Detection and Classification of Facades

Panagiotis Panagiotopoulos and Anastasios Delopoulos
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece[*]
Multimedia Understanding Group, Information Processing Lab, Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece

Keywords: Facades, Probabilistic Modeling, Context-free Grammars, Recursion, Learning, Classification, Window Detection.

Abstract: This paper presents a framework that exploits the expressive power of probabilistic geometric grammars to cope with the task of facade classification. In particular, we work on a dataset of rectified facades and we attempt to discover the origin of a number of query facade segments, contaminated with noise. The building block of our description are the windows of the facade. To this direction we develop an algorithm that achieves to accurately detect them. Our core contribution though, lies on the probabilistic manipulation of the geometry of the detected windows. In particular, we propose a simple probabilistic grammar to model this geometry and we propose a methodology for learning the parameters of the grammar from a single instance of each facade through a MAP estimation procedure. The produced generative model is essentially a detector of the particular facade. After producing one model per facade in our dataset, we proceed with the classification of the query segments. Promising results indicate that the simultaneous use of an appearance model together with our geometric formulation always achieved superior classification rates than the exclusive use of the appearance model itself, justifying the value of probabilistic geometric grammars for the task of facade classification.

1 INTRODUCTION

Perhaps one of the most challenging problems in machine vision is the task of matching images of the same object that have been photographed under different conditions (viewpoint, lighting conditions, occlusions, etc). In this paper we present a methodology that classifies noisy query facade instances against an original facade images. To this direction, we construct a detector for each facade in our dataset, i.e., a generative model that evaluates a query facade instance and we proceed to the classification task by evaluating all instances against all the detectors. Since facades exhibit repetitive structures and symmetries, it is not possible to directly apply traditional matching techniques, like SIFT matching (Lowe, 2004).

In this paper, we assume that a facade is generated from a context-free grammar with built-in geometric information, which uses elementary entities (windows) as an alphabet. Such a grammar is called a Probabilistic Geometric Grammar (PGG). In our setting, each grammar defines a generative facade model, whose parameters are learned by a supervised, MAP classifier.

The idea of using grammars for object modeling and recognition/detection is not new. In the literature, the study of syntactic pattern recognition was pioneered by Fu et al (Fu, 1981; You and Fu, 1979). In the mid 90's, L-systems exploited the recursive nature of grammars to model fractal structures, such as plants and leaves (Holliday and Samal, 1995; Holliday and Samal, 1994). In particular, stochastic grammatical models were used in order to express the potential uncertainty of the observed tree structures. Unlike our approach however, L-systems did not model the uncertainty of the produced geometry itself, as each rule produced a particular geometry in a deterministic way. In recent years, more sophisticated grammars, such as attribute graph grammars (Baumann, 1995; Feng and Zhu, 2005; Zhu et al., 2010b; Zhu et al., 2010a) and context sensitive graph grammars (Rekers and Schurr, 1997) have been developed to enable more powerful expressiveness and visual inference mechanisms. Additionally, in the relevant context of iterative and/or recursive patterns, there are several approaches that examine the use of Frieze and

[*]This work was partially supported by the EU-funded VITALAS project (FP6-045389)

Panagiotopoulos P. and Delopoulos A..
Detection and Classification of Facades.
DOI: 10.5220/0004283506410650
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 641-650
ISBN: 978-989-8565-47-1

Wallpaper groups for image modeling, matching or Geo-tagging (Liu et al., 2004; Schindler et al., 2008). However, they are not interested in the use of grammatical models. Finally, there is a number of recent studies on the application of parsing facade images. These studies use grammatical models for the procedural modeling of buildings and facade reconstruction (Muller et al., 2006; Wu et al., 2010; Wonka et al., 2003; Ripperda and Brenner, 2009), or for scene interpretation and segmentation (Teboul et al., 2011; Teboul et al., 2010).

Intuitively, the use of grammars in facade classification is an attractive choice due to their ability to describe compactly both the hierarchical and the recursive structure of the particular objects. Although there are several recent approaches that cope with the task of image parsing of facades, we are not aware of any approaches that proceed (after parsing) to their classification. To that direction, we propose a novel approach that is capable of modeling both the structural and the geometric uncertainty of such structures and evaluating images against these models.

Throughout this paper, we will use the grammar of Table 1 to describe facades. The proposed grammar describes facades as strings of entities that correspond to specific parse trees. The leaf nodes of the parse tree forming the string, i.e., the so called terminal symbols, correspond to the visible substructures of the facade, which in our case are the windows of the building (symbol “w”). These structures are represented only by their 2D position in our analysis. On the other hand, symbols “B” and “F” are the internal nodes of the parse tree. These non-terminal symbols correspond to the floors and the building itself. Their position is not measured from the examined image. In that sense we use the term invisible parts for non-terminal symbols.

Perhaps the most important aspect in the definition of our PGG is the inclusion of probability distributions determining the relative position of the right symbols of each replacement rule (children) with respect to the left symbol (parent).

Our framework is essentially a part based model (Felzenszwalb and Huttenlocher, 2005; Felzenszwalb and Schwartz, 2007; Fergus et al., 2006). However, the incorporation of grammars allows us to model facades whose size and structure may vary significantly among the various instantiations, due to the existence of replacement rules that produce repetitions. More importantly though and unlike traditional part-based models, the use of grammars allows us to adopt a unified description that models the original facade itself and any other partial instance of the particular facade. Which means that if we only see a part of a building, the rest of it does not have to be considered occluded, since it can be described by the adopted model. As an additional note, although we do propose an appearance model in our experiments (Section 5), we mainly focus on the ability of the grammatical model to improve the classification results when it is used together with this appearance model compared to the efficiency of the exclusive use of the appearance model itself.

In Section 2, we present a formal representation of PGGs and propose a modeling scheme that captures the statistical variance of positions. In Section 3 we formulate the bottom-up and top-down equations that indicate how children nodes define the positions of their parents and vice-versa. Based on these equations, we end up with closed-form expressions for estimating the parameters of our geometric distributions and we propose a method for learning these parameters from a single image. In Section 4 we present our window detection framework. Note that the overall methodology is not dependent on it since any algorithm that accurately detects the positions of the windows could be used instead. Section 5 presents the classification performance of the proposed framework. Finally, conclusions are discussed in Section 6.

2 PROBABILISTIC MODELING

2.1 Probabilistic Geometric Grammars

A PGG is a 5-tuple $(V,\Sigma,R,S,\mathcal{F})$, where V is a set of symbols, $\Sigma \subset V$ is the set of terminal symbols, $R \subset (V-\Sigma) \times V^*$ is a finite set of rules, $S \in (V-\Sigma)$ is the starting symbol and $\mathcal{F}$ is a set with $f^{k,j} \in \mathcal{F}$ denoting the parameters of the generative geometric model that produces the j-th child of the k-th rule. It can be seen that PGGs are in essence context free grammars (pages 113-120 in (Lewis and Papadimitriou, 1998)), including $\mathcal{F}$.

According to the aforementioned definition, Table 1 shows the PGG that is used in this paper to describe facades and Figure 1 displays a modeling example. In this grammar, terminal symbols “w” express the windows of the building and one can think of “F” as representing the floors and “B” the building itself.

On the other hand, $f^{k,j} = \{\bar{\mathbf{x}}_{k,j}, \Sigma_{k,j}\}$ are defined in Section 2.2 as the mean and covariance of a normal distribution on the relative positions between a parent node i and its j-th child in the parse tree, when production rule r_k is used.

Since PGGs are context free, the choice of a rule does not affect the choice of other rules. Additionally,

Table 1: A PGG for facades

$V=\{B,F,w\}, \Sigma=\{w\}, S\equiv B, R=\{r_1,...,r_4\}$	
$r_1: B\rightarrow BF$	$r_2: B\rightarrow FF$
$r_3: F\rightarrow Fw$	$r_4: F\rightarrow ww$

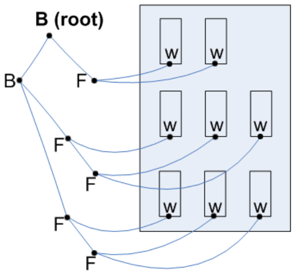

Figure 1: A facade produced by the proposed grammar.

we assume that geometric relations among the children of the same rule are independent to each other. Geometric dependencies exist only among each child and its parent. Since the employed grammar G in Table 1 is in Chomsky Normal Form, let i index a non-terminal node in a parse tree and $ch_1(i)$ and $ch_2(i)$ the indices of the left and right child of i, respectively. Moreover, let T be the set of all the parse trees of this grammar and $t\in T$ one of these parse trees. We denote the probability of observing t, given that the examined facade is some outcome of the grammar as $P(t|G)$. It can be interpreted as the probability of the union of all the geometric relations in t. If we let the relative position $\mathbf{x}_{(i,j)}$, $j=1,2$ of node i with respect to its j-th child represent the geometric relation of these nodes, taking advantage of our independence assumptions we can write:

$$P(t|G)=\prod_{\mathbf{x}_{(i,j)}\in t} P(\mathbf{x}_{(i,j)}) \tag{1}$$

2.2 Probabilistic Modeling of Geometric Relations $\mathbf{x}_{(i,j)}$

Consider a rule r_k from Table 1 and an instance of this rule, consisting of a parent indexed as i and its two children. We denote by $\mathbf{y}_i$ the position of the parent with respect to some global coordinate system and $\mathbf{y}_{ch_1(i)}$ and $\mathbf{y}_{ch_2(i)}$ the corresponding absolute positions of the two children. Then we consider:

$$\mathbf{x}_{(i,j)}=\mathbf{y}_{ch_j(i)}-\mathbf{y}_i \tag{2}$$

as a random variable depending on $\mathbf{y}_i$.

Assume that we can measure the absolute positions of the two children of parent i, i.e., $\mathbf{y}_{ch_j(i)}$ for $j=1,2$. In order to estimate the position of the parent, we would have to find the MAP estimate of $\mathbf{y}_i$:

$$\begin{aligned}(\mathbf{y}_i)_{\text{MAP}} &= \arg\max_{\mathbf{y}_i}\left\{Pr\left(\mathbf{y}_i|\{\mathbf{y}_{ch_j(i)}\}_{j=1,2}\right)\right\}\\ &= \arg\max_{\mathbf{y}_i}\left\{Pr\left(\mathbf{x}_{(i,j)}|\{\mathbf{y}_{ch_j(i)}\}_{j=1,2}\right)\right\}\end{aligned} \tag{3}$$

It is natural to let the probability on the right hand side of Equation (3) follow a normal distribution. Since our grammar is context-free and we have assumed statistical independence among the geometric relations of the children of the same rule (Equation (1)), we can write:

$$\begin{aligned}P(\mathbf{x}_{(i,j)}|\mathbf{y}_{ch_j(i)}) &= \prod_{j=1}^{2} P(\mathbf{x}_{(i,j)}|\mathbf{y}_{ch_j(i)})\\ \propto \exp &\left\{-\frac{1}{2}\sum_{j=1}^{2}\left[(\mathbf{x}_{(i,j)}-\bar{\mathbf{x}}_{k,j})^T\Sigma_{k,j}^{-1}(\mathbf{x}_{(i,j)}-\bar{\mathbf{x}}_{k,j})\right]\right\}\end{aligned} \tag{4}$$

where $\bar{\mathbf{x}}_{k,j}$ and $\Sigma_{k,j}$ are the mean and the covariance matrix of $\mathbf{x}_{(i,j)}$ respectively, for all the instances i of the rule k and for $j=1,2$.

3 GEOMETRIC DISTRIBUTION PARAMETER ESTIMATION

Let us initially examine how to determine the positions of all the nodes in a parse tree, when the parameters of the normal distribution (means $\bar{\mathbf{x}}_{k,j}$ and covariances $\Sigma_{k,j}$) are known. We identify two scenarios. In the first one, we are aware of the positions of the children and we want to estimate recursively the positions of the parent nodes (bottom-up). In the second one, we know the position of a parent node and we want to predict the positions of its children (top-down).

Bottom-up Estimation. Consider two sibling nodes produced by rule r_k. If we knew the positions of these two nodes and the values for $\bar{\mathbf{x}}_{k,j}$ and $\Sigma_{k,j}$ where would their invisible parent be? In order to find the MAP estimate of the parent, we set the first derivative of Equation (4) with respect to $\mathbf{y}_i$ equal to zero, resulting to:

$$\hat{\mathbf{y}}_i=\left(\sum_{j=1}^{2}\Sigma_{k,j}^{-1}\right)^{-1}\sum_{m=1}^{2}\Sigma_{k,m}^{-1}\left(\hat{\mathbf{y}}_{ch_m(i)}-\bar{\mathbf{x}}_{k,m}\right), \tag{5}$$

where $\hat{\mathbf{y}}_{ch_m(i)}$, for $m=1,2$ denotes the previously estimated positions for the children.

Since we assumed that the statistical parameters are known, $\hat{\mathbf{y}}_i$ can be easily estimated. Moreover, we can estimate all the non-terminal nodes by applying Equation 5 recursively from the leaves to the root of the parse tree. In the special case that $ch_m(i)$ for $m = 1,2$ corresponds to leaf nodes, we make the reasonable assumption that $\hat{\mathbf{y}}_{ch_m(i)} \equiv \mathbf{y}_{ch_m(i)}$, so that Equation 5 holds for all the nodes of the parse tree. Equation (5) estimates a parent from its children and thus, we call it **bottom-up** (*BU*) equation.

Top-down Prediction. Consider now the reverse scenario where given the position of a parent node we want to predict the positions of its children resulting from replacement rule k. The predicted children positions are given by

$$\tilde{\mathbf{y}}_{ch_j(i)} = \tilde{\mathbf{y}}_i + \bar{\mathbf{x}}_{k,j}, \tag{6}$$

for $j = 1,2$, where $\tilde{\mathbf{y}}_i$ is the measured or estimated position of parent node i and $\bar{\mathbf{x}}_{k,j}$ are the mean relative positions of rule r_k.

If we denote the position of the root node estimated by the BU equations as $\hat{\mathbf{y}}_0$, we can define $\tilde{\mathbf{y}}_0 \equiv \hat{\mathbf{y}}_0$, so that Equation 6 holds for the whole parse tree. Applying Equation 6 recursively from the root to the leaves, we can predict all the positions of the parse tree. We shall refer to Equation (6) as the **top-down** (*TD*) equation.

Throughout the rest of this paper, positions marked with hats will be associated with BU estimates, while positions marked with tildes will refer to TD predictions. Moreover, we will employ the two aforementioned notation assumptions that will be used in order to proceed with the parameter estimation, throughout Section 3.

3.1 Optimization Criteria

Consider a training set that consists of several parse trees. Although we are aware of the structure of these trees, we have no information regarding the position of the non-terminal nodes. On the other hand we are only able to measure the positions of the windows. Our goal is to estimate the distribution parameters (means $\bar{\mathbf{x}}_{k,j}$ and covariances $\Sigma_{k,j}$) of the generative model.

Let us now examine Equation (5). We can see that the position of the parent depends on the relation between the covariances of the children positions. Covariances on the other hand do not participate in TD equations. Indeed, TD equations construct ideal trees, based only on the mean values of the Gaussian distributions. Although we could try to discover a set of parameters that would bring the TD and BU trees as close as possible, this seems to be too demanding, since the visible (and measurable) information is captured only on the leaves of the parse tree. Our generative model is in fact interested in producing accurately only what is visible. Therefore, any set of parameters that sufficiently explains the observed leaves of the dataset can be accepted to be valid. We seek these parameters that bring the estimated leaves as close as possible to the observed ones. In order to achieve this, we adopt the following optimality criterion:

Definition 1. *The optimal means and covariances estimated are those that, if used in the BU procedure, yield a parse tree root node that subsequently and via the TD procedure generates leaf estimates that are as close to the observed ones as possible.*

In more detail, let our dataset consist of M facades or, equivalently, M parse trees. Each parse tree t, $t = 1,\dots,M$ has a number of m_t leaf nodes, and let $P = \sum_{t=1}^{M} m_t$ be the total number of leaves. Assume an indexed collection I of all the nodes in the dataset. Further assume that the indices of the leaf nodes within I are i_p, $p = 1\dots P$, so that all the leaf nodes have absolute positions $\mathbf{y}_{i_p}$. If

$$\begin{aligned}\Psi = [(\bar{\mathbf{x}}_{1,1},\Sigma_{1,1}),(\bar{\mathbf{x}}_{1,2},\Sigma_{1,2}),\\ \dots,(\bar{\mathbf{x}}_{4,1},\Sigma_{4,1}),(\bar{\mathbf{x}}_{4,2},\Sigma_{4,2})],\end{aligned} \tag{7}$$

we seek:

$$\hat{\Psi} = \arg\min_{\Psi} \sum_{p=1}^{P} |\mathbf{y}_{i_p} - \tilde{\mathbf{y}}_{i_p}|^2 \tag{8}$$

For the sake of clarity, Table 2 introduces some useful operators that will be used in the next sections.

3.2 Estimating Position Means

Consider a rule r_k that produces one pair of terminal nodes and let their parent be indexed with p. Since the parent node is invisible, the only information we can extract is the statistical behavior of one terminal node, as observed from the other one, i.e, the quantity:

$$\delta\mathcal{Y}(p) = \mathbf{y}_{ch_1(p)} - \mathbf{y}_{ch_2(p)} \tag{9}$$

We now seek these statistical parameters ($\bar{\mathbf{x}}_{k,1}$, $\Sigma_{k,1}$, $\bar{\mathbf{x}}_{k,2}$,$\Sigma_{k,2}$) that can reproduce the statistical behavior of $\delta\mathcal{Y}(p)$. Let's focus on the covariance matrices; the covariance of the positional difference of the children will be:

$$\begin{aligned}C_0 &= cov(\mathbf{y}_{ch_1(p)} - \mathbf{y}_{ch_2(p)})\\ &= cov(\mathbf{x}_{(p,1)} - \mathbf{x}_{(p,2)}) = C_1 + C_2,\end{aligned} \tag{10}$$

where cov denotes the covariance and $\mathbf{x}_{(p,j)} = (\mathbf{y}_{ch_j(p)} - \mathbf{y}_p)$ corresponds to the position of the parent with respect to its j-th child. The last equation holds because $\mathbf{x}_{(p,1)}$ and $\mathbf{x}_{(p,2)}$ are assumed independent. Any pair of C_1 and C_2 that satisfies Equation

Table 2: Operators.

$par(i_n)$: index of the parent of node i_n.

$ind(i_n)$: rule index and subindex of the rule that produced i_n. If for example i_n is the right child of rule k, $ind(i_n) = [k, 2]$.

$path(i_n)$: all the indices of the nodes in the path from i_n to the root (including i_n and excluding the root).

$term(d)$: indices of all the terminal nodes at depth d.

$node(d)$: indices of all the nodes at depth d.

(10) is an acceptable choice for $\Sigma_{k,1}$ and $\Sigma_{k,2}$ respectively. Therefore, we are free to choose:

$$\begin{aligned} \hat{C}_1 &= \hat{\Sigma}_{k,1} = \lambda C_0 \\ \hat{C}_2 &= \hat{\Sigma}_{k,2} = (1-\lambda) C_0 \end{aligned} \tag{11}$$

with $0 < \lambda < 1$.

This observation provides us with two important benefits:

1. We reduce the parameter space because we do not have to estimate both covariances.
2. Equation (5) transforms to a covariance free expression. In the following, we choose $\lambda = 0.5$, so that Equation (5) becomes:

$$\hat{\mathbf{y}}_i = \frac{1}{2}\left(\hat{\mathbf{y}}_{ch_1(i)} - \bar{\mathbf{x}}_{k,1} + \hat{\mathbf{y}}_{ch_2(i)} - \bar{\mathbf{x}}_{k,2}\right) \tag{12}$$

It can be proved that using the same rationale, if we denote with $\Sigma_{k,\delta\mathbf{y}}$ the statistical covariance of the positional differences of all the children produced by rule k, we can generalize so that we can find covariances, such that:

$$\hat{\Sigma}_{k,1} = \hat{\Sigma}_{k,2} = \Sigma_{k,\delta\mathbf{y}}/2 \tag{13}$$

for all rules $k = 1, ..., 4$ that can produce the statistical behavior of the observed leaves of a parse tree. We adopt this choice and thus, in the sequel we shall use Equation (12) instead of Equation (5).

In accordance to Definition 1, we will use the BU and TD equations to find an expression of the predicted leaf positions $\tilde{\mathbf{y}}_{i_p}$, with respect to the measured window positions $\mathbf{y}_{i_p}$ and the mean positions of the model. Consequently, we will estimate the position means that minimize the distance between $\tilde{\mathbf{y}}_{i_p}$ and $\mathbf{y}_{i_p}$.

According to Equation (12), the position of the root can be written as:

$$\begin{aligned} \hat{\mathbf{y}}_0 = 0.5\hat{\mathbf{y}}_{ch_1(0)} - 0.5\bar{\mathbf{x}}_{ind(ch_1(0))} \\ + 0.5\hat{\mathbf{y}}_{ch_2(0)} - 0.5\bar{\mathbf{x}}_{ind(ch_2(0))} \end{aligned}$$

By recursively eliminating the internal node positions, we come up with:

$$\hat{\mathbf{y}}_0 = \sum_{d=1}^{D}\left[\left(\frac{1}{2}\right)^d \left(\sum_{i\in term(d)} \mathbf{y}_i - \sum_{j\in node(d)} \bar{\mathbf{x}}_{ind(j)}\right)\right], \tag{14}$$

where D is the maximum depth of the particular tree. Equation (14) expresses the position of the root node with respect to the measured window positions and the mean positions.

In order to predict the position of the leaf node i_p with respect to the position of the root node $\hat{\mathbf{y}}_0$, we apply Equation (6) recursively, resulting to:

$$\begin{aligned} \tilde{\mathbf{y}}_{i_p} &= \tilde{\mathbf{y}}_{par(i_p)} + \bar{\mathbf{x}}_{ind(i_p)} = \\ &= \tilde{\mathbf{y}}_{par(par(i_p))} + \bar{\mathbf{x}}_{ind(par(i_p))} + \bar{\mathbf{x}}_{ind(i_p)} = \cdots \\ &= \hat{\mathbf{y}}_0 + \sum_{j\in path(i_p)} \bar{\mathbf{x}}_{ind(j)} \end{aligned} \tag{15}$$

The last equality in Equation (15) holds because, as explained previously, the roots of the BU and the TD trees are common so that $\hat{\mathbf{y}}_0 \equiv \tilde{\mathbf{y}}_0$.

Combining Equations (15) and (14) we can write:

$$\tilde{\mathbf{y}}_{i_p} - \mathbf{y}_{i_p} = M_{i_p}\mathcal{X} + c_{i_p} \tag{16}$$

so that

$$c_{i_p} = \sum_{d=1}^{D}\left[\left(\frac{1}{2}\right)^d \left(\sum_{i\in term(d)} \mathbf{y}_i\right)\right] - \mathbf{y}_{i_p}, \tag{17}$$

$$\mathcal{X} = [\bar{\mathbf{x}}_{1,1}^T, \bar{\mathbf{x}}_{1,2}^T, \cdots, \bar{\mathbf{x}}_{4,2}^T]^T \tag{18}$$

and

$$\begin{aligned} M_{i_p} = \big[&G_{i_p}([1,1]), G_{i_p}([1,2]), G_{i_p}([2,1]), \cdots \\ &\cdots, G_{i_p}([4,2])\big] \end{aligned} \tag{19}$$

where

$$\begin{aligned} G_{i_p}([a,b]) = \sum_{j\in path(i_p)} \delta(ind(j) - [a,b]) I_2 - \\ \sum_{d=1}^{D}\left(\left(\frac{1}{2}\right)^d \sum_{p\in node(d)} \delta(ind(p) - [a,b]) I_2\right) \end{aligned} \tag{20}$$

where I_2 is the 2×2 identity matrix and $\delta(A,B) = 1$ iff $A = B$ and 0 otherwise. Minimizing the L_2 norm of Equation (16) for all the leaf nodes, we get:

$$\sum_{p=1}^{P}\left(M_{i_p}^T M_{i_p}\right)\mathcal{X} + \sum_{k=1}^{P}\left(M_{i_k}^T c_{i_k}\right) = 0 \tag{21}$$

Equation (21) has a unique solution, so that:

$$\mathcal{X} = -\left[\sum_{p=1}^{P}\left(M_{i_p}^T M_{i_p}\right)\right]^{-1} \sum_{k=1}^{P}\left(M_{i_k}^T c_{i_k}\right) \tag{22}$$

Matrix $\sum_{p=1}^{P}\left(M_{i_p}^T M_{i_p}\right)$ encodes the structure of the parse trees and does not depend on the measured positions of the leaf nodes. It is in general rank deficient. Regarding the employed grammar of Table 1, it is sufficient to define one of the mean values on each non-recursive rule (for example $\bar{\mathbf{x}}_{2,1}$ and $\bar{\mathbf{x}}_{4,1}$). Thus, we set arbitrary values the particular $\mathcal{X}$ entries and solve for the remaining ones. The obtained solution, $\mathcal{X}_{est}$, is certainly a minimizer of Equation (8) and hence, it describes the observations in the LSE sense.

3.3 Covariance Estimation

Since we have estimated the mean positions, we can use the BU equations to estimate the positions of all the internal nodes of the parse tree.

As explained before (Equation (13)), we assumed that $\hat{\Sigma}_{k,1}$=$\hat{\Sigma}_{k,2} = \Sigma_{k,\delta\mathbf{y}}/2$, for all $k = 1,...,4$ and all we need is a way to estimate $\Sigma_{k,\delta\mathbf{y}}$. We proceed with the estimation using the following procedure, for all the rules in the grammar:

1. Pick a rule r_k.
2. Identify all the instances of the particular rule in the dataset, and let the corresponding parents be indexed as $k_1...k_N$.
3. Estimate the sample mean of $\delta\mathcal{Y}(k_j), j = 1,...N$ (Equation (9)), over the N instances of rule r_k.
4. Estimate the sample covariance $\Sigma_{k,\delta\mathbf{y}}$ of $\delta\mathcal{Y}(k_j), j = 1,...N$, over the N instances of rule r_k.
5. Choose: $\hat{\Sigma}_{k,1} = \hat{\Sigma}_{k,2} = \Sigma_{k,\delta\mathbf{y}}/2$.

3.4 Learning from a Single Image

Assume for a moment that we have achieved to detect the windows of a facade and we have constructed its parse tree (see Section 4). Since our ultimate goal is to construct a detector for this facade, we are limited to use a single image to produce the desired geometric model. However, due to the employed grammar, the second rule will appear only once in each facade. Additionally, windows are not spread on a regular grid in general, since the distance among windows may vary across a building and we would like to capture this behavior in the covariance matrices.

In order to to create a sufficient dataset per facade, we define a $n \times m$ segment as a part of the parse tree that contains n floors and m windows per floor. We construct a dataset that includes the original parsetree and all the possible 3×3 parsetrees. If this is not possible, in the case for example that n (or m) in the original parse tree is 2, we set n (and/or m) equal to 2, accordingly.

Conclusively, we learn one geometrical model per facade, using the original parse tree and the selected $n \times m$ segments, using the techniques described in Section 3.

4 WINDOW DETECTION

The proposed window detection framework produces the positions of the windows along with the horizontal and vertical period and a set of characteristic windows of each facade. Since we expect the detected windows to lie on a grid, it is straightforward to define their ordering on the plane and therefore, the corresponding parse tree (see also Figure 1). On the other hand, the produced periods and characteristic windows of each facade will be used for the detection of windows on our test images, in Section 5.

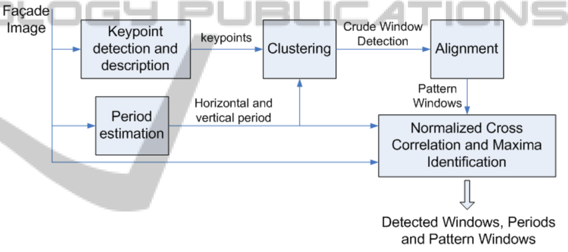
Façade Image
Keypoint detection and description
keypoints
Clustering
Crude Window Detection
Alignment
Pattern Windows
Horizontal and vertical period
Period estimation
Normalized Cross Correlation and Maxima Identification
Detected Windows, Periods and Pattern Windows

Figure 2: The window detection diagram.

We initially approximate the horizontal and the vertical period of each facade, namely T_x and T_y. This is achieved by cross-correlating the image with itself and calculating the distance between the two highest consecutive peaks along each dimension (Figure 3).

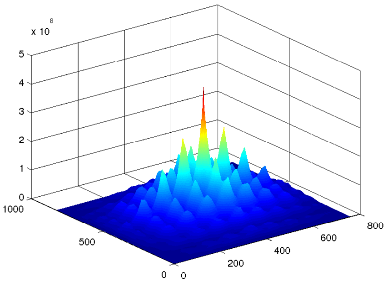

Figure 3: Cross correlation of an image with itself. The distance between two consecutive peaks along each dimension approximate the horizontal and vertical period.

In order to detect the windows we modify the bottom-up approach described in the PhD thesis of Olivier Teboul (Teboul, 2011). In particular we use the FAST (Rosten et al., 2009) corner detector to detect our interest points and we use the SIFT descriptor with the same scale and orientation for all points.

From the large pool of the detected keypoints, we have to determine which of them refer to windows and choose one keypoint per window, for as many windows as possible. This keypoint should describe the same part of the window for every facade (a particular corner for example). To this direction, we utilize a three step clustering scheme using the algorithm proposed in (Komodakis et al., 2008). We justify the choice of this algorithm for two reasons. First of all, the number of clusters K is an output of the formulation and in our case, K is unknown. Secondly, cluster centers are necessarily members of the data set. We use the L_1 norm as our distance function.

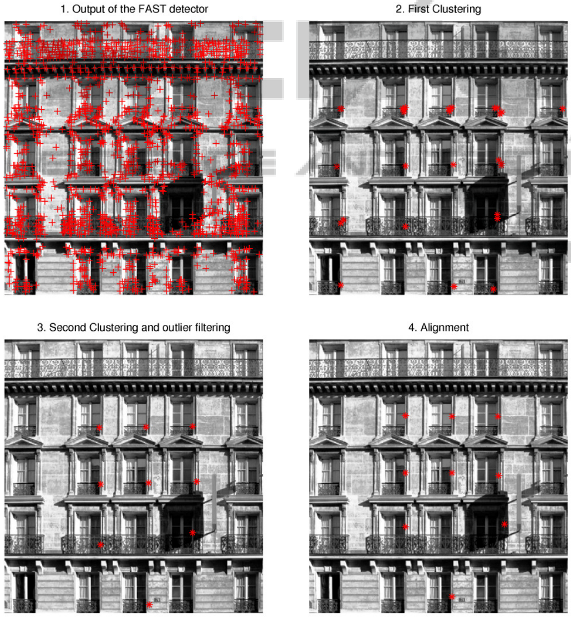
1. Output of the FAST detector
2. First Clustering
3. Second Clustering and outlier filtering
4. Alignment

Figure 4: The first clustering produces a number of clusters, some of which refer to windows. We can see such a cluster in the second image. In order to obtain one keypoint per window, we perform the second clustering and we drop potential outliers (third figure). We can see that in this case, the outlier detection disposed several good windows. This happened because this facade is not strictly periodic. The windows on the left and right edges are further from their neighbors, than the ones in the middle and T_x expresses the period in the middle windows. Finally in the fourth figure, the detected windows are aligned. We can see that the shaded window did not align very well. On the contrary, the other window of the same floor moved to align with the rest detected windows. The scope of this procedure is to choose the patterns for the normalized cross correlation that follows, in both the training and the testing phase.

First Clustering. In the first clustering phase we cluster the SIFT descriptors. The output of the first phase is a number of clusters, some of which do not refer to windows. On the other hand, since it is natural for neighboring keypoints that lie on the same edge to have similar SIFT descriptors, clusters that do refer to windows will have more than one keypoints per window, in general (see the second image of Figure 4).

Second Clustering. Our second clustering phase aims at choosing exactly one keypoint per window. We cluster the positions of the members of each cluster from the first clustering, so that windows are described by the emerged cluster centers. We assume that cluster centers should satisfy the estimated periods. Therefore, we examine each center and if there are no other centers that lie in a horizontal (vertical) distance close to T_x (T_y), we consider the particular center to be an outlier and we drop it (see the third image of Figure 4). Although this is too strict and we might drop points that lie on windows, we are still interested to make a crude estimation that will be enhanced later.

Third Clustering. Our third clustering phase aims at identifying which of the initial clusters refer to windows. We separately cluster the horizontal and vertical distance between all the pairs of cluster centers of the second phase. For each dimension, if we discover a cluster with respectable cardinality whose distance is close to the corresponding period (T_x and T_y), we consider that the cluster from the first phase is referring to windows. For all the potential clusters that refer to windows, we choose the one with the largest cardinality.

Once we have chosen which of the initial clusters we will use, we consider the corresponding cluster centers from the second phase as an initial crude estimation of some window positions. If we have detected N windows, we choose N image segments of $T_x \times T_y$ area, centered at the detected positions $y_1, \ldots, y_N$.

Alignment and Choise of Pattern Windows. Since we want the window centers to represent the same part of the window, we perform an iterative alignment of the image segments. For each segment i, we compute the normalized cross-correlation of i with the rest $j = 1, 2, \ldots, i-1, i+1, \ldots, N$ segments. For each one of the j segments, we locate the maximum y_j^0 and we deviate y_j towards y_j^0, i.e.,$y_j' = x_j + d(y_j^0 - y_j)$, where $0 < d < 1$. We perform the same procedure for 60 epochs until convergence. If some of the window centers fail to converge or fall out of the image boundaries, we drop them. At the end of the alignment we have M windows of size $T_x \times T_y$ centered at $y_1^a, \ldots, y_M^a$. Let's call them patterns.

Final Detection and Ordering of the Windows on the plane. So far we have managed to detect some windows in various locations of the facade and we want to detect as many as possible. To this direction, we compute the normalized cross-correlation of the original image with each one of the M patterns. We aggregate the results using the max operator (Figure 5) and we search for the maxima that correspond to windows. In particular, we assume that one of the highest peaks is a window. If the coordinates of this peak are $[p_x^w \;\; p_y^w]^T$, we search for new windows in an area of $T_x \times T_y$, around $[p_x^w \pm T_x \;\; p_y^w]^T$ and $[p_x^w \;\; p_y^w \pm T_y]^T$. The maxima within each one of these four areas are our new windows. We continue the same procedure recursively, making sure that we do not search twice within the same area. The output of this procedure is our final choice of windows (Figure 6), along with their ordering. The order of windows is defined by the progress of window searching. If, for example, we detect the i-th window of the j-th floor at position $[p_x^w \;\; p_y^w]^T$ and then we manage to detect another window in the area around $[p_x^w + T_x \;\; p_y^w]^T$, the new window will be the $i+1$ window of the j-th floor. If on the other hand the new window is detected in the area around $[p_x^w \;\; p_y^w - T_y]^T$, the new window will be the i-th window of the $j-1$ floor.

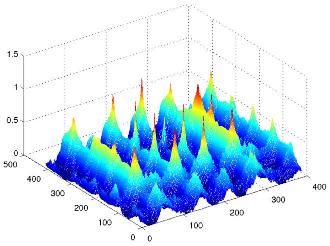

Figure 5: Cross Correlation of the original image with the patterns. The peaks indicate the positions of the windows.

By examining the ordering, we can argue if this procedure has missed any windows. There are two cases of occlusion. In the first case we identify gaps in the ordering. For example if the ordering for one floor is [1 2 4 5], we assume that the third window is missing. In the second case, there are floors that have less windows than the maximum number of windows per floor, or the first window of a floor is missing. In the first case we interpolate the missing windows between the previous and the next detected ones. In the second case, we extrapolate the missing windows from the last or the first detected window of the floor that exhibits the occlusion, according to T_x.

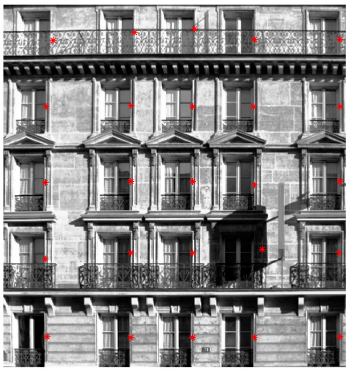

Figure 6: The final detected windows.

5 RESULTS

We apply our approach on the *Ecole Centrale Paris Facades Database* (Teboul, 2012) produced and maintained by Olivier Teboul. In particular we focus on the *Paris, France* collection of 215 rectified images. From each one of these images we crop a small segment and our main goal is to discover the origin of a distorted version of each segment wrt the original 215 images. Therefore, we want to compute 215 geometric models and evaluate each test segment against all models.

In order to compute these models, for each facade we apply the window detection techniques described in Section 4, we produce the parse trees (along with the horizontal and vertical periods and the pattern windows) and we learn the parameters, as described in Section 3.

Then for each one of the cropped test segments we follow similar steps as in the window detection phase; we evaluate the normalized cross correlation of the segment with each of the M patterns of the particular model, we aggregate using the max operator and we search for windows using the periods T_x and T_y that were estimated for the production of the particular model. Therefore we detect some windows and their ordering and we produce the parse tree.

We proceed by utilizing an appearance model. In particular, when our searching algorithm detects a window, we crop a $T_x \times T_y$ area centered at the detection point and we compare it with the pattern window that gave the maximum cross-correlation value at the particular position. Let M' be the number of the detected windows in the segment. We compute the mean squared error S_j between all the pixels of the j-th detected window and the corresponding pattern and we define the appearance likelihood of the image to be:

$$P_a = \frac{\sum_{i=1}^{M'} S_i}{M'}. \tag{23}$$

In order to estimate the geometric likelihood, we estimate the frames of the non-terminal nodes using the BU equations and we would normally use Equation (1). However, geometric likelihood is a decreasing function of the size of the parse tree. In order to compensate the fact that different models detect different number of windows, we evaluate the geometric likelihood as:

$$P_g = \frac{\log P(t|G)}{M'}, \tag{24}$$

where $P(t|G)$ is evaluated from Equation (1).

The overall expression of the likelihood is

$$P = P_g - kP_a, \tag{25}$$

where k is a normalizing factor.

We perform the evaluation 25 times by gradually blurring and adding noise to the segments. In particular we use a Gaussian kernel with $\sigma_k = 0.8, 1.3, 1.8, 2.3, 2.8$ to blur the images and we add random noise from a uniform distribution in $[0, N]$, where $N = 90, 130, 170, 210, 250$. With reference to Figure 7, the original segment is blurred with the Gaussian filter H_1 whose variance is σ_k. The random noise is initially filtered with a Gaussian filter H_2 whose variance is 1. We normalize the filtered noise by subtracting its mean value and we add it to the blurred segment.

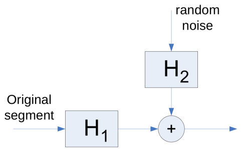

Figure 7: Blurring and adding noise to the original image segments

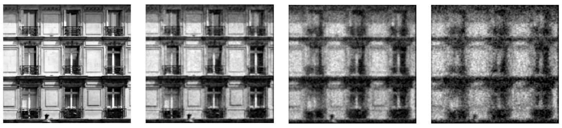

Figure 8: The original test image and three noisy instances.

For each one of the 25 noise scenarios, we compute the number of correct classifications and the mean reciprocal rank (MRR) (Voorhees, 1999) for two cases. In the first one, we use only the appearance model and in the second one we use both the appearance and the geometric model. If there is no noise present, our appearance model manages to classify all the samples correctly, making the use of our geometric model unnecessary. However, we can see in figures 9 and 10 that the contribution of the geometric model becomes significant, as the noise increases. The less our appearance model achieves to discover the correct classification, the more our geometrical model contributes to the overall performance. As a final remark, we can see that the surfaces that represent the combined use of both models are constantly above the surfaces that represent the explicit use of the appearance model.

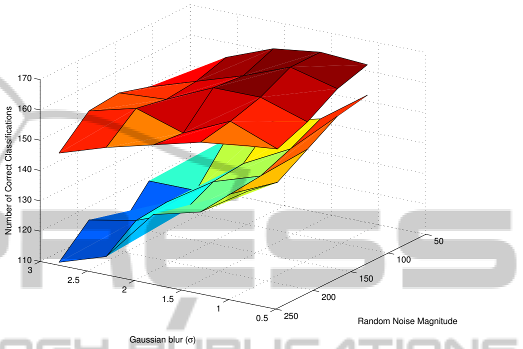

Figure 9: Number of correct classifications against noise. The upper surface corresponds to the combined use of appearance and geometry. We see that under the presence of noise, the contribution of the geometric model is significant in the overall classification rate.

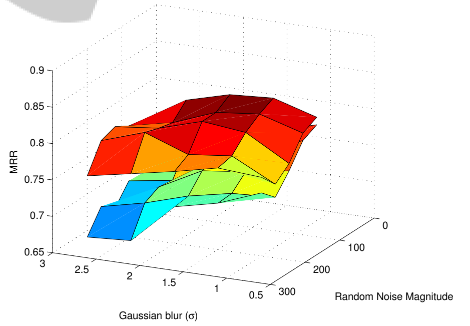

Figure 10: MRR against noise. As in Figure 9, the upper surface corresponds to the combined use of appearance and geometry. The more the noise, the more necessary it is to take advantage of the geometric information.

6 CONCLUSIONS

This paper examined the effectiveness of PGG's in modeling and classifying facades. We employed a description where terminal symbols correspond to windows, generated by the geometric grammar. We derived closed-form expressions for estimating the geometric parameters of our grammar and we managed to learn the parameters from a single image. We developed a window detection algorithm and we applied our framework on a dataset of 215 rectified facades. Our geometric model was tested against a proposed appearance model. The performance of the proposed methodology was very promising, as the simultaneous use of the geometric and the appearance model constantly achieved better classification performance than the exclusive use of the appearance model itself, in all examined cases. Results justify our intuition to use grammatical models for facade classification.

The proposed method requires the a priori definition of the producing rules but not their geometric statistics. Despite the fact that the simplicity of the adopted grammar proved to be very effective, more complex grammatical models could be used instead, in order to capture the different horizontal periodic patterns that may exist in facades. Moreover, we currently work on the extension of PGGs to include rotation and scale relations, so that they could be applied to different object classes, such as plants, aerial urban images, etc.

ACKNOWLEDGEMENTS

The authors would like to thank Nikos Komodakis for providing the source code for the clustering algorithm that was used in Section 4.

REFERENCES

Baumann, S. (1995). A simplified attribute graph grammar for high level music recognition. In *ICDAR*, volume 2, pages 1080–1083. IEEE.

Felzenszwalb, P. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *IJCV*, 61(1):55–79.

Felzenszwalb, P. and Schwartz, J. (2007). Hierarchical matching of deformable shapes. In *CVPR*. IEEE.

Feng, H. and Zhu, S. C. (2005). Bottom-up/top-down image parsing by attribute graph grammar. In *ICCV*, 10, pages 1778–1785. IEEE.

Fergus, R., Perona, P., and Zisserman, A. (2006). Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 71:273–303.

Fu, K. S. (1981). *Syntactic Pattern Recognition and Applications*. Prentice Hall.

Holliday, D. J. and Samal, A. (1994). Recognizing plants using stochastic l-systems. In *ICIP*, volume 1, pages 183–187. IEEE.

Holliday, D. J. and Samal, A. (1995). A stochastic grammar of images. *Object recognition using L-system fractals*, 16:33–42.

Komodakis, N., Paragios, N., and Tziritas, G. (2008). Clustering via lp-based stabilities. In *NIPS*.

Lewis, H. R. and Papadimitriou, D. (1998). *Elements of the Theory of Computation*. Prentice Hall.

Liu, Y., Collins, R., and Tsin, Y. (2004). A computational model for periodic pattern perception based on frieze and wallpaper groups. *Transactions on PAMI*, 26(3):354–371.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *IJCV*, 2(60):91–110.

Muller, P., Wonka, P., Haegler, S., Ulmer, A., and Gool, L. V. (2006). Procedural modeling of buildings. *ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH*, 25:614–623.

Rekers, J. and Schurr, A. (1997). Defining and parsing visual languages with layered graph grammars. *Journal of Visual Language and Computing*, 8(1):27–55.

Ripperda, N. and Brenner, C. (2009). Application of a formal grammar to facade reconstruction in semiautomatic and automatic environments. In *12th AGILE Conference on GIScience*.

Rosten, E., Porter, R., and Drummond, T. (2009). Faster and better: A machine learning approach to corner detection. *Transactions on PAMI*, 32(1):105–119.

Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., and Dellaert, F. (2008). Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*. IEEE.

Teboul, O. (2011). *Shape Grammar Parsing: Application to Image-based Modeling*. PhD thesis, Ecole Centrale Paris.

Teboul, O. (2012). *Ecole Centrale Paris Facades Database*. http://vision.mas.ecp.fr/Personnel/teboul/data.php.

Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., and Paragios, N. (2011). Shape grammar parsing via reinforcement learning. In *CVPR*. IEEE.

Teboul, O., Simon, L., Koutsourakis, P., and Paragios, N. (2010). Segmentation of building facades using procedural shape prior. In *CVPR*. IEEE.

Voorhees, E. M. (1999). Trec-8 question answering track report. In *8th Text Retrieval Conference*.

Wonka, P., Wimmer, M., Sillon, F., and Ribarsky, W. (2003). Instant architecture. *ACM Transactions on Graphics*, 22(3):669–677.

Wu, C., Frahm, J., and Pollefeys, M. (2010). Detecting large repetitive structures with salient boundaries. In *ECCV*. Springer.

You, F. and Fu, K. S. (1979). A syntactic approach to shape recognition using attributed grammars. *Transactions on SMC*, 9:334–345.

Zhu, L., Chen, Y., Freeman, W., and Yuille, A. (2010a). Latent hierarchical structural learning for object detection. In *CVPR*. IEEE.

Zhu, L., Chen, Y., Torralba, A., Freeman, W., and Yuille, A. (2010b). Part and appearance sharing: Recursive compositional models for multi-view multi-object detection. In *CVPR*. IEEE.