Towards Semantic City Models

LUC VAN GOOL, ANDELO MARTINOVIC, MARKUS MATHIAS, Leuven, Belgium

ABSTRACT

3D city modeling is a very demanding task. It suffers from the same problems as general bottom-up 3D acquisition processes. Whatever the 3D acquisition system, there are always objects and surfaces that yield meager performance. Often the problems are fundamental and cannot be resolved through more sophisticated bottom-up processing. If however, one knows what kind of objects are to be modeled – e.g. buildings – strong prior knowledge can be invoked. Inverse procedural modeling is a case in point. It yields more compact and more realistic models, yet requires a grammar to be created and then activated for the style at hand. The paper discusses style classification and the (initial) use of more generic architectural guidelines as ways to mitigate the problems with procedural grammars.

1. INTRODUCTION

3D data acquisition, including that for city modeling, has traditionally been handled in a very bottom-up fashion. One of the noteworthy developments has been the increasing success of image-based mobile mapping. Increasingly, platforms only equipped with cameras and a GPS receiver extra 3D data comparable in quality to that obtained with more heavily equipped platforms, carrying e.g. a laser scanner, and INS unit, several cameras, and GPS. Such image-based capture is a commercial fact by now.

Yet, a new wave of innovations can be expected to arrive soon, again driven by the cameras on the mobile mapping platforms. Indeed, over the last decade or so, the computer vision community has developed methods to detect instances of object classes in images, e.g. to find all the people, trees, cars, etc. in images. This is useful *per se*, also for city modeling, as cities obviously do not only contain buildings but also such objects. Yet, at least as important is the role the object detection can be expected to have on the 3D acquisition process itself. This process still has many pitfalls, whatever capturing technology is used. As an example, image-based 3D capture finds it difficult to handle untextured or specular surfaces, thin objects like poles or fragmented volumes like tree canopies. Yet, strong priors can be used as soon as one knows what it is that one is trying to model. In this paper, we give a short overview of our semantic modeling work for buildings. In particular, we describe our inverse procedural approach.

Procedural modeling describes buildings through the instantiations of a series of rules. Together these rules form a grammar. Typically such grammar starts from the overall structure of a building, to then add more and more detail to its geometry as one applies rule after rule. Rules may e.g. add windows to walls, or ledges along a floor. A grammar is designed for a specific style and its creation requires quite some expertise about that style. Grammar creation therefore may be non-trivial. As such, a procedural modeling approach to architecture is a graphics-like process, where a user wants to create a building model. City modeling requires the opposite: starting from (in our case images of) the existing reality, one wants to create procedural models of structures as-built. We refer to such process as *inverse procedural modeling*. When producing a procedural modeling of a building through inverse procedural modeling, one needs to select the appropriate rule to apply, and also their parameters. The search space of this optimization problem tends to be quite large.
Grammars being non-trivial to create and then the optimization for inverse procedural modeling being involved, why would one want to produce procedural modeling of buildings in the first place? For

one, procedural models are very compact, yet can be detailed. Thus, detailed models would still fit in reasonably sized memory. As a matter of fact, one can infer aspects that allow for a more realistic rendering than would be possible from pure 3D capturing. For instance, windows can be made reflective and to lie a bit deeper than the façade, even if such small depth difference would be difficult to infer from the 3D data. Moreover, procedural models are rich in terms of semantics. Semantic concepts like windows, floors, doors, balconies, etc. are made explicit. Procedural city models can therefore be explored at a high semantic level. One could ask how many buildings have at least 10 floors, what the total window area in a city district is, etc. In the case of animation, the model is prepared to let virtual people only walk via doors, rather than through walls, and to automatically determine the flux in or out of a building based on its size.

The structure of the remainder of the paper is as follows. First, we give a short overview of the related literature (section 2). Section 3 discusses inverse procedural modeling in a bit more detail, for the case of classical temples. Section 4 then concisely describes how appropriate style grammars can be identified automatically, through the visual recognition of the style of a building. Section 5 continues with showing that – for most buildings – one could actually work from rather style-independent architectural principles. This is exemplified for the case of façade parsing. Section 6 concludes the paper.

2. RELATED WORK

Urban reconstruction. For a more extensive overview of the city modeling, we refer the reader to the survey of (Musialski et al., 2012). We restrict our discussion to the contributions that are closest to the approach in our paper, e.g. to approaches that also focus on ground imagery. First, we want to mention several of the papers that focus on preprocessing steps that should also be applied before some or our techniques could kick in. (Zhao et al., 2010) presented an algorithm that segments ground images into buildings, grass and sky, followed by the partitioning of buildings into individual facades. In this paper, we do not consider such segmentation and assume it has been carried out beforehand (mainly for Section 5). The facade splitting problem was also studied by (Wendel et al., 2010; Recky et al., 2011), who exploit repetitive patterns to get cues for the transitions between the facades of different buildings. In previous work (Mathias et al., 2011a), we developed a scene classification step that identifies input images containing facades. After automated image rectification, buildings are split into individual facades.

As to the modeling of the actual buildings, (Xiao et al., 2008; Xiao et al., 2009) target realistic visualization with a low level of semantic encoding in the reconstruction. In their work, facades are represented with planes or simple developable surfaces. On the other hand, several approaches use higher-order knowledge for building reconstruction. Probabilistic approaches find their origin in the seminal work of (Dick et al., 2004), where a building is treated as a combination of parameterized primitives. An expert is needed to set the model parameters and prior probabilities, while inference is performed using a Markov Chain Monte Carlo (MCMC) approach. Another early grammar-based approach that fitted stochastic grammars with MCMC was (Alegre and Dellaert, 2004), while a bit later (Ripperda and Brenner, 2006) used rjMCMC for the construction of a grammar tree. Assumptions about the existence of a facade grid structure were employed in (Korah and Rasmussen, 2008; Yang and Förstner, 2011; Han et al., 2012). Multiple grids are estimated in the work of (Shen et al., 2011).

(Inverse) procedural modeling. In the introduction we have already pointed out that we use assumptions about building and façade structures in the form of a style grammar. We also introduced

the concept of inverse procedural modeling (IPM) as an umbrella term for approaches that attempt to discover the rules and their parameter values of the procedural models of existing buildings. Most IPM methods have confined the framework to cases where the style grammar is known in advance (i.e. the set of relevant rules is considered given). The relevant rules and parameters are then still to be selected. This top-down model is then fitted to bottom-up cues derived from the data. The first such method is probably (Mueller et al., 2007). The authors assume a certain degree of facade regularity and fit procedural grammar rules to the detected subdivision of the facade. The approach was extended in (Van Gool et al., 2008), where images with strong perspective distortions are used to infer vanishing points and 3D information from a single image. (Han and Zhu, 2009) propounded a hybrid bottom-up/top-down approach. (Vanegas et al., 2010) used a simple grammar for buildings that follow the Manhattan world assumption. A grammar was fitted from laser-scan data in (Toshev et al., 2010). (Mathias et al., 2011b) reconstructed Greek Doric temples using template procedural models. We concisely summarize this approach in Section 2. (Teboul et al., 2011) presented an efficient parsing scheme for Haussmannian shape grammars using Reinforcement Learning. Recently, (Riemenschneider et al., 2012) proposed CYK parsing. Although these approaches produce good facade parsings, they assume strong priors.

As already indicated, all these grammar-based methods share a common drawback. They assume that a manually designed grammar is available from the outset. This is a serious constraint, as it limits the reconstruction techniques to a handful of building styles for which pre-written grammars exist. Creating style-specific grammars is a tedious and time-consuming process, which can typically only be achieved through collaboration between architectural experts in combination with people verse in the writing of the rules for the grammar type of choice. We will come back to this limitation later, especially in Section 4. Moreover, if a style-specific grammar is to be of any use in the context of automated, large-scale city modeling, then the style of the buildings need to be recognized swiftly, as to activate the relevant style. In the next section, we discuss the state-of-the-art in architectural style classification.

Architectural style classification. As matter of fact, so far very little research has been carried out in the field of automated architectural style identification. (Romer and Plumer, 2010) aims at detecting buildings of the Wilhelminian style from a simplified 3D city model. Their approach is based on a few coarse features (building footprint and height) and exploits no image support. This would only work in a city where buildings of that epoch are all of the Wilhelminian style.

It stands to reason that better classification can be achieved if further image information is used. Available image classification systems such as the one of (Bosch et al., 2008) often distinguish between images whose appearances are very different. For instance, much attention has been paid to distinguishing indoor from outdoor scenes (Payne and Singh, 2005, Szummer and Picard, 2002). Conversely, facade pictures share many common features no matter their styles. For instance, colour or edges would seem to be overly weak cues when wanting to distinguish Haussmannian from Neoclassical buildings. The system of (Mathias et al., 2011a) was the first to tackle the problem of image-based architectural style identification. Their approach provides a systematic and comprehensive way of estimating the building style from a single street-side image, incorporating steps of scene classification (where are the facades?), image rectification, facade splitting and style classification.

3. INVERSE PROCEDURAL MODELING WITH FIXED GRAMMARS

This section describes an approach that is explained in more detail in (Mathias et al., 2011b). That paper discusses the 3D modeling of classical Doric temples. We will use this as an example case for inverse procedural modeling. We have developed a grammar for such buildings and the IPM pipeline described here knows this to be the grammar that has to be used in advance. Classical temples, like the more recent Hausmannian architecture, conform to strict architectural rules as described in (Summerson, 1996). These rules have been converted into the shape grammar.

Our IPM pipeline creates 3D building models – i.c. of classical temples – by combining image-based Structure-from-Motion (SfM), building element ('asset') detectors (e.g. of elements like pillars and capitals in our application example), and inverse procedural modeling. The latter component incorporates a shape grammar interpreter that drives the process. The usage of asset detectors replaces fragile segmentation processes by top-down, semantic influences. This has only become possible by leveraging recent progress in visual object class recognition. The detectors are trained from single images, where a user has to draw bounding boxes around exemplars of the intended class. This is a rather tedious process and it is therefore good if part of such training can be automated. More about this is to follow soon. It is also important to note that our images, which have been mined from the Internet, often do not allow for a complete SfM reconstruction. In such case, the strong shape priors from the grammar combined with the asset detections obtained from single images, are a necessary addition to the SfM approach.

The approach combines the robustness of a top-down grammar-based approach with the flexibility of the bottom-up SfM image-based approach. The extra building asset detectors act as mid-level catalyzers that help speed up the interaction between bottom-up and top-down processing. They can be regarded a top-down process from an in-between, semantic level. By proposing such pipeline, our main contributions have been the following. (1) The reconstruction process is guided by the grammar. Instead of the developers having style-specific guidelines in mind when producing the system, a grammar interpreter tool renders the process more generic. Thus, the pipeline uses a specific

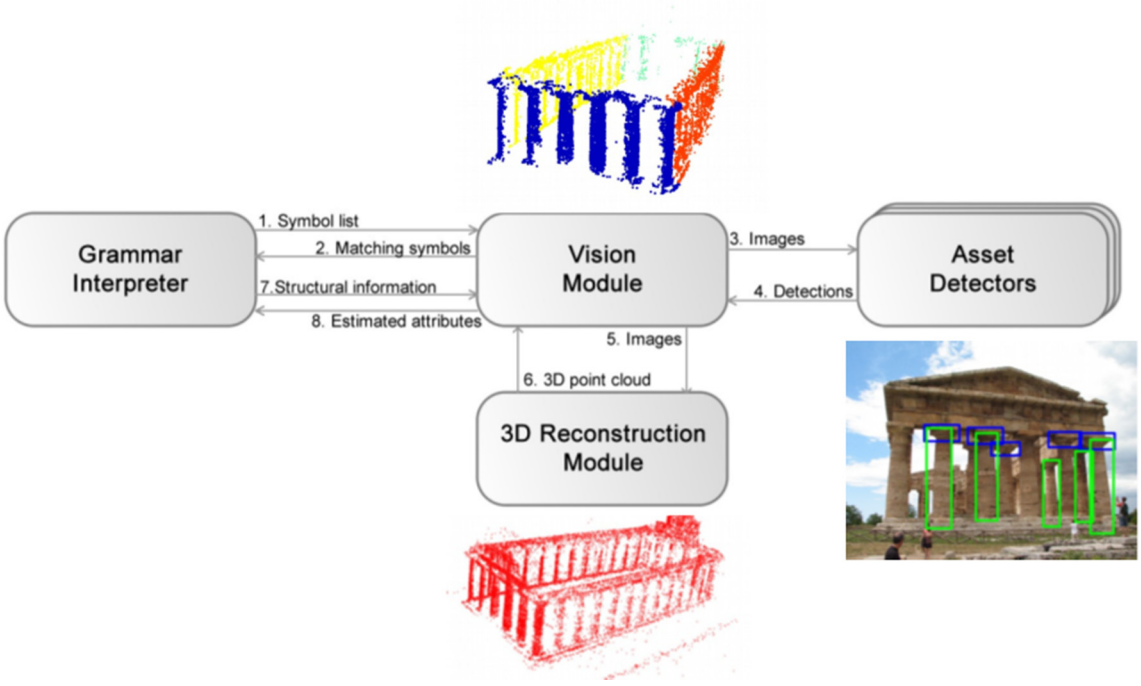

Figure 1: Reconstruction of Doric temples combining SfM, column and capital detectors, and an appropriate classical temple grammar.

grammar, but can be used in combination with another one just the same. No style knowledge is baked into the pipeline itself. It is the grammar that decides on what to do when. Moreover, structures that may not even be visible can be filled in. (2) Rather than relying on fragile segmentation processes to kick-start the semantic analysis, the grammar uses the available detectors to assign initial semantic labels to image regions. (3) The system learns from its previous results. Asset detectors self-improve by using modeling results as additional training material. If the entire modeling sequence has been successfully closed, this is a strong indication that the surviving detectors are correct. This also allows us to start with rather generic asset detectors, which have not been developed uniquely for the targeted style, but can then be specialized towards style-specific detectors. This is very useful, as this avoids having to train detectors for each style separately, which would a very time-consuming process given the need for manual annotations (bounding boxes around the exemplars).
As a result, not only is the 3D modeling helped by the detectors, but the training of detectors is also helped by the 3D modeling.

Figure 1 shows how the parts of the system interact. First (1), the grammar interpreter initializes the vision module with a list of shape symbols automatically extracted from the grammar. They are then compared with the list of symbols that represent trained asset detectors from our database. The matching symbols (assets) are identified, reported to the grammar interpreter (2) and the detection process is initialized for those assets resulting in detection bounding boxes in all input images (3-4). The images are also fed into the 3D reconstruction module ARC3D (Vergauwen and Van Gool, 2006) to obtain a sparse 3D point cloud and the camera parameters from the building (5-6). For the matched symbols (detectable assets) the grammar interpreter parses the grammar to find structural information like spatial configuration or repetitions of these symbols (step 7). The vision module uses a plane fitting algorithm to extract the dominant planes of the building. The detections from all images are projected into 3D and re-weighted based on consensus in 3D and the structural information. The output of this vision module are the sizes of the detected assets and their color, the footprint for the building and the parameters for the structural configurations (step 8). Then the building can be instantiated by the grammar interpreter by using the extracted parameters.

We show the resulting reconstructions of three Greek Doric temples in Figure 2: the Temple of Athena (also known as Temple of Ceres); the Temple of Poseidon, where the latter two are both archaic Doric temples in the ancient city of Paestum, and the Parthenon in Nashville, a full-scale replica of the

Figure 2: Reconstruction of Greek Doric temples (Mathias et al. 2011). Left: Temple of Athena, Paestum, Italy. Center: Temple of Poseidon, Paestum, Italy. Right: Parthenon replica, Nashville, USA.

original Parthenon in Athens. The figure shows original images and final the 3D temple models superimposed. Note how in the former two cases the grammar is strong enough a driving force to allow for the completion of these ruined temples.

4. ARCHITECTURAL STYLE RECOGNITION

In the previous section, we still had to assume that the system would know about the relevant style grammar before the modeling pipeline would start. This is OK for individual landmarks, but when a mobile mapping system is driven through a city, many buildings are observed and the style often changes between them, even within a single street. It would therefore be good if such mobile system would be capable of automatically recognizing the style of buildings, such that it can in each case activate the appropriate grammar.

In (Mathias et al., 2011a) we proposed a 4-stage method for the automated classification of architectural building styles. We demonstrate this approach on three distinct architectural styles: Flemish Renaissance, Haussmannian, and Neoclassical. We also consider a class `other'. Probably to the dismay of the experts, we use a loose interpretation of these architectural terms, as our main goal is to enable automated 3D modeling pipelines to get sufficient prior information to succeed. Hence we actually focus on the categorization of building appearance, not actual historic provenance. For example, our Flemish Renaissance dataset also contains buildings from the Flemish Renaissance Revival style, which share their visual features.

Last but not least, we have also created a publicly available dataset of facade images spanning the aforementioned three styles. The images were taken from example buildings in the cities of Leuven, Antwerp and Brussels, in Belgium. This database will allow other researchers to test and compare their approaches on the same images.

As a matter of fact, if mobile mapping is to be automated, other preprocessing steps need to be automated as well. For instance, the system should know where there are buildings (and e.g. not vegetation) and it needs to identify single facades, which then are to be rectified. We have worked on the automation of these steps as well.

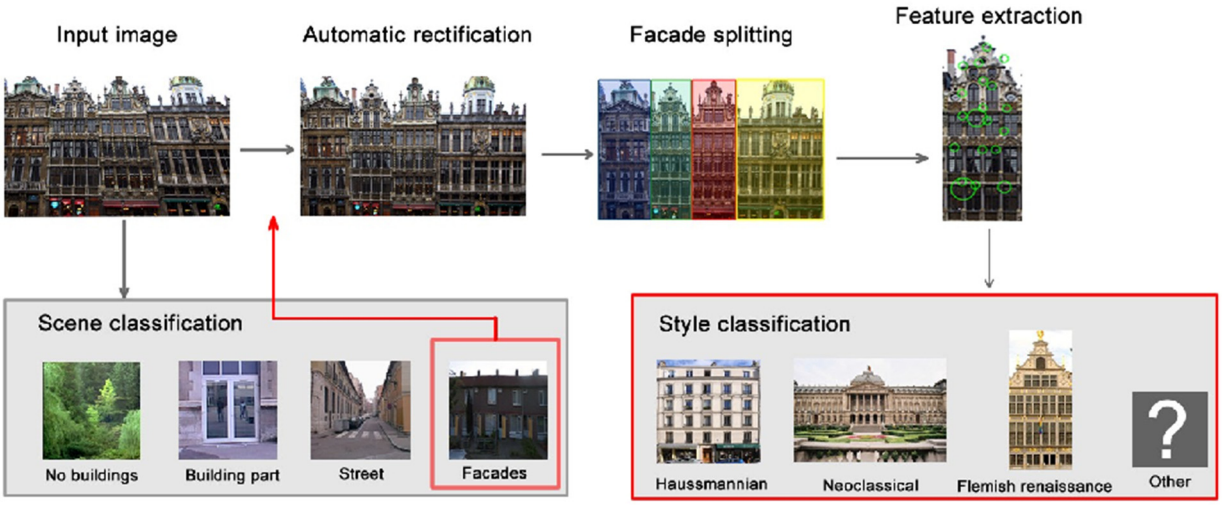

Figure 3: A system for architectural style recognition (Mathias et al., 2011a).

Figure 3 gives an overview of our preprocessing system. The first step determines if the image actually contains building facades and where. If this condition is met, the façade portions are rectified, as the images of buildings taken from the street usually contain significant projective distortions. After the image has been rectified, we still face the problem of segmenting individual facades out of the typical long, unbroken building blocks. This is important as the architectural style may vary from facade to facade. We use edge information to find individual building separators. For more details on façade segmentation we refer to the original paper.

As to the determination where the buildings are, we actually distinguish between the four most common cases in street-side imagery:

- No buildings – images not containing any buildings. Typical examples in urban scenarios are parks, gardens and waterfronts.
- Street – images containing facades captured at a large angle from the facade planes, occurring when the camera orientation coincides with the street direction (forward looking cameras on the mobile mapping van).
- Facades – images containing one or more whole facades (typically found in the obliquely forward looking cameras).
- Building part – images containing only a small part of a facade, not enough for a complete reconstruction (often the case for the sideways looking cameras).

We used a similar approach to (Torralba et al., 2003). The extracted features correspond to a steerable pyramid of Gabor filters, tuned to 4 scales and 8 orientations. Filter outputs are then averaged over a 4x4 grid. This produces a feature vector of 512 features. Classification is performed using a Support Vector Machine (SVM) with a Gaussian radial basis kernel function. The SVM is trained using a one-versus-all approach.

Buildings	None	Part	Street	Facades
None	100	0	0	0
Part	2.8	85.6	2.4	9.2
Street	0.8	1.2	98	0
Facades	0	7.2	0.4	92.4

Table 1: Results in scene classification.

From this result table, one can see that most classes are well distinguished from others. Misclassification mostly occurs between 'Building part' and 'Facades', as could be expected given their high visual similarity.

To differentiate between the different styles, namely "Flemish renaissance", "Haussmannian", "Neoclassical" and "Unknown", we got convincing results using the Naive-Bayes Nearest-Neighbor (NBNN) classifier proposed by (Boiman et al., 2008). Despite its simplicity, it has many advantages. This non-parametric classifier does not need time-consuming offline learning and, by design, it can handle many different classes. This means that new styles can easily be added. Furthermore it avoids over-fitting, which is a serious issue for learning-based approaches.

We cross-validated our style detector, using both SIFT (Lowe, 2004) and SSIM (Shechtman and Irani, 2007) feature descriptors. Our dataset contains 949 images: 318 background facades (i.e. facades belonging to none of the trained styles), 286 images for Neoclassical, 180 for Haussmannian and 165 for Flemish Renaissance. We have taken these images ourselves, except for the Haussmannian style images that come from (Teboul, 2010).

Style	Haussman	Neoclassical	Renaissance	Unknown
Haussman	0.98	0	0	0.02
Neoclassical	0.02	0.76	0	0.22
Renaissance	0	0	0.59	0.41
Unknown	0.03	0.005	0.005	0.96

Table 2: Results in style classification.

Table 2 shows the confusion matrix after cross-validation for the case with SIFT descriptors. This feature choice yielded the best performance throughout our experiments. While the Haussmannian style is clearly separated from other classes, many building of the Flemish Renaissance type are classified as ”Unknown“. Probably in part due to the fact that we have the least number of images for the Flemish Renaissance style, our implicitly derived definition for that particular class is still quite imprecise, where the great but sparsely sampled diversity of the facades of that class could not steer the classification process sufficiently well. As said, SIFT features outperformed SSIM features in our case: the mean detection rate of the SIFT features was 84% while for the self-similarity descriptor (SSIM) it reached only 78%.

Figure 4 shows the regions of the SIFT interest points colored in different colors. The colors indicate to which style the given feature had the minimum distance. The colors associated with the appropriate styles clearly dominate the images. The features that respond correctly for the style at hand are mostly attached to architectural elements that are typical for that style, e.g. the features responding to the capitals in a neoclassical building.

A current limitation is that the system focuses on facades, i.e. on dominantly planar structures where 2D image features are the most obvious candidates. For more complicated buildings such as landmarks (public buildings, churches, museums, …), it stands to reason to expand the feature set towards 3D shape features as well, which then can also capture volumetric aspects (but would probably still benefit from façade image features).

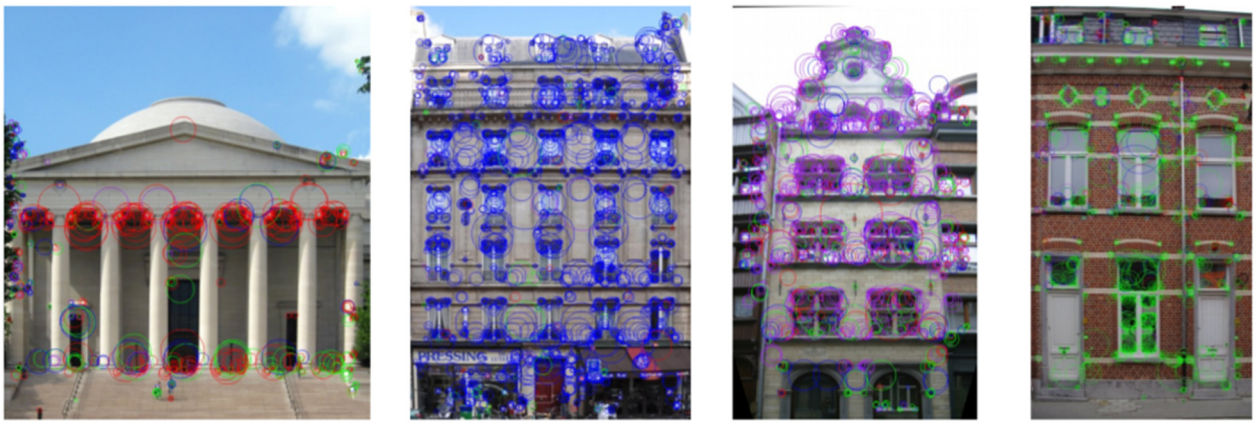

Figure 4: Detected, style-specific features. Red: Neo-classical, blue: Hausmannian, purple: Flemish Renaissance, green: other. The figure shows the confusion for the Flemish Renaissance building third from the left.

5. USING WEAK ARCHITECTURAL PRINCIPLES IN THE ABSENCE OF STYLE INFORMATION

Even if the style of a building can be recovered, the production of many style-specific grammars remains non-trivial. Thus, there are recent attempts to by-pass the initial need for such grammars and to start with weaker intuitive or learnt priors (Martinovic et al., 2012, Dai et al., 2012). Here we discuss automated façade parsing, i.e. the subdivision of facades into their main components (windows, doors, floors, balconies, …) without needing a strong grammar from the start.

Our proposed facade parsing method (Martinovic et al., 2012) consists of three distinct layers. In the first, a supervised training method learns the facade labeling based on an initial over-segmentation. For this purpose we utilize the recently developed Recursive Neural Networks (RNN) (Socher et al. 2011). In the middle layer we introduce knowledge about distinct facade elements, such as doors and windows. In the third and top layer, the raw RNN output is then combined with information coming from object detectors trained to detect architectural elements. Figure 5 gives a schematic overview of the system (from left to right rather than bottom to top).

We pose the merging of RNN and detector outputs as a pixel labeling problem, modeled as a 2D Markov Random Field over the pixels. The multi-label MRF is solved using graph cuts. Finally, the top layer introduces the weak architectural concepts. These are guidelines that encourage regularities like horizontal or vertical alignments of windows. An important advantage of our guidelines over grammar rules is that the former are directly observable in the images, whereas the latter keep some concepts implicit. Thus, even if the combined application of a number of façade rules may necessarily lead to, say, the vertical alignment of windows across floors, there could be no single rule explicitly prescribing such alignment. An issue with style grammars can therefore be the very indirect coupling

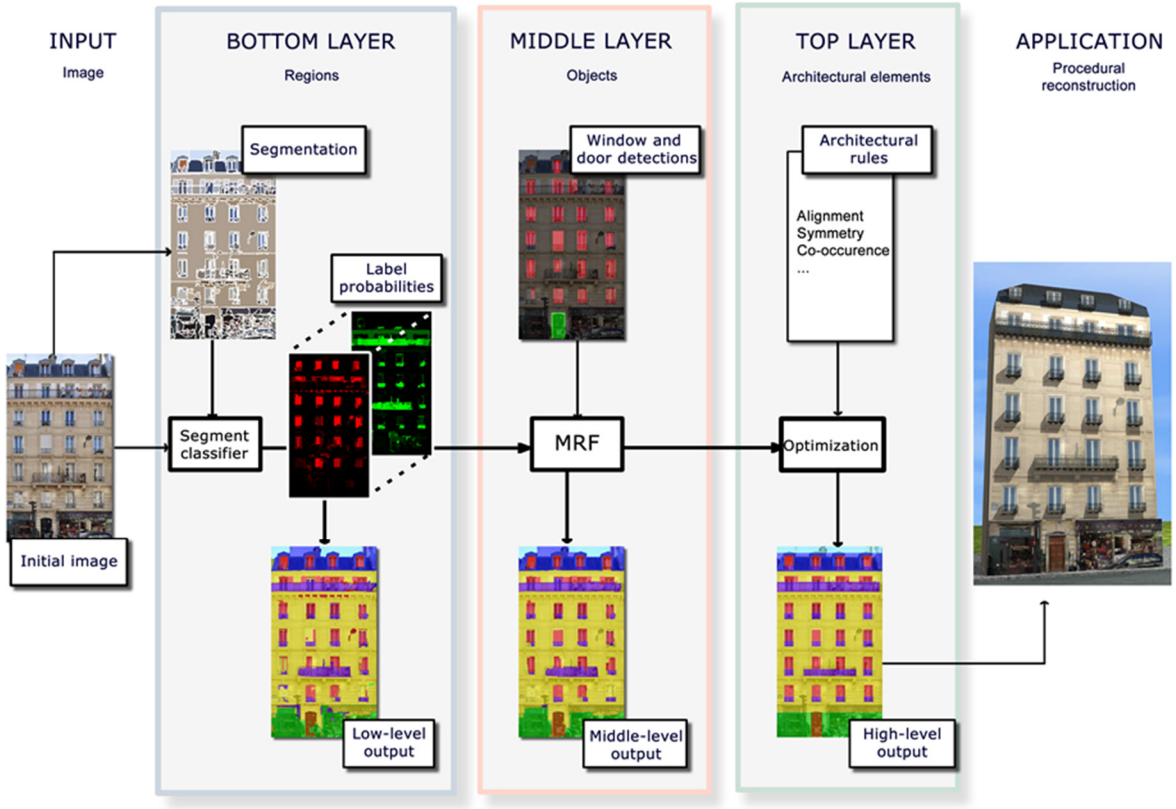

Figure 5: A three-layered approach to facade parsing (Martinovic et al., 2012).

between what they specify and what could easily be verified in the images. Our approach also enables the modeling of irregular facades, as we use the architectural concepts as guidelines, not as hard constraints.

Next, we describe the layers of this process that works without a style-specific grammar, in more detail.

For the first layer, we follow the approach of (Gould et al., 2009a), with some modifications. First, the input image is over-segmented. The segmented regions are created by use of the mean-shift segmentation algorithm of (Comaniciu and Meer, 2002). We prefer a fine-grained segmentation so as not to combine different facade elements into a single region. On average, we obtain 643 regions per image (average image size is 600*400 pixels). Next, appearance (color and texture), geometry, and location features are extracted for each region using the procedure of (Gould et al., 2009a). We use the default parameters from the implementation in the STAIR Vision Library (Gould et al., 2009b), which results in feature vectors of size 225. The trained RNN also builds a parse tree for this image, assigning a score to each segment merger and a multinomial label distribution to each region. We then read out the probabilities in the leaves of the tree and assign to the regions the most likely label. Every region is thus assigned one of the predefined labels, e.g. w*indow, wall, balcony, door, roof, sky, shop.*

At the middle layer, the results of object detectors are introduced. The RNN requires pre-segmented images as input, but the results of the bottom layer are still quite noisy. Object detectors (e.g. of doors, windows, balconies, ….) provide labeling information from a second source, so we can estimate better boundaries for detected elements. We use our own GPU-based implementation of Dollar's Integral Channel Features detector (Dollar et al. 2009, Benenson et al., 2012). This detector provides state-of-the-art results for pedestrian detection and proved to be equally suited for the task of window and door detection (see Figure 6).

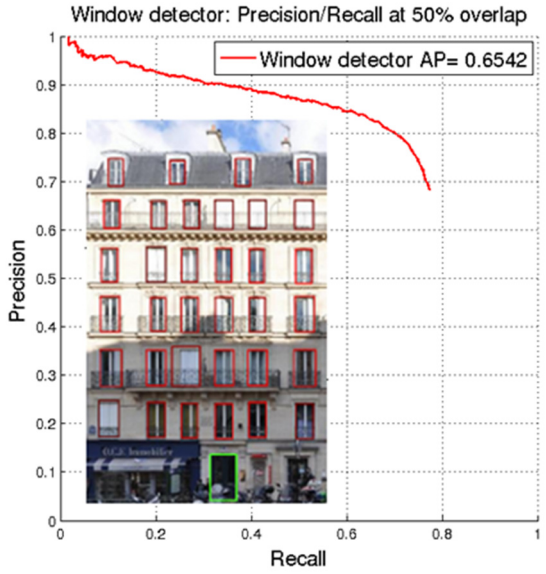

Figure 6: Performance of the window detector.

To merge the information coming from the lower and the middle level of the pipeline (i.e. mean-shift and detector-based segmentations), we formulate a labeling problem by placing a 2D Markov

Random Field over the image pixels. We seek to minimize the total energy, defined as the sum of unary potentials for each node, and the sum of all pairwise potentials between neighboring pixels:

$$E(\psi) = \sum_{x_i} \Phi_s(\psi_i \mid x_i) + \lambda \sum_{x_i} \sum_{x_j \sim x_i} \Phi_p(\psi_i, \psi_j \mid x_i, x_j)$$

where x_i is an image pixel, while the relation $\sim$ represents the 4-pixel neighborhood. Here, λ corresponds to the smoothing parameter, as the pairwise potentials follow the Potts model. The unary potential of a pixel is a weighted sum of the low-level information (RNN labeling) and detector potentials.

Both the output of the initial over-segmentation and the boundaries of the detectors are imprecise. From these two sources alone one cannot expect the MRF to derive a clean semantic segmentation yet. Yet, in these two first layers we have not used any information about the facade structure. The results up to that point may already be convincing quantitatively, but suffer from visually salient errors such as missing or misplaced facade elements. To combat this problem, we exploit *weak architectural principles*, summarized in Table 3.

Principle	Alter	Add	Remove	ECP	eTrims
(Non-)alignment: vertical and horizontal	x	-	-	x	x
Similarity of different *windows* of the same *facade*	-	x	-	x	x
Facade symmetry	-	x	x	x	x
Co-occurence of elements	-	x	x	x	-
Equal width/height in a row or column	x	-	-	x	-
Door hypothesis: first floor, touching ground	x	x	x	x	-
Vertical region order: {*shop**, *facade*$^+$, *roof**, *sky**}	x	-	-	x	-
Running *balcony* in the 2nd and 5th floor	x	x	x	x	-

Table 2: Weak architectural principles used to complement the first 2 layers. An 'x' in the 'alter' column denotes that the principle adjusts element borders. The principle may also remove or add new elements. Last two columns indicate which principles are used for each of the two datasets used in the experiments.

The principles listed above are used to encode high-level architectural knowledge, and they can be directly evaluated in facade images. Most of them can be applied on the majority of facades

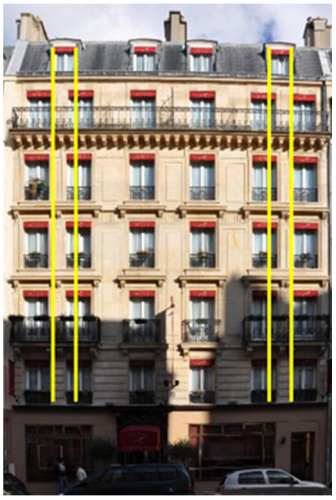

Figure 7: The (non-)alignment principle states that facade elements should be either aligned or clearly off-center.

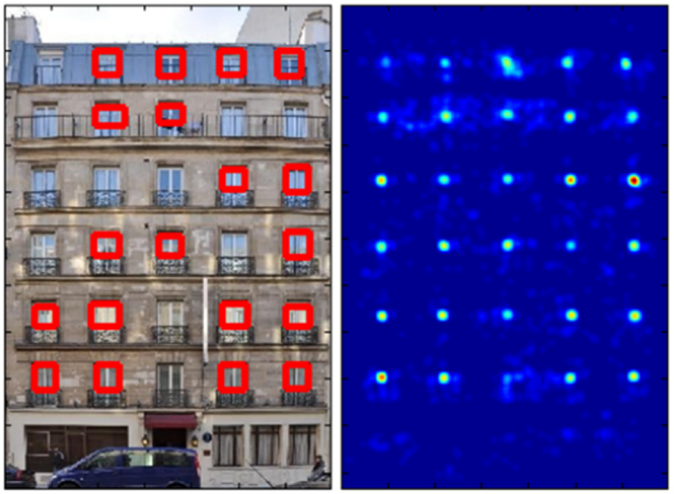

Figure 8: Similarity principle: Left: windows marked with red rectangles are the initially discovered windows. Right: the similarity voting space contains strong peaks on previously undetected windows.

irrespective of their styles, while others are less generally applicable. In any case, the weak architectural principles are weak enough to cover several styles instead of being style-specific. Furthermore, we used the ground-truth labeling of the validation sets for the benchmarks on which we tested to automatically deduce which principles should hold.

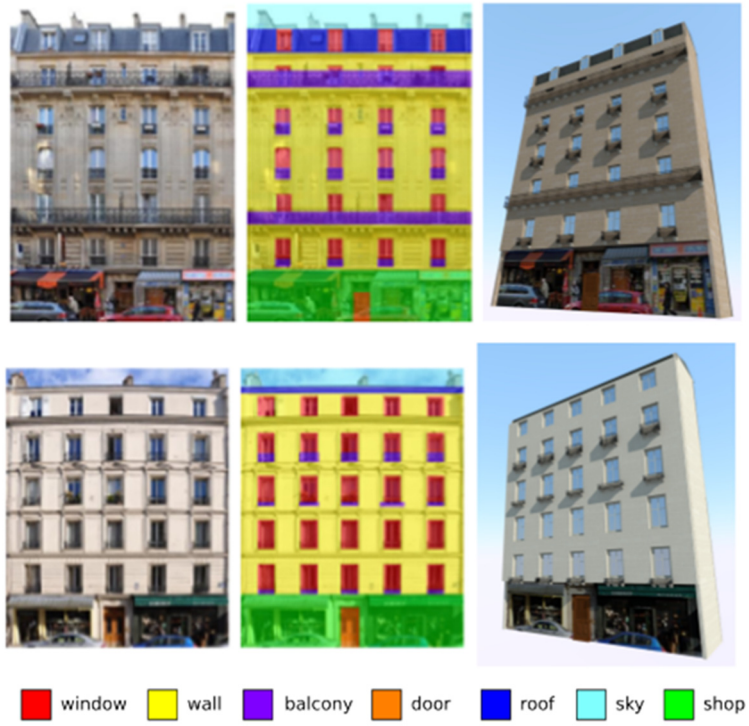

Figure 9: Some modeling results on the ECP dataset. Left: original image. Middle: semantically segmented facade. Right: procedural building model.

Figure 10: Results on the eTrims dataset. (a) Original image. (b-d) Outputs from bottom, middle and top layer. (e) Ground truth.

Figures 7 and 8 illustrate two of the weak architectural guidelines. Figure 7 shows how the system prefers aligned or sufficiently un-aligned configurations (in this case vertical alignment of windows across floors). Figure 8 shows how the system looks for the similarity among detected elements (here windows). Similarity detection helps to suggest the presence of other elements, which went unnoticed by the detector. The system thus infers the presence of additional elements.

Figure 9 shows results for Hausmannian style buildings. The input images are shown on the left. The middle column shows the façade parsing results for the 3-layer system. The right column shows reconstructions. These are cleaner than what pure 3D points clouds would support. Although the rendering was not done very carefully (thereby staying below photo-realism quite a bit still), there are 3D details like deeper lying windows that actually come from the semantics and not from any 3D capturing process. As a matter of fact, in this case the input only consisted of the single images on the left.

Figure 10 shows results for the eTrims dataset. This dataset contains mixed styles and quite diverse sizes of buildings (in contrast to the Hausmann dataset). The system was tested on this benchmark with the same parameters as used for the Hausmann benchmark. In this benchmark, additional object types appear (cars, vegetation, etc.). As the results show, the 3-layer system again produces reasonable results. In this system, all 3 layers perform an important role and yield improvements beyond what lower layers can achieve.

Ongoing work replaces the rather intricate RNN component by a simpler and faster alternative. This and other modifications have already produces a similar but superior system than the one described here.

6. CONCLUSIONS

Semantic 3D modeling uses knowledge about the types of objects to be handled. In the case of buildings, inverse procedural modeling has already made convincing inroads. Yet, in order to apply such schemes at a large scale, it is important that architectural styles are automatically recognized, as to let systems autonomously pick the most appropriate grammar. We have presented some early results to achieve this. That work shows that it is possible to identify styles without first having to fully parse facades, which is a step where one would hope to make use of a style-specific grammar already. Style recognition therefore can take the role of a preprocessing step.

Style-specific grammars are not easy to come by. One needs to bring together experts in the corresponding architectural styles and people versed in writing the grammar rules. This is not obvious. Even if the style-specific grammars have been created, some grammars complicate the inverse implementation because features visible in images and information made explicit in the grammars do not always coincide. For instance, the vertical alignment between windows across floors may only follow indirectly by stating that similar floors need to be stacked upon each other. This implies vertical alignments but never explicitly states their existence. This is a simple example still, but some characteristics that are visually salient could be buried deeper in the rules. These grammars have originally been developed with the purpose of graphics applications, i.e. the creation of virtual models. This is an important cause of such disconnection. Therefore, we have proposed to use weaker architectural guidelines, with a broad applicability. As our façade parsing results have shown, one can go a long way with these for most regular buildings. One can then still consider the use of a style-specific grammar as a final refinement step in the inverse procedural modeling (e.g. as a 4-layer system).

This said, it is probable that the inverse 3D modeling of complicated buildings – e.g. gothic cathedrals – will need a style-specific grammar from the start. The image material will not generate convincing 3D models. Even if it has been claimed that images mined from the Internet allow for such modeling, the vast majority of landmarks can only be reconstructed incompletely or not at all. Tourists tend to focus on the same parts and the same vantage points. Therefore, for such cases the combination of good detectors and strong priors will be required. We therefore expect that there still is a need for the development of new style grammars, but preferentially geared towards the combination with *inverse* procedural models, i.e. taking account of what is visible in images.

Ideally, such vision-oriented grammars could be created automatically, by just presenting the system with example images or models for the relevant style. If only computers generate and use the models, there is no longer a need for generating grammars that are easy to read by humans. We are in the process of generating such grammars automatically. A future pipeline could then look as follows. First, images of a style are collected and the style classification system is extended to cover it. This system is then used to automatically look for more examples, which can be pruned by a human. Then a grammar is built for the larger set of examples. That grammar is used for inverse procedural modeling. This would allow the IPM system to work with style-specific grammars from the start.

Acknowledgements The authors gratefully acknowledge support from the ERC Advanced Grant VarCity (Variation & the City). The object detectors were developed with support of the EC Strep project `Europa' and the inverse procedural modeling pipelines for the classical temples was funded by Integrated Project `3D-Coform'.

7. REFERENCES

Alegre O, Dellaert F (2004): A probabilistic approach to the semantic interpretation of building facades. In: Workshop on Vision Techniques Applied to the Rehabilitation of City Centres, pp. 1-12.

Benenson R, Mathias M, Timofte R, Van Gool L (2012): Pedestrian detection at 100 frames per second. In: IEEE Conference on Computer Vision and Pattern Recognition.

Boiman O, Shechtman E, Irani M (2008): In defense of nearest-neighbor based image classification. In: CVPR.

Bosch A, Zisserman A, Muoz X (2008): Scene classification using a hybrid generative/discriminative approach. Pattern Analysis and Machine Intelligence, IEEE Transactions on 30(4): pp. 712-727.

Comaniciu D, Meer P (2002): Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5): pp. 603-619.

Dai D, Prasad M, Schmitt G, Gool LV (2012): Learning domain knowledge for facade labeling. In: European Conference on Computer Vision.

Dick AR, Torr PHS, Cipolla R (2004): Modelling and interpretation of architecture from several images. International Journal of Computer Vision 60: pp. 111-134.

Dollar P, Belongie S, Perona P (2010): The fastest pedestrian detector in the west. In: Proceedings of the British Machine Vision Conference, BMVA Press, pp. 68.1-68.11.

Van Gool L, Zeng G, Van den Borre F, Müller P (2007): Towards mass-produced building models. In: Photogrammetric Image Analysis, pp. 209-220.

Gould S, Fulton R, Koller D (2009a): Decomposing a scene into geometric and semantically consistent regions. In: International Conference on Computer Vision, pp. 1-8.

Gould S, Russakovsky O, Goodfellow I, Baumstarck P, Ng AY, Koller D (2009b): The stair vision library (v2.2). http://ai.stanford.edu/~sgould/svl.

Han F, Zhu SC (2009): Bottom-up/top-down image parsing with attribute grammar. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(1).

Han T, Liu C, Tai CL, Quan L (2012): Quasi-regular facade structure extraction. In: Asian Conference on Computer Vision, pp. 552-564.

Korah T, Rasmussen C (2008): Analysis of building textures for reconstructing partially occluded facades. In: European Conference on Computer Vision, pp. 359-372.

Lowe DG (2004): Distinctive image features from scale-invariant keypoints. IJCV 60(2):91.

Martinović A, Mathias M, Weissenberg J, Van Gool L(2012): A three-layered approach to facade parsing. In: European Conference on Computer Vision, pp. 416-429.

Mathias M, Martinović A, Weissenberg J, Gool LV (2011a): Procedural 3d building reconstruction using shape grammars and detectors. In: International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, pp. 304-311.

Mathias M, Martinović A, Weissenberg J, Haegler S, Gool LV (2011b): Automatic architectural style recognition. In: ISPRS international workshop 3D-ARCH.

Muller P, Zeng G, Wonka P, Van Gool L (2007): Image-based procedural modeling of facades. SIGGRAPH 26(3):85.

Musialski P, Wonka P, Aliaga DG, Wimmer M, Van Gool L, Purgathofer W (2012): A survey of urban reconstruction. In: EUROGRAPHICS 2012 State of the Art Reports, Eurographics Association, EG STARs, pp. 1-28.

Oliva A, Torralba A (2001): Modeling the shape of the scene: A holistic representation of the spatial envelope. Int J Comput Vision 42: pp. 145-175.

Payne A, Singh S (2005): Indoor vs. outdoor scene classification in digital photographs. Pattern Recognition 38(10): pp. 1533-1545.

Recky M, Wendel A, Leberl F (2011): Façade segmentation in a multi-view scenario. In: International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, pp. 358-365.

Riemenschneider H, Krispel U, Thaller W, Donoser M, Havemann S, Fellner DW, Bischof H (2012): Irregular lattices for complex shape grammar facade parsing. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1640-1647.

Ripperda N, Brenner C (2006): Reconstruction of façade structures using a formal grammar and rjmcmc. In: Pattern Recognition – DAGM Symposium, pp. 750-759.

Romer C, Plumer L (2010): Identifying architectural style in 3d city models with support vector machines. Photogrammetrie - Fernerkundung - Geoinformation 2010:371-384(14).

Shechtman E, Irani M (2007a): Matching local self-similarities across images and videos. In: IEEE Conference on Computer Vision and Pattern Recognition.

Shen CH, Huang SS, Fu H, Hu SM (2011): Adaptive partitioning of urban facades. ACM Transactions on Graphics 30(6):184.

Socher R, Lin CC, Ng AY, Manning CD (2011): Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In: International Conference on Machine Learning.

Summerson J (1996): The Classical Language of Architecture. MIT Press, 1st edition.

Szummer M, Picard R (2002): Indoor-outdoor image classification. In: Content-Based Access of Image and Video Database, 1998. Proceedings, 1998 IEEE International Workshop on, IEEE, pp. 42-51.

Teboul O (2010): Ecole centrale paris facades database. http://www.mas.ecp.fr/vision/Personnel/teboul/data.php.

Teboul O, Kokkinos I, Simon L, Koutsourakis P, Paragios N (2011): Shape grammar parsing via reinforcement learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2273-2280.

Torralba A, Murphy KP, Freeman WT, Rubin MA (2003): Context-based vision system for place and object recognition. In: Proceedings of the Ninth IEEE International Conference on Computer Vision – Volume 2, IEEE Computer Society, Washington, DC, USA.

Toshev A, Mordohai P, Taskar B (2010): Detecting and parsing architecture at city scale from range data. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 398-405.

Vanegas C, Aliaga D, Benes B (2010a): Building reconstruction using manhattan-world grammars. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 358-365.

M. Vergauwen and L. V. Gool (2006): Web-based 3d reconstruction service. Machine Vision and Applications, 17(6): pp. 411-426.

Wendel A, Donoser M, Bischof H (2010): Unsupervised facade segmentation using repetitive patterns. In: Pattern Recognition – DAGM Symposium, pp. 51-60.

Xiao J, Fang T, Tan P, Zhao P, Ofek E, Quan L (2008): Image-based façade modeling. In: SIGGRAPH Asia.

Xiao J, Fang T, Zhao P, Lhuillier M, Quan L (2009a): Image-based street-side city modeling. SIGGRAPH 28(5).

Yang MY, Förstner W (2011): Regionwise classification of building facade images. In: Photogrammetric Image Analysis, Springer, LNCS 6952, pp. 209-220.

Zhao P, Fang T, Xiao J, Zhang H, Zhao Q, Quan L (2010): Rectilinear parsing of architecture in urban environment. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 342-349.