Parameter-free/Pareto-driven Procedural 3D Reconstruction of Buildings from Ground-Level Sequences

Loic Simon [1,2] Olivier Teboul[1] Panagiotis Koutsourakis[1] Luc Van Gool[4] Nikos Paragios[1,3,5]

[1] Center for Visual Computing, Ecole Centrale Paris, [2] CMLA, ENS Cachan, [3] LIGM (UMR CNRS), Ecole des Ponts ParisTech

[4] BIWI, ETH Zurich, [5] Equipe Galen, INRIA Saclay

Abstract

In this paper we address multi-view reconstruction of urban environments using 3D shape grammars. Our formulation expresses the solution to the problem as a shape grammar parse tree where both the tree and the corresponding derivation parameters are unknown. Besides the grammar constraint, the solution is guided by an image support that is twofold. First, we seek for a derivation that induces optimal semantic partitions in the different views. Second, using structure-from-motion, noisy depth maps can be determined towards minimizing their distance from to the ones predicted by any potential solution. We show how the underlying data structure can be efficiently optimized using evolutionary algorithms with automatic parameter selection. To the best of our knowledge, it is the first time that the multi-view 3D procedural modeling problem is tackled. Promising results demonstrate the potentials of the method towards producing a compact representation of urban environments.

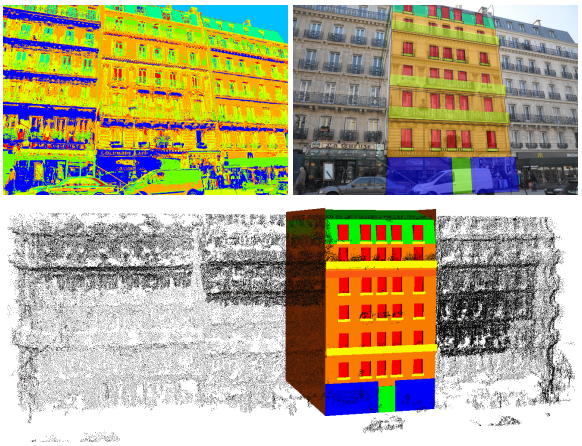


Figure 1. Based one 2D classification data (upper left) and automatically computed 3D points cloud (second row), we retrieve the structural information of the building (right and second row).

1. Introduction

Urban environment modeling has been a burning issue in the computer vision, remote sensing and geoscience communities over the past few decades. While, the theory behind stereo and structure-from-motion has became more mature, new research perspectives have emerged. One can think of the sensational impact of [11] and the many studies that followed on browsing large collections of images. Another challenge that arose recently is related to the reconstruction of large outdoor scenes. To treat efficiently such a problem, there is need to exploit compact representations. Towards this end, [4] have proposed an hybrid representation, combining predefined primitives (*e.g.* cylinders) and classical meshes. However, existing approaches lack of semantics while at the same time scalability might be an issue in terms of representation, transmission, *etc*.

Shape grammars are not so recent, they were introduced in a mathematical formalism by [12] as a generative specification of shape classes in order to analyze paintings and sculptures. During the two decades that followed, many studies made use of this framework to decode the unifying rules behind certain architectural styles [13].

This concept was emancipated in computer graphics, under the name of **procedural modeling** [18, 7]. In this domain, they were used as random generators of realistic 3D representations of buildings from a given architectural style.

Approximately in the same period, image parsing based on shape grammars have drawn a great interest. For instance, [8] and [3] focused on detecting repetitive structures to produce a grammatical explanation of an input image. Again in the computer vision domain, [1] and [16] tackled image parsing as a classification problem where the grammar can generate an extensive set (referred to as the procedural space) of semantic image segmentations. Last, [15] took up the same philosophy but endeavored to improve the procedural space exploration relying on reinforcement learning.

Despite the great interest of these methods, they fail to

provide the real 3D structure. Orthogonally, [17] used shape grammars to retrieve complex mass models of modern architectures. Ancient-heritage inverse procedural modeling has been presented in [6]. In this paper, we focus on automatic multi-view procedural reconstruction of urban 3D models embedding a mass model and faithful facade details. To the best of our knowledge, comparable approaches rely on heavy user interaction [9].

Our approach differs from prior work using shape grammars for building reconstruction in terms of output as well as in terms of means to achieve it. It assumes as input a sequence of N images $I_1, \ldots, I_N$ along with their calibration matrices $\pi_1, \ldots, \pi_N$. Similarly to existing approaches, when $N = 1$ (a case mainly considered for comparison purpose), only 2D parsing can be solved. Otherwise, we can recover the 3D structure of the building. To this end, procedural techniques are developed to generate 2D/3D geometrico-semantic layouts to be evaluated with respect to two criteria[1], leading to a multi-objective optimization where semantics and geometry are placed on an equal footing. The first criterion, evaluates the likelihood of the visual properties of the different architectural elements. The concurrent criterion accounts for the consistency between noisy reference depth-maps (derived through structure-from-motion) and the ones predicted via the procedural model. The unknowns are expressed as a grammar parse tree and are simultaneously recovered using evolutionary computation methods.

The remainder of the paper is organized as follows. In Section 2, the grammar formalism is explained and illustrated on both facade and building procedural modeling. Then, in Section 3, we present our search strategy based on evolutionary algorithms. Section 4 describes the energy functions used to implement the single-view and multi-view criteria. Last, we validate our approach in Section 5, where we show that it is competitive with the state-of-the-art for the procedural facade segmentation, and present promising quantitative/qualitative results for the newly introduced 3D procedural reconstruction problem.

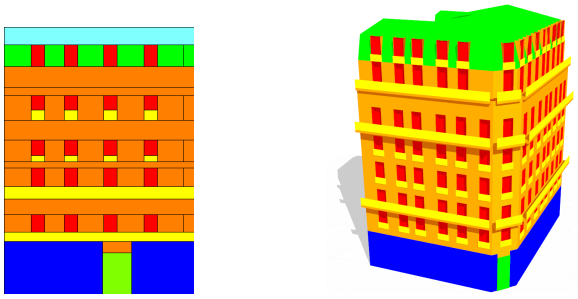
2. Procedural Building Modeling

The central idea developed here is that a shape grammar might be thought as a formal specification of how to randomly generate 2D/3D layouts (Figure 2) following some design principles. We shall see that such layouts not only bear a geometric description but also a semantic one, making them compliant to image understanding techniques.

2.1. A Shape Grammar Paradigm

In this framework, we call **shape** a collection of primitives $s = \{p_1, \ldots, p_k\}$. A **primitive** p is characterized by a class c_p (*e.g.* floor, window, balcony) and a geometric description (a rectangle in 2D or a mesh in 3D). A **rule** $r : c_r \rightarrow rhs[x](.)$ describes how to replace a primitive p by primitives produced by its right-hand-side procedure $r[\mathbf{x}](p)$, parametrized by a real-valued vector $\mathbf{x} \in \mathbb{R}^n$. These parameters usually stand for geometric properties, *e.g.* dimensions, orientations, *etc*.

[1]Note that only one criterion can be evaluated if $N = 1$.



(a) (b)

Figure 2. Grammars as semantic layout generators: random layouts obtained with a 2D grammar (a) and a 3D grammar (b).

Note that r can be applied to p only if c_p matches c_r. We denote $\mathcal{R}_p = \{r \text{ s.t. } c_r = c_p\}$ the set of such rules. Based on this, a primitive (or by extension its class c_p) will be said terminal if $\mathcal{R}_p$ is void and non-terminal otherwise.

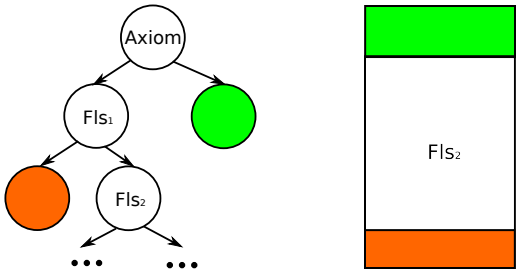
Then, a shape grammar is defined by an axiom primitive, a set of primitives $\mathcal{V}$ called vocabulary and a set of rules $\mathcal{R}$. Let us now describe how a shape grammar can be turned into a stochastic shape factory.

2.2. Derivation of a Shape Grammar

Algorithm 1 Shape grammar derivation process

$s \leftarrow \{axiom\}$
while $\exists p \in s \; s.t. \mathcal{R}_p \neq \emptyset$ **do**
  Pick $r \in \mathcal{R}_p$ randomly ($r : c_p \rightarrow rhs[](.)$)
  Sample $\mathbf{x}$ randomly
  $\{p_1, \ldots, p_k\} = rhs[\mathbf{x}](p)$
  $s \leftarrow s \cup_{i=1}^{k} p_i \setminus p$
end while

The process explained in algorithm 1 describes the random shape generation. The initial shape is composed of the axiom only. At each iteration, a non-terminal primitive p is selected in the current shape. A rule $r \in \mathcal{R}_p$ is randomly chosen along with the values of the necessary parameters $\mathbf{x}$. Then, p is replaced by the primitives emerging from the application of r to p. The process stops as soon as the shape is composed of terminal primitives only. This iterative replacement process can naturally be represented with a **parse tree** (Figure 3) where internal nodes are non-terminals, and leaves are either terminals or unprocessed non-terminals. For each leave under processing, the chosen rule application yields as many children as the produced primitives.



(a) (b)

Figure 3. Parse tree (a) associated with a partial derivation (b) of the grammar in Table 1. Terminals correspond to colored nodes.

Implementing the derivation process provides a random generator of shapes following the formal specification encoded in the grammar. The resulting shape depends directly on the random choices made during the derivation. In the perspective of optimizing them, we store the rule applied to any node and its parameters $\mathbf{x}$ in the concerned node.

2.3. 2D Modeling of Facades

Following the ideas developed in [16], we can use this procedural framework to decompose a frontal-facade image into a set of labeled rectangles. The labels corresponds to the classes of the primitives, and their geometry is always rectangular. Here we provide a simple 2D grammar to help the reader's understanding of the procedural paradigm and give concrete examples of procedures used in the rules.

The grammar rules are provided in Table 1 and few steps of a derivation are shown in Figure 4. The only procedures used here are the `x/ySplit` that carve a primitive into two along the x/y direction (*e.g.* left image in Figure 4).

r_1	Axiom	→	ySplit[h_c]	Fls_1,**Rf**
r_2	Fls_1	→	ySplit[h]	**Wa**,Fls_2
r_3	Fls_2	→	ySplit[h']	Fl_1,Fls_1
r_4	Fl_1	→	xSplit[w]	**Wa**,Fl_2
r_5	Fl_2	→	xSplit[w']	**Wi**,Fl_1

Table 1. A toy example of 2D grammar.

A more complex 2D grammar was actually designed to express sophisticated facade layouts (see Figure 2 (a)). In this version, additional procedures create dynamic links between the parameters of the rules at different scopes in the derivation so as to enforce symmetries and alignments. This kind of dynamic constraints generalizes the grammar factorization concept proposed in [16] and turns out to be more flexible towards 3D modeling of buildings.

2.4. 3D Modeling of Buildings

To generate 3D layouts as depicted in Figure 2 (b), we follow the general scheme proposed by [7]: build a mass model from a footprint, and then refine this coarse representation by further splitting each facade in both directions to first grow the floors, then the windows, wall and balconies and other terminal elements of the facade.

Figure 4. Few steps of a 2D grammar derivation.

For conciseness, the rules of the grammar are not listed here[2]. They rely mainly on 7 kinds of procedures: `Move` and `Scale` change the position and scale of a primitives, `Extrude` and `Facetize` switch between 2D and 3D primitive, `Insert` plugs a new primitive in replacement of another, `Roof` creates a mansard roof over a polygon and `x/ySplit`.

Once the grammar has been defined, the next step consists in searching among the possible designs the most suitable one to account for the images.

3. Efficient Exploration Strategy through Evolutionary Algorithms

In this section, we introduce an elegant and appropriate inference framework based on evolutionary algorithms. Recent surveys of this field can be found in [5].

3.1. Evolutionary Algorithms

Evolutionary algorithms are part of meta-heuristics which are stochastic search optimization methods. A fundamental notion in this strategy refers to elitism which favors promising individuals. The underlying concept of these algorithms is presented in Figure 5. Each iteration aims at improving a generation P made of previously evaluated individuals. Evaluation of an individual consists in computing its objective function value(s). Then promising individuals P^* are selected for mating (recombination) leading to O as offspring. After a subsequent mutation step, the offspring O is evaluated, and reinserted in the current population towards replacing (partly) the previous generation.

3.2. Optimization of the Grammar Derivation

To apply the previous methodology to our particular problem, we need to specify the kind of individuals we are studying and the evolution processes applied to them.

[2]This grammar and the 2D one are detailed in supplementary material.

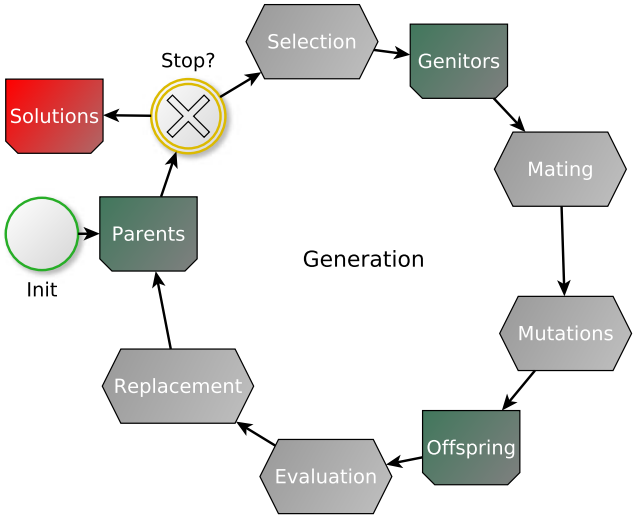


Figure 5. Typical pipeline of evolutionary algorithms.

One usually discriminates between two representations. On the one hand, the genotype is convenient to operate evolutions. On the other hand, the phenotype is a "physical" representation, more natural for fitness evaluation.

Genotype

A parse tree is made of nodes involving three components $n = (r, \mathbf{x}, \mathbf{p})$. r and $\mathbf{x}$ are the variables to be optimized. The last component, p, is the semantic primitive created during the derivation and is entirely determined by the nodes above n in the tree. For a given node, the rule can be chosen among the finite set $\mathcal{R}_p$ of rules applicable to p. On the contrary, the parameters correspond to a real-valued vector.

A **mutation** acts on a single individual. After a node $n = (r, \mathbf{x}, p)$ is chosen randomly, two options are available.

1. Rule mutations consist in replacing r by a random rule of $\mathcal{R}_p$, if $\#\mathcal{R}_p > 1$.
2. Parameter mutations add a uniform perturbation u to a randomly chosen element $x_i \in \mathbf{x}$:

$$x_i \leftarrow x_i + u, \tag{1}$$

To **recombine** two individuals, we select two nodes n and n' (one per tree), such that $c_p = c_{p'}$. Then their contents r, x and r', x' are swapped.

Note that mutations and recombinations can jeopardize the validity of specific sub-trees. Any invalid sub-tree is pruned and re-derived at random.

Phenotype and evaluation

A parse tree stores all the necessary information to derive a semantic layout $\mathcal{L}$ (Figure 2). This layout is the physical expression of the individual: its phenotype. In the single-view case, the individual relevance with respect to the observations will be evaluated thanks to an appearance energy $E_a(\mathcal{L})$. In the multiple-view case, an individual will be also evaluated thanks to a depth energy $E_d(\mathcal{L})$. The energy definitions will be detailed later.

Based on the fitness of any individual is known, an elitist process called **selection** is designed to ensure the preservation of strong features. When only $E_a(\mathcal{L})$ is considered, individuals are selected by running 2-tournaments. Two candidates are randomly picked and the fittest is kept as a genitor (*i.e.* appended to P^*).

Otherwise, $E_{total}(\mathcal{L}) = (E_a(\mathcal{L}), E_d(\mathcal{L})) \in \mathbb{R}^2$ we can compare individuals using Pareto partial ordering. Formally, we will note $\mathcal{L}_1 >_p \mathcal{L}_2$, if $\mathcal{L}_1$ has all energies smaller than $\mathcal{L}_2$. Given this partial ordering, we seek to approximate a set called the **Pareto frontier**. It consists of all individuals that are dominated by none other solution (Figure 6).

Among different options, we have chosen the modified Strength Pareto Evolutionary Algorithm (**SPEA-II**) selection scheme [19]. The process (Algorithm 2) performs both the pareto front estimation and the selection of genitors. Selection is performed by using a classical tournament on the current estimate of the front. The front is approximated by a fixed size archive A, which is initialized with all non-dominated solutions found so far and it then pruned or completed at need. Note that diversity is promoted among genitors by preferably pruning solutions located at dense positions in the current Pareto front estimate.

Algorithm 2 SPEA-II algorithm

Require: P: current population
Require: A: archive of non-dominated solutions
Require: a: maximum size of archive
  $A \leftarrow$ non-dominated solutions of $P \cup A$
  if $\#A < a$ **then**
    extend A with $(a - \#A)$ fittest dominated solutions from P
  else
    remove from A its $(\#A - a)$ individuals at dense locations
  end if
  $P^* \leftarrow$ tournament selection on A
  return P^*, A

3.3. Optimal solution in the multi-objective case

Successive iterations of the multi-objective algorithm produce a set of candidates approximating the Pareto frontier. In practice, we are looking for a unique optimal solution. Thus, we combine the two energies into a single one:

$$E(\mathcal{L}) = \alpha E_a(\mathcal{L}) + (1 - \alpha) E_d(\mathcal{L}) \tag{2}$$

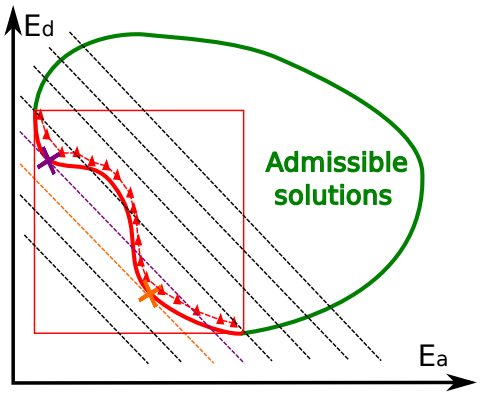


Figure 6. Illustrative example: Pareto frontier (red line) and its estimate (red triangles). Level sets of a linearized fitness E (dashed black). A local minimum of E (purple cross) far away from the global one (orange cross).

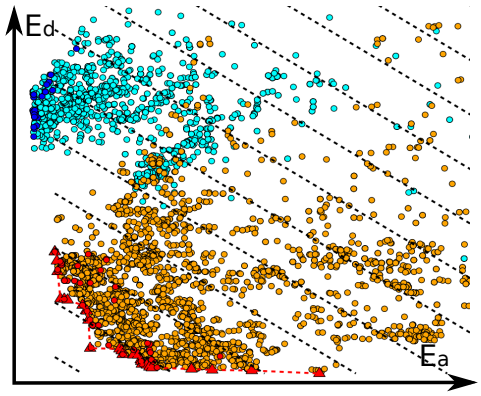


Figure 7. A population driven by **SPEA-II** in orange and one driven by a combined energy in cyan. Red/blue dots correspond the last generations and red line to the estimated Pareto set.

The level sets of E correspond to dashed straight lines in Figure 6. The optimal solution with respect to this energy is easily determined from the complete set of Pareto solutions. Note that among the approximate Pareto solutions, only those belonging to the Pareto convex hull might be optimal for a linear combination of the objectives.

Better exploration behavior

One may think that we could have used a combined energy from the beginning and rely on classical tournaments. This assumption is strongly objected in the theory of evolutionary optimization where the consensus states that keeping multiple objectives reduces the risk of early convergence to a local minimum. Concerning our problem, the validity of this statement has been observed experimentally (Figure 7). A typical case of local minima is illustrated in Figure 6 where the Pareto set is concave.

Automatic weight selection

It is also important to note that the choice of α plays a key role on the final outcome of the optimization process. Defining such a value beforehand is not straightforward. The knowledge of the Pareto set makes this task easier since its bounding box brings valuable insights regarding the appropriate balance between the two energy components. A practical value of α corresponds to iso-lines of E that are parallel to the diagonal of the Pareto set bounding box.

4. Concurrent Energy Models

In this section, we propose two energy models, one extending the appearance model proposed in [15] to non-frontal views of a 2D/3D layout and another evaluating the quality of a 3D layout with respect to depth profiles.

4.1. Appearance Energy

The first energy makes the link between the symbolic world of grammars and the statistical visual properties of the associated architectural elements. It can be used as soon as a calibrated image of the building is available. Therefore, it is adapted to facade segmentation and 3D reconstruction.

We adopt a Bayesian formulation where we aim at defining the posterior probability of the procedural layout $\mathcal{L}$ knowing the set of views I: $P(\mathcal{L}|I)$. We first express the posterior with the likelihood and the prior using the Bayes rule: $P(\mathcal{L}|I) \propto P(I|\mathcal{L})P(\mathcal{L})$. Given that the grammar already expresses a very strong prior on the semantic layout, we do not consider additional ones. Then, assuming independence between the different views and among the pixels of each image, the likelihood is factorized as:

$$P(I|\mathcal{L}) = \prod_{k=1}^{N} P(I_k|\mathcal{L}) = \prod_{k=1}^{N} \prod_{x} P(I_k(x)|\mathcal{L}) \quad (3)$$

In the factorization, we only have to express the pixel likelihood $P(I_k(x)|\mathcal{L})$. We make the natural assumption that the distribution of $I_k(x)$ given the model $\mathcal{L}$ only depends on the voxel $\pi_k^{-1}(x)$ which projects on the pixel x in I_k. Then $P(I_k(x)|\mathcal{L})$ can be rewritten as $P(I_k(x)|\pi_k^{-1}(x))$.

In the single-view case, π_1 is a one-to-one mapping from the image domain to the 2D layout. Otherwise the π_k's are not necessarily bijective and the latter expression is only valid when $\pi_k^{-1}(x)$ exists (*i.e.* when the line of sight passing through x intersects $\mathcal{L}$). In the opposite case, $\pi_k^{-1}(x)$ is considered latent and $P(I_k(x)|\mathcal{L})$ is averaged over the set of admissible values for $\pi_k^{-1}(x)$.

$P(I_k(x)|\pi_k^{-1}(x))$ is estimated based on the semantics predicted by $\mathcal{L}$ at x *i.e.* as $P(I_k(x)|c(\pi_k^{-1}(x)))$. In our settings, for any class c the probability distribution $P(I_k(x)|c)$ is learned using a Gaussian Mixture Model on the RGB values of pixels in a training set. In practice, the training set is

created by a user who paints some brush strokes on one of the input images for all the terminal classes of the grammar.

Last, the appearance energy E_a is obtained by taking the negative log-likelihood as:

$$E_a(\mathcal{L}) = \sum_k \sum_x -\log\left(P(I_k(x)|\mathcal{L})\right) \quad (4)$$

It uses appearance as a way to distinguish between the terminal semantic elements. However, it fails to account for real 3D evidence with respect to the voxels belonging to different or even to the same class when located at different depths. This can be addressed through a depth energy.

4.2. Depth Energy

As an alternative, a depth energy is obtained by comparing the model $\mathcal{L}$ with the reference 3D point cloud $\mathcal{P}_{ref}$. For each visible facade, we extract two depth maps. One is derived from the point cloud while the second corresponds to $\mathcal{L}$. Practically, each depth map is extracted from the Z-buffer of a virtual orthographic camera facing the facade. We end up with a list of pairs of depth maps $(D^1_{\mathcal{L}}, D^1_{ref}), \ldots, (D^M_{\mathcal{L}}, D^M_{ref})$, where M is the number of facades. Then the energy of $\mathcal{L}$ is:

$$E_d(\mathcal{L}) = \sum_m \sum_x ||D^m_{\mathcal{L}}(x) - D^m_{ref}(x) - \bar{d}^m||^2, \quad (5)$$

where $\bar{d}^m$ is the mean distance between $D^m_{\mathcal{L}}$ and D^m_{ref}. Therefore, the energy is minimal when both depth maps differ by a constant (not required to be null for better robustness to small inaccuracies in the calibration data).

5. Experimental Validation

5.1. Single-view Performance

The following tests were launched on the data-set proposed by [16]. 10000 hypotheses were tested for each facade. Motivated by empirical observations, the population and offspring sizes were set respectively to 32 and 16.

We computed the confusion matrix, where at row i and column j is the percentage of pixels attached to class i according to the ground truth and to j with our approach.

							[16]	[15]	c_p
70	23	5	0	2	0	0	−11	−11	window
2	**91**	6	0	0	0	1	+8	+7	wall
7	19	**73**	0	0	0	1	+1	+10	balcony
0	1	0	**76**	0	0	22	+5	−8	door
6	4	0	0	**88**	2	0	+8	+2	roof
1	0	0	0	3	**96**	0	+2	+2	sky
0	2	0	4	0	0	**93**	−1	−4	shop

Comparisons with [16, 15] regarding detection rates (diagonal entries) are shown in blue/red. This experiment shows that evolutionary algorithms provide equivalent performance as both alternatives, with a competitive number of hypotheses as in the state-of-the-art. This is very promising for the 3D multi-view extension.

5.2. Multi-view Quantitative Validation

Settings

We use tens of images to run the automatic structure-and-motion tool Bundler [10]. Then, a dense point cloud is generated using the PMVS tool [2]. For practical reasons, these steps can be performed for multiple buildings at once. Afterward, to reduce the computational burden, we limit ourselves to 2 views per visible facade.

The grammar axiom is derived from a footprint retrieved from OpenStreetMap, using the address of the targeted building. The footprint must be expressed in the same euclidean coordinate system as the calibrated cameras. We have met this requirement by providing few correspondences between the cadastrial map and the structure-from-motion point cloud. Note that this step could be automated [14] and allows the output building models to be seamlessly embedded in a complete **GIS** environment.

Our complete data set is made of 10 buildings. For a given building, a ground-truth procedural layout $\mathcal{L}_{gt}$ can be built manually. This task is time-consuming and was done carefully for each building of our data set.

The evolutionary algorithm runs 200 generations producing each at most 100 new individuals.

Performance criteria

Given that the model incorporates geometric and semantic information, it is natural to evaluate the relevance of both aspects. Regarding the semantic aspect, we keep the same line with confusion matrices, although the ground-truth labeling is derived from $\mathcal{L}_{gt}$.

To assess geometric accuracy, we can compute the average point-to-surface distance between the inferred model and the reference one.

$$d(\mathcal{L}, \mathcal{L}_{gt}) = \frac{1}{\#\mathcal{L}} \sum_{x \in \mathcal{L}} \min_{x_{gt} \in \mathcal{L}_{gt}} d(x, x_{gt}). \quad (6)$$

Results

We first consider the classification criterion presented in the following confusion matrix. Globally, the numerical values are equivalent to these obtained in the single-view experiments. We do not provide a detailed comparison with the previous experiments as the data sets differ.

$$\begin{pmatrix} \mathbf{70} & 24 & 5 & 0 & 1 & 0 \\ 3 & \mathbf{83} & 13 & 0 & 0 & 0 \\ 10 & 7 & \mathbf{82} & 0 & 1 & 0 \\ 0 & 2 & 0 & \mathbf{84} & 0 & 14 \\ 8 & 6 & 7 & 0 & \mathbf{79} & 0 \\ 0 & 4 & 0 & 2 & 0 & \mathbf{94} \end{pmatrix} \begin{matrix} \text{window} \\ \text{wall} \\ \text{balcony} \\ \text{door} \\ \text{roof} \\ \text{shop} \end{matrix}$$

In Table 2, we show the statistics obtained with Equation 6 on a per-semantics basis. Overall, the deviation is bounded by 30 centimeters. Large gaps between the ground truth and the inferred model concern mainly the roof and the shops. But in such cases, this difference has little impact on the visual quality of the model since the corresponding architectural elements are large. On the contrary, the error for windows averages to 11 centimeters and is less satisfactory given that the depth of a window amounts to approximately 50 centimeters. Nevertheless when considering the final model, this is hardly noticeable.

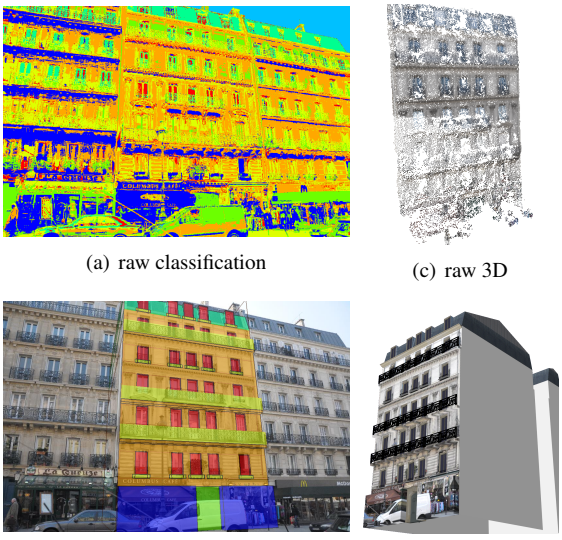
c	$d_c(\mathcal{L}, \mathcal{L}_{gt})$
window	11cm
wall	4cm
balcony	13cm
door	1cm
roof	31cm
shop	27cm

Table 2. Geometric accuracy.

We have allowed $20,000$ hypotheses which shows that despite a greater complexity in the inference task, the number of candidates remains comparable to state-of-the-art facade parsing [15]. However, even though the algorithmic complexity is harnessed, the necessary time to reconstruct a single building remains significant mainly because of the computation of the energy terms. The inference typically requires around one hour.

5.3. Examples of Multi-view Reconstructions

We present here a few qualitative results on different buildings from Paris. For each building, we show the classification induced by the 3D semantic layout and the textured 3D model in which structural elements are replaced by highly detailed models. In particular, Figure 8 draws a parallel between these results and the raw classification and depth cues. The gain is obvious with respect to both aspects. Despite the complexity of the problem and the very noisy level of appearance and depth information, the models are very accurate in terms of geometry and of topology. Different results obtained on other examples are depicted in Figure 9. Last, Figure 10 shows the reconstruction obtained with a building presenting many street facades. This example illustrates the benefit of handling the building as a whole instead of dealing with each facade separately.



(a) raw classification (c) raw 3D

(b) Optimal 3D layout (d) Optimal 3D model

Figure 8. Optimal procedural reconstruction compared to raw inputs. First row: raw classification and point cloud. Second row: the optimal 3D layout and the derived 3D model obtained by adding detailed 3D models and back-projecting a texture.

Although the results are very satisfying, few failure cases must be deplored. They occur when the actual layout cannot be generated by the grammar, as for instance balconies running only partially along the facade (third floor in Figure 8 (b)). In such a case, the reconstruction is a consensus between the language of the grammar and the observations. Besides, the registration of the point cloud and the camera is sometimes not as accurate as expected, and it can therefore introduce some bias in the final model.

6. Conclusion

In this paper, we have introduced an innovative grammar-based approach to multi-view 3D reconstruction. Our method inherits the strength of grammar representations like the modularity with respect to the class of architectures that could be derived, the compactness of the representation and the semantic understanding of the environment. This is achieved through an aggregation of segmentation and reconstruction objectives. The resulting paradigm opposite to prior grammar-based segmentation methods provide the real 3D geometry of the scene.

As a by-product, we have demonstrated the potentials of evolutionary algorithms to infer optimal grammar derivation. Because of its flexibility, this paradigm is amenable to many extensions. For instance, the estimation of the camera parameters on the fly, as well as the footprint of the building as part of the evolutionary algorithm could lead to a fully automatic approach to 3D modeling. We are also considering the fusion of range data with aerial/street-level images

Figure 9. 3D reconstruction of different buildings.

as another extension. Eventually, looking at efficient means of accelerating the convergence process through data-driven mutations is a very promising direction.

Figure 10. 3D reconstruction of a building with multiple facades visible from the street. Note that floors align seamlessly along the different facades.

References

[1] F. Alegre and F. Dellaert. A probabilistic approach to the semantic interpretation of building facades. In *International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres*, 2004. 1

[2] Y. Furukawa and J. Ponce. Patch-based multi-view stereo software. http://grail.cs.washington.edu/software/pmvs/. 6

[3] P. Koutsourakis, L. Simon, O. Teboul, G. Tziritas, and N. Paragios. Single view reconstruction using shape grammars for urban environments. In *ICCV*, 2009. 1

[4] F. Lafarge, R. Keriven, M. Brédif, and V. Hiep. Hybrid multi-view reconstruction by jump-diffusion. In *CVPR*, 2010. 1

[5] S. Luke. *Essentials of Metaheuristics*. Lulu, 2009. 3

[6] M. Mathias, A. Martinovic, J. Weissenberg, and L. Gool. Procedural 3d building reconstruction using shape grammars and detectors. In *3DIMPVT*, 2011. 2

[7] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. *ACMTOG*, 2006. 1, 3

[8] P. Müller, G. Zeng, P. Wonka, and L. Van Gool. Image-based procedural modeling of facades. *ACMTOG*, 2007. 1

[9] F. Niccolucci, M. Dellepiane, S. P. Serna, H. Rushmeierand, and L. V. Gool. Reconstructing and exploring massive detailed cityscapes. In *VAST*, 2011. 2

[10] N. Snavely. Structure from motion for unordered image collections. http://phototour.cs.washington.edu/bundler/. 6

[11] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 2007. 1

[12] G. Stiny and J. Gips. Shape grammars and the generative specification of painting and sculpture. *Information processing*, 1972. 1

[13] G. Stiny and W. Mitchell. The palladian grammar. *Environment and Planning B*, 1978. 1

[14] C. Strecha and P. Fua. Dynamic and Scalable Large Scale Image Reconstruction. In *CVPR*, 2010. 6

[15] O. Teboul, I. Kokkinos, P. Koutsourakis, L. Simon, and N. Paragios. Shape grammar parsing via reinforcement learning. In *CVPR*, 2011. 1, 5, 6, 7

[16] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape priors. In *CVPR*, 2010. 1, 3, 6

[17] C. Vanegas, D. Aliaga, and B. Benes. Building Reconstruction using Manhattan-World Grammars. In *CVPR*, 2010. 2

[18] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky. Instant architecture. *ACMTOG*, 2003. 1

[19] E. Zitzler, M. Laumanns, L. Thiele, and Others. SPEA2: Improving the strength Pareto evolutionary algorithm. In *Eurogen*, 2001. 4