Segmentation of Building Facades Using Procedural Shape Priors

Olivier Teboul[1,2] Loïc Simon[1*] Panagiotis Koutsourakis[1,3] Nikos Paragios[1,4]

[1] Laboratoire MAS, Ecole Centrale Paris, [2] Microsoft France

[3] Computer Science Department, University of Crete, [4] INRIA Saclay, GALEN Group

Abstract

In this paper we propose a novel approach to the perceptual interpretation of building facades that combines shape grammars, supervised classification and random walks. Procedural modeling is used to model the geometric and the photometric variation of buildings. This is fused with visual classification techniques (randomized forests) that provide a crude probabilistic interpretation of the observation space in order to measure the appropriateness of a procedural generation with respect to the image. A random exploration of the grammar space is used to optimize the sequence of derivation rules towards a semantico-geometric interpretation of the observations. Experiments conducted on complex architecture facades with ground truth validate the approach.

1. Introduction

Segmentation of building facades has become an important problem in computer vision of late. It is vital for accurate image-based modeling and urban scene understanding. The purpose of this activity is to discover the different regions of a facade image and assign each of them a particular semantic label (walls, windows, roof, etc). The main difficulty rests on the significant variations that may exist between buildings, even for ones corresponding to the same architectural style. Furthermore, their visual appearance spans an infinite space, due to either their internal characteristics (walls, windows, roof) or their external ones (occlusions, lighting, reflectance properties, etc.).

Traditionally, segmentation problems have been tackled through model-based or model-free methods. Model-free approaches make no assumption on the geometry and the appearance of facade components and aim at grouping pixels according to feature similarities. Among these techniques, normalized cuts [18], MRFs [10], mean shift [3] and snakes/active contours/level sets [9, 16] have been very successful. Unfortunately, these methods are prone to failure simply because pixels belonging to the same facade element do not necessarily share common visual characteristics. Model-based methods provide a segmentation which is a compromise between the available image support and a prior knowledge. In some cases, the solution is constrained to rely on a manifold (active shape [4], active appearance models [5]), while in other cases the distance from a manifold is penalized like for example snakes and level sets with priors or wavelet-like representations [6, 15]. The main assumption of these methods is that the space of solutions can be parameterized in a convenient way. This is far from being sufficient when dealing with building facades, simply because no static parameterization is generic enough to encompass different facade layouts.

Figure 1. Overview of the method. From left to right: original image, classification-based segmentation, and segmentation with procedural shape prior.

Recently, a third class of methods has been introduced in order to cope with the aforementioned limitations. Procedural models like shape grammars [7, 20] offer a flexible and powerful tool to account for such variability while being compact and providing a semantic representation of the obtained results. The idea is to represent buildings using a set of replacement rules and a dictionary of basic shapes. These methods have been extensively used in computer graphics [13, 17]. In spite of the well-known curse of dimensionality, some attempts have been made to tighten the image support with the dynamic nature of a procedural model in computer vision. The authors in [14] use mutual information to identify a repetitive pattern on a regular grid. The

*The first two authors have equally contributed to this work. Contact author: olivier.teboul@ecp.fr

result is then turned into a grammar formulation. Such an approach is based on the very strong assumption of global statistical correlation between the generation process and the observed image. In [1] the authors also describe the use of a shape grammar, but somewhat differently explore the space of building through a reversible jumps Markov Chain Monte-Carlo, and measure the appropriateness of the obtained models on the image itself. However, the supported buildings turn out to present a rather regular layout, with well defined visual appearance. Another solution has been recently proposed in [11] through an MRF formulation to address a similar problem. Such an approach consists of a fixed derivation tree that encompasses the maximum allowable variation, and is based on appearance self-similarities. Yet, it proposes a rather coarse shape representation. Such a method seems to be well suited for image-based building modeling but not for facade segmentation. Indeed, it provides repetitive patterns but does not accurately separate one semantic element from another. The found patterns show at the same time wall, window, and balcony, whereas an accurate facade segmentation is expected to distinguish between these different classes.

Our work's main contribution is to combine a powerful machine learning approach with procedural modeling as a shape prior so as to fit the purpose of multi-class facade segmentation. Generic shape grammars are constrained so as to express buildings only. Non-linear classification methods, namely randomized forests, are used to determine a relationship between the semantic elements of the grammar and the observed image support. Then the segmentation task can be considered as an optimization problem with respect to the derivation of the grammar. The main difficulty here is to efficiently decrease the energy without losing any of the flexibility provided by the grammar. In order to do so, we adopt an active search technique based on random walks. The outcome of this method is an accurate semantic representation of the building that is able to deal with the above-mentioned limitations. Promising results on complex Parisian architectures demonstrate the potential of this method. The work-flow of our approach is depicted in Fig.1.

The remainder of this paper is organized as follows: section 2 describes the 2D shape grammar while section 3 shows how to learn the appearance of semantics using randomized forests. In section 4 we present our segmentation framework which ties image support and procedural shape prior together. Section 5 provides experimental results along with both qualitative and quantitative evaluations.

2. 2D Shape Grammar for Architecture

Shape grammars can be used to efficiently describe complex but highly structured geometries, like fractals, plants and buildings. Inspired from the pioneering work of Stiny and Gips [7, 20], they have emerged as a powerful tool to describe a great variety of architectural styles. Unlike classical representations which are static, shape grammars provide a dynamic way of describing the geometry as the result of a generation procedure. *Shapes* (unlike conventional vertex, edge and polygon representations) are incrementally built using a sequence of *rules* combined with some basic dictionary elements. As a consequence, they are intrinsically tying the geometry with semantics. The rules manipulate shapes through semantico-geometric building blocks called *basic shapes*.

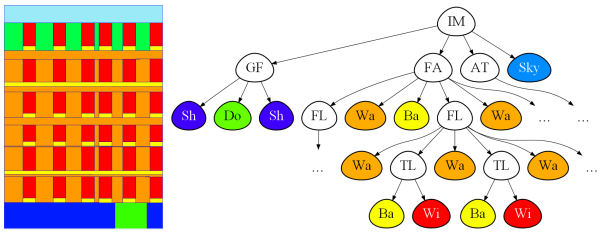


Figure 2. Derivation tree representing the procedural geometry. Each node represents an *atomic shape*, carrying a *scope* and a *basic shape*. The leaves (colored nodes) define the final shape, whereas the white nodes correspond to intermediate states.

2.1. Generic 2D Split Grammar

A shape grammar $\mathcal{G}$ is described by a dictionary of K basic shapes $\mathcal{D} = \{b_1, b_2, \ldots, b_K\}$ and a set of N rules $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$.

A basic shape is the combination of a semantic symbol and an appearance. In this work we consider the appearance of the basic shape as a square. It does not carry a notion of absolute geometry. Basic shapes are positioned in the 2D plane through a bounding box called *scope* creating an *atomic shape*. Atomic shapes are manipulated through *replacing rules*, that turn a left-hand side atomic shape LHS into some atomic shapes on the right-hand side RHS under a possible condition. We usually write a rule as follows $condition : LHS \rightarrow RHS$.

The split rule is the only kind of rule we allow in our framework. It decomposes the scope of the LHS along a splitting direction into several chunks. Their geometries are given by a fixed number s of non negative parameters $(x_1, \ldots, x_s)$ that correspond to the sizes of the chunks. We constrain the sum $x = \sum_i x_i$ of the parameters to be equal to the length of the scope of the LHS along the split direction, by adapting the parameters. Therefore, any set of parameters leads to a valid split. This property allows us to deal with different facade topologies using a single rule.

The derivation process takes an atomic shape called *axiom*, and keeps replacing the existing atomic shapes using the grammar rules. In order to keep track of the sequence of rules, rather than replacing them, we add the RHS atomic shapes as the children of the LHS, constructing a derivation tree. The final shape is defined by the leaves of the derivation tree (see Fig.2).

2.2. A Specific Grammar for Facades

In order to create a realistic facade, we propose the following derivation scheme made of 6 rules. We use a regular expressions-like formalism, in which the '.' character refers to the concatenation operator.

Rule	LHS	$\rightarrow$	RHS	DoF
1.	IM	$\rightarrow$	$GF.FA.AT.Sky$	3
2.	GF	$\rightarrow$	$Sh.Do.Sh$	2
3.	FA	$\rightarrow$	$(Ba?.FL.Wa)^*$	n_1
4.	FL	$\rightarrow$	$Wa.(TL.Wa)^*$	n_2
5.	AT	$\rightarrow$	$Rf.(TL.Rf)^*$	n_2
6.	TL	$\rightarrow$	$Ba?.Wi$	1

Therefore the vocabulary of basic shapes is defined by the symbols: *IM* (Image), *AT* (Attic), *FA* (Facade), *GF* (GroundFloor), *FL* (Floor), *TL* (Tile), **Sky**, **Wa** (Wall), **Wi** (Window), **Sh** (Shop), **Do** (Door), **Rf** (Roof) **Ba** (Balcony), where the bold face symbols are said to be *terminals*.

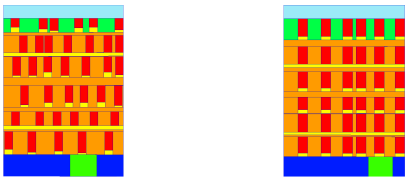
On average, the grammar has around 40 degrees of freedom, but this number may vary since the number of degrees of freedom of rules 3, 4, and 5 is not clearly defined. We make n_1 and n_2 big enough so as to handle the majority of facade layouts. For this paper we set $n_1 = 13$ and $n_2 = 11$. This allows the generated model to exhibit up to 6 floors and 10 windows per floor.

With the same derivation tree topology, a different set of parameters leads to different geometric instances. However, by following the derivation process as it has been defined, we allow a huge number of shapes to emerge whereas a lot of them are very unlikely to represent real buildings. It turns out that a pretty natural constraint can be enforced in the derivation to handle this problem. We give more details about it in the next section.

2.3. Grammar Factorization

The left hand-side of Fig.3 illustrates the kind of inconsistent facades the current derivation scheme can lead to. For instance, the first floor may be split differently from the second one, which can prevent windows from being vertically aligned. This is not usually the case for the vast majority of buildings. On the contrary, the right hand-side shows the kind of constraint we want to impose on the outgoing facade.

Therefore, the grammar tree is factorized so as to enforce the derivation to produce realistic facades only. Hence, rather than deriving the atomic shapes independently, we now force the ones sharing the same semantics (for example *FL*) to be derived in exactly the same way. In other words, the exact same rule with the exact same parameters will be applied on all the nodes from a given semantic label. For our example in Fig.2 this basically means that the rule applied on each of the sub-trees representing the floors is the same. We also force the attic (AT) to be split in the same way as any other floor (FL).



(a) No Factorization (b) Factorization

Figure 3. Effect of grammar factorization on randomly generated buildings. In (a) the windows are not aligned along the facade. In (b), the building structure is consistent from a floor to another.

By applying a factorization of the grammar, we end up with a fixed grammar tree spanning a smaller space of shapes. The factorized grammar better expresses realistic architecture, and still covers a large space of facades.

2.4. The Procedural Space

After factorization, it turns out that all the buildings are ultimately defined by a fixed number of rules to be applied. We denote $\pi = (r_1, r_2, \ldots, r_M)$ this sequence of rules or *policy*. Applied on an axiom A, the grammar provides a specific building instance. Thus a shape and a policy are two representations of the same object.

Given an axiom shape A, we can now define the procedural space of A with respect to a grammar $\mathcal{G}$ as the set of shapes $\mathcal{S}$ that can be generated using $\mathcal{G}$ starting from A:

$$\mathcal{S}(A, \mathcal{G}) = \{\pi = (r_1, r_2, \ldots, r_M) | r_i \in \mathcal{G},\ i \leq M\} \quad (1)$$

The dimension of the procedural space is:

$$d(\mathcal{S}(A, \mathcal{G})) = \sum_{i=1}^{M} DoF(r_i) \quad (2)$$

In the case of our building grammar, the dimension is 30. If for instance, each parameter lives in a discrete space of cardinal number 10, then the cardinal number of $\mathcal{S}(A, \mathcal{G})$ is 10^{30}. Without factorization, we can expect to have a different split for each of the 6 possible floors, 1 for the attic, and a different small balcony in front of each window. Thus the number of buildings grows to $10^{19} \times (10^{11})^7 \times (10^1)^{70} = 10^{166}$. If we do not force the (at most) 10 windows to have

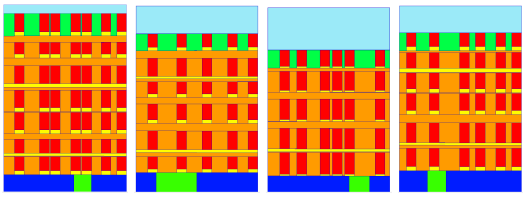


Figure 4. Randomly generated facades using the proposed factorized parametric grammar. The parameters (e.g. floor height, wall width) are randomly chosen among predefined sets.

the same size, then rules 4 and 5 have 16 degrees of freedom, and the number of building grows to 10^{201}. Factorization prevents the dimension of the procedural space from growing exponentially with the depth of the derivation tree.

As we can see from this simple computation, inconsistent shapes represent the vast majority of the procedural space before factorization. Not only does factorization reduce the dimension of the procedural space, but it also makes sense from an architectural point of view. It leads to a relatively small dimension of the space of shapes while allowing the grammar to express a large amount of facades. In Fig.4 we show some results of random generations using the proposed factorized grammar on a single axiom. As we can see the buildings show diverse structures.

3. Learning a Shape Dictionary with Randomized Forests

In this section, we introduce a supervised method to learn the appearance of the grammar dictionary in the images, in order to be able to identify them in the later segmentation task (see section 4).

3.1. Randomized Forest for Image Classification

Randomized Forests [2] are quite powerful classifiers. They have been used in numerous problems in computer vision such as object recognition [12], object classification coupled with bags of words techniques [19] or with graph cuts [21]. We adopt this machine learning technique in order to identify systematically the semantic elements of the facade. The output of the randomized forest is then used to set up a cost function for each model given an image (see section 4).

A randomized forest is used for supervised classification among C classes and is made of a set of T random decision trees. The leaves keep track of the visits of input feature vectors (see Fig.5). Internal nodes consist of a simple random test on a feature vector given as input.

During the training phase the trees are fed with labeled feature vectors. When a feature vector, associated to a label, is dropped into the tree and reaches an internal node, it goes down to the left or the right child depending on whether

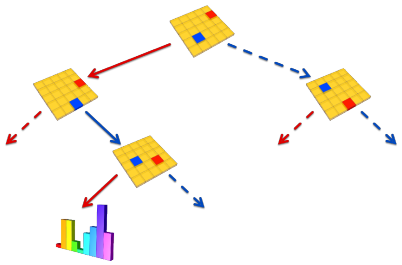


Figure 5. Principle of Randomized Tree. A patch is dropped in the decision tree. The tests are randomly built over the elements of the patch. Depending on the outcome of the test, the path is sent to one of the two children. The tree leaves record the visits of different classes, that can be seen as posterior probabilities at testing time.

the test is true or not. After d tests, the vector falls into a leaf where the number of visits per label is updated. Thus, each leaf holds a histogram $h = (h_1, \ldots, h_C)$ containing the number of feature vectors which have fallen there for each of the C classes.

During the testing phase, an unlabeled feature vector is dropped in each tree of the forest. In a given tree τ, the feature vector falls in the leaf l_τ in which a histogram of visits is stored. Once normalized, this histogram actually provides an estimation of the posterior probability for the feature vector to belong to each class c, given the leaf l_τ in which the patch has ended up:

$$P(c|l_\tau) = \frac{h_c}{\sum_i h_i} \tag{3}$$

Then, the probability over the forest is given by averaging the probabilities of all the trees.

$$P(c|(l_1, \ldots, l_T)) = \frac{1}{T} \sum_{\tau=1}^{T} P(c|l_\tau) \tag{4}$$

In our case, we want to classify all the pixels of an image. The feature vectors we consider are patches centered on the pixels. The number of trees of the forest, the size of the patches as well as the depth of the decision trees are discussed in section 3.2. Ultimately, the decision tests can be of two varieties: comparing the values of two pixels of the patch, or comparing the value of a pixel with a random threshold.

We train the randomized forest with annotated examples from Parisian buildings. This choice is not too restrictive since it enables us to target a very broad class of buildings found outside of Paris as well. The pictures are supposed to be rectified. For each image, we label by hand the various semantic elements of the facade: windows, walls, balconies, doors, roof, shop, sky and outliers. Images were taken in different lighting and weather conditions. Fig.7(a) depicts the kind of data we use for training. The whole database,

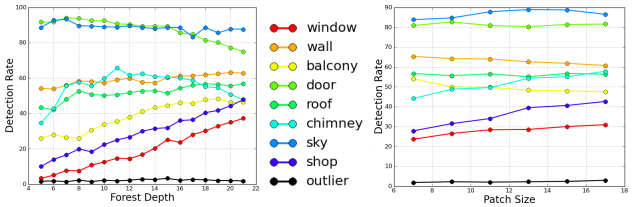


Figure 6. Evolution of the detection rates with respect to the depth of the randomized trees, and the size of the patches. They are mainly sensitive to the depth of the forest.

including the ground truth images, is available on-line at: www.mas.ecp.fr/vision/Personnel/teboul/data.html.

3.2. Randomized Forest Parameters

A randomized forest has 3 main degrees of freedom: the depth of the trees, the number of trees and the size of the feature vectors. According to [12], 10 trees ensure a healthy level of robustness. In order to choose the two other parameters, we basically train forests with depths from 5 to 21, and with patch sizes from 7 to 17. We then test the trained forest on a data set of 10 new images outside of the training set for which we have built a ground truth by hand. We attribute for each pixel the most probable label according to the obtained posterior probability. Then, we compare the classified label with the ground truth. Fig.6 sums up the detection rates for each class.

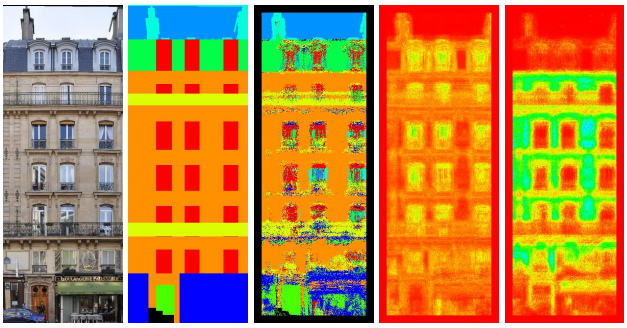
We can notice that the size of the patches has small impact on the classification, whereas the detection rates are quite sensitive to the depth of the forest. Especially difficult classes, such as windows or shops, need more decision tests to be well classified, whereas easy classes, such as *sky*, tend to have slightly better results on shallow forests.

As we can see in the first graph of Fig.6, the curves are first increasing and then occasionally decreasing. This second trend is the consequence of *over-fitting*. A very deep forest will tend to over-fit the training data, making impossible any generalization. The depth for which the forest is over-fitting actually depends on the visual complexity of each class. Therefore, deciding a fixed depth entails always a trade-off. Since windows are key elements in the building structure, we give more weight to a good window detection rate, and therefore decide to use a forest made of 10 trees, of depth 18 and with patches of size 15×15.

3.3. Segmentation via Independent Classification

Given that we can decide what is the best class for each pixel according to the classifier, we can perform a first segmentation of the image, in which each pixel is associated to a class independently from any other pixel.

Fig.7(c) shows the probabilities obtained for each class on a single image. We then build a confusion matrix obtained after applying the classification on the testing set of 10 images mentioned above. On line i and column j of the confusion matrix is the percentage of pixels that belong to class i according to the ground truth and have been classified as j by the Randomized Forest classifier. In the ideal case, the confusion matrix should be diagonal.



(a) image (b) truth (c) result (d) window (e) wall

Figure 7. (a) Original data. (b) Label image (by hand). Each color represents a label. (c) Most probable label (the same color space). (d), (e) probability for each pixel to be a *window* and a *wall*. The values increase from red (p=0) to blue (p=1).

$$\begin{pmatrix} \mathbf{30} & 11 & 13 & 7 & 12 & 14 & 3 & 8 & 2 \\ 3 & \mathbf{62} & 12 & 4 & 1 & 6 & 1 & 10 & 1 \\ 11 & 9 & \mathbf{48} & 7 & 7 & 4 & 0 & 13 & 1 \\ 1 & 2 & 2 & \mathbf{81} & 0 & 0 & 0 & 14 & 0 \\ 6 & 9 & 6 & 0 & \mathbf{57} & 12 & 9 & 0 & 1 \\ 9 & 13 & 8 & 0 & 12 & \mathbf{55} & 2 & 0 & 1 \\ 1 & 0 & 0 & 0 & 4 & 6 & \mathbf{89} & 0 & 0 \\ 6 & 7 & 9 & 28 & 6 & 1 & 1 & \mathbf{40} & 2 \\ 9 & 10 & 15 & 6 & 14 & 18 & 11 & 14 & \mathbf{3} \end{pmatrix} \begin{matrix} window \\ wall \\ balcony \\ door \\ roof \\ chimney \\ sky \\ shop \\ other \end{matrix}$$

Unsurprisingly the windows are poorly detected (30%), but the classification is still better than random (11%). Indeed, windows are not visually invariant, and even worse, they show the appearance of other objects of the scene through reflexion or transparency. If we consider the columns of the confusion matrix, we can conclude that when an object is classified as a *window*, it is truly a window 40% of the time. The randomized forest still separates windows from the other classes, even though it does not detect them very well. Finally, such a segmentation definitely lacks structure and is not directly exploitable. Therefore, we then combine this rather good but unstructured classification with our procedural framework.

4. Procedural Segmentation Framework

Grammar factorization implies that a facade is equivalent to a specific sequence of M rules, or policy π (in our case $M = 6$). As a consequence, segmenting the facade

is viewed as choosing the M rules which generate a facade that best fits the image. Once we have defined an energy function $E(\pi)$ (see section 4.1) our procedural segmentation problem can be casted as an enegy minimization:

$$\pi^* = arg \min_{\pi \in \mathcal{S}(\mathcal{A},\mathcal{G})} E(\pi) \quad (5)$$

Let us consider, without loss of generality, that we are dealing with rectified images. This preprocessing step can be performed either automatically [8] or manually, when the automated procedure fails.

4.1. Classifier-based Facade Energy

The energy or score function should quantify the appropriateness of a given policy π with respect to the image. The leaf nodes of the facade tree are taken into account. Therefore a building provides a segmentation of the image into regions R_s, where s is a semantic label such as *window*, *door*, *balcony*, etc.

Given this segmentation, the energy is based on the probability estimations provided by the randomized forest. As discussed in section 3, for each region R_c, we can compute the probability $p(x \in c)$ of each pixel x belonging to the class c expected from the grammar. The probability of the whole region R_c belonging to class c can be computed as the joint probability of all its pixels. Assuming that the pixels have independent labels, the joint probability is factorized:

$$p(R_c \in c) = p(x \in c, x \in R_c) = \prod_{x \in R_c} p(x \in c) \quad (6)$$

We can turn this into an energy using Boltzmann's transformation:

$$E(R_c \in c) = -\sum_{x \in R_c} \log p(x \in c) \quad (7)$$

Then, the energy of a whole policy π is computed as the sum of the energies of all the regions $\{R_i\}$ (with label c_i) in the segmentation provided by π:

$$E(\pi) = E(R_i \in c_i, \forall i) = -\sum_i \sum_{x \in R_i} \log p(x \in c_i) \quad (8)$$

For a given image, comparing the energies of two different buildings π_1 and π_2 is meaningful since they are both defined as a sum of positive numbers over the same image.

4.2. The Random Walk Algorithm

Although it seems at first sight that the energy is easily differentiable, and therefore candidate to gradient descent methods, the number of parameters may vary. A given derivation of the grammar will lead, for instance, to a certain number of floors and windows, making it hopeless to avoid local minima. On the other hand, a random walk is better suited to deal with the dynamic nature of the problem. It is important to notice that in spite of a varying number of parameters, the energies are always comparable, since they are all computed as a sum over the entire image.

The idea of the optimization algorithm is quite simple. We start from an initial seed policy, and randomly consider in a neighborhood of the seed if there could be policies that score better with respect to the input image and the defined energy. If so, the seed is updated, and we go on exploring the neighborhood of the new one, as specified by the following algorithm. We resample a policy by perturbing all its rules. Section 4.3 explains how to sample new rules on the fly.

```
Initial Seed π_0 = (r_1^0, ..., r_6^0)
while n < n_max:
    Perturb π_n k times:   {π_n^1, ..., π_n^k}
    compute E(π_n^i),   ∀i
    π_{n+1} = arg min_i E(π_n^i)
```

The initial seed is chosen by using a regular split, according to the image dimensions and probable values of the window dimensions. At the beginning of the process, we allow the optimizer to search for buildings far from the initial seed. As the optimization procedure progresses, we constrain the search for the optimal solution to be closer to the current seed by decreasing the standard deviation of the Gaussian law used for sampling (see equation 9). This is known as the exploitation-exploration trade-off. The more we explore the procedural space, the more knowledge we have about it and the more we want to use it. However, we still have to explore the procedural space from time to time, in order to avoid falling into local minima.

After about 50 iterations, the algorithm usually converges towards a minimum (see Fig.8). Section 5 shows results of the proposed method.

4.3. Perturbation Model

From the discussion at section 2 we conclude that a rule can be viewed as a vector in $\mathbb{R}^d_+$, where d is the number of degrees of freedom of the rule. In order to sample a new rule r in the neighborhood of a rule r_0, we basically follow the sampling equation 9, using centered Gaussian laws of given standard deviation σ.

$$r = r_0 + \sum_{j=1}^{d} x_j \delta_{ij} \quad where \quad x_j \sim \mathcal{N}(0, \sigma), \forall j \quad (9)$$

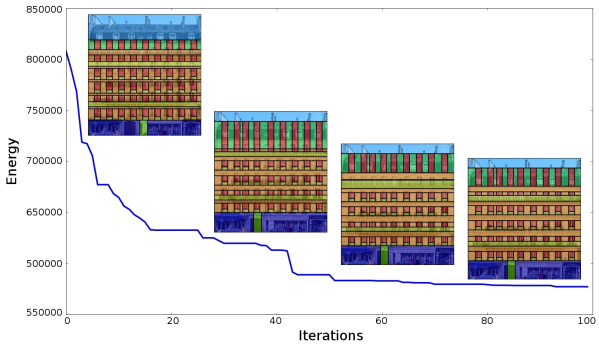


Figure 8. Evolution of the energy with the iterations. The energy is decreasing quickly in the first iterations. The random walk algorithm converges after approximately 50 iterations.

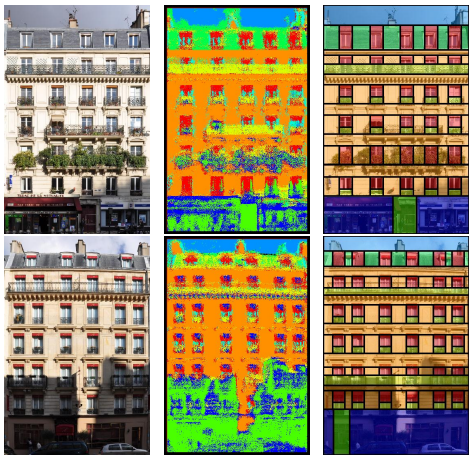


Figure 9. Facade segmentation on a Haussmannian buildings. The images show challenging illumination variations and occlusions.

where $\delta_{ij} \in \mathbb{R}^d$ is the vector with 0 everywhere except the ith component which is 1.

5. Experimental Validation

5.1. Quantitative Validation

The random walk optimization provides a well structured segmentation of the input image. We compare here the confusion matrix obtained by segmenting using the classification method as described in section 3.2, with the procedural segmentation, on the same set of 10 images with known ground truth.

$$\begin{pmatrix} \mathbf{81} & 9 & 6 & 0 & 4 & 0 & 0 & 0 & 0 \\ 5 & \mathbf{83} & 8 & 1 & 0 & 0 & 0 & 3 & 0 \\ 13 & 13 & \mathbf{72} & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & \mathbf{71} & 0 & 0 & 0 & 29 & 0 \\ 8 & 12 & 0 & 0 & \mathbf{80} & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 19 & \mathbf{0} & 75 & 0 & 0 \\ 2 & 0 & 0 & 0 & 4 & 0 & \mathbf{94} & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 & \mathbf{95} & 0 \\ 23 & 8 & 3 & 14 & 16 & 0 & 10 & 25 & \mathbf{0} \end{pmatrix} \begin{matrix} window \\ wall \\ balcony \\ door \\ roof \\ chimney \\ sky \\ shop \\ other \end{matrix}$$

Quantitatively, the confusion matrix is clearly less noisy and the individual performances per class are significantly improved. As the grammar imposes on the labeling map to be piecewise constant, local outliers are easily corrected to their actual labels. By looking carefully at the relations between semantic labels, one can notice an emulation phenomenon. Labels within a rule are helping each other (e.g. walls and windows). Eventually, as the designed grammar does not deal with chimneys, their "detection rate" is obviously zero.

5.2. Qualitative Validation

In this section, we show the diversity of results obtained using the proposed method. Combining an efficient parametric grammar with a fixed derivation scheme and some simple classifiers, we are able to handle many challenging situations.

First, we present three very challenging examples of our testing set in Fig.1 and Fig.9. In each of them, an impressive amount of information is missing in the raw segmentation obtained from the classifier only. In the first example of Fig.9, no walls or windows are detected by the classifier on the first floor and part of the second one. In the second one, the classifier is clearly misguided by the strong shadow. However, the proposed method takes advantage of the intrinsic repetitions and regularities of the grammar to recover the missing windows.

Moreover, Fig.10 shows extra results of segmentation of Haussmannian buildings (first two rows). On the last row, we present results obtained with the same classifier on facades from other architectural styles. These architectures are not explicitly present in the training set, but do share similar basic shapes with the Haussmannian style. Therefore, the Randomized Forest classifiers still provide some relevant information, and our method is still able to cope with this kind of data.

6. Conclusion and Future Work

In this paper we have proposed a novel modular approach to facade image segmentation using procedural grammars. We first constrain shape grammars towards fixed tree representations that are able to capture a wide variety of typologies of architectures. Then, we reduce the grammar complexity by factorization. We establish a connection between grammar semantics and images using machine learning techniques, and we propose a hierarchical, dynamic way to perform search through a perturbation model. Promising experimental results demonstrate the potential of the

Figure 10. Segmentation results on different buildings. The first two rows belong to Haussmannian style. The first image of the last row belongs to Restauration style (1830), the second one belongs to Louis XIV style (1680).

method.

The use of more appropriate optimization techniques that fully exploit the dynamic nature of our generation process is also a challenging but very promising path. Methods like neuro-dynamic programming and multi-armed bandits are some of the methods currently under consideration.

Acknowledgments This work has been partially supported by the Conseil General de Hauts-de-Seine and the Region Ile-de-France under the TERRA NUMERICA grant of the Pole de competitivite CapDigital and by Microsoft Research Cambridge through its PhD Scholarship Program. The authors would like to thank Aristeidis Sotiras, Chaohui Wang for their fruitful discussions and Ali Ezzatyar for his precious comments.

References

[1] F. Alegre and F. Dellaert. A probabilistic approach to the semantic interpretation of building facades. *International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres*, 2004. 2

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 4

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Tr. PAMI*, 24(5):603–619, 2002. 1

[4] T. Cootes. Active shape models. In *BMVC*, pages 266–275. Springer-Verlag, 1992. 1

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Tr. PAMI*, 23:681–685, 1998. 1

[6] S. Essafi, G. Langs, and N. Paragios. Hierarchical 3d diffusion wavelets shape priors. In *ICCV*, 2009. 1

[7] J. Gips. *Shape Grammars and Their Uses*. Birkhäuser, 1975. 1, 2

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, second edition, 2003. 6

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988. 1

[10] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Tr. PAMI*, 26(2):147–159, 2004. 1

[11] P. Koutsourakis, L. Simon, O. Teboul, G. Tziritas, and N. Paragios. Single view reconstruction using shape grammars for urban environments. In *ICCV*, 2009. 2

[12] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Tr. PAMI*, 28(9):1465–1479, 2006. 4, 5

[13] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. J. V. Gool. Procedural modeling of buildings. *ACM Tr. Graph.*, 25(3):614–623, 2006. 1

[14] P. Müller, G. Zeng, P. Wonka, and L. J. V. Gool. Image-based procedural modeling of facades. *ACM Tr. Graph.*, 26(3):85, 2007. 1

[15] D. Nain, S. Haker, A. Bobick, and A. Tannenbaum. Multiscale 3d shape representation and segmentation using spherical wavelets. *TMI*, 26(4):598–618, 2007. 1

[16] S. Osher and N. Paragios. *Level Set Methods in Imaging, Vision and Graphics*. Springer, 2003. 1

[17] Y. I. H. Parish and P. Müller. Procedural modeling of cities. In *SIGGRAPH*, pages 301–308, 2001. 1

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Tr. PAMI*, 22:888–905, 1997. 1

[19] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008. 4

[20] G. Stiny. *Pictorial and Formal Aspects of Shape and Shape Grammars*. PhD thesis, Birkhäuser, 1975. 1, 2

[21] J. M. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, pages 37–44, 2006. 4