

Modeling Symmetries for Stochastic Structural Recognition

Radim Tyleček and Radim Šára

Center for Machine Perception, Czech Technical University in Prague

tylecrl@cmp.felk.cvut.cz

Abstract

We propose a method for stochastic parsing of images with regular structures exhibiting symmetries, such as facades of buildings. The translational symmetry is represented by an array of elements (windows) that is generated with a stochastic grammar which allows structural exceptions and spatial deviations for individual elements. The reflection symmetry of the elements is automatically inferred as a part of the learning process, where a set of random weak features is boosted into a final mixture.

A hierarchical probability model is built for the attributed 'words' generated by the proposed grammar. The image parsing result is then found as the most probable interpretation visited with MCMC sampler which is designed to efficiently explore the space of possible configurations.

1. Introduction

Stochastic grammars are becoming increasingly popular as a way of representing the prior knowledge about structure and relationship of objects in the images. They were found suitable for both generation and recognition of structured scenes in 2D and 3D, and as such have applications in both computer graphics and vision, i.e. in reconstruction of urban scenes [8]. In this paper we will be interested in their use for recognition of symmetric structures in images.

Our goal is to model regular structures rather than general relations between parts and objects in part-based models [4, 3], but we establish a similar hierarchy of layers which is in our case given explicitly by different nested regions - arrays, elements and features. We build on a general assumption that the embedding of constraints such as symmetry, similarity and regularity in the model allows to achieve better performance than the general approach.

Shape grammars were introduced in [5] and later picked up by [21] as the basic essence for a number of recent parsing methods to overcome the limitations of traditional segmentation techniques. The idea of shape grammars is that image can be constructed as a word in a specific language by applying rules on symbols.

In facade modeling, a wide-spread class of shape grammars are split-based grammars that originate in procedural modeling [13, 16, 1]. They hierarchically split the entire image into orthogonal regions of different classes. A recent representative [19] combines trained randomized forest classifiers with shape grammar to segment Haussmannian facades into eight classes. Their model assumes windows form a grid while allowing different intervals. In the second step, positions of rows and columns are stochastically estimated by a specific random walk algorithm. Subsequently they have proposed a new parser based on reinforcement learning to speed up the process in [18]. Their results were quantitatively evaluated on a dataset of Haussmannian facades in Paris which is available on-line. An approach closer to our was chosen in [20], where the structure is modeled by weak pairwise priors.

The recognition of symmetries and their application for vision problems has also received more interest in the last years. The case of translational symmetry was treated in [7, 14], where lattices are detected in areas seen as near-regular textures. A similar approach applied to rotational and reflective symmetries was implemented in [12]. A large set of symmetries, including helical, is discovered in 3D data in [15], where the usability of symmetries is demonstrated on the task of surface completion. While the mentioned methods detect arbitrary repetitive patterns, our interest is in objects of specific appearance.

The goal of this paper is to show a flexible structural model suitable for parsing of raw images with possibly multiple instances of symmetric structures of image elements, such as facades with windows.

We follow an approach that stands on the concept of *scopes* [17], where a configuration of objects is placed in a 'container', which is an image interval (in the simple case a rectangle). It allows us to directly work with nesting relations, represent structures that overlay each other and frees us from the need to explain the entire image as opposed to the split-based grammars. Also we do not explicitly assume we are given a structured (facade) image on the input, and can answer 'there is probably no facade', when the model prefers to model entire image as the background.

For a higher level of flexibility, we introduce spatial and structural exceptions into otherwise regular arrays of elements. They bring the benefits of the possibility to represent a wider range of structures and it has also showed that the proposed stochastic recognition can faster traverse the space of possible configurations.

2. Grammar

Our goal is to describe a semi-regular array of elements as in Fig. 2. The spatial layout of the terminal elements is a lattice with possible individual deviations of location (*spatial deviations*). We allow empty positions in the array (holes), i.e. an item of the array is left out (*structural exception*) and introduce additional attributes for representation of these exceptions.

We will represent this with attributed context-sensitive stochastic grammar, with contextual sensitivity through attribute constraints. The language of our grammar describes how the image \mathcal{I} contains sub-regions we will call scopes Σ , each scope S defines an array A that splits into rows R or columns C containing terminal elements w , which can be formally written as in Fig. 1. We will use the convention that bold face letters denote vectors or matrices and subscripts their variable dimension throughout this paper.

2.1. Attributes

Scope set Σ_l and scope S . Each of l scopes in the set has defined scope rectangle $\mathbf{x}_i \in (0, 1)^4$ and array dimensions $m_i, n_i \in \mathbb{N}$. The values of all spatial attributes are relative to the parent, i.e. for scopes it is the image rectangle. In this paper, we will proceed only with a single scope in the image, its actual implementation is subject of our current work, thus we fix $l = 1$.

Array $A_{m \times n}$, rows R_n and column C_m . All terminal elements in the array have common size $\mathbf{d} = \{h, w\} \in (0, 1)^2$ and define row and column positions $r_i, c_i \in (0, 1)$. An array can be expanded either by removing the last row or column.

Terminal elements $w(\mathbf{d}, r, c, \Delta, h)$. Each element has independently defined position deviation $\Delta = \{\delta_r, \delta_c\} \in (0, 1)^2$. Its actual rectangle of size \mathbf{d} is then centered on the position given by $(r + \delta_r, c + \delta_c)$. The structural exceptions are represented with binary hole labels $h \in \{0, 1\}$, where $h = 0$ marks presence of an element and $h = 1$ a structural exception, what results in treating the element area as the background. For simpler view the reader can identify the terminal elements with windows throughout this paper.

3. Probability Model

We build a probability model on the set of attributes θ of the structural model generated by the proposed grammar and the image data I , and express it as the product of likelihood and prior:

$$p(\theta, I) = p(I|\theta)p(\theta). \quad (1)$$

3.1. Structure Prior

The prior for structural attributes is defined as

$$p(\theta) = p(\mathbf{x})p(\mathbf{r}|w, m)p(\mathbf{c}|h, n)p(\Delta|h, w, m, n) \quad (2)$$

$$p(H|m, n)p(m, n|h, w)p(h, w).$$

Scope prior. The scope is expected to fill the image, which is modeled by setting $p(x) = \text{Be}(\|x\|, \beta_x, 1)$, $\beta_x = 10$ for the occupied area, its actual position is uniformly distributed.

Window size prior. It is independently set to $p(h, w) = p(h|w)p(w)$ with Beta distributions for width and aspect that were fitted to the training data.

Array size. Array size prior is $p(m, n|h, w) = p(m|x, w)p(n|x, h)$ with binomial distributions

$$p(m|x, w) = \text{Bi}(m, m_{max}, \alpha_m), \quad (3)$$

where $m_{max} = \lfloor \frac{mw}{|x|} \rfloor$ is the limit given by the terminal element and scope widths, and $\alpha_m = 0.5$ models the expected free space between the terminal elements (analogically for columns in $p(n|x, h)$).

Row and column positions. The array rows are expected to be equispaced, this is modeled with a homogeneous Dirichlet distribution on inter-row distances:

$$p(\mathbf{r}|w, m) = \frac{1}{B(\alpha)} \prod_{i=0}^m (r_{i+1} - r_i - w)^{\alpha_i - 1}, \quad (4)$$

where we additionally define $r_0 = 0$ and $r_{m+1} = 1$ to include the border intervals. The parameter $\alpha_i = 3$ controls the width of modes of marginal Beta distributions of row positions and where the non-overlapping constraint is implicitly enforced. Column position prior $p(\mathbf{c}|h, n)$ is defined analogically.

Spatial deviations. The local deviations from the grid defined by \mathbf{r} and \mathbf{c} are normally distributed with the scale of the terminal element's size:

$$p(\Delta|h, w) = \prod_{i=1}^m \prod_{j=1}^n p(\delta_{ij}^c | w) p(\delta_{ij}^r | h), \quad (5)$$

$$\begin{aligned}
\mathcal{I} &\longrightarrow \Sigma_l(\mathbf{x}_l, \mathbf{m}_l, \mathbf{n}_l) \\
\Sigma_{l+1}(\mathbf{x}_{l+1}, \mathbf{m}_{l+1}, \mathbf{n}_{l+1}) &\longrightarrow \Sigma_l(\mathbf{x}_l, \mathbf{m}_l, \mathbf{n}_l) \circ S(x_{l+1}, m_{l+1}, n_{l+1}) \\
S(x, m, n) &\longrightarrow A_{m \times n}(\mathbf{d}, \mathbf{r}_m, \mathbf{c}_n) \\
A_{m+1 \times n}(\mathbf{d}, \mathbf{r}, \mathbf{c}_{n+1}) &\longrightarrow A_{m \times n}(\mathbf{d}, \mathbf{r}_m, \mathbf{c}_n) \circ R_n(\mathbf{d}, r_{m+1}, \mathbf{c}_n) \\
A_{m \times n+1}(\mathbf{d}, \mathbf{r}, \mathbf{c}_{n+1}) &\longrightarrow A_{m \times n}(\mathbf{d}, \mathbf{r}_m, \mathbf{c}_n) \circ C_m(\mathbf{d}, r_m, c_{n+1}) \\
R_{n+1}(\mathbf{d}, r, \mathbf{c}_{n+1}) &\longrightarrow R_n(\mathbf{d}, r, \mathbf{c}_n) \circ w(\mathbf{d}, r, c_{n+1}, \Delta, h) \\
C_{m+1}(\mathbf{d}, r_{m+1}, c) &\longrightarrow C_m(\mathbf{d}, r_m, c) \circ w(\mathbf{d}, r_{m+1}, c, \Delta, h) \\
A_{0 \times 0} &\longrightarrow \varepsilon \\
R_0 &\longrightarrow \varepsilon \\
C_0 &\longrightarrow \varepsilon
\end{aligned}$$

Figure 1. Proposed grammar in Chomsky Normal Form [9] with start symbol \mathcal{I} , non-terminal symbols in capitals and terminal symbol and attributes in lower-case.

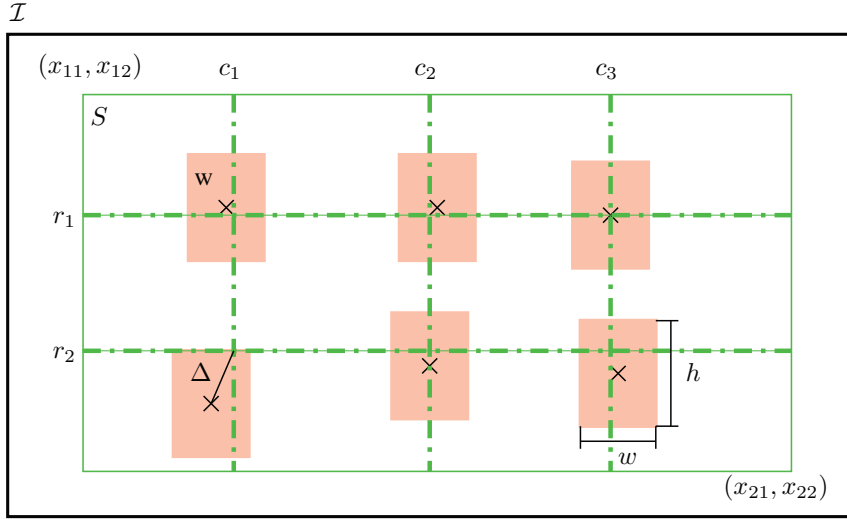


Figure 2. Scope and array parametrization (green), terminal elements (red) with centers.

where $p(\delta_{ij}^c | w) = \mathcal{N}\left(\frac{\delta_{ij}^c}{w}, 0, \sigma_\delta\right)$ scales horizontal deviation with width, analogically for vertical and height.

Structural exceptions. The presence of holes is modeled by

$$p(H) = \prod_{i=1}^m \prod_{j=1}^n p(h_{ij}), \quad (6)$$

where $p(h_{ij} = 1) = p_h = 0.1$ and $p(h_{ij} = 0) = 1 - p_h = 0.9$.

3.2. Image Likelihood

The likelihood of the given input image I is defined for a combination of foreground (interiors of terminal elements)

and background areas I_f and I_b

$$p(I|\theta) = \prod_{i \in I} p(I_i|\theta) = \prod_{i \in I_b} p_b(I_i|\theta) \prod_{j \in I_f} p_f(I_j|\theta). \quad (7)$$

The background pixels I_b are evaluated individually and uniformly, foreground pixels are grouped by elements in

$$p(I|\theta) = p_b^{|I|} \prod_{i=1}^m \prod_{j=1}^n \left(\frac{p_f(\mathbf{I}_{ij})}{p_b(\mathbf{I}_{ij})} \right)^{|\mathbf{d}|}, \quad (8)$$

where \mathbf{I}_{ij} denotes all pixels in the element rectangle at row i and column j , and $|I|, |\mathbf{d}|$ are dimensions of the image and element in pixels. The foreground probability is given by a mixture of weak features Φ

$$p_f(\mathbf{I}_{ij}) = \sum_{\varphi \in \Phi} p(\varphi) p_f(\mathbf{I}_{ij}|\varphi). \quad (9)$$

If the terminal element is excluded from the array (hidden, $h_{ij} = 1$), then its area is treated as a background, it is not evaluated and we set $p_f(\mathbf{I}_{ij}) = p_b$. In the standard case ($h_{ij} = 0$) the probability pdf is given by a specific feature. We designed three families of features to discriminate foreground from the background, because they are computed frequently during sampling, they must be simple enough to be fast. To cope with this limitation, we choose features calculated from integral images. They are described in the next sections as well as the actual selection of the feature set.

3.2.1 Differential features

The first family is based on randomly generated basic set of Haar-like features [11], which are expected to work well in the largely orthogonal world of arrays. A feature is specified by its rectangle within terminal element scope, type (single split, double split, central), orientation (horizontal/vertical), split position within the rectangle and the number of image channel on which a feature is evaluated. We work with six channels generated from the input RGB image: hue, saturation, intensity, edges (detected by standard Canny) and horizontal and vertical gradients of the intensity image. Since the surrounding area of the terminal element of interest plays an important role for its localization, we extend the actually evaluated area for input to the classifier by 20% in all directions.

The Haar-like feature responses (differences between the split parts) are then fitted to the range of $(-1, 1)$ and the classification threshold is shifted by t_0 . The transformation is based on the statistics of positive and negative samples in the training set, which should ideally have clearly separated modes. The classification threshold $t_0 \in (-1, 1)$ is computed in Bayesian way from Beta distributions fitted to the training samples as in Fig. 3.

Instead of taking directly the likelihood ratio, we can now define sigmoidal pdf (Fig. 4) on this response $t \in (-1, 1)$

$$\text{Sig}(t|t_0, \sigma) = \frac{1}{Z(\sigma)} \frac{1}{1 + e^{-\sigma(t-t_0)}},$$

where $Z(\sigma) = \frac{1}{\sigma}(\log(1 + e^\sigma) - \log(1 + e^{-\sigma}))$ is a partition function and σ controls the cutoff slope, which is set as the mean of the standard deviations of the fitted Beta distributions. This pdf can be now simply inserted into the set of features (9) as

$$p_f(\mathbf{I}_{ij}|\varphi) = \text{Sig}(\mathbf{H}_\varphi \cdot \mathbf{I}_{ij}|t_\varphi, \sigma_\varphi) \quad (10)$$

where \mathbf{H}_φ is the Haar-like feature mask.

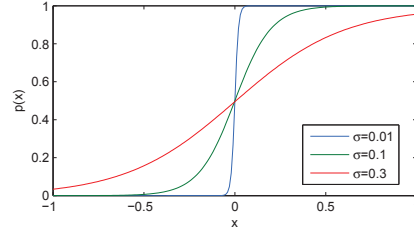


Figure 4. Sigmoidal pdf for different values of σ .

3.2.2 Symmetric features

While the features described in the previous section are based on differences of feature's rectangles, we can also in the same fashion evaluate their similarity, which corresponds to the reflection symmetry. The only change is that we expect the means of values in both Haar-like rectangles to be equal, and model it with a zero-mean normal distribution on values of the differences of responses t :

$$p(t|\sigma) = \mathcal{N}(t, 0, \sigma),$$

which is fit to the positive samples and inserted into set as in the previous section.

This gives us the potential to learn any reflection symmetry with horizontal or vertical axis that is present within the element area (including the symmetry of entire element area) by selection of discriminative features described later in Sec. 3.3. Depending on the training data, the reflection symmetries can be included in the final model or not, unlike translational symmetry of arrays which is enforced explicitly by grammar (though allows exceptions).

3.2.3 Similarity features

The appearance of terminal elements in the model should be most probably equal. We model this by introducing a hidden variable \bar{I}_f that represents the appearance, and then evaluate

$$p_{ch}(I_f(r_m, c_n)|\bar{I}_f) = \mathcal{N}\left(\|I_{f,ch}^{m,n} - \bar{I}_{ch}\|, 0, \sigma_{ch}\right).$$

where $\|\cdot\|$ is a measure of similarity, such as SSD between terminal element area and its appearance template. However, this computation is expensive and grows with image resolution. The actual measure we used is difference of mean values of element and template, which can be computed easily with integral images. We deterministically set the appearance to be the mean over all terminal elements $\bar{I}_{ch} = \frac{1}{MN} \sum_{mn} I_{ch}^{m,n}$ in the image, and then insert these features into the generated set for all channels. It acts in the model as an adaptive component that can increase the ability of generalization in the proposed method.

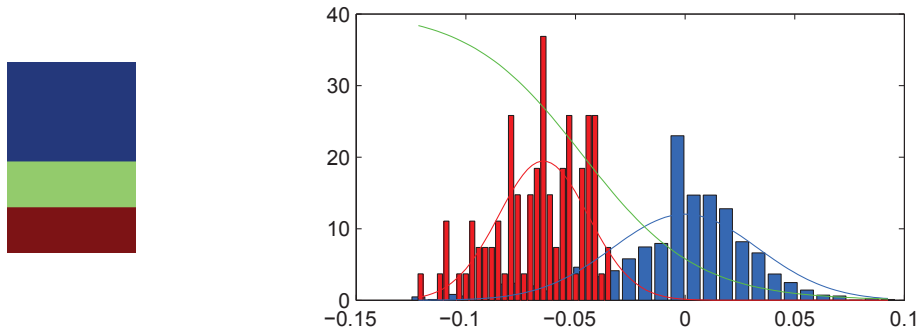


Figure 3. Left: Haar-like feature for vertical gradient channel. Right: histogram of its responses for positive class (red) and negative class (blue), fitted sigmoidal pdf in green (scaled).

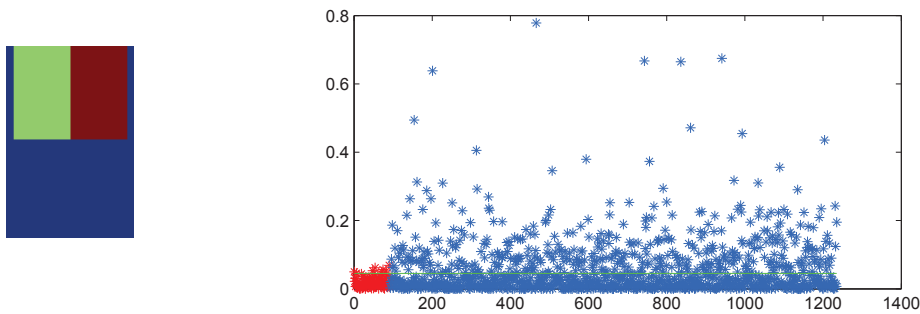


Figure 5. Left: Symmetric feature for intensity channel evaluates similarity of the two rectangles. Right: Responses for positive class are small (red) and higher for negative class (blue).

3.3. Learning

Most parameters of the structural model were inferred by fitting respective distributions (as given in the probability model) to the training samples. In certain cases (when a specific value is given in the text) we assigned them manually because it was not obvious how to extract them or there were not enough samples to build a statistics. However, our intention is to train all parameters automatically in the future.

To learn the the feature mixture for image likelihood in (9), we employ the well known concept of boosting to combine a set of weak classifiers into a single, powerful one. The output feature set F and the weights $p(f)$ of the individual features in (9) are selected with FloatBoost [10] from the set of 100 generated and filtered weak features (unpromising ones are directly discarded). Moreover the least weighted features that fail to improve results on the validation set are removed in order to reduce overfitting and computation cost. The whole selection process is repeated $100\times$ and ultimately all selected features are combined into one set and boosted once more. The resulting set typically contains ~ 30 features.

4. Recognition

The solution θ^* for a given input image is found as the most probable sample

$$\theta^*(I) = \arg \max_{\theta} p(\theta, I)$$

visited in the run of Markov Chain Monte Carlo. It allows us to efficiently explore the space of all possible attribute configurations by means of a random walk and to traverse different dimensions at the same time.

4.1. Independent sampling

For initialization of the chain we can sample $\theta_0 \sim p(\theta)$ directly from the prior distributions, first scope x and terminal element size h, w , then array dimensions m, n and subsequently all other attributes.

4.2. Conditional sampling

The conditional sampler $\theta' \sim q(\theta'|\theta, I)$ is a mixture of individual samplers such that each modifies a subset of parameters θ based on a specific proposal distribution $q_s(\theta'|\theta, I)$. The main sampler chooses from $q(s|\theta)$ which of the individual samplers s will be used to propose the next

move. We will now propose the set of samplers that will explore the space of parameters θ . Their design must fulfill Markov Chain properties of detailed balance and reversibility of all moves, i.e. given a move there must always exist a reverse move s' , and their probability ratio must be reflected in the acceptance of Metropolis-Hastings (MH) algorithm:

$$A = \min \left\{ 1, \frac{p(\theta', I)}{p(\theta, I)} \cdot \frac{q_{s'}(\theta|\theta', I)}{q_s(\theta'|\theta, I)} \cdot \frac{q(s'|\theta')}{q(s|\theta)} \right\}. \quad (11)$$

Change element size h, w

Adjust the terminal element dimensions by adding a random vector either to height $h' = h + dh$, $dh \sim \mathcal{N}(0, \sigma_h)$ or similarly to width w .

Move scope x . Adjust the position of the scope rectangle by adding a random vector $x' = x + dx$, $dx \sim \mathcal{N}(0, \sigma_x)$.

Resample row and column positions r, c .

Uniformly draw a row (or column) i . The distribution for its new position within the corresponding interval is $\text{Be}(\alpha_\tau, \beta_\tau)$ with mode centered in a random portion $\tau \in \{\frac{1}{2}, \frac{1}{3}, \frac{2}{3}\}$

$$r'_i \sim q(r_i|r_{i-1}, r_{i+1}). \quad (12)$$

For the first and last position, the support distance is given by positions of the nearest two rows/columns $r'_1 \sim q(r_1|r_2, r_3)$.

Resample positions deviations δ_{ij} .

Uniformly choose a terminal element at position ij , then resample its deviation from the prior $\delta'_{ij} \sim p(\delta_{ij}|w, h)$.

Change structural labels h_{ij} .

Uniformly choose a terminal element at position ij , then invert its hole label $h'_{ij} = 1 - h_{ij}$.

Add and remove rows or columns m, n .

This proposal changes the dimension of the parameter vector θ and such operation requires additional factors in (11), as described in [6].

Add (birth). It is possible to insert a new row in between of two present rows, before the first or after the last. In the last two cases the scope is extended to fit a new row if necessary. The new position is sampled as in (12). The correcting factor in acceptance ratio for dimension matching is in this case $\frac{1}{p(r_i|R_{-i})}$.

Remove (death). Remove random row or column. The correcting factor in acceptance ratio for dimension matching is inverse of Add, thus $p(r_i|R_{-i})$.

Resample positions deviations δ .

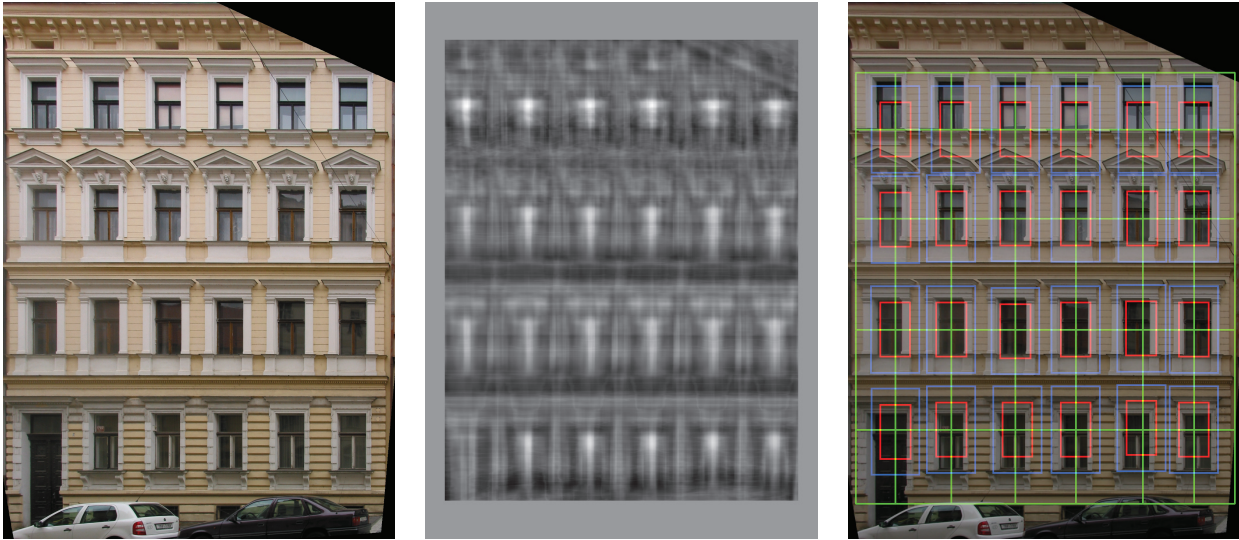
Uniformly choose a terminal element at position ij , then resample its deviation from the prior $\delta'_{ij} \sim p(\delta_{ij}|w, h)$.

5. Experiments

We have performed experiments on the rectified facades of images to evaluate the performance of the proposed method. We have used two facade image databases [19, 2] and trained both classifier and the structural model to detect arrays of windows. The images in [2] were automatically rectified and both databases were split into three parts for training, validation and testing purposes. Positive samples (windows) were manually filtered to exclude unrepresentative ones (occluded, distorted, etc.). Negative samples were taken randomly around the positive ones such that the pixel in the center does not belong to the positive class. The MCMC was run for a fixed number of 5×10^4 samples, what corresponds roughly to visiting 30% of image pixels and 10 minutes of CPU time. Both sampling and its implementation are subject to further optimization.

The typical results are presented in Fig. 6, where certain amount of vertical ambiguity can be observed in the image likelihood b), which is resolved by the row position prior in the output result c). Note that also doors are included in the array since they have similar appearance and the classifier is not trained to distinguish between the door and window. The black areas in the corners were outside the original (unrectified) images. More results can be seen in Fig. 7, notice how is our method successful in determining the scope position and to handle the structural exceptions (missing window). In some cases here the spatial deviations from the regular positions help to better localize the element.

Results on the Parisian database were compared with split-based methods proposed in [18, 19], however direct comparison is here not possible as they solve multi-class problem, but still it is the only result we can set against. Quantitative results on relevant *window* and *wall* classes are given in Tab. 1 and selected images are displayed in Fig. 8. The difficulty in our case rises from the fact we purposely model the background uniformly, in contrast to [18, 19] where the single *wall* classifier can with ease fit its ochre color over the entire dataset. Though no strong conclusions can be made and without intention to compete, we evaluate the performance of our method as comparable with the previous ones while the model is more biased towards false negatives.



a) input image b) image likelihood c) parsing result
 Figure 6. Results on a typical image, its likelihood b) is computed with the correct window size as seen in c).

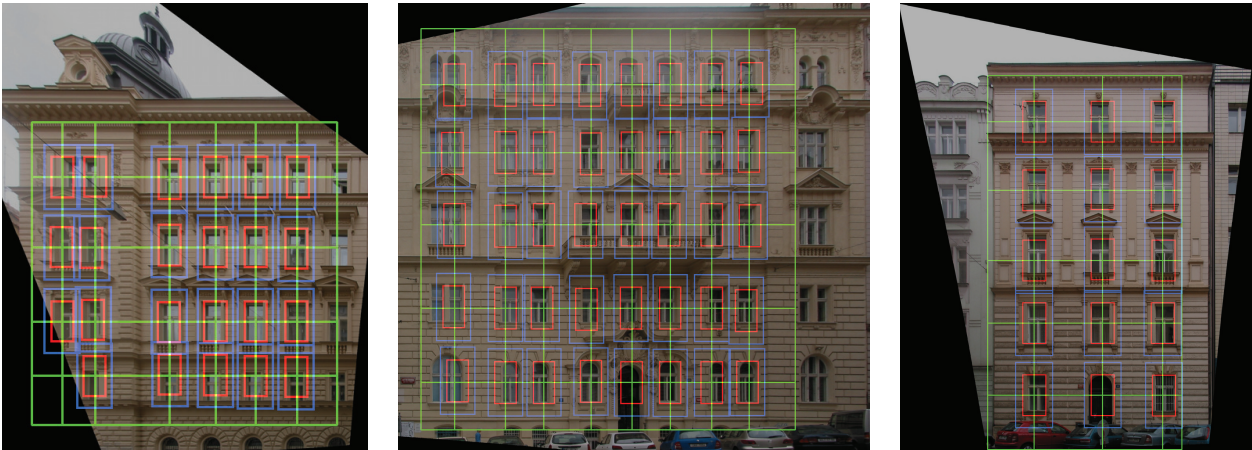


Figure 7. Results on selected images from Prague [2]

Class	PS [19]	RL [18]	WSM [20]	our
window	81	81	83	78
wall	83	84	84	89

Table 1. Quantitative evaluation on Paris dataset [19], recognition rate in percents.

6. Conclusion

The proposed method based on the concept of *scopes* was presented as a competitive alternative to existing mostly split-based methods, while allowing wider range of possible applications thanks to its higher generality. The overall hierarchy of nested scopes ‘image - array scope - terminal element - feature’ comes out as a flexible representation that

is open to future extensions and suitable for intended applications such as structural priors for 3D reconstructions.

Our current interest is to work with multiple array scopes in one image, which will allow us for example to segment streets into individual houses and extension to more object classes. Simultaneously we want to complete the learning of all parameters from the training data and explore unsolved problems related to sampling, such as termination criteria and learning proposal densities.

Acknowledgment

This work was supported in part by the Czech Ministry of Education under Project MSM6840770012, in part by Grant Agency of the CTU Prague under Project SGS10/278/OHK3/3T/13 and in part by Google Research Award.



Figure 8. Results on three selected images from the Parisian dataset [19]

References

- [1] F. Alegre and F. Dellaert. A probabilistic approach to the semantic interpretation of building facades. In *International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres*, 2004. 1
- [2] eTRIMS Image Database (60 annotations). <http://www.ipb.uni-bonn.de/projects/etrimis.db>. 6, 7
- [3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010. 1
- [4] S. Fidler, M. Boben, and A. Leonardis. Similarity-based cross-layered hierarchical representation for object categorization. In *Proc. CVPR*, 2008. 1
- [5] J. Gips. *Shape grammars and their uses*. Birkhäuser, 1975. 1
- [6] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995. 6
- [7] J. Hays, M. Leordeanu, A. Efros, and L. Yanxi. Discovering texture regularity as a higher-order correspondence problem. In *Proc. ECCV*, pages 522–535. Springer, 2006. 1
- [8] B. Hohmann, U. Krispel, S. Havemann, and D. Fellner. CITYFIT: High-quality urban reconstructions by fitting shape grammars to images and derived textured point cloud. In *Proc. of the International Workshop 3D-ARCH*, 2009. 1
- [9] J. E. Hopcroft, R. Motwani, and J. D. Ullman. Introduction to automata theory, languages, and computation, 2nd edition. *SIGACT News*, 32:60–65, March 2001. 3
- [10] S. Li and Z. Zhang. Floatboost learning and statistical face detection. *Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1112–1123, sept. 2004. 5
- [11] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proc. Image Processing*, volume 1, pages I–900 – I–903 vol.1, 2002. 4
- [12] G. Loy and J.-O. Eklundh. Detecting symmetry and symmetric constellations of features. In *Proc. ECCV*, pages 508–521, 2006. 1
- [13] P. Müller, G. Zeng, P. Wonka, and L. Van Gool. Image-based procedural modeling of facades. *Transactions on Graphics*, 26(3):85, 2007. 1
- [14] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu. Translation-symmetry-based perceptual grouping with applications to urban scenes. In *Proc. ACCV*, volume 6492 of *LNCS*, pages 329–342, 2011. 1
- [15] M. Pauly, N. Mitra, J. Wallner, H. Pottmann, and L. Guibas. Discovering structural regularity in 3D geometry. *Transactions on Graphics*, 27(3):43–43, 2008. 1
- [16] N. Ripperda and C. Brenner. Data driven rule proposal for grammar based facade reconstruction. *Photogrammetric Image Analysis*, 36(3/W49A):1–6, 2007. 1
- [17] G. Stiny. Introduction to shape and shape grammars. *Environment and Planning B*, 3(8):241–274, 1977. 1
- [18] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios. Shape grammar parsing via reinforcement learning. In *Proc. CVPR*, 2011. 1, 6, 7
- [19] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape prior. In *Proc. CVPR*, 2010. 1, 6, 7, 8
- [20] R. Tyleček and R. Šára. A weak structure model for regular pattern recognition applied to facade images. In *Proc. ACCV*, volume 6492 of *LNCS*, pages 450–463, 2010. 1, 7
- [21] S. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):362, 2006. 1