# A MRF Shape Prior for Facade Parsing with Occlusions

Mateusz Koziński, Raghudeep Gadde, Sergey Zagoruyko, Guillaume Obozinski and Renaud Marlet
Université Paris-Est, LIGM (UMR CNRS 8049), ENPC, F-77455 Marne-la-Vallée, e-mail: {*name.surname*}@enpc.fr

## Abstract

*We present a new shape prior formalism for the segmentation of rectified facade images. It combines the simplicity of split grammars with unprecedented expressive power: the capability of encoding simultaneous alignment in two dimensions, facade occlusions and irregular boundaries between facade elements. We formulate the task of finding the most likely image segmentation conforming to a prior of the proposed form as a MAP-MRF problem over a 4-connected pixel grid, and propose an efficient optimization algorithm for solving it. Our method simultaneously segments the visible and occluding objects, and recovers the structure of the occluded facade. We demonstrate state-of-the-art results on a number of facade segmentation datasets.*

## 1. Introduction

The goal of facade parsing is to segment a rectified image of a building facade into regions corresponding to architectural elements, like windows, balconies and doors. Applications of facade parsing include creating 3D models of buildings for games, thermal simulations, or architectural design. A specificity of facade parsing as compared to general image segmentation, is that we have strong prior knowledge on which combinations of facade elements are semantically valid. For example, windows in a given floor are usually aligned and a balcony needs to be adjacent to the lower part of at least one window. We consider that the set of semantic constraints on the layout of facade elements is specified by the user for a given dataset. The quality of facade segmentation, as perceived by a human, suffers a lot if these semantic constraints are not satisfied.

### 1.1. Related work

One possible approach to the problem is to enforce the structural constraints on results of a general-purpose segmentation algorithm. Martinović *et al*. [7] combine results of a Recursive Neural Network with object detections to form unary potentials of a Markov Random Field encoding an initial image segmentation. The initial segmentation is modified to satisfy a number of 'weak architectural principles': some elements are given rectangular shapes; rectangles, boundaries of which are sufficiently close, are aligned; doors are inserted into the lower parts of facades. However, the set of 'architectural principles' is different for each dataset and no formal way of specifying them has been proposed. Moreover, applying local corrections to a segmentation (e.g., aligning lines that are close enough) does not necessarily yield a semantically correct segmentation.

The structural constraints can also be hard-coded in the parsing algorithm. In the work by Cohen *et al*. [1] a sequence of dynamic programs (DPs) is run on an input image, each of which makes the current labeling more detailed. The first DP operates along the vertical axis and identifies the floors. The following ones identify window columns, the boundary between the sky and roof, the doors, etc. However the algorithm is limited to segmentations that assume the hierarchical structure encoded in the dynamic programs. Besides, the approach neither enforces nor favors simultaneous alignment of shapes in two dimensions.

Teboul *et al*. introduced split grammars as shape priors for facade segmentation [15]. Shape derivation with a split grammar is analogous to string derivation in formal languages, except that the symbols correspond to rectangular image regions and productions split them along one of the coordinate axes. The advantage of this framework is the simplicity and the expressive power of split grammars. The disadvantage is that approximating the optimal segmentation requires randomly generating a large number of shapes and keeping the best one as the final result. Even with robust strategies of data driven exploration of the space of grammar derivations [14, 11, 8], the method still cannot be relied on to repeatedly produce optimal results.

Riemenschneider *et al*. have shown that parsing an image with a two-dimensional grammar can be performed using a variant of the CYK algorithm for parsing string grammars [9]. They also introduced production rules modeling symmetry in facade layouts. However, the high computational complexity of the algorithm makes its direct application on the input image impractical. Instead, the authors subsample images forming irregular grids of approximately 60 by 60 cells and run the algorithm on the subsampled images.

Koziński *et al*. [5] proposed a shape prior formalism

where facade parsing is formulated as a binary linear program. The method enforces horizontal and vertical alignment of facade element simultaneously and yields state of the art results on the ECP and Graz50 datasets. However, the principle of global alignment makes the priors very restrictive. A separate class is needed to model each misaligned facade element (e.g., each floor misaligned with the other ones). This, and the time of around 4 minutes required to segment a single image, make the algorithm impractical for datasets with a high level of structural variation. Moreover, the prior formalism does not allow for modeling non-rectangular shapes or occlusions.

### 1.2. Contribution

We present a facade segmentation framework based on user-defined shape priors. Our shape prior formalism is based on a hierarchical partitioning of the image into grids, possibly with non-linear boundaries between cells. Its advantage over the split grammar formalism [15, 14, 11, 9] is that it explicitly encodes simultaneous alignment in two dimensions. Encoding this constraint using a split grammar requires an extension which makes the grammars context-dependent. While a method of encoding bidirectional alignment has been proposed in [5], the priors defined in that formalism enforce global alignment in a very restrictive way: all segments of the same class must be aligned, so that, for example, a separate window class needs to be defined for each floor with a distinct pattern of windows. Our shape prior formalism has the advantage of being conceptually simpler and more flexible thanks to explicit encoding of the alignment constraints.

In the proposed framework, parsing is formulated as a MAP-MRF problem over a 4-connected pixel grid with hard constraints on the classes of neighboring pixels. The existing shape prior-based parsers are based on randomized exploration of the space of shapes derived from the grammar [14, 11, 8] or require severe image subsampling [9]. Although a linear formulation that does not require sampling was proposed recently [5] our formulation is simpler and more intuitive, and results both in significantly shorter running times and more accurate segmentations. In our experiments, our method systematically yields accuracy superior to existing methods given the same per-pixel costs.

Last but not least, our new shape prior formalism allows two extensions: we show that unlike existing prior formalisms [14, 11, 9, 5], that are limited to rectangular tilings of the image, we can model more general boundaries between segments. We also extend our prior formalism to model possible occlusions and to recover both the occluding object boundaries and the structure of the occluded parts of the facade.

Table 1. Comparison of selected properties of state-of-the-art facade parsing algorithms.

| | [14] | [9] | [7] | [1] | [5] | ours |
|---|---|---|---|---|---|---|
| User-defined shape prior | ✓ | ✓ | – | – | ✓ | ✓ |
| Occlusions and irregular shapes | – | – | ✓ | ✓ | – | ✓ |
| Simultaneous alignment in 2D | – | ✓ | ✓ | – | ✓ | ✓ |
| No need of image subsampling | ✓ | – | ✓ | ✓ | ✓ | ✓ |
| No need of sampling from a grammar | – | ✓ | ✓ | ✓ | ✓ | ✓ |

### 1.3. Outline of the paper

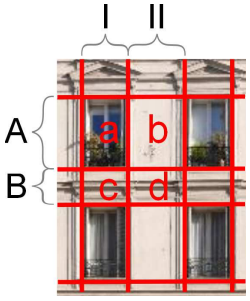
In the next section, we present the new shape prior formalism and show that it can be expressed in terms of classes assigned to image pixels and constraints on classes of pairs of neighboring pixels. This enables formulating the problem of optimal facade segmentation in terms of the most likely configuration of a Markov Random Field with hard constraints on neighbor classes. We present this formulation in section 3. In section 4 we show how to apply dual decomposition to perform inference in our model. We present the experiments in section 5.

## 2. Adjacency patterns as shape priors

Simultaneous vertical and horizontal alignments are prevalent in facade layouts. To encode shape priors expressing such alignments, as well as more complex shapes, we introduce the notion of adjacency patterns.

### 2.1. From grid patterns to pixel adjacencies

Consider a shape prior encoding a grid pattern, which can be specified in terms of the set of column classes $\mathcal{C}$ and the set of row classes $\mathcal{R}$. By assigning a column class $c \in \mathcal{C}$ to each image column, and a row class $r \in \mathcal{R}$ to each image row, we implicitly label each pixel with a pair $(c, r)$ of a column class and a row class. We call such pairs $(c, r) \in \mathcal{R} \times \mathcal{C}$ 'pre-semantic' classes. We define a set of 'semantic' classes $K$ encoding types of facade elements (like wall, window, etc), and a mapping $\Psi$ that assigns to each pre-semantic class $(c, r) \in \mathcal{R} \times \mathcal{C}$ a semantic class $k \in K$. For facade parsing it is reasonable to prohibit some combinations of neighboring row or column classes. For example, segmentations where 'roof' is above 'sky' can be viewed as invalid. To encode such preferences, we can specify the set of ordered pairs of column classes that can be assigned to adjacent image columns $\mathcal{H} \subset \mathcal{C} \times \mathcal{C}$, and the set of ordered pairs of adjacent row classes, $\mathcal{V} \subset \mathcal{R} \times \mathcal{R}$. We call a shape prior of the form $\mathcal{G} = (\mathcal{C}, \mathcal{R}, \mathcal{H}, \mathcal{V})$ a grid pattern.




$\mathcal{R} = \{A, B\}, \quad \mathcal{C} = \{I, II\},$
$\Psi(A, I) = \text{window},$
$\Psi(A, II) = \text{wall},$
$\Psi(B, I) = \text{wall},$
$\Psi(B, II) = \text{wall},$
$\mathcal{V} = \{(A, B), (B, A)\},$
$\mathcal{H} = \{(I, II), (II, I)\}.$

$(A, I) = a,$
$(A, II) = b,$
$(B, I) = c,$
$(B, II) = d.$

horiz. neighbors

| h↱ | a | b | c | d |
|---|---|---|---|---|
| a | + | + | – | – |
| b | + | + | – | – |
| c | – | – | + | + |
| d | – | – | + | + |

vert. neighbors

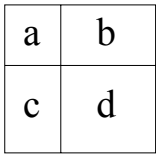
| v↱ | a | b | c | d |
|---|---|---|---|---|
| a | + | – | + | – |
| b | – | + | – | + |
| c | + | – | + | – |
| d | – | + | – | + |

Figure 1. *Top left*: grid-shaped segmentation with row, column and pixel classes. *Top right*: Specification of the corresponding grid pattern using row and column classes. *Bottom*: Specification of the same grid pattern using allowed vertical and horizontal pixel neighbors ('+' denotes an allowed adjacency, '–' a forbidden one).

We now introduce an alternative encoding of shape priors, that it is capable of expressing grid patterns and more general priors. We define an 'adjacency pattern' as a triple $A = (S, V, H)$ where $S$ is a finite set of (pre-semantic) classes, and $V \subset S \times S$ and $H \subset S \times S$ are sets of ordered pairs of classes that can be assigned to vertically and horizontally adjacent pixels. A pair of vertically adjacent pixels can be labeled in such a way that a pixel of class $s_1$ is immediately below a pixel of class $s_2$ only if $(s_1, s_2) \in V$. The same holds for any pair of horizontally adjacent pixels and the set $H$.

To show that the expressive power of adjacency patterns is at least as high as that of grid patterns, we construct an adjacency pattern $A^{\mathcal{G}} = (S^{\mathcal{G}}, V^{\mathcal{G}}, H^{\mathcal{G}})$ equivalent to a given grid pattern $\mathcal{G} = (\mathcal{C}, \mathcal{R}, \mathcal{H}, \mathcal{V})$. We set $S^{\mathcal{G}} = \mathcal{R} \times \mathcal{C}$. In consequence, the sets of classes assigned to image pixels are the same for both types of priors. For a pixel class $s = (r_s, c_s)$, $r_s \in \mathcal{R}$, $c_s \in \mathcal{C}$ we denote its row-class component by $r(s) = r_s$ and its column-class component by $c(s) = c_s$. We enforce that the rows of a labeling conforming to the adjacency pattern are valid rows of the grid pattern by requiring that each two horizontally adjacent pixels receive classes with the same row-class component, and similarly for vertically adjacent pixels and the column-class component of pixel classes. We also reformulate the constraints on classes of neighboring rows and columns of the grid pattern in terms of the row- and column-class components of pixel classes of the adjacency pattern. We define the sets of allowed classes of adjacent pixels as:

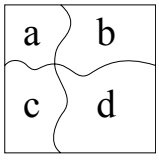
$$V^{\mathcal{G}} = \big\{(s_1, s_2) | c(s_1) = c(s_2) \wedge \big(r(s_1), r(s_2)\big) \in \mathcal{V}\big\}, \quad \text{(1a)}$$
$$H^{\mathcal{G}} = \big\{(s_1, s_2) | r(s_1) = r(s_2) \wedge \big(c(s_1), c(s_2)\big) \in \mathcal{H}\big\}. \quad \text{(1b)}$$




| h↱ | a | b | c | d |
|---|---|---|---|---|
| a | + | + | – | – |
| b | – | + | – | – |
| c | – | – | + | + |
| d | – | – | – | + |

| v↱ | a | b | c | d |
|---|---|---|---|---|
| a | + | – | – | – |
| b | – | + | – | – |
| c | + | – | + | – |
| d | – | + | – | + |

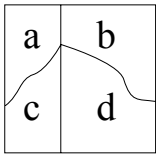
(a) A non-repeating pattern with straight, axis-aligned boundaries.




| h↱ | a | b | c | d |
|---|---|---|---|---|
| a | + | + | + | – |
| b | – | + | – | + |
| c | + | – | + | + |
| d | – | + | – | + |

| v↱ | a | b | c | d |
|---|---|---|---|---|
| a | + | + | – | – |
| b | + | + | – | – |
| c | + | – | + | + |
| d | – | + | + | + |

(b) A non-repeating pattern with winding, axis-driven boundaries.




| h↱ | a | b | c | d |
|---|---|---|---|---|
| a | + | + | + | – |
| b | – | + | – | – |
| c | – | – | + | + |
| d | – | – | – | + |

| v↱ | a | b | c | d |
|---|---|---|---|---|
| a | + | – | – | – |
| b | – | + | – | – |
| c | + | – | + | – |
| d | – | + | – | + |

(c) A non-repeating pattern on grid with monotonic boundaries.

Figure 2. Shape patterns and corresponding horizontal and vertical compatibility tables for neighboring pixel classes: '+' denotes a pair of allowed neighbors in this order, '–' denotes forbidden pairs.

Fig. 1 presents a grid pattern specification, the equivalent adjacency pattern specification and a corresponding image segmentation.

## 2.2. Handling complex patterns and boundaries

In real images, the boundaries between some semantic classes, like 'roof' and 'sky', are often irregular and cannot be modeled by straight axis-aligned line segments. Priors expressing patterns with such complex boundaries can be encoded in terms of adjacency patterns by properly designing the sets of allowed neighbor classes, $V$ and $H$.

The pattern presented in fig. 1 has straight, axis-aligned boundaries. The pattern can be repeated an indefinite number of times in the horizontal and vertical directions. Fig. 2a presents a non-repeating pattern on a grid with straight axis-aligned boundaries. The difference with respect to the previous case is that here the prior does not allow for repetition of the pattern along the vertical or horizontal direction. As shown in fig. 2b, these straight borders can be turned into irregular winding boundaries by allowing a controlled interpenetration of classes. For instance, on a horizontal line, an 'a' can now be followed by a 'c' and then again by an 'a', but a 'c' on this line still cannot be followed by 'b'. Fig. 2c displays another variant where monotonicity is imposed to a boundary, to represent a rising and a descending border. Such a pattern can be used to model a roof, which is expected to have an ascending slope in the beginning and a descending slope at the end.

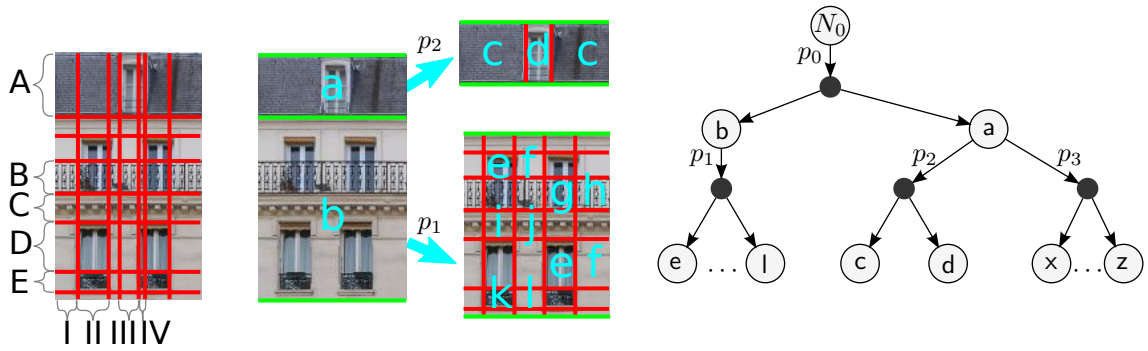



Figure 3. *Left*: modeling a pattern with vertical misalignment as a single grid requires each column class to encode the type of both the element occupying the lower part of the column and the element occupying its upper part: *I - (wall, roof)*, *II - (window, roof)*, *III - (wall, attic window)*, *IV - (window, attic window)*. The number of resulting pixel classes is exponential in the number of misalignments (20 in the depicted case). *Middle*: a hierarchical grid model, where cells of a coarser grid (green) are further subdivided into finer grids (red), results in a set of terminal pixel classes of cardinality linear in the number of misalignments (10 in the example). *Right:* a hierarchy of adjacency patterns corresponding to the labeling in the middle. Large, circled nodes correspond to pixel classes. Small, filled nodes correspond to adjacency patterns. Productions are marked next to arrows that map pixel classes to adjacency patterns. Note that the hierarchy encodes a structural alternative between production $p_2$ and production $p_3$ (not used in the segmentation shown in the middle).

### 2.3. Hierarchical adjacency patterns

Even when it is axis-aligned, the layout of facade elements is usually more complex than a grid and contains many misaligned elements. Encoding such patterns as a single grid requires a number of pixel classes that grows exponentially with the number of misalignments. This is illustrated in fig. 3.

To address this issue, we define a shape prior consisting of a hierarchy of adjacency patterns. The concept is that the pre-semantic pixel classes of an adjacency pattern on a coarser level of the hierarchy are mapped to adjacency patterns on a finer level. A connected region of pixels that received the same pixel class of an adjacency pattern on a coarser level of the hierarchy can be further segmented using a prior encoded by the adjacency pattern on a finer level.

A hierarchical adjacency pattern is a quadruple $\hat{A} = (\mathcal{N}, \mathcal{T}, N_0, \mathcal{P})$ where $\mathcal{N}$ is a finite set of nonterminal classes, $\mathcal{T}$ is a finite set of terminal classes, disjoint from $\mathcal{N}$, $N_0 \in \mathcal{N}$ is the start symbol and $\mathcal{P}$ is a set of production rules of the form $p = N_p \rightarrow A_p$ where $N_p \in \mathcal{N}$ and $A_p = (S_p, V_p, H_p)$ is an adjacency pattern such that $S_p \subset \mathcal{N} \cup \mathcal{T}$. Additionally, we impose that the productions contain no cycle and that the sets of pixel classes in each adjacency pattern $A_p$ are all disjoint.

Now we define conditions of conformance of a segmentation to a hierarchical adjacency pattern. We denote the set of classes descending in the hierarchy from production $p$ by $Desc(p)$, and the set of classes descending from a class $s$ by $Desc(s)$. For a production $p$ and class $s \in Desc(p)$, we define the ancestor class of $s$, belonging to the adjacency pattern $A_p$, by $Anc_p(s) = s'$ s.t. $s' \in S_p$ and $s \in Desc(s')$. For each production $p \in \mathcal{P}$, each region of the labeling that contains only classes $s \in Desc(p)$, must conform to the adjacency pattern $A_p$, when labels of its pixels are changed to their ancestors in $A_p$. We denote the set of indexes of pixels excluding the last image column by $\mathcal{I}_h$, and the set of pixel indexes without the last row by $\mathcal{I}_v$. We denote the class of pixel $(i, j)$ by $s_{ij}$. The conformance conditions:

$$\forall (i,j) \in \mathcal{I}_h,\ \forall p \in \mathcal{P},\ \text{s.t. } s_{ij}, s_{ij+1} \in Desc(p) \\ \big(Anc_p(s_{ij}), Anc_p(s_{ij+1})\big) \in H_p\ , \tag{2a}$$

$$\forall (i,j) \in \mathcal{I}_v,\ \forall p \in \mathcal{P},\ \text{s.t. } s_{ij}, s_{i+1\,j} \in Desc(p) \\ \big(Anc_p(s_{ij}), Anc_p(s_{i+1\,j})\big) \in V_p\ . \tag{2b}$$

A hierarchical adjacency pattern $\hat{A}{=}(\mathcal{N}, \mathcal{T}, N_0, \mathcal{P})$ can be represented as a simple, flattened adjacency pattern $A^f = (S^f, V^f, H^f)$, where $S^f = \mathcal{T}$. The definition of the sets of pairs of classes that can be assigned to vertically and horizontally adjacent pixels, $V^f$ and $H^f$, follows directly from the conformance conditions (2):

$$V^f = \Big\{(t_1, t_2) \in \mathcal{T}^2 \mid \forall p \in \mathcal{P}\ \text{s.t. } t_1, t_2 \in Desc(p) \\ \big(Anc_p(t_1), Anc_p(t_2)\big) \in V_p\Big\} \tag{3a}$$

$$H^f = \Big\{(t_1, t_2) \in \mathcal{T}^2 \mid \forall p \in \mathcal{P}\ \text{s.t. } t_1, t_2 \in Desc(p) \\ \big(Anc_p(t_1), Anc_p(t_2)\big) \in H_p\Big\}\ . \tag{3b}$$

While the hierarchical representation is more conveniently specified by a human user, because it requires defining a lower number of constraints on the classes of adjacent pixels, the 'flat' representation enables formulating the inference in terms of the MAP-MRF problem, as shown in sec. 3.

### 2.4. Handling Occlusions

Occlusions are omnipresent in urban scenes. For facade parsing, the most common occlusions are by trees and lamp

posts. Lower parts of facades can also be occluded by other types of vegetation, street signs, cars and pedestrians.

Given an adjacency pattern $A = (S, V, H)$, we define another adjacency pattern $A^o = (S^o, V^o, H^o)$, encoding shapes consistent with $A$, with possible occlusions by objects of classes from the set $O$, disjoint from the set of pre-semantic classes $S$ and from the set of semantic classes of facade elements $K$. We define a pixel class $\sigma \in S^o$ to have a 'pre-semantic' and a 'semantic' component $\sigma = (s, \kappa)$, where $s \in S$ and $\kappa \in (O \cup K)$. Only a small number of combinations of occluder and pre-semantic classes is semantically meaningful (e.g., pedestrians can occlude the lower part of a facade, but not the roof). We represent the semantically meaningful pairs by a set $\mathfrak{S} \subset S \times O$. We define the set of pixel classes as $S^o = \{(s, \Psi(s)) | s \in S\} \cup \mathfrak{S}$. That is, for a class $\sigma = (s, \kappa)$ representing a non-occluded facade element $\kappa = \Psi(s)$, $\kappa \in K$. For a class $\sigma = (s, \kappa)$ representing an occlusion $(s, \kappa) \in \mathfrak{S}$, $\kappa \in O$. This practically limits the number of classes. In our experiments, it never increased by a factor of more than 2.5, compared to the model without occlusions. We denote the pre-semantic component of class $\sigma = (s_\sigma, \kappa_\sigma)$ by $s(\sigma) = s_\sigma$. The sets $V^o$ and $H^o$ are defined as:

$$V^o = \Big\{(\sigma_1, \sigma_2) | \sigma_1, \sigma_2 \in S^o,\ \big(s(\sigma_1), s(\sigma_2)\big) \in V\Big\}, \quad (4a)$$

$$H^o = \Big\{(\sigma_1, \sigma_2) | \sigma_1, \sigma_2 \in S^o,\ \big(s(\sigma_1), s(\sigma_2)\big) \in H\Big\}. \quad (4b)$$

We define a pairwise potential $\theta_{\sigma\sigma'}$, penalizing frequent transitions between classes $\sigma, \sigma' \in S^o$, to limit noise in the resulting segmentations. The mapping of a pixel class $\sigma = (s, \kappa)$ to semantic or occluder class becomes $\Psi^o(\sigma) = \kappa$.

## 3. Formulation of optimal segmentation

In this section we propose a formulation of the optimal image segmentation that conforms to an adjacency pattern. We denote image height and width by $h$ and $w$, the set of image row indexes $I = \{1, \dots h\}$, the set of column indexes $J = \{1, \dots w\}$, and the set of pixel indexes by $\mathcal{I} = I \times J$. We encode the assignment of a class $\sigma \in S^o$ to a pixel $(i, j) \in \mathcal{I}$ by variables $z_{ij\sigma} \in \{0, 1\}$, where $z_{ij\sigma} = 1$ if $\sigma$ is the class assigned to pixel $(i, j)$ and $z_{ij\sigma} = 0$ otherwise. To enforce the satisfaction of the constraints on classes of neighboring pixels, we also introduce variables $v_{ij\sigma\sigma'} \in \{0, 1\}$ and $u_{ij\sigma\sigma'} \in \{0, 1\}$, such that $u_{ij\sigma\sigma'} = 1$ if pixel $(i, j)$ is assigned class $\sigma$ and pixel $(i, j+1)$ is assigned class $\sigma'$, and $u_{ij\sigma\sigma'} = 0$ otherwise, and similarly for $v_{ij\sigma\sigma'}$ and vertically neighboring pixels. We denote the vectors of all $z_{ij\sigma}$, $u_{ij\sigma\sigma'}$, $v_{ij\sigma\sigma'}$ by $\mathbf{z}$, $\mathbf{u}$, $\mathbf{v}$, respectively. The goal is to find an assignment that minimizes the sum of costs $\phi_{ij\kappa}$ of assigning class $\kappa \in O \cup K$ to pixel $(i, j) \in \mathcal{I}$. We denote the set of all pixels except for the last row by $\mathcal{I}_v = (I \setminus \{h\}) \times J$, and the set of all pixels without the last column by $\mathcal{I}_h = I \times (J \setminus \{w\})$. The objective is

$$\min_{\mathbf{z},\mathbf{v},\mathbf{u}} \sum_{\substack{(i,j)\in\mathcal{I} \\ \sigma\in S^o}} \phi_{ij\Psi^o(\sigma)} z_{ij\sigma} + \sum_{\substack{(i,j)\in\mathcal{I}_v \\ \sigma,\sigma'\in S^o}} \theta_{\sigma\sigma'} v_{ij\sigma\sigma'} + \sum_{\substack{(i,j)\in\mathcal{I}_h \\ \sigma,\sigma'\in S^o}} \theta_{\sigma\sigma'} u_{ij\sigma\sigma'}. \quad (5)$$

We require that exactly one class is assigned to each pixel,

$$\forall (i, j) \in \mathcal{I}, \quad \sum_{\sigma \in S^o} z_{ij\sigma} = 1\ . \quad (6)$$

We impose consistency between variables encoding pixel labels and pairs of labels: $\forall (i, j) \in \mathcal{I}_v, \forall \sigma \in S^o$,

$$\sum_{\sigma' \in S^o} v_{ij\sigma\sigma'} = z_{ij\sigma}, \qquad \sum_{\sigma' \in S^o} v_{ij\sigma'\sigma} = z_{i+1\,j\sigma}, \quad (7)$$

and $\forall (i, j) \in \mathcal{I}_h, \forall \sigma \in S^o$,

$$\sum_{\sigma' \in S^o} u_{ij\sigma\sigma'} = z_{ij\sigma}, \qquad \sum_{\sigma' \in S^o} u_{ij\sigma'\sigma} = z_{ij+1\sigma}. \quad (8)$$

We constrain the pairs of neighboring classes according to:

$$\forall (i, j) \in \mathcal{I}_v, \forall (\sigma, \sigma') \notin V^o, \quad v_{ij\sigma\sigma'} = 0\ , \quad (9a)$$

$$\forall (i, j) \in \mathcal{I}_h, \forall (\sigma, \sigma') \notin H^o, \quad u_{ij\sigma\sigma'} = 0\ . \quad (9b)$$

The model resembles a linear formulation of the most likely configuration of a MRF [16], with the difference of hard constraints on classes of neighboring pixels.

## 4. Inference algorithm

To solve problem (5-9) we assume the dual decomposition approach. We adopt the most standard decomposition of a 4-connected grid into Markov chains over image rows and columns. The resulting subproblems can be solved independently and efficiently using the Viterbi algorithm. For a comprehensive treatment of dual decomposition we refer the reader to [3, 13]. We derive an algorithm specialized to our problem in the supplementary material.

## 5. Experiments

We evaluated the accuracy of our algorithm in segmenting facade images on a wide range of datasets and for unary terms of various quality. We emphasize that our goal is not to establish a new state of the art performance by using more accurate classification algorithms, better features or detections. Instead we demonstrate that the proposed optimization scheme leads to better segmentations given the same bottom-up cues. Moreover, we show that imposing the structural constraints improves parsing results, while previous work [7] suggested that structural correctness comes at a cost of decreased accuracy.

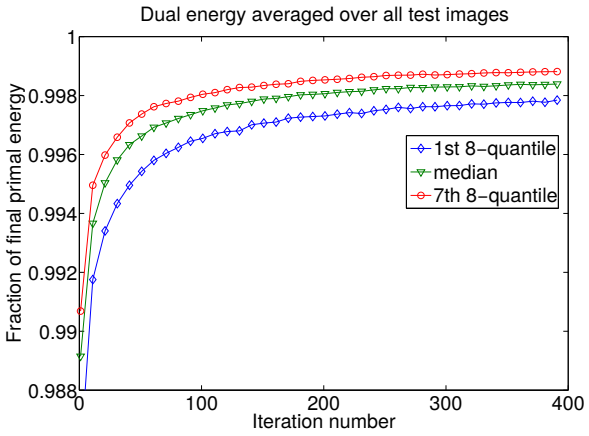



Figure 4. Statistics of the ratio of dual energy to the final primal energy with respect to iteration number. Experiment performed on the ECP dataset.

**Convergence and duality gap** The algorithm operates on the dual problem, yielding a lower bound on the optimal energy. The gap between the dual energy and the energy of the primal binary solution can be seen as a measure of suboptimality of the obtained solution. We analyze the performance of the algorithm on the ECP dataset [14] against the ground truth proposed by Martinović *et al*. [7]. For each image of the test set we record the dual energy in each iteration of the algorithm. We normalize the dual energies with respect to the energy of the final primal solution. We present the statistics in figure 4. For a vast majority of the images the primal-dual gap is not more than 0.2% of the final energy, which indicates that only a very small fraction of the pixel labels are different than at the primal optimum.

**Performance on the ECP dataset** We apply our method to the ECP dataset [14], consisting of 104 images of Haussmannian building facades. We use the ground truth annotations proposed by Martinović *et al*. [7]. We apply the procedure described by Cohen *et al*. [1] to obtain the per-pixel energies: a multi-feature extension of TextonBoost implemented by Ladický *et al*. [6]. We use SIFT, ColorSIFT, Local Binary Patterns and location features. Feature vectors are clustered to create dictionary entries and the final feature vector is a concatenation of histograms of appearance of cluster members in a neighborhood of 200 randomly sampled rectangles. The per-pixel energies are output by a multi-class boosting classifier [10]. Like in [7] and [1] we perform experiments on five folds with 80 training and 20 testing images. The used shape prior models a wide range of structural variation, including possible vertical misalignment of the attic and top floors with the rest of the facade, balconies of two different heights in a single floor and shop windows. The resulting adjacency pattern has 80 classes.

Table 2. Performance on the ECP dataset with unary potentials obtained using a Recursive Neural Network and a variant of TextonBoost [6]. The rows corresponding to classes present class accuracy. The bottom rows contain average class accuracy and total pixel accuracy. In columns, starting from left: performance of the RNN; result of [7]; our result for the same unaries; performance resulting from classifying each pixel separately using the TextonBoost scores; results of Cohen et al. [1]; results of the binary linear program by Koziński *et al*.; our results.

| | RNN unaries | | | TextonBoost unaries | | | |
|---|---|---|---|---|---|---|---|
| | raw | [7] | Ours | raw | [1] | [5] | Ours |
| roof | 70 | 74 | 78 | 89 | 90 | 91 | 91 |
| shop | 79 | 93 | 90 | 95 | 94 | 95 | 97 |
| balcony | 74 | 70 | 76 | 90 | 91 | 90 | 91 |
| sky | 91 | 97 | 94 | 94 | 97 | 96 | 97 |
| window | 62 | 75 | 67 | 86 | 85 | 85 | 87 |
| door | 43 | 67 | 44 | 77 | 79 | 74 | 79 |
| wall | 92 | 88 | 93 | 90 | 90 | 91 | 90 |
| pixel accur. | 82.6 | 84.2 | **86.2** | 90.1 | 90.8 | 90.8 | **91.3** |

As shown in table 2 we outperform state-of-the-art methods that use the same unaries by a small margin. Additionally our algorithm can accept user-defined shape priors, while [1] has hard-coded constraints. Some advantage over [5] comes from a more flexible prior. We also outperform [5] in terms of running time: 100 iterations of our algorithm takes less that 30 seconds (a CPU implementation running on a 3GHz Corei7 processor), compared to 4 minutes in the latter case. For a fair comparison with [7], we perform another experiment on the ECP dataset using the same bottom-up cues as in their paper: the output of a Recursive Neural Network [12], which is less accurate than TextonBoost. For this experiment we use a simple pairwise Potts potential. We set the off-diagonal entries of pairwise cost tables to 0.5, a value determined by grid search on a subset of the training set. The results are presented in table 2. We outperform the baseline [7], even though their segmentation is obtained using window, balcony and door detections in addition to RNN. The influence of the detections on the performance of the baseline can be seen on results for the window and door class, for which the baseline outperforms our algorithm. Our algorithm guarantees semantic correctness of the segmentations, while the baseline aligns facade elements only locally and can yield, for example, balconies ending in the middle of a window.

**Performance on the Graz50 dataset** The Graz50 dataset [9] contains 50 images of various architectural styles labeled with 4 classes. We compare the performance of our algorithm to the method of Riemenschneider *et al*. [9] and Koziński *et al*. [5]. As in the case of the ECP dataset we use the TextonBoost to get unaries. We note that Riemenschneider *et al*. [9] use a different kind of per-pixel energies,

Table 3. Left: results on the Graz50 dataset. The diagonal entries of the confusion matrices for results reported by Riemenschneider et al. [9], Koziński et al. [5], and our results. Right: results on the ArtDeco dataset; raw[1] – pixel classification for a classifier without the vegetation class, raw[2] – pixel classification for a classifier with the vegetation class; ours[3] – the facade structure extracted by our algorithm; ours[4] – the segmentation produced by our algorithm.

| | Graz50 | | | | ArtDeco | | | |
|---|---|---|---|---|---|---|---|---|
| | [9] | [5] | Ours | | raw[1] | raw[2] | ours[3] | ours[4] |
| sky | 91 | 93 | 93 | roof | 82 | 82 | 81 | 82 |
| window | 60 | 82 | 84 | shop | 96 | 95 | 97 | 97 |
| door | 41 | 50 | 60 | balcony | 88 | 87 | 82 | 87 |
| wall | 84 | 96 | 96 | sky | 97 | 97 | 98 | 97 |
| | | | | window | 87 | 85 | 82 | 82 |
| | | | | door | 64 | 63 | 57 | 57 |
| | | | | wall | 77 | 87 | 89 | 88 |
| | | | | vegetation | – | 90 | – | 90 |
| pix. acc. | 78.0 | 91.8 | **92.5** | | 83.5 | 88.4 | **88.8** | **88.8** |

obtained using a random forest classifier. On the other hand the energies used in [5] are the same as in our algorithm. As shown in table 3, our algorithm outperforms the state of the art and yields shorter running times: less than 30 seconds per image compared to 4 minutes for [5]. The increased accuracy can be attributed to a different formulation of the optimization problem, which is solved more efficiently.

**Performance on the ArtDeco dataset** The ArtDeco dataset [2] consists of 80 images of facades of consistent architectural style. The dataset features occlusion of facades by trees and more structural complexity than the ECP or Graz50 datasets. Again, we use TextonBoost to obtain the unary potentials. We use Potts' form of pairwise potentials penalizing transitions between different classes with a fixed coefficient, determined by grid search on a subset of the training set. We test the algorithm in two tasks: extracting the structure of the facades, even when they are occluded, and segmenting the objects visible in the images, including the trees. We evaluate performance of the algorithm in the first task with respect to the original ground truth, which does not contain annotations of vegetation. The accuracy of the segmentations including the trees occluding the facades has been evaluated with respect to the ground truth that we produced by annotating vegetation in all the images. The results are presented in table 3. In this challenging setting our method yields segmentations of higher accuracy than the ones obtained by maximizing the unary potentials.

**Performance on the eTrims dataset** We test our algorithm on the challenging eTrims dataset [4], consisting of 60 images of facades of different styles. We perform a 5-fold cross validation as in [7] and [1], and each time the dataset is divided into 40 training and 20 testing images. We use per-pixel energies generated by a Recursive Neural Network, like in [7] and [1]. We assume the Potts model of pairwise potentials, with the parameter determined by grid search on a subset of the training set. The results are presented in table 4. Our algorithm outperforms the result of [7] and yields result slightly inferior to [1]. The possible reason is the constraints assumed in the latter paper are less restrictive than our grammars. However, our method is still the first algorithm with a user-specified shape grammar to be tested on eTrims and its performance is a close match to the two baseline methods, which offer no flexibility with respect to prior definition.

Table 4. Performance on the eTrims dataset with RNN-based unaries. Starting from left: score using raw unaries, layer 3 of [7], results of [1] and our results.

| | eTrims | | | |
|---|---|---|---|---|
| | raw | [7]-L3 | [1] | Ours |
| building | 88 | 87 | 91 | 92 |
| car | 69 | 69 | 70 | 70 |
| door | 25 | 19 | 18 | 20 |
| pavement | 34 | 34 | 33 | 33 |
| road | 56 | 56 | 57 | 56 |
| sky | 94 | 94 | 97 | 96 |
| vegetation | 89 | 88 | 90 | 91 |
| window | 71 | 79 | 71 | 70 |
| pixel accur. | 81.9 | 81.6 | **83.8** | 83.5 |

## 6. Conclusion

We have shown how complex, grid-structured patterns, possibly with irregular boundaries between regions corresponding to different semantic classes, can be encoded by specifying which pairs of classes can be assigned to pairs of vertically- and horizontally-adjacent pixels. We have argued that these patterns can be specified more conveniently in a hierarchical fashion and shown that the induced flattened set of rules can automatically be translated into the structure of a Markov random field. The formulation lends itself to a more efficient optimization scheme than the previous approaches. Finally, our formulation makes it possible to easily handle occlusion.

**Acknowledgements** We thank Anđelo Martinović from KU Leuven for sharing the texture classification results and Andrea Cohen from ETH Zürich for a useful discussion. This work was carried out in IMAGINE, a joint research project between Ecole des Ponts ParisTech (ENPC) and the Scientific and Technical Centre for Building (CSTB). It was partly supported by ANR project Semapolis ANR-13-CORD-0003.


Figure 5. Parsing results in triples: original image, result of per-pixel classification, parsing result. Each row corresponds to a different dataset. Row labels after hyphen indicate the method used to obtain unary potentials: TB - TextonBoost, RNN - Recursive Neural Network.

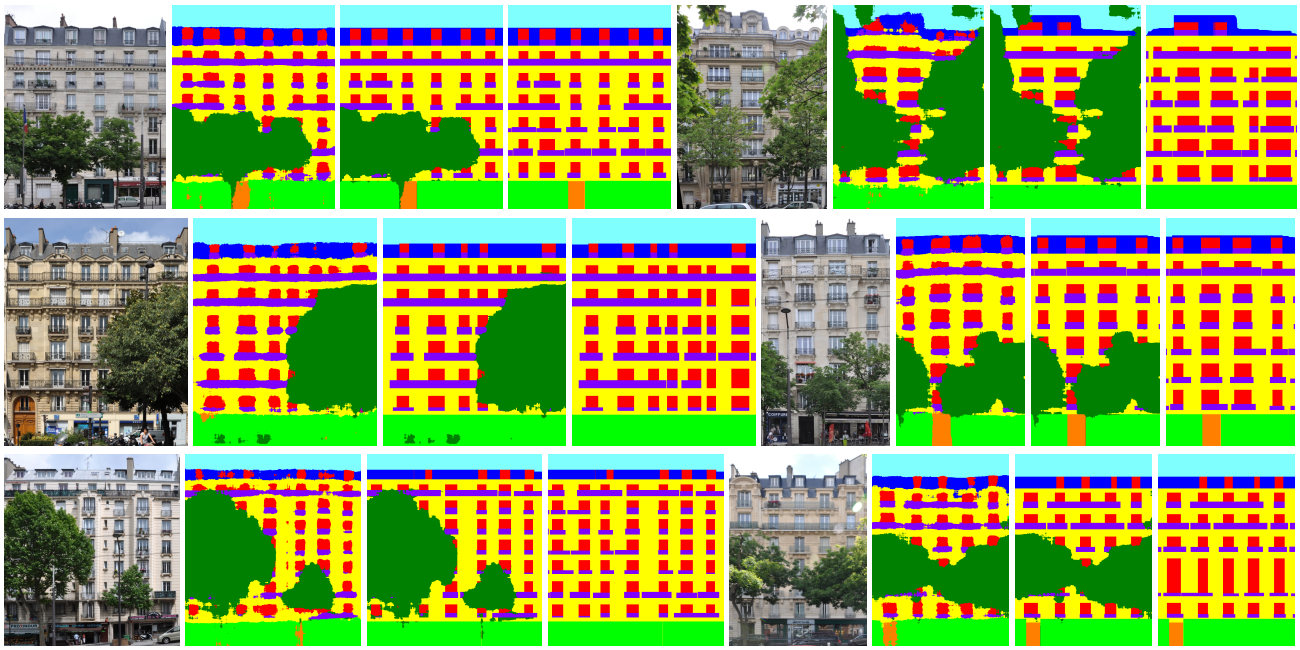


Figure 6. Parsing results for the ArtDeco dataset. In quadruples: original image, unary classification, segmentation with occluder classes, extracted facade structure. The last image is a typical failure case.

## References


[1] A. Cohen, A. Schwing, and M. Pollefeys. Efficient structured parsing of facades using dynamic programming. In *CVPR*, 2014.

[2] R. Gadde, R. Marlet, and P. Nikos. Learning grammars for architecture-specific facade parsing. Research Report RR-8600, Sept. 2014.

[3] N. Komodakis, N. Paragios, and G. Tziritas. Mrf energy minimization and beyond via dual decomposition. *IEEE Trans. PAMI*, 33(3):531–552, 2011.

[4] F. Korč and W. Förstner. eTRIMS Image Database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01, April 2009.

[5] M. Koziński, G. Obozinski, and R. Marlet. Beyond procedural facade parsing: bidirectional alignment via linear programming. In *ACCV*, 2014.

[6] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints):1, 2013.

[7] A. Martinovic, M. Mathias, J. Weissenberg, and L. Van Gool. A three-layered approach to facade parsing. In *ECCV 2012*. Springer, 2012.

[8] D. Ok, M. Kozinski, R. Marlet, and N. Paragios. High-level bottom-up cues for top-down parsing of facade images. In *2nd Joint 3DIM/3DPVT Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, 2012.

[9] H. Riemenschneider, U. Krispel, W. Thaller, M. Donoser, S. Havemann, D. Fellner, and H. Bischof. Irregular lattices for complex shape grammar facade parsing. In *CVPR*, 2012.

[10] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV (1)*, pages 1–15, 2006.

[11] L. Simon, O. Teboul, P. Koutsourakis, L. Van Gool, and N. Paragios. Parameter-free/pareto-driven procedural 3d reconstruction of buildings from ground-level sequences. In *CVPR*, 2012.

[12] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *ICML)*, 2011.

[13] D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.

[14] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios. Shape grammar parsing via reinforcement learning. In *CVPR*, pages 2273–2280, 2011.

[15] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape priors. In *CVPR*, pages 3105–3112, 2010.

[16] T. Werner. A linear programming approach to max-sum problem: A review. *Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1165–1179, July 2007.