An efficient representation of 3D buildings: application to the evaluation of city models

Oussama Ennafii, Oussama Ennafii, Arnaud Le Bris, Florent Lafarge, Clément Mallet

AN EFFICIENT REPRESENTATION OF 3D BUILDINGS: APPLICATION TO THE EVALUATION OF CITY MODELS

Oussama Ennafii[1,2,3,*], Arnaud Le Bris[1], Florent Lafarge[2], Clément Mallet[1]

[1] LASTIG, Univ Gustave Eiffel, ENSG, IGN, F-94160 Saint-Mandé, France
[2] INRIA, TITANE, 06902 Sophia Antipolis, France
[3] Gambi-M, SARA, 77420 Champs-sur-Marne, France
oussama.ennafii@gambi-m.com

Commission II, WG II/4

KEY WORDS: Error taxonomy, 3D building models, Quality evaluation, 3D feature representation, ScatNet, Graph kernels, Classification.

ABSTRACT:

City modeling consists in building a semantic generalized model of the surface of urban objects. These could be seen as a special case of Boundary representation surfaces. Most modeling methods focus on 3D buildings with Very High Resolution overhead data (images and/or 3D point clouds). The literature abundantly addresses 3D mesh processing but frequently ignores the analysis of such models. This requires an efficient representation of 3D buildings. In particular, for them to be used in supervised learning tasks, such a representation should be scalable and transferable to various environments as only a few reference training instances would be available. In this paper, we propose two solutions that take into account the specificity of 3D urban models. They are based on graph kernels and Scattering Network. They are here evaluated in the challenging framework of quality evaluation of building models. The latter is formulated as a supervised multilabel classification problem, where error labels are predicted at building level. The experiments show for both feature extraction strategy strong and complementary results (F-score $> 74\,\%$ for most labels). Transferability of the classification is also examined in order to assess the scalability of the evaluation process yielding very encouraging scores (F-score $> 86\,\%$ for most labels).

1. INTRODUCTION

1.1 3D urban modeling

City modeling consists in building a geometric abstraction of urban objects enriched with semantics. It is an approximation of the real world. From an overhead remote sensing perspective (satellite/airborne images or point clouds), necessity mainly lies in buildings (Musialski et al., 2013). The literature heavily focuses on *surface reconstruction*. Three dimensional (3D) meshes provide high geometric fidelity but often neglect semantics (Blaha et al., 2016) and insufficiently describe urban objects (Biljecki et al., 2016). Conversely, *3D modeling* targets to represent the urban environment with a certain degree of generalization and compactness (i.e., Level of Detail), which depends on the spatial resolution of the remote sensing data (Figure 1). 3D models, in fact, include geometry, semantics, and potentially attributes, related to their shape, type, use, etc (Figure 1). The explicit knowledge of the semantics efficiently helps to process the 3D geometry for simplification, spatial analysis, rendering or visualization purposes.

1.2 Evaluating 3D building models

Automated 3D modeling of urban scenes has been widely studied but no solution guarantees perfect models whatever the types of buildings and urban environments (Musialski et al., 2013, Förstner, 2016). This entails a need for quality evaluation, which has been, so far, manually performed and highly time consuming. Automating this evaluation step requires efficient building representations as a basis for comparison with reference data. It has been barely investigated in the literature. Indeed, this has usually been achieved by reporting some global geometric metrics such as Root Mean Square Error (RMSE) which result from comparing these models to their reference (Kaartinen et al., 2005, Zebedin et al., 2008, Zeng et al., 2014, Rottensteiner et al., 2014, Nguatem and Mayer, 2017, Zeng et al., 2018). The defects are, in general, not localized and, more importantly, not very informative for a later correction step (Michelin et al., 2013).

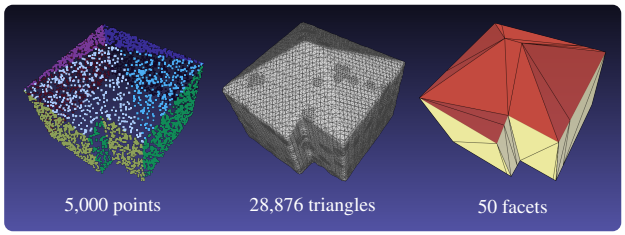

Figure 1. 3D modelling of a building from remote sensing data. *Left*: raw data (a 3D point cloud), without topological information. *Middle*: a mesh, many details but no semantics. *Right*: a model, high compaction and explicit semantics for each architectural feature (`roof` - `façade`).

In contrast, there have been some recent works where the problem has been formulated as a supervised classification one (Boudet et al., 2006, Michelin et al., 2013, Ennafii et al., 2019). In fact, quality is evaluated by the detection (or lack of) of errors in surface modeling. This is usually done, manually or semi-automatically, by comparing the 3D model to reference data. In order to scale this procedure, the problem is cast as a super-

* Corresponding author.

vised learning one: based on a predefined list of errors, learn to detect defects using the statistical properties of a pre-annotated dataset, as well as the comparison to external raw sensor data. This hinges on two main ideas:

- Developping an adapted error categorization that can describe all possible errors.
- Proposing an efficient model representation that can describe the quality of models efficiently.

1.3 Contributions

In this paper, we adopt and improve on the framework proposed in (Ennafii et al., 2019) to evaluate 3D models. For this purpose, we present:

(i) A **new feature extraction procedure** that is adapted to the building models and generic Boundary representation (B-rep) surfaces in general. For that purpose, we look at the 3D models as graphs of facets which are then classified using graph kernels.

(ii) We also make use of **external modalities** (height maps or satellite/aerial images) with the help of Scattering Networks (ScatNets) (Sifre and Mallat, 2013, Oyallon and Mallat, 2015) in order to better represent the similarity of the 3D model in question to the sensor acquired data.

(iii) We achieve a **scalable** quality evaluation workflow that does not depend on the origin of the model, in contrast with the work in (Ennafii et al., 2019). This is achieved thanks to this new 3D model representation.

2. RELATED WORK

In this section, we present the state-of-the-art in terms of evaluation of 3D building models. It will also be the opportunity to summarize the evaluation framework that will be used in the rest of the paper. We will have also the opportunity to describe the related works on graph kernels and ScatNet which will be used for the feaure extraction.

2.1 3D city model evaluation

2.1.1 General overview The state-of-the-art methods can be sorted according to the input data and desired outputs. A comprehensive analysis is available in (Ennafii et al., 2019).

Input Two kinds of complementary inputs exist. First, accurate reference building models may be available. The task then consists in defining suitable metrics for efficient comparison. These ground truth models are manually derived either from field measurements (Vögtle and Steinle, 2003, Dick et al., 2004) or multi-view image interpretation (Zebedin et al., 2008). Few models are available and such costly and time-consuming strategy does not scale well. Secondly, Very High Resolution (VHR) remote sensing data provide a better solution and are more readily available. Both multi-view aerial images (Boudet et al., 2006, Michelin et al., 2013, Zeng et al., 2018) and 3D Light Detection And Rangings (LiDARs) point clouds (Kaartinen et al., 2005, Akca et al., 2010, Zhu et al., 2018) have shown their relevance when the spatial resolution fits with the required evaluation accuracy. The main task then lies in defining suitable features (e.g., color, geometry consistency).

Dim.	Issue		Level	`Atomic` error
2D	Seg.	Under	Building	`Building Under Segmentation (BUS)`
			Facet	`Facet Under Segmentation (FUS)`
		Over	Buidling	`Building Over Segmentation (BOS)`
			Facet	`Facet Over Segmentation (FOS)`
	Border	Imp.	Building	`Building Imprecise Borders (BIB)`
			Facet	`Facet Imprecise Borders (FIB)`
		Inc.	Buidling	`Building Inaccurate Topology (BIT)`
			Facet	`Facet Inaccurate Topology (FIT)`
3D	Imp.		Building	`Building Imprecise Geometry (BIG)`
			Facet	`Facet Imprecise Geometry (FIG)`

Table 1. Table summarizing all `atomic` errors. Dim. (*resp.* Seg., Imp. and Inc.) stands for dimensionality (*resp.* segmentation, imprecision and inaccuracy).

Output Most approaches compute fidelity metrics i.e., evaluate the local geometric precision of the model. The object of interest can be either 3D points (Kaartinen et al., 2005, Elberink and Vosselman, 2011, Landes et al., 2012, Zeng et al., 2014), lines (Kaartinen et al., 2005, Elberink and Vosselman, 2011, Michelin et al., 2013), surfaces (Zebedin et al., 2008, Landes et al., 2012, Rottensteiner et al., 2014, Zeng et al., 2014) or volumes (Zeng et al., 2014, Nguatem and Mayer, 2017). A score can be extracted per building or building facet, which is sufficient for local quantitative assessment. However, it does not result in a binary (*correct/erroneous*) answer. Alternatively, casting 3D model evaluation as a supervised classification task is more adapted. Labels can be either the degree of acceptability (user-specific e.g., "*generalized*" (Boudet et al., 2006)) or an exhaustive set of potential errors (e.g., "*under/over-segmentation*" (Michelin et al., 2013, Ennafii et al., 2019). Such solution targets to be independent to the urban scene and modeling approach. Training data is required, time-consuming, and can only be obtained in limited size (Ennafii et al., 2019).

2.1.2 Learning based quality evaluation Herein, we describe in more details the learning based quality evaluation of 3D building models presented in (Ennafii et al., 2019). This evaluation framework is adopted later on with the proposed feature representations that are discussed further in Section 3.

This method relies on two ideas. First is the fact that error categorization is adaptable as it does not depend on the origin of the model (the modeling method and the urban scene). Secondly, the model feature representation does not only rely on the existance of external remote sensing data that are compared against. It can also make use of the characteristics of the model shape which can be used as intrinsic features.

Error taxonomy Errors have been categorized in a hierarchical and modular taxonomy. All defects could be described up to a certain specificity (called `finesse`) and Level of Detail (LoD). At the highest `finesse` level, `atomic` errors are the most informative. Reporting defects in modeling could necessitate more than one `atomic` error simultaneously. These are summarized in Table 1.

Feature extraction for quality evaluation `Atomic` errors are learned and predicted independently from each other. The

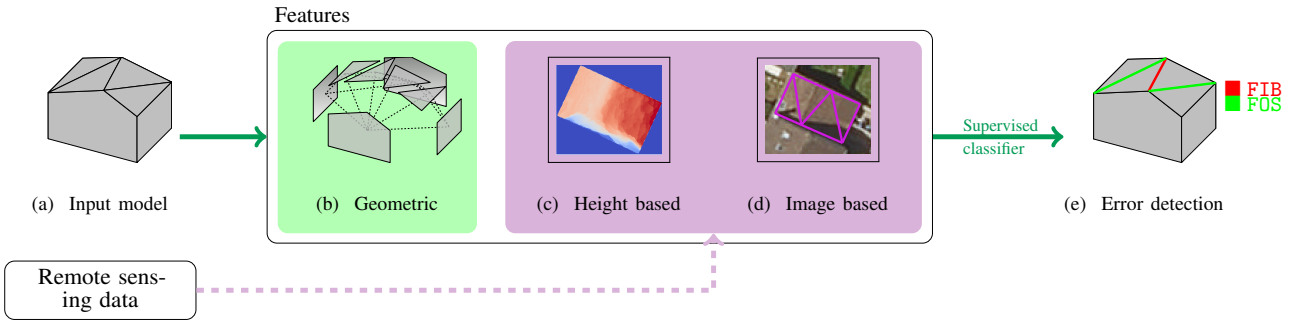

Figure 2. Quality evaluation workflow presented in (Ennafii et al., 2019) for 3D building models. Both intrinsic attributes (Figure 2b) and external modalities (Figures 2c and 2d) are used to describe the input model (Figure 2a). Based on these features, a trained classifier predicts the presence of each error label (Figure 2e).

exception being `BIG` which is redundant with facet level errors. As a result, the problem is actually cast as a multilabel ($\underbrace{4}_{\text{Building errors}} + \underbrace{5}_{\text{Facet errors}}$ labels) classification. Errors are identified only at building level even if the label is a facet error.

As illustrated in Figure 2, the input model can be described using only intrinsic geometric features, but also, comparing it to sensor acquired data. Intrinsic features are computed by considering its dual graph where nodes represent facets and edges encode their adjacency. Extrinsic features, on the other hand, rely on some readily available remote sensing data: a two dimensional (2D) height grid called Digital Surface Model (DSM) and VHR RGB optical images which are corrected from the relief and geometric distortions and called orthoimages.

The baseline approach yielded good enough prediction scores for models that share the same urban scene as the ones used for training. However, the scalability of the process proved hard to attain. In fact, the `Transferability` experiments in (Ennafii et al., 2019) showed how the prediction of some error labels was highly dependent on the urban scene of origin. In this paper, we present a new intrinsic and extrinsic representation which aims at alleviating this issue. This is addressed by taking into account the inherent structure of the data. Geometric features make use of graph kernels (cf. Section 2.2) for geometric features and ScatNets (cf. Section 2.3) for grid structures based ones.

2.2 Graph kernels

A valid representation for graphs should address three issues: (i) incorporate all possible graph sizes, (ii) notably small graphs (since building models contain few facets), and (iii) be invariant to graph isomorphisms. This would involve computing explicit feature maps to possibly infinite dimensional Hilbert spaces. Fortunately, kernels offer a more efficient approach by comparing pairs of observations. This is particularly true for graphs (Ghosh et al., 2018, Kriege et al., 2020). Five variants are tested, for their availability and numerical stability.

2.2.1 Random Walk Kernel (RWK) One way to compare two graphs is to perform a simultaneous random walk on both graphs. This is equivalent to a random walk on the Cartesian product of both graphs (Vishwanathan et al., 2010). Such a kernel (RWK) has two special cases: the exponential and geometric RWKs (Gärtner et al., 2003, Vishwanathan et al., 2010). They involve heavy computations and are numerically unstable. They also ignore node and edge attributes. More importantly, for tottering reasons, they exhibit the major drawback of focusing greatly on central nodes and ignoring isolated ones.

2.2.2 SVM ϑ Kernel (STK) This kernel takes only the graph structure into account and is agnostic to attributes. It is a tractable version of the Lovász ϑ kernel (Johansson et al., 2014). The graph comparison strategy consists in evaluating the difference between the Lovász number of their subgraphs. A direct solution is often intractable but can be approximated (Jethava et al., 2013). Such a difference is also computed for a subset of subgraphs for both graphs.

2.2.3 Multiscale Laplacian Kernel (MLK) Graphs are characterized by their Laplacian matrix. When both graphs have the same number of vertices, the Bhattacharyya kernel can be adopted to compute the similarity between the Laplacian of both graphs (Kondor and Pan, 2016). It is based on the probability distributions associated to the Gaussian graphical model on each graph. For the general case, a linear transform permits to be invariant to permutations. Such features are able to represent graphs with different sizes in the same feature space, therefore comparing any pairs of graphs (Kondor and Pan, 2016). To take node attributes (i.e., vectors associated to each graph node) into account, another linear map is applied, resulting in a Laplacian kernel. Then, scale information is incorporated: vertices from both graphs are recursively compared at increasing size neighborhoods using this Laplacian kernel. This results in a multiscale aware base kernel that is used, in a final iteration, with the Laplacian kernel to compare both graphs.

2.2.4 Propagation Kernel (PK) Similar to RWK, the concept lies in propagating information through the graph. At each iteration, the pairwise comparison of graph nodes is aggregated using the Naive kernel (Neumann et al., 2016). A propagation scheme allows to recover the graph structural information lost with the kernel. For numerical efficiency, the vertex base kernel is chosen: feature vectors are computed, avoiding pairwise comparisons. These features rely on representing each vertex using a hashing function and binning the resulting values (Shervashidze et al., 2011, Neumann et al., 2016). For node attributes, a probability distribution is assigned at each vertex (a mixture of Gaussian distributions centered on node attributes). The coefficients are propagated at each iteration, after which the vertex probability distribution is hashed. The propagation transition matrix can be user defined or, by default, the normalized adjacency matrix of the graph.

2.2.5 Graph Hopper Kernel (GHK) In order to avoid the tottering issue, path comparison between two graphs is used to computed their similarity, as adopted for the shortest path kernel (Borgwardt and Kriegel, 2005). Being intractable in practice, paths are rather compared by simultaneously "hopping"

along there vertices and comparing them. This is called the GHK (Feragen et al., 2013): the scalable version of the shortest path kernel.

2.3 Scattering Networks (ScatNets)

Convolutional Neural Networks (ConvNets) are state-of-the-art feature extractors in image classification. However, they require a great load of training images in order to learn good representations. For our challenging task, ScatNets were chosen instead. They can be seen as a reverse engineered ConvNet. They are built mainly by wavelet filters, mimicking filters from ConvNets. The latter learn, end-to-end, an image representation by minimize a certain loss function. Conversely, ScatNets make use of mathematical properties of the image signal to guarantee some properties that were observed for ConvNets: translation invariance (or covariance), stability towards local small deformations, etc. Like ConvNets, ScatNets consist in applying linear convolutional operators followed by a non-linearity and some pooling operators. The learned filters are replaced by a specific wavelet decomposition. Non-linearity is retrieved using the modulus operator, and pooling through a low pass filter. This step is critical for invariance purposes. Lost high frequencies are retrieved using the wavelet convolutions.

Formally, the linear mapping followed by the pointwise modulus operator is denoted by:

$$U[\lambda] : x \mapsto |x \circledast_G \psi_\lambda| , \tag{1}$$

where ψ_λ denotes a wavelet with parameters λ and x the input image. The scattering output of such a mapping is also refered to as coefficients. The convolution $\circledast_G$ depends on the targeted invariance group. A simple translation invariance involves the usual convolution $\circledast_{\mathbb{R}^3} \equiv \star$. Morlet wavelets are chosen with specific parameters to guarantee the required properties (Sifre and Mallat, 2013, Oyallon and Mallat, 2015). The U operator (Equation (1)) is covariant to the corresponding group (translation or rigid transformation). The subsequent low pass filtering amounts to a weighted averaging operator that enforces the invariance property up to the scale I of the filter ϕ_I. This defines the first layer coefficients:

$$S_1(x, \lambda_1) \triangleq U[\lambda_1](x) \circledast_G \phi_I. \tag{2}$$

In the next layer, the same operator U with a different parameter λ_2 is applied to each image $U[\lambda_1](x)$ from layer 1. Then, the average pooling is applied once again to result in:

$$S_2(x, \lambda_1, \lambda_2) \triangleq U[\lambda_2]\left(U[\lambda_1](x)\right) \circledast_G \phi_I. \tag{3}$$

This process can be infinitely applied. The scattering coefficient at layer 0 is simply defined as $S_0(x) \triangleq x \star \phi_I$.

The convolution at pooling step (Equations (2) and (3)) is not always the same as the convolution by wavelets (Equation (1)). This depends greatly on the required set of invariances and covariances: (Sifre and Mallat, 2013) target a roto-translation invariance, while (Oyallon and Mallat, 2015) favor translation invariance along with rotation covariance. This leads the first to average through rotations as well as translation vectors, and the second to average only on translation. In both cases, the first layer always involves a regular convolution $\circledast_{\mathbb{R}^3}$, and the subsequent ones rely on convolutions defined on the special Euclidean group $\circledast_{SE(2)}$. In practice, not all coefficients are computed as most information ($>$98 %) is concentrated in layers 0, 1 and 2, and is carried along increasing scale paths (Sifre and Mallat, 2013, Oyallon and Mallat, 2015). As a consequence, we compute the coefficients along these paths only up to the second layer resulting in a number n_S of scattering outputs.

3. EFFICIENT FEATURE EXTRACTION

In this section, we explain how these previously described feature extractors fit in the quality evaluation framework presented in Section 2.1.

3.1 Geometric features

The attributes calculated for each graph node have distinct behaviours and cannot be handled with a single graph. Instead of normalizing, concatenating and associating the resulting node attribute vectors into one graph, we preferred instead to isolate each geometric feature in a specific duplicated graph (3 in total). These graphs share the same structure. The first takes the face normals as node attributes, the second face centroids, and the last one a composite vector grouping the face degree, area and circumference. In fact, separately taken, the last ones had a limited relevance in error prediction, according to the feature importance measures provided by training Random Forests (RFs) in earlier experiments.

Each graph can take multiple types of kernels at once. Since all graphs share the same structure, kernels such as the RWK and the STK, that ignore node attributes, would yield the same results. We also experiment with three other types of kernels: the MLK, the PK and the GHK. The latter depends on the choice of the base kernel which compares node attributes. The Radial Basis Function (RBF) was discarded as it did not yield desirable results when experimented with. There are two alternatives: the linear kernel and Brownian bridge one (Borgwardt and Kriegel, 2005). This results in total in 14 graph kernels.

$$\underbrace{2}_{\text{STK \& RWK}} + \underbrace{3}_{\substack{\text{attributed}\\ \text{graphs}}} \times \Big(\underbrace{2}_{\text{MLK \& PK}} + \underbrace{1}_{\text{GHK}} \times \underbrace{2}_{\substack{\text{base}\\ \text{kernels}}} \Big) = 14.$$

These are aggregated into one kernel using a linear combination. This is possible thanks to Multiple Kernel Learning (MKL) which consists in finding the convex linear combinations of kernels which optimizes the Support Vector Machine (SVM) function (Rakotomamonjy et al., 2008, Aiolli and Donini, 2015). Other kernel types were briefly experimented, namely the Lovász ϑ, Graphlet Sampling, Subgraph Matching and Shortest Path kernels. However, they did not yield any valuable results, mostly failing numerically.

3.2 Height based features

A significant discrepancy between the model and measured depth exhibits specific textures (Ennafii et al., 2019), suited for ScatNets (Bruna and Mallat, 2013, Sifre and Mallat, 2013). The height data can be fed directly to a ScatNet without requiring any normalization or preprocessing. In fact, by construction, these extractors can admit any type of 2D signal.

In practice, the residual maps have different sizes $h_{\mathsf{M}} \times w_{\mathsf{M}}$ depending on the input model. Consequently, concatenating ScatNet coefficients into a single vector results in variable feature vector dimensions. One solution is to resize all images at a certain fixed size beforehand. However, this solution negatively

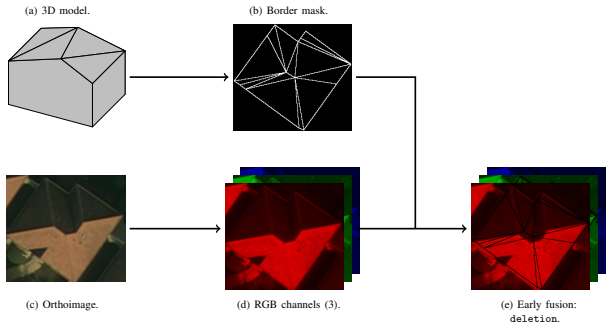

Figure 3. Early fusion scheme (`deletion`). Pixels that coincide with the 3D model border mask are assigned the zero value.

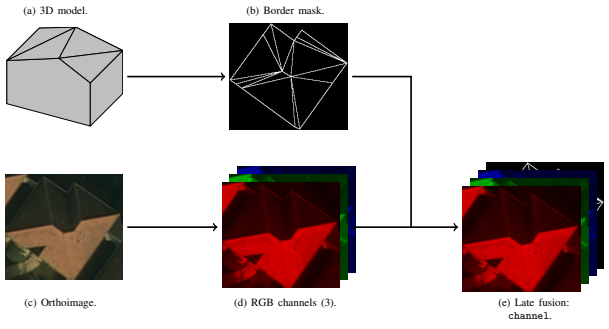

Figure 4. Late fusion scheme (`channel`). The mask indicating the pixels that intersect the edges of the nadir projection of the model is added as a fourth channel.

impacted our preliminary results: this either looses valuable structural information or adds undesired blur and it completely deforms the input signal. 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, 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. To that end, for each scattering output, some statistics are computed by applying the function:

$$\chi : l \mapsto \left(\max\left(l\right), \min\left(l\right), \mathrm{mean}\left(l\right), \mathrm{median}\left(l\right), \mathrm{stdev}\left(l\right)\right). \tag{4}$$

Stacking the results of these operations on all scattering outputs guarantee a fixed size ($5 \times \underbrace{n_S}_{\substack{\text{Total number of} \\ \text{scattering outputs}}}$) feature vector:

$$\begin{aligned}\left(v_{\text{height}}\left(\mathsf{M}\right)\right)^{\intercal} \triangleq & \left(\chi\left(S_0\left(R_{\mathsf{M}}\right)\right)\right) \oplus \left(\chi\left(S_1\left(R_{\mathsf{M}}, \lambda_1\right)\right)\right)_{\lambda_1} \\ & \bigoplus_{\lambda_1} \left(\chi\left(S_2\left(R_{\mathsf{M}}, \lambda_1, \lambda_2\right)\right)\right)_{\lambda_1,\lambda_2}.\end{aligned} \tag{5}$$

3.3 Image based features

ScatNets are well suited for edge detection as they use Morlet wavelets for convolution operations (Zhang et al., 2007). As a result, they are used to compare real images to edge masks.

Two options for image and mask fusion are possible: an early and a late scheme. Both settings are later experimented and compared (Section 4.2). The resulting images are then fed to a ScatNet. Equation (4) is then applied so as to yield feature vectors with the same dimensions per channel ($5 \times n_S$).

`Deletion` **(early fusion)** Pixels that coincide with the 3D model border mask are assigned the zero value in the three channels (Figure 3e). This results in an image $I_{\mathsf{M}}^{\mathsf{dl}} \in \mathbb{R}^{h_{\mathsf{M}} \times w_{\mathsf{M}} \times 3}$.

`Channel` **(late fusion)** The 3D model border mask is simply added to the orthoimage as a fourth channel (Figure 4e). This results in an image $I_{\mathsf{M}}^{\mathsf{ch}} \in \mathbb{R}^{h_{\mathsf{M}} \times w_{\mathsf{M}} \times 4}$.

We end up with the following feature vector ($5 \times n_S$ per channel):

$$\begin{aligned}\left(v_{\text{image},o}\left(\mathsf{M}\right)\right)^{\intercal} \triangleq & \left(\chi\left(S_0\left(I_{\mathsf{M}}^{\mathsf{o}}\right)\right)\right) \oplus \left(\chi\left(S_1\left(I_{\mathsf{M}}^{\mathsf{o}}, \lambda_1\right)\right)\right)_{\lambda_1} \\ & \bigoplus_{\lambda_1} \chi\left(S_2\left(I_{\mathsf{M}}^{\mathsf{o}}, \lambda_1, \lambda_2\right)\right)_{\lambda_1,\lambda_2},\end{aligned} \tag{6}$$

where o = del/ch is the fusion option scheme.

4. EVALUATION OF 3D CITY MODELS

4.1 Setup

4.1.1 Dataset Three urban areas are studied (Ennafii et al., 2019): **Elancourt**, **Nantes**, and the XIII$^{\text{th}}$ district of Paris (**Paris-13**). The first scene is rich in terms of building diversity containing residential areas with gable and hip roof buildings as well as districs with large industrial flat roof buildings. **Nantes** and **Paris-13** reprensent a much denser urban setting where flat roof high towers coexist with Haussmann style buildings that typically exhibit highly fragmented roofs. Both these scenes were merged into a single dataset **Na-P13**.

Building models were automatically obtained (Durupt and Taillandier, 2006). Thanks to a homegrown tool (`proj.city`), we can project building models in the nadir direction and produce the corresponding height maps. 3,235 building models were manually annotated (2,007 and 1,226 for **Elancourt** and **Na-P13**, *resp.*). Classes are highly imbalanced, resulting in poor discrimination performances for rare labels with the baseline presented in (Ennafii et al., 2019).

4.1.2 Implementation details The used DSMs and orthorectified optical image have the same spatial resolution as the 3D models (0.06 m and 0.1 m for **Elancourt** and **Na-P13**, *resp.*). Two classifiers are used in these experiments: (1) RF with the same parameters as in (Ennafii et al., 2019) and (2) a SVM classifier with a standard RBF kernel for ScatNet derived features. Its parameters are set to $C = 0.1$ and $\gamma = 0.001$ following a grid search. Regarding Multiple Kernel Learning (MKL), the already implemented `EasyMKL` (Aiolli and Donini, 2015) approach is adopted. Graph kernel attributes are implemented with the help of a `Python` module called `GraKel` (Siglidis et al., 2018).

4.1.3 Feature configurations The representations showed in Section 3 are compared to the baseline proposed in (Ennafii et al., 2019). The added value of each proposed modality is evaluated both individually and jointly, resulting in three feature combinations.

(a) Intrinsic features only: graph kernels (**KG**) applied to building models are fed alone to the classifier.

(b) Baseline geometric features (**G**) coupled with ScatNet derived ones. Starting with three possible configurations in (Ennafii et al., 2019), added to the two options for ScatNet features, we end up with five extrinsic feature configurations: (i) intrinsic and height-based features (**G** $\oplus$ **SH**);

(ii) intrinsic and image-based features with deletion (**G** $\oplus$ **SdI**); (iii) intrinsic and image-based features with channel (**G** $\oplus$ **ScI**); (iv) all features with deletion (**SdA**), and (v) all features with channel (**ScA**).

(c) All proposed features aggregated with `EasyMKL`, leading to five additional extrinsic feature configurations: (i) intrinsic and height-based features (**KG** $\oplus$ **SH**); (ii) intrinsic and image-based features with deletion (**KG** $\oplus$ **SdI**); (iii) intrinsic and image-based features with channel (**KG** $\oplus$ **ScI**); (iv) all features with deletion (**KSdA**), and (v) all features with channel (**KScA**).

4.2 Experimental results

We focus in this paper on the prediction power of these new features and more importantly on the scalability of the quality evaluation. Some labels being rare, we choose to report, for each error label, the F-score as a prediction metric.

4.2.1 Prediction power In order to assess the prediction power of this new representation, we conducted the so called `Vanilla` experiments. It involves learning and testing on the same urban zone with a 10-fold cross validation. As seen in the previous section, there is a great number of possible combinations. To simplify the reasoning, we only focus on two comparisons. In the first, only intrinsic features are used for learning. In the second, we add the external modalities and compare the best combination with this new representation to the best one using the baseline of (Ennafii et al., 2019).

Intrinsic attributes Here, we compare graph kernels to the baseline for intrinsic attributes (Figure 5). For context, we add the F-scores obtained using the RMSE, as used in the state-of-the-art, as a feature to predict errors on **Elancourt** (Ennafii et al., 2019).

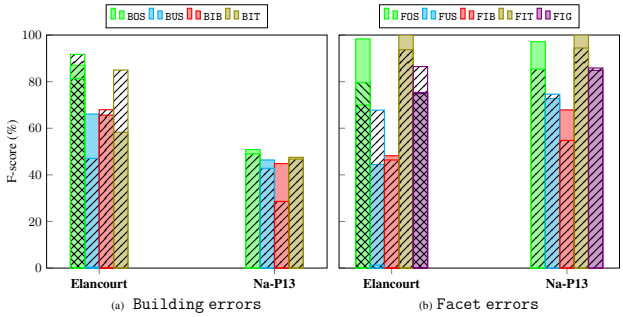

Figure 5. `Vanilla` F-scores obtained with a SVM using graph kernel features (colors). Hash bars: north east lines for SVM baseline and north west lines for the RMSE.

Regarding **Na-P13**, all labels either record comparable F-scores or better benefit from using graph kernels. On **Elancourt**, however, the `BIT`, `FUS` and `FIG` labels show poorer results. This is due to the baseline intrinsic features overfitting to the learned set as will be explained later in the discussion. The F-scores obtained with RMSE are very weak with the exception of `BOS`, `FOS` and `FIG`. This is simply due to the fact that these labels are very frequent that the classifier predicts all instances to be erroneous.

All attributes We add the extrinsic feature configurations ((b) and (c) in Section 4.1) to the comparison. In order to compare different configurations, only the best combinations, for each

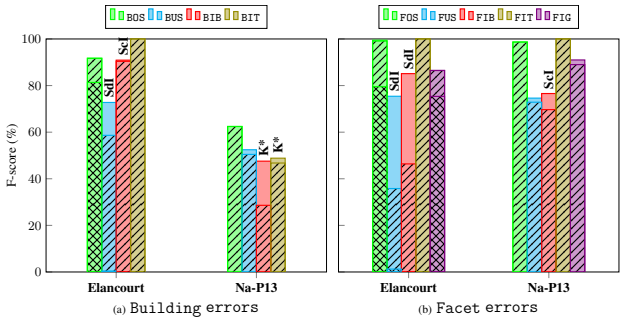

Figure 6. Best `Vanilla` F-scores out of all possible combinations with our proposed feature extractors (colors). Hash bars: north east lines for SVM and RF baseline and north west lines for the RMSE.

label, are reported here for both feature extraction approaches (Figure 6).

On **Na-P13**, there is no particular gain in using ScatNet with extrinsic features. On **Elancourt**, it is almost the same case except for `BUS`, `FUS` and `FIB` which benefits greatly from the image modality (**S*I**). Moreover, especially for the last zone, ScatNet based extrinsic attributes are more critical for error detection compared to baseline ones. Additionally, Image based features proved to be the most helpful overall. No clear pattern could be detected trying to distinguish between both fusion schemes.

Discussion Quality evaluation based solely on intrinsic features hinges on finding significant statistical patterns in the training dataset. For instance, if one of two building models with close feature representations is known to have certain errors then the other will be predicted as having the same errors. This may be true in the same urban scene but would not hold for larger scales. Consequently, two situations are expected to occur. First is that, no matter the used feature extractor being used, intrinsic attributes would not help scale the quality evaluation. This is studied later on in the next set of experiments. The second consists in the fact that intrinsic features could easily overfit to some false patterns. This could explain the mixed results of using graph kernels compared to the baseline. In fact, one can suspect that the baseline has overfitted on **Elancourt** as shown in Figure 5. This is further confirmed by the poor transferability of the baseline intrinsic features as shown in (Ennafii et al., 2019).

Regarding extrisinc features, they usually yield: (i) the same results as the baseline when these were good ($>$60 %); (ii) better results otherwise. From a general perspective, using a ScatNet did not improve largely the prediction power of the extrinsic features on the same scene used for training.

4.2.2 Scalability of the evaluation process For the quality evaluation to be scalable, learned classifiers should have sensibly the same prediction power on instances from different urban scenes as on the ones from the same urban scene used for learning. To that end, we conduct the `Transferability` experiments, we train on one zone Z_i and test on another one Z_j. The test results on Z_j are then compared to the test results of `Vanilla` experiments on the same zone. If transferable, the `Transferability` test results on Z_j should be at least as good as the `Vanilla` test results. Ideally, these test results should depend only on the number of instances in the training set and not its origin. In fact, this would mean that annotating massively on one well studied zone should suffice to predict on other unseen

zones. In contrast, the baseline in (Ennafii et al., 2019) produced mixed `Transferability` results and necessitates an annotation step to adapt to new areas. As with the previous comparisons, we report detailed comparisons using intrinsic attributes alone as well as the best combinations using external modalities.

Intrinsic attributes Only graph kernel based representation of models is compared here to the baseline (Figure 7).

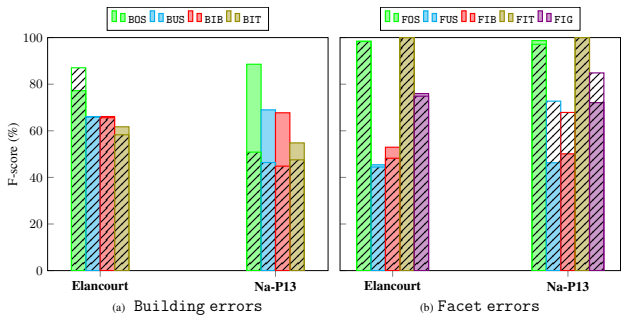

Figure 7. `Transferability` F-scores obtained with an SVM using graph kernels features (colors). Hash bars: `Vanilla` F-scores of Figure 5.

For **Elancourt**, a decrease in F-score is noted for `BOS` alone, while `BIT` and `FIB` benefit from training on **Na-P13**. Training on **Elancourt** proves to be beneficial for all `Building errors` on **Na-P13** but detrimental for `Facet errors`. This is due to the fact that dense urban scenes are more suited for training `Facet errors` (but not `Building errors`), as already illustrated in (Ennafii et al., 2019) with baseline features. Otherwise, in general, F-scores prove to be more stable compared to the `Transferability` experiments conducted in (Ennafii et al., 2019).

Best combinations The combinations with the best `Transferability` results are reported here (Figure 8).

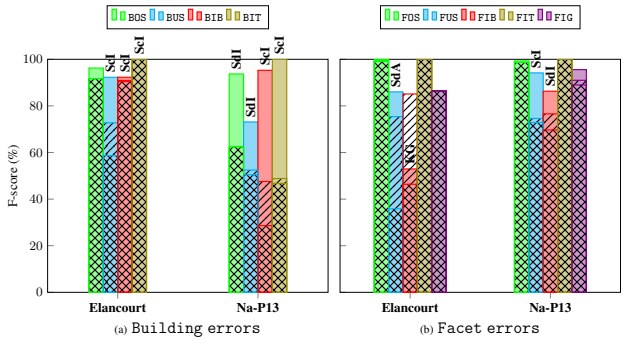

Figure 8. `Transferability` experiment F-scores obtained with an RF or an SVM using ScatNet features (in color). Hash bars: north east lines for `Vanilla` F-scores (Figure 6) and north west lines for the `Vanilla` baseline F-scores.

In general, a better transferability is observed. In fact, regarding **Elancourt**, except for `FIB`, training on **Na-P13** yields the same and even slightly better F-scores compared to training on the same zone. For **Na-P13**, results are even more satisfactory: better F-scores are obtained on `Building errors`, as previously, albeit with a larger margin. On the other hand, we record either a gain or stability in F-scores when training on **Elancourt** for `Facet errors`. In addition, Not only are the `Transferability` results good with the use of the proposed representation (with the exception of `FIB` on **Elancourt**), but better F-scores are achieved by training on different scenes than the baseline **Vanilla** results. This is especially helpful with **Na-P13**, where `Building errors` were difficult to detect.

Discussion As discussed earlier, the evaluation based on intrinsic features depend highly on the training scene. In fact, if it were not the case, then test F-scores, when training on **Elancourt** and testing on **Na-P13** (which contains fewer samples than the other scene), should be consistently higher than the `Vanilla` test results on that same zone. This was obviously not the case as `Facet errors` recorded lower test ratios while it was the inverse situation for `Building errors`.

Extrinsic features on the other hand rely on comparing the 3D model to real patterns measured from remote sensing data. As a result, they are expected to transfer well from one scene to the other. In fact, great improvements are observed for **Na-P13** for almost all errors while test results are mostly stable when transfering to the larger **Elancourt** set.

5. CONCLUSION

3D building models are pivotal in many applications, in which they are transformed into 3D meshes. This leads to a lack of efficient representations of semantized models. As such, we have proposed specific representations for such models that proved to be efficient and, more importantly, transferable from one urban environment to another. For that purpose, this representation relied on applying graph kernels to intrinsic geometric attributes of these models. In addition, ScatNets were also used to extract extrinsic features by comparing the actual model to Very High Resolution aerial images or height maps. We applied the quality evaluation framework of 3D city models developed in (Ennafii et al., 2019). Not only the proposed feature extractors yield better results than the baseline on different scene types (especially for rare classes), but also proves to be almost perfectly transferable from one scene to another. The last point is an important issue since error annotation is very resource intensive. Our solution would help keeping this manual task to a minimum. This evaluation framework along with the proposed representation could be generalized to B-rep surfaces other than 3D building models. It could also be used for a automatic or interactive model correction workflow in industrial applications.

REFERENCES

Aiolli, F., Donini, M., 2015. EasyMKL: a scalable multiple kernel learning algorithm. *Neurocomputing*, 169, 215–224.

Akca, D., Freeman, M., Sargent, I., Gruen, A., 2010. Quality assessment of 3D building data. *The Photogrammetric Record*, 25(132), 339–355.

Biljecki, F., Ledoux, H., Stoter, J., 2016. An improved LOD specification for 3D building models. *CEUS*, 59, 25–37.

Blaha, M., Vogel, C., Richard, A., Wegner, J. D., Pock, T., Schindler, K., 2016. Large-scale semantic 3d reconstruction: An adaptive multi-resolution model for multi-class volumetric labeling. *CVPR*.

Borgwardt, K. M., Kriegel, H.-P., 2005. Shortest-path kernels on graphs. *ICDM*.

Boudet, L., Paparoditis, N., Jung, F., Martinoty, G., Pierrot-Deseilligny, M., 2006. A supervised classification approach towards quality self-diagnosis of 3D building models using digital aerial imagery. *ISPRS Archives*, 36(3), 136–141.

Bruna, J., Mallat, S., 2013. Invariant scattering convolution networks. *PAMI*, 35(8), 1872–1886.

Dick, A. R., Torr, P. H., Cipolla, R., 2004. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2), 111–134.

Durupt, M., Taillandier, F., 2006. Automatic building reconstruction from a Digital Elevation Model and cadastral data: an operational approach. *ISPRS Archives*, 36(3), 142–147.

Elberink, S. O., Vosselman, G., 2011. Quality analysis on 3D building models reconstructed from airborne laser scanning data. *ISPRS J. of Photogrammetry and Remote Sensing*, 66(2), 157–165.

Ennafii, O., Le Bris, A., Lafarge, F., Mallet, C., 2019. A learning approach to evaluate the quality of 3D city models. *Photogrammetric Engineering & Remote Sensing*, 85, 865-878.

Feragen, A., Kasenburg, N., Petersen, J., de Bruijne, M., Borgwardt, K., 2013. Scalable kernels for graphs with continuous attributes. *NIPS*.

Förstner, W., 2016. A future for learning semantic models of man-made environments. *ICPR*.

Gärtner, T., Flach, P., Wrobel, S., 2003. On graph kernels: Hardness results and efficient alternatives. *Learning theory and kernel machines*, Springer, 129–143.

Ghosh, S., Das, N., Gonçalves, T., Quaresma, P., Kundu, M., 2018. The journey of graph kernels through two decades. *Computer Science Review*, 27, 88–111.

Jethava, V., Martinsson, A., Bhattacharyya, C., Dubhashi, D., 2013. Lovász ϑ function, SVMs and finding dense subgraphs. *JMLR*, 14(1), 3495–3536.

Johansson, F., Jethava, V., Dubhashi, D., Bhattacharyya, C., 2014. Global graph kernels using geometric embeddings. *ICML*.

Kaartinen, H. et al., 2005. Accuracy of 3D city models: EuroSDR comparison. *ISPRS Archives*, 36(3/W19), 227–232.

Kondor, R., Pan, H., 2016. The multiscale laplacian graph kernel. *NIPS*.

Kriege, N. M., Johansson, F. D., Morris, C., 2020. A survey on graph kernels. *Applied Network Science*, 5(1), 1–42.

Landes, T., Boulaassal, H., Grussenmeyer, P., 2012. Quality assessment of geometric façade models reconstructed from TLS data. *The Photogrammetric Record*, 27(138), 137–154.

Michelin, J.-C., Tierny, J., Tupin, F., Mallet, C., Paparoditis, N., 2013. Quality evaluation of 3D city building models with automatic error diagnosis. *ISPRS Annals*, XL-7/W2, 161–166.

Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., Van Gool, L., Purgathofer, W., 2013. A survey of urban reconstruction. *Computer Graphics Forum*, 32(6), 146–177.

Neumann, M., Garnett, R., Bauckhage, C., Kersting, K., 2016. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102(2), 209–245.

Nguatem, W., Mayer, H., 2017. Modeling urban scenes from pointclouds. *ICCV*.

Oyallon, E., Mallat, S., 2015. Deep roto-translation scattering for object classification. *CVPR*.

Rakotomamonjy, A., Bach, F. R., Canu, S., Grandvalet, Y., 2008. SimpleMKL. *JMLR*, 9, 2491–2521.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J. D., Breitkopf, U., Jung, J., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS J. of Photogrammetry and Remote Sensing*, 93, 256–271.

Shervashidze, N., Schweitzer, P., Leeuwen, E. J. v., Mehlhorn, K., Borgwardt, K. M., 2011. Weisfeiler-lehman graph kernels. *JMLR*, 12, 2539–2561.

Sifre, L., Mallat, S., 2013. Rotation, scaling and deformation invariant scattering for texture discrimination. *CVPR*.

Siglidis, G., Nikolentzos, G., Limnios, S., Giatsidis, C., Skianis, K., Vazirgianis, M., 2018. Grakel: A graph kernel library in Python. *arXiv:1806.02193*.

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., Borgwardt, K. M., 2010. Graph kernels. *JMLR*, 11, 1201–1242.

Vögtle, T., Steinle, E., 2003. On the quality of object classification and automated building modeling based on laserscanning data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(Part 3), W13.

Zebedin, L., Bauer, J., Karner, K., Bischof, H., 2008. Fusion of feature-and area-based information for urban buildings modeling from aerial imagery. *ECCV*.

Zeng, C., Zhao, T., Wang, J., 2014. A multicriteria evaluation method for 3-D building reconstruction. *IEEE Geoscience and Remote Sensing Letters*, 11(9), 1619–1623.

Zeng, H., Wu, J., Furukawa, Y., 2018. Neural procedural reconstruction for residential buildings. *ECCV*.

Zhang, L., Qian, T., Zeng, Q., 2007. The Radon measure formulation for edge detection using rotational wavelets. *Commun. Pure Appl. Anal*, 6(3), 899–915.

Zhu, L., Shen, S., Gao, X., Hu, Z., 2018. Large scale urban scene modeling from mvs meshes. *ECCV*.