Computer Science and Information Systems 13(1):23–43 DOI: 10.2298/CSIS150222062V

Obtaining Structural Descriptions of Building Façades

Petar Vračar, Igor Kononenko, and Marko Robnik-Šikonja

University of Ljubljana
Faculty of Computer and Information Science
Tržaška 25, 1000 Ljubljana, Slovenia
{petar.vracar,igor.kononenko,marko.robnik}@fri.uni-lj.si

Abstract. We describe a method for learning and recognizing windows as basic structural elements of façades and organizing them into interpretable models of building façades. The method segments an input image into a hierarchical structure of window candidates. The candidates are used to create a likelihood map of window locations that is explained by a structural façade model based on a formal grammar. We use a look-ahead greedy search method in the grammar derivation space to select the (sub)optimal façade model. Empirical evaluation results reveal that, on average, the generated façade model covers 45% of the actual windows present in the input image. On the other hand, 56% of the modeled windows actually cover façade windows present in the input image.

Keywords: façade segmentation, window detection, formal grammar, urban environment, image segmentation.

1. Introduction

In recent years there is a growing interest in the field of modeling, recognition and interpretation of urban environments. The main motivation for this study is to provide a tool which would help to annotate the windows (and later also other parts) on the façade images. Several types of users can benefit from such a tool: architects, analysts of architectural history, a visual navigation system, etc. Currently, for example, an architect has to manually annotate the windows, which is a tiresome and time consuming task for hundreds of façade images for a certain region of a larger city. The idea is to provide a semi-automatic tool which gives the first approximation of window locations, and users would only have to make the necessary corrections in case of a demand for greater accuracy. This would significantly reduce the annotation time required.

In this paper we focus on window recognition as a basic step in the description of façades. The recognition process starts from local structures at the pixel level. Most windows are rectangular in shape. Therefore, the linear edge segments in the façade images are a reasonable starting point for generating hypotheses about the position of windows on the façade. One can also use the fact that buildings are often divided into floors which are further divided into rooms. The room configuration is usually uniform for the whole building and, as a consequence, windows align horizontally (for individual floors) and vertically (for individual rooms on different floors). This is a key property of window candidates. Although they generally vary in shape and size, the windows on the same façade are often similar to each other, which can be used for recognition. In other words,

multiple repetition of a certain part of the façade image increases the probability that this part corresponds to the window. Finally, it is possible to use machine learning methods with an appropriate set of features to develop a window detector based on the learned window appearance. Once the likely location of windows are known, one can generate a structural description of a façade that is interpretable and allows a further high-level processing.

The rest of the paper is structured as follows. In the next section we briefly present the related work. Section 3 describes the segmentation of input images which generates the hierarchical structure of window candidates which is then used in façade modeling via formal grammar rules. Section 4 provides the experimental results which are discussed in Section 5. The paper concludes in Section 6 where we give some ideas for further work.

2. Related work

In the early seventies Stiny and Gips [1] introduced shape grammars as a generative approach to shape design, laying a foundation for three-dimensional architectural grammars. In recent years, research on the use of grammars in architectural design has been intensified. Wonka et al. [2] developed a method for the automatic generation of detailed building models using a split grammar, a formal context-free grammar that uses parametrized shapes as primitive elements rather than letters or symbols. A large number of production rules that consist of geometric split operations is required to hierarchically transform the shapes and derive building models with rich geometric detail for several different architectural styles. During the derivation process, a second kind of grammar, called a control grammar, is used in order to propagate the split grammar attributes. Müller et al. [3] extended split grammars to context-sensitive shape grammars for more complex and consistent procedural modelling of buildings, including their façade elements.

There has been a notable tendency in recent years towards stochastic methods for building modeling, and especially façade modeling. Alegre and Dellaert [4] used stochastic context-free grammars and the Markov Chain Monte Carlo (MCMC) method [5] to obtain a semantic description of building façades by applying hierarchical partitioning based on local appearance characteristics. Mayer and Reznik [6] combined appearance-based, generative modeling employing MCMC, and Implicit Shape Models (ISM) for the 3D interpretation of building façades. Dick et al. [7] proposed a Bayesian model for automatic 3D building reconstruction using evidence from multiple images. Prior information of typical buildings characteristics such as parallelism and orthogonality is integrated in a stochastic process which is sampled using a Reversible Jump MCMC (rjMCMC) sampler [8]. Ripperda and Brenner [9] used rjMCMC to guide the grammar-based derivation of a structural façade description. The proposed method is data driven and uses several measures on terrestrial façade images and laser scans to model jumping distribution. Teboul et al. [10] addressed shape grammar parsing for facade segmentation formulated in the framework of a Hierarchical Markov Decision Process and solved it using Reinforcement Learning. Becker [11] presented a system which attempts to automatically discover a formal grammar for geometric modelling of façades from LiDAR point clouds and image data. Martinovic and Van Gool [12] used the Bayesian Model Merging technique to automatically learn stochastic context-free grammars from a set of labeled building facades.

Their induced grammar can be sampled to create novel instances of the same building style, and also achieves excellent results in the facade parsing task.

There are also stochastic methods for façade segmentation that are not grammar-based. Yang and Förstner [13] formulated façade segmentation as a labeling problem. They used randomized decision forest classifiers to classify façade image regions. Then, a conditional random field is used to enforce spatial consistency between neighboring regions. Martinovic et al. [14] presented a three-layered approach for façade parsing. In the first layer they employ recursive neural networks to obtain a semantic segmentation of a façade, which is then augmented with object detectors in the second layer. A third, architectural layer, is then used to establish the architectural plausibility of the final segmentation. Tyleček and Šára [15] used local low-level data evidence (regularity of spacing, shape similarity, alignment) to establish a weakly regular structure of façade windows that is optimized using the rjMCMC framework.

Approaches besides stochastic models include Müller et al. [16] who introduced mutual information to derive a meaningful hierarchical façade description by detecting repetitions. Lee and Nevatia [17] proposed a profile projection method to frame floors and window columns using alignment of the building windows. They segmented the window candidates by intersecting the vertical and horizontal projection profile histograms and then refined the candidates using local information. Haugeard et al. [18] presented a window extraction method which is also based on a profile projection. They used it to find linear contours in the input image which are organized in the form of a relational graph. Façade windows, represented as subgraphs, are extracted from the graph by means of a kernel similarity function for structured sets of contours. Liu et al. [19] described a method based on the Kronecker product for detection of repeated patterns in façade images. The method is based on a theoretical foundation, but assumes a regularity of structure and cannot model aperiodic façades.

Our approach can be characterized as grammar-based with a look-ahead greedy search strategy in the grammar derivation space. Unlike the traditional approach that uses a context-free grammar to reconstruct the input façade image directly, our basic idea is to create a likelihood map of window locations that is explained by a façade model in the form of a derivation tree according to predetermined grammar rules. The main advantages of our look-ahead greedy search strategy are that it is deterministic, does not require additional tuning or learning steps, it is simple to implement, and it is computationally less demanding than the usually used random sampling approach.

3. Façade segmentation

Façades, like many man-made objects, usually have a highly regular structure. On the other hand, photographs of façades do not preserve parallelism and orthogonality between structural elements due to perspective transformation. In order to simplify further processing we perform two pre-processing steps. First, we rectify the input image i.e. remove the perspective projection from the image by reconstructing the orthogonal view to the plane that models the spatial orientation of the façade surface (for more details about image rectification see [20]). As the second pre-processing step, we blur the rectified image in order to eliminate, or at least reduce, the presence of any façade texture patterns

such as a brick wall. The windows' edges are usually more prominent than those of the façade texture and, as such, will be preserved in the output image.

From now on, we assume rectified input images preserving parallelism and orthogonality of the façade elements' edges. For the purpose of generating hypotheses about the position and size of façade windows, we also assume that the windows are rectangular in shape and have their edges aligned with the image coordinate axes. In this case, we can expect more vertical edges in the image sections corresponding to the horizontal rows of windows in same floor, and fewer vertical edges in the image sections corresponding to the wall between floors. Similarly, more horizontal edges can be expected in image areas that correspond to columns of the façade windows and fewer between them.

3.1. Creating candidates

Lee and Nevatia [17] proposed a method for window detection. They presented the concept of a projection profile that represents the histogram of the edge distribution in the input image. The vertical projection profile is calculated by projecting the vertical edges to the vertical axis. The vertical edges of the aligned windows are projected to the same part of the vertical axis. The accumulation of these projections gives a histogram of the vertical edge distribution. The horizontal projection profile can be calculated analogously, by accumulating the projections of the horizontal edges to the horizontal axis. The window candidates may be segmented by intersecting the strips that correspond to the peaks in the horizontal and vertical projection profile histograms. We have chosen a slightly different approach similar to [18]. In order to use local information, we decided to segment the façade image using a greedy method which recursively divides the façade in the vertical or horizontal direction. A decision on the division direction is made using the location of the global minimum of the local projection profiles. In case the vertical projection profile contains the global minimum, a vertical division will be performed. Otherwise, a horizontal division will be used. In each iteration, the input image is split into three parts: the central part (corresponding to the valley around the global minimum with respect to both projection profiles) and the two marginal parts – one on each side.

We assume that the central part does not contain any façade structural element, and therefore it is excluded from further analysis. Each marginal part is recursively divided by the same procedure. In this way we build a hierarchical structure of the observed façade using local information. Subdivision into smaller parts is stopped when the size of the observed area becomes smaller than a predetermined minimum, or when the projection profiles are evenly distributed so that subsequent divisions are meaningless. Using the described procedure we can build a structural decomposition of the façade in the form of a tree (see Figure 1). Each node of that tree covers a certain part of the input image. The root node corresponds to the whole image. All internal nodes have three children, according to the division procedure. From our construction it follows that the internal nodes of the tree are increasingly framed due to the fact that in each recursive step we eliminate parts of the image without edges. The construction of the tree also provides rectangular nodes with sides aligned to the image axes, which is in accordance with the simplified definition of a window. Each node of the generated tree structure is a candidate for the window.

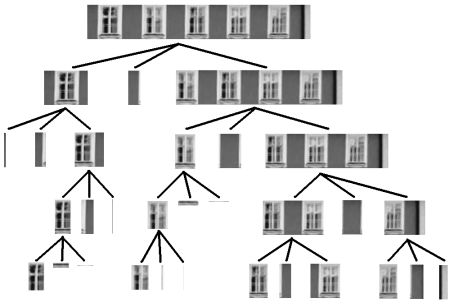


Fig. 1. Structural decomposition of the façade in the form of a tree

3.2. Evaluating candidates

For the segmentation of the façade, it is necessary to evaluate all candidates and to declare the best ones as windows. We evaluate candidates using heuristics based on the position in the tree structure, edge distribution, similarity, and the learned window appearance.

The simplest heuristic (f_1) is based on the position in the tree structure. It is derived directly from the hierarchy. Each internal node of the tree has three children, and the middle child covers the least amount of edge pixels. From this we can conclude that the middle child covers a wall of the façade, so we can assign a negative rating to it, and thus reduce the expectation that it corresponds to a window. The heuristic is neutral to the other two child nodes.

The second heuristic (f_2) is based on the edge distribution, and is similar to the original method of Lee and Nevatia. We want to positively evaluate candidates that are located at intersections between histogram peaks of projection profiles. With this objective we set up a two-dimensional accumulator array of the same size as the input image. Every histogram peak votes for its vertical or horizontal strip in the accumulator array, depending on the projection profile it belongs (see Figure 2). The evaluation of a candidate is defined as the average value of the accumulator array cells from the area that corresponds to that candidate.

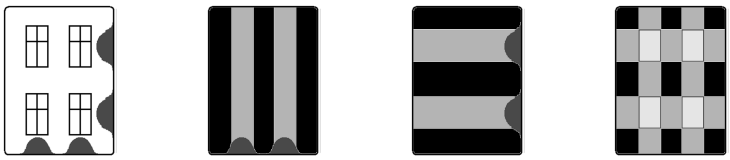


Fig. 2. Accumulation of histogram peaks: a) peaks correspond to actual windows, b) horizontal peaks update vertical strips, c) vertical peaks update horizontal strips, and d) accumulator array cells after updating

The idea of the next heuristic (f_3) follows from the observation that individual windows on the same façade are similar to each other. The quality of each candidate is as-

sessed as the degree of visual similarity with other candidates, which can be measured, for example, using the normalized correlation coefficient. A nice property of this measure is its insensitivity to the changes in image brightness and contrast, which is a common phenomenon in the façade images.

Viola and Jones [21] have developed a framework to train a cascade classifier for the face detection task, which due to its generality is suitable for the detection of arbitrary types of objects. We used this framework to learn the visual appearance of façade windows. We have created a training set using the reference image library ZuBuD [22] and TSG-60 [1]. The positive example set consists of 1529 manually marked windows (see Figure 3). The negative examples set contains 4668 objects that often appear in input images, but are not windows (e.g. façade ornaments, traffic signs, trees, parked vehicles, passers-by etc.). Some examples from the negative training set are shown in Figure 4. Using the described training set we have built an 18-stage cascade classifier. Again, we have used an accumulator array that has been incremented in regions where the classifier detected window patterns. As before, the evaluation of a candidate (f_4) is defined as the average of its accumulator array cells.

Fig. 3. Examples from the positive training set

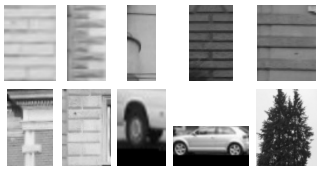


Fig. 4. Examples from the negative training set

Each of the described heuristics can produce false window detections (e.g. the lower levels nodes in the tree structure are less reliably estimated by the heuristics f_1 due to the larger impact of local noise; not all intersections between histogram peaks correspond to the actual windows, as it is provided by the heuristics f_2, etc.). Due to the fact that the heuristics are based on different criteria, we can expect to reduce probability for systematic errors in the same regions of the input image. For this reason, the final evaluation of the quality Q of a candidate C is obtained as a linear combination of individual heuristic evaluations:

$$Q(C) = \sum_{i=1}^{N} \alpha_i f_i(C) \tag{1}$$

where N denotes the number of used heuristics f_i, while α_i are weights (experimentally determined using a validation set of façade images, see Section 4 for details). Thus, we favor candidates correctly estimated by the majority of heuristics and reduce the impact of individual false classification.

[1] obtained from http://dib.joanneum.at/cape/TSG-60/

3.3. Façade modeling using a formal grammar

The last step in forming a hypothesis on the location and size of windows is a selection of the best candidates. We used a context-free grammar, whose production rules enforce the symmetry, alignment, and repetitive patterns of the final hypothesis. Their role is to guide the search in a space of all possible structural descriptions of the input façade image.

The basic idea of our method is to create a likelihood map of windows locations that is explained by a facade model in the form of a derivation tree according to the provided production rules. The aforementioned likelihood map is formed as follows: We first build a structural decomposition of the façade in the form of a tree and then use the described heuristics to assess the individual nodes i.e. window candidates. Each candidate is then projected into a two-dimensional accumulator array. By doing so, a candidate updates the corresponding accumulator cells using its estimated quality Q(C), encoded as a pixel gray level. Therefore, the accumulator field can be interpreted as a grayscale image where brighter pixels are more likely to belong to the façade's windows. In order to simplify processing and increase the robustness of the procedure, we perform thresholding of the likelihood map. A threshold can be set manually or automatically – its aim is to remove the noise from the image i.e. pixels with low probability of belonging to the actual windows on the façade. Figure 5 shows the creation of the window likelihood map for an input image.

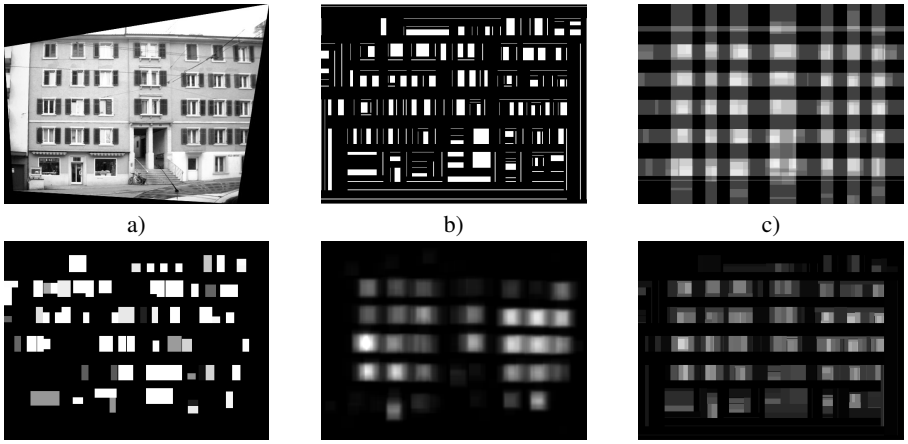


Fig. 5. Creation of the window likelihood map. a) input image. b) - e) cell values in the accumulator array after applying the heuristics f_1, f_2, f_3, and f_4, respectively; lighter dots correspond to higher values. The resulting images for heuristics f_1 and f_3 were obtained by averaging the assessments of individual candidates. f) window likelihood map; lighter dots correspond to higher probabilities

Once the likelihood map has been computed, the search for an appropriate façade structural description can be performed. We are interested to find such a description that puts windows on the bright areas, and the wall on the dark areas of the likelihood map. At the same time we want a description in accordance to the structures that are usually

perceived in the façade images. To describe symmetry, alignment, and repetitive patterns of the façade's elements, we use a formal grammar.

Motivated by the work of Alegre and Dellaert [4], we used a grammar consisting of production rules for creating an image, similar to the input likelihood map. The grammar's start symbol represents the entire region of the input image. A set of segmentation rules provide a division of the starting region into rectangular sub-regions in horizontal or vertical directions by generating new non-terminal symbols. Each newly created non-terminal symbol corresponds to one or more sub-regions, depending on the particular rule. In this way, it is possible to use a single non-terminal symbol for modeling different parts of a façade, representing a repetitive pattern. A sequential application of segmentation rules creates a hierarchical structure of sub-regions that can be transformed into terminal symbols using the termination rules. Our grammar contains two terminal symbols corresponding to the window (represented as a white rectangle on a black background) and the wall (represented as a black area). The termination rules are illustrated in Table 1.

Table 1. Grammar termination rules

Termination rule	Graphical presentation	Description
$A \rightarrow window\ (x, y, width, height)$		The parameters define the relative position and dimensions of the window within a region corresponding to a non-terminal symbol A. In the likelihood map representation, the inserted window appears as a white rectangle on a black background.
$A \rightarrow wall$		In the likelihood map representation, a region corresponding to a non-terminal symbol A appears as a black area.

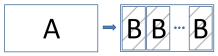
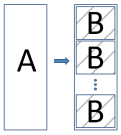
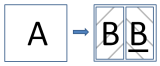
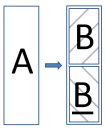
The segmentation rules are of the form:

$$A \rightarrow op\ (V_1, V_2, \ldots, V_n)\ (id_1, id_2, \ldots, id_m)\ (r_1, r_2, \ldots, r_n)$$

where op denotes the direction of division – we use $hsplit$ for horizontal subdivisions and $vsplit$ for vertical subdivisions, V_k is a non-terminal symbol, id_1 to id_m form a sub-region association pattern, and r_1 to r_n are dividing parameters such that $\sum_{i=1}^{n} r_i = 1$. This rule splits a region corresponding to a non-terminal symbol A into m sub-regions ($m \geq n$) along the specified direction. The created sub-regions are associated with non-terminals V_1 to V_n ($n > 0$) in accordance to the specified pattern. Each id_j represents a reference to a non-terminal V_{id_j}, thus $id_j \in \{1, \cdots, n\}$. If a non-terminal symbol V_k is referenced more than once, several sub-regions will be associated with it, which means that derivations from that non-terminal symbol will be applied to all corresponding sub-regions. This allows a uniform modeling of identical or very similar parts of the façade. The association pattern can be extended with an optional label '$\sim$' before a reference id_j that indicates an identical copy, but flipped along the dividing direction. The actual

dimensions of the created sub-regions are determined indirectly by the parameters r_1 to r_n. Each parameter r_k defines the proportion of the total area that will be assigned to the V_k. The segmentation rules that we have used are shown in Tables 2 and 3.

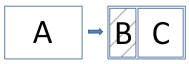
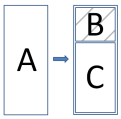
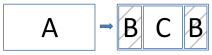
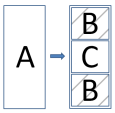
Table 2. Uniform segmentation rules

Segmentation rule	Graphical presentation	Description
$A \rightarrow hsplit\ (B)\ (\underbrace{1, \cdots, 1}_{m})\ (1.0)$		A uniform segmentation into m subregions along the horizontal direction.
$A \rightarrow vsplit\ (B)\ (\underbrace{1, \cdots, 1}_{m})\ (1.0)$		A uniform segmentation into m subregions along the vertical direction.
$A \rightarrow hsplit\ (B)\ (1, \sim 1)\ (1.0)$		A region is split in half along the horizontal direction. An underlined non-terminal symbol indicates an identical copy that is flipped horizontally.
$A \rightarrow vsplit\ (B)\ (1, \sim 1)\ (1.0)$		A region is split in half along the vertical direction. An underlined non-terminal symbol indicates an identical copy that is flipped vertically.

Words derived from such a grammar represent possible configurations of façade windows. Each word has a graphical depiction that can be considered as an ideal likelihood map of window locations. Table 4 illustrates the derivation of a simple façade configuration with the corresponding likelihood map representations.

Searching through the hypothesis space is done as follows: We are interested in finding a word from the language of the grammar, such that its graphical depiction is maximally similar to the input likelihood map. As a measure of similarity between two images we have used the normalized correlation coefficient (although there are other possibilities like the normalized sum of squared differences). The derivation tree for this word represents a structural description of the input façade image. We are interested in finding a simple structural description since, by Occam's razor, it is intuitively clear that simpler hypotheses are more likely to be correct. As a measure of the hypothesis complexity, we use the size of the derivation tree.

Table 3. Non-uniform segmentation rules

Segmentation rule	Graphical presentation	Description
$A \rightarrow hsplit\ (B,C)\ (1,2)\ (r_B, r_C)$		A region is divided along the horizontal direction into two independent sub-regions with the ratio $r_B : r_C$.
$A \rightarrow vsplit\ (B,C)\ (1,2)\ (r_B, r_C)$		A region is divided along the vertical direction into two independent sub-regions with the ratio $r_B : r_C$.
$A \rightarrow hsplit\ (B,C)\ (1,2,1)\ (r_B, r_C)$		A region is divided along the horizontal direction into three sub-regions with the ratio $\frac{r_B}{2} : r_C : \frac{r_B}{2}$; marginal sub-regions are identical.
$A \rightarrow vsplit\ (B,C)\ (1,2,1)\ (r_B, r_C)$		A region is divided along the vertical direction into three sub-regions with the ratio $\frac{r_B}{2} : r_C : \frac{r_B}{2}$; marginal sub-regions are identical.

In the presented work, we expressed the quality of a hypothesis as a simple linear combination of the form:

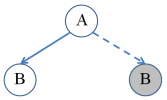
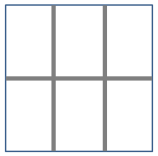
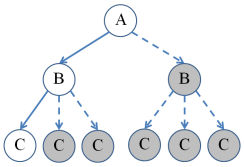
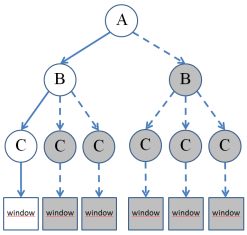
$$Q(H) = \alpha C(H) + (1 - \alpha) S(H) \tag{2}$$

where $Q(H)$, $C(H)$, and $S(H)$ denote the quality, correctness (similarity), and size [2] of a hypothesis H, respectively. Parameter $\alpha \in [0, 1]$ in Equation 2 determines the trade-off between the correctness and complexity of the observed hypothesis. Smaller values favor compact hypotheses. The appropriate value of the parameter should be selected using a validation set of façade images, see Section 4 for details.

The hypothesis space is huge, so it is a common practice to use a stochastic method like the Markov Chain Monte Carlo (MCMC) to guide the search for the (sub)optimal solution. Although this method gives satisfactory results, it is not very efficient. As is usual in MCMC applications, the number of samples tends to be very large – typically in

[2] S(H) is actually inversely proportional to the size of the derivation tree that represents a hypothesis H

Table 4. Example of the derivation process. The left column shows the derivation of a word using production rules and the likelihood map presentation of corresponding façade structure. The right column presents a derivation tree. The gray nodes are references to the nodes of the same name, which means that they can not expand independently. They are shown for better understanding of the derivation process

 A	
 $vsplit\ (B)\ (1,1)\ (1.0)$	
 $vsplit\ (\ hsplit\ (C)\ (1,1,1)\ (1.0)\ )\ (1,1)\ (1.0)$	
 $vsplit\ (\ hsplit\ (\ window(0.25,0.25,0.5,0.5)\ )\ (1,1,1)\ (1.0)\ )\ (1,1)\ (1.0)$	

order of hundreds of thousands samples. It is also very important to properly design the proposal distribution, otherwise the chain will converge very slowly.

We tried a different approach, based on the following observation: façades usually have a highly regular structure. This means that, in practice, the number of different façade configurations is relatively small, and it is possible to systematically inspect virtually all of them. We used a look-ahead greedy search method in the grammar derivation space to select the best façade model. The procedure starts with a non-terminal symbol A associated to the entire input likelihood map I_A representing the root of a derivation tree T_A. At each iteration a non-terminal symbol V in T_A is chosen, and its local likelihood map I_V is calculated by averaging all regions of I_A that correspond to that symbol. Then, a model T_V that describes I_V is built using an exhaustive search in the space of all derivation trees with a maximum depth of 3 (the root node is at depth zero). This space is large enough to capture the basic structure of the input image, usually by applying horizontal and vertical splits in internal nodes of a tree containing terminal symbols in the leaves. The right side of the first production in T_V is then used to replace V in T_A and the iterative procedure is repeated. This hierarchical subdivision identifies simple parts of the input image that can be described using only terminal symbols. The procedure stops when it runs out of non-terminal symbols. The described procedure is shown as Algorithm (1). An example of an iterative derivation is shown in Figure 6.

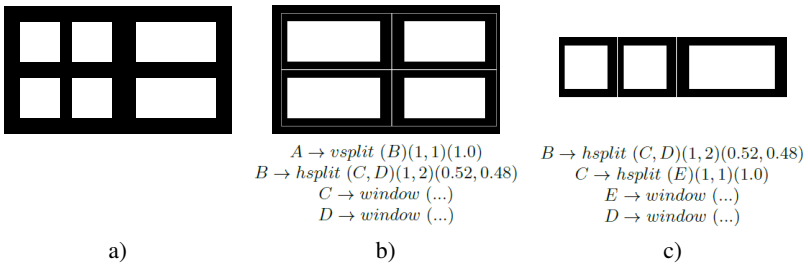


Fig. 6. Example of an iterative derivation procedure: a) input likelihood map; b) derivation tree of depth 3 is not descriptive enough to capture the structure of input image, the final model preserves only the first production; c) the next iteration is focused on describing the vertical stripes

An exhaustive search in the hypotheses space requires the construction of all possible derivation trees. Moreover, it is necessary to try all possible combinations of parameters in the production rules, which is computationally too expensive. For this reason, we have to settle for a properly selected, small subset of parameter combinations. In determining the appropriate values for dividing parameters r_i in the non-uniform segmentation rules, we used the projection profiles again. In general, it is desirable that each window is defined by a single terminal symbol (which does not exclude the possibility that the individual non-terminal symbol corresponds to many similar façade windows). When using production rules for the region segmentation we want to avoid divisions through aligned windows, therefore a natural choice is to divide regions in the middle between the peaks of the profile's histogram (see Figure 7).

Algorithm 1 Grammar-based method

Require: Window likelihood map I,
α - a parameter in Equation 2,
Ensure: A derivation tree T that fits to I.
$A \leftarrow$ non-terminal symbol.
$region(A) \leftarrow I$.
$root(T) \leftarrow A$.
repeat
  Select a non-terminal symbol V in T.
  $I_V \leftarrow region(V)$.
  $B \leftarrow$ an empty tree.
  $Q_B \leftarrow 0$.
  for all trees T_V of depth ≤ 3 with the root V **do**
    $M \leftarrow$ likelihood map representation of T_V.
    $C_V \leftarrow$ similarity between M and I_V.
    $S_V \leftarrow 1/($size of $T_V)$.
    $Q_V \leftarrow \alpha * C_V + (1 - \alpha) * S_V$
    if $Q_V > Q_B$ **then**
      $B \leftarrow T_V$.
      $Q_B \leftarrow Q_V$.
    end if
  end for
  Replace V in T with the $root(B)$.
  for all child nodes V_i of V **do**
    $V_i \leftarrow$ non-terminal symbol.
    $region(V_i) \leftarrow$ average image of corresponding
      sub-regions of I_V.
  end for
until no non-terminal symbol left in T.

To improve the detection of symmetrical and repetitive window patterns and reduce the derivation tree, each non-terminal symbol can be extended with the definition of its region of interest (ROI) within its corresponding region. The ROI is constructed to cut off the irrelevant parts of the region without edge points and is positioned so that the histogram peaks are centered (see Figure 8).

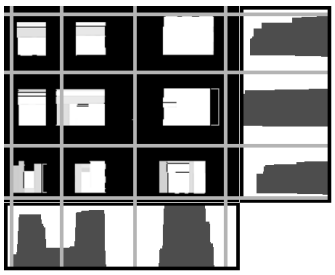


Fig. 7. Region division boundaries. The coordinates of the lines between the peaks of histograms are used as dividing parameters in the non-uniform segmentation rules

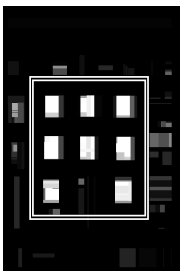


Fig. 8. Example of a window likelihood map and a properly selected region of interest

4. Empirical evaluation

We implemented the described methods in C++ using the open source library OpenCV.

As a validation set for parameter tuning, we used 96 photographs of building façades taken in Ljubljana, Slovenia. All photographs were manually annotated concerning ground truth window locations. The values of weights α_i in Equation 1 were obtained by minimizing the sum of the squared differences between the calculated window likelihood maps (using a particular set of weights) and the actual window position maps (according to the ground truth images). The weights for the heuristics f_1, f_2, f_3 and f_4 were found to be 0.1, 0.51, 0.05, and 0.34, respectively.

The value of the parameter α in Equation 2 was obtained by maximizing the mean F-measure over the validation set. The F-measure [23] is the harmonic mean of sensitivity and precision of generated hypotheses about position and size of windows in the input images. We define the sensitivity of the generated hypothesis as a proportion of actual window pixels which are correctly identified. Similarly, the specificity of the generated hypothesis is a proportion of pixels that do not belong to façade windows and are correctly identified as such. The precision is defined as a proportion of pixels covered by the generated hypothesis that actually belong to façade windows. The mean F-measure as a function of the parameter α is shown in Figure 9. In order to avoid overfitting, we set α to 0.8.

The window recognition performance was tested on 60 photographs from the eTRIMS image database [24]. For a 512x768 input image the average processing time was 32s (see Figure 10). Window candidates generation typically requires 0.57s. The heuristic evaluation of generated candidates takes 2s. The generation of a grammar-based hypothesis

requires, on average, the construction of 7551 derivation trees. Processing times were measured on a 1.8GHz Intel Core i7-4500U processor.

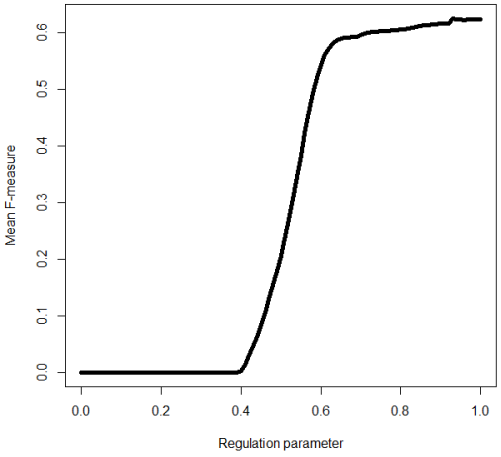


Fig. 9. Mean F-measure as a function of the parameter α in Equation 2 which determines the trade-off between the correctness and complexity of the hypothesis. The results were obtained on the Ljubljana image dataset. Small parameter values are too restrictive, while large values may lead to overfitting

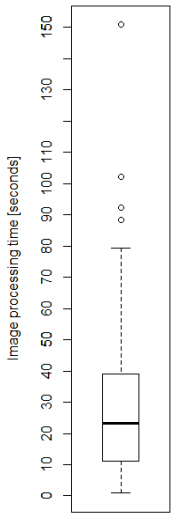


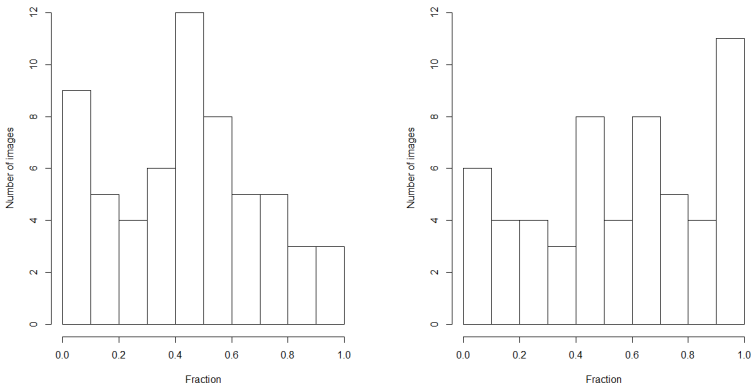
Fig. 10. Boxplot of the processing times measured on the eTRIMS image database

In the first experiment we measured the sensitivity, specificity, and precision of generated hypotheses about the position and size of windows in the input images. The average sensitivity, specificity and precision of hypotheses obtained by the proposed grammar-based method are 51%, 93%, and 57%, respectively. For comparison, the state-of-the-art pixel-wise façade parsing methods achieve higher average window detection sensitivity (80%[14], 78%[25], 75%[13], 71%[26]), and precision (60%[13]), while the specificity is comparable. The reasons for this sub-par performance of our method is discussed in the next section.

Since we are not (primarily) interested in a pixel labelling task, but rather in obtaining a structural description of building façades depicted in the input images, pixel-wise sensitivity and precision are not the most appropriate measures. In the second experiment the final hypothesis was not considered as a set of pixels, but as a set of separate hypotheses about the individual windows (hereinafter referred to as 'hypothesised windows'). This time, we measured the proportion of actual windows which are properly covered by the final hypothesis (which can be considered as "windows-wise" sensitivity of the hypothesis). An appropriate definition of coverage should prefer solutions that exhibit a one-to-one correspondence between the windows in the input image and the hypothesised windows. In other words, we are interested in the solutions where every window in the

input image is (sufficiently) covered with exactly one hypothesised window, and every hypothesised window (precisely enough) covers exactly one window in the input image.

In this manner, a hypothesised window is declared as a valid cover if at least 25% of its pixels overlap the actual windows in the input image, and at least 95% of these pixels overlap the same window. This definition permits a correctly-sized hypothesised window to be misaligned up to half its height and width (or a correctly aligned hypothesised window to be twice in height and width compared to the actual size), as long as it (almost) exclusively overlaps a single actual window. An actual window in the input image was declared as covered, if at least 25% of its pixels are overlapped with valid covers (i.e. hypothesised windows) and at least 95% of these pixels are overlapped by a single valid cover. Our rationale is similar to that in the previous case, but this time we allow coverage by the hypothesised window that is the half-height and half-width of the actual window. Such a definition of coverage allows a certain misalignment between the hypothesised and actual windows, as long as we have a clear one-to-one correspondence between them. The experiment revealed that, on average, the generated hypotheses cover 45% of the actual windows present in the input image (which can be considered as "windows-wise" precision of the hypothesis). The results are shown in Figure 11a. We also measured the fraction of hypothesised windows that actually cover façade windows (which can be considered as "windows-wise" precision of the hypothesis) and obtained an average value of 56% (see Figure 11b for more detailed results).



a) b)

Fig. 11. Fraction of a) actual windows which are properly covered by the generated hypotheses, and b) hypothesised windows that actually cover façade windows, measured on the eTRIMS image dataset

5. Discussion

The proposed method has proven successful in analyzing the highly structured façades with large numbers of aligned windows. This is not surprising, as the candidates generating process and the heuristic evaluation function are based on the projection profile histograms, which are more pronounced in the case of aligned edges. The actual window alignment also improves the formation of the final hypotheses, which favors grid structured candidates. Figure 12 shows some examples of good final hypotheses obtained with the proposed method.

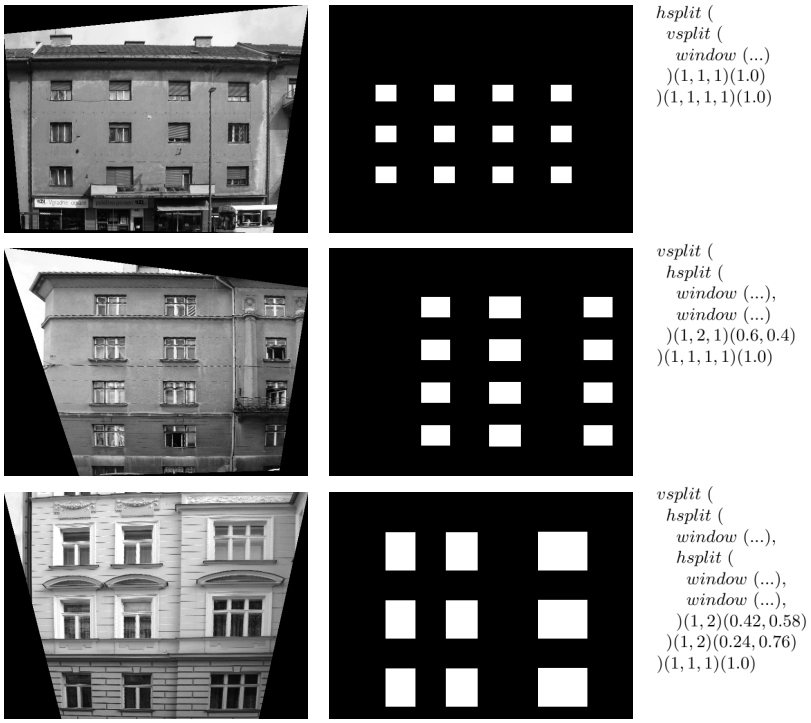


Fig. 12. Examples of good final hypotheses

The results shown in Figure 11 suggest that there are quite a few test images on which the proposed method proved to be completely unsuccessful. These are mainly the images of façades either with a small number of large windows, or with widely spaced groups of small windows. In the first case, the evaluation heuristics (especially those based on the distribution of edges and similarity) are usually weak, which results in hypotheses that are composed of many undersized and often randomly positioned windows. In the second case, the whole group of windows are modeled as a single large window. Figure 13 shows some examples of problematic façades and poor final hypotheses.

The mean pixel-wise sensitivity and precision of generated hypotheses are not impressive, especially in comparison to the state-of-the-art pixel classification methods. The

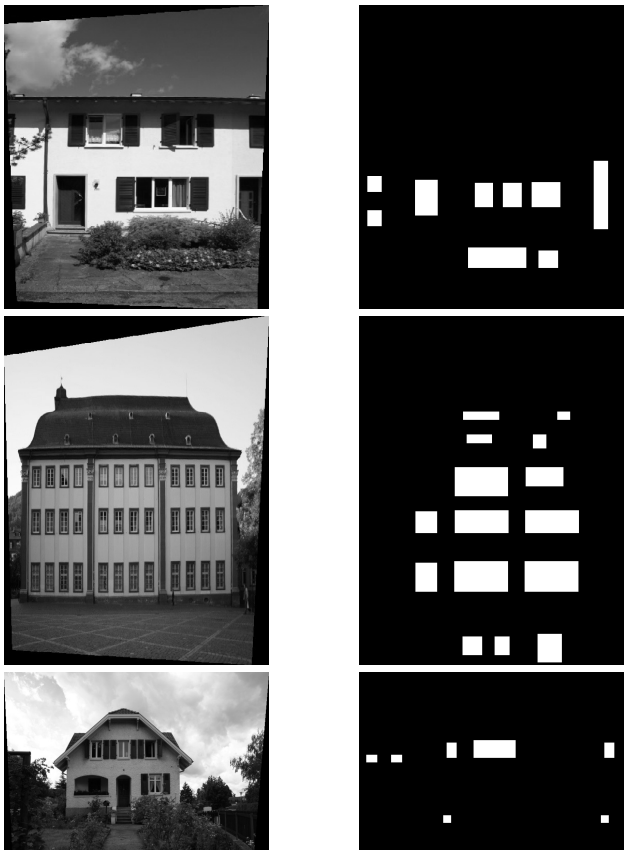
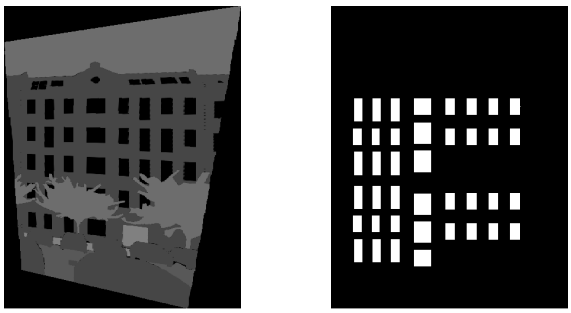


Fig. 13. Examples of a poor final hypothesis. The top row represents an example of a façade with very few large windows. In such cases, the generated hypothesised windows are too small and do not cover a sufficiently large proportion of pixels belonging to the actual windows. The middle row illustrates a situation where the input façade consists of widely spaced groups of small windows. In such cases, the generated hypothesised windows are too large and cover more than a single actual window. The bottom row shows an example of an inaccurate final hypothesis generated when the input image contains a lot of clutter which reduces the effectiveness of the heuristics used to predict the location and size of windows

inferior performance can be attributed to the fact that the pixel labelling process is not carried out separately for individual pixels, but governed by a set of grammar production rules. The production rules enforce symmetry, alignment, and repetitive patterns of the final hypothesis. Although such a rule-based approach may reduce the overall flexibility of the model (reflected by a lower sensitivity and precision), it has a huge advantage – interpretability, which is the basis for further high-level processing. We conclude the discussion with an observation that extracting the regular structure in façade images and pixel labelling are sometimes conflicting tasks. Figure 14 illustrates a situation where a part of the façade is occluded by nearby trees. Some of the pixels in the occluded part of the façade can be classified as a window. According to the actual façade structure, this

would be considered correct, while according to the ground truth image, this is wrong. The opposite applies if the pixels are classified as a tree.



a) b)

Fig. 14. Example of windows hallucination: a) ground truth - the building façade is partially occluded by trees; b) due to the presence of a regular pattern, the final hypothesis assumes the existence of windows in the occluded parts of the façade. Although the result is not entirely correct, it represents a promising capability of the rule-based models

6. Conclusion

We have described a method for learning and recognizing windows as basic structural elements of façades. The method assumes rectangular windows and is adapted to the segmentation of façades which are horizontally and vertically aligned. The experimental results show that the automatically provided annotations are accurate enough to serve as the first approximation and the time for the necessary interventions by the user is significantly lowered. As such, our work is ready to become part of a semi-automatic tool which can be used in navigation, architecture, urban planning, art history, etc. The output of our approach in the form of grammar offers new possibilities for matching and fine-tuning of the results. In the further work we will develop additional heuristics for windows detection that utilize more background knowledge about structure of symmetrical and repeated patterns of façades. One interesting idea is to use inductive logic programming (ILP) to find a logical definition of a window as a certain configuration of edges. We are also interested in extending this work to include other façade structural elements such as doors and balconies.

References

1. Stiny, G., Gips, J.: Shape Grammars and the Generative Specification of Painting and Sculpture. Information processing 71, 1460–1465 (1971)
2. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W.: Instant architecture. ACM Trans. Graph., 22 (3), 669–677 (2003)
3. Müller, P., Wonka, P., Haegler, S., Ulmer, A., Van Gool, L.: Procedural modeling of buildings. ACM Trans. Graph., 25 (3), 614–623 (2006)

4. Alegre, F., Dellaert, F.: A Probabilistic Approach to the Semantic Interpretation of Building Facades. CIPA International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres (2004)
5. Neal, R. M.: Probabilistic inference using Markov Chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 144 pages (1993)
6. Mayer, H., Reznik, S.: MCMC Linked with Implicit Shape Models and Plane Sweeping for 3D Building Facade Interpretation in Image Sequences. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 36 (3), 130–135 (2006)
7. Dick, A. R., Torr, P. H. S., Cipolla, R.: Modelling and Interpretation of Architecture from Several Images. *International Journal of Computer Vision*, 60 (2), 111–134 (2004)
8. Green, P. J.: Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination.Biometrika, 82, 711–732 (1995)
9. Ripperda, N., Brenner, C.: Reconstruction of Façade Structures Using a Formal Grammar and RjMCMC. In: K. Franke, K.-R. Müller, B. Nickolay & R. Schäfer (eds), Pattern Recognition, Proceedings of the 28th DAGM Symposium, 750–759 (2006)
10. Teboul, O.,Kokkinos, I., Simon, L., Koutsourakis, P., Paragios, N.: Shape grammar parsing via Reinforcement Learning. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2273–2280 (2011)
11. Becker, S.: Generation and application of rules for quality dependent façade reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing, 64 (6), 640–653 (2009)
12. Martinovic, A., Van Gool, L.: Bayesian Grammar Learning for Inverse Procedural Modeling. Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 201–208 (2013)
13. Yang, M.Y., Förstner, W.: Regionwise classification of building facade images. Photogrammetric Image Analysis, 209–220 (2011)
14. Martinović, A., Mathias, M., Weissenberg, J., Van Gool, L.: A three-layered approach to facade parsing. Computer Vision–ECCV 2012, 416–429 (2012)
15. Tyleček, R., Šára, R.: Stochastic Recognition of Regular Structures in Facade Images. IPSJ Transactions on Computer Vision and Applications, 4(1), 63–70 (2012)
16. Müller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. ACM Trans. Graph., 26 (3), 9 pages (2007)
17. Lee, S. C., Nevatia, R.: Extraction and integration of window in a 3D building model from ground view images. Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition (CVPR 2004), 113–120 (2004)
18. Haugeard, J.-E., Philipp-Foliguet, S., Precioso, F., Lebrun, J.: Extraction of Windows in Facade Using Kernel on Graph of Contours. Proceedings of the 16th Scandinavian Conference on Image Analysis (SCIA '09), 646–656 (2009)
19. Liu, J., Psarakis, E., Stamos, I.: Automatic Kronecker Product Model Based Detection of Repeated Patterns in 2D Urban Images. Proceedings of the 2013 IEEE International Conference on Computer Vision, 401–408 (2013)
20. Liebowitz, D., Zisserman, A.: Metric rectification for perspective images of planes. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 482–488 (1998)
21. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), 511–518 (2001)
22. Shao, H., Svoboda, T., Van Gool, L.: ZuBuD - Zurich Buildings Database for Image Based Recognition. Technical Report No. 260, Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland (2003)
23. van Rijsbergen, C.: Information Retireval, 2nd edn., Butterworths, London (1979)
24. Korč, F., Förstner, W.: eTRIMS Image Database for Interpreting Images of Man-Made Scenes, Technical Report TR-IGG-P-2009-01, Dept. of Photogrammetry, University of Bonn (2009)

25. Gatta, C., Ciompi, F.: Stacked Sequential Scale Space Taylor Context. IEEE Transaction on Pattern Analysis and Machine Intelligence, 36(8), 1694–1700 (2014)
26. Cohen, A., Schwing, A., Pollefeys, M.: Efficient structured parsing of facades using dynamic programming. Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 3206–3213 (2014)

Petar Vračar received his MSc in 2010 in computer science from University of Ljubljana, Slovenia. He is a teaching assistant at the Faculty of Computer and Information Science, where he assists in teaching several courses on algorithms, data structures, and machine learning. His research interests include sports forecasting, team sports simulation, data mining, and machine learning based image processing. He is a (co)author of 8 publications in scientific journals and conference proceedings.

Igor Kononenko received his Ph.D. in 1990 in computer science from University of Ljubljana, Slovenia. He is a professor at the Faculty of Computer and Information Science in Ljubljana, the head of Laboratory for Cognitive Modeling and the Head of AI department. His research interests include artificial intelligence, machine learning, neural networks and cognitive modeling. He is the (co)author of 225 scientific papers and 12 textbooks, among others the book Machine Learning and Data Mining: Introduction to Principles and Algorithms, Horwood 2007. Kononenko is a member of the editorial board of Applied Intelligence and Informatica journals.

Marko Robnik-Šikonja received his Ph.D. in computer science in 2001 from the University of Ljubljana. He is an Associate Professor at the University of Ljubljana, Faculty of Computer and Information Science. His research interest include machine learning, data and text mining, knowledge discovery, cognitive modeling, and their practical applications. He is a (co)author of more than 50 publications in scientific journals and international conferences and three open-source data analytics tools.

Received: February 22, 2015; Accepted: October 27, 2015.