.77	$\frac{3}{21}$

Table 5.6: Test results expressed in percentage for the **eFin** = 1 case. All four modality configurations, using baseline features, are compared across both family errors.

on **Nantes** and **Paris-13** and testing on **Elancourt** with approximatly −4 % and −3 % respectively. This confirms the fact that **Elancourt** was best for training `Building errors` as established in Section 5.3.2. Same as with the ablation study we can see how the stability in `FOS` and `FIG` is reflected by a stability of `Facet errors`. For this reason, we skip the generalization study, all together, at this section.

The representativeness study conducted for the **eFin** level 2 results in the F-scores that are illustrated in Figure 5.16c. Family detection scores are very stable across all different tested split ratios. Moreover, in contrast to `atomic` errors results (cf. Figure 5.15), F-scores do not vary by more than 1 % in mean and standard deviation. This further proves that at **eFin** level 2, error family prediction is evened out independent of different split ratios, as opposed to higher `finesse` errors. Again, it benefits from the higher heterogeneity of the training set with multiple areas.

5.4.2 Detection of erroneous models

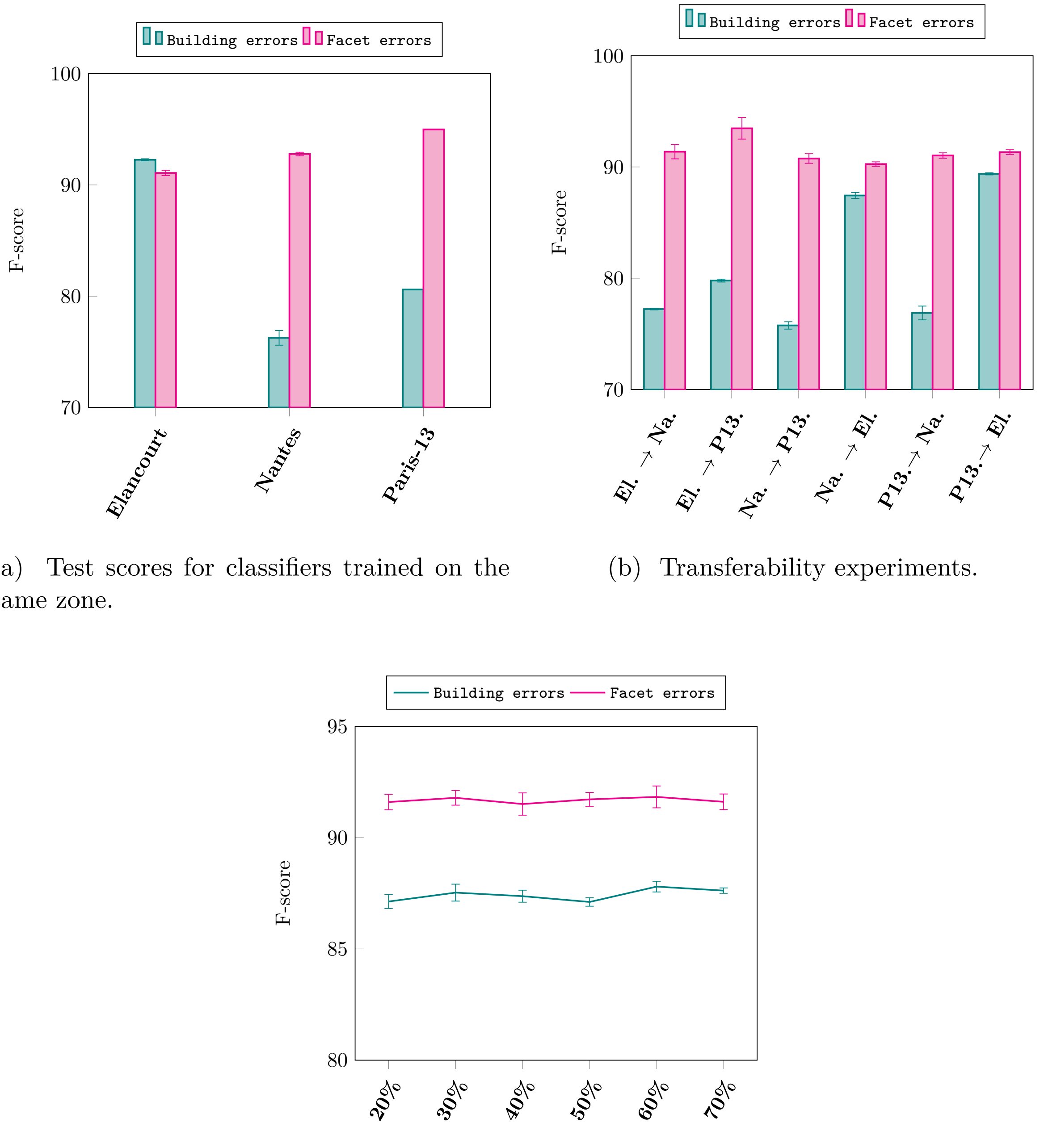
For the level 1 in **eFin**, we start with the feature ablation experiments. Since valid samples are very rare in our case, it is expected that it will be very difficult to detect these instances. In consequence, in Table 5.6, we choose to report correctly `Valid` buildings instead of computing the precision score in percentage.

At this level, even more than the error family semantic degree, feature configurations have virtually no impact on test results: **Elancourt** was the only exception when image features are added to geometric ones. Furthermore, we confirm expectations as, at most, only 1 out of 57 (*resp.* 0 out of 55 and 3 out of 21) valid instances are detected for **Elancourt** (*resp.* **Nantes** and **Paris-13**). As a consequence, we do not report the rest of previously conducted experiments for this **eFin** level. Indeed, There is no interest in comparing prediction transferability, generalization or representativeness if we hardly detect them at all on the same training scene.

5.4.3 Summary

To summarize, we have applied the same study, that was previously conducted at the `atomic` errors level, at the **eFin** levels 2 and 1. We have experimentally proven that:

- `Facet errors` are detected with around a 90 % F-score unaffected by the urban scenes.
- As seen previously on **eFin** level 3, `Building errors` are hard to train on **Nantes** and **Paris-13**, with an F-score around 75 %;
- We confirm once again that:
  - `Building errors` learning is harder to transfer than `Facet errors`;
  - **Elancourt** is the best choice to learn `Building errors` on;
  - **Nantes** and **Paris-13** are similar in composition;
- 20 % is sufficient for learning on the mixed dataset;
- As expected, the distinction between `Erroneous` and `Valid` proved to be difficult to learn.



(a) Test scores for classifiers trained on the same zone.

(b) Transferability experiments.

(c) Representativeness experiments.

Figure 5.16: F-score mean and standard deviation for the baseline feature study results per zone for **eFin** level 2.

6 Computing a better representation

Contents

As seen in the previous chapter, some error labels are poorly detected with the baseline features. The aim here is to present a new set of features so as to achieve better prediction scores. First, Section 6.1, we introduce some advanced feature extractors from the literature that can fit our experimental constraints. In Section 6.2, we show how these ideas are utilized in our evaluation workflow. Last, Section 6.3 presents the implementation details related to these advanced feature extractors.

6.1 Image and graph classification in the literature

In this section, we review two state-of-the-art tools that are used to extract better features of 3D models. First, we present the ScatNet which can be seen as a Convolutional Neural Network (ConvNet) emulator. Second, we show how kernels could be used to classify graphs in the literature.

6.1.1 ScatNet for image classification

Scattering Networks (ScatNets) can be seen as a reverse engineered convolutional network. They are built mainly relying on wavelet filters which are used to imitate filters from ConvNets. The latter learns automatically an image representation Φ trying to minimize a learning loss function. ScatNets, on the other hand, make use of mathematical properties of the image signal.

Some properties are always required in order to achieve a good representation Φ of an image. A scene that is captured from different points of view, while yielding different images, should have close representations. Just as with intersect point detection (Lowe, 2004), the representation of an image should be invariant to scale, translation and rotation (Mallat, 2012; Sifre et al., 2013; Bruna et al., 2013). Since images illustrate not only rigid objects, a good representation should allow small local deformations in the signal that can be attributed to distortions of non-rigid objects.

Mathematical notations.

To formalize these properties, we introduce some mathematical notations.

Action of a group. Let $x : u \mapsto x(u)$ be an image (or any signal in general) and $(G, *)$ a group such as, for example, the set of translation operators on signals. The action of $g \in G$ on x is written as:

$$L_g(x) : u \mapsto x\left(g^{-1}(u)\right). \tag{6.1}$$

As an example, we can study the case of $T_{\mathbb{R}^2}$ a translation group on $\mathbb{R}^2$. This group is isomorphic to $\mathbb{R}^2$: i.e., $(T_{\mathbb{R}^2}, \circ) \cong (\mathbb{R}^2, +)$. Let v be a vector in $\mathbb{R}^2$. We denote a translation by vector v by $t_v : u \mapsto u + v$. Its inverse verifies $t_v^{-1} = t_{-v}$. For $v \in \mathbb{R}^2$, we can express the translation applied to signal as:

$$L_{t_v}(x) : u \mapsto x\left(u - v\right). \tag{6.2}$$

Equally, if we consider the rotation group $SO(2)$ which is isomorphic to the unit circle, and write a rotation of angle $\theta \in [0, 2 \cdot \pi)$ as r_θ, we express the action of that group as:

$$L_{r_\theta}(x) : u \mapsto x\left(r_{-\theta}(u)\right). \tag{6.3}$$

Rotations and translations are not commutative in the general case. This justifies the definition of a roto-translation group, called also the rigid motion group and denoted $SE(2)$. Such a group is isomorphic to $\mathbb{R}^2 \times SO(2)$. Let $v \in \mathbb{R}^2$ and $\theta \in [0, 2 \cdot \pi)$, a member of such a group is defined as:

$$g_{\theta,v} \triangleq t_v \circ r_\theta = r_\theta \circ t_{r_{-\theta}(v)} \tag{6.4}$$

and its inverse is expressed then as $g_{\theta,v}^{-1} = r_{-\theta} \circ t_{-v}$. Hence we can write:

$$L_{g_{\theta,v}}(x) : u \mapsto x\left(r_{-\theta}(u - v)\right). \tag{6.5}$$

Group convolution. A convolution over a locally compact topological group G is defined as:

$$x \circledast y : u \mapsto \int_{h \in G} x(h) \cdot y(h^{-1}(u)) \cdot dh \tag{6.6}$$

where:

dh : is the Haar measure on G.

For signals x and y that take inputs on the plane $\mathbb{R}^2$, we can recognize the usual definition of a convolution:

$$x \star y : u \mapsto \int_{v \in \mathbb{R}^2} x(v) \cdot y(u - v) \cdot dv \tag{6.7}$$

Applied in the roto-translation case to signals $x(u, \theta)$ on pairs from $\mathbb{R}^2 \times [0, 2 \cdot \pi)$, we write:

$$x \circledast_{SE(2)} y : (u, \theta) \mapsto \int_{\substack{v \in \mathbb{R}^2 \\ \omega \in [0, 2 \cdot \pi)}} x(v, \omega) \cdot y(r_{-\omega}(u - v), \theta - \omega) \cdot dv d\omega \tag{6.8}$$

Invariance: formulation.

Φ is said to be invariant to g if and only if:

$$\Phi \circ L_g\,(x) = \Phi(x), \tag{6.9}$$

and covariant if and only if:

$$\Phi \circ L_g\,(x) = L_g \circ \Phi(x). \tag{6.10}$$

Images of similar objects should have also similar representations. Using Euclidean distances as a similarity measure, this property implies Φ to be contractive (Bruna et al., 2013). Given two images x and y, we write:

$$\|\Phi(x) - \Phi(y)\| \leq \|x - y\|. \tag{6.11}$$

The mapping Φ should be stable under small local deformations. A local deformation is modeled as a diffeomorphism[30] τ. A small one is formalized as a diffeomorphism τ that verifies $\|\nabla \tau\|_\infty < 1$ (Mallat, 2012; Bruna et al., 2013; Sifre et al., 2013). Consequently a good transform Φ should be Lipschitz stable to deformations:

$$\exists C > 0, \forall x \in \mathscr{I} \quad \|\Phi(x) - \Phi \circ L_\tau(x)\| \leq C \cdot \|x\| \cdot \|\nabla \tau\|_\infty. \tag{6.12}$$

where:

$\mathscr{I}$: is the set of all possible signals;
$L_\tau(x) : u \mapsto x(u - \tau(u))$ is the action of the deformation τ on a signal x.

ScatNet: a reverse engineered ConvNet.

ConvNets learn such a representation in the hopes of verifying these previous conditions stated in Equations 6.11, 6.9 and 6.12. In fact, it is possible by virtue of some basic building blocs. First, since the weights are learned based on a classification objective, images with similar content are supposed to be mapped to the same region. Secondly, pooling layers in ConvNets provide a solution for invariance to translations and local

[30] A differentiable bijection which inverse is also differentiable.

deformations up to a scale.

ScatNets formalize these ideas using wavelet transforms. Actually, ScatNets were originally developed to better understand ConvNets. The idea is to apply linear convolutional operators followed by a non-linearity and some pooling operators. In a ScatNet the learned filters are replaced by a specific wavelet decomposition, the non-linearity is achieved using the modulus operator and pooling is obtained through a low pass filter. The low-pass filter is critical for the invariance condition, while the lost high frequencies are retrieved using the wavelet convolutions. The non-linearity, just as in ConvNets, is necessary in order to obtain interesting non-linear final representations.

Wavelets filter bank. We need two sets of wavelets: spatial and angular ones. Spatial wavelet filters are derived based on a mother wavelet $\psi : \mathbb{R}^2 \rightarrow \mathbb{C}$ as follows:

$$\psi_{i,\theta} : u \mapsto 2^{-2\cdot i} \cdot \psi\left(2^{-i} \cdot r_{-\theta}(u)\right). \tag{6.13}$$

We also need a low-pass averaging filter $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ scaled up to a desired size:

$$\phi_I : u \mapsto 2^{-2\cdot I} \cdot \phi\left(2^{-I} \cdot u\right) \tag{6.14}$$

The angular wavelets are computed using $2 \cdot \pi$-periodic mother wavelet $\bar{\psi} : \mathbb{R}^2 \rightarrow \mathbb{C}$ scaled as follows:

$$\bar{\psi}_\xi : \omega \mapsto 2^{-\xi} \cdot \bar{\psi}\left(2^{-\xi} \cdot \omega\right) \tag{6.15}$$

We also define the constant averaging filter[31] defined as:

$$\bar{\phi} : \theta \mapsto \frac{1}{2 \cdot \pi} \tag{6.16}$$

Based on the spatial filter bank defined in Equation 6.13 and the angular one from Equation 6.15, we define rigid motion wavelets in the next equations:

$$\forall i \neq I, \theta \neq 0, \xi \neq 0 \quad \tilde{\psi}_{i,\theta,\xi}(u,\omega) = \psi_{i,\theta}(u) \cdot \bar{\psi}_\xi(\omega) \tag{6.17}$$

$$\forall \xi \neq 0 \quad \tilde{\psi}_{I,0,\xi}(u,\omega) = \phi_I(u) \cdot \bar{\psi}_\xi(\omega) \tag{6.18}$$

$$\forall i \neq I, \theta \neq 0 \quad \tilde{\psi}_{i,\theta,0}(u,\omega) = \psi_{i,\theta}(u) \cdot \bar{\phi}(\omega). \tag{6.19}$$

The corresponding rigid motion averaging filter is obtained as:

$$\begin{aligned}\tilde{\phi}(u,\omega) &= \tilde{\psi}_{I,0,0}(u,\omega) = \phi_I(u) \cdot \bar{\phi}(\omega) \\ \tilde{\phi}(u,\omega) &= \frac{1}{2 \cdot \pi} \cdot \phi_I(u)\end{aligned} \tag{6.20}$$

These are separable filters that can help speed up the convolution computations as shown in Sifre et al. (2013). In practice, ψ(*resp.* $\bar{\psi}$) are taken as two (*resp.* one) dimensional Morlet wavelets (cf. Figure 6.1), while ϕ is taken as a Gaussian (Sifre et al., 2013; Bruna et al., 2013; Oyallon et al., 2015). A finite number L of orientations are possible for angle indices: $\theta \in \left\{\frac{l \cdot \pi}{L} : l = 1, 2 \ldots, L\right\}$. In Figure 6.3, we can see how the designed filter bank is similar to the one learned by a ConvNet.

[31] It corresponds to a zero scale filter.

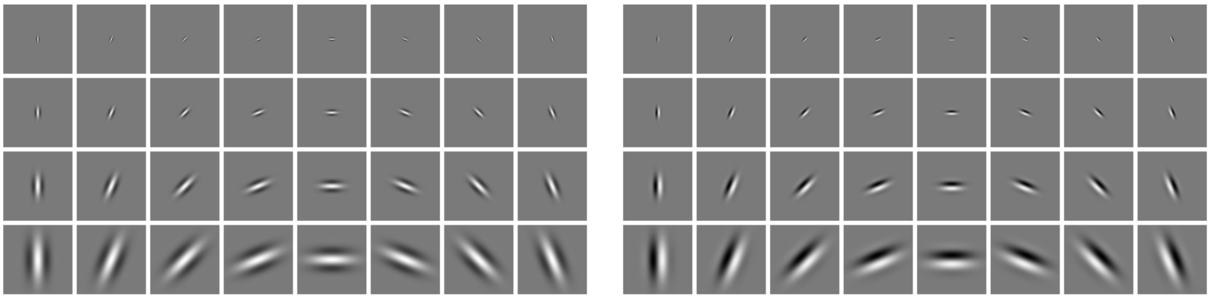


Figure 6.1: Morlet wavelets at different scales ($I = 5$) and orientations ($L = 8$). Left are presented the real parts while the imaginary parts are on the right. Image taken from (Sifre et al., 2013).

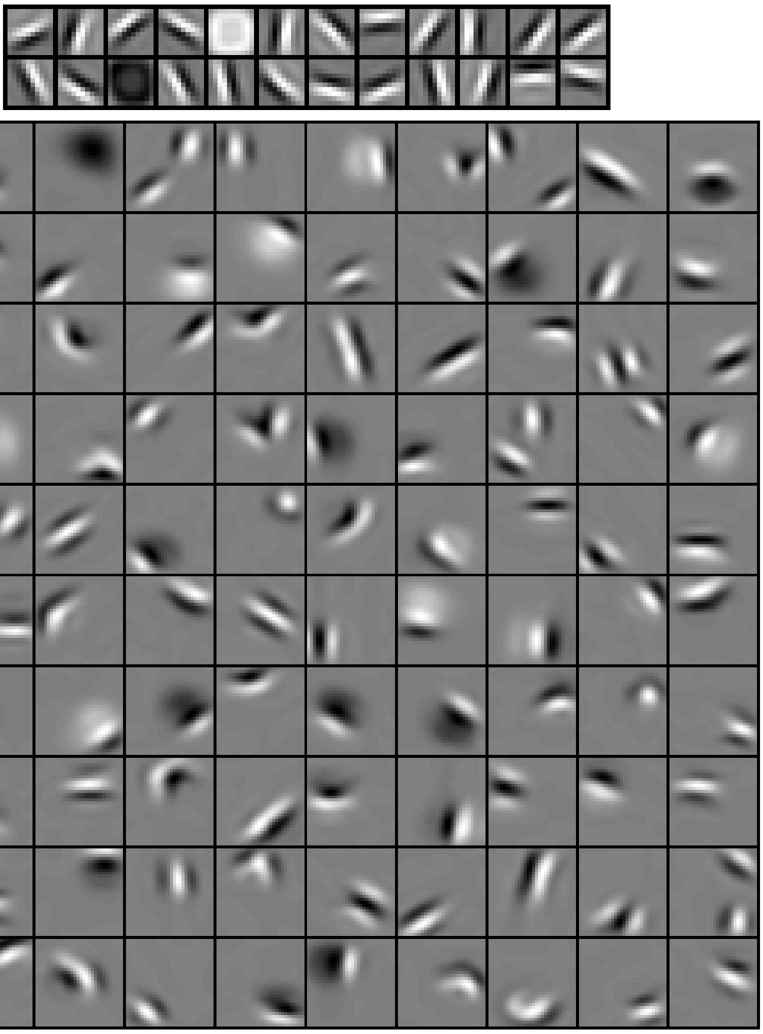


Figure 6.2: Learned filters at the first (top) and second (bottom) layers of a ConvNet. Image taken from (Lee et al., 2009).

Figure 6.3: Comparison of the Morlet wavelet bank and the first and second layer filters learned from a ConvNet. The latter can be distinguished into two classes: an averaging filter that could be likened to ϕ and rotated and scaled (only in the second layer) filters that looks like Morlet wavelets.

The wavelet-modulus operator. A wavelet transform of a signal x consists in representing it using a wavelet filter bank: $(x \circledast \psi_\lambda)_\lambda$. Depending on the transformed signals, two versions of wavelet transforms are possible. If the signal takes one variable $u \in \mathbb{R}^2$ then we write:

$$W : x \mapsto (x \star \psi_{i,\theta})_{\substack{i=1,2\ldots,I \\ \theta\in\left\{\frac{l\cdot\pi}{L}:l=1,2\ldots,L\right\}}} \quad , \tag{6.21}$$

if it takes two variables $(u, \omega) \in \mathbb{R}^2 \times [0, 2 \cdot \pi)$ it becomes:

$$\widetilde{W} : x \mapsto \left(x \circledast_{SE(2)} \psi_{i,\theta,\xi}\right)_{\substack{i=1,2\ldots,I \\ \theta\in\left\{\frac{l\cdot\pi}{L}:l=1,2\ldots,L\right\} \\ \xi=1,2\ldots,\lfloor\log_2(L)\rfloor}} \quad . \tag{6.22}$$

An image is a discretization of a two dimensional signal. As a consequence, we can only apply the first transform in Equation 6.21. The result of such a mapping is an example of a signal to which one can apply the second transform from Equation 6.22. For the right choice of wavelets, these operators can be proven to be invertible, contractive and Lipschitz stable to deformations (Mallat, 2012).

By applying a modulus we get the basic building bloc of ScatNets called the wavelet-modulus operator: $x \mapsto |x \circledast \psi_\lambda|$. This is delineated for the two cases as follows:

$$U_{i,\theta} : x \mapsto |x \star \psi_{i,\theta}| \tag{6.23}$$

$$\widetilde{U}_{i,\theta,\xi} : x \mapsto |x \circledast_{SE(2)} \psi_{i,\theta,\xi}| \tag{6.24}$$

These operators are proven to be covariant to translations and rotations (Mallat, 2012; Sifre et al., 2013): i.e., for a couple $(v, \vartheta) \in \mathbb{R}^2 \times [0, 2 \cdot \pi)$:

$$L_{g_{v,\vartheta}} \circ U_{i,\theta} = U_{i,\theta} \circ L_{g_{v,\vartheta}} \tag{6.25}$$

$$L_{g_{v,\vartheta}} \circ \widetilde{U}_{i,\theta,\xi} = U_{i,\theta,\xi} \circ L_{g_{v,\vartheta}}. \tag{6.26}$$

These conditions may remind the reader of the work of Cohen et al. (2016) on Group equivariant Convolutional Neural Network (G-ConvNet).

The average pooling. Contrarily to (Cohen et al., 2016), the pooling operations differ. G-ConvNets rely on max-pooling as standard practice in ConvNets. This operator is actually covariant to actions of the rigid movement group. (Bruna et al., 2013; Sifre et al., 2013; Oyallon et al., 2015), however, rely on averaging (or low-pass) filters as pooling operators.

Considering the rigid movement as a nuisance, (Sifre et al., 2013) considers invariance with regards to the roto-translation group. To do so they filter any signal $x(u)$ using ϕ_I yielding a signal:

$$P_I : x \mapsto x \star \phi_I(u). \tag{6.27}$$

For signals $\tilde{x}(u, \omega)$ they are averaged using $\tilde{\phi}_I$ and giving as ouput:

$$\widetilde{P}_I : x \mapsto \tilde{x} \circledast_{SE(2)} \tilde{\phi}_I(u, \omega). \tag{6.28}$$

These operators are actually invariant to translation and rotation:

$$P_I = P_I \circ L_{t_v} \tag{6.29}$$

$$\tilde{P}_I = \tilde{P}_I \circ L_{g_{v,\vartheta}}. \tag{6.30}$$

Scattering coefficients. Applying the first operation 6.27 to the input image defines the first scattering coefficient:

$$S_0[x](u) \triangleq x \star \phi_I(u). \tag{6.31}$$

To the image, the wavelet-modulus operator is applied giving coefficients $U_{i_1,\theta_1}(x)$. The latter is averaged using the second operation from Equation 6.28. This defines the second layer of scattering coefficients:

$$S_1[x](u, i_1, \theta_1) \triangleq U_{i_1,\bullet} \circledast_{SE(2)} \tilde{\phi}_I(u, \theta_1). \tag{6.32}$$

At the second level is applied the wavelet-modulus operator retrieving the high-frequencies lost after the low-pass filter using yet another time the wavelet-modulus operator yielding:

$$\widetilde{U}_{i_2,\xi_2,\theta_2} \circ U_{i_1,\theta_1}(x) \tag{6.33}$$

Once again, an average pooling is applied to the latter:

$$S_2[x](u, i_1, \theta_1, i_2, \theta_2, \xi_2) \triangleq \widetilde{U}_{i_2,\xi_2,\theta_2} \circ U_{i_1,\bullet} \circledast_{SE(2)} \tilde{\phi}_I(u, \theta_1). \tag{6.34}$$

This can be reapplied further giving cascaded scattering coefficients at level m:

$$S_m[x](u, p_m) \triangleq \widetilde{U}_{\lambda_m} \circ \widetilde{U}_{\lambda_{m-1}} \cdots \circ \widetilde{U}_{\lambda_2} \circ U_{i_1,\bullet} \circledast_{SE(2)} \tilde{\phi}_I(u, \theta_1). \tag{6.35}$$

where:

p_m : $i_1, \theta_1, \lambda_2 \ldots, \lambda_{m-1}, \lambda_m$ and is called a path;
λ_k : i_k, θ_k, ξ_k.

The scattering coefficients are proved to be contractive and Lipschitz stable to deformations (Mallat, 2012). Moreover, they are invariant to actions of the group of rigid movement. In fact, concatenating a covariant operator and an invariant one yields an invariant operator (Mallat, 2012; Sifre et al., 2013).

The energy of the signal is concentrated along increasing scale paths: i.e., $\forall k = 1, 2, \ldots, m-1\ i_{k+1} > i_k$. This implies that computing coefficients along these paths is sufficient (Bruna et al., 2013; Sifre et al., 2013; Oyallon et al., 2015). Furthermore, only the first two layers are computed as they concentrate most the energy of the signal (Bruna et al., 2013; Sifre et al., 2013; Oyallon et al., 2015). This yields an efficient way of computing a scattering transform as discussed by Sifre et al. (2013) and Oyallon et al. (2015). This means that total number of the possible paths is in practice:

$$n_S = \underbrace{1}_{\substack{\text{layer}\\ l=0}} + \underbrace{L \cdot I}_{\substack{\text{layer}\\ l=1}} + \underbrace{\frac{L^2 \cdot I \cdot (I-1)}{2}}_{\substack{\text{layer}\\ l=2}}. \tag{6.36}$$

Sifre et al. (2013) go on to propose a way to make the scattering invariant to scale effects also. This is done by introducing a logarithm that linearizes the dependency of scattering coefficients to scales i_1 and i_2 (Sifre et al., 2013; Oyallon et al., 2015). A scale-space averaging is thus applied to achieve the sought invariance (Sifre et al., 2013).

In contrast, Oyallon et al. (2015) propose to keep only the translation invariance and let the classifier decide on the relevance of rotation and scale invariance, much like in (Cohen et al., 2016). To do so only a spatial averaging convolution is applied as $x \mapsto \tilde{x}(\bullet, \omega) \star \phi_I$ which is covariant to rotations. Similarly, they drop the scale-space averaging at the end, while the logarithm guaranties the covariance of the signal to scaling effects. The operations are also conducted in a way that renders the shape of the ScatNet, shown in Figure 6.4, more like that of ConvNet.

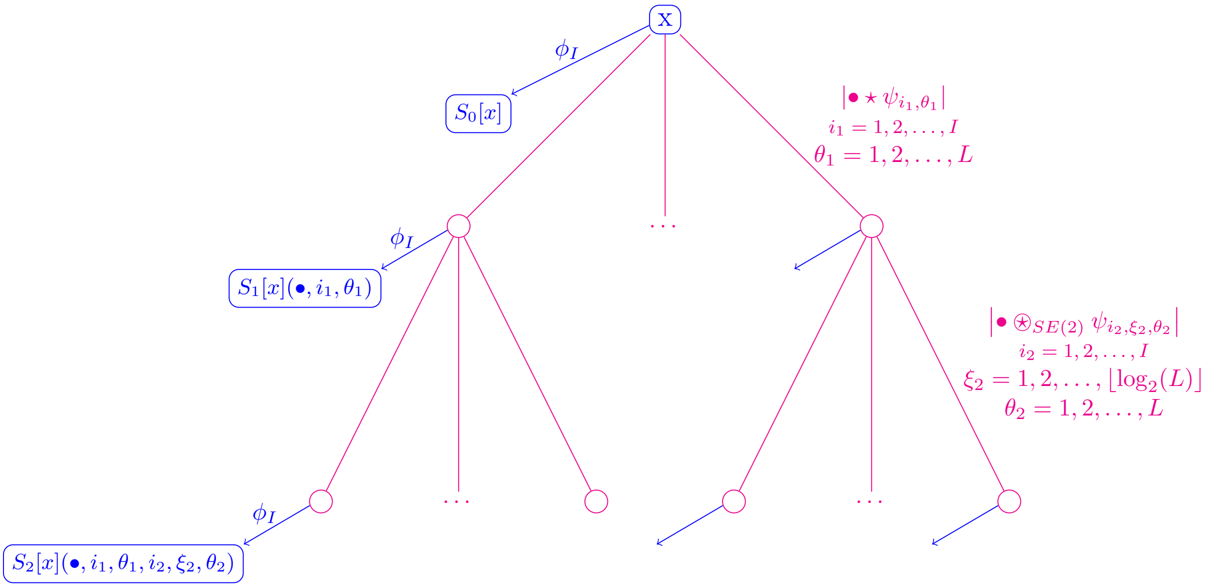

X
ϕ_I
$S_0[x]$
$|\bullet \star \psi_{i_1,\theta_1}|$
$i_1 = 1, 2, \dots, I$
$\theta_1 = 1, 2, \dots, L$
$\dots$
ϕ_I
$S_1[x](\bullet, i_1, \theta_1)$
$|\bullet \circledast_{SE(2)} \psi_{i_2,\xi_2,\theta_2}|$
$i_2 = 1, 2, \dots, I$
$\xi_2 = 1, 2, \dots, \lfloor \log_2(L) \rfloor$
$\theta_2 = 1, 2, \dots, L$
ϕ_I
$\dots$
$\dots$
$S_2[x](\bullet, i_1, \theta_1, i_2, \xi_2, \theta_2)$

Figure 6.4: Illustration of a ScatNet. At each level are computed convolutions with a filter bank followed by a modulus operator. The scattering coefficients are obtained then by a low-pass filter (in blue). In practice, scattering coefficients are only computed for increasing scale paths up to level 2.

6.1.2 Kernels for graph classification

Standard machine learning practice usually assumes that the observed instances live in a finite dimensional space. This is not always the case. In fact, graphs can have varying numbers of nodes and edges. Moreover, they can be different while providing the same structural information: we say they are isomorphic. A valid representation for graphs should then take care of these two issues: incorporate all possible graph sizes and be invariant to graph isomorphisms. As accustomed in statistical learning, one way to alleviate these issues is to directly compare graphs using kernels as seen in Section B.2.

This Section does not aim at presenting a thorough survey of graph kernels. The work of Ghosh et al. (2018) categorizes graph kernels depending on the used methodology. A different approach is proposed in Kriege et al. (2020), where graph kernels are studied based on the underlying graphs.

Apart from the structure of the graph, kernels should also take into account the labels or continuous attributes assigned to a graph. These could be given at node level or edge level. We will present hereafter some graph kernels which were used in this work, namely, continuously node attributed ones, as well as those devoted only to its structural properties.

Basic kernels.

Let $G = (V, E)$ be an undirected graph with vertices $v \in V$ and edges $e \in E = \{\{u, v\} : (u, v) \in V \times V\}$. Attributes can be associated to each node $a : V \to \mathbb{R}^{d_V}$ or edge $b : E \to \mathbb{R}^{d_E}$.

The most basic graph kernel would correspond to a scalar product of a global hashing vector of all attributes. This is possible through the use of a histogram function for instance. This type of functions is described in details in Section 4.2.1.

Let $S\left((a(v)_i)_{v\in V}\right) : \mathbb{R}^{|V|} \to \mathbb{R}^{l}$ (*resp.* $S\left((b(e)_i)_{e\in E}\right) : \mathbb{R}^{|E|} \to \mathbb{R}^{m}$) be a hashing function that describes the distribution of attributes i of all nodes (*resp.* edges) of graph G as a $\mathbb{R}^{l_i}$ (*resp.* $\mathbb{R}^{m_i}$) vector. We can build a node based feature vector for the graph G:

$$\Phi_V(G) \triangleq \begin{bmatrix} S\left((a(v)_1)_{v\in V}\right) \\ S\left((a(v)_2)_{v\in V}\right) \\ \vdots \\ S\left((a(v)_{d_V})_{v\in V}\right) \end{bmatrix} \in \mathbb{R}^{d_V \cdot l} \tag{6.37}$$

and an edge based one:

$$\Phi_E(G) \triangleq \begin{bmatrix} S\left((b(e)_1)_{e\in E}\right) \\ S\left((b(e)_2)_{e\in E}\right) \\ \vdots \\ S\left((b(e)_{d_E})_{e\in E}\right) \end{bmatrix} \in \mathbb{R}^{d_E \cdot m}. \tag{6.38}$$

Based on a base kernel on vectors κ (cf. Section B.2), we can compute the similarity between two graphs $G = (V, E)$ and $G' = (V', E')$:

$$k_V(G, G') \triangleq \kappa(\Phi_V(G), \Phi_V(G')) \tag{6.39}$$

$$k_E(G, G') \triangleq \kappa(\Phi_E(G), \Phi_E(G')). \tag{6.40}$$

Concatenating both feature vectors would amount to a simple addition of these kernels:

$$k(G, G') \triangleq k_V(G, G') + k_E(G, G'). \tag{6.41}$$

This type of kernels is versatile as it can be applied to node and edge attributes as well as labels (with the right choice of hashing function). However, it does not take into account the structure of the graph. It is mainly used as a baseline for graph feature extraction. The work of Shervashidze et al. (2011) is an example of kernels that uses the same idea to describe graphs while taking account of their structure.

Random walk kernel.

In order to define this kernel, we need to define the adjacency matrix A of a graph G:

$$A \triangleq \left(\delta_{\{u,v\}\in E}\right)_{(u,v)\in V\times V} \tag{6.42}$$

and the diagonal matrix of node degrees $D \triangleq \operatorname{diag}\left(\sum_{v\in V} A_{uv}\right)_{u\in V}$. We also denote the normalized adjacency matrix as:

$$P \triangleq A \cdot D^{-1} \tag{6.43}$$

The latter could be interpreted as a transition probability matrix of a random walk on the graph: P_{uv} is the probability of choosing u as the next node to visit starting from v. Similarly, $\left(P^k\right)_{uv}$ expresses the probability of being in node u after k iterations, starting from v. A random walk starts with an initial distribution p over the nodes. After k iterations, the distribution is $P^k \cdot p$. At any time, the walk can end with a probability q_u at node u. p and q are used to encode prior information of the graph (Vishwanathan et al., 2010).

Simultaneous random walk. To compare two graphs G and G' using random walks, we start by defining the direct product graph $G_\times = (V_\times, E_\times)$ where:

$$V_\times \triangleq V \times V' \tag{6.44}$$

$$E_\times \triangleq \{\{(u,u'),(v,v')\} : \{u,v\} \in E \wedge \{u',v'\} \in E'\}\,. \tag{6.45}$$

This is vizualized in Figure 6.5.

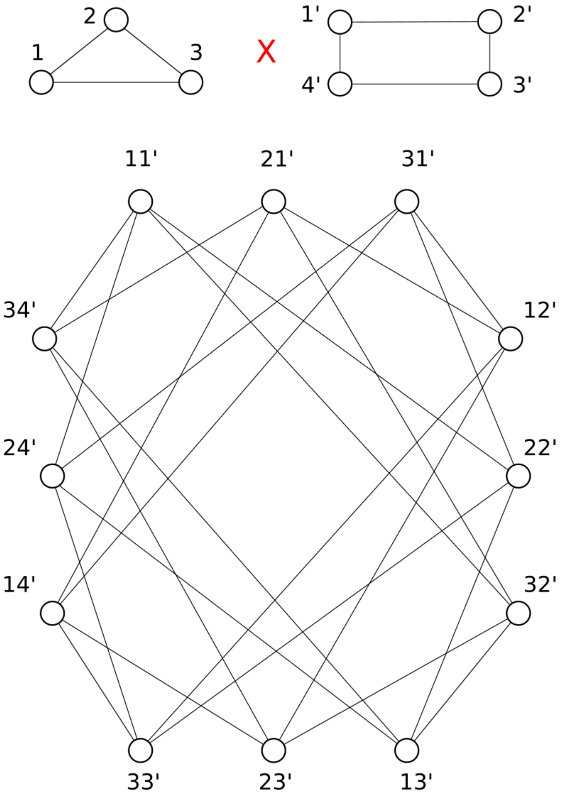


Figure 6.5: Depiction of a direct product (bottom) of two graphs (top). Nodes adjacent in the direct product graph $G_\times$ correspond to adjacent nodes in both graphs. Image taken from (Vishwanathan et al., 2010).

For the direct product graph, we compute adjacency matrices, initial probability distribution and stopping probabilities as follows:

$$A_\times = A \otimes A' \tag{6.46}$$

$$P_\times = P \otimes P' \tag{6.47}$$

$$p_\times = p \otimes p' \tag{6.48}$$

$$q_\times = q \otimes q' \tag{6.49}$$

where:

$\otimes$: refers to the Kronecker product.

In fact, a random walk on $G_\times$ is equivalent to a simultaneous random walk on G and G' (Hammack et al., 2011).

A random walk on a graph depends heavily on the its structure. In consequence, a random walk on the direct product graph can be used as an indicator of the similarity in structure of both graphs G and G'. It is defined as:

$$k_{rw}(G,G') \triangleq \sum_{k=0}^{\infty} \lambda_k \cdot q_\times^\intercal \cdot P_\times \cdot p_\times\,. \tag{6.50}$$

The choice of the series $(\lambda_k)_{k\in\mathbb{N}}$ is critical, as the kernel defined in Equation 6.50 can diverge. It is taken to be non-negative in order in ensure the positive semi-definite of the kernel.

Special cases. This defintion actually generalizes a family of graph kernels based on walks on graphs (Vishwanathan et al., 2010). Setting $p_\times \propto 1$ and $q_\times \propto 1$ and, there are two cases of interest:

- Let $\lambda \in \mathbb{R}_+^*$, we set $\forall k \in \mathbb{N}\ \lambda_k = \lambda^k$: this gives rise to a geometric random kernel:

$$k_{grw}(G, G') = \begin{bmatrix}1, 1 \ldots, 1\end{bmatrix} \cdot (I - \lambda \cdot A_\times)^{-1} \cdot \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}. \tag{6.51}$$

  Let $\lambda_\times$ the largest eigenvalue of $A_\times$. For this kernel to be valid, the following condition must hold:

$$\lambda < \frac{1}{\lambda_\times} \tag{6.52}$$

- Let $\lambda \in \mathbb{R}_+^*$, we set $\forall k \in \mathbb{N}\ \lambda_k = \frac{\lambda^k}{k!}$: this yields the so called exponential random kernel:

$$k_{erw}(G, G') = \begin{bmatrix}1, 1 \ldots, 1\end{bmatrix} \cdot \exp(\lambda \cdot A_\times) \cdot \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}. \tag{6.53}$$

These graph kernels were already defined in (Gärtner et al., 2003).

Computationally, these kernels involve heavy computations. Vishwanathan et al. (2010) proposed efficient numerical algorithms to alleviate this problem. However, these graphs suffer from other issues. First, the kernel does take into account the attributes of graphs if they exist, although it is possible to adapt. Secondly, these kernels suffer from tottering. The latter being the fact that random walks are mostly consistent of multiple movements between a small subset of vertices in a row. Third, related to the previous point, the random walk would spend much time on central nodes that connect to most other nodes adding to their contribution to the kernel, while peripherial ones that can hold the most distinctive feature of the structure are watered down. It is possible to address these issues as demonstrated by Horváth et al. (2004) and Mahé et al. (2004). However these kernels cannot be guaranteed to be computed in polynomial time (Vishwanathan et al., 2010).

SVM ϑ kernel.

This is also a kernel which takes advantage only of the structure of graphs. It relies on the definition of Lovász number of a graph $\vartheta(G)$. For each vertex $v \in V$, we can assign a unit vector $\boldsymbol{w}_v \in \left\{u \in \mathbb{R}^d : \|u\| = 1\right\}$. The set of vectors $W(G) \triangleq (\boldsymbol{w}_v)_{v\in V}$ is said to be an orthonormal representation of a graph G if and only if:

$$\forall \{u, v\} \notin E \Rightarrow \boldsymbol{w}_u^\intercal \cdot \boldsymbol{w}_v = 0.$$

Lovász ϑ kernel. The Lovász number (Lovász, 1979) is defined as:

$$\vartheta(G) \triangleq \min_{\substack{\boldsymbol{c}\in\mathbb{R}^d \\ W(G)}} \max_{v\in V} \left(\frac{1}{\boldsymbol{c}^\intercal \cdot \boldsymbol{w}_v}\right)^2 \tag{6.54}$$

Which can be interpreted as the smallest cone enclosing a valid orthonormal representation of graph G.
Equally, we define the Lovász number over a subset of vertices $B \in V$ as follows:

$$\vartheta_B(G) \triangleq \min_{\boldsymbol{c}\in\mathbb{R}^d} \max_{\boldsymbol{w}_v\in W_B^*(G)} \left(\frac{1}{\boldsymbol{c}^\intercal \cdot \boldsymbol{w}_v}\right)^2 \tag{6.55}$$

where:

$W_B^*(G)$: is the restriction of $W^*(G)$ over the subset B;
$W^*(G)$: is the maximizer of the problem in Equation 6.54.

The Lovász kernel (Johansson et al., 2014) is defined based on a base kernel $\kappa : \mathbb{R}\times\mathbb{R} \to \mathbb{R}_+$ as:

$$k_\vartheta(G, G') \triangleq \sum_{\substack{B\subseteq V \\ B'\subseteq V' \\ |B|=|B'|}} \frac{1}{\binom{|V|}{|B|}\cdot\binom{|V|}{|B'|}} \cdot \kappa\left(\vartheta_B(G), \vartheta_{B'}(G')\right) \tag{6.56}$$

Lovász number approximation. The Lovász number is, however, hard to compute (Johansson et al., 2014). An approximation is possible using the work of Jethava et al. (2013). It presents an alternative definition of Lovász number. We define

$$L \triangleq \left\{K \in S_{|V|}^+ : \forall v \in V, K_{vv} = 1 \land \forall \{u, v\} \notin E, K_{uv} = 0\right\}$$

where $S_{|V|}^+$ is the set of $|V| \times |V|$ positive semi-definite matrices. We can hence write:

$$\vartheta(G) = \min_{K\in L} \omega(K) \tag{6.57}$$

where:

$$\omega(K) = \max_{\alpha_v>0,\forall v\in V} 2 \cdot \sum_{v\in V} \alpha_v - \sum_{(u,v)\in V\times V} \alpha_u \cdot \alpha_v \cdot K_{uv} \tag{6.58}$$

is dual one-class SVM (cf. Section B.2).

If λ_m is the minimum eigenvalue of the adjacency matrix A (cf. Equation 6.42), for $\rho \geq -\lambda_m$, the matrix

$$K_{LS} \triangleq \frac{1}{\rho} \cdot A + I \in S_{|V|}^+ \tag{6.59}$$

is interesting as $\omega(K_{LS}) = \sum_{v\in V} \alpha_v$ (Jethava et al., 2013). It is even more interesting, since for Erdos–Rényi random graphs, $\omega(K_{LS})$ is a constant factor approximation to the Lovász number (Jethava et al., 2013) with high probability.

This justifies the definition of the kernel:

$$k_{svm}(G, G') \triangleq \sum_{\substack{B\subseteq V \\ B'\subseteq V' \\ |B|=|B'|}} \frac{1}{\binom{|V|}{|B|}\cdot\binom{|V|}{|B'|}} \cdot \kappa\left(\sum_{v\in B} \alpha_v, \sum_{v\in B'} \alpha_v\right). \tag{6.60}$$

This is still very prohibitive in terms of computational complexity as one has to sum over all $2^{|V|}$ (*resp.* $2^{|V'|}$) subsets of V (*resp.* V'). To avoid this issue, for each graph, only a set $\mathscr{S}$ (*resp.* $\mathscr{S}'$) of few of the subsets of G (*resp.* G') are visited:

$$\hat{k}_{svm}(G, G') \triangleq \sum_{\substack{B\in\mathscr{S} \\ B'\in\mathscr{S}' \\ |B|=|B'|}} \frac{1}{\binom{|V|}{|B|}\cdot\binom{|V|}{|B'|}} \cdot \kappa\left(\sum_{v\in B} \alpha_v, \sum_{v\in B'} \alpha_v\right). \tag{6.61}$$

Multiscale Laplacian kernel.

Related to random walks on graphs, the Laplacian characterizes the structure of graphs, especially its low eigenvalue eigenvectors (Kondor et al., 2016). It is defined as:

$$\widetilde{L} \triangleq D - A \tag{6.62}$$

Comparing same size Laplacians. When $|V| = |V'|$, we can directly compare the two matrices. The idea of a Laplacian graph kernel k_l is to compare the two matrices by comparing related Gaussian probability distribution. In fact, using a Gaussian graphical model, based on a graph G and node variance $\frac{1}{\eta}$, for a random variable $\boldsymbol{x}$ is equivalent to:

$$\boldsymbol{x} \sim \mathscr{N}\left(0, (L + \eta \cdot I)^{-1}\right). \tag{6.63}$$

Using a Bhattacharyya kernel on probabilities and a regulizer parameter $\gamma > 0$, we define (Kondor et al., 2016):

$$k_l(G, G') \triangleq \frac{\left|\left(\frac{1}{2} \cdot (L^{-1} + \gamma \cdot I)^{-1} + \frac{1}{2} \cdot (L'^{-1} + \gamma \cdot I)^{-1}\right)^{-1}\right|^{\frac{1}{2}}}{|L^{-1} + \gamma \cdot I|^{\frac{1}{4}} \cdot |L'^{-1} + \gamma \cdot I|^{\frac{1}{4}}}. \tag{6.64}$$

The γ regulizer term is added in so as to avoid numerical issues when the one of the Laplacians has eigenvalues equal or close to zero. The Laplacian is not invariant to permutations as the graph as described in Kondor et al. (2016). Moreover, both graphs are required to be of the same size.

This kernel is not well adapted. To alleviate this issue, Kondor et al. (2016) propose to describe the graph nodes by vertex permutation invariant features. If U (*resp.* U') is a matrix which encodes such a transformation for graph G (*resp.* G'), k_l is adapted in what is called feature space Laplacian graph kernel:

$$k_{fl}(G, G') \triangleq \frac{\left|\left(\frac{1}{2} \cdot S^{-1} + \frac{1}{2} \cdot S'^{-1}\right)^{-1}\right|^{\frac{1}{2}}}{|S|^{\frac{1}{4}} \cdot |S'|^{\frac{1}{4}}} \tag{6.65}$$

where

$S = \ U \cdot L^{-1} \cdot U^{\intercal} + \gamma \cdot I;$

$S' = \ U \cdot L'^{-1} \cdot U^{\intercal} + \gamma \cdot I;$

$L = \ \widetilde{L} + \eta \cdot I.$

Node attribute aware Laplacian comparison. Up to now, only the structure of the graph is taken into account. To that extent, utilizing a base kernel κ on node attributes, first is defined the Gram matrix

$$K = (\kappa\,(u, v))_{(u,v) \in V \cup V'} \in \mathbb{R}^{(|V|+|V'|) \times (|V|+|V'|)}.$$

Let $\{(\lambda_1, e_1), (\lambda_2, e_2) \ldots, (\lambda_p, e_p)\}$ be all[32] its eigenvalues and their eigenvectors such that $\forall i = 1, 2 \ldots, p,\ \lambda_i > 0$. We also need to define

$$\widetilde{Q} = \left[\sqrt{\lambda_1} \cdot e_1, \sqrt{\lambda_2} \cdot e_2 \ldots, \sqrt{\lambda_p} \cdot e_p\right] \in \mathbb{R}^{p \times p}.$$

[32] $p \leq |V| + |V'|$.

For graph G (*resp.* G'), the first $|V|$ (*resp.* the last $|V'|$) rows of $\widetilde{Q}$ are taken from Q (*resp.* Q'). Both these matrices are needed to define

$$\bar{S} = Q^\intercal \cdot L^{-1} \cdot Q + \gamma \cdot I$$
$$\bar{S}' = Q'^T \cdot L'^{-1} \cdot Q' + \gamma \cdot I.$$

The generalized feature space Laplacian graph kernel is hence defined as:

$$k_{gfl}^{\kappa}(G, G') \triangleq \frac{\left| \left(\frac{1}{2} \cdot \bar{S}^{-1} + \frac{1}{2} \cdot \bar{S}'^{-1} \right)^{-1} \right|^{\frac{1}{2}}}{\left|\bar{S}\right|^{\frac{1}{4}} \cdot \left|\bar{S}'\right|^{\frac{1}{4}}}. \tag{6.66}$$

Multiscale Laplacian comparison based kernel. The last kernel restricted to a subgraph can in fact be used as a base kernel for the same type of kernels, but at a larger scale. In fact, considering a nested sequence of L sets (neighborhoods) containing $v \in V$:

$$v \in N_1(v) \subseteq N_2(v) \cdots \subseteq N_L(v) \subseteq V. \tag{6.67}$$

Let us denote by $G[A]$ the induced subgraph by $A \subseteq V$:

$$G[A] \triangleq (A, \{\{u, v\} \in E : (u, v) \in A \times A\}). \tag{6.68}$$

The Multiscale Laplacian Subgraph kernels are base kernels:

$$\kappa_0(v, v') \triangleq \kappa(v, v') \tag{6.69}$$
$$\forall l = 1, 2 \ldots, L \quad \kappa_l(v, v') \triangleq k_{gfl}^{\kappa_{l-1}} \left(G[N_1(v)], G'[N_1'(v)] \right). \tag{6.70}$$

Finally, the Multiscale Laplacian Graph kernel is defined using the last base kernel:

$$k_{msl}(G, G') \triangleq k_{gfl}^{\kappa_L}(G, G') \tag{6.71}$$

These kernels could be estimated efficiently by computing once the $\bar{S}$ matrices for all graphs, all multiscale base kernels for all nodes and using a low rank approximation. More details are available in the original paper (Kondor et al., 2016).

Propagation kernel.

The propagation kernel was proposed by (Neumann et al., 2016) and combines ideas from (Shervashidze et al., 2011) with random walks. It relies on a simple kernel:

$$k_s(G, G') \triangleq \sum_{(v, v') \in V \times V'} \kappa(v, v') \tag{6.72}$$

where κ is a base kernel on node attributes[33].

In order to take advantage of the structure of the graph, attributes are propagated using a matrix T giving a new graph G_t at each time t, where $G_0 = G$. If not given by the user, T is taken to be the normalized adjacency matrix P (cf. Equation 6.43). Once the attributes are propagated, the kernel from Equation 6.72 is computed for the new graphs. At time $t_{\max}$, we compute the propagation kernel:

$$k_p(G, G') = \sum_{t=1}^{t_{\max}} k_s(G_t, G_t'). \tag{6.73}$$

Two points are still to be discussed. First, the type of base kernels to use, and secondly, the attribute propagation scheme.

[33] It can accomodate also the case of labeled and partially labeled graphs.

Efficient base kernels through hashing. Knowing the corresponding feature vector extractor $\phi_s(G)$ makes the computation of the simple graph kernel efficient (Shervashidze et al., 2011; Neumann et al., 2016). This is possible provided a base kernel of the form: $\kappa(u,v) = \mathbb{1}_{h(u)=h(v)}$, where h is a hash function defined over nodes. By binning values $h(u)$ in a graph G and encoding the results in a vector $\phi_s(G) = b_h(G)$, the simple graph kernel can be expressed as:

$$k_s(G, G') \triangleq b_h(G)^\intercal \cdot b_h(G'). \tag{6.74}$$

The used hash function is the Locality sensitivity hashing (Neumann et al., 2016).

Node attribute propagation. Nodes attributes are not directly hashed. Instead, the latter are taken, at time t and for node u, as samples of mixtures of Gaussian multivariate distributions $q_{t,u}$ using coefficients W_t at each time t:

$$q_{t,u} \sim \sum_{v \in V} W_{uv} \cdot \mathscr{N}(a(v), \Sigma) \tag{6.75}$$

where:

Σ : is the $d_V \times d_V$ covariance matrix based on attributes $(a(v))_{v\in V}$.

To propagate these distributions at the next iteration, the mixture coefficients are diffused using the already predefined T: $W_{t+1} = T \cdot W_t$. At initialization, $W_0 = I$ and hence $W_t = T^t$.

Graph hopper kernel.

Random walk kernels, just as the basic kernels defined in Equations 6.39 and 6.40 and propagation kernels, are instances of R-convolution kernels (Haussler, 1999): i.e., they can be written as a sum of kernels of substructures of graphs. In the case of the class of kernels defined in Equation 6.50, it can be decomposed into a sum of kernels on all equal length walks from both graphs G and G' (Vishwanathan et al., 2010). In order to deal with issues of these kernels, the shortest path kernel (Borgwardt et al., 2005) proposes to replace walks by shortest paths between pairs of vertices. The graph hopper kernel proposes also to compare pairs of nodes from two graphs in a scalable way (Feragen et al., 2013).

Path kernel. A path π between two vertices $(v_s, v_e) \in V \times V$ is a sequence of nodes $(\pi_i)_{i=1,2\dots,|\pi|}$ such that:

$$\begin{cases} \forall i = 1,2\dots,|\pi|-1,\ \{\pi_i, \pi_{i+1}\} \in E \\ v_s = \pi_1 \\ v_e = \pi_{|\pi|} \end{cases}$$

The set of all shortest paths between nodes of graph G (*resp.* G') is denoted $\mathscr{P}$ (*resp.* $\mathscr{P}'$).

Let $(\pi, \pi') \in \mathscr{P} \times \mathscr{P}'$. To compare both paths, a path kernel is defined:

$$k_p(\pi, \pi') \triangleq \sum_{i=1}^{|\pi|} \kappa\left(\pi_i, \pi'_i\right) \cdot \mathbb{1}_{|\pi|=|\pi'|} \tag{6.76}$$

where:

κ : is a base kernel on node attributes.

This kernel compares paths with similar length by hopping along them simultaneously.

Efficient path based graph kernel. Based on this path kernel, one can compare two graphs G and G' by comparing paths from both:

$$k_{gh}(G, G') \triangleq \sum_{(\pi,\pi')\in\mathscr{P}\times\mathscr{P}'} k_p(\pi, \pi'). \tag{6.77}$$

Defining $w(v, v')$ as the number of times v and v' appear at the same coordinate i of some shortest paths π and π' with the same length $|\pi| = |\pi'|$, this kernel can be computed efficiently as it can be transformed into:

$$k_{gh}(G, G') = \sum_{\substack{v\in V\\ v'\in V'}} w(v, v') \cdot \kappa(v, v'). \tag{6.78}$$

Let $\delta \triangleq \max_{\pi\in\mathscr{P}} |\pi|$ the maximal length of shortest paths in G. Define also the $\delta \times \delta$ matrix

$$M_G(v) \triangleq (|\{\pi \in \mathscr{P} : \pi_i = v \wedge |\pi| = j\}|)_{\substack{i=1,2\ldots,\delta\\ j=1,2\ldots,\delta}}$$

which in row i and column j stores how many times does v appear as the i^{th} member of paths of length j. We can see that:

$$w(v, v') = \langle M_G(v), M_{G'}(v')\rangle_F \tag{6.79}$$

where:

$\langle \bullet, \bullet\rangle_F$: is Frobenius inner product on matrices.

This quantity can be computed efficiently for each graph with a time complexity two orders of magnitude less than that of basic shortest paths kernel (Feragen et al., 2013).

6.2 Evaluation using ScatNet and graph kernels

In the previous studys (cf. Chapters 4 and 5), features are, by construction, taken to be as simple as possible. We aimed at keeping features as simple as possible in order to evaluate the feasibility of our learning approach.

In constrast, we present here features that better exploits structural information of the input models that were missed by the baseline. In the previous section, we identifed two types of instances from which features are extacted: graph-like and image-like data. Advanced graph based feature extractors are proposed in Section 6.2.1. Regarding image-like structures, better attributes are also presented in the next Section 6.2.2.

6.2.1 Graph kernels

In Section 6.1.2, were discussed some kernels that can adequately describe graphs. The geometric baseline features we provided in Section 4.2.1, more precisely in Equation 4.4, could actually be seen as a concatenation of the basic feature maps from Equations 6.37 and 6.38. This corresponds, in fact, to the basic kernel in Equation 6.41.

The biggest disadvantage of this basic kernel is its disregard towards the structural information stored in the graph. That is why we propose to use the other kernels that are presented in Section 6.1.2. None of these kernels takes into account edge attributes. This is actually not an issue as both the edge attributes of the facet graph (cf. Equation 4.1) are in fact a function of node attributes: centroid $f \mapsto \mathscr{G}(f)$ and normal $f \mapsto \vec{n}(f)$. Some of these kernels do not utilize the node attributes either as they take only account of the structure of the graph.

Face normals are unit vectors with coefficients in the interval $[0, 1]$, while centroids are free to be roam in $\mathbb{R}^3$. From a global standpoint, each one of the face geometric features have a specific dynamic. As a consequence, taking all these attributes into account by one graph kernel is going to raise some issues. One possible solution is to normalize all geometric features, concatenate them and associate the resulting node attribute vectors to one graph. We preferred instead to isolate each geometric feature in a specific graph. All graphs would share the same structure but each one takes as node attributes a type of geometric features. This results in three graphs. The first takes the face normals $f \mapsto \vec{n}(f)$ as node attributes. The second graph has its nodes assigned face centroids $f \mapsto \mathscr{G}(f)$. The last one has a composite vector

$$f \mapsto \begin{bmatrix} d(f) \\ \mathscr{A}(f) \\ \mathscr{C}(f) \end{bmatrix}$$

as node attributes. The degree, area and circumference, contrarily to the normal and centroid, of facets where inconsequential in error predictions according to the feature importances that were computed when training RFs. This explains why these were grouped into one vector in contrast with the other two features.

Each graph can take multiple types of kernels. We use the kernels described in Section 6.1.2. Since all graphs share the same structure, kernels that ignore node attributes would yield the same results, no matter which node attribute is used. There are two such kernels: the random walk kernel and the SVM ϑ kernel. We also experimented with three other types of kernels: the Multiscale Laplacian kernel, the propagation kernel and the graph hopper kernel. The latter depends on the choice of the base kernel which compares node attributes. The Radial Basis Function (RBF) was briefly experimented and did not yield desirable results. Two alternatives are utilized:

Linear kernel: As shown in Equation B.14, this is the most simple choice;

Brownian bridge kernel: This base kernel was originally proposed for the Shortest Path kernel (Borgwardt et al., 2005) and is also valid for its scalable derivation.

This results, in total, in

$$\underbrace{2}_{\substack{\text{kernels ignoring} \\ \text{node attributes}}} + \underbrace{3}_{\substack{\text{attributed} \\ \text{graphs}}} \times \left(\underbrace{2}_{\substack{\text{Multiscale Laplacian \&} \\ \text{Propagation}}} + \underbrace{1}_{\text{Graph hopper}} \times \underbrace{2}_{\substack{\text{base} \\ \text{kernels}}} \right) = 14$$

graph kernels. These are aggregated into one kernel using a linear combination. This is possible thanks to Multiple Kernel Learning (MKL) as explained in Section B.2. Other types of kernels were briefly experimented with, namely the Lovász ϑ, Graphlet Sampling, Subgraph Matching and Shortest Path kernels. However, they did not yield any valuable results and most of the time failed numerically.

6.2.2 ScatNet feature extractor

ConvNets have proven to be the standard feature extractors in image classification. However, they require a great load of images in order to learn good enough representations.

This is not our case as explained later in Section 5.1.2. As a consequence, we choose instead to use ScatNets which mimic classical ConvNets and can yield good image representations in an unlearned manner as shown in Section 6.1.1.

Height based features.

Discrepancies between the 3D model extracted height map and the DSM manifest in textures in computed residuals (cf. Figure 4.2c). As a matter of fact, ScatNets can handle very well texture discrimination as proven theoretically by Mallat (2012) and experimentally by Bruna et al. (2013) and Sifre et al. (2013). In addition, theoretically the height data can be fed directly to a ScatNet without requiring any normalization or preprocessing since, by construction, they can admit any type of 2D signal[34]. This explains why ScatNets were chosen as a height based feature extractor.

From a practical standpoint, the residuals computed as in Section 4.2.2 come as images in different sizes $h_{\mathsf{M}} \times w_{\mathsf{M}}$ (cf. Section 4.2.3) depending on the input model. Consequently, concatenating ScatNet coefficients into a single vector is going to result in variable feature vector dimensions. One solution is to resize all images to a certain fixed size beforehand. However, this solution was quickly ruled out based on few experiments. In fact, aside from the fact that this process either looses valuable structural information or adds undesired blur, it completly deforms the input signal as the $\frac{w_{\mathsf{M}}}{h_{\mathsf{M}}}$ ratio is not guaranteed to be constant for all inputs resulting in squashed or elongated image. Moreover, since ScatNets yield a great deal of coefficients that can easily surpass the number of training instances which hinders the learning ability of any classifier. As a consequence, we propose to add a function to help extract meaningful feature vectors with the same length.

Suppose, for any $(\tilde{h}, \tilde{w}) \in \mathbb{N}^* \times \mathbb{N}^*$, we have a function $\chi : \mathbb{R}^{\tilde{h} \times \tilde{w}} \to \mathbb{R}^d$ such as the ones presented in Equations 4.2 and 4.3 which has the same output dimension d no matter the input size $\tilde{h} \times \tilde{w}$. It can be applied on the output of each scattering output $S_l[dsm - alt](\bullet, p)$:

$$\chi\left(S_m[dsm_{\mathsf{M}} - alt_{\mathsf{M}}]\left(\bullet, p_m\right)\right) \in \mathbb{R}^d, \tag{6.80}$$

where:

l : is the scattering output layer;
p_m : is a valid path at layer l (cf. Equation 6.35).

The resulting coefficients defined in Equation 6.80 can be concatenated for all n_S scattering paths to form a feature vector:

$$v_{\text{scattered height}}\left(\mathsf{M}\right) = \begin{bmatrix} \chi\left(S_0[dsm_{\mathsf{M}} - alt_{\mathsf{M}}]\left(\bullet\right)\right) \\ \vdots \\ \chi\left(S_1[dsm_{\mathsf{M}} - alt_{\mathsf{M}}]\left(\bullet, i_1, \theta_1\right)\right) \\ \vdots \\ \chi\left(S_2[dsm_{\mathsf{M}} - alt_{\mathsf{M}}]\left(\bullet, i_1, \theta_1, i_2, \theta_2, \xi_2\right)\right) \\ \vdots \\ \chi\left(S_m[dsm_{\mathsf{M}} - alt_{\mathsf{M}}]\left(\bullet, p_m\right)\right) \end{bmatrix}_{\substack{i_1 \in [\![1, I]\!] \\ \theta_1 \in \frac{\pi}{L} \cdot [\![1, L]\!] \\ i_2 \in [\![i_1 + 1, I]\!] \\ \theta_2 \in \frac{\pi}{L} \cdot [\![1, L]\!] \\ \xi_2 \in [\![1, \lfloor \log_2(L) \rfloor]\!] \\ \vdots \\ \lambda_m \in \Lambda_m}} \in \mathbb{R}^{d \cdot n_S}, \tag{6.81}$$

[34] There are other versions of ScatNets taking one dimensional signals (Andèn et al., 2014) or even graphs (Eickenberg et al., 2018).

where:

Λ_m : is the space of all possible values of parameter λ_m at layer m.

Regarding the χ function it is taken herein as follows:

$$\chi = \chi_{\mathrm{max,min,mean,med,std}} : l \mapsto \begin{bmatrix} \max(l) \\ \min(l) \\ \mathrm{mean}(l) \\ \mathrm{median}(l) \\ \mathrm{std}(l) \end{bmatrix}, \tag{6.82}$$

where:

$\mathrm{std}(l)$: computes the standard deviation over the tuple l.

Image based features.

ScatNets seem then to be good choice for image based feature extractors. In fact, as shown in Figure 3.3, image textures could be also useful for error detection. More importantly, as shown with baseline image based features (cf. Section 4.2.3), edges are key image attributes for comparing building models to orthoimages. Actually, ScatNets are well suited for edge detection as they use Morlet wavelets for convolution operations (cf. Section 6.1.1). These filters are adapted to edge detection (Zhang et al., 2007), as depicted in Figure 6.1.

In order to draw features comparing orthoimages to buildings models, we start first by rasterizing the borders of polygons $f^q \in \mathsf{F}_\mathsf{M}$ of the model into a grid structure mask:

$$Q_\mathsf{M} \triangleq \left(\mathbb{1}_{g_{i,j} \cap \left(\bigcup_{f^q \in \mathsf{F}_\mathsf{M}} f^q \right)} \right)_{\substack{i \in [\![1, h_\mathsf{M}]\!] \\ j \in [\![1, w_\mathsf{M}]\!]}}, \tag{6.83}$$

where:

$g_{i,j}$: is the rectangle[35] representing the pixel at row i and column j.

Two options are possible:

`Deletion`: Pixels g in the corresponding orthoimage which are part of a polygon border (i.e., $Q_\mathsf{M}(g) = 1$) are made black, as shown in Figure 6.6:

$$I_\mathsf{M}^{\mathrm{dl}} \triangleq I_\mathsf{M} \odot (J - Q_\mathsf{M})^{\otimes 3}, \tag{6.84}$$

where:

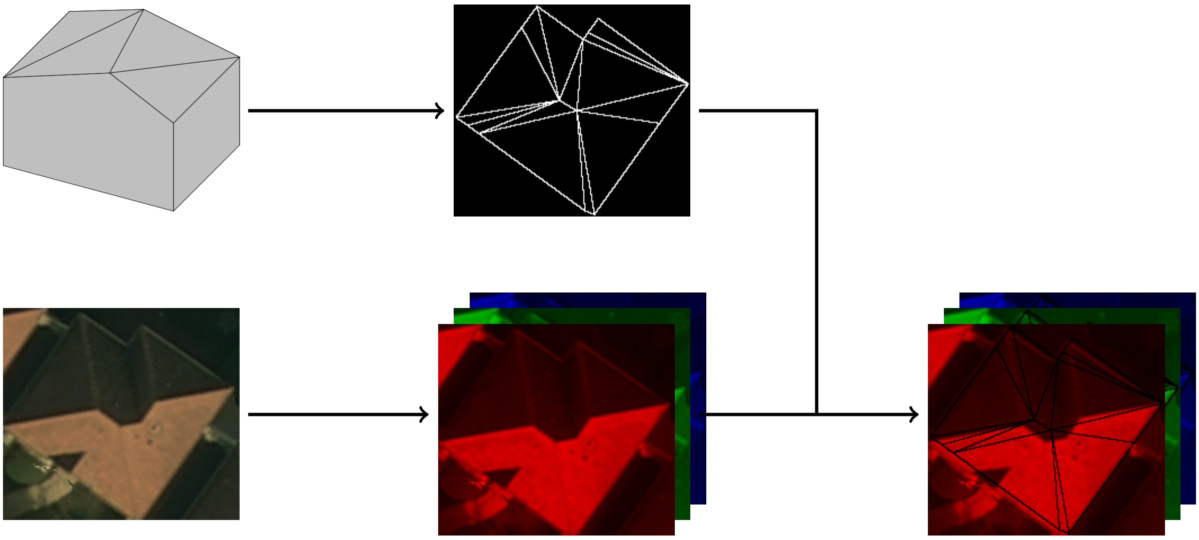
$J_{h_\mathsf{M}, w_\mathsf{M}} = (1)_{\substack{i \in [\![1, h_\mathsf{M}]\!] \\ j \in [\![1, w_\mathsf{M}]\!]}}$: is the matrix of ones of size $h_\mathsf{M} \times w_\mathsf{M}$;

$P^{\otimes 3} = P \otimes P \otimes P$: is the tensor obtained by stacking in depth three copies of the same matrix P;

$A \odot B = (A_{ij} \cdot B_{ij})_{ij}$: denotes the Hadamard/Schur product of any two matrices A and B.

`Channel`: The mask Q_M is simply added to the orthoimage as a fourth channel, as depicted in Figure 6.7:

$$I_\mathsf{M}^{\mathrm{ch}} \triangleq I_\mathsf{M} \otimes Q_\mathsf{M}. \tag{6.85}$$



(a) 3D model.

(b) Nadir Projection.

(c) Orthoimage.

(d) RGB channels.

(e) Early fusion: `deletion`.

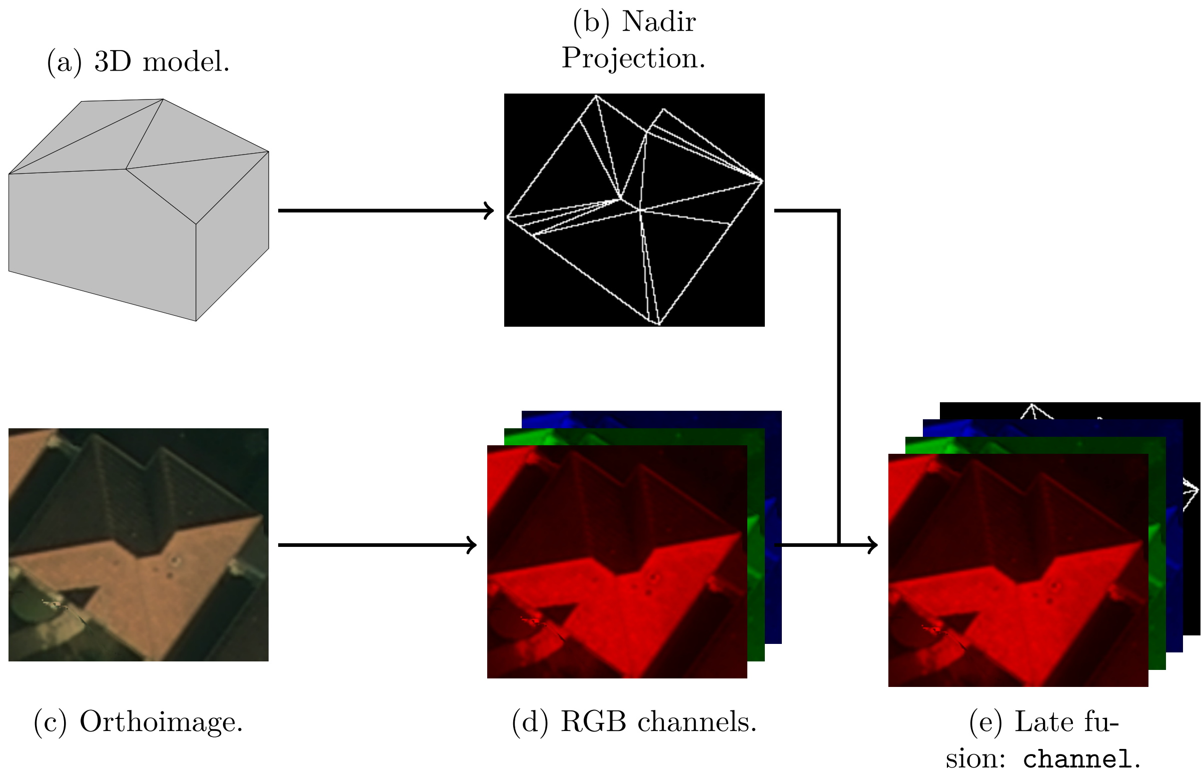
Figure 6.6: Illustration of the early fusion scheme denoted `deletion`. Pixels that intersect the edges of the nadir projection of the model are blackened.

The first situation corresponds to an early fusion scheme while the second represent a late fusion case. Both settings are experimented with and compared later in Section 7.3.1.

Now we can apply the ScatNet on any of the previously defined images. We apply then the same post-processing to yield feature vectors with the same dimensions per channel $d \cdot n_S$:

`Deletion:`

$$v^{\mathrm{dl}}_{\text{scattered image}}(\mathsf{M}) \triangleq \left[\begin{matrix} \chi\left(S_0[I^{\mathrm{dl}}_{\mathsf{M}}](\bullet)\right) \\ \vdots \\ \chi\left(S_1[I^{\mathrm{dl}}_{\mathsf{M}}](\bullet, i_1, \theta_1)\right) \\ \vdots \\ \chi\left(S_2[I^{\mathrm{dl}}_{\mathsf{M}}](\bullet, i_1, \theta_1, i_2, \theta_2, \xi_2)\right) \\ \vdots \\ \chi\left(S_m[I^{\mathrm{dl}}_{\mathsf{M}}](\bullet, p_m)\right) \end{matrix}\right]_{\substack{i_1 \in [\![1, I]\!] \\ \theta_1 \in \frac{\pi}{L}\cdot[\![1, L]\!] \\ i_2 \in [\![i_1+1, I]\!] \\ \theta_2 \in \frac{\pi}{L}\cdot[\![1, L]\!] \\ \xi_2 \in [\![1, \lfloor \log_2(L) \rfloor]\!] \\ \vdots \\ \lambda_m \in \Lambda_m}} \in \mathbb{R}^{3 \cdot d \cdot n_S}. \quad (6.86)$$



(a) 3D model.

(b) Nadir Projection.

(c) Orthoimage.

(d) RGB channels.

(e) Late fusion: `channel`.

Figure 6.7: Illustration of the late fusion scheme denoted `channel`. The mask indicating the pixels that intersect the edges of the nadir projection of the model is added as a fourth channel.

`Channel`:

$$v^{\text{ch}}_{\text{scattered image}}(\mathsf{M}) \triangleq \begin{bmatrix} \chi\left(S_0[I^{\text{ch}}_{\mathsf{M}}](\bullet)\right) \\ \vdots \\ \chi\left(S_1[I^{\text{ch}}_{\mathsf{M}}](\bullet,i_1,\theta_1)\right) \\ \vdots \\ \chi\left(S_2[I^{\text{ch}}_{\mathsf{M}}](\bullet,i_1,\theta_1,i_2,\theta_2,\xi_2)\right) \\ \vdots \\ \chi\left(S_m[I^{\text{ch}}_{\mathsf{M}}](\bullet,p_m)\right) \end{bmatrix}_{\substack{i_1\in[\![1,I]\!] \\ \theta_1\in\frac{\pi}{L}\cdot[\![1,L]\!] \\ i_2\in[\![i_1+1,I]\!] \\ \theta_2\in\frac{\pi}{L}\cdot[\![1,L]\!] \\ \xi_2\in[\![1,\lfloor\log_2(L)\rfloor]\!] \\ \vdots \\ \lambda_m\in\Lambda_m}} \in \mathbb{R}^{4\cdot d\cdot n_S}. \tag{6.87}$$

Eventhough we have used the χ function to reduce the dimension, these feature vector still contains a sizable amount of coefficients. This will prove to be difficult to learn on for both of the considered classifiers.

6.3 Implementation details

As in Section 4.3, we give here a detailed account of how every ingredient of our pipeline is parameterized. First, Section 4.3.1 gives a detailed account of how the different feature configurations of these advanced features were implemented. Secondly, in Section 4.3.2, we present the classification process.

6.3.1 Feature configurations

Baseline features are replaced by more advanced ones as shown in Section 6.2. These are denoted as follows:

- **K-Geom.** refers to geometric features with graph kernels;
- **S-Hei.** refers to ScatNet height based features;
- **S(d)-Im.** (*resp.* **S(c)-Im.**) corresponds to ScatNet image based features with `deletion` (*resp.* `channel`) option;
- **S(d)-All** (*resp.* **S(c)-All**) $\equiv$ **Geom.** $\oplus$ **S-Hei.** $\oplus$ **S(d)-Im.** (*resp.* **S(c)-Im.**);
- **K-S(d)-All** (*resp.* **K-S(c)-All**) $\equiv$ **K-Geom.** $\oplus$ **S-Hei.** $\oplus$ **S(d)-Im.** (*resp.* **S(c)-Im.**);

We lay out herein how their parameters were determined.

Graph kernels.

In Section 6.2.1, we have seen how 14 graph kernels are aggregated to describe graphs. We relied, in experiments, on an available `Python` module called `GraKel`[36] (Siglidis et al., 2018). We provide herein the parameters of each kernel type.

- **Random walk** The exponential version fails numerically and was left out. After a grid search λ was set to be 1×10^{-3} for the geometric random walk. This is actually a very low value as λ has to verify the condition stated in Equation 6.52 for all pairs of graphs in the training dataset. One case where the largest eigenvalue of the direct product of the adjacency matrix of two graphs in the dataset suffices to considerably lower the maximal value λ can take. To compute such a kernel it takes approximatly 1.14 s/building^2.
- **SVM** ϑ This kernel takes no parameters. It takes on average 2.01×10^{-5} s/building^2 to compute a kernel comparison.
- **Multiscale Laplacian** Our building models have a varying number of nodes from 4 up to 20 and more facets in some cases. That is why we choose to keep the same radius size[37] and depth level L set to 3 as in the original paper (Kondor et al., 2016). The same goes for the regularization terms fixed at a value of 1×10^{-2}. Computing one kernel comparison requires around 10.6 s/building^2.
- **Propagation** The choice of the transition matrix, as explained in Section 6.1.2, is set by default to be the normalized adjacency matrix. The maximal iteration number $t_{\max}$ is similar to the height parameter of the Weisfeler-Lehman kernel and is set to 5. Other parameters regarding the hashing and binning processed are kept at their default values. Comparing two graph instances takes around 8.12×10^{-2} s/building^2.
- **Graph hopper** This kernel takes no other parameter than the choice of base kernels. Comparisons between takes on average 34.5 s/building^2.

[36] `GraKel`: `https://github.com/ysig/GraKeL`

[37] The neighboorhoods $N_l(v)$ are taken as balls around vertex v (Kondor et al., 2016).

ScatNet.

For ScatNets, we rely on a modern and vestatile `Python` module: `Kymatio`[38] (Andreux et al., 2018). The parameterization of the ScatNet does not depend on the signal content as much as it relates to the size of input images. This is true because the filter banks, the parameterization of which is the most related to the signal dynamics, were set beforehand. As a consequence, the same parameters were applied for image and height based features.

The number of possible orientations L is fixed at its default value 8 as it was the optimal choice corresponding to the already defined Morlet filter banks. I the scale of the ScatNet pooling operator was set to 3. This corresponds to models M verifying: $w_{\mathsf{M}} \geq 2^3 = 8$ and $h_{\mathsf{M}} \geq 8$. In **Elancourt**, it implies that building models have to be at least larger in length and width than $8 \times 0.06\,\mathrm{m} = 0.48\,\mathrm{m}$, while in **Paris-13** and **Nantes** the minimal dimensions of a building are $8 \times 0.10\,\mathrm{m} = 0.80\,\mathrm{m}$. This is reasonable and fails only for some rare cases were the building is obviously over segmented an can be detected as such by simply applying a building model size threshold.
The maximal number of layers m is 2. As a consequence, the total number, according to Equation 6.36 of scattering outputs is $n_S = 217$. This implies that the length of ScatNet based features is $d \times n_S = 5 \times 217 = 1085$ per channel.

The `Kymatio` implementation of ScatNet uses GPU to accelerate computations. Using an `NVIDIA GeForce GTX 750 Ti` graphics card, it takes around 14.06 s/building on average to compute a height based features and 20.86 s/building. As with previous features onces, once computed they are cached for later use.

6.3.2 Classification settings

Compared to the setup described in Section 4.3, there are some major differences that we report herein.

First, considering the conclusions of Section 5.4, it is not interesting to experiment with the new feature extractors at **eFin** levels 1 and 2. That is why in the following, we will only conduct experiments at **eFin** levels 3.

Regarding the used classifiers, in addition to RFs, we also employ the SVM classifier.

RF: We use the same parameters as in Section 4.3.

SVM: The linear SVM is not well suited for the type of features that we use, even for baseline features as experiments do not yield any results in time. As a consequence, the latter was left out and we experimented only with the kernel SVM using the standard RBF kernel (cf. Equation B.15) when instances are vectors not graphs. Just as with the RF classifier, we conducted a grid search using baseline geometric features only in order to determine both parameters C and γ by limiting the range between 1×10^1 and 1×10^{-3} for both. All values yielded sensibly the same scores. As a result we set these parameters as follows: $C = 1 \times 10^{-1}$ to not overpenalize nor underfit during learning and $\gamma = 1 \times 10^{-3}$ to avoid overfitting. Regarding Multiple Kernel Learning (MKL), we made use of the already implemented EasyMKL (Aiolli et al., 2015) approach. It was the only method, to our knowledge, that was readily available as a library in `Python`[39].

For the assessement metrics, we keep the same ones as in Section 4.3.

[38] `Kymatio`: https://github.com/kymatio/kymatio

[39] `MKLpy`: `https://github.com/IvanoLauriola/MKLpy`

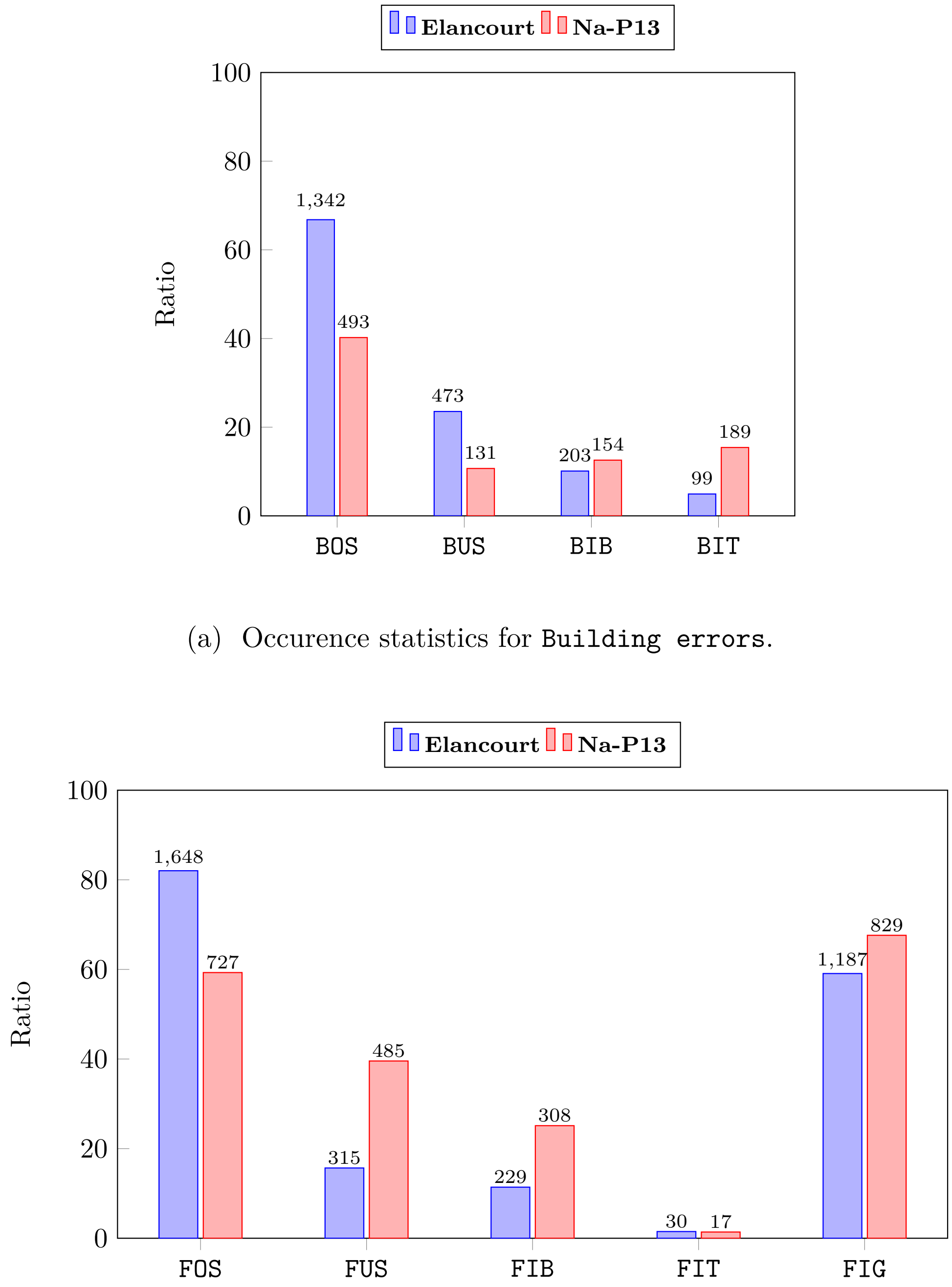
7 Assessing the advanced features

Contents

The goal of this chapter is to apply the feature configurations presented in Chapter 6 and analyse the experimental results. First, in Section 7.1, we explain how the dataset is setup for the new experiments. Next, in Section 7.2, both classifiers, SVM and RF, are trained with this new dataset setup using always the baseline features (cf. Section 4.2). Third, in Section 7.3, we present the results of the new representation for 3D models and compare them to the baseline results.

7.1 Fusing Paris-13 and Nantes

According to the findings of the previous sections (cf. Sections 5.3.1 and 5.4.1), **Paris-13** and **Nantes** are similar compared to **Elancourt**. Moreover, the latter area contains a lot more instances than the others which is not ideal for comparisons. As a consequence, both **Paris-13** and **Nantes** were fused in one set denoted from now by **Na-P13**. It contains 1226 buildings compared to 2007 instances of **Elancourt**. In Figure 7.1, we remind the reader of the **Elancourt** area error distributions which are compared this time to statistics from the fused set **Na-P13**.



(a) Occurence statistics for `Building errors`.

(b) Occurence statistics for `Facet errors`.

Figure 7.1: Detailed error statistics depending on the new experimental sets. The height of bars indicates the frequency of each errors while the number of occurences is displayed over.

Naturally, as **Nantes** contains more (around 1.56 times more) instances than **Paris-13**, **Na-P13** error statistics profile looks a bit more like the one of **Nantes** than the other area as shown in Figure 5.8. According to Sections 5.3.1 and 5.4.1, we can expect that **Na-P13** would be better suited to learn `Facet errors`, with the exception of `FIT`, compared to **Elancourt**, while the latter is also the best alternative for `Building errors`.

7.2 Classifier choice analysis

The aim of this section is to find out how beneficial the use of SVMs can be if used instead of RFs. Two reasons motivate this experimental comparison:

i) SVMs are more adapted to kernels, as we plan trying graph kernels as feature extractors;

ii) SVMs are better suited for unbalanced labels, which is the case of `BIT` and `FIT` for instance.

Hereafter, we first describe the urban scenes that are studied. Next, we compare the results obtained using the SVM classifier to the ones resulting from the RF. We end with a comparison of feature importances computed for the two classifiers.

RF results.

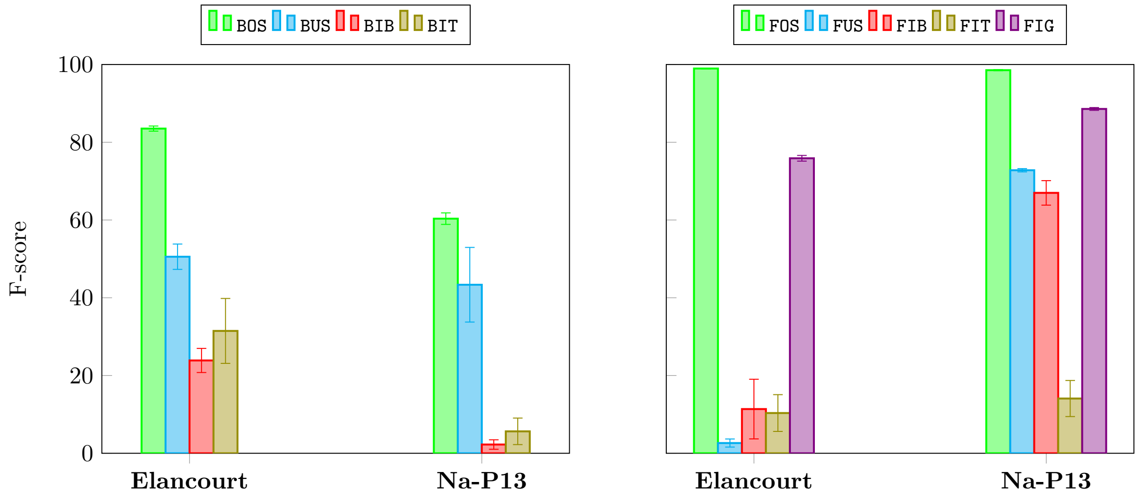
For **Elancourt** results remain unchanged using the RF classifier and are reported along the newer results on **Na-P13** in Table 7.1.

	Elancourt							
	Geom.		**Geom. $\oplus$ Hei.**		**Geom. $\oplus$ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	**93.96**	**76.15**	91.43	77.76	91.51	76.08	90.83	76.14
BUS	32.98	76.47	**41.86**	**75.57**	40.38	71.00	39.32	71.81
BIB	12.32	67.57	12.81	68.42	16.26	67.35	**16.75**	**68.0**
BIT	**25.25**	**92.59**	20.20	90.91	20.20	95.24	11.11	91.67
FOS	98.91	99.07	**98.91**	**99.30**	98.99	98.84	98.91	98.84
FUS	**1.90**	**54.55**	0.63	66.67	1.61	50	1.27	66.67
FIB	**9.17**	**87.5**	0	—	8.30	82.61	7.42	100
FIT	6.67	100	**8.73**	**95.24**	3.33	100	3.33	100
FIG	**80.54**	**73.14**	80.45	72.62	78.69	72.12	79.02	71.82

	Na-P13							
	Geom.		**Geom. $\oplus$ Hei.**		**Geom. $\oplus$ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	**51.65**	**78.93**	47.84	81.75	48.15	77.74	47.43	78.57
BUS	19.85	100	22.90	100	**36.64**	**92.31**	34.61	93.75
BIB	**1.96**	**100**	0.65	100	0.65	100	1.31	100
BIT	**5.32**	**100**	3.19	100	2.13	100	1.06	100
FOS	98.62	98.22	98.62	98.21	98.48	98.76	**98.62**	**98.76**
FUS	68.80	77.44	68.18	77.10	**68.80**	**78.54**	67.83	78.15
FIB	55.23	78.60	53.59	78.47	65.47	74.44	**65.35**	**74.63**
FIT	6.25	100	6.25	100	6.25	100	**11.76**	**100**
FIG	94.55	82.54	95.15	82.72	94.55	83.43	**95.15**	**83.60**

Table 7.1: RF results on the two datasets of interest at **eFin** level 3. Test results are expressed in percentage. All possible modality configurations are tested using baseline features.

Regarding **Na-P13**, one natural prediction is that scores would average out with the same ratios as the frequency of error labels (cf. Figure 7.1). By accounting for the random nature of the choice in training instances during the cross validation, this could be argued to be true for `FOS` or `FIG` and, in a lesser extent, for `BUS` too. However, it is far from being true for the rest of errors as shown in Table C.18. In fact, for the other six error labels, F-scores, on the fused set, are better than those on both **Nantes** and **Paris-13**. Notably, some instances of `BIT` and `FIT` are now detected, in best cases, at arounf 11 % and 21 % respectively. On the contrary, they were not detected at all on the separate areas (cf. Table 5.2). This can be explained by the fact that, although having around the same statistical distribution of errors as **Nantes** and **Paris-13** in a lesser extent, the fused set contains enough instances to better learn than before. It can also be the result of the fact that the two areas complement each other, as better shown in Figure 5.3, with `BIT`, where training on **Nantes** and testing on the other and vice versa proved to be better than training and testing on the same zone.



(a) `Building errors.` (b) `Facet errors.`

Figure 7.2: Mean F-score and standard deviation obtained with an RF using baseline features.

Mean and standard deviation F-scores are vizualized in Figure 7.2. Obviously, everything remains unchanged for **Elancourt**. However, we see how the standard deviations on the fused set seem to be greater than what observed previously in Figure 5.9. Added to the labels `BUS` and `FIB` that were previously improved by the use of image based features as shown in Table C.2, `BIT` and `FIT` F-scores are also greatly impacted on the new fused set. In fact, the first better performs when only geometric features are used as previously explained before in Section 5.2.2. For the Second, it was image based features that proved to be better suited. This agrees with the fact that, for the **Elancourt** → **Nantes** experiment, image based features were instrumental in better detecting `FIT` than training on **Nantes** itself (cf. Figure 5.3).

7.2.1 SVM results

Now that we discussed the RF results on the fused set, we can move on to the SVM experimental results on both identified urban sets: **Elancourt** and **Na-P13**. Results are reported in Table 7.2.

Two observations are worth noting herein:

	Elancourt							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	**97.67**	**86.44**	91.29	91.57	91.29	91.56	41.51	76.70
BUS	32.27	86.85	30.15	90.45	30.14	90.45	**42.89**	**92.66**
BIB	97.02	52.27	**91.09**	**89.75**	91.08	89.75	67.98	45.10
BIT	100	73.88	**100**	**100**	**100**	**100**	**100**	**100**
FOS	53.88	99.71	51.87	99.70	51.87	99.70	**63.06**	**94.08**
FUS	**96.49**	**52.24**	98.73	21.86	98.73	21.87	90.79	17.06
FIB	33.77	74.03	17.54	88.89	17.54	88.89	**71.93**	**93.71**
FIT	100	88.24	**100**	**100**	**100**	**100**	**100**	**100**
FIG	**84.57**	**88.47**	65.59	83.14	65.76	83.08	52.20	62.99

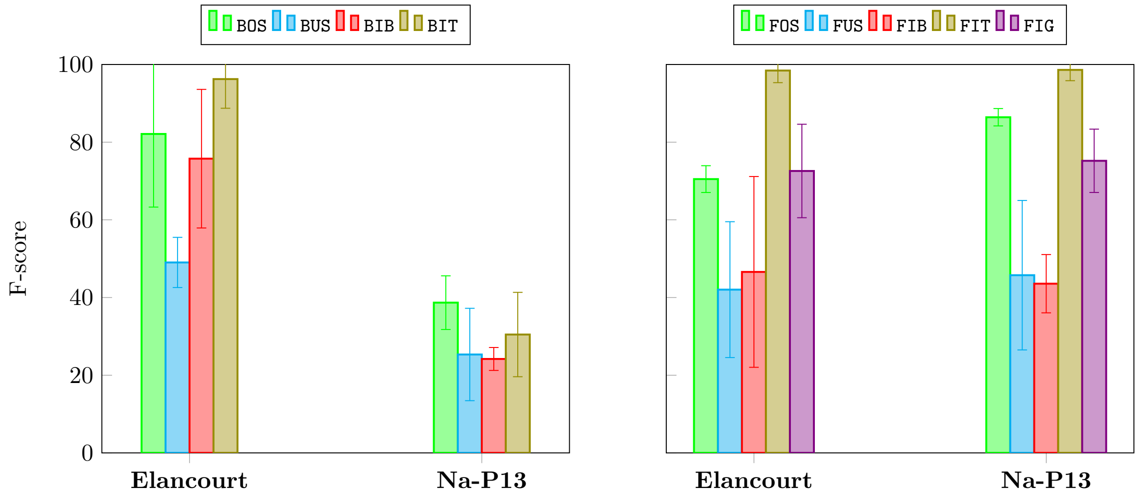
	Na-P13							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	**44.86**	**54.09**	29.98	42.69	29.98	42.69	29.98	42.69
BUS	**98.46**	**27.35**	36.15	13.51	41.54	15.21	30.0	11.44
BIB	**82.35**	**17.31**	70.59	13.53	70.58	13.53	70.59	13.53
BIT	**95.74**	**30.93**	50.26	16.67	50.26	16.67	50.26	16.67
FOS	98.90	75.08	99.31	74.77	99.31	74.69	**99.17**	**81.98**
FUS	**87.40**	**65.08**	30.79	43.70	30.79	43.70	30.79	43.70
FIB	**97.06**	**38.17**	71.90	27.88	71.90	27.88	70.36	27.07
FIT	100	89.47	**100**	**100**	**100**	**100**	**100**	**100**
FIG	**95.64**	**77.89**	71.39	77.91	71.27	77.88	60.36	72.81

Table 7.2: SVM results on the two datasets at **eFin** level 3.

- First is the fact that, contrarily to RF results, adding either height or image based features alone to the intrinsic features produced the same scores. However, when adding both, it yields different scores, as can be shown in Table C.19. This was the case for all errors but BIT and FIT on both sets, as well as FUS on **Na-P13**. One possible explanation is that both features have the same dynamic as they are both histogram values, as designed in Section 4.2. As a consequence, in this case, the SVM considers both feature configurations to be similar, unless when fed together to the SVM. This also could be the result of the fact that the parameterization of the classifier was not ideal and does not learn properly when external modalities are added.

- Secondly, in some cases, the SVM, for some particular feature configurations, yields results that exceed the other ones by a large margin. This was the case of FUS on **Elancourt** (*resp.* **Na-P13**) with a jump of around 31 % with the **Geom.** configuration (*resp.* 38 %) in F-score and 35 % for FIB on **Na-P13** with the **All** configuration. This could be owed to two possible reasons. Either these feature configurations are actually the best alternatives which is not conflicting with previous findings. Indeed, the extrinsic features were designed in the first place to detect fidelity errors such as FIB while FUS is a topological error that can be suitably detected using intrinsic features only. However, these large margins could be also explained by the fact that the SVM overfitted in these special cases. The last reason cannot be ruled out either, as the γ hyper-parameter was not optimized for these features as seen in Section 6.3.2, due to the high number of possible combinations.

As with the RF classifier, mean and standard deviation F-scores are computed and vizualized in Figure 7.3. Further commentary is left for the next sub-subsection as these results are compared to the RF ones.



(a) Building errors.

(b) Facet errors.

Figure 7.3: Mean F-score and standard deviation obtained with an SVM using baseline features.

7.2.2 SVM compared to RF

In Table 7.3, are compared the SVM F-scores (cf. Table C.19) to RF ones (cf. Table C.18). The same color scheme is used as with the Tables 5.3 and 5.4.

	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt	🟩	⬜ **All**	🟩	🟩	🟧 **All**	🟩	🟩 **Geom.**	🟩	🟩 **Geom.**
Na-P13	🟧 **Geom.**	🟧 **Geom.**	🟩 **Geom.**	🟩 **Geom.**	🟧 **All**	🟥	🟧 **Geom.**		⬜ **Geom.**

Table 7.3: Evolution of the F-score value, for each error using SVM compared to RF and based on baseline features. Feature sets having a significant impact on the classification results are mentioned in the corresponding cell (cf. Table C.19). The used color scheme is presented in figure 5.13.

As suspected, SVM yields better, or at least stable, results on highly unbalanced labels. In fact, BIB (*resp.* BIT and FIT) with less than 16 % (*resp.* 16 % and 1.5 %) occurence ratio in both sets gained in terms of F-scores when training was conducted using an SVM. The same pattern is observed for BUS (*resp.* FUS and FIB) with less than 25 % (*resp.* 19 % and 12 %) frequency but on **Elancourt** only. However, the same set of labels, as well as BOS, performs worse when an SVM is used on **Na-P13**, eventhough they both have a presence ratio strictly under 41 %. On the other hand, regarding the labels that are very frequent [40] (with more than 50 %), it is expected that they would underperform with the use of SVMs. Although this is true for FOS, on both sets, it is not the case for the other error labels.

In order to explain these exceptions, we tried to look at the best performing modalities, per label and set. Hence, we can compare between the two classifiers easily as shown in Table 7.4. Most labels perform best with the same kind of feature configurations. Most notably, we have previously suspected that using only geometric features resulted in overfitting with an SVM when training for FUS on both sets. Comparing to the RF,

[40] These are BOS on **Elancourt**, in addition to FOS and FIG on both sets.

	Elancourt		**Na-P13**	
	RF	SVM	RF	SVM
BOS	**Geom.**	**Geom.**	**Geom.**	**Geom.**
BUS	**Hei.**	**All**	**Im.**	**Geom.**
BIB	**All**	**Hei.**, **Im.**	**Geom.**	**Geom.**
BIT	**Geom.**	**Hei.**, **Im.**	**Geom.**	**Geom.**
FOS	**Hei.**	**All**	**All**	**All**
FUS	**Geom.**	**Geom.**	**Im.**	**Geom.**
FIB	**Geom.**	**All**	**All**	**Geom.**
FIT	**Hei.**	**Hei.**, **Im.**	**All**	**Hei.**, **Im.**
FIG	**Geom.**	**Geom.**	**All**	**Geom.**

Table 7.4: The best performing feature configuration per zone, label and classifier. This summarizes all comparisons between Tables C.18 and C.19. The features, that stand out compared to the others in these last tables, are underlined.

Geom. was also the best alternative. This leads us to speculate that, contrarily to what was discussed in Section 7.2.1, **Geom.** did not lead to overfitting. It was rather the SVM that did not learn properly (underfitting) using the other feature configurations.

Hereafter, we analyse the cases where RF and SVM did not share the same best performing feature configuration.

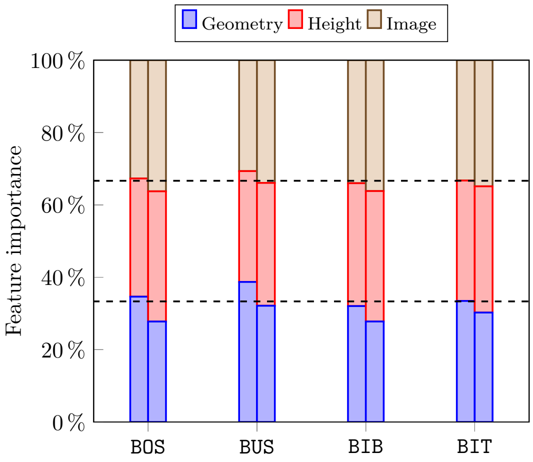
- On **Elancourt**, for BUS, FOS and FIT, the RF performed better with height based features, compared to the SVM which works better with other feature configurations containing extrinsic features. Moreover, BIT was better detected with intrinsic features only (*resp.* any extrinsic feature based configuration) when trained with an RF (*resp.* SVM). This does not contradict previous findings in Sections 5.2.2 and 5.3.

- On **Na-P13**, the only labels where the better performing feature configuration changed were BUS, FUS, FIB and FIG. In fact, only geometric features yielded better results using an SVM, in contrast with the fact that **All** or **Im.** were the best alternative with an RF.

This reinforces the idea that the SVM was actually not adequately parameterized for the extrinsic features as was announced in Section 6.3.2. In fact, as discussed earlier in Section B.2, SVMs are very hard to parameterize compared to RFs. This seems to be a reasonable cause why the SVM classifier underperforms compared to the RF one on **Na-P13**.

7.2.3 Modality contributions comparison

In a linear SVM, feature importance can be natively computed by looking at the weight vector $\boldsymbol{w}$ as shown by Guyon et al. (2002). This is, unfortunatly, not the case. Instead we operate with Kernel SVMs where features are fused using MKL with weights $(\mu_i)_{i=1,2,3}$ (cf. Equation B.18) that are on the simplex, as discussed in Section B.2. Consequently, when training with the last feature configuration, the resulting weights could be interpreted as feature importance ratios. These are vizualized in Figure 7.4.



(a) `Building errors.`

Figure 7.4: Modality contribution for the SVM classifier based on the coefficients computed by EasyMKL. The first (*resp.* second) column represents **Elancourt** (*resp.* **Na-P13**).

We can see how these ratios stay mostly around 1/3 with more leeway compared to RF (cf. Figure 5.10). This actually confirms how these MKL weights could be interpreted as features importances. The larger margins to the 1/3 ratio could be explained by the sensibility of the weights to the noise when selecting training instances. There is, however, one exception that could be noted between Figures 7.4 and 5.10. In fact, for `FOS`, geometric features have a larger importance when training with SVM compared to the RF case. This is actually not an issue as it could be explained by the topological nature of the error label.

Eventhough geometric features alone yield better results in some cases, the MKL cannot ignore the extrinsic features. As in the RF case, this may prove to be helpful for the transferability of learning. Consequently, it would be interesting to rerun the same scalability experiments, as in Section 5.3, with an SVM. This was not the case, due to time limitations.

7.2.4 Summary

To summarize, the aim of this study was to assess the impact of the classifier choice. Based on previous findings (cf. Sections 5.2, 5.3 and 5.4), we fused both sets **Nantes** and **Paris-13** into one: **Na-P13**. Consequently, we learned that:

- Fusing **Nantes** and **Paris-13** helps better train the RF classifier for previously difficult labels `FIT` and `BIT`;
- As previously expected, the SVM was not well parameterized to take full advantage of the extrinsic features;

- The SVM classifier is much better than the RF one, for highly unbalanced labels: `BIB`, `BIT` and `FIT`.
- `FOS`, being very present in both sets, loses in F-score when the SVM is used;
- Just as the RF classifier, it is hard for the SVM to learn on **Na-P13**;
- Like when using RF, all modalities are equally important for the SVM classifier.

7.3 Advanced features contributions

The goal, herein, is to find out if it is possible to achieve better results than the ones obtained with our handcrafted baseline features. As we do not have enough learning instances to leverage deep learning methods, we choose instead to use graph kernels as well as ScatNets as shown in Section 6.2 and 6.3.1.

Three possibilities are investigated:

i) We keep the baseline geometric features and replace the basic image and height based features by the ScatNet derived extractors, as explained in Section 6.2.2;

ii) We compare both the baseline features and graph kernels for geometric features alone;

iii) We combine both the graph kernels and the ScatNet derived extractors and compare them to the previous results.

As with the previous section (cf. Section 7.2), the experiments are conducted on the two sets: **Elancourt** and **Na-P13**.

7.3.1 ScatNet to baseline comparison

We start by the ScatNet comparisons. We run the same experiments, as in Section 7.2, where this time height and image based features are replaced by derived ones. There are two options when employing ScatNet with image based features: `channel` and `deletion` (cf. Section 6.3.1). This makes the number of possible feature configurations equal to six.

Both the RF and SVM classifiers are used. Results from both are first compared to the baseline results reported in Section 7.2. Afterwhat, we examine the differences between results from both classifiers.

RF results.

Results, using the RF classifier, on both sets, are reported in Table 7.5. In general, we can see how external modalities seem to play a more important role in detecting errors. This can be confirmed with the larger standard deviations depicted in Figure 7.5 compared to the baseline features (cf. Figure 7.2).

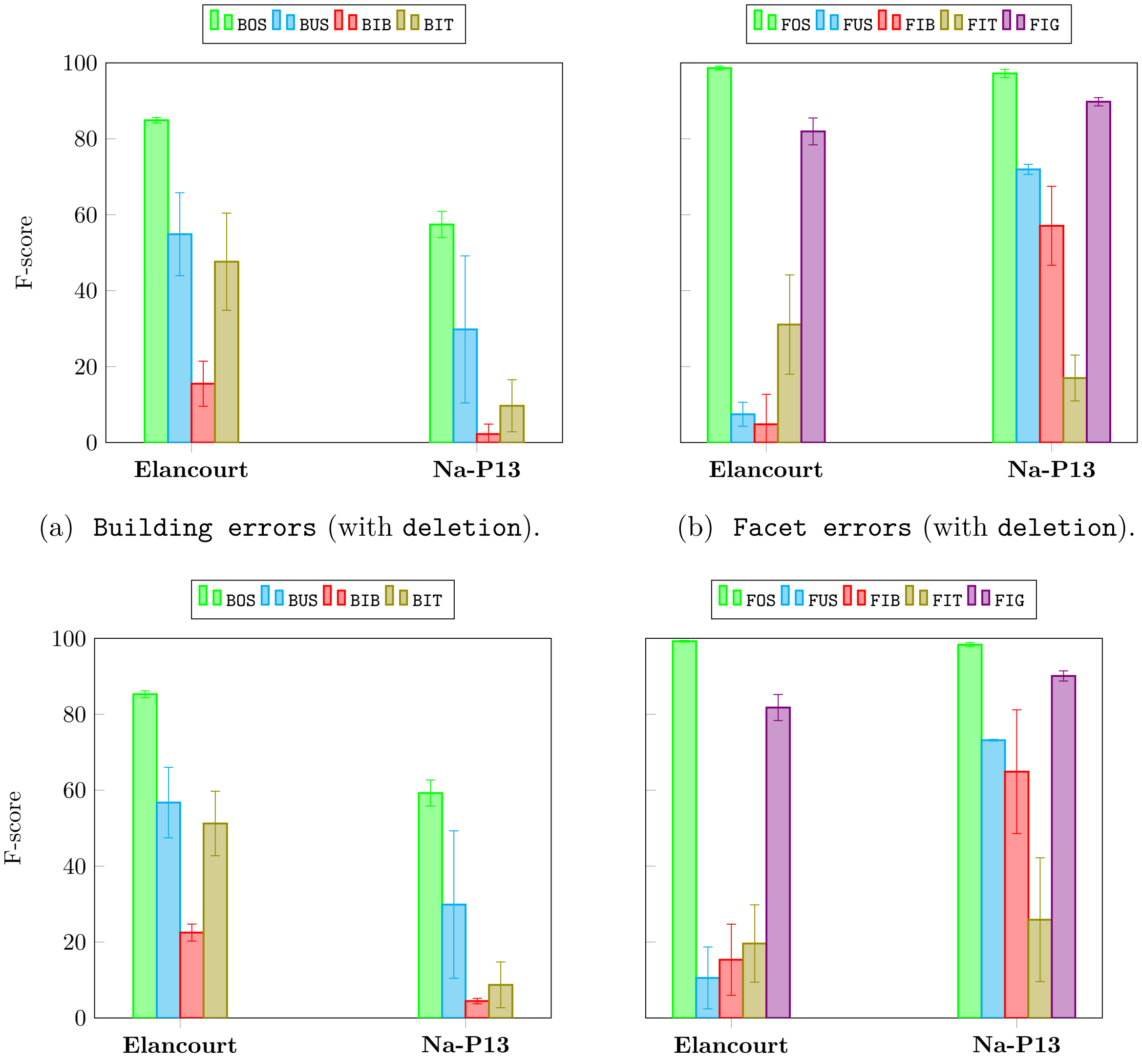
In addition, based on the same Figure 7.5, we can deduce the best option to be used for ScatNet image based features. In most cases, the best options seems to be almost always `channel`. This is understandable as this option does not modify the signal, from the get go, and preserves both the image and model information until the last possible opportunity letting the classifier handle the fusion. The early fusion conducted when

Elancourt												
	Geom.		**Geom. $\oplus$ S-Hei.**		**Geom. $\oplus$ S(d)-Im.**		**S(d)-All**		**Geom. $\oplus$ S(c)-Im.**		**S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	93.96	76.15	91.97	79.13	97.75	74.49	94.52	78.49	96.55	77.24	**94.89**	**78.80**
`BUS`	32.98	76.47	**52.44**	**84.88**	29.51	92.67	50.32	87.13	36.09	91.89	49.79	90.38
`BIB`	12.32	67.57	11.38	100	5.45	100	5.45	100	13.37	100	**14.36**	**100**
`BIT`	25.25	92.59	**42.86**	**100**	20.41	100	39.80	100	34.69	100	36.73	100
`FOS`	98.91	99.07	99.14	99.14	99.30	97.25	99.46	96.82	99.61	99.23	**99.69**	**99.23**
`FUS`	1.90	54.55	3.18	100	4.78	100	5.73	100	5.41	100	**12.42**	**100**
`FIB`	9.17	87.5	0.87	100	0.44	100	0	—	**12.28**	**100**	11.84	100
`FIT`	6.67	100	20.69	100	**27.59**	**100**	20.69	100	10.34	100	6.90	100
`FIG`	80.54	73.14	94.83	74.85	96.27	73.57	**97.12**	**74.13**	93.98	74.23	95.17	74.87

Na-P13												
	Geom.		**Geom. $\oplus$ S-Hei.**		**Geom. $\oplus$ Im.**		**All**		**Geom. $\oplus$ S(c)-Im.**		**S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	**51.65**	**78.93**	39.92	85.84	39.71	96.5	40.95	90.87	44.65	95.59	43.21	93.75
`BUS`	19.85	100	0.76	100	**27.69**	**100**	25.95	100	**27.69**	**100**	26.15	100
`BIB`	1.96	100	**2.61**	**100**	0	—	0	—	2.60	100	1.96	100
`BIT`	5.32	100	0	—	**8.51**	**100**	6.91	100	7.41	100	5.82	100
`FOS`	98.62	98.22	99.31	95.87	99.31	94.99	99.17	92.65	98.62	98.62	**98.62**	**98.75**
`FUS`	68.80	77.44	70.10	76.75	61.57	84.18	61.57	82.55	**65.08**	**84.0**	65.57	82.81
`FIB`	55.23	78.60	26.80	95.35	46.41	94.04	43.46	94.33	68.62	85.71	**68.95**	**86.12**
`FIT`	6.25	100	6.25	100	12.5	100	12.5	100	**25.0**	**100**	**25.0**	**100**
`FIG`	94.55	82.54	98.42	83.45	97.70	83.71	98.18	83.85	97.58	85.10	**97.94**	**84.96**

Table 7.5: RF applied to ScatNet based features. Results are expressed in percentage on the two datasets at **eFin** level 3.

choosing the `deletion` option deforms the input signal. As the ScatNet convolves filters and applies non linear functions to the input, the classifier cannot separate, in this case, between information that is derived from the image or from the evaluated model.



(a) `Building errors` (with `deletion`).

(b) `Facet errors` (with `deletion`).

(c) `Building errors` (with `channel`).

(d) `Facet errors` (with `channel`).

Figure 7.5: Mean F-score and standard deviation obtained with an RF based on ScatNet features. This is a vizualization of scores recorder in Table C.25.

There are however exceptions that we divide into two cases. The first is where `deletion` was better with a small margin that can be explained by the noise of training data selection. This was the case of `BIT` on **Elancourt** and `FIG` on **Na-P13**. On the contrary, the Second, and more important, case is where the margin is large which was the case of `FIT` on **Elancourt** where **Geom.** $\oplus$ **S(d)-Im.** had at least 9 % more in F-score than the other configurations. At this point we do not have any explication to why this is the case.

The new results are compared, for each ScatNet option, error label and set, to the baseline configuration scores (cf. Table C.21). These are shown in Table 7.6 with the same color scheme as in previous comparisons. On `Building errors`, ScatNet with `deletion` yields better (*resp.* worse and stable) results 3 (*resp.* 3 and 2) times. On `Facet errors`, the same option yields better (*resp.* worse and stable) results 3 (*resp.* 2 and 5) times. In

	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt		**S-Hei.**	**Geom.**	**S-Hei.**		**S(d)-Im.**	**Geom.**	**S(d)-Im.**	
Na-P13	**Geom.**	**S(d)-Im.**		**S(d)-Im.**				**S(d)-Im.**	

(a) Comparison with `deletion` option.

	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt		**S-Hei.**		**S-Hei.**		**S(c)-All**	**S(c)-Im.**	**S-Hei.**	
Na-P13		**S(c)-Im.**					**S(c)-Im.**	**S(c)-Im.**	

(b) Comparison with `channel` option.

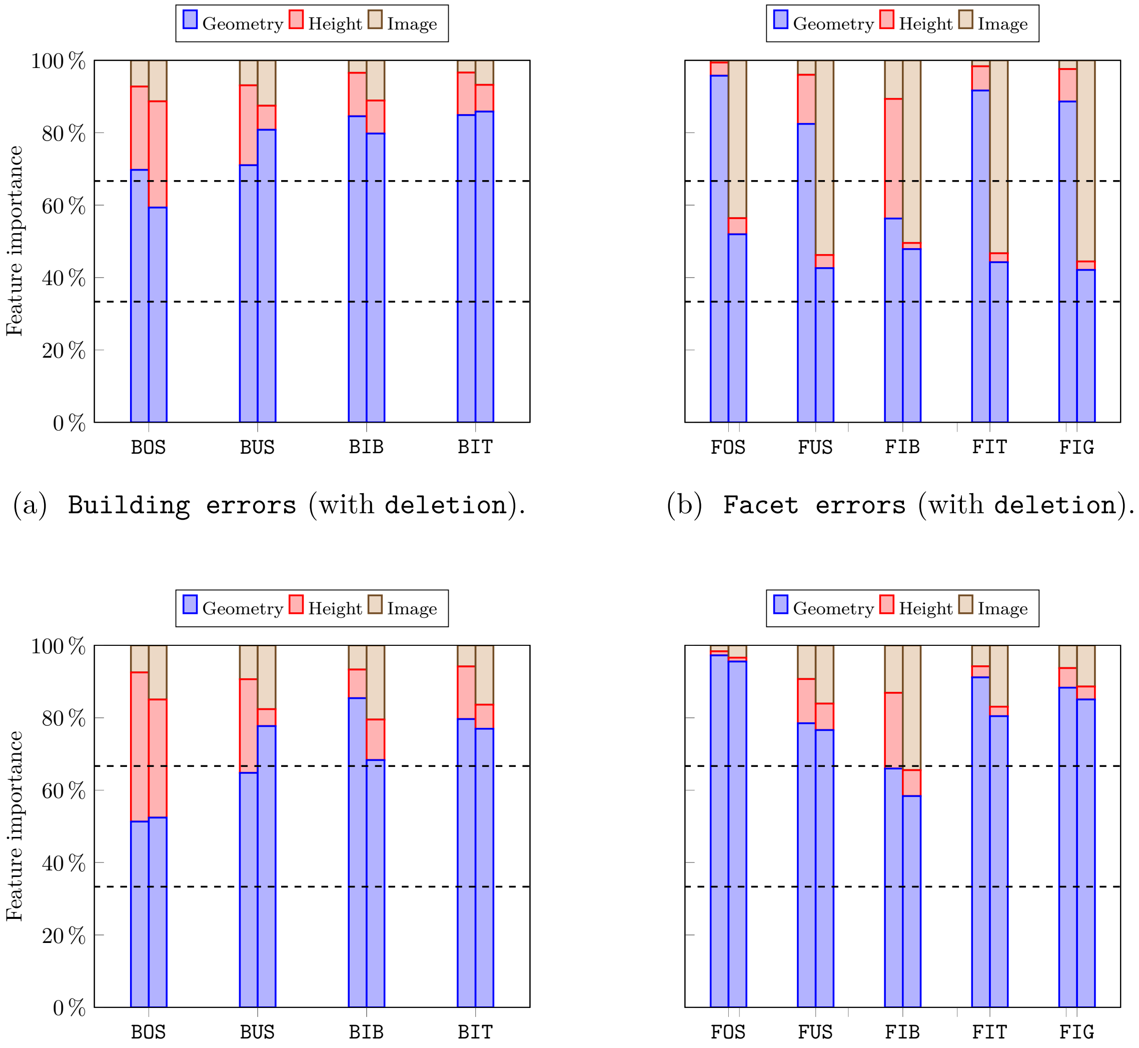
Table 7.6: Evolution of the F-score value, for each error, with the RF classifier, using ScatNet compared to baseline features. Feature sets having a significant impact on the classification results are mentioned in the corresponding cell (cf. Table C.21). The used color scheme is presented in figure 5.13.

constrast, regarding `Building errors`, the `channel` option helps ScatNet achieve, better (*resp.* worse and stable) results 2 (*resp.* 1 and 5) times. On `Facet errors`, the same option yields better (*resp.* worse and stable) results 6 (*resp.* 0 and 3) times. This confirms again the fact that the `channel` option was better than `deletion`.

We can also observe how height based features with ScatNet are more instrumental than the baseline features, especially for `BIT` and `FIT` on **Elancourt**. For `BUS` on **Elancourt**, baseline height based features were also instrumental but in the same capacity as image based ones. It is not the case anymore when adding ScatNet based ones: it is better than both baseline and advanced image based features. On **Na-P13**, this modality does not play an important role even with the more advanced feature extractor. This may be attributed to the fact that on both zones, **Nantes** and **Paris-13**, building height profiles are mostly the same, especially since the types of these buildings are less heterogeneous than on **Elancourt**.

Regarding image based features with ScatNet, they proved to be more decisive in error prediction, for both options. Although it fails to give better results for `BIB` for both sets and `BUS` on **Na-P13**, it is more helpful than baseline features, especially for topological error labels. In fact, contrarily to baseline image features, it is crucial in better detecting `BIT` on **Na-P13** (with `deletion`), `FUS` and `FIG` on **Elancourt**, as well as, `FIB` (with `channel`) and `FIT` on both sets.

As with previous experiments, we also computed feature importances using the new configurations. These are normalized and shown for both sets and options in Figure 7.6. The normalization consists in weighting the importance ratios by the inverse of the corresponding feature vector length. This is conducted in order to put all modalities at an equal footing eventhough the ScatNet produces a lot of features.



(a) `Building errors` (with `deletion`).

(b) `Facet errors` (with `deletion`).

(c) `Building errors` (with `channel`).

(d) `Facet errors` (with `channel`).

Figure 7.6: Normalized modality importance using the RF classifier and ScatNet features. Height and Image based features use the ScatNet with both `deletion` and `channel` options. The first (*resp.* second) column represents **Elancourt** (*resp.* **Na-P13**).

First of all, we see how the depth based modality plays a more important role on **Elancourt** compared to the other set as seen in Table 7.6. In general, it has an importance ratio less than 20 % except for `BOS` and `FIB` on both sets, as well as `BUS` on **Elancourt**.

Regarding image based features, we can notice that the `channel` option results in less importance than the `deletion` one. This can be explained by the fact that the latter contains 3/4 times less coefficients than the other case. However, this reason fails to describe the case where the discrepancy between the two cases is too important. In fact, for `FIB` on **Na-P13**, as well as `FIT` and `FIG` on both sets, the image based features are too important with `deletion` than with `channel`. This is actually not beneficial for learning as height based features were essential to achieve better results. This means that the fact that the `channel` option was better not only because it is on its own better than the alternative, but also because it works better with height based features.

When examnining the `channel` option (cf. Figures 7.6c and 7.6d) more closely, we see how for both sets the image based modality importance falls in the range 5 to 10 % in most cases. There are two cases where this is not the case:

- `FOS`, on both sets, with an importance below 1 %. This is understandable as this error is of topological nature and is better detected with intrinsic features. Indeed, for this error even height based features score less than 1 % in importance ratio.

- `FIB`, on **Na-P13** with a ration around 1/3. This can be explained by the fact that this modality was decisive in getting the best F-score possible (cf. Table 7.6).

On another note, building typology is so important that, even with advanced extrinsic features, baseline geometric features have a very large importance ratio.

SVM results.

We run the same experiments and follow the same layout as in Section 7.3.1. Results are reported in Table 7.7. The mean and standard deviation F-scores are shown in Figure 7.7. Just as with the RF experiments, external modalities seem to play a more crucial role in error prediction than with baseline features (cf. Figure 7.3).

Based on Figure 7.7, we can compare the options of ScatNet derived image based features based on the produced results. In constrast with RF, in most cases, it is this time the `deletion` option that yields the best results. The opposite occurs only three times: the `channel` option was best on **Elancourt** for `BOS` and `BIT` with a small margin, and for `BIB` with a jump of more that 20 % in terms of F-score. As in the previous sub-subsection, we do not have any theory explaining why the last exception occurs.

On the other hand, regarding the inversion of fusion scheme preference, there are two possible explanations:

- The analysis that was presented in Section 7.3.1 is false;

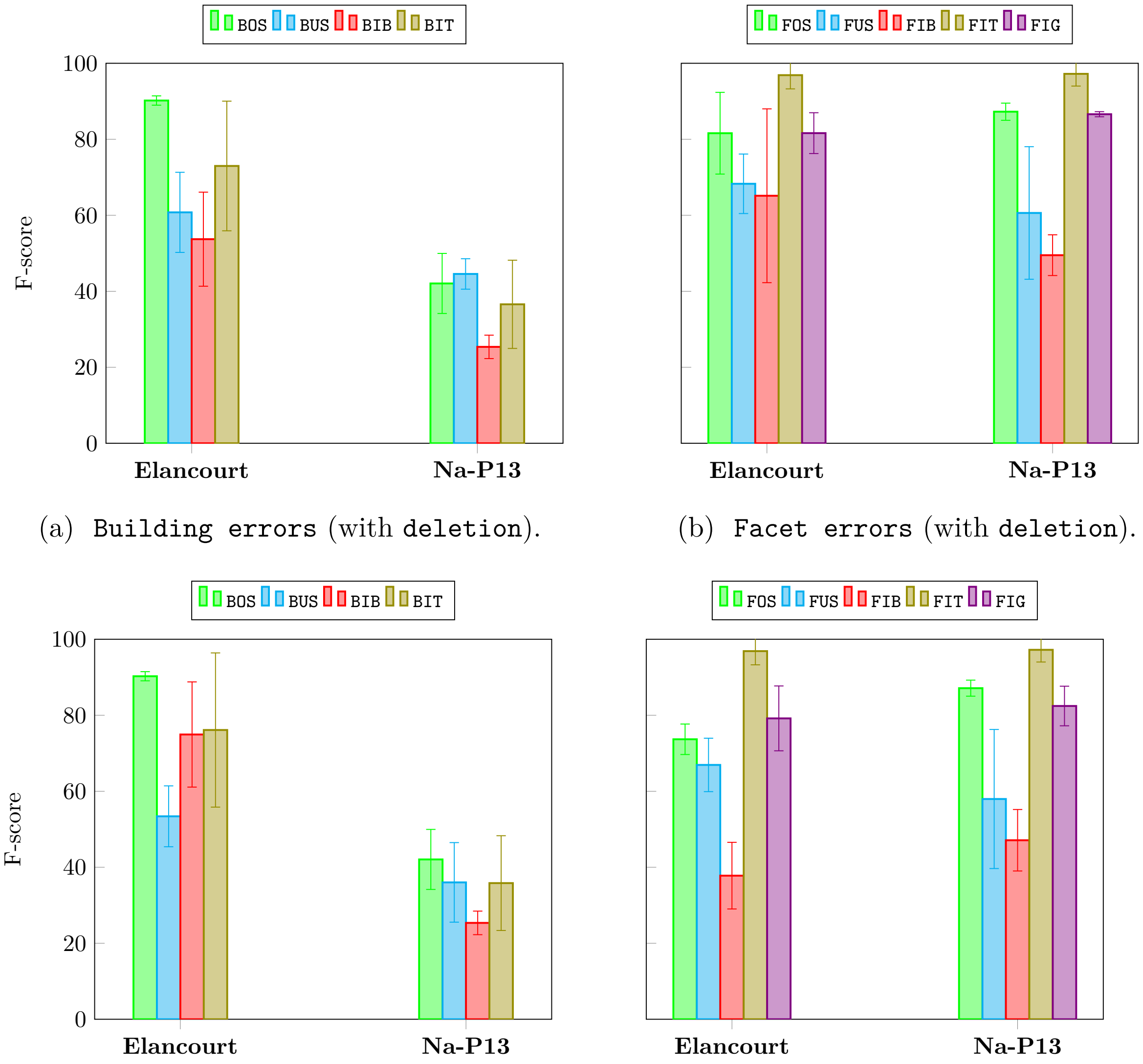
- The issue is rather with the SVM that was not well parameterized.

The last reason seems to be more probable as the late fusion schemes induces larger feature vector dimensions (1085 more than the other option) which may have caused problems for the SVM. In addition, this is consistent with the large drop in the performances of the SVM classifier when baseline external features are added, as shown in Table 7.2.

Elancourt												
	Geom.		**Geom.** ⊕ **S-Hei.**		**Geom.** ⊕ **S(d)-Im.**		**S(d)-All**		**Geom.** ⊕ **S(c)-Im.**		**S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	**97.67**	**86.44**	97.07	81.99	95.12	86.43	97.15	83.16	91.67	89.71	93.32	86.56
BUS	32.27	86.85	89.81	46.48	**60.08**	**92.18**	89.60	47.42	30.79	90.63	89.41	44.42
BIB	97.02	52.27	89.60	44.80	98.51	25.68	97.03	30.53	**91.13**	**90.69**	94.06	71.43
BIT	100	73.88	91.84	38.96	98.98	82.20	94.90	46.5	**100**	**100**	93.88	49.46
FOS	53.88	99.71	70.22	87.42	**94.17**	**97.04**	78.46	88.05	54.59	99.72	69.98	83.96
FUS	96.49	52.24	96.82	40.92	**95.59**	**62.26**	94.27	58.85	98.09	51.85	97.13	60.52
FIB	33.77	74.03	31.58	74.23	**89.04**	**81.53**	87.72	81.97	17.98	89.13	18.42	89.36
FIT	100	88.24	100	88.24	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
FIG	**84.57**	**88.47**	86.28	85.56	64.92	91.30	68.31	91.49	68.56	83.75	63.05	76.00

Na-P13												
	Geom.		**Geom.** ⊕ **S-Hei.**		**Geom.** ⊕ **S(d)-Im.**		**S(d)-All**		**Geom.** ⊕ **S(c)-Im.**		**S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	**44.86**	**54.09**	44.65	53.71	29.98	42.69	29.98	42.69	29.98	42.69	29.98	42.69
BUS	98.46	27.35	97.69	25.76	93.08	29.30	**93.07**	**34.28**	37.40	14.04	71.54	27.84
BIB	**82.35**	**17.31**	79.74	16.55	70.59	13.55	70.59	13.55	70.59	13.53	70.59	13.53
BIT	**95.74**	**30.93**	95.21	30.76	53.49	17.63	52.66	17.71	50.26	16.67	50.26	16.67
FOS	98.90	75.08	95.45	82.78	**99.72**	**74.54**	99.17	81.89	99.45	74.64	98.34	81.67
FUS	**87.40**	**65.08**	87.60	60.31	30.79	43.70	59.50	61.02	30.79	43.70	45.45	54.73
FIB	**97.06**	**38.17**	97.39	36.79	79.08	30.75	81.37	31.60	71.90	27.88	71.90	27.78
FIT	100	89.47	100	89.47	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
FIG	95.64	77.89	95.39	78.62	97.70	78.63	**97.82**	**78.65**	72.12	78.08	83.76	81.68

Table 7.7: SVM applied to ScatNet based features. Results are expressed in percentage on the two datasets at **eFin** level 3.



(a) `Building errors` (with `deletion`).

(b) `Facet errors` (with `deletion`).

(c) `Building errors` (with `channel`).

(d) `Facet errors` (with `channel`).

Figure 7.7: Mean F-score and standard deviation obtained with an SVM based on ScatNet features.

	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt		🟩 **S(d)-Im.**	🟧 **Geom.**	🟨 **S(d)-Im.**	🟩 **S(d)-Im.**	🟩 **S(d)-Im.**	🟩 **S(d)-Im.**	**S(d)-Im.**	
Na-P13		🟩 **S(d)-All**				🟩		**S(d)-Im.**	

(a) Comparison with `deletion` option.

	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt		**S-Hei.**	**S(c)-Im.**	**S(c)-Im.**		🟩 **S(c)-All**		**S(c)-Im.**	**Geom.**
Na-P13						🟩		**S(c)-Im.**	

(b) Comparison with `channel` option.

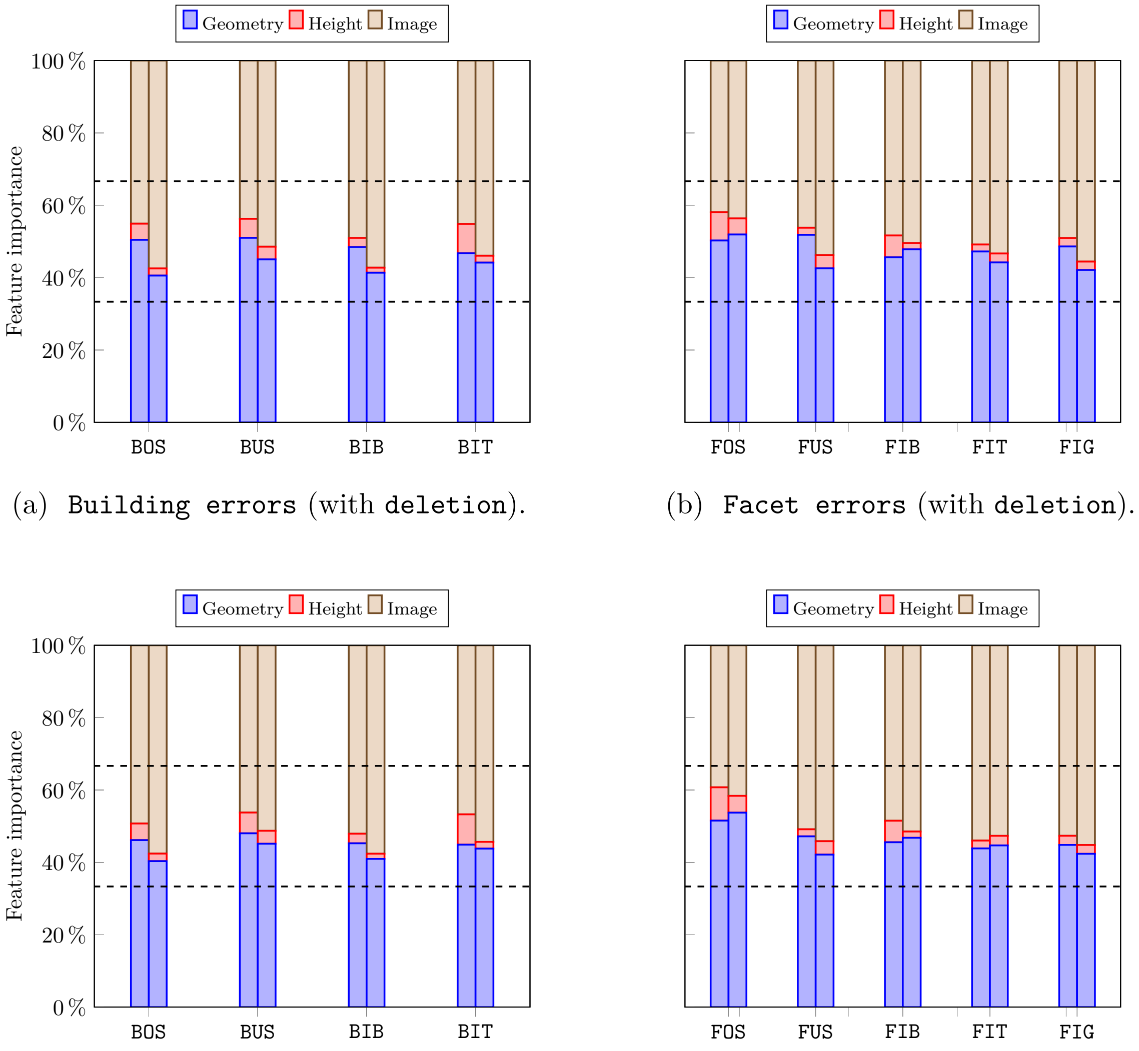
Table 7.8: Evolution of the F-score value, for each error, with the SVM classifier, using ScatNet compared to baseline features. Feature sets having a significant impact on the classification results are mentioned in the corresponding cell (cf. Table C.22). The used color scheme is presented in figure 5.13.

The scores obtained using SVM and ScatNet features are compared to the baseline (cf. Table C.22) with the same color scheme as earlier. These comparisons are summarized in Table 7.8. On `Building errors`, ScatNet with `deletion` yields better (*resp.* worse and stable) results 2 (*resp.* 4 and 2) times. On `Facet errors`, the same option yields better (*resp.* worse and stable) results 4 (*resp.* 6 and 0) times. In constrast, regarding `Building errors`, the `channel` option helps ScatNet achieve, better (*resp.* worse and stable) results 0 (*resp.* 0 and 6) times. On `Facet errors`, the same option yields better (*resp.* worse and stable) results 2 (*resp.* 0 and 8) times. This confirms the fact than `deletion` was, in general, the best option when using an SVM.

Just as with the RF classifier, height based features with ScatNet help yield better results. This was the case for `BUS` on **Elancourt** with the `channel` option. This confirms once again the results fo earlier experiments where the same modality always proved to be important for predicting `BUS` on **Elancourt**. Regarding image based features, Table 7.8 shows that they are, as always, crucial for error detection. In the SVM case, it was with the `deletion` option compared to RF, as discussed earlier in the previous paragraph. This was suspected to be due to the fact SVM was not well parameterized, as was the case with baseline features in Section 7.2.

Once again, feature importances are computed, using the MKL weights, and vizualized in Figure 7.8. There is little difference, this time, between both fusion schemes, eventhough thery produce different results. Geometric and image based features are the most important modalities with ratios around 45 % in all sets and for both options.

Height based features, on the other hand, have a ratio of importance below 10 %. Actually, in most cases, it is below 4 %, except for `BIT`, `FOS` and `FIB` on **Elancourt**. This is different from baseline features, where height based features had comparable importance ratios to other modalities (`cf.` Figure 7.4). It seems to be contradictory with the fact



(a) `Building errors` (with `deletion`).

(b) `Facet errors` (with `deletion`).

(c) `Building errors` (with `channel`).

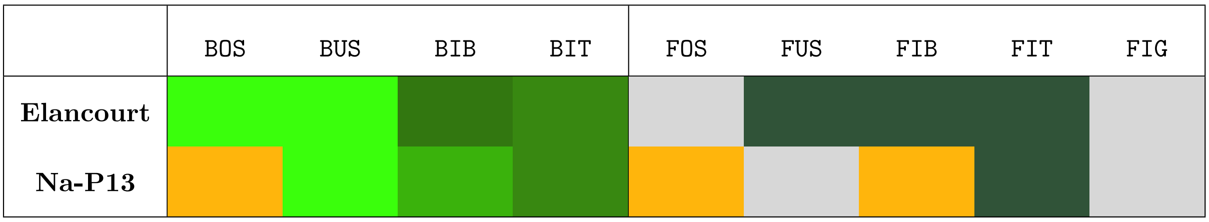
(d) `Facet errors` (with `channel`).

Figure 7.8: Modality contribution for the SVM classifier based on the coefficients computed by EasyMKL. Height and Image based features use the ScatNet with both `deletion` and `channel` options. The first (*resp.* second) column represents **Elancourt** (*resp.* **Na-P13**).

that ScatNet derived height features performed overall better than baseline features (cf. Tables C.18 and C.19). It is, however, consistent with the findings of the scalability analysis in Section 5.3, where height based features proved to not be important. This further bolsters the fact that the SVM was not well parameterized for the image based features.

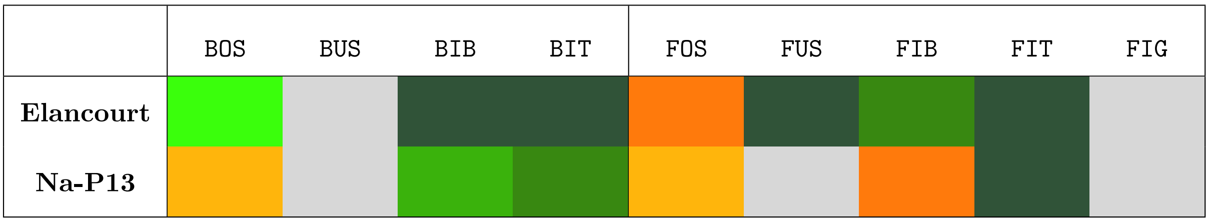
SVM compared to RF.

Just as in the Section 7.2, we will compare preditction scores resulting from both classifiers. These are compiled in Table 7.9.



	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt	🟩	🟩	🟩	🟩	▨	🟩	🟩	🟩	▨
Na-P13	🟨	🟩	🟩	🟩	🟨	▨	🟨	🟩	▨

(a) Comparison with `deletion` option.



	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt	🟩	▨	🟩	🟩	🟧	🟩	🟩	🟩	▨
Na-P13	🟨	▨	🟩	🟩	🟨	▨	🟧	🟩	▨

(b) Comparison with `channel` option.

Table 7.9: Evolution of the F-score value, for each error using SVM compared to RF and based on ScatNet features. The color indicates the magnitude: ■: (−100, −45 %]– ■: [−45, −35 %)– ■: [−35, −25 %) – ■: [−35, −25 %) – ■: [−15, −5 %) – ■: [−5, 5 %) – ■: [5, 15 %) – ■: [15, 25 %) – ■: [25, 35 %) – ■: [35, 45 %) – ■: [45, 100 %] – When two null F-scores are compared, the cell is colored in white □: neither positive nor negative.

On **Elancourt**, the SVM classifier yields better results than the RF one, especially with the `deletion` option. On **Na-P13**, the SVM classifier is not always better, but `deletion` remains the best option. On both sets, using ScatNet derived features helped the SVM score better than RF compared to the baseline features. Hereafter, in Table 7.10, based on this comparison, as well as, the ones in Sections 7.3.1 and 7.3.1, we can deduce the best F-score, feature configuration and classifier, per error label.

We can observe that for **Elancourt** the SVM classifier yields always the best results, with the exception of `FOS`. This exception could be explained easily by the fact that `FOS` is very frequent, as discussed in Section 7.2.2. Moreover, on this set, most error labels achieve an F-score is over 85 %, excluding `BUS` and `FUS` with at least 70 %. Regarding feature configurations, the ScatNet derived ones are always the best with two exceptions: `BOS` and `FIG`. In these two cases, geometric baseline features yield slightly better results than the more advanced ScatNet based ones.

On **Na-P13**, results are not as positive as in the other set. Although for `Facet errors`, F-scores are as good[41] as in **Elancourt**, they are poor for `Buiding errors`. Besides, advanced features did not yield the best results for most cases (6 over 9 labels).

[41] Over 74 %, for `FUS` and `FIB`, and 90 % for the rest.

		Elancourt	
	$\boldsymbol{F_{score}}$	Feature configuration	Classifier
`BOS`	91.71	**Geom.**	SVM
`BUS`	72.75	**Geom. ⊕ S(d)-Im.**	SVM
`BIB`	90.91	**Geom. ⊕ S(c)-Im.**	SVM
`BIT`	100	**Geom. ⊕ Hei.** **Geom. ⊕ Im.** **All** **Geom. ⊕ S(c)-Im.**	SVM
`FOS`	99.46	**S(c)-All**	RF
`FUS`	75.41	**Geom. ⊕ S(d)-Im.**	SVM
`FIB`	85.12	**Geom. ⊕ S(d)-Im.**	SVM
`FIT`	100	**Geom. ⊕ Hei.** **Geom. ⊕ Im.** **All** **Geom. ⊕ S(d)-Im.** **S(d)-All** **Geom. ⊕ S(c)-Im.** **S(c)-All**	SVM
`FIG`	86.48	**Geom.**	SVM

		Na-P13	
	$\boldsymbol{F_{score}}$	Feature configuration	Classifier
`BOS`	62.44	**Geom.**	RF
`BUS`	52.46	**Geom. ⊕ Im.**	RF
`BIB`	28.61	**Geom.**	SVM
`BIT`	46.76	**Geom.**	SVM
`FOS`	98.69	**All.**	RF
`FUS`	74.61	**Geom.**	SVM
`FIB`	76.58	**S(c)-All.**	RF
`FIT`	100	**Geom. ⊕ Hei.** **Geom. ⊕ Im.** **All** **Geom. ⊕ S(d)-Im.** **S(d)-All** **Geom. ⊕ S(c)-Im.** **S(c)-All**	SVM
`FIG`	90.99	**S(c)-All**	RF

Table 7.10: For each set and each error label, we report the best F-score as well as feature configurations and classifiers.

Added to that the fact that the SVM classifier was only the best for rare errors.

Overall, we can see how `Building errors` are best learned on **Elancourt**. This is in accord with what we have seen in Section 5.2.2, 5.3 and 7.2. On the other hand, for `Facet errors`, according to the same sections, we would expect **Na-P13** to be the best alternative. However, there is one exception as `FIB` yield a better F-score on **Elancourt** than on **Na-P13**. We do not have any explanation to why this happens other than the fact that the SVM was not better parameterized.

7.3.2 Graph kernels to baseline comparison

Herein, we aim at assessing the added value of graph kernels in error classification, compared to the baseline. This means that we have only one feature configuration to experiment with. We only use the SVM which naturally takes kernels into account. Results are reported in Table 7.11 and visualized in Figure 7.9.

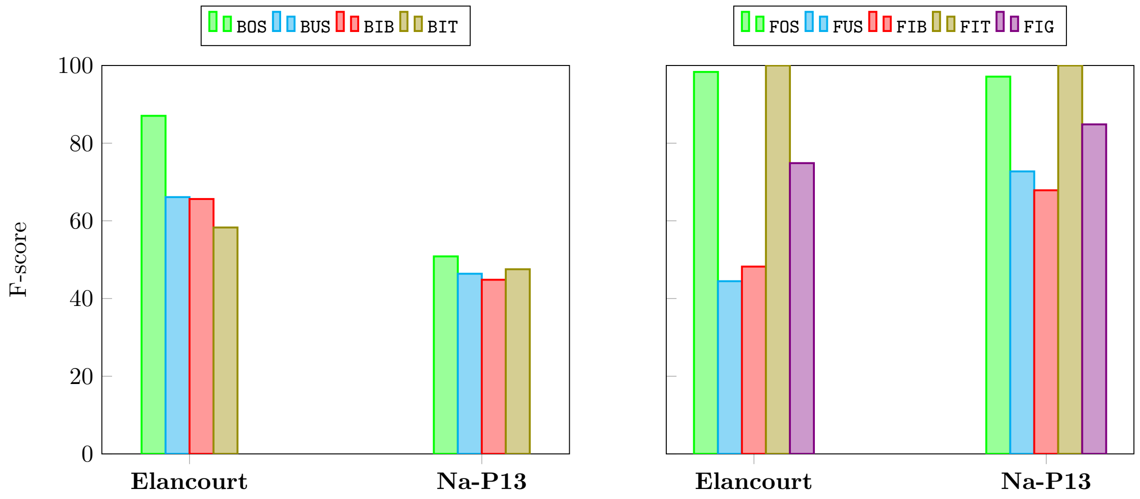
Elancourt						
	Geom.			**K-Geom.**		
	Rec	***Prec***	$\boldsymbol{F_{score}}$	***Rec***	***Prec***	$\boldsymbol{F_{score}}$
`BOS`	97.67	86.44	**91.71**	87.99	86.11	87.04
`BUS`	32.27	86.85	47.06	92.99	51.29	**66.11**
`BIB`	97.02	52.27	**67.94**	72.28	60.08	65.62
`BIT`	100	73.88	**84.98**	96.94	41.67	58.29
`FOS`	53.88	99.71	69.96	97.12	99.60	**98.34**
`FUS`	96.49	52.24	**67.78**	84.39	30.18	44.46
`FIB`	33.77	74.03	46.38	94.74	32.34	**48.22**
`FIT`	100	88.24	93.75	100	100	**100**
`FIG`	84.57	88.47	**86.48**	69.66	80.91	74.86

Na-P13						
	Geom.			**K-Geom.**		
	Rec	***Prec***	$\boldsymbol{F_{score}}$	***Rec***	***Prec***	$\boldsymbol{F_{score}}$
`BOS`	44.86	54.09	49.04	43.42	61.34	**50.85**
`BUS`	98.46	27.35	42.81	86.15	31.73	**46.38**
`BIB`	82.35	17.31	28.61	75.16	31.94	**44.83**
`BIT`	95.74	30.93	46.76	87.23	32.67	**47.54**
`FOS`	98.90	75.08	85.36	95.45	98.86	**97.13**
`FUS`	87.40	65.08	**74.61**	79.34	67.13	72.73
`FIB`	97.06	38.17	54.79	92.16	53.71	**67.87**
`FIT`	100	89.47	94.44	100	100	**100**
`FIG`	95.64	77.89	**85.86**	80.73	89.40	84.84

Table 7.11: Comparison between the baseline geometric features and graph kernels using SVM. Results, expressed in percentage, on the two datasets at **eFin** level 3.

As in the previous subsection, we draw comparisons between the graph kernel and baseline results. These are summarized in Table 7.12.

We can see that on **Na-P13**, grapĥ kernels yield better scores than baseline features for `BIB`, `FOS`, `FIB` and `FIG`, while they are stable on the rest. On **Elancourt**, it is a different situation. In fact, while it is, same as on the other set, better on `BOS` and `FIT`, the F-score remains stable for `BIB` and `FIB`. Moreover, it is better on `BUS` but worse on `BIT`, `FUS` and `FIG`.



(a) Building errors.

(b) Facet errors.

Figure 7.9: F-score obtained with an SVM using graph kernels features.

	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt	⬜	🟩	⬜	🟥	🟩	🟧	⬜	🟩	🟨
Na-P13	⬜	⬜	🟩	⬜	🟩	⬜	🟩	🟩	⬜

Table 7.12: Evolution of the F-score value, for each error using on graph kernels compared to baseline features. The used color scheme is presented in figure 5.13.

These discrepancies could be explained by looking at each set composition. Actually, FIT is a purely topological error label in nature. This justifies why advanced geometric features which take into account the structure of the building models, are better in prediction. The same could be said about FOS, which explains why both these labels benefited from the usage of graph kernels. However, it was not the case of BIT which is contradictory with the previous explanation.

The second factor that should be taken into account is the frequency of the label. The more the latter is large, the larger the impact of graph kernels. In fact, this, added to the statistics shown in Figure 7.1, can explain the change in prediction scores for each error:

- BUS was better on **Elancourt**;
- BIB was better on **Na-P13**;
- BIT was worse on **Elancourt**;
- FOS was better on **Elancourt**;
- FUS was worse on **Elancourt**;
- FIB was better on **Na-P13**;
- FIT was as better on **Elancourt** as on **Na-P13**;
- FIG was as worse on **Elancourt**.

Based on these comparisons, and the Table 7.10, we can try to anticipate the changes that will occur when using both graph kernels and ScatNet features. In fact, baseline geometric features were the best configuration for some error labels. Improving on these with graph kernels, we can only expect better scores for `BIB` and `FIT` on **Na-P13**. However, on the rest, we cannot predict how these features will interact with the other modalities.

7.3.3 Graph kernels and ScatNet to baseline comparison

Now that we have studied the contribution of each of the proposed advanced features, these are combined in order to assess their impact on the prediction results. As in the previous subsection, using graph kernels forces us to abandon RFs and to experiment only with SVMs. Results are reported in Table 7.13.

In figure 7.10, we can see how standard deviations are very low[42] for all errors and on both sets. The only exception is when image based features with ScatNet are added on **Na-P13** and the F-score imporves by almost 4.5 % for `BIB`. This is consistent with the design of the image based features as well as the previous experimental results.

Since graph kernels did not prove to always be better than the baseline features, this situation could not be explained by the fact that geometric features with graph kernels were sufficient enough, as in Section 5.2. Indeed, `FIB` proved to be stable when changing the baseline features for the graph kernels. However, when the latter was put together with ScatNet features it yielded a worse F-score compared the **Geom.** $\oplus$ **S(d)-Im.** configuration.

Table 7.14 compiles comparisons, for both options, sets and all labels, between graph kernel and baseline geometric features when used with ScatNet features. We can see how, no matter the chosen option, the set nor the label, follow tightly the evolution of **K-Geom.**.

There are two reasons that can explain this situation. First, as seen before, the SVM classifier was not well parameterized for the ScatNet derived features. Most importantly, the second explanation involves the high number of kernel used for geometric features compared to one for each extrinsic feature. In fact, when adding up the importance ratio of all graph kernels, the intrinsic feature ratio was always over 80 %. This explains very well how geometric features faded out the contributions of the other modalities.

In Figure 7.11, we illustrated the normalized (in the number of graph kernels) MKL weights. There is no notable difference between the two options. Once again, as with ScatNet features and baseline geometric features, the height based modality is not as important as the others.

Based on this study, we can update Table 7.10 based on the recent results. The graph kernel based feature combinations were helpful twice. Both cases where on **Na-P13** as shown in Table 7.15.

[42] It is under 1 %, in most cases, and 3 %, in the worst case.

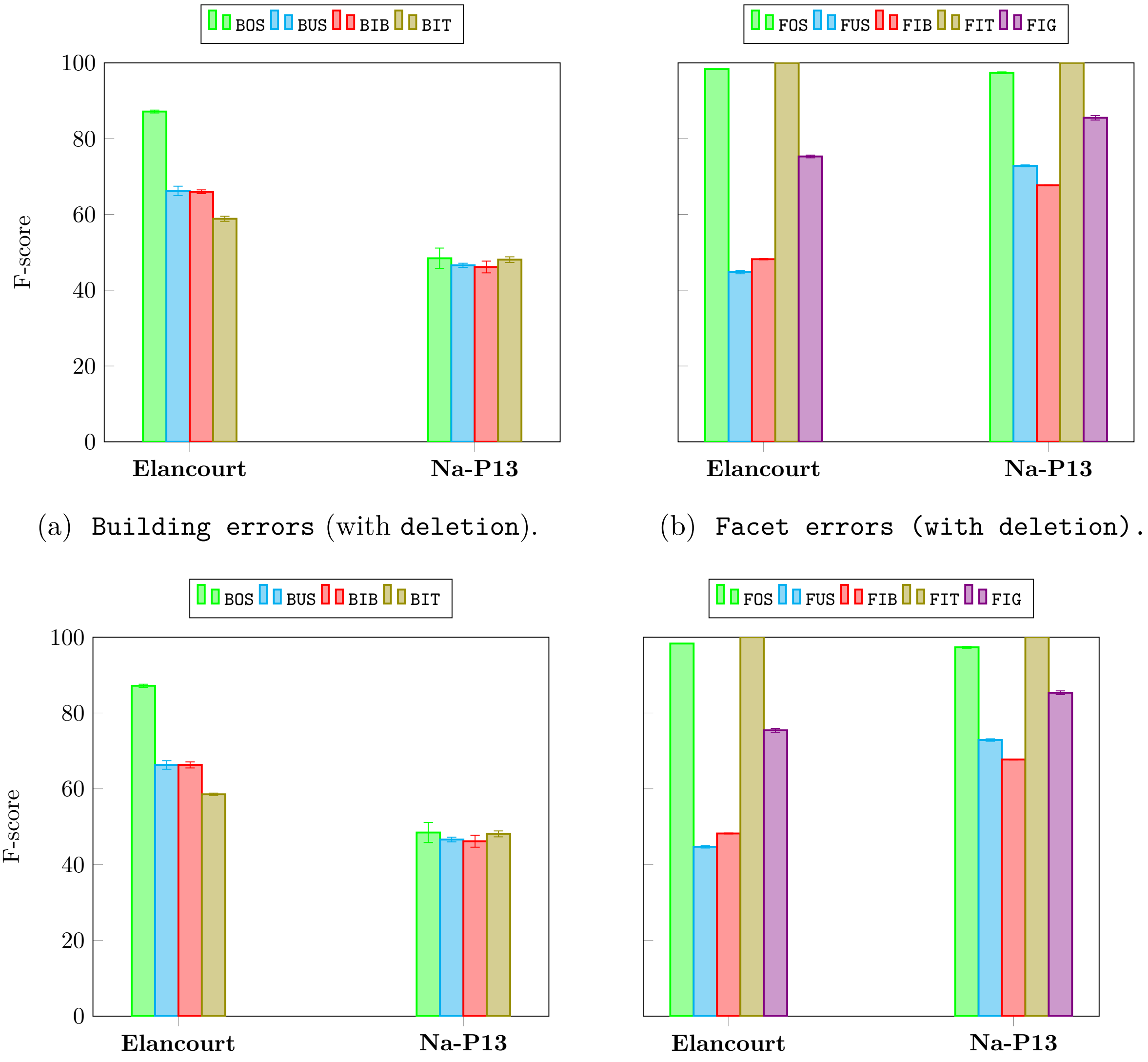
Elancourt

	K-Geom.		**K-Geom. ⊕ S-Hei.**		**K-Geom. ⊕ S(d)-Im.**		**K-S(d)-All**		**K-Geom. ⊕ S(c)-Im.**		**K-S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	87.99	86.11	87.16	86.19	88.29	86.85	87.84	86.80	**88.66**	**86.58**	88.30	86.54
`BUS`	92.99	51.29	**93.42**	**53.27**	93.84	49.50	93.86	50.92	93.63	49.94	93.63	51.04
`BIB`	72.28	60.08	73.40	59.36	70.94	61.80	72.77	61.51	72.27	61.86	**73.76**	**61.83**
`BIT`	96.94	41.67	96.94	41.67	**97.96**	**42.86**	96.94	42.60	96.94	42.22	95.96	42.41
`FOS`	**97.12**	**99.60**	**97.12**	**99.60**	**97.12**	**99.60**	**97.12**	**99.60**	97.05	99.60	97.05	99.60
`FUS`	84.39	30.18	84.39	30.08	**85.03**	**30.83**	84.71	30.75	84.71	30.61	84.71	30.61
`FIB`	94.74	32.34	94.32	32.24	**96.49**	**32.31**	96.05	32.06	96.49	32.26	96.51	32.12
`FIT`	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
`FIG`	69.66	80.91	69.92	81.28	70.93	80.64	71.10	80.91	71.19	80.92	**71.44**	**81.06**

Na-P13

	K-Geom.		**K-Geom. ⊕ S-Hei.**		**K-Geom. ⊕ S(d)-Im.**		**K-S(d)-All**		**K-Geom. ⊕ S(c)-Im.**		**K-S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	**43.42**	**61.34**	43.42	60.81	38.89	56.59	38.89	56.59	38.89	56.93	38.89	56.59
`BUS`	86.15	31.73	85.38	31.36	87.69	32.11	87.69	32.11	**87.02**	**32.39**	87.69	32.11
`BIB`	75.16	31.94	75.16	31.86	79.74	33.80	79.74	33.80	**79.74**	**33.89**	79.74	33.80
`BIT`	87.23	32.67	87.23	32.48	89.89	33.40	89.89	33.40	**89.89**	**33.53**	89.89	33.40
`FOS`	95.45	98.86	**96.59**	**98.86**	95.86	98.86	95.86	98.86	95.59	98.86	95.86	98.86
`FUS`	79.34	67.13	79.55	66.61	80.79	66.61	80.79	66.61	**80.79**	**66.95**	80.79	66.61
`FIB`	**92.16**	**53.71**	92.16	53.51	92.81	53.18	92.81	53.18	92.51	53.48	92.81	53.18
`FIT`	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
`FIG`	80.73	89.40	81.33	89.35	**83.03**	**89.19**	**83.03**	**89.19**	81.94	89.18	**83.03**	**89.19**

Table 7.13: SVM results using graph kernels and ScatNets, expressed in percentage, on the two datasets at **eFin** level 3.



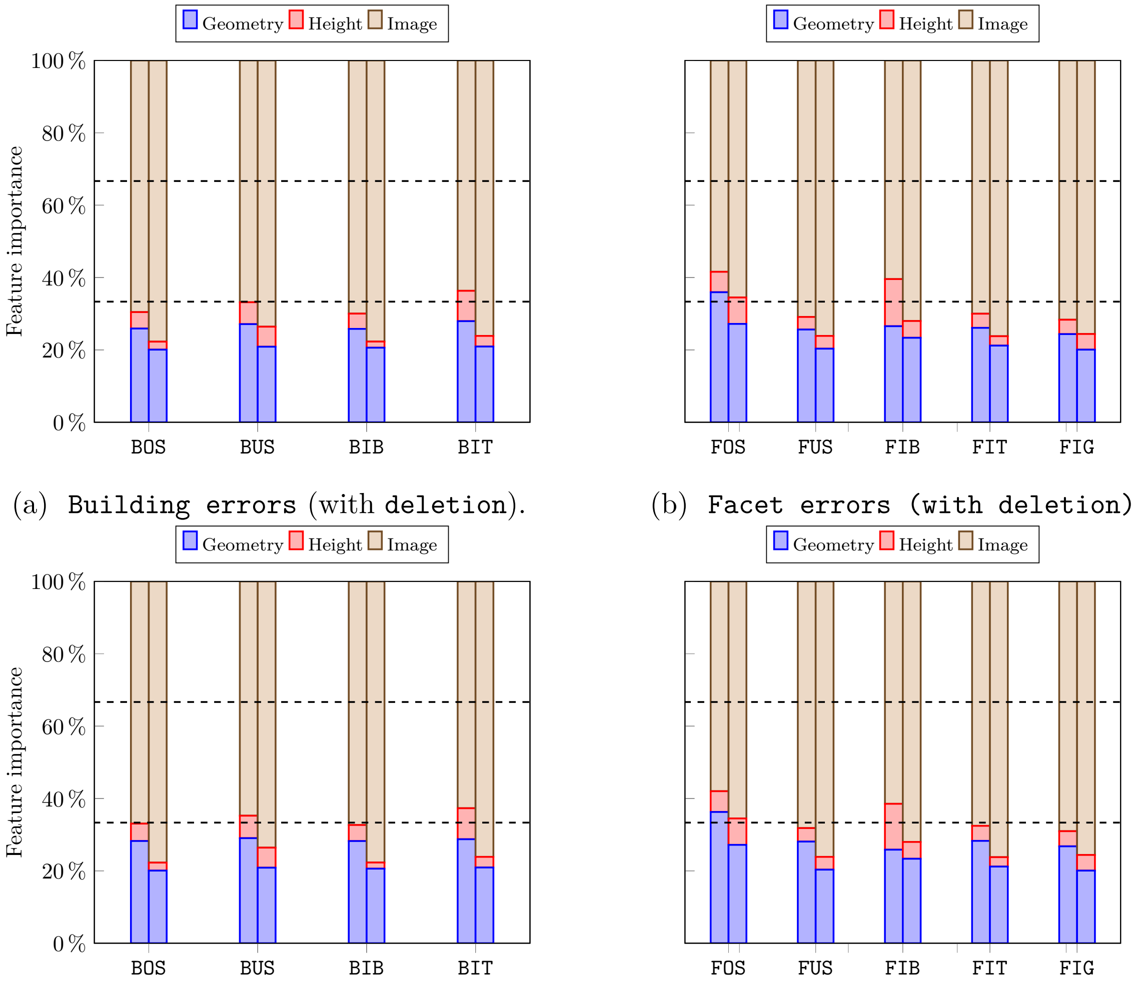
(a) `Building errors` (with `deletion`).

(b) `Facet errors (with deletion).`

(c) `Building errors (with channel).`

(d) `Facet errors (with channel).`

Figure 7.10: Mean F-score and standard deviation obtained with an SVM. The geometric modality is based on graph kernels while height and image based features use the ScatNet with `deletion` and `channel` options.



(a) `Building errors` (with `deletion`).

(b) `Facet errors (with deletion).`

(c) `Building errors (with channel).`

(d) `Facet errors (with channel).`

Figure 7.11: Normalized modality importance for the SVM classifier based on the coefficients computed by EasyMKL. The geometric modality uses on graph kernels while height and image based features use the ScatNet with `deletion` and `channel` options. The first (*resp.* second) column represents **Elancourt** (*resp.* **Na-P13**).

	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt		🟨	🟧	🟫		🟥	🟫		🟨
Na-P13			🟩 **S(d)-Im.**		🟩		🟩		

(a) Comparison with `deletion` option.

	BOS	BUS	BIB	BIT	FOS	FUS	FIB	FIT	FIG
Elancourt		🟨	🟧	🟫	🟩	🟥	🟫		🟨
Na-P13			🟩 **S(c)-Im.**		🟩		🟩		

(b) Comparison with `channel` option.

Table 7.14: Evolution of the F-score value, for each error using on graph kernels and ScatNet compared to when ScatNet was used with geometric baseline features. Feature sets having a significant impact on the classification results are mentioned in the corresponding cell. The used color scheme is presented in figure 5.13.

		Elancourt	
	$\boldsymbol{F_{score}}$	Feature configuration	Classifier
BOS	91.71	**Geom.**	SVM
BUS	72.75	**Geom.** $\oplus$ **S(d)-Im.**	SVM
BIB	90.91	**Geom.** $\oplus$ **S(c)-Im.**	SVM
BIT	100	**Geom.** $\oplus$ **Hei.** **Geom.** $\oplus$ **Im.** **All** **Geom.** $\oplus$ **S(c)-Im.**	SVM
FOS	99.46	**S(c)-All**	RF
FUS	75.41	**Geom.** $\oplus$ **S(d)-Im.**	SVM
FIB	85.12	**Geom.** $\oplus$ **S(d)-Im.**	SVM
FIT	100	**Geom.** $\oplus$ **Hei.** **Geom.** $\oplus$ **Im.** **All** **Geom.** $\oplus$ **S(d)-Im.** **S(d)-All** **Geom.** $\oplus$ **S(c)-Im.** **S(c)-All** **K-Geom.** **K-Geom.** $\oplus$ **S-Hei.** **K-Geom.** $\oplus$ **S(d)-Im.** **K-S(d)-All** **K-Geom.** $\oplus$ **S(c)-Im.** **K-S(c)-All**	SVM
FIG	86.48	**Geom.**	SVM

		Na-P13	
	$\boldsymbol{F_{score}}$	Feature configuration	Classifier
BOS	62.44	**Geom.**	RF
BUS	52.46	**Geom.** $\oplus$ **Im.**	RF
BIB	47.56	**K-Geom.** $\oplus$ **S(c)-Im.**	SVM
BIT	48.84	**K-Geom.** $\oplus$ **S(c)-Im.**	SVM
FOS	98.69	**All.**	RF
FUS	74.61	**Geom.**	SVM
FIB	76.58	**S(c)-All.**	RF
FIT	100	**Geom.** $\oplus$ **Hei.** **Geom.** $\oplus$ **Im.** **All** **Geom.** $\oplus$ **S(d)-Im.** **S(d)-All** **Geom.** $\oplus$ **S(c)-Im.** **S(c)-All** **K-Geom.** **K-Geom.** $\oplus$ **S-Hei.** **K-Geom.** $\oplus$ **S(d)-Im.** **K-S(d)-All** **K-Geom.** $\oplus$ **S(c)-Im.** **K-S(c)-All**	SVM
FIG	90.99	**S(c)-All**	RF

Table 7.15: For each set and each error label, we report the best F-score as well as feature configurations and classifiers.

7.3.4 Summary

In this section, we experimentally examined the added value of the advanced features that we proposed: We have seen how:

- For both sets, ScatNet proved to be more helpful than baseline features in predicting errors;
- In contrast to the baseline version, height based features with ScatNet proved to be crucial for detecting error labels;
- The `channel` option was best when used with an RF, while `deletion` worked best with the SVM classifier;
- Using ScatNet derived features, SVM proved to be better overall, especially on **Elancourt**, than using an RF;
- Once again, the SVM proved to be not well parameterized to take full advantage of the extrinsic features;
- Graph kernels, compared to baseline features, improve prediction results of labels provided they are frequent enough;
- Although being helpful, height based features have, by a large margin, the least importance ratio compared to the other modalities for both classifiers;
- Graph kernels, outnumbering the kernels for extrinsic modalities, have reduced the potency of the ScatNet derived features;
- **Elancourt** the easiest to learn on with a minimum F-score of 72 % and 85 % on 7 out of 9 of the error labels;
- `Building errors` proved again to yield worse scores on **Na-P13**.

8 Conclusion

Contents

In this chapter, we conclude the work presented in this thesis. First, we provide a summary of the porposed approach and the related experimental results in Section 8.1. Secondly, Section 8.2 addresses some issues that remain open to investigation.

8.1 Summary

A learning framework was proposed to semantically evaluate the quality of 3D models of buildings. For that purpose, based on our observation over three different urban areas, a general categorization of errors is drawn in a hierarchical and modular manner. It aims to handle the large diversity of urban environments and varying requirements stemming from end-users (geometric accuracy and level of details). Based on the desired evaluation Level of Detail, exclusivity and evaluation Finesse, an error collection is considered. Model quality is then predicted using either a supervised Random Forest or Support Vector Machine classifier. Each model provides intrinsic geometrical characteristics that are compiled in a handcrafted feature vector. Remote sensing modalities can be introduced. This helps better describing building models and detecting errors. Attributes can indeed be extracted by comparing the 3D model with optical images or depth data at the spatial resolution at least similar to the input 3D model. Experiments shows it helps detecting hard cases both for geometrical and topological errors with the right choice of classifiers. In addition to these baseline features, we also made use of graph kernels in order to better take into account the geometric structure of the evaluated model. We also utilized the Scattering Network so as to better extract information from the extrinsic modalities.

This new framework was applied to the case of aerial urban reconstruction, where features are extracted from Very High Resolution airborne images and a Digital Surface Model. A fully annotated dataset containing 3,235 aerial reconstructed building models with high diversity and from three distinct areas was used to assess our method. External remote sensing data consisted in multimodal RGB optical and Digital Surface Model features. Although being mitigated over under-represented errors, results are satisfactory in the well balanced cases. Moreover, with the right choice of the classifier as well as the feature configuration, we can achieve better results than those obtained with the baseline features. More importantly, we proved that the urban scene composition affects greatly error detection. In fact, some predictions scores are not only stable, when training on a different urban scene, they even outperform when learning on the same scene. Additionally, we reported how, for a heterogeneous training dataset, the size of the training set have, practically, no effect as test score stay stable for all errors. This demonstrates that the proposed framework can be easily scaled with the right choice of training samples with little manually generated data. This exactly answers to the raised issue, contrarily to the present state-of-the-art literature. We believe our framework is robust enough to evaluate unseen areas. It represents also a strong basis for subsequent manual or automatic 3D building model correction.

8.2 Perspectives

The goal of this work was to introduce an new paradigm of large scale and automated semantic evaluation of building models which was largely understudied in the community. Thanks to a thorough experimental analysis, we have proven the feasibility of our approach which is based on the supervised formulation of the error detection as well as the error taxonomy definition. Yet some questions went unanswered. We present hereafter some issues that should be adressed further.

The proposed taxonomy was mainly developed based on observations in the overhead case. **Terrestrial data** has become lately more ubiquituous. As a consequence, the number of façade modeling methods have proliferated accordingly. The relevance of the proposed errors was briefly discussed for this case in Section 3.2 but is still to be proven experimentally.

This error taxonomy, although providing valuable information on the type of defects that can affect the building model, is not very thorough. It was indeed developed so as to guide human operators correct models through a concise and meaningful description of the failures altering said models. However, our approach does not **quantify the detected errors** to further help the model correction process. For instance, if we imagine a model with a planar facet with an imprecisely estimated slope. This corresponds to a `Facet Imprecise Geometry (FIG)` error, but the operator would have no idea by how much the facet was tilted. As a consequence, a possible extension of this work is to associate appropriate metrics to each `atomic` error which would provide actionable information on the detected error. Such metrics have already been studied profusely in the literature as shown in Section 2.2.

Always related to building model correction, error existence was predicted at building level even for **facet level** defects. This is actually not an ideal setting for a postprocessing step as errors are not localized. This could be solved by relying on the facet graph structure of the models (cf. Section 4.2.1) and labeling individual facets.

In order to scale the whole building modeling pipeline and linked to automatic quality evaluation, the question of model **correction automation** steps into light. Indeed, one could imagine how an automatic evaluation of a building model yielding semantic localized errors along with specific metrics can be useful for automated model correction. In fact, to each error could be associated a set of unitary actions based on the discussion in Section 3.2. Hence, the correction step could be designed as an optimization problem which would determine the best combination of actions that can solve the raised issues while making use of the corresponding error metrics as a prior. This would be an iterative process depending on a preestablished prioritization of errors.

In the experimental study, all models originated from one modeling approach. The scalability was only tested in terms of the considered urban area. Actually, it would be interesting to test the transferability of learned classifiers depending on the **model origin** especially as intrinsic features play an important role in error prediction.

Connected to the learning issues, the proposed advanced features had a positive but limited impact on the predictability of errors. **Deep learning** proved in the last decade to be the state-of-the-art approach in Computer Vision tasks. However, it requires a large number of training instances. Manual annotation necessitates a considerable effort especially for expert tasks as ours. To alleviate this issue there are two possibilities. First, we can develop an **active learning** approach that can help scale the annotation effort. Secondly, we can **simulate errors** by altering ground truth models like the ones presented in (Rottensteiner et al., 2012).

A List of publications

Contents

Provided herein are the articles published during this thesis. They are classified by publication type.

A.1 Journal articles

Ennafii, **Oussama**, Arnaud Le Bris, Florent Lafarge, and Clément Mallet (Dec. 2019a). "A learning approach to evaluate the quality of 3D city models." In: *Photogrammetric Engineering and Remote Sensing.*

A.2 Peer-reviewed conference papers

Ennafii, **Oussama**, Arnaud Le Bris, Florent Lafarge, and Clément Mallet (Aug. 2019b). "Scalable evaluation of 3D city models." In: *IEEE Geoscience and Remote Sensing Society (IGARSS).* IEEE.

Ennafii, **Oussama**, Arnaud Le Bris, Florent Lafarge, and Clément Mallet (May 2019c). "The necessary yet complex evaluation of 3D city models : a semantic approach." In: *Joint Urban Remote Sensing Event (JURSE).*

A.3 Peer-reviewed conference abstracts

Ennafii, **Oussama**, Arnaud Le Bris, Florent Lafarge, and Clément Mallet (June 2018d). "Qualification sémantique de modèles 3D de bâtiments." In: *Conférence Française de Photogrammétrie et de Télédétection (CFPT).* SFPT.

A.4 Technical reports

Ennafii, **Oussama**, Arnaud Le-Bris, Florent Lafarge, and Clément Mallet (Sept. 2018b). *Semantic evaluation of 3D city models.* Tech. rep.

B

Supervised classifiers

Contents

In this appendix, are presented notions from statistical learning were useful in this work. This constitutes a simple reminder and is not a thorough survey of the field.

First, in Section B.1, we present the notations used in this chapter. Next, Section B.2 presents details about the inner workings of the SVM algorithm. Third and last, RFs are presented in Section B.3.

B.1 Notations

In this subsection, we denote a family of n observations and their classes by

$$\left(O^i, y^i\right)_{i=1,2,\dots,n}.$$

One way of representing an observation O is to compute a number $d \in \mathbb{N}^*$ of features (or attributes):

$$\forall j = 1, 2, \dots, d\ , x_j \in \mathscr{X}_j$$

These coefficients are aggregated to form a feature vector

$$\boldsymbol{x} \triangleq (x_1, x_2, \dots, x_d) \in \prod_{i=1}^{d} \mathscr{X}_j$$

that corresponds to the observation O. Such a mapping $O \mapsto \boldsymbol{x}$ is called a feature map. A special case is when $\forall j = 1, 2, \dots, d\ , \mathscr{X}_j = \mathbb{R}$ and feature vectors are Euclidean ones.

$$\forall i = 1, 2, \dots, n\ , \boldsymbol{x^i} \in \mathbb{R}^d.$$

In practice, observations are often confused with fetaure vectors.

Classes are modeled by a finite set bijective to the set $\{1, 2, \dots, C\}$:

$$\forall i = 1, 2, \dots, n\ , y_i \in \{1, 2, \dots, C\}\,.$$

The goal in supervised classification is to learn, based on the training set $\left(\boldsymbol{x^i}, y^i\right)_{i=1,2,\dots,n}$, some statistical characteristics in order to predict the classes of some new observations:[43]

$$\left(\boldsymbol{x^i}\right)_{i=n+1,n+2,\dots,n+n'}.$$

This type of problems is said to be multi-class: each instance has one possible class out of a number of possibilities. In the special case of binary classification, only two classes are possible:

$$\forall i = 1, 2, \dots, n\ , y^i \in \{0, 1\}$$

On another hand, the multi-label classification problem is the case where to each instance corresponds a number L of binary labels:

$$\forall i = 1, 2, \dots, n\ , y^i = (y^i_1, y^i_2, \dots, y^i_L) \in \{0, 1\}^L\,.$$

In this case, associating features with each label $\left(\boldsymbol{x^i}, y^i_l\right)_{i=1,2,\dots,n}$ can be seen as an independent binary classification problem. This is different from multi-class problems, as for the latter only one class is possible while in the other multiple labels can be detected per instance.

Hereafter, are presented the two supervised classifiers that were used in this work.

[43] This family is usually called the test set.

B.2 Support Vector Machine

SVMs are a special type of linear classifiers. The initial classes $\left(y^k\right)_{k=1,2,\ldots,n}$ are transformed into $\left(\tilde{y}^k\right)_{k=1,2,\ldots,n}$ using the map: $y \mapsto \tilde{y} = 2 \cdot y - 1$. There are always two possible classes $\{1, -1\}$. This simple transformation is use so as to simplify the latter equations.

We remind the reader of the decision function[44] of a linear classifier:

$$\begin{aligned} \mathbf{D}_{\text{linear},\boldsymbol{w},b} : \mathbb{R}^d &\to \{-1, 1\} \\ \boldsymbol{x} &\mapsto 2 \cdot \mathbb{1}_{\boldsymbol{w}^\intercal \cdot \boldsymbol{x} + b \geq 0} - 1 \quad . \end{aligned} \tag{B.1}$$

where:

$\boldsymbol{w}$: is the weight vector;
b : is the bias.

B.2.1 Hard margin

In order to understand the idea behind the SVM classifier, we start by assuming that the dataset to be classified $\left((\boldsymbol{x}^k, y^k)\right)_{k=1,2,\ldots,n}$ is linearly separable. It means that there is at least one hyperplane (H_0) that can separate perfectly the two classes. We can order points of one class based on their distance to the hyperplane (H_0): $\boldsymbol{x} \mapsto d(\boldsymbol{x}, H_0)$. The closest positive (*resp.* negative) points (i.e., of class 1 (*resp.* -1)) to (H_0) are called positive (*resp.* negative) support vectors. Support hyperplanes are the hyperplanes that are parallel to the separator and pass through the support vectors. This can be summarized as:

$$\left\{\boldsymbol{x}_s^+ : s = 1, 2, \ldots, n_{psp}\right\} \triangleq \arg\min \left\{d(\boldsymbol{x}_k, H_0) : k = 1, 2, \ldots, n \wedge \tilde{y}^k = 1\right\} \tag{B.2}$$

$$\left\{\boldsymbol{x}_s^- : s = 1, 2, \ldots, n_{nsp}\right\} \triangleq \arg\min \left\{d(\boldsymbol{x}_k, H_0) : k = 1, 2, \ldots, n \wedge \tilde{y}^k = -1\right\} \tag{B.3}$$

where:

n_{psp} : is the number of positive support vectors.
n_{nsp} : is the number of negative support vectors.

We define:

$$Sol \triangleq \left\{\omega \in \mathbb{R}^d : \omega.\boldsymbol{x} + b = 0\right\}.$$

We verify that:

$$\forall \lambda \in \mathbb{R}^*,\ (\boldsymbol{w}, b) \in Sol \Leftrightarrow (\lambda.\boldsymbol{w}, \lambda.b) \in Sol.$$

In other words, the solution $(\boldsymbol{w}^*, b^*)$ is unique up to multiplicative term $\lambda \in \mathbb{R}^*$. In this context, $(\boldsymbol{w}, b)$ is chosen so that $\boldsymbol{w}$ points to the positive instances[45] and the support hyperplanes verify:

$$\begin{cases} \forall s = 1, 2, \ldots, n_{psp} & \boldsymbol{w}^\intercal \cdot \boldsymbol{x}_s^+ + b = 1 \\ \forall s = 1, 2, \ldots, n_{nsp} & \boldsymbol{w}^\intercal \cdot \boldsymbol{x}_s^- + b = -1 \end{cases} \tag{B.4}$$

Figure B.1 illustrates these notions. The uncertainty of a point classification is an increasing function of the distance of that point to the separator. In the hard margin case, the given samples are supposed to be certain. Since support vectors are the closest certain points to (H_0), the class of points $\boldsymbol{p}$ that verify $|\boldsymbol{w}^\intercal \cdot \boldsymbol{p} + b| \geq 1$ is known with

[44] A decision function assigns a class to given observation.

[45] The positive instances are inside the positive half-space.

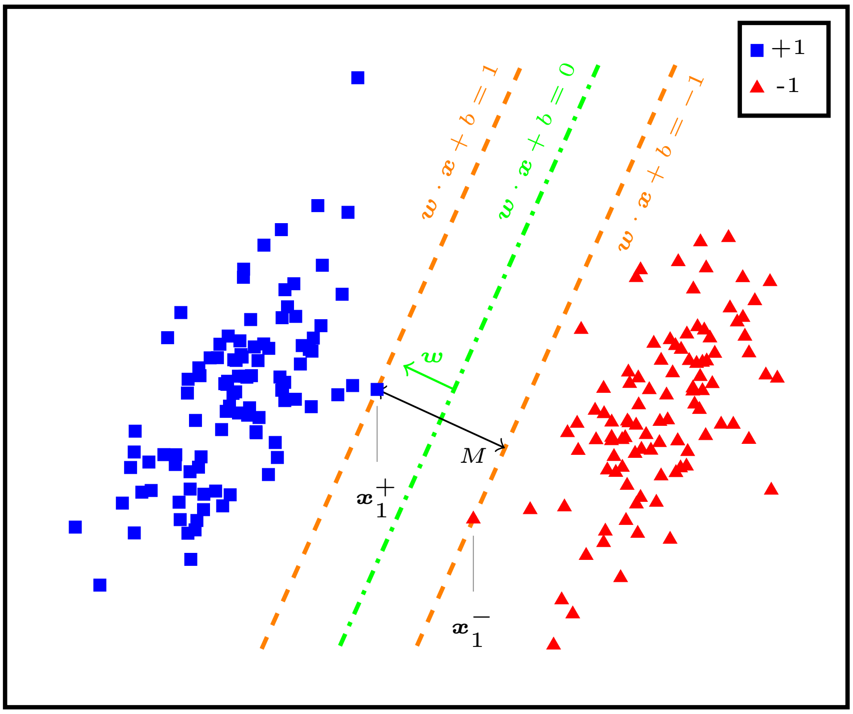


Figure B.1: Illustration of the hard-margin SVM. A hyperplane separator in $\mathbb{R}^2$ corresponds to a line. In this case, $n_{psp} = n_{nsp} = 1$. The support lines are plotted in orange. M is the margin.

probability 1. Conversely, the points that lie between the support hyperplanes are the most uncertain. The bigger the distance between these two lines, the more nuanced the classifier decision is. The main idea behind SVMs is to maximize this distance. It is called margin and denoted by M.

In order to compute the margin, we can interpret it as the length of the orthogonal projection of any vector $\boldsymbol{v}_{st}$[46] going from a negative support vector $\boldsymbol{x}_s^-$ to a positive one $\boldsymbol{x}_t^+$ on any line carried by $\boldsymbol{w}$. Since all support vectors of the same class lay on the same line, the choice of (s,t) is inconsequential. We can hence deduce:

$$\begin{aligned} M &= \frac{\boldsymbol{w}^\intercal}{\|\boldsymbol{w}\|} \cdot (\boldsymbol{x}_1^+ - \boldsymbol{x}_1^-) \\ M &= \frac{2}{\|\boldsymbol{w}\|} \end{aligned} \tag{B.5}$$

Maximizing M is actually equivalent to minimizing $\|\boldsymbol{w}\|$. In the purpose of simplifying the resolution of the problem, we drop the square root and minimize instead $\|\boldsymbol{w}\|^2$. The certainty in the given samples is translated by the inequalities:

$$\begin{cases} \boldsymbol{w}^\intercal \cdot \boldsymbol{x}^i + b \geq 1 & \forall i \in \{1,2,\ldots,n : \tilde{y}^i = 1\} \\ \boldsymbol{w}^\intercal \cdot \boldsymbol{x}^i + b \leq -1 & \forall i \in \{1,2,\ldots,n : \tilde{y}^i = -1\} \end{cases}.$$

These can be summed in one, as follows:

$$\tilde{y}^i.(\boldsymbol{w}^\intercal \cdot \boldsymbol{x}^i + b) \geq 1 \; \forall i = 1,2,\ldots,n. \tag{B.6}$$

Thus, the SVM problem can be formalized as a convex constrained quadratic optimization one:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \|\boldsymbol{w}\|^2 \\ \text{s.t.} \quad & \tilde{y}^i.(\boldsymbol{w}^\intercal \cdot \boldsymbol{x}^i + b) \geq 1 \; \forall i = 1,2,\ldots,n \end{aligned}. \tag{B.7}$$

[46] $\forall (s,t) \in \{1,2,\ldots,n_{nsp}\} \times \{1,2,\ldots,n_{psp}\}$.

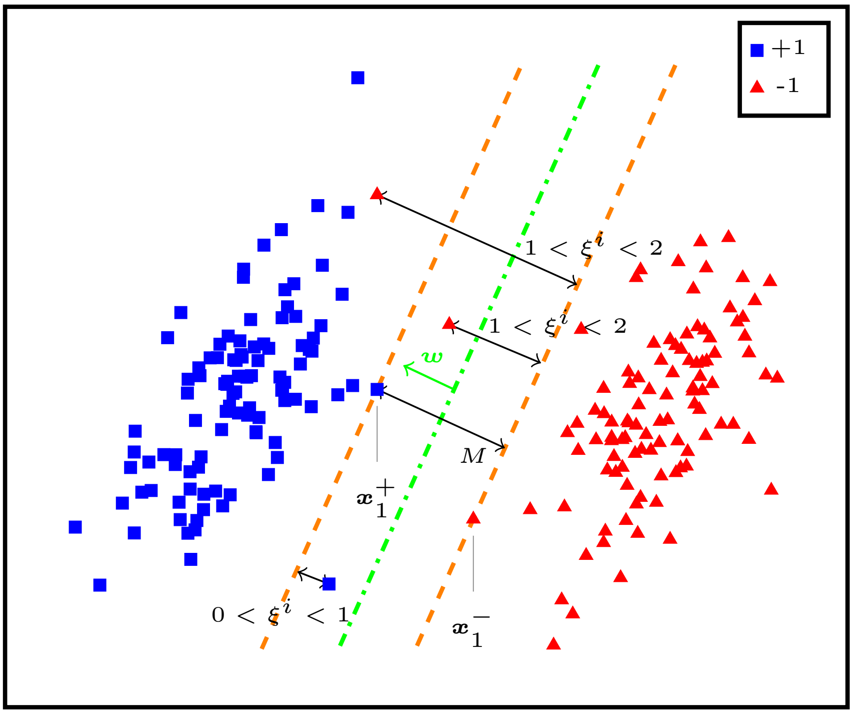


Figure B.2: Illustration of a soft margin SVM. The orange lines (*resp.* green line) correspond to the support hyperplanes (*resp.* separator). Dataset points are allowed to be inside the margin and even in the other side of the separator. It helps fit a natively linear classifier in a non linearly separable case.

B.2.2 Soft margin

In the previous case, we assumed that we are always certain about the classes of the given samples. The margin cannot contain any of those points. This explains the name hard margin. We are always guaranteed to have a solution of a hard margin SVM in the case of linear separability. However, if the last condition is not met, there is no such solution. To alleviate this problem, Cortes et al. (1995) allowed some uncertainty in the given samples in order to fit the linear model.

To put this into mathematical terms, we start by the reminding the constraint of the hard margin problem in Equation B.6. Allowing uncertainty for some given point $\boldsymbol{x}^i$ comes back to allowing it to be in the wrong side of the support hyperplane: i.e., $\tilde{y}^l.(\boldsymbol{w}^\intercal \cdot \boldsymbol{x}^l + b) < 1$. In this case, we define: $\xi^i := 1 - \tilde{y}^i.(\boldsymbol{w}^\intercal \cdot \boldsymbol{x}^i + b) > 0$. Otherwise, $\xi^i := 0$. These defined values are called slack variables. They express how much each point is uncertain. They can be expressed in a compressed form called hinge loss:

$$\xi^i \triangleq \max\left(1 - \tilde{y}^i.(\boldsymbol{w}^\intercal \cdot \boldsymbol{x}^i + b), 0\right). \tag{B.8}$$

The soft margin constraint translates now to:

$$\tilde{y}^i.(\boldsymbol{w}^\intercal \cdot \boldsymbol{x}^i + b) \geq 1 - \xi^i. \tag{B.9}$$

Four cases can be distinguished for slack variables:

- $\xi^i = 0$: the class of the point is certain;
- $0 < \xi^i \leq 1$: the point is of the same class but is uncertain: i.e., between the support hyperplane and the separator;
- $1 < \xi^i < 2$: the point is of the opposite class and is uncertain: i.e., between the separator and the opposite support hyperplane;

- ► $2 \leq \xi^i$: the point is certainly of the opposite class: i.e., beyond the opposite support hyperplane.

As a consequence, when a dataset is not linearly separable, at least one sample i_0 cannot fit in a linear model and $1 < \xi^{i_0}$. All these situations are illustrated in Figure B.2.

The idea is to find a configuration where most slack variables are null or near 0. Sparsity is also required: we prefer having one wrong point rather than a lot of points that are uncertain. For this purpose, these variables would be penalized against using an L_1 norm. Since all slack variables are positive, the soft margin SVM problem becomes:

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^d} \quad & \|\boldsymbol{w}\|^2 + C \cdot \sum_{i=1}^{n} \xi^i \\ \text{s.t.} \quad & \tilde{y}^i.(\boldsymbol{w}^\intercal \cdot \boldsymbol{x}^i + b) \geq 1 - \xi^i, \ \forall i = 1, 2, \ldots, n^{.} \\ & \xi^i \geq 0, \ \forall i = 1, 2, \ldots, n \end{aligned} \tag{B.10}$$

where:

C : is the penalization constant.

A small value of C means the slack variables are allowed to get big: the margin is expected to be big. A high value of C leads to a tight margin as it penalizes any uncertainty. The special case where $C = \infty$ corresponds simply to the hard margin case. Hence, the penalization constant is tuned, using cross-validation for instance, so as to achieve the best generalization power.

Since the problem stated in Equation B.10 is a convex optimization problem, solving it is equivalent to solving the dual:

$$\begin{aligned} \max_{\alpha_1,\alpha_2,\ldots,\alpha_n\in\mathbb{R}_+} \quad & \sum_{1\leq i\leq n} \alpha_i - \frac{1}{2} \cdot \sum_{\substack{1\leq l\leq n \\ 1\leq p\leq n}} \alpha_l \cdot \alpha_p \cdot \tilde{y}^l \cdot \tilde{y}^p \cdot (\boldsymbol{x}^l)^\intercal \cdot \boldsymbol{x}^p \\ \text{s.t.} \quad & \sum_{1\leq i\leq n} \tilde{y}^i \cdot \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \ \forall i = 1, 2, \ldots, n \end{aligned} \tag{B.11}$$

where:

α_i : is the Lagrange multiplier corresponding to i^{th} observation.

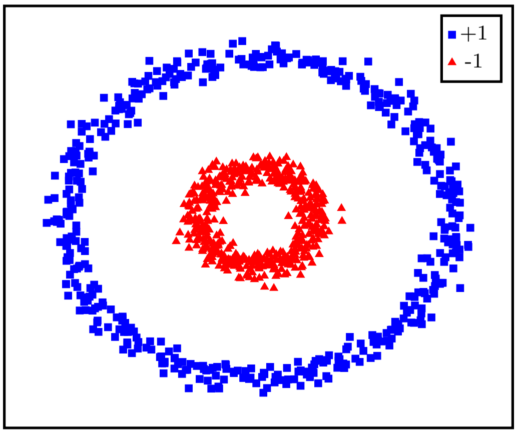
A fast solution of this problem is possible using the Sequential Minimal Optimization (SMO) developed by Platt (1998). This will not be detailed herein.

B.2.3 Kernel SVM

Not all classification problems can be solved in a linear manner. Unfortunately, SVM is inherently linear. However, there is a way to generalize this classifier.

The original feature space where the data is represented is not always an ideal[47] one. A solution is to find a transformation $\Phi : \mathbb{R}^d \rightarrow \mathscr{H}$ that maps initial feature vectors into a Hilbert space $(\mathscr{H}, \langle\cdot,\cdot\rangle_{\mathscr{H}})$ where distances between instances are meaningful. There is no unique map that satisfy this type of properties. We are particularly interested in a map

[47] An ideal space is one where distances between points are meaningful.



(a) Original representation of the dataset in the feature space.

Figure B.3: Example of a transformation of a dataset that can be linearly separable in the maped space.

which transforms the data into a space where they are hopefully linearly separable (Boser et al., 1992). This illustrated in Figure B.3. Using a polar coordinate transformation

$$\begin{aligned} \Phi : \mathbb{R}^2 &\to \mathbb{R}^2 \\ \begin{bmatrix} x \\ y \end{bmatrix} &\mapsto \begin{bmatrix} \sqrt{x^2+y^2} \\ \arctan_2(y,x) \end{bmatrix}, \end{aligned}$$

the dataset can be linearly separable. In this particular case, the original probability distribution is known as it was generated manually. Moreover, the map codomain is a vector space of the same dimension as the map domain. This is usually not the case: in practice, the right mapping is difficult to come by[48] and the codomain is usually of a higher dimension, possibly infinite. Finding such a map involves usually heavy computations. However, as we are interested in distances between dataset points, there is an easier way around. Using the "kernel trick", one can compute any distance in the target space $\mathscr{H}$ using only the scalar product:

$$\begin{aligned} k : \mathbb{R}^d \times \mathbb{R}^d &\to \mathbb{R} \\ (\boldsymbol{x}, \boldsymbol{y}) &\mapsto \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle_{\mathscr{H}}. \end{aligned} \tag{B.12}$$

Going back to the dual optimization problem stated in Equation B.11, we see that only the scalar product between samples is needed. Assuming the existence of an adequate mapping Φ, and if we substitute each point by its representation in the new space, we can rewrite, so called, kernel SVM.

$$\begin{aligned} \max_{\alpha_1,\alpha_2,\ldots,\alpha_n \in \mathbb{R}_+} \quad & \sum_{1\le i\le n} \alpha_i - \frac{1}{2} \cdot \sum_{\substack{1\le l\ge n \\ 1\le p\ge n}} \alpha_l \cdot \alpha_p \cdot \tilde{y}^l \cdot \tilde{y}^p \cdot k(\boldsymbol{x}^l, \boldsymbol{x}^p) \\ \text{s.t.} \quad & \sum_{1\le i\le n} \tilde{y}^i \cdot \alpha_i = 0 \\ & 0 \le \alpha_i \le C \ \forall i = 1, 2, \ldots, n \end{aligned} \tag{B.13}$$

There are more theoretical details for kernels which are not discussed herein. For more details on the subject, one can check the work by Aronszajn (1950), Shawe-Taylor et al. (2004), and Vapnik (2013).

[48] If it exists at all.

Usual kernels.

The choice of kernels is not easy. Practitioners experiment with different kernel types and parameters. One can engineer a kernel specific for their application. However, in general, there are some kernels that are frequently chosen:

- the native linear kernel:
$$k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\intercal \cdot \boldsymbol{y}; \tag{B.14}$$
- the RBF kernel: for some parameter $(\gamma) \in \mathbb{R}$
$$k_\gamma(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\gamma \cdot \|\boldsymbol{x} - \boldsymbol{y}\|^2\right); \tag{B.15}$$
- the polynomial kernel: for some parameters $(c, d) \in \mathbb{R}_+ \times \mathbb{N} \setminus \{0\}$
$$k_{\gamma,c}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\intercal \cdot \boldsymbol{y} + c)^d; \tag{B.16}$$
- the sigmoid kernel: for some parameters $(\gamma, c) \in \mathbb{R}^2$
$$k_{\gamma,c}(\boldsymbol{x}, \boldsymbol{y}) = \gamma \cdot \tanh(\boldsymbol{x}^\intercal \cdot \boldsymbol{y} + c). \tag{B.17}$$

The choice of kernel parameters in important not only for achieving good classification results but also to avoid overfitting problems.

Multiple Kernel Learning.

Some operations over kernels always yield valid kernels. This is true for instance for summation. In fact, fixing the number of kernels $K \in \mathbb{N} \setminus \{0\}$, for basis kernels $k_i\ \forall i = 1, 2, \ldots, K$ weigthed by $\mu_i \in \mathbb{R}_+\ \forall i = 1, 2, \ldots, K$,

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{K} \mu_i \cdot k_i(\boldsymbol{x}, \boldsymbol{y}) \tag{B.18}$$

is also a valid kernel. Such kernels proved to be better in practice than using single kernels (Lanckriet et al., 2004). To simplify the problem, the solution is chosen in the simplex of the basis kernels: i.e., $(\mu_1, \mu_2, \ldots, \mu_K) \in (\mathbb{R}_+)^K\ s.t. \sum_{i=1}^{K} \mu_i = 1$. Multiple schemes were presented in order to solve the MKL SVM problem (Rakotomamonjy et al., 2008; Varma et al., 2009; Sun et al., 2010).

B.2.4 Properties

Herein are presented properties of the SVM classifier based on a practical point of view. We start by stating some of its advantages:

- Mathematically guaranteed solution: the convex formulation of the problem implies the existence of a global optimum that can be computed efficiently.
- The SMO can be adapted to an online setting (Bordes et al., 2005).
- The soft margin SVM, with a right tuning of the slackness parameter C, can achieve a good generalization.
- The SVM is inherently sparse in the sense that it relies only on a small subset of the data which are support vectors. This implies that with a right kernel the SVM can handle imbalanced datasets.

On the other hand it suffers also from some drawbacks, listed hereafter:

- Finding a the right kernel to represent that data is not a simple task. Even with an expert knowledge, one can easily overfit to the training data.
- In the same sense, parameter fitting is not straight-forward and time-consuming. Practically, it is a result of a trial and error or grid search approach.
- Eventhough it can be adapted to multi-class or multi-output settings, the SVM is natively designed for binary classification.
- SVM does not handle well attributes of different types. Feature vectors are preprocessed, usually normalized, before training.
- SVM does not yield probabilities natively. Although it is possible to compute scores that are in the unit interval, these are difficult to interpret as probabilities.

B.3 Random Forest

A RF is an agregation of decision tree classifiers. First, is explained how the latter works. Afterwards, we describe how the aggregation of basic classifiers can be beneficial for classification, especially in the case of decision trees. We end this Section discussing some RF properties.

B.3.1 Decision tree classifier

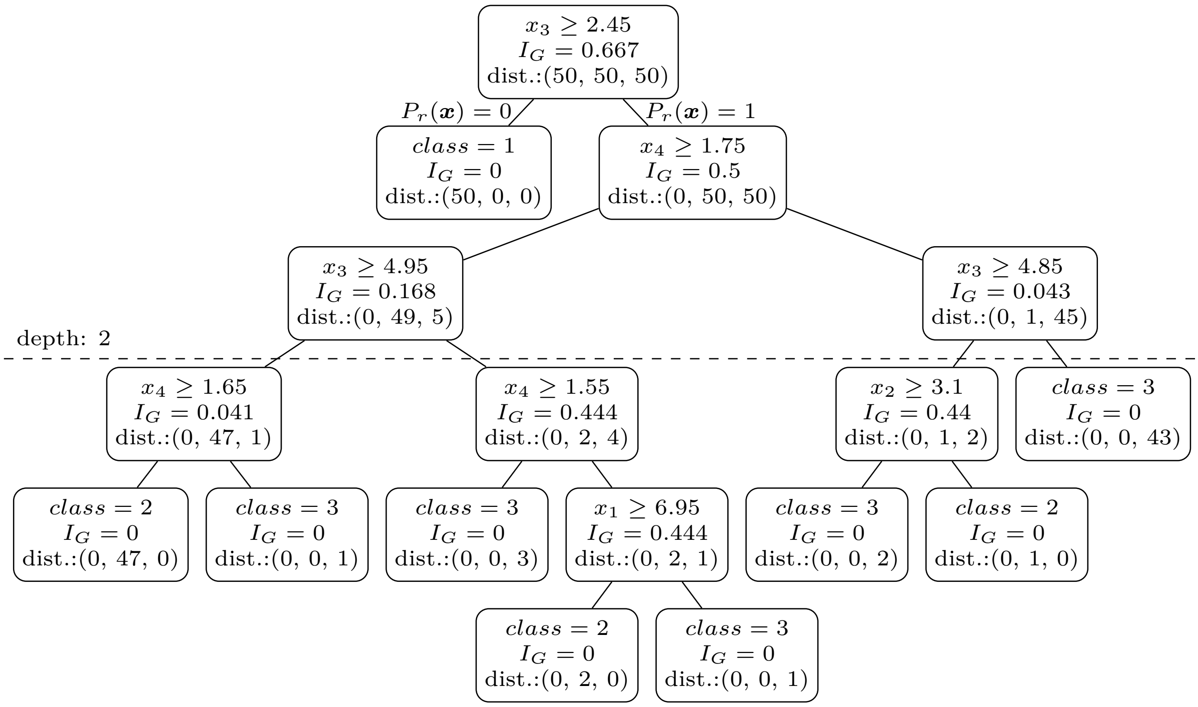


Figure B.4: A visualization of a decision tree classifier learned using I_G Gini index over the iris dataset (Fisher, 1936). Three classes are possible. Each non leaf node states a thresholding condition. For a leaf, its class is presented instead. “dist.”stands for distribution of classes in each node. The whole decision tree overfits to the data with a 100 % overall accuracy ratio. However, when cut at depth 2, the overall accuracy ratio does not drop noticeably (96 %).

The decision tree classifier consists in successive conditions. It is easily modeled as a tree graph rooted at r. Each non leaf node b in tree takes as input an observation $\boldsymbol{x}$ which is fed to logical predicate P_b[49]: exactly two children are possible per node. These predicates are of a specific form: only one dimension i_b of the data is taken into account. The output is then computed by applying a threshold t_b to x_{i_b}. This can be simplified as $P_b(\boldsymbol{x}) = \mathbb{1}_{x_{i_b} \geq t_b}$. The chosen child is denoted $d_b(\boldsymbol{x}) \leftarrow \text{child}(b, \mathbb{1}_{x_{i_b} \geq t_b})$. Leafs are the terminal nodes of the tree and they do not apply any predicate to their input. Instead, based on the assigned class $c(l) \in \{1, 2, \ldots, C\}$ of the leaf l, they predict the label of the input. Each dimension $x_i \in \mathscr{X}_i$ is treated separately. Hence, decision trees can handle heterogenous feature vectors that are no longer restricted to be $\mathbb{R}^d$. The decision function of a decision tree classifier can be written as:

$$\begin{aligned} D_{\text{decision tree}} : \prod_{i=1}^{d} \mathscr{X}_i &\to \{1, 2, \ldots, C\} \\ \boldsymbol{x} &\mapsto c\left(d_{d_{\ddots_{d_r(\boldsymbol{x})}}}(\boldsymbol{x})\right) \end{aligned} \tag{B.19}$$

This representation is complicated and too abstract and is never used. The intuitive representation of decision tree classifiers, shown in Figure B.4, is generaly prefered. In Figure B.5, we can clearly attribute the fact that the separator is an aggregation of vertical and horiontal segments to the fact that only one dimension is taken into acount at a time.

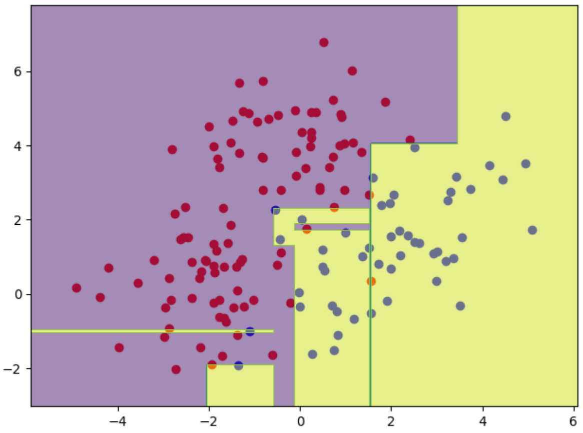


Figure B.5: Visualization of a decision tree separator trained over generated data in a two dimensional feature space. The separator is composed exclusively of horizontal and vertical lines. We can also see how the decision tree overfits by adding narrow splits to accomodate lone points surrounded by others with an opposite class.

Up to now, we assumed the decision tree to be already in place. The training step of this supervised classifier consists in determining its structure and all the thresholds. The goal is to have leafs with no prediction errors. In other words, all observations that end up in a leaf must be of the same class: the leaf is called pure. Consequently, to each node that is not pure would be associated a predicate that would hopefully distinguish amongst incoming observations. This is done recursively starting from the root.

[49] $P_b(\boldsymbol{x})$ can takle only two values.

The choice of the splitting dimension and the threshold at each node is made in the aim of achieving the most gain in the "purity" of child nodes. As a consequence, there is a need of a metric I that can describe the heterogeneity, as the opposite of purity, of a node. These are usually based on a probabilistic interpretation as they compute the fraction of observations p_c that go through a node of class $c \in \{1, 2, \ldots, C\}$. Three examples are provided:

Gini index:

$$I_G\left((p_c)_{c=1,2,\ldots,C}\right) = \sum_{c=1}^{C} p_c \cdot \sum_{\substack{l=1,2,\ldots,C \\ l \neq c}} p_l; \tag{B.20}$$

Entropy:

$$I_H\left((p_c)_{c=1,2,\ldots,C}\right) = -\sum_{c=1}^{C} p_c \cdot \log_2(p_c); \tag{B.21}$$

Variance:

$$I_V\left(\left(x^i_{d_s}\right)_{i \in S}\right) = \frac{1}{2 \cdot |S|^2} \cdot \sum_{(i,j) \in S \times S} \left(x^i_{d_s} - x^j_{d_s}\right), \tag{B.22}$$

where:

d_s : is the chosen dimension to split over;
S : $\subset \{1, 2, \ldots, n\}$ is an set of indices.

Considering S_b the set of indices of inputs going through a node b, one can compute the gain of a split at node b as:

$$G_b \triangleq I(S_b) - \sum_{c \in \{d_b(\boldsymbol{x}^i) : i \in S_b\}} I(S_c) \tag{B.23}$$

The optimal splitting dimension and threshold are chosen so that they maximizes the gain G_b. Details of how this is computed is outside ths scope of this manuscript and is not provided herein. For more information on the subject the reader may refer to the work of Breiman et al. (1984).

Stopping only when total purity is achieved can yield complicated decision trees that overfit easily. This motivates the use for some early stopping criteria:

- a minimal number of observations going through each node.
- a minimal purity ratio computed as the ratio of all instances going through the node having the dominant class.
- a maximal depth of the tree.

To conclude, the class at each leaf is taken as the dominant class of instances entering it.

B.3.2 Bagging decision trees

While reducing the complexity of the decision tree can help avoid overfitting problems, one can risk, on the other hand, an underfitting in the classification. In order to find a compromise, an ensemble method can be adopted. The principle of this type of approaches is to multiply different underperforming classifiers and aggregate them together. These classifiers should be taken as diverse as possible in order to cover the whole feature space.

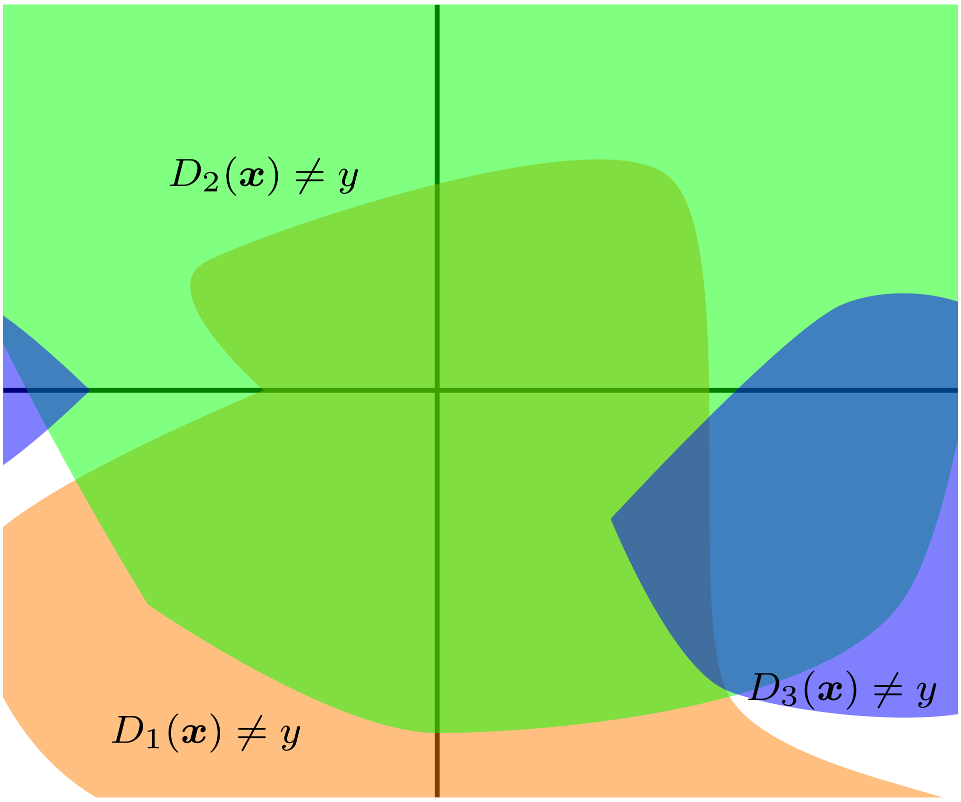
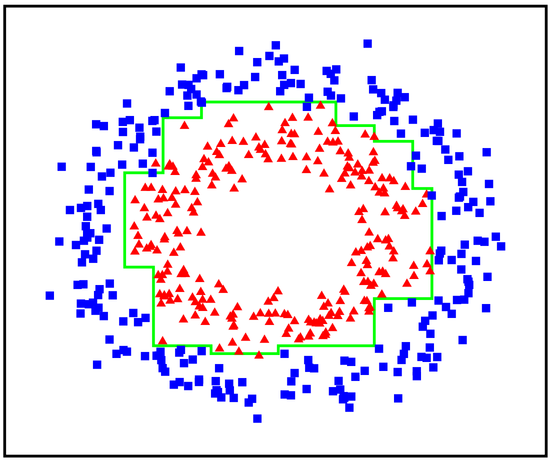

$D_2(\boldsymbol{x}) \neq y$
$D_3(\boldsymbol{x}) \neq y$
$D_1(\boldsymbol{x}) \neq y$

Figure B.6: Illustration of the principle of ensemble methods. For each decision function $\forall l = 1,2,3$ D_l, is represented the set where the each classifier fails $F(D_l) \leftarrow \left\{(\boldsymbol{x}, y) \in \prod_{i=1}^{d} \mathscr{X}_i \times \{1, 2, \ldots, C\} : D_l(\boldsymbol{x}) \neq y\right\}$. In the case of unweigthed bagging, the aggregated classifier will fail when more than half of the classifiers fail. In this case, the set $F(D_{\text{bagging}})$ is the union of intersections of two different $F(D_l)$: $F(D_{\text{bagging}}) = \bigcup_{l \neq p} F(D_l) \cap F(D_p)$. In this case where classifiers are diverse, the aggregated one fails less frequently than each single one.



(a) Decision tree separator

Figure B.7: Difference between a single decision tree and an RF visualized in feature space for a generated toy data. We see how an RF aggregates multiple shallow decision trees in order to achieve a good generalization power instead of overfitting to the sampled data.

The aggregation is achieved through a majority vote and, if the classifiers can provide probabilities, the vote can be weighted by the latter. This is illustrated in Figure B.6. Formally, the decision function of an unweighted aggregation of classifiers $(D_l)_{l=1,2,\ldots,L}$ can be written as:

$$\begin{aligned} D_{\text{hard ensemble},(D_l)_{l=1,2,\ldots,L}} : \prod_{i=1}^{d} \mathscr{X}_i &\to \{1,2,\ldots,C\} \\ \boldsymbol{x} &\mapsto \arg\max_{c=1,2,\ldots,C} \left|\{l \in \{1,2,\ldots,L\} : D_l(\boldsymbol{x}) = c\}\right| \end{aligned} , \tag{B.24}$$

while the weighted aggregation using classifier output probabilities $(p_l)_{l=1,2,\ldots,L}$ is expressed as:

$$\begin{aligned} D_{\text{weighted ensemble},((D_l,p_l))_{l=1,2,\ldots,L}} : \prod_{i=1}^{d} \mathscr{X}_i &\to \{1,2,\ldots,C\} \\ \boldsymbol{x} &\mapsto \arg\max_{c=1,2,\ldots,C} \sum_{l=1}^{L} p_l\left(D_l(\boldsymbol{x}) = c\right) \end{aligned} . \tag{B.25}$$

Bagging is an instance of ensemble approach. In order to have different classifiers specializing in a certain pattern, each one is trained on a randomly determined subset of the training data (Breiman, 1996). The idea is that each one of these subsets would exhibit a particular aspect that is crucial for the classification. The aggregation of each classifier knowledge would eventually help generalize at prediction time.

RF do not only rely on a simple bagging but also on a second random sampling: this time it is on the feature dimensions. When splitting a node at training time, only a subset of randomly chosen dimensions is considered (Breiman, 2001). We depict in Figure B.7 the difference between an RF and a single decision tree.

B.3.3 Properties

RFs are often used in practice as they offer some advantages. Hereafter are listed some of these:

- As decision trees deal with each feature independently, they can natively, as well as RFs, handle heterogenous data. It can handle boolean features along with integer or real ones.
- Since at each time only a subset of features is considered, RFs can scale easily under high dimensionality.
- The ensemble character of RFs allows them to adapt to outliers and hence, provided a large enough trainning dataset, can achieve a good generalization power.
- Prediction relies on simple comparisons and can be computed quickly.
- They yield probabilities that are consistent.
- They are inherently multi-class and do not need to be adapted.

As seen with SVMs, RFs have also some issues:

- It is not simple to interpret results of RFs and connect specific trees to recognizable patterns.

- They can take up a lot of memory to store all trees. They can also require a lot of computations to train.
- They do not allow the possibility of using kernels to represent the data.
- They do not handle well imbalanced datasets and must be adapted accordingly.

C Further experimental details

Contents

In this appendix, we provide numerical details of the experimental study. First, in Section C.1, we present in greater details statistics for each urban scene. Next, Section C.2 presents detailed account of each experiment that was mentioned before.

C.1 Dataset statistics

Table C.1 provides the number of each label at `finesse` levels 2 and 3 as well as their presence ratios, on each urban scene. These are vizualized in Figure 5.8.

Elancourt						
`Error family`	**Number**	**Ratio**	`Atomic error`	**Number**	**Ratio/Fam.**	**Overall ratio**
`Unqualifiable`	62	3.09	—	—	—	—
`Building Errors`	1,420	70.68	`BOS`	1,342	94.51	66.80
			`BUS`	473	33.31	23.54
			`BIB`	203	14.30	10.10
			`BIT`	99	6.97	4.93
`Facet Errors`	1,648	82.03	`FOS`	1,289	78.22	64.16
			`FUS`	315	19.11	15.68
			`FIB`	229	13.90	11.40
			`FIT`	30	1.82	1.49
			`FIG`	1,187	72.03	59.08

Nantes						
`Error Family`	**Number**	**Ratio**	`Atomic error`	**Number**	**Ratio/Fam.**	**Overall ratio**
`Unqualifiable`	56	7.49	—	—	—	—
`Building Errors`	435	58.16	`BOS`	291	66.90	38.90
			`BUS`	68	15.63	9.09
			`BIB`	99	22.76	13.24
			`BIT`	113	25.98	15.11
`Facet Errors`	568	75.94	`FOS`	478	84.15	63.90
			`FUS`	210	36.97	28.07
			`FIB`	164	28.87	21.93
			`FIT`	11	1.94	1.47
			`FIG`	446	78.52	59.63

Paris-13						
`Error Family`	**Number**	**Ratio**	`Atomic error`	**Number**	**Ratio/Fam.**	**Overall ratio**
`Unqualifiable`	23	4.81	—	—	—	—
`Building Errors`	303	63.39	`BOS`	202	66.67	42.26
			`BUS`	63	20.79	13.18
			`BIB`	55	18.15	11.51
			`BIT`	76	25.08	15.90
`Facet Errors`	411	85.98	`FOS`	249	60.58	52.09
			`FUS`	275	66.91	57.53
			`FIB`	144	35.04	30.13
			`FIT`	6	1.46	1.26
			`FIG`	383	93.19	80.13

Table C.1: Ground truth detailed statistics over the annotated datasets.

C.2 Experimental results

For every conducted experiment, we provide herein the F-score at each zone and for each feature configuration and error label. We also report the mean and standard deviation of these scores across all feature configurations.

C.2.1 Baseline feature analysis

Herein are reported the F-scores of baseline feature experiments analysed in Section 5.2.

	Elancourt			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	84.12	84.04	83.08	82.84
BUS	46.08	53.88	51.48	50.82
BIB	20.84	21.58	**26.20**	**26.88**
BIT	**39.68**	33.06	33.33	19.82
FOS	98.99	99.10	98.91	98.87
FUS	3.67	1.25	3.12	2.49
FIB	16.60	0.00	15.08	13.81
FIT	12.51	15.99	6.45	6.45
FIG	76.66	76.33	75.26	75.25

	Nantes			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	47.13	45.40	46.22	44.11
BUS	13.15	12.98	**40.82**	**37.50**
BIB	0.00	0.00	1.98	1.98
BIT	3.28	6.56	0.00	5.03
FOS	98.33	98.33	98.12	98.01
FUS	36.83	37.66	34.10	32.56
FIB	46.97	46.34	**54.54**	**52.66**
FIT	0.00	0.00	0.00	0.00
FIG	82.00	82.16	81.52	80.87

	Paris-13			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	53.64	55.45	57.71	55.95
BUS	11.60	14.29	**34.57**	14.09
BIB	0.00	0.00	0.00	0.00
BIT	5.00	0.00	2.57	0.00
FOS	97.19	97.19	97.98	97.19
FUS	79.73	78.91	79.46	78.91
FIB	57.46	56.06	57.89	56.93
FIT	0.00	0.00	0.00	0.00
FIG	90.67	91.33	90.67	91.33

Table C.2: F-scores for the baseline features ablation study results at **eFin** level 3. These are deduced from Table 5.2. Modalities, which stand out with at least 4.5 % in F-score, are distinguished in bold.

	Elancourt	**Nantes**	**Paris-13**
BOS	83.52 ± 0.66	45.71 ± 1.28	55.13 ± 4.10
BUS	50.56 ± 3.26	26.11 ± 15.12	18.64 ± 10.69
BIB	23.87 ± 3.10	0.99 ± 1.14	0 ± 0
BIT	31.47 ± 8.35	3.72 ± 2.82	1.89 ± 2.40
FOS	98.97 ± 0.10	98.20 ± 0.16	97.39 ± 0.40
FUS	2.63 ± 1.04	35.29 ± 2.37	79.25 ± 0.41
FIB	11.37 ± 7.67	50.13 ± 4.10	57.09 ± 0.79
FIT	10.35 ± 4.73	0 ± 0	0 ± 0
FIG	75.88 ± 0.73	81.64 ± 0.58	91.00 ± 0.38

Table C.3: Mean F-score and standard deviation for the feature baseline study.

C.2.2 Transferability study

We report in this subsection the F-scores drawn from the transferability experiments that were analysed in Section 5.3.1. We also provide the precision and recall scores that were too cumbersome to include in the main text.

	Elancourt → Nantes								**Elancourt → Paris-13**							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**		**Geom.**		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	99.66	42.03	**100**	**42.05**	90.72	43.64	100	42.05	96.53	43.82	**96.53**	**44.22**	74.75	44.02	78.71	42.97
`BUS`	0	0	0	—	11.76	66.67	**19.12**	**54.17**	0	—	0	—	**26.98**	**44.74**	4.76	27.27
`BIB`	0	—	0	—	0	—	0	—	0	—	0	—	**1.85**	**33.33**	**1.85**	**33.33**
`BIT`	2.65	50.0	1.77	66.67	1.77	66.67	**14.24**	**40.71**	**3.95**	**50.0**	2.63	50.0	1.32	100	0	—
`FOS`	98.33	98.12	98.33	98.12	**98.54**	**98.13**	100	69.38	97.19	97.58	97.17	97.58	98.80	95.72	**98.80**	**96.47**
`FUS`	2.38	38.46	0.95	33.33	**17.62**	**59.68**	15.71	56.90	8.36	95.83	3.63	90.91	**30.95**	**90.28**	20.73	91.94
`FIB`	18.90	81.58	16.46	84.38	**54.88**	**64.75**	83.54	43.08	11.80	60.71	11.11	64.0	**42.36**	**61.62**	39.58	64.04
`FIT`	0	—	0	—	**9.09**	**100**	9.09	33.33	0	0	0	0	0	0	0	0
`FIG`	**93.05**	**72.81**	94.39	70.52	93.05	72.55	100	64.45	86.16	88.47	87.73	86.82	87.21	87.89	**90.86**	**86.14**

	Nantes → Elancourt								**Nantes → Paris-13**							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**		**Geom.**		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	91.76	61.48	90.72	62.97	**86.90**	**66.40**	86.47	64.99	15.84	66.67	**19.31**	**75.0**	17.33	70.0	17.33	70.0
`BUS`	11.40	77.19	12.83	83.08	24.70	53.33	**23.75**	**57.47**	0	0	3.17	33.33	**6.35**	**50.0**	**6.35**	**50.0**
`BIB`	6.52	65.22	6.96	64.0	15.22	46.05	**15.65**	**46.75**	0	—	0	—	0	—	0	—
`BIT`	**8.14**	**87.5**	5.81	100	4.07	87.5	4.70	100	**17.11**	**34.21**	13.16	26.32	13.16	23.81	10.53	23.53
`FOS`	98.76	98.99	98.76	98.99	**98.84**	**98.92**	98.76	98.92	**97.59**	**97.2**	97.59	96.81	97.99	93.85	97.99	92.08
`FUS`	1.44	37.5	0.96	44.44	**4.32**	**72.0**	1.68	77.78	**51.27**	**81.03**	43.27	82.64	44.36	82.99	42.18	82.86
`FIB`	10.03	81.08	8.70	92.86	**27.09**	**69.23**	23.75	73.96	53.47	66.96	45.14	65.66	**54.86**	**66.39**	54.86	65.83
`FIT`	3.57	50.0	**3.57**	**100**	3.57	50.0	**3.57**	**100**	0	0	0	—	0	—	0	—
`FIG`	**86.10**	**68.78**	86.69	67.83	85.76	68.88	86.35	67.97	71.10	95.10	**91.64**	**95.47**	72.06	94.20	72.06	94.85

	Paris-13 → Nantes								**Paris-13 → Elancourt**							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**		**Geom.**		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	23.30	60.0	23.30	64.29	**25.57**	**66.95**	24.60	65.52	88.68	66.89	88.11	68.52	**82.74**	**72.78**	82.14	69.59
`BUS`	2.30	25.0	2.30	28.57	**9.20**	**40.0**	5.75	31.25	15.49	76.92	15.49	82.35	15.04	75.56	**25.0**	**50.49**
`BIB`	0	—	0	—	0	—	0	—	14.78	66.67	14.29	69.05	**26.60**	**43.90**	19.89	48.05
`BIT`	**30.48**	**30.77**	20.19	25.93	19.23	26.32	16.35	23.94	**10.88**	**84.21**	5.44	88.89	5.44	100	4.44	100
`FOS`	97.77	97.53	**98.02**	**97.54**	98.51	94.99	98.76	93.01	98.71	98.63	**98.71**	**98.71**	98.87	98.31	99.06	97.14
`FUS`	**40.12**	**77.97**	38.08	76.61	34.59	78.29	31.69	76.76	**4.86**	51.02	3.50	54.55	**4.86**	**75.76**	3.53	89.47
`FIB`	47.42	63.89	43.30	64.62	**53.61**	**63.41**	53.61	63.03	7.32	69.70	3.82	70.59	17.52	65.48	**21.86**	**70.11**
`FIT`	0	0	0	0	0	—	0	—	3.45	50.0	3.45	50.0	3.45	33.33	**4.34**	**33.33**
`FIG`	80.35	84.80	**80.93**	**84.55**	77.82	83.86	79.18	83.92	83.40	74.46	**85.19**	**74.05**	82.93	73.48	82.38	73.08

Table C.4: Transferability results of all six combinations reported, in percentage, at **eFin** level 3, based on baseline features. Bold indicates the best performing feature configuration in terms of F-score.

	Elancourt → Nantes				**Elancourt → Paris-13**			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**	**Geom.**	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
`BOS`	59.12	59.20	58.93	59.20	**60.28**	**60.65**	55.41	55.59
`BUS`	0.00	0.00	19.99	**28.26**	0.00	0.00	**33.66**	8.11
`BIB`	0.00	0.00	0.00	0.00	0.00	0.00	3.51	3.51
`BIT`	5.03	3.45	3.45	**21.10**	7.32	5.00	2.61	0.00
`FOS`	98.22	98.22	98.33	81.92	97.38	97.37	97.24	97.62
`FUS`	4.48	1.85	**27.21**	**24.62**	15.38	6.98	**46.10**	33.83
`FIB`	30.69	27.55	**59.41**	**56.85**	19.76	18.93	**50.21**	**48.92**
`FIT`	0.00	0.00	**16.67**	**14.28**	0.00	0.00	0.00	0.00
`FIG`	81.70	80.73	81.53	78.38	87.30	87.27	87.55	88.44

	Nantes → Elancourt				**Nantes → Paris-13**			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**	**Geom.**	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
`BOS`	73.63	74.34	75.28	74.21	25.60	30.71	27.78	27.78
`BUS`	19.87	22.23	**33.76**	**33.61**	0.00	5.79	**11.27**	**11.27**
`BIB`	11.85	12.55	**22.88**	**23.45**	0.00	0.00	0.00	0.00
`BIT`	**14.89**	10.98	7.78	8.98	**22.81**	17.55	16.95	14.55
`FOS`	98.87	98.87	98.88	98.84	97.39	97.20	95.88	94.94
`FUS`	2.77	1.88	**8.15**	3.29	**62.80**	56.80	57.82	55.90
`FIB`	17.85	15.91	**38.94**	**35.95**	59.46	53.50	60.08	59.85
`FIT`	6.66	6.89	6.66	6.89	0.00	0.00	0.00	0.00
`FIG`	76.47	76.11	76.40	76.07	81.37	**93.52**	81.66	81.90

	Paris-13 → Nantes				**Paris-13 → Elancourt**			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**	**Geom.**	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
`BOS`	33.57	34.20	37.01	35.77	76.26	77.09	77.44	75.35
`BUS`	4.21	4.26	**14.96**	9.71	25.79	26.08	25.09	**33.44**
`BIB`	0.00	0.00	0.00	0.00	24.20	23.68	**33.13**	28.13
`BIT`	**30.62**	22.70	22.22	19.43	**19.27**	10.25	10.32	8.50
`FOS`	97.65	97.78	96.72	95.80	98.67	98.71	98.59	98.09
`FUS`	52.98	50.87	47.98	44.86	8.87	6.58	9.13	6.79
`FIB`	54.44	51.85	**58.10**	**57.94**	13.25	7.25	27.64	**33.33**
`FIT`	0.00	0.00	0.00	0.00	6.45	6.45	6.25	7.68
`FIG`	82.52	82.70	80.73	81.48	78.68	79.23	77.92	77.45

Table C.5: F-scores for transferability results at **eFin** level 3 using baseline features. These are deduced from Table C.4. Modalities, which stand out with at least 4.5 % in F-score, are distinguished in bold.

	Elancourt → Nantes	**Elancourt → Paris-13**	**Nantes → Elancourt**
BOS	59.12 ± 0.13	57.98 ± 2.87	74.36 ± 0.68
BUS	12.06 ± 14.33	10.44 ± 15.94	27.37 ± 7.36
BIB	0 ± 0	1.75 ± 2.02	17.68 ± 6.34
BIT	8.26 ± 8.59	3.73 ± 3.15	10.66 ± 3.12
FOS	94.18 ± 8.17	97.40 ± 0.16	98.87 ± 0.02
FUS	14.54 ± 13.22	25.57 ± 17.69	4.02 ± 2.81
FIB	43.62 ± 16.83	34.46 ± 17.46	27.16 ± 11.96
FIT	7.74 ± 8.99	0 ± 0	6.78 ± 0.13
FIG	80.58 ± 1.53	87.64 ± 0.55	76.26 ± 0.20

	Nantes → Paris-13	**Paris-13 → Nantes**	**Paris-13 → Elancourt**
BOS	27.97 ± 2.10	35.14 ± 1.55	76.53 ± 0.93
BUS	7.08 ± 5.38	8.29 ± 5.14	27.60 ± 3.92
BIB	0 ± 0	0 ± 0	27.28 ± 4.37
BIT	17.96 ± 3.48	23.75 ± 4.81	12.09 ± 4.86
FOS	96.35 ± 1.16	96.99 ± 0.92	98.51 ± 0.29
FUS	58.33 ± 3.08	49.17 ± 3.53	7.84 ± 1.35
FIB	58.22 ± 3.16	55.58 ± 3.01	20.37 ± 12.16
FIT	0 ± 0	0 ± 0	6.71 ± 0.65
FIG	84.61 ± 5.94	81.86 ± 0.92	78.32 ± 0.79

Table C.6: F-score mean and standard deviation per `atomic` error for the transferability experiments using baseline features.

C.2.3 Generalization study

As with the transferability results, we report herein the recall, precision and F-scores of the generalization study from Section 5.3.2.

Elancourt ∪ Nantes → Paris-13								
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	26.73	50.47	26.73	62.07	31.68	69.57	**32.18**	**69.89**
`BUS`	0	—	0	—	**7.94**	**35.71**	3.17	33.33
`BIB`	0	—	0	—	0	—	0	—
`BIT`	6.58	50.0	**9.21**	**53.85**	2.63	100	0	—
`FOS`	97.19	97.19	97.59	97.2	**97.99**	**96.83**	98.39	94.23
`FUS`	**34.18**	**85.45**	32.0	84.62	33.09	85.85	31.64	87.88
`FIB`	15.97	62.16	13.89	58.82	37.5	66.67	**39.58**	**66.28**
`FIT`	0	—	0	—	0	—	0	—
`FIG`	69.97	94.37	73.37	93.36	72.06	94.52	**74.15**	**92.50**

Elancourt ∪ Paris-13 → Nantes								
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	33.66	52.26	30.10	56.71	37.86	61.58	**38.19**	**62.43**
`BUS`	0	—	0	—	**10.34**	**40.91**	5.75	35.71
`BIB`	0	—	0	—	0	0	0	0
`BIT`	**17.31**	**33.33**	14.42	28.30	10.58	34.38	11.54	34.29
`FOS`	97.77	97.77	98.02	97.78	**98.51**	**97.31**	98.76	94.77
`FUS`	26.16	81.08	22.09	85.39	**25.87**	**85.58**	24.71	85.86
`FIB`	14.95	63.04	13.91	61.36	**39.18**	**63.87**	37.63	65.77
`FIT`	0	—	0	—	0	0	0	—
`FIG`	82.10	83.40	**82.30**	**83.27**	81.32	82.77	81.91	82.71

Nantes ∪ Paris-13 → Elancourt								
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`BOS`	**96.76**	**55.49**	96.09	55.66	87.80	58.14	88.37	57.19
`BUS`	17.42	80.56	17.12	**89.06**	**22.82**	79.17	21.62	81.82
`BIB`	3.70	60.0	3.70	60.0	**7.82**	**57.58**	7.41	54.55
`BIT`	**8.62**	**64.52**	6.47	68.18	4.31	83.33	3.02	100
`FOS`	98.33	98.49	**98.33**	**98.56**	98.73	97.64	98.65	97.48
`FUS`	1.76	47.83	0.80	71.43	**10.22**	**81.01**	3.19	76.92
`FIB`	12.02	70.15	7.16	75.68	**46.29**	**61.99**	40.92	62.75
`FIT`	3.33	25.0	**3.33**	**33.33**	**3.33**	**33.33**	3.33	25.0
`FIG`	**88.95**	**74.23**	90.92	72.79	89.02	73.78	89.70	73.19

Table C.7: Results of the generalization study, reported in percentage, at the **eFin** level 3 using baseline features. Classifiers are trained on two zones and tested on the one that was left out.

	Elancourt			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	34.95	37.37	**43.54**	**44.07**
BUS	0.00	0.00	**12.99**	5.79
BIB	0.00	0.00	0.00	0.00
BIT	11.63	**15.73**	5.13	0.00
FOS	97.19	97.39	97.41	96.27
FUS	48.83	46.44	47.77	46.53
FIB	25.41	22.47	**48.00**	**49.56**
FIT	0.00	0.00	0.00	0.00
FIG	80.36	82.17	81.78	82.31

	Nantes			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	40.95	39.33	**46.89**	**47.39**
BUS	0.00	0.00	**16.51**	9.91
BIB	0.00	0.00	0.00	0.00
BIT	22.79	19.11	16.18	17.27
FOS	97.77	97.90	97.91	96.72
FUS	39.56	35.10	39.73	38.38
FIB	24.17	22.68	**48.57**	**47.87**
FIT	0.00	0.00	0.00	0.00
FIG	82.74	82.78	82.04	82.31

	Paris-13			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	70.53	70.49	69.96	69.44
BUS	28.65	28.72	**35.43**	**34.20**
BIB	6.97	6.97	**13.77**	**13.05**
BIT	**15.21**	11.82	8.20	5.86
FOS	98.41	98.44	98.18	98.06
FUS	3.40	1.58	**18.15**	6.13
FIB	20.52	13.08	**53.00**	**49.54**
FIT	5.88	6.06	6.06	5.88
FIG	80.93	80.85	80.69	80.61

Table C.8: F-scores for the generalization experiments at **eFin** level 3 using baseline features. These are deduced from Table C.7. Modalities, which stand out with at least 4.5 % in F-score, are distinguished in bold.

	Paris-13	**Nantes**	**Elancourt**
BOS	39.98 ± 4.53	43.64 ± 4.10	70.10 ± 0.51
BUS	4.70 ± 6.17	6.60 ± 8.09	31.75 ± 3.58
BIB	0 ± 0	0 ± 0	10.19 ± 3.73
BIT	8.12 ± 6.96	18.84 ± 2.90	10.27 ± 4.10
FOS	97.06 ± 0.54	97.58 ± 0.57	98.27 ± 0.18
FUS	47.39 ± 1.13	38.19 ± 2.15	7.31 ± 7.46
FIB	36.36 ± 14.41	35.82 ± 14.33	34.04 ± 20.18
FIT	0 ± 0	0 ± 0	5.97 ± 0.10
FIG	81.65 ± 0.89	82.47 ± 0.36	80.77 ± 0.15

Table C.9: Mean F-score and standard deviation for the generalization experiments using baseline features.

C.2.4 Representativeness study

As with the transferability and generalization results, we report herein the recall, precision and F-scores of the representativeness study from Section 5.3.3.

	20 %								30 %							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**		**Geom.**		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	77.63	73.44	**80.66**	**70.97**	83.99	68.10	81.82	68.15	81.49	68.76	76.96	73.84	**77.31**	**75.44**	78.50	72.07
BUS	36.44	74.46	**39.37**	**73.62**	28.92	75.13	31.36	71.96	32.72	75.66	35.84	71.84	33.18	75.27	**35.70**	**74.87**
BIB	8.07	67.65	7.42	87.5	**9.96**	**43.07**	0.33	33.33	4.86	75.0	2.77	77.78	6.32	69.57	**7.78**	**71.43**
BIT	**10.57**	**63.16**	5.73	72.22	5.33	36.11	2.59	60.0	**11.17**	**68.75**	7.11	73.68	11.34	59.46	3.90	80.0
FOS	98.19	98.62	**98.52**	**98.89**	98.02	98.94	98.57	98.63	98.67	98.67	**99.08**	**98.32**	98.22	98.36	98.80	98.45
FUS	**40.31**	**67.54**	29.17	73.05	35.11	71.87	25.95	72.15	**38.38**	**73.65**	28.73	69.87	29.33	73.39	37.50	67.09
FIB	22.09	62.91	5.20	76.67	**21.87**	**67.61**	16.70	66.67	19.16	73.0	17.49	67.0	23.77	69.70	**33.24**	**57.94**
FIT	0	—	0	—	**2.70**	**100**	2.86	9.09	**6.67**	**100**	3.13	100	0	—	2.78	50.0
FIG	81.77	76.52	**89.05**	**72.35**	88.04	72.13	89.48	71.94	**89.05**	**75.0**	87.36	75.62	84.14	78.62	85.57	74.78

	40 %								50 %							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**		**Geom.**		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	83.85	69.93	**78.65**	**74.55**	75.52	74.71	75.80	73.39	**80.39**	**70.64**	76.95	73.22	77.79	71.67	76.09	73.61
BUS	**39.54**	**79.77**	30.94	80.0	34.47	78.57	30.93	76.32	25.17	71.70	**32.58**	**81.45**	28.76	69.92	32.77	72.39
BIB	6.64	66.67	7.25	71.43	**9.76**	**58.82**	4.21	64.29	6.67	66.67	5.41	90.91	**7.43**	**81.25**	6.56	75.0
BIT	**5.03**	**81.82**	1.72	100	2.92	71.43	3.05	83.33	**11.03**	**57.69**	4.76	100	4.32	100	2.10	100
FOS	98.60	98.68	98.29	98.53	**98.84**	**98.68**	98.59	97.94	98.79	98.49	**98.81**	**98.81**	99.01	98.33	98.12	98.60
FUS	33.40	70.80	33.91	71.69	33.76	70.35	**35.41**	**72.37**	**41.12**	**76.47**	35.47	74.23	38.97	67.86	36.18	76.92
FIB	26.06	59.31	26.25	62.69	**34.19**	**63.69**	26.75	67.20	20.22	62.92	24.44	59.63	**33.46**	**59.06**	30.21	69.60
FIT	0	—	**3.57**	**100**	0	—	0	0	0	—	0	—	**4.76**	**50.0**	0	0
FIG	82.78	76.08	**89.74**	**74.50**	84.92	73.85	88.78	73.50	**85.20**	**76.97**	86.39	76.43	84.18	75.22	85.42	76.38

	60 %								70 %							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**		**Geom.**		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	**79.37**	**71.95**	78.53	71.85	76.98	72.44	76.05	72.79	74.83	73.69	81.01	68.41	**80.32**	**72.48**	77.36	74.26
BUS	30.67	74.19	**36.25**	**85.29**	35.50	74.55	36.84	71.19	31.47	78.48	**37.36**	**80.0**	33.52	72.84	30.65	78.21
BIB	6.92	84.62	8.0	85.71	8.33	75.0	**13.93**	**70.83**	6.38	75.0	4.90	71.43	4.81	62.5	**9.26**	**76.92**
BIT	**7.69**	**75.0**	5.88	100	4.88	100	2.42	100	**13.10**	**100**	5.62	100	1.76	100	3.95	100
FOS	97.53	97.89	98.46	98.71	98.77	98.53	**98.89**	**98.52**	98.72	98.72	**98.86**	**98.86**	99.16	98.18	98.68	98.68
FUS	33.13	73.51	**41.45**	**73.68**	36.56	70.35	35.67	65.64	**35.83**	**69.35**	30.87	72.45	31.15	68.47	34.30	66.4
FIB	28.57	65.26	12.73	77.78	**36.36**	**64.41**	29.27	63.16	12.12	86.96	21.43	71.74	30.13	61.04	**34.19**	**67.09**
FIT	0	—	0	—	0	—	**4.76**	**100**	0	—	0	—	**5.26**	**100**	**5.26**	**100**
FIG	83.13	78.53	**90.94**	**74.38**	83.78	77.58	87.40	69.67	86.73	75.04	**88.66**	**74.22**	82.45	74.32	83.19	74.10

Table C.10: Representativeness study on the fused dataset at different training sizes using baseline features. Results are reported in percentage. Bold is synonym of the best performing configuration of modalities F-score wise.

	20 %				**30 %**			
	Geom.	**Geom. $\oplus$ Hei.**	**Geom. $\oplus$ Im.**	**All**	**Geom.**	**Geom. $\oplus$ Hei.**	**Geom. $\oplus$ Im.**	**All**
BOS	75.48	75.51	75.21	74.36	74.59	75.37	76.36	75.15
BUS	48.93	51.30	41.76	43.68	45.68	47.82	46.06	48.35
BIB	14.42	13.68	16.18	0.65	9.13	5.35	11.59	14.03
BIT	**18.11**	10.62	9.29	4.97	19.22	12.97	19.05	7.44
FOS	98.40	98.70	98.48	98.60	98.67	98.70	98.29	98.62
FUS	50.49	41.69	47.17	38.17	50.46	40.72	41.91	48.11
FIB	32.70	9.74	33.05	26.71	30.35	27.74	**35.45**	**42.24**
FIT	0.00	0.00	**5.26**	**4.35**	**12.51**	6.07	0.00	5.27
FIG	79.06	79.84	79.29	79.76	81.42	81.07	81.29	79.81

	40 %				**50 %**			
	Geom.	**Geom. $\oplus$ Hei.**	**Geom. $\oplus$ Im.**	**All**	**Geom.**	**Geom. $\oplus$ Hei.**	**Geom. $\oplus$ Im.**	**All**
BOS	76.26	76.55	75.11	74.58	75.20	75.04	74.60	74.83
BUS	**52.87**	44.62	47.92	44.02	37.26	**46.54**	40.76	**45.12**
BIB	12.08	13.16	16.74	7.90	12.13	10.21	13.61	12.06
BIT	9.48	3.38	5.61	5.88	**18.52**	9.09	8.28	4.11
FOS	98.64	98.41	98.76	98.26	98.64	98.81	98.67	98.36
FUS	45.39	46.04	45.63	47.55	53.48	48.00	49.51	49.21
FIB	36.21	37.01	**44.49**	38.27	30.60	34.67	**42.72**	**42.13**
FIT	0.00	**6.89**	0.00	0.00	0.00	0.00	**8.69**	0.00
FIG	79.29	81.41	79.00	80.42	80.88	81.11	79.45	80.65

	60 %				**70 %**			
	Geom.	**Geom. $\oplus$ Hei.**	**Geom. $\oplus$ Im.**	**All**	**Geom.**	**Geom. $\oplus$ Hei.**	**Geom. $\oplus$ Im.**	**All**
BOS	75.48	75.04	74.64	74.38	74.26	74.18	76.20	75.78
BUS	43.40	50.88	48.10	48.55	44.93	50.93	45.91	44.04
BIB	12.79	14.63	14.99	**23.28**	11.76	9.17	8.93	**16.53**
BIT	13.95	11.11	9.31	4.73	**23.17**	10.64	3.46	7.60
FOS	97.71	98.58	98.65	98.70	98.72	98.86	98.67	98.68
FUS	45.67	**53.05**	48.12	46.22	47.25	43.29	42.82	45.23
FIB	39.74	21.88	**46.48**	40.00	21.27	33.00	40.35	**45.30**
FIT	0.00	0.00	0.00	**9.09**	0.00	0.00	**9.99**	**9.99**
FIG	80.76	81.83	80.56	77.53	80.46	80.80	78.17	78.38

Table C.11: F-scores for representativeness results at **eFin** level 3 using baseline features. These are deduced from Table C.10. Modalities, which stand out with at least 4.5 % in F-score, are distinguished in bold.

	20 %	**30 %**	**40 %**	**50 %**	**60 %**	**70 %**
BOS	75.14 ± 0.53	75.37 ± 0.74	**75.62 ± 0.93**	74.92 ± 0.26	74.89 ± 0.48	75.10 ± 1.04
BUS	46.42 ± 4.45	46.98 ± 1.31	47.36 ± 4.06	42.42 ± 4.23	**47.73 ± 3.13**	46.45 ± 3.08
BIB	11.23 ± 7.13	10.02 ± 3.70	12.47 ± 3.64	12.0 ± 1.39	**16.43 ± 4.67**	11.60 ± 3.53
BIT	10.75 ± 5.47	**14.67 ± 5.63**	6.09 ± 2.52	10.00 ± 6.08	9.77 ± 3.87	11.22 ± 8.49
FOS	98.55 ± 0.13	98.57 ± 0.19	98.52 ± 0.22	98.62 ± 0.19	98.41 ± 0.47	**98.73 ± 0.09**
FUS	44.38 ± 5.50	45.30 ± 4.73	46.15 ± 0.97	**50.05 ± 2.38**	48.27 ± 3.36	44.65 ± 2.02
FIB	25.55 ± 10.93	33.95 ± 6.39	**38.99 ± 3.76**	37.53 ± 5.89	37.03 ± 10.57	34.98 ± 10.44
FIT	2.40 ± 2.80	**5.96 ± 5.13**	1.72 ± 3.45	2.17 ± 4.35	2.27 ± 4.54	5.00 ± 5.77
FIG	79.49 ± 0.37	80.20 ± 0.74	80.03 ± 1.11	**80.52 ± 0.74**	80.17 ± 1.84	79.45 ± 1.37

Table C.12: Mean F-score and standard deviation for the representativeness experiments.

C.2.5 The evaluation Finesse study

Herein are reported the F-scores of **eFin** experiments analysed in Section 5.4. We start with the ablation study at **eFin** level 2.

	Elancourt	**Nantes**	**Paris-13**
Building errors	92.26 ± 0.09	76.26 ± 0.66	80.60 ± 0.0
Facet errors	91.08 ± 0.24	92.78 ± 0.16	94.99 ± 0.0

Table C.13: F-score mean and standard deviation for the baseline feature ablation study results per zone for **eFin** level 2.

Next are presented the detailed results of the transferability study at **eFin** level 2. As with the `atomic` error labels, we also provide the precision and recall ratios.

	Elancourt → Nantes							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
Building errors	**100**	**63.04**	100	62.86	100	62.86	100	62.86
Facet errors	**95.25**	**89.42**	97.18	85.05	95.42	87.56	95.95	86.92

	Elancourt → Paris-13							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
Building errors	99.34	66.45	99.67	66.52	99.67	66.52	**100**	**66.59**
Facet errors	92.94	91.17	**97.81**	**90.74**	96.35	91.24	96.84	91.28

	Nantes → Elancourt							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
Building errors	95.25	80.26	97.56	79.08	99.60	78.07	**99.41**	**78.56**
Facet errors	89.78	90.45	90.90	89.23	**91.21**	**89.70**	90.34	90.51

	Nantes → Paris-13							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
Building errors	85.15	68.62	84.16	68.55	**84.49**	**69.19**	83.50	68.75
Facet errors	88.56	93.33	**89.29**	**93.38**	87.10	93.72	87.59	93.75

	Paris-13 → Nantes							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
Building errors	87.14	67.88	87.58	67.99	88.03	68.33	**89.36**	**68.77**
Facet errors	90.25	91.64	89.75	91.75	**89.92**	**92.72**	90.08	92.25

	Paris-13 → Elancourt							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
Building errors	99.24	81.19	**99.56**	**81.29**	99.24	81.19	99.49	81.15
Facet errors	93.02	89.39	93.57	88.98	94.42	88.36	**95.63**	**88.00**

Table C.14: Transferability study on **eFin** level 2 using baseline features.

As before, we present the precision, recall and F-score of representativeness study at **eFin** level 2.

	Elancourt → Nantes	**Elancourt → Paris-13**
`Building errors`	77.23 ± 0.07	79.79 ± 0.13
`Facet errors`	91.37 ± 0.64	93.47 ± 0.97
	Nantes → Elancourt	**Nantes → Paris-13**
`Building errors`	87.44 ± 0.27	75.76 ± 0.33
`Facet errors`	90.26 ± 0.20	90.76 ± 0.43
	Paris-13 → Nantes	**Paris-13 → Elancourt**
`Building errors`	76.88 ± 0.62	89.38 ± 0.09
`Facet errors`	91.03 ± 0.24	91.33 ± 0.22

Table C.15: Mean F-score and standard deviation in each zone on **eFin** level 2 using baseline features.

	20 %							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`Building errors`	94.62	80.26	97.16	79.24	99.06	77.38	**99.38**	**78.15**
`Facet errors`	92.53	90.40	95.29	87.74	**95.68**	**88.81**	93.18	89.86

	30 %							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`Building errors`	99.59	78.52	96.90	79.09	**99.65**	**78.59**	99.76	77.65
`Facet errors`	**93.32**	**91.07**	97.57	86.89	91.04	91.85	94.22	89.17

	40 %							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`Building errors`	98.54	78.28	**99.52**	**78.44**	99.86	77.70	99.86	77.21
`Facet errors`	92.65	90.22	88.96	92.79	**93.81**	**90.07**	93.59	90.23

	50 %							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`Building errors`	100	76.81	**99.75**	**77.59**	99.75	77.55	99.83	77.12
`Facet errors`	95.03	88.07	96.12	87.31	93.78	90.05	**95.97**	**88.49**

	60 %							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`Building errors`	**99.28**	**79.05**	99.69	77.93	99.28	78.94	100	78.19
`Facet errors`	92.02	90.37	**97.20**	**87.96**	93.72	90.32	92.67	90.94

	70 %							
	Geom.		**Geom. ⊕ Hei.**		**Geom. ⊕ Im.**		**All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
`Building errors`	99.31	78.51	**99.59**	**78.40**	100	77.72	99.86	78.02
`Facet errors`	94.72	88.77	96.08	86.67	94.55	89.00	**94.09**	**89.94**

Table C.16: Representativeness study on **eFin** level 2.

	20 %	**30 %**	**40 %**
`Building errors`	87.13 ± 0.31	87.53 ± 0.38	87.37 ± 0.27
`Facet errors`	91.60 ± 0.35	91.79 ± 0.33	91.51 ± 0.50
	50 %	**60 %**	**70 %**
`Building errors`	87.11 ± 0.19	**87.80 ± 0.24**	87.62 ± 0.12
`Facet errors`	91.72 ± 0.31	**91.83 ± 0.49**	91.61 ± 0.35

Table C.17: F-score mean and standard deviation for the representativeness experiments on **eFin** level 2 using baseline features.

C.2.6 SVM to RF comparison

In this subsection, we provide the F-score results obtained with baseline features using both RF and SVM classifiers. A detailed analysis is presented in Section 7.2.

	Elancourt			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	84.12	84.04	83.08	82.84
BUS	46.08	53.88	51.48	50.82
BIB	20.84	21.58	**26.20**	**26.88**
BIT	**39.68**	33.06	33.33	19.82
FOS	98.99	99.10	98.91	98.87
FUS	3.67	1.25	3.12	2.49
FIB	16.60	0.00	15.08	13.81
FIT	12.51	15.99	6.45	6.45
FIG	76.66	76.33	75.26	75.25

	Na-P13			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	62.44	60.36	59.47	59.15
BUS	33.12	37.27	**52.46**	**50.56**
BIB	3.84	1.29	1.29	2.59
BIT	**10.10**	6.18	4.17	2.10
FOS	98.42	98.41	98.62	98.69
FUS	72.86	72.37	73.35	72.63
FIB	64.87	63.69	**69.67**	**69.68**
FIT	11.76	11.76	11.76	**21.05**
FIG	88.14	88.50	88.64	89.00

Table C.18: F-scores on the two experimental sets of interest at **eFin** level 3 using an RF and baseline features. These are deduced from Table 7.1. Modalities, which stand out with at least 4.5 % in F-score, are distinguished in bold.

	Elancourt			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	91.71	91.43	91.42	53.87
BUS	47.06	45.23	45.21	**58.64**
BIB	67.94	**90.42**	**90.41**	54.23
BIT	84.98	100	100	100
FOS	69.96	68.24	68.24	**75.51**
FUS	67.78	35.79	35.81	28.72
FIB	**46.38**	29.30	29.30	81.39
FIT	93.75	100	100	100
FIG	**86.48**	73.33	73.41	57.09

	Na-P13			
	Geom.	**Geom. ⊕ Hei.**	**Geom. ⊕ Im.**	**All**
BOS	**49.04**	35.22	35.22	35.22
BUS	**42.81**	19.67	22.27	16.56
BIB	**28.61**	22.71	22.71	22.71
BIT	**46.76**	25.04	25.04	25.04
FOS	85.36	85.31	85.26	**89.76**
FUS	74.61	36.13	36.13	36.13
FIB	**54.79**	40.18	40.18	39.10
FIT	94.44	100	100	100
FIG	**85.86**	74.51	74.43	66.00

Table C.19: F-scores on the two datasets of interest at **eFin** level 3 using an SVM and baseline features. These are deduced from Table 7.2. Modalities, which stand out with at least 4.5 % in F-score, are distinguished in bold. The scores suspected to result from overfitting are underlined.

	RF		SVM	
	Elancourt	**Na-P13**	**Elancourt**	**Na-P13**
BOS	83.52 ± 0.66	60.35 ± 1.48	82.11 ± 18.83	38.68 ± 6.91
BUS	50.56 ± 3.26	43.35 ± 9.60	49.03 ± 6.46	25.33 ± 11.89
BIB	23.87 ± 3.10	2.25 ± 1.22	75.75 ± 17.84	24.18 ± 2.95
BIT	31.47 ± 8.35	5.64 ± 3.41	96.24 ± 7.51	30.47 ± 10.86
FOS	98.97 ± 0.10	98.54 ± 0.14	70.49 ± 3.45	86.42 ± 2.23
FUS	2.63 ± 1.04	72.80 ± 0.42	42.03 ± 17.49	45.75 ± 19.24
FIB	11.37 ± 7.67	66.98 ± 3.15	46.59 ± 24.56	43.56 ± 7.50
FIT	10.35 ± 4.73	14.08 ± 4.64	98.44 ± 3.12	98.61 ± 2.78
FIG	75.88 ± 0.73	88.57 ± 0.36	72.58 ± 12.03	75.20 ± 8.15

Table C.20: Mean F-score and standard deviation using RF and SVM and baseline features. These were computed based on results reported in Tables C.18 and C.19.

C.2.7 ScatNet to baseline comparison

As in the previous subsection, we present herein the F-scores of both classifiers but, this time, using ScatNet features. A detailed analysis is provided in Section 7.3.1.

	Elancourt					
	Geom.	**Geom. ⊕ S-Hei.**	**Geom. ⊕ S(d)-Im.**	**S(d)-All**	**Geom. ⊕ S(c)-Im.**	**S(c)-All**
BOS	84.12	85.07	84.55	85.76	85.82	86.10
BUS	46.08	**64.83**	44.76	**63.80**	51.83	**64.21**
BIB	20.84	20.43	10.34	10.34	23.59	25.11
BIT	39.68	**60.00**	33.90	**56.94**	51.51	53.73
FOS	98.99	99.14	98.26	98.12	99.42	99.46
FUS	3.67	6.16	9.12	10.84	10.26	**22.10**
FIB	16.60	1.72	0.88	0.00	**21.87**	**21.17**
FIT	12.51	34.29	**43.25**	34.29	18.74	12.91
FIG	76.66	83.66	83.40	84.08	82.95	83.81

	Na-P13					
	Geom.	**Geom. ⊕ S-Hei.**	**Geom. ⊕ S(d)-Im.**	**S(d)-All**	**Geom. ⊕ S(c)-Im.**	**S(c)-All**
BOS	62.44	54.50	56.27	56.46	60.87	59.16
BUS	33.12	1.51	43.37	41.21	43.37	41.46
BIB	3.84	5.09	0.00	0.00	5.07	3.84
BIT	10.10	0.00	15.69	12.93	13.80	11.00
FOS	98.42	97.56	97.10	95.80	98.62	98.68
FUS	72.86	73.27	71.12	70.53	73.34	73.19
FIB	64.87	41.84	62.15	59.50	**76.22**	**76.58**
FIT	11.76	11.76	22.22	22.22	**40.00**	**40.00**
FIG	88.14	90.32	90.17	90.45	90.91	90.99

Table C.21: F-scores on the two datasets of interest at **eFin** level 3 using an RF based on ScatNet derived features. These are deduced from Table 7.5. Modalities, which stand out with at least 4.5 % in F-score, are distinguished in bold.

	Elancourt					
	Geom.	**Geom. ⊕ S-Hei.**	**Geom. ⊕ S(d)-Im.**	**S(d)-All**	**Geom. ⊕ S(c)-Im.**	**S(c)-All**
BOS	91.71	88.89	90.57	89.61	90.68	89.81
BUS	47.06	61.26	**72.75**	62.02	45.96	59.35
BIB	67.94	59.73	40.74	46.45	**90.91**	81.20
BIT	84.98	54.71	89.81	62.42	**100.00**	64.79
FOS	69.96	77.88	**95.58**	82.98	70.56	76.34
FUS	67.78	57.53	**75.41**	72.46	67.84	74.57
FIB	46.38	44.31	**85.12**	**84.75**	29.92	30.54
FIT	93.75	93.75	100.00	100.00	100.00	100.00
FIG	**86.48**	**85.92**	75.88	78.22	75.40	68.92

	Na-P13					
	Geom.	**Geom. ⊕ S-Hei.**	**Geom. ⊕ S(d)-Im.**	**S(d)-All**	**Geom. ⊕ S(c)-Im.**	**S(c)-All**
BOS	**49.04**	**48.76**	35.22	35.22	35.22	35.22
BUS	42.81	40.77	44.57	**50.11**	20.42	40.08
BIB	28.61	27.41	22.74	22.74	22.71	22.71
BIT	**46.76**	**46.50**	26.52	26.51	25.04	25.04
FOS	85.36	**88.66**	85.31	**89.71**	85.28	**89.23**
FUS	**74.61**	**71.44**	36.13	60.25	36.13	49.66
FIB	**54.79**	**53.41**	44.28	45.52	40.18	40.08
FIT	94.44	94.44	100.00	100.00	100.00	100.00
FIG	85.86	86.20	87.13	87.19	74.98	82.71

Table C.22: F-scores on the two datasets of interest at **eFin** level 3 using an SVM based on ScatNet derived features. These are deduced from Table 7.7. Modalities, which stand out with at least 4.5 % in F-score, are distinguished in bold.

	Elancourt											
	Geom.		**Geom. ⊕ S-Hei.**		**Geom. ⊕ S(d)-Im.**		**S(d)-All**		**Geom. ⊕ S(c)-Im.**		**S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	97.67	86.44	97.67	84.81	82.07	81.89	82.00	81.75				
BUS	32.27	86.85	36.31	87.69	75.80	57.95	76.22	58.28				
BIB	97.02	52.27	96.54	55.24	65.35	33.17	64.85	33.08				
BIT	100.0	73.88	100.0	75.57	89.80	29.24	88.78	28.9				
FOS	53.88	99.71	56.14	99.72	82.81	90.95	82.66	90.93				
FUS	96.49	52.24	96.50	52.06	88.85	31.14	88.85	31.10				
FIB	33.77	74.03	32.89	74.26	87.77	30.09	87.72	29.90				
FIT	100.0	88.24	100.0	88.24	100.0	69.77	100.0	69.77				
FIG	84.57	88.47	84.66	88.25	66.10	80.50	66.02	80.39				

	Na-P13											
	Geom.		**Geom. ⊕ S-Hei.**		**Geom. ⊕ S(d)-Im.**		**S(d)-All**		**Geom. ⊕ S(c)-Im.**		**S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS												
BUS												
BIB												
BIT												
FOS												
FUS												
FIB												
FIT												
FIG												

Table C.23: SVM applied to ScatNet based features. Results are expressed in percentage on the two datasets at **eFin** level 3 with $\gamma = 1e - 4$.

Elancourt												
	Geom.		**Geom. ⊕ S-Hei.**		**Geom. ⊕ S(d)-Im.**		**S(d)-All**		**Geom. ⊕ S(c)-Im.**		**S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS	97.67	86.44	85.21	86.05								
BUS	32.27	86.85	84.50	52.23								
BIB	97.02	52.27	89.60	21.75								
BIT	100.0	73.88	96.94	36.12								
FOS	53.88	99.71	68.35	90.15								
FUS	96.49	52.24	90.13	29.95								
FIB	33.77	74.03	51.09	74.52								
FIT	100.0	88.24	100.0	88.24								
FIG	84.57	88.47	74.94	74.52								

Na-P13												
	Geom.		**Geom. ⊕ S-Hei.**		**Geom. ⊕ S(d)-Im.**		**S(d)-All**		**Geom. ⊕ S(c)-Im.**		**S(c)-All**	
	Rec	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***	***Rec***	***Prec***
BOS												
BUS												
BIB												
BIT												
FOS												
FUS												
FIB												
FIT												
FIG												

Table C.24: SVM applied to ScatNet based features. Results are expressed in percentage on the two datasets at **eFin** level 3 with $\gamma = 1e - 2$.

	RF			
	Elancourt		**Na-P13**	
	Deletion	Channel	Deletion	Channel
BOS	84.88 ± 0.71	**85.28 ± 0.89**	57.42 ± 3.46	**59.24 ± 3.44**
BUS	54.87 ± 10.93	**56.74 ± 9.29**	29.80 ± 19.37	**29.87 ± 19.42**
BIB	15.49 ± 5.95	**22.49 ± 2.24**	2.23 ± 2.63	**4.46 ± 0.71**
BIT	47.63 ± 12.80	**51.23 ± 8.50**	**9.68 ± 6.84**	8.72 ± 6.03
FOS	98.63 ± 0.51	**99.25 ± 0.23**	97.22 ± 1.09	**98.32 ± 0.52**
FUS	7.45 ± 3.17	**10.55 ± 8.16**	71.95 ± 1.33	**73.17 ± 0.21**
FIB	4.80 ± 7.90	**15.34 ± 9.38**	57.09 ± 10.40	**64.88 ± 16.29**
FIT	**31.08 ± 13.08**	19.61 ± 10.19	16.99 ± 6.04	**25.88 ± 16.30**
FIG	**81.95 ± 3.54**	81.77 ± 3.43	89.77 ± 1.09	**90.09 ± 1.34**

	SVM			
	Elancourt		**Na-P13**	
	Deletion	Channel	Deletion	Channel
BOS	90.20 ± 1.22	**90.27 ± 1.20**	42.06 ± 7.90	42.06 ± 7.90
BUS	**60.77 ± 10.54**	53.41 ± 8.01	**44.56 ± 4.01**	36.02 ± 10.47
BIB	53.71 ± 12.38	**74.94 ± 13.84**	25.37 ± 3.08	25.36 ± 3.10
BIT	72.98 ± 17.06	**76.12 ± 20.29**	**36.57 ± 11.61**	35.83 ± 12.47
FOS	**81.60 ± 10.75**	73.68 ± 4.01	**87.26 ± 2.26**	87.13 ± 2.11
FUS	**68.29 ± 7.83**	66.93 ± 7.03	**60.61 ± 17.44**	57.96 ± 18.30
FIB	**65.14 ± 22.87**	37.79 ± 8.77	**49.50 ± 5.36**	47.11 ± 8.09
FIT	96.88 ± 3.61	96.88 ± 3.61	97.22 ± 3.21	97.22 ± 3.21
FIG	**81.62 ± 5.37**	79.18 ± 8.53	**86.60 ± 0.67**	82.44 ± 5.21

Table C.25: Mean F-score and standard deviation using RF and SVM based on ScatNet features. Bold indicates the best option between `channel` and `deletion`.

C.2.8 Graph kernels and ScatNet to baseline comparison

In this subsection, we present the F-score results obtained with graph kernels and ScatNet using the SVM classifier. A detailed analysis is provided in Section 7.3.3.

	Elancourt					
	K-Geom.	**K-Geom. ⊕ S-Hei.**	**K-Geom. ⊕ S(d)-Im.**	**K-S(d)-All**	**K-Geom. ⊕ S(c)-Im.**	**K-S(c)-All**
BOS	87.04	86.67	87.56	87.32	87.61	87.41
BUS	66.11	67.85	64.81	66.02	65.14	66.07
BIB	65.62	65.64	66.06	66.67	66.66	67.27
BIT	58.29	58.29	59.63	59.19	58.82	58.82
FOS	98.34	98.34	98.34	98.34	98.31	98.31
FUS	44.46	44.35	45.25	45.12	44.97	44.97
FIB	48.22	48.05	48.41	48.07	48.35	48.20
FIT	100.00	100.00	100.00	100.00	100.00	100.00
FIG	74.86	75.17	75.47	75.69	75.74	75.95
	Na-P13					
	K-Geom.	**K-Geom. ⊕ S-Hei.**	**K-Geom. ⊕ S(d)-Im.**	**K-S(d)-All**	**K-Geom. ⊕ S(c)-Im.**	**K-S(c)-All**
BOS	**50.85**	**50.66**	46.10	46.10	46.21	46.10
BUS	46.38	45.87	47.01	47.01	47.21	47.01
BIB	44.83	44.75	47.48	47.48	47.56	47.48
BIT	47.54	47.33	48.70	48.70	48.84	48.70
FOS	97.13	97.71	97.34	97.34	97.20	97.34
FUS	72.73	72.51	73.02	73.02	73.22	73.02
FIB	67.87	67.71	67.62	67.62	67.78	67.62
FIT	100.00	100.00	100.00	100.00	100.00	100.00
FIG	84.84	85.15	86.00	86.00	85.41	86.00

Table C.26: F-scores on the two datasets of interest at **eFin** level 3 using an SVM based on graph kernels and ScatNet derived features. These are deduced from Table 7.7. Modalities, which stand out with at least 4.5 % in F-score, are distinguished in bold.

	Elancourt		**Na-P13**	
	Deletion	Channel	Deletion	Channel
BOS	87.15 ± 0.38	87.18 ± 0.41	48.43 ± 2.69	48.46 ± 2.66
BUS	66.20 ± 1.25	66.29 ± 1.13	46.57 ± 0.55	46.62 ± 0.61
BIB	65.99 ± 0.49	66.30 ± 0.81	46.13 ± 1.55	46.16 ± 1.58
BIT	58.85 ± 0.67	58.55 ± 0.31	48.07 ± 0.74	48.10 ± 0.78
FOS	98.34 ± 0.00	98.33 ± 0.02	97.38 ± 0.24	97.34 ± 0.26
FUS	44.80 ± 0.46	44.69 ± 0.33	72.82 ± 0.25	72.87 ± 0.32
FIB	48.19 ± 0.16	48.21 ± 0.12	67.70 ± 0.12	67.74 ± 0.11
FIT	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
FIG	75.30 ± 0.36	75.43 ± 0.50	85.50 ± 0.59	85.35 ± 0.49

Table C.27: Mean F-score and standard deviation using SVM with graph kernels and ScatNet based features.

Acronyms

2.5D two and half dimensional. xv, 42, 46

2D two dimensional. xv, 36, 37, 41, 51, 73, 80, 90, 91, 140

3D three dimensional. i, v, xi, xii, xv, xvi, 35–53, 55–57, 59–62, 64–67, 70–72, 74–80, 85–88, 92–94, 97, 98, 100, 105, 124, 140, 147, 178

BIB Building Imprecise Borders. xv, 72, 73, 77, 81

BIG Building Imprecise Geometry. xv, 73, 75, 81

BIM Building information model. 43–46

BIT Building Inaccurate Topology. xv, 73, 74, 81

BOS Building Over Segmentation. xv, 71, 72, 75

BUS Building Under Segmentation. xv, 70–72, 74, 75

CAD Computer aided design. xv, 41–43, 45

ConvNet Convolutional Neural Network. xvi, 124–129, 139, 140

DoF Degree of Freedom. 38

DSM Digital Surface Model. v, 42, 51, 56, 66, 75, 77, 80, 87, 88, 90, 91, 95, 98, 99, 140, 178

DTM Digital Terrain Model. 42

eFin evaluation Finesse. xii, xiv, xvi, xix–xxi, 82, 83, 86, 87, 95, 97, 100, 105, 109, 118–121, 145, 149, 151, 156, 161, 167, 170, 178, 197, 199, 201, 202, 204, 205, 208–215, 217

eLoD evaluation Level of Detail. xix, 82, 83, 86, 87, 95, 178

ENSG Ecole Nationale des Sciences Géographique. i, xxiv, 99

FIB Facet Imprecise Borders. xvi, 76–78, 81

FIG Facet Imprecise Geometry. xvi, 79–81, 179

FIT Facet Inaccurate Topology. xvi, 78, 79, 81

FOS Facet Over Segmentation. xvi, 75–77, 81

FUS Facet Under Segmentation. xv, xvi, 74–76, 78, 81

G-ConvNet Group equivariant Convolutional Neural Network. 128

GIS Geographic Information Science. 36, 43–46, 50, 52, 60, 61

IFC Industry Foundation Classes. xv, 46

IGN Institut National de l'Information Géographique et Forestière. i, 39

ISPRS International Society for Photogrammetry and Remote Sensing. 51

LCC Linear Cell Complex. 50

LiDAR Light Detection And Ranging. xv, 36, 41, 47, 48, 60, 65, 66, 70, 88

LoD Level of Detail. xv, xvi, 46, 47, 52–54, 61, 62, 69, 70, 73, 76, 80–83, 86, 87, 95, 99, 101, 107

MKL Multiple Kernel Learning. 139, 145, 154, 163, 169, 190

NDVI Normalized Difference Vegetation Index. 65

OGC Open Geospatial Consortium. 50

RaDAR Radio Detection And Ranging. 47

RBF Radial Basis Function. 139, 145, 190

RF Random Forest. v, xiii, xiv, xvi, xvii, xix–xxi, 88, 95, 107, 108, 139, 145, 147, 149–160, 163, 165, 166, 169, 174, 175, 178, 183, 191, 194, 195, 197, 211–213, 216, 235

RMSE Root Mean Square Error. xii, xix, 46, 49, 62, 63, 65, 66, 88, 91, 97, 105, 108

ScatNet Scattering Network. xiii, xiv, xvii, xix–xxi, 56, 57, 88, 123–126, 128–130, 138–142, 144, 145, 147, 155–165, 169–173, 175, 178, 197, 212–217

SfM Structure-from-Motion. 88

SMO Sequential Minimal Optimization. 188, 190

SVM Support Vector Machine. v, xiii, xiv, xvii, xix–xxi, 88, 133, 134, 139, 144, 145, 147, 149–155, 160–172, 174, 175, 178, 183, 185–191, 195, 197, 211–217, 235

VHR Very High Resolution. v, 56, 70, 91, 99

Bibliography

Adeline, Karine R.M., M. Chen, X. Briottet, S.K. Pang, and Nicolas Paparoditis (2013). “Shadow detection in very high spatial resolution aerial images: A comparative study.” In: *ISPRS Journal of Photogrammetry and Remote Sensing* 80, pp. 21–38 (cit. on p. 60).

Aiolli, Fabio and Michele Donini (2015). “EasyMKL: a scalable multiple kernel learning algorithm.” In: *Neurocomputing* 169, pp. 215–224 (cit. on p. 145).

Akca, Devrim, Mark Freeman, Isabel Sargent, and Armin Gruen (2010). “Quality assessment of 3D building data.” In: *The Photogrammetric Record* 25.132, pp. 339–355 (cit. on p. 65).

Alam, Nazmul, Detlev Wagner, Mark Wewetzer, Julius von Falkenhausen, Volker Coors, et al. (2014). “Towards automatic validation and healing of CityGML models for geometric and semantic consistency.” In: *Innovations in 3D Geo-Information Sciences*. Springer, pp. 77–91 (cit. on p. 50).

Albrecht, F, J Moser, and I Hijazi (2013). “Assessing façade visibility in 3D city models for city marketing.” In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 1–5 (cit. on p. 39).

Andèn, Joakim and Stéphane Mallat (2014). “Deep scattering spectrum.” In: *IEEE Transactions on Signal Processing* 62.16, pp. 4114–4128 (cit. on p. 140).

Andreux, Mathieu, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, et al. (2018). *Kymatio: Scattering Transforms in Python*. Tech. rep. arXiv preprint arXiv:1812.11214 (cit. on p. 145).

Ardeshir, Shervin, Amir Roshan Zamir, Alejandro Torroella, and Mubarak Shah (2014). “GIS-assisted object detection and geospatial localization.” In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 602–617 (cit. on p. 38).

Armagan, Anil, Martin Hirzer, and Vincent Lepetit (2017). “Semantic segmentation for 3D localization in urban environments.” In: *Joint Urban Remote Sensing Event (JURSE)*. IEEE, pp. 1–4 (cit. on p. 42).

Aronszajn, Nachman (1950). “Theory of reproducing kernels.” In: *Transactions of the American mathematical society* 68.3, pp. 337–404 (cit. on p. 189).

Arroyo Ohori, Ken, Hugo Ledoux, and Jantien Stoter (2015). “A dimension-independent extrusion algorithm using generalised maps.” In: *International Journal of Geographical Information Science* 29.7, pp. 1166–1186 (cit. on p. 61).

Arth, Clemens, Christian Pirchheim, Jonathan Ventura, Dieter Schmalstieg, and Vincent Lepetit (2015). “Instant outdoor localization and slam initialization from 2.5 d maps.” In: *IEEE Transactions on Visualization and Computer Graphics* 21.11, pp. 1309–1318 (cit. on p. 38).

Aubry, Mathieu, Bryan C Russell, and Josef Sivic (2014). “Painting-to-3D model alignment via discriminative visual elements.” In: *ACM Transactions on Graphics (ToG)* 33.2, p. 14 (cit. on p. 39).

Baillard, Caroline, Cordelia Schmid, Andrew Zisserman, and Andrew Fitzgibbon (Sept. 1999). “Automatic line matching and 3D reconstruction of buildings from multiple views." In: *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*. Vol. XXXII. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 3-2W5, pp. 69–80 (cit. on pp. 61, 72).

Bao, Fan, Dong-Ming Yan, Niloy J Mitra, and Peter Wonka (2013). "Generating and exploring good building layouts." In: *ACM Transactions on Graphics (ToG)* 32.4, p. 122 (cit. on p. 62).

Berger, Matthew, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Joshua Levine, et al. (Apr. 2014). "State of the Art in Surface Reconstruction from Point Clouds." In: *Eurographics 2014 - State of the Art Reports*. Vol. 1. EUROGRAPHICS star report 1. Strasbourg, France, pp. 161–185 (cit. on p. 51).

Biljecki, Filip and Y. Dehbi (2019). "Raise the roof: towards generating LoD2 models without aerial surveys using machine learning." In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* IV.4/W8, pp. 27–34 (cit. on p. 60).

Biljecki, Filip, Gerard Heuvelink, Hugo Ledoux, and Jantien Stoter (2015a). "Propagation of positional error in 3D GIS: estimation of the solar irradiation of building roofs." In: *International Journal of Geographical Information Science* 29.12, pp. 2269–2294 (cit. on pp. 26, 37).

Biljecki, Filip, Hugo Ledoux, Xin Du, Jantien Stoter, Kean Huat Soon, et al. (2016a). "The most common geometric and semantic errors in citygml datasets." In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (cit. on pp. 27, 50).

Biljecki, Filip, Hugo Ledoux, and Jantien Stoter (2016b). "An improved LOD specification for 3D building models." In: *Computers, Environment and Urban Systems* 59, pp. 25–37 (cit. on pp. 26, 45 sqq.).

Biljecki, Filip, Hugo Ledoux, and Jantien Stoter (2017). "Generating 3D city models without elevation data." In: *Computers, Environment and Urban Systems* 64, pp. 1–18 (cit. on p. 60).

Biljecki, Filip, Hugo Ledoux, Jantien Stoter, and Junqiao Zhao (2014). "Formalisation of the level of detail in 3D city modelling." In: *Computers, Environment and Urban Systems* 48, pp. 1–15 (cit. on p. 46).

Biljecki, Filip, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin (2015b). "Applications of 3D City Models: State of the Art Review." In: *ISPRS International Journal of Geo-Information* 4.4, pp. 2842–2889 (cit. on pp. 26, 36 sqq., 44).

Billen, Roland and Sisi Zlatanova (2003). "3D spatial relationships model: a useful concept for 3D cadastre?" In: *Computers, Environment and Urban Systems* 27.4, pp. 411–425 (cit. on p. 36).

Bonin-Font, Francisco, Alberto Ortiz, and Gabriel Oliver (2008). "Visual navigation for mobile robots: A survey." In: *Journal of intelligent and robotic systems* 53.3, pp. 263–296 (cit. on p. 38).

Bordes, Antoine, Seyda Ertekin, Jason Weston, and Léon Bottou (2005). "Fast kernel classifiers with online and active learning." In: *Journal of Machine Learning Research* 6.Sep, pp. 1579–1619 (cit. on p. 190).

Borgwardt, Karsten M. and Hans-Peter Kriegel (Nov. 2005). "Shortest-path kernels on graphs." In: *Fifth IEEE International Conference on Data Mining*. ICDM '05. IEEE, pp. 74–81 (cit. on pp. 137, 139).

Boser, Bernhard E, Isabelle M Guyon, and Vladimir N Vapnik (1992). "A training algorithm for optimal margin classifiers." In: *fifth annual workshop on Computational learning theory*. ACM, pp. 144–152 (cit. on p. 189).

Boudet, Laurence, Nicolas Paparoditis, Franck Jung, Gilles Martinoty, and Marc Pierrot-Deseilligny (2006). "A supervised classification approach towards quality self-diagnosis of 3D building models using digital aerial imagery." In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXVI.3, pp. 136–141 (cit. on pp. 28, 63, 65 sq., 82, 88).

Brachmann, Eric and Carsten Rother (2018). "Learning less is more-6d camera localization via 3d surface regression." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 4654–4662 (cit. on p. 47).

Brasebin, Mickaël, Julien Perret, Michaël Borne, Paul Chapron, Imran Lokhat, et al. (2014–2019). *SimPLU3D*. `https://simplu3d.github.io` (cit. on p. 38).

Brasebin, Mickaël, Julien Perret, Sébastien Mustière, and Christiane Weber (2018). "3D urban data to assess local urban regulation influence." In: *Computers, Environment and Urban Systems* 68, pp. 37–52 (cit. on p. 36).

Brasebin, Mickaël, Julien Perret, and Romain Reuillon (2017). "Stochastic buildings generation to assist in the design of right to build plans." In: *Advances in 3D Geoinformation*. Springer, pp. 373–384 (cit. on p. 38).

Brechbühler, Ch, Guido Gerig, and Olaf Kübler (1995). "Parametrization of closed surfaces for 3-D shape description." In: *Computer Vision and Image Understanding* 61.2, pp. 154–170 (cit. on p. 63).

Brédif, M., D. Boldo, M. Pierrot-Deseilligny, and H. Maître (2007). "3D building reconstruction with parametric roof superstructures." In: *IEEE International Conference on Image Processing*. IEEE, pp. 537–540 (cit. on pp. 61, 76, 90).

Brédif, Mathieu (May 2010). "3D Building Modeling: Topology-Aware Kinetic Fitting of Polyhedral Roofs and Automatic Roof Superstructure Reconstruction." PhD thesis. Télécom ParisTech (cit. on p. 102).

Breiman, L, JH Friedman, R Olshen, and CJ Stone (1984). "Classification and Regression Trees." In: (cit. on p. 193).

Breiman, Leo (1996). "Bagging predictors." In: *Machine learning* 24.2, pp. 123–140 (cit. on p. 195).

Breiman, Leo (2001). "Random forests." In: *Machine Learning* 45.1, pp. 5–32 (cit. on p. 195).

Bruna, Joan and Stéphane Mallat (2013). "Invariant scattering convolution networks." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8, pp. 1872–1886 (cit. on pp. 124 sqq., 128 sq., 140).

Budroni, Angela and Jan Böhm (2010). "Automatic 3D modelling of indoor manhattan-world scenes from laser data." In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 115–120 (cit. on p. 40).

Castellazzi, Giovanni, Antonio D'Altri, Gabriele Bitelli, Ilenia Selvaggi, and Alessandro Lambertini (2015). "From laser scanning to finite element analysis of complex buildings by using a semi-automatic procedure." In: *Sensors* 15.8, pp. 18360–18380 (cit. on p. 48).

Cazals, Frédéric and Joachim Giesen (2006). "Delaunay triangulation based surface reconstruction." In: *Effective Computational Geometry for Curves and Surfaces*. Springer, pp. 231–276 (cit. on p. 51).

Cham, Tat-Jen, Arridhana Ciptadi, Wei-Chian Tan, Minh-Tri Pham, and Liang-Tien Chia (2010). "Estimating camera pose from a single urban ground-view omnidirectional image and a 2D building outline map." In: *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 366–373 (cit. on p. 38).

Chauve, Anne-Laure, Patrick Labatut, and Jean-Philippe Pons (June 2010). "Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data." In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1261–1268 (cit. on p. 61).

Chen, Liang-Chien, Chia-Hao Wu, Tzu-Sheng Shen, and Chien-Cheng Chou (2014). "The application of geometric network models and building information models in geospatial environments for fire-fighting simulations." In: *Computers, Environment and Urban Systems* 45, pp. 1–12 (cit. on p. 39).

Chen, Yuyu, Avraham Ebenstein, Michael Greenstone, and Hongbin Li (2013). "Evidence on the impact of sustained exposure to air pollution on life expectancy from China's Huai River policy." In: *National Academy of Sciences* 110.32, pp. 12936–12941 (cit. on p. 37).

Christie, Gordon, Garrett Warnell, and Kevin Kochersberger (2016). *Semantics for UGV Registration in GPS-denied Environments.* Tech. rep. arXiv preprint arXiv:1609.04794 (cit. on p. 38).

Cohen, Taco and Max Welling (2016). "Group equivariant convolutional networks." In: *International conference on machine learning*, pp. 2990–2999 (cit. on pp. 128 sq.).

Colomb, Maxime, Mickaël Brasebin, Julien Perret, and Cécile Tannier (Sept. 2017). "Simulation of a realistic residential development with the integration of two existing models." In: *European Colloquium on Theoretical and Quantitative Geography* (cit. on p. 36).

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks." In: *Machine learning* 20.3, pp. 273–297 (cit. on p. 187).

Dalal, Navneet and Bill Triggs (June 2005). "Histograms of Oriented Gradients for Human Detection." In: *International Conference on Computer Vision & Pattern Recognition (CVPR '05).* Ed. by Cordelia Schmid, Stefano Soatto, and Carlo Tomasi. Vol. 1. San Diego, United States: IEEE Computer Society, pp. 886–893 (cit. on p. 91).

Damiand, Guillaume and Pascal Lienhardt (2014). *Combinatorial maps: efficient data structures for computer graphics and image processing.* CRC Press (cit. on p. 50).

Demir, Ilke, Daniel G. Aliaga, and Bedrich Benes (2015). "Procedural editing of 3d building point clouds." In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2147–2155 (cit. on pp. 48, 61).

Deng, Yichuan, Jack CP Cheng, and Chimay Anumba (2016). "Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison." In: *Automation in Construction* 67, pp. 1–21 (cit. on p. 45).

Devaux, A., C. Hoarau, M. Brédif, and S. Christophe (2018). "3D URBAN GEOVISUALIZATION: IN SITU AUGMENTED AND MIXED REALITY EXPERIMENTS." In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-4 (cit. on pp. 39, 43).

Devaux, Alexandre, Mathieu Brédif, and Nicolas Paparoditis (2012). "A web-based 3d mapping application using webgl allowing interaction with images, point clouds and models." In: *International Conference on Advances in Geographic Information Systems.* ACM, pp. 586–588 (cit. on p. 43).

Diakité, Abdoulaye Abou, Guillaume Damiand, and Dirk Van Maercke (2014). "Topological reconstruction of complex 3D buildings and automatic extraction of levels of detail." In: *Eurographics Workshop on Urban Data Modelling and Visualisation.* Eurographics Association, pp. 25–30 (cit. on p. 50).

Dick, Anthony R, Philip H.S. Torr, and Roberto Cipolla (2004). "Modelling and interpretation of architecture from several images." In: *International Journal of Computer Vision* 60.2, pp. 111–134 (cit. on p. 65).

Dimitropoulos, Kosmas, K Köse, Nikos Grammalidis, and Enis Cetin (2010). "Fire detection and 3D fire propagation estimation for the protection of cultural heritage areas." In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 38, pp. 620–625 (cit. on p. 37).

Duan, Liuyun and Florent Lafarge (2016). "Towards large-scale city reconstruction from satellites." In: *European Conference on Computer Vision (ECCV).* Springer, pp. 89–104 (cit. on p. 63).

Durupt, M and F Taillandier (2006). “Automatic building reconstruction from a Digital Elevation Model and cadastral data: an operational approach.” In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXVI.3, pp. 142–147 (cit. on pp. 31, 60, 64, 71, 73 sq., 99, 102).

Eickenberg, Michael, Georgios Exarchakis, Matthew Hirn, Stéphane Mallat, and Louis Thiry (2018). “Solid harmonic wavelet scattering for predictions of molecule properties.” In: *The Journal of chemical physics* 148.24, p. 241732 (cit. on p. 140).

Elberink, Sander Oude and George Vosselman (2011). “Quality analysis on 3D building models reconstructed from airborne laser scanning data.” In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.2, pp. 157–165 (cit. on pp. 62 sq.).

Ennafii, Oussama, Arnaud Le-Bris, Florent Lafarge, and Clément Mallet (Sept. 2018a). *Semantic evaluation of 3D city models*. Tech. rep. (cit. on p. 80).

Ennafii, Oussama, Arnaud Le Bris, Florent Lafarge, and Clément Mallet (June 2018b). “Qualification sémantique de modèles 3D de bâtiments.” In: *Conférence Française de Photogrammétrie et de Télédétection (CFPT)* (cit. on p. 95).

Fabri, Andreas, Geert-Jan Giezeman, Lutz Kettner, Stefan Schirra, and Sven Schönherr (2000). “On the design of CGAL a computational geometry algorithms library.” In: *Software: Practice and Experience* 30.11, pp. 1167–1202 (cit. on pp. 31, 98).

Feragen, Aasa, Niklas Kasenburg, Jens Petersen, Marleen de Bruijne, and Karsten Borgwardt (2013). “Scalable kernels for graphs with continuous attributes.” In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 216–224 (cit. on pp. 137 sq.).

Fisher, Ronald A (1936). “The use of multiple measurements in taxonomic problems.” In: *Annals of eugenics* 7.2, pp. 179–188 (cit. on p. 191).

Gärtner, Thomas, Peter Flach, and Stefan Wrobel (2003). “On graph kernels: Hardness results and efficient alternatives.” In: *Learning theory and kernel machines*. Springer, pp. 129–143 (cit. on p. 133).

Ghosh, Swarnendu, Nibaran Das, Teresa Gonçalves, Paulo Quaresma, and Mahantapas Kundu (2018). “The journey of graph kernels through two decades.” In: *Computer Science Review* 27, pp. 88–111 (cit. on p. 130).

Gilbert, Ben (Dec. 2018). *I’m blown away by the virtual New York City of ‘Spider-Man’ on PlayStation 4 — here’s how it compares to the real thing.* `https://www.businessinsider.nl/spider-man-ps4-new-york-city-2018-9` (cit. on p. 38).

Gorszczyk, Benjamin, Guillaume Damiand, Sylvie Servigne, Abdoulaye Abou Diakite, and Gilles Gesquière (Dec. 2016). “An Automatic Comparison Approach to Detect Errors on 3D City Models.” In: *Eurographics Workshop on Urban Data Modelling and Visualisation.* Ed. by The Eurographics Association. Proceedings. Liège, Belgium, pp. 25–30 (cit. on p. 50).

Gröger, Gerhard, Thomas H. Kolbe, Claus Nagel, and Karl-Heinz Häfele, eds. (2012a). *OGC City Geography Markup Language (CityGML) Encoding Standard*. 2nd ed. OpenGIS® Encoding Standard. Open Geospatial Consortium (cit. on p. 50).

Gröger, Gerhard and Lutz Plümer (2012b). “CityGML–Interoperable semantic 3D city models.” In: *ISPRS Journal of Photogrammetry and Remote Sensing* 71, pp. 12–33 (cit. on pp. 27, 46).

Gröger, Gerhard and Lutz Plümer (2011). “How to achieve consistency for 3D city models.” In: *GeoInformatica* 15.1, pp. 137–165 (cit. on p. 50).

Gupta, Agrim, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi (2018). “Social gan: Socially acceptable trajectories with generative adversarial networks.” In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, pp. 2255–2264 (cit. on p. 38).

Guyon, Isabelle, Jason Weston, Stephen Barnhill, and Vladimir Vapnik (Jan. 2002). "Gene Selection for Cancer Classification using Support Vector Machines." In: *Machine Learning* 46.1, pp. 389–422 (cit. on p. 154).

Hammack, Richard, Wilfried Imrich, and Sandi Klavžar (2011). *Handbook of product graphs*. CRC press (cit. on p. 132).

Harrach, Mouna, Alexandre Devaux, and Mathieu Brédif (2019). "Interactive image geolocalization in an immersive web application." In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII.2/W9, pp. 377–380 (cit. on p. 43).

Haussler, David (1999). *Convolution kernels on discrete structures*. Tech. rep. Department of Computer Science, University of California at Santa Cruz (cit. on p. 137).

Henricsson, O. and E. Baltsavias (1997). "3-D Building reconstruction with ARUBA: a qualitative and quantitative evaluation." In: *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pp. 65–76 (cit. on pp. 63, 65).

Hofer, Manuel, Michael Maurer, and Horst Bischof (2017). "Efficient 3D scene abstraction using line segments." In: *Computer Vision and Image Understanding* 157, pp. 167–178 (cit. on p. 75).

Holzmann, Thomas, Michael Maurer, Friedrich Fraundorfer, and Horst Bischof (Sept. 2018). "Semantically Aware Urban 3D Reconstruction with Plane-Based Regularization." In: *European Conference on Computer Vision (ECCV)* (cit. on p. 61).

Horna, Sébastien, Guillaume Damiand, Daniel Meneveaux, and Yves Bertrand (Mar. 2007). "Building 3D indoor scenes topology from 2D architectural plans." In: *GRAPP*. Proc. of 2nd International Conference on Computer Graphics Theory and Applications. Barcelona, Spain, pp. 37–44 (cit. on p. 60).

Horváth, Tamás, Thomas Gärtner, and Stefan Wrobel (2004). "Cyclic pattern kernels for predictive graph mining." In: *tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 158–167 (cit. on p. 133).

Huck, Andreas and Jochen Monstadt (2019). "Urban and infrastructure resilience: Diverging concepts and the need for cross-boundary learning." In: *Environmental Science & Policy* 100, pp. 211–220 (cit. on p. 37).

Iovan, Corina, Didier Boldo, and Matthieu Cord (2008). "Detection, characterization, and modeling vegetation in urban areas from high-resolution aerial imagery." In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 1.3, pp. 206–213 (cit. on pp. 42, 44).

Jamet, Olivier, Olivier Dissard, and Sylvain Airault (1995). "Building extraction from stereo pairs of aerial images: accuracy and productivity constraint of a topographic production line." In: *Automatic Extraction of Man-Made Objects from Aerial and Space Images*. Springer, pp. 231–240 (cit. on p. 48).

Jaynes, Christopher, Edward Riseman, and Allen Hanson (2003). "Recognition and reconstruction of buildings from multiple aerial images." In: *Computer Vision and Image Understanding* 90.1, pp. 68–98 (cit. on pp. 62 sq., 65).

Jethava, Vinay, Anders Martinsson, Chiranjib Bhattacharyya, and Devdatt Dubhashi (2013). "Lovász ϑ function, SVMs and finding dense subgraphs." In: *The Journal of Machine Learning Research* 14.1, pp. 3495–3536 (cit. on p. 134).

Johansson, Fredrik, Vinay Jethava, Devdatt Dubhashi, and Chiranjib Bhattacharyya (2014). "Global graph kernels using geometric embeddings." In: *31st International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014* (cit. on p. 134).

Kaartinen, H, J Hyyppä, E Gülch, G Vosselman, H Hyyppä, et al. (2005). "Accuracy of 3D city models: EuroSDR comparison." In: *International archives of photogrammetry,* *remote sensing and spatial information sciences* XXXVI.3/W19, pp. 227–232 (cit. on pp. 48, 62 sq., 65).

Kedzierski, Michal and Anna Fryskowska (2014). “Terrestrial and aerial laser scanning data integration using wavelet analysis for the purpose of 3D building modeling.” In: *Sensors* 14.7, pp. 12070–12092 (cit. on p. 47).

Kim, Young Min, Sangwoo Ryu, and Ig-Jae Kim (2019). “Planar Abstraction and Inverse Rendering of 3D Indoor Environment.” In: *IEEE Transactions on Visualization and Computer Graphics*. Ed. by Paolo Cignoni and Eder Miguel. The Eurographics Association, pp. 81–84 (cit. on p. 39).

Kolbe, Thomas H, Gerhard Gröger, and Lutz Plümer (2005). “CityGML: Interoperable access to 3D city models.” In: *Geo-information for Disaster Management*. Springer, pp. 883–899 (cit. on pp. 26, 45).

Kondor, Risi and Horace Pan (2016). “The multiscale laplacian graph kernel.” In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2990–2998 (cit. on pp. 135 sq., 144).

Koutsoudis, Anestis, Fotis Arnaoutoglou, and Christodoulos Chamzas (2007). “On 3D reconstruction of the old city of Xanthi. A minimum budget approach to virtual touring based on photogrammetry.” In: *Journal of Cultural Heritage* 8.1, pp. 26–31 (cit. on p. 39).

Koutsourakis, Panagiotis, Loic Simon, Olivier Teboul, Georgios Tziritas, and Nikos Paragios (2009). “Single view reconstruction using shape grammars for urban environments.” In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, pp. 1795–1802 (cit. on p. 61).

Kovashka, Adriana, Olga Russakovsky, Li Fei-Fei, Kristen Grauman, et al. (2016). “Crowdsourcing in computer vision.” In: *Foundations and Trends® in Computer Graphics and Vision* 10.3, pp. 177–243 (cit. on p. 54).

Kowdle, Adarsh, Yao-Jen Chang, Andrew Gallagher, and Tsuhan Chen (2011). “Active learning for piecewise planar 3D reconstruction.” In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 929–936 (cit. on pp. 53 sq., 88).

Kriege, Nils M, Fredrik D Johansson, and Christopher Morris (2020). “A survey on graph kernels.” In: *Applied Network Science* 5.1, pp. 1–42 (cit. on p. 130).

Kurakula, Vinaykumar, AK Skidmore, H Kluijver, J Stoter, K Dabrowska-Zielinska, et al. (2007). “A GIS based approach for 3D noise modelling using 3D city models.” MA thesis. Enschede, The Netherlands: International Institute for Geo-information Science and Earth Observation (cit. on p. 40).

Kwan, Mei-Po and Jiyeong Lee (2005). “Emergency response after 9/11: the potential of real-time 3D GIS for quick emergency response in micro-spatial environments.” In: *Computers, Environment and Urban Systems* 29.2, pp. 93–113 (cit. on p. 39).

Lafarge, Florent (2015). “Some new research directions to explore in urban reconstruction.” In: *Joint Urban Remote Sensing Event (JURSE)*. IEEE, pp. 1–4 (cit. on pp. 26, 46 sq.).

Lafarge, Florent, Xavier Descombes, Josiane Zerubia, and Marc Pierrot-Deseilligny (2008). “Structural approach for building reconstruction from a single DSM.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.1, pp. 135–147 (cit. on pp. 61, 75).

Lafarge, Florent and Clement Mallet (2012). “Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation.” In: *International Journal of Computer Vision* 99.1, pp. 69–85 (cit. on pp. 26, 41, 46, 48 sq., 60 sqq., 65, 70, 91).

Lanckriet, Gert RG, Tijl De Bie, Nello Cristianini, Michael I Jordan, and William Stafford Noble (2004). “A statistical framework for genomic data fusion.” In: *Bioinformatics* 20.16, pp. 2626–2635 (cit. on p. 190).

Landes, Tania, Hakim Boulaassal, and Pierre Grussenmeyer (2012). “Quality assessment of geometric façade models reconstructed from TLS data.” In: *The Photogrammetric Record* 27.138, pp. 137–154 (cit. on pp. 62 sq.).

Langlois, Pierre-Alain, Alexandre Boulch, and Renaud Marlet (Sept. 2019). “Surface Reconstruction from 3D Line Segments.” In: *International Conference on 3D Vision (3DV)* (cit. on p. 75).

Ledoux, Hugo (2013). “On the validation of solids represented with the international standards for geographic information.” In: *Computer-Aided Civil and Infrastructure Engineering* 28.9, pp. 693–706 (cit. on pp. 27, 50).

Ledoux, Hugo (2018). “val3dity: validation of 3D GIS primitives according to the international standards.” In: *Open Geospatial Data, Software and Standards* 3.1, p. 1 (cit. on pp. 50, 69).

Ledoux, Hugo and Martijn Meijers (2011). “Topologically consistent 3D city models obtained by extrusion.” In: *International Journal of Geographical Information Science* 25.4, pp. 557–574 (cit. on pp. 60 sq.).

Lee, Honglak, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng (2009). “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations.” In: *26th annual international conference on machine learning*. ACM, pp. 609–616 (cit. on p. 127).

Li, Minglei, Peter Wonka, and Liangliang Nan (2016). “Manhattan-world urban reconstruction from point clouds.” In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 54–69 (cit. on pp. 48, 55, 61, 78, 93).

Løvås, Gunnar G (1994). “Modeling and simulation of pedestrian traffic flow.” In: *Transportation Research Part B: Methodological* 28.6, pp. 429–443 (cit. on p. 41).

Lovász, László (1979). “On the Shannon capacity of a graph.” In: *IEEE Transactions on Information theory* 25.1, pp. 1–7 (cit. on p. 134).

Lowe, David G (2004). “Distinctive image features from scale-invariant keypoints.” In: *International Journal of Computer Vision* 60.2, pp. 91–110 (cit. on pp. 91, 124).

Ludlow, David (2006). “Urban sprawl in Europe: The ignored challenge.” In: (cit. on p. 36).

Mahé, Pierre, Nobuhisa Ueda, Tatsuya Akutsu, Jean-Luc Perret, and Jean-Philippe Vert (2004). “Extensions of marginalized graph kernels.” In: *twenty-first international conference on Machine learning*. ACM, p. 70 (cit. on p. 133).

Mallat, Stéphane (2012). “Group invariant scattering.” In: *Communications on Pure and Applied Mathematics* 65.10, pp. 1331–1398 (cit. on pp. 32, 124 sq., 128 sq., 140).

Martinovic, Andelo and Luc Van Gool (2013). “Bayesian grammar learning for inverse procedural modeling.” In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 201–208 (cit. on p. 61).

Mathias, Markus, Andelo Martinovic, Julien Weissenberg, and Luc Van Gool (2011). “Procedural 3D building reconstruction using shape grammars and detectors.” In: *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*. IEEE, pp. 304–311 (cit. on p. 61).

Mayunga, Selassie, Yun Zhang, and David Coleman (2005). “Semi-automatic building extraction utilizing Quickbird imagery.” In: XXXVI.3/W24, pp. 1–136 (cit. on p. 48).

McWhertor, Michael (Sept. 2013). *Under the hood of Infamous: Second Son’s hyper-real Seattle*. `https://www.polygon.com/2013/9/25/4702318/under-the-hood-of-infamous-second-son-hyper-real-seattle` (cit. on p. 38).

Mezian, Miloud, Bruno Vallet, Bahman Soheilian, and Nicolas Paparoditis (2016). “Uncertainty propagation for terrestrial mobile laser scanner.” In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 41, pp. 331–335 (cit. on p. 47).

Michelin, J.-C., J. Tierny, F. Tupin, C. Mallet, and N. Paparoditis (2013). "Quality evaluation of 3D city building models with automatic error diagnosis." In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL.7/W2, pp. 161–166 (cit. on pp. 28, 63 sqq., 76, 81, 88, 91).

Michelin, Jean-Christophe, Clément Mallet, and Nicolas David (2012). "Building edge detection using small-footprint airborne full-waveform lidar data." In: I.3 (cit. on p. 60).

Mnih, Volodymyr and Geoffrey E. Hinton (2010). "Learning to detect roads in high-resolution aerial images." In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 210–223 (cit. on p. 42).

Mohamed, Mostafa (2013). "Quality assessment of 3D building models in airborne digital photogrammetry." PhD thesis. Université de Strasbourg (cit. on p. 63).

Monnier, Fabrice (2014). "Amélioration de la localisation 3D de véhicules mobiles à l'aide de cartes ou modèles 3D." PhD thesis. Université Paris-Est (cit. on p. 48).

Monnier, Fabrice, Bruno Vallet, Nicolas Paparoditis, Jean-Pierre Papelard, and Nicolas David (Oct. 2013). "Registration of terrestrial mobile laser data on 2D or 3D geographic database by use of a non-rigid ICP approach." In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* II.5/W2, pp. 193–198 (cit. on p. 47).

Mooney, Peter, Padraig Corcoran, and Adam C Winstanley (2010). "Towards quality metrics for OpenStreetMap." In: *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS)*. ACM, pp. 514–517 (cit. on p. 51).

Musialski, Przemyslaw, Peter Wonka, Daniel G. Aliaga, Michael Wimmer, Luc Van Gool, et al. (2013). "A survey of urban reconstruction." In: *Computer graphics forum*. Vol. 32. 6. Wiley Online Library, pp. 146–177 (cit. on pp. 26, 45 sq., 48 sq., 60).

Nagel, Claus, Alexandra Stadler, and Thomas H. Kolbe (2009). "Conceptual requirements for the automatic reconstruction of building information models from uninterpreted 3D models." In: *Academic Track of the Geoweb*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. ISPRS, pp. 46–53 (cit. on p. 46).

Nan, Liangliang, Caigui Jiang, Bernard Ghanem, and Peter Wonka (2015). "Template assembly for detailed urban reconstruction." In: *Computer Graphics Forum*. Vol. 34. 2. Wiley Online Library, pp. 217–228 (cit. on pp. 61, 72).

Nan, Liangliang and Peter Wonka (2017). "Polyfit: Polygonal surface reconstruction from point clouds." In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2353–2361 (cit. on pp. 48, 61, 74, 78).

Neis, Pascal, Marcus Goetz, and Alexander Zipf (2012). "Towards automatic vandalism detection in OpenStreetMap." In: *ISPRS International Journal of Geo-Information* 1.3, pp. 315–332 (cit. on p. 55).

Neumann, Marion, Roman Garnett, Christian Bauckhage, and Kristian Kersting (2016). "Propagation kernels: efficient graph kernels from propagated information." In: *Machine Learning* 102.2, pp. 209–245 (cit. on pp. 136 sq.).

Nguatem, William and Helmut Mayer (2017). "Modeling Urban Scenes From Pointclouds." In: *IEEE Internatinal Conference on Computer Vision (ICCV)*. IEEE, pp. 3837–3846 (cit. on pp. 60, 63).

OECD.stat (Mar. 2020). *Land cover in Functional Urban Areas*. `https://stats.oecd.org/Index.aspx?DataSetCode=LAND_COVER_FUA` (cit. on p. 42).

Omasa, Kenji, Fumiki Hosoi, and Atsumi Konishi (2006). "3D lidar imaging for detecting and understanding plant responses and canopy structure." In: *Journal of Experimental Botany* 58.4, pp. 881–898 (cit. on p. 41).

Ortner, Mathias, Xavier Descombes, and Josiane Zerubia (2007). "Building outline extraction from Digital Elevation Models using marked point processes." In: *International Journal of Computer Vision* 72.2, pp. 107–132 (cit. on p. 91).

Oyallon, Edouard and Stéphane Mallat (2015). "Deep roto-translation scattering for object classification." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2865–2873 (cit. on pp. 32, 126, 128 sq.).

Pascal, Mathilde, Magali Corso, Olivier Chanel, Christophe Declercq, Chiara Badaloni, et al. (2013). "Assessing the public health impacts of urban air pollution in 25 European cities: results of the Aphekom project." In: *Science of the Total Environment* 449, pp. 390–400 (cit. on p. 37).

Pătrăucean, Viorica, Iro Armeni, Mohammad Nahangi, Jamie Yeung, Ioannis Brilakis, et al. (2015). "State of research in automatic as-built modelling." In: *Advanced Engineering Informatics* 29.2, pp. 162–171 (cit. on p. 44).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, et al. (2011). "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12, pp. 2825–2830 (cit. on p. 95).

Piasco, Nathan, Désiré Sidibé, Cédric Demonceaux, and Valérie Gouet-Brunet (2018). "A survey on visual-based localization: On the benefit of heterogeneous data." In: *Pattern Recognition* 74, pp. 90–109 (cit. on p. 38).

Plante, Chris (June 2013). *How Spider-Man PS4's New York City compares to the real thing.* `https://www.polygon.com/e3/2018/6/12/17453588/spider-man-ps4-new-york-city-avengers-demo-preview` (cit. on p. 38).

Platt, John (Apr. 1998). "Sequential minimal optimization: A fast algorithm for training support vector machines." In: *Advances in Kernel Methods: Support Vector Learning*, p. 21 (cit. on p. 188).

Poullis, Charalambos (2013). "A Framework for Automatic Modeling from Point Cloud Data." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.11, pp. 2563–2575 (cit. on pp. 61, 91).

Poullis, Charalambos and Suya You (2009). "Automatic reconstruction of cities from remote sensor data." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2775–2782 (cit. on p. 60).

Powers, David (2011). "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." In: *International Journal of Machine Learning Technology* 2.1, pp. 37–63 (cit. on p. 96).

Previtali, M., Luigi Barazzetti, R. Brumana, Branka Cuca, Daniela Oreni, et al. (2014). "Automatic façade modelling using point cloud data for energy-efficient retrofitting." In: *Applied Geomatics* 6.2, pp. 95–113 (cit. on pp. 26, 37).

Rakotomamonjy, Alain, Francis R Bach, Stéphane Canu, and Yves Grandvalet (2008). "SimpleMKL." In: *Journal of Machine Learning Research* 9.Nov, pp. 2491–2521 (cit. on p. 190).

Rau, Jiann-Yeou, Liang-Chien Chen, Fuan Tsai, Kuo-Hsin Hsiao, and Wei-Chen Hsu (2006). "Lod generation for 3d polyhedral building model." In: *Pacific-Rim Symposium on Image and Video Technology.* Springer, pp. 44–53 (cit. on p. 46).

Redweik, Paula, Cristina Catita, and Miguel Brito (2013). "Solar energy potential on roofs and facades in an urban landscape." In: *Solar Energy* 97, pp. 332–341 (cit. on p. 41).

Ref3DNat®: Spécifications du noyau du référentiel (Sept. 2017). 1.0.2. Institut national de l'information géographique et forestière (IGN). 73 avenue de Paris, 94165 Saint-Mandé CEDEX (cit. on p. 44).

Rottensteiner, Franz, Gunho Sohn, Markus Gerke, Jan Dirk Wegner, Uwe Breitkopf, et al. (2014). "Results of the ISPRS benchmark on urban object detection and 3D building reconstruction." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 93, pp. 256–271 (cit. on pp. 51, 63 sq.).

Rottensteiner, Franz, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, et al. (2012). "The ISPRS benchmark on urban object classification and 3D building reconstruction." In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* I-3.1, pp. 293–298 (cit. on pp. 51 sq., 81, 179).

Rüppel, Uwe and Kristian Schatz (2011). "Designing a BIM-based serious game for fire safety evacuation simulations." In: *Advanced Engineering Informatics* 25.4, pp. 600–611 (cit. on p. 39).

Russell, Bryan C, Josef Sivic, Jean Ponce, and Helene Dessales (2011). "Automatic alignment of paintings and photographs depicting a 3D scene." In: *IEEE International Conference on Computer Vision (ICCV) Workshops (ICCV Workshops).* IEEE, pp. 545–552 (cit. on p. 39).

Rüther, Heinz, Hagai M Martine, and EG Mtalo (2002). "Application of snakes and dynamic programming optimisation technique in modeling of buildings in informal settlement areas." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 56.4, pp. 269–282 (cit. on p. 48).

Schuster, H.-F. and U. Weidner (2003). "A new approach towards quantitative quality evaluation of 3D building models." In: *ISPRS Commission IV Joint Workshop on Challenges in Geospatial Analysis, Stuttgart, Germany*, pp. 614–629 (cit. on pp. 63, 65).

Shao, Wei and Demetri Terzopoulos (2007). "Autonomous pedestrians." In: *Graphical Models* 69.5-6, pp. 246–274 (cit. on p. 41).

Shawe-Taylor, John, Nello Cristianini, et al. (2004). *Kernel methods for pattern analysis.* Cambridge university press (cit. on p. 189).

Shervashidze, Nino, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt (2011). "Weisfeiler-lehman graph kernels." In: *Journal of Machine Learning Research* 12.Sep, pp. 2539–2561 (cit. on pp. 131, 136 sq.).

Sifre, Laurent and Stéphane Mallat (2013). "Rotation, scaling and deformation invariant scattering for texture discrimination." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, pp. 1233–1240 (cit. on pp. 32, 124 sqq., 140).

Siglidis, Giannis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, et al. (2018). *Grakel: A graph kernel library in python.* Tech. rep. arXiv preprint arXiv:1806.02193 (cit. on p. 144).

Simon, Loic, Olivier Teboul, Panagiotis Koutsourakis, and Nikos Paragios (2011). "Random exploration of the procedural space for single-view 3d modeling of buildings." In: *International Journal of Computer Vision* 93.2, pp. 253–271 (cit. on p. 61).

Sinha, Sudipta, Drew Steedly, and Rick Szeliski (Sept. 2009). "Piecewise planar stereo for image-based rendering." In: *Twelfth IEEE International Conference on Computer Vision (ICCV) (ICCV 2009).* IEEE (cit. on p. 74).

Soheilian, Bahman, Nicolas Paparoditis, and Bruno Vallet (2013). "Detection and 3D reconstruction of traffic signs from multiple view color images." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 77, pp. 1–20 (cit. on pp. 42, 45).

Stoter, Jantien, Henk De Kluijver, and Vinaykumar Kurakula (2008). "3D noise mapping in urban areas." In: *International Journal of Geographical Information Science* 22.8, pp. 907–924 (cit. on p. 37).

Stoter, JE, GAK Arroyo Ohori, and Hugo Ledoux (2018). "Geo-BIM data integration: easier said than done?" In: *Geospatial World* 9.4 (cit. on p. 45).

Sun, Zhaonan, Nawanol Ampornpunt, Manik Varma, and Svn Vishwanathan (2010). "Multiple kernel learning and the SMO algorithm." In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2361–2369 (cit. on p. 190).

Taillandier, Franck and Rachid Deriche (2004). “Automatic buildings reconstruction from aerial images: a generic Bayesian framework.” In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXV.3A (cit. on pp. 60 sq., 74).

Taneja, Aparna, Luca Ballan, and Marc Pollefeys (2013). “City-scale change detection in cadastral 3d models using images.” In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 113–120 (cit. on p. 54).

Tannier, Cécile, Jean-Christophe Foltête, and Xavier Girardet (2012). “Assessing the capacity of different urban forms to preserve the connectivity of ecological habitats.” In: *Landscape and Urban Planning* 105.1, pp. 128–139 (cit. on p. 36).

Tavani, Herman T (2011). *Ethics and technology: Controversies, questions, and strategies for ethical computing*. John Wiley & Sons (cit. on p. 41).

Thornton, James (2010). “Individual privacy rights with respect to services such as Google Street View.” In: *ACM SIGCAS Computers and Society* 40.4, pp. 70–76 (cit. on p. 41).

Tran, H., K. Khoshelham, and A. Kealy (2019). “Geometric comparison and quality evaluation of 3D models of indoor environments.” In: *ISPRS Journal of Photogrammetry and Remote Sensing* 149, pp. 29–39 (cit. on p. 62).

Uden, Matthias and Alexander Zipf (2013). “Open building models: Towards a platform for crowdsourcing virtual 3D cities.” In: *Progress and New Trends in 3D Geoinformation Sciences*. Springer, pp. 299–314 (cit. on p. 45).

Ujang, Uznir, François Anton, and Alias Abdul Rahman (2013). “Unified Data Model of Urban Air Pollution Dispersion and 3D Spatial City Models: Groundwork Assessment towards Sustainable Urban Development for Malaysia.” In: *Journal of Environmental Protection* 4.7, pp. 701–712 (cit. on p. 38).

Vanegas, Carlos A., Daniel G. Aliaga, and Bedřich Beneš (2010). “Building reconstruction using manhattan-world grammars.” In: *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 358–365 (cit. on pp. 48, 61).

Vanhoey, Kenneth, Carlos Eduardo Porto de Oliveira, Hayko Riemenschneider, András Bódis-Szomorú, Santiago Manén, et al. (2017). “VarCity - the Video: The Struggles and Triumphs of Leveraging Fundamental Research Results in a Graphics Video Production.” In: *ACM SIGGRAPH 2017 Talks*. SIGGRAPH ’17. Los Angeles, California: ACM, 48:1–48:2 (cit. on pp. 37, 41).

Vapnik, Vladimir (2013). *The nature of statistical learning theory*. Springer science & business media (cit. on p. 189).

Varduhn, Vasco, Ralf-Peter Mundani, and Ernst Rank (2015). “Multi-resolution models: Recent progress in coupling 3D geometry to environmental numerical simulation.” In: *3D Geoinformation Sciences*. Springer, pp. 55–69 (cit. on p. 37).

Varma, Manik and Bodla Rakesh Babu (2009). “More generality in efficient multiple kernel learning.” In: *26th Annual International Conference on Machine Learning*. ACM, pp. 1065–1072 (cit. on p. 190).

Verdie, Yannick, Florent Lafarge, and Pierre Alliez (2015). “Lod generation for urban scenes.” In: *ACM Transactions on Graphics (ToG)* 34.ARTICLE, p. 30 (cit. on pp. 48, 61, 76).

Verma, Vivek, Rakesh Kumar, and Stephen Hsu (2006). “3D building detection and modeling from aerial LIDAR data.” In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2213–2220 (cit. on pp. 77, 89).

Vishwanathan, S Vichy N, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt (2010). “Graph kernels.” In: *Journal of Machine Learning Research* 11.Apr, pp. 1201–1242 (cit. on pp. 131 sqq., 137).

Vögtle, T and E Steinle (2003). “On the quality of object classification and automated building modeling based on laserscanning data.” In: *The International Archives of the*

Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIV.3/W13, pp. 149–155 (cit. on pp. 62, 65).

Wate, Parag and Volker Coors (2015). “3D Data Models for Urban Energy Simulation.” In: *Energy Procedia* 78. 6th International Building Physics Conference, IBPC 2015, pp. 3372–3377 (cit. on pp. 26, 37).

Watson, Benjamin, Pascal Müller, Oleg Veryovka, Andy Fuller, Peter Wonka, et al. (2008). “Procedural urban modeling in practice.” In: *IEEE Computer Graphics and Applications* 28.3, pp. 18–26 (cit. on p. 38).

Werner, Tomás and Andrew Zisserman (2002). “New techniques for automated architectural reconstruction from photographs.” In: *European Conference on Computer Vision (ECCV).* Springer, pp. 541–555 (cit. on pp. 61, 72).

Willenborg, Bruno (2015). “Simulation of explosions in urban space and result analysis based on CityGML-City Models and a cloud-based 3D-Webclient.” MA thesis. Munich, Germany: Technische Universität München (cit. on p. 40).

Wolff, Markus and Hartmut Asche (2008). “Geospatial modelling of urban security: A novel approach with virtual 3D city models.” In: *International Conference on Computational Science and Its Applications.* Springer, pp. 42–51 (cit. on p. 40).

Wolff, Markus and Hartmut Asche (2009). “Towards geovisual analysis of crime scenes–a 3D crime mapping approach.” In: *Advances in GIScience.* Springer, pp. 429–448 (cit. on p. 40).

Wu, Huayi, Zhengwei He, and Jianya Gong (2010). “A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes.” In: *Computers, Environment and Urban Systems* 34.4. Geospatial Cyberinfrastructure, pp. 291–298 (cit. on p. 37).

Xiong, Biao, Sander Oude Elberink, and George Vosselman (2014). “A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds.” In: *ISPRS Journal of Photogrammetry and Remote Sensing* 93, pp. 227–242 (cit. on pp. 73, 81).

Yan, Wai Yeung, Ahmed Shaker, Ayman Habib, and Ana Paula Kersting (2012). “Improving classification accuracy of airborne LiDAR intensity data by geometric calibration and radiometric correction.” In: *ISPRS Journal of Photogrammetry and Remote Sensing* 67, pp. 35–44 (cit. on p. 47).

You, Rey-Jer and Bo-Cheng Lin (2011). “A quality prediction method for building model reconstruction using LiDAR data and topographic maps.” In: *IEEE Transactions on Geoscience and Remote Sensing* 49.9, pp. 3471–3480 (cit. on p. 62).

Yun, Zhengqing, Magdy F Iskander, Soo Yong Lim, Donya He, and Ralph Martinez (2007). “Radio wave propagation prediction based on 3-D building structures extracted from 2-D images.” In: *IEEE Antennas and Wireless Propagation Letters* 6, pp. 557–559 (cit. on p. 37).

Zebedin, Lukas, Joachim Bauer, Konrad Karner, and Horst Bischof (2008). “Fusion of feature-and area-based information for urban buildings modeling from aerial imagery.” In: *European Conference on Computer Vision (ECCV).* Springer, pp. 873–886 (cit. on pp. 63, 65, 74, 90).

Zeng, Chuiqing, Ting Zhao, and Jinfei Wang (2014). “A multicriteria evaluation method for 3-D building reconstruction.” In: *IEEE Geoscience and Remote Sensing Letters* 11.9, pp. 1619–1623 (cit. on pp. 28, 62 sq.).

Zeng, Huayi, Jiaye Wu, and Yasutaka Furukawa (2018). “Neural procedural reconstruction for residential buildings.” In: *European Conference on Computer Vision (ECCV).* Springer, pp. 737–753 (cit. on pp. 48, 61, 63, 66).

Zhang, Liming, Tao Qian, and Q Zeng (2007). “The Radon measure formulation for edge detection using rotational wavelets.” In: *Commun. Pure Appl. Anal* 6.3, pp. 899–915 (cit. on p. 141).

Zhang, Liqiang and Liang Zhang (2017). “Deep learning-based classification and reconstruction of residential scenes from large-scale point clouds.” In: *IEEE Transactions on Geoscience and Remote Sensing* 56.4, pp. 1887–1897 (cit. on p. 63).

Zhao, Z, Hugo Ledoux, and JE Stoter (2013). “Automatic repair of CityGML LOD2 buildings using shrink-wrapping.” In: II-2.W1, pp. 309–317 (cit. on p. 50).

Zhou, Qian-Yi and Ulrich Neumann (2010). “2.5 d dual contouring: A robust approach to creating building models from aerial lidar point clouds.” In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 115–128 (cit. on pp. 48, 61, 90).

Zhu, Lingjie, Shuhan Shen, Xiang Gao, and Zhanyi Hu (2018). “Large Scale Urban Scene Modeling from MVS Meshes.” In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 614–629 (cit. on pp. 60 sq., 63, 65).

Zlatanova, Sisi, Alias Abdul Rahman, and Morakot Pilouk (2002). “Trends in 3D GIS development.” In: *Journal of Geospatial Engineering* 4.2, pp. 71–80 (cit. on p. 40).

Abstract

The automatic generation of 3D building models from geospatial data is now a standard procedure. An abundant literature covers the last two decades and several softwares are now available. However, urban areas are very complex environments. Inevitably, practitioners still have to visually assess, at city-scale, the correctness of these models and detect frequent reconstruction errors. Such a process relies on experts, and is highly time-consuming with approximately 2 h/km^2/expert. This work proposes an approach for automatically evaluating the quality of 3D building models. Potential errors are compiled in a novel hierarchical and modular taxonomy. This allows, for the first time, to disentangle fidelity and modeling errors, whatever the level of details of the modeled buildings. The quality of models is predicted using the geometric properties of buildings and, when available, Very High Resolution images and Digital Surface Models. A baseline of handcrafted, yet generic, features is fed into a Random Forest or Support Vector Machine classifiers. Advanced features, relying on graph kernels as well as Scattering Networks, were proposed to better take into consideration structure. Both multi-class and multi-label cases are studied: due to the interdependence between classes of errors, it is possible to retrieve all errors at the same time while simply predicting correct and erroneous buildings. The proposed framework was tested on three distinct urban areas in France with more than 3,000 buildings. 80 to 99 % F-score values are attained for the most frequent errors. For scalability purposes, the impact of the urban area composition on the error prediction was also studied, in terms of transferability, generalization, and representativeness of the classifiers. It shows the necessity of multi-modal remote sensing data and mixing training samples from various cities to ensure a stability of the detection ratios, even with very limited training set sizes.

Résumé

La génération automatique de modèles de construction 3D à partir de données géospatiales est maintenant une procédure standard. Une littérature abondante couvre les deux dernières décennies et plusieurs solutions logicielles sont maintenant disponibles. Cependant, les zones urbaines sont des environnements très complexes. Inévitablement, les producteurs de données doivent encore évaluer visuellement, à l'échelle de villes, l'exactitude de ces modèles et détecter les erreurs fréquentes de reconstruction. Un tel processus fait appel à des experts et prend beaucoup de temps, soit environ 2 h/km^2/expert. Cette thèse propose une approche d'évaluation automatique de la qualité des modèles de bâtiments 3D. Les erreurs potentielles sont compilées dans une nouvelle taxonomie hiérarchique et modulaire. Cela permet, pour la première fois, de séparer erreurs de fidélité et de modélisation, quelque soit le niveau de détail des bâtiments modélisés. La qualité des modèles est estimée à l'aide des propriétés géométriques des bâtiments et, lorsqu'elles sont disponibles, d'images géospatiales à très haute résolution et des modèles numériques de surface. Une base de référence de caractéristiques *ad hoc* génériques est utilisée en entrée d'un classificateur par Random Forests ou par Séparateurs à Vaste Marge. Des attributs plus riches, s'appuyant sur des noyaux de graphes ainsi que sur des réseaux de type Scattering ont été proposées pour mieux prendre en compte la structure dans la donnée 3D. Les cas multi-classes et multi-étiquettes sont étudiés séparément : de par l'interdépendance entre les classes d'erreurs, il est possible de détecter toutes les erreurs en même temps tout en prédisant au niveau sémantique le plus simple des bâtiments corrects et erronés. Le cadre proposé dans cette thèse a été testé sur trois zones urbaines distinctes en France avec plus de 3 000 bâtiments étiquetés manuellement. Des valeurs de F-score élevées sont atteintes pour les erreurs les plus fréquentes (80 – 99 %). Pour une problématique de passage à l'échelle, l'impact de la composition de la zone urbaine sur la prédiction des erreurs a également été étudié, en termes de (i) transférabilité, de (ii) généralisation et de (iii) représentativité des classificateurs. Cette étude montre la nécessité de disposer de données de télédétection multimodale et de mélanger des échantillons d'entraînement provenant de différentes villes pour assurer une stabilité des taux de détection, même avec des tailles d'ensembles d'entraînement très limitées.