This is a preprint version of the article published in International Journal of Imaging System and Technology, volume 19, issue 2, pp. 66-99, 2009. Wiley Periodicals, Inc. DOI 10.1002/ima.20181

Languages for Constrained Binary Segmentation based on Maximum Aposteriori Probability Labeling

Jan Čech and Radim Šára
Center for Machine Perception
Faculty of Electrical Engineering, Czech Technical University
Prague, Czech Republic
{cechj,sara}@cmp.felk.cvut.cz

ABSTRACT: We use a MRF with asymmetric pairwise compatibility constraints between direct pixel neighbors to solve a constrained binary image segmentation task. The model is constraining shape and alignment of individual contiguous binary segments by introducing auxiliary labels and their pairwise interactions. Such representation is not necessarily unique. We study several ad-hoc labeling models for binary images consisting of non-overlapping rectangular contiguous regions. Nesting and equivalence of these models are studied. We observed a noticeable increase in performance even in cases when the differences between the models were seemingly insignificant.

We use the proposed models for segmentation of windowpanes and windows in orthographically rectified façade images. Segmented window patches are always axis-parallel non-overlapping rectangles which must also be aligned in our strongest model. We show experimentally that even very weak data model in the MAP formulation of the optimal segmentation problem gives very good segmentation results.

1 Introduction

Markov Random Fields (MRFs) have been used in image analysis for a long time (Woods, 1972; Geman & Geman, 1984). There are many papers on image segmentation using MRFs, e.g. (Wilson & Li, 2002; Kato & Pong, 2006). The spatial relationships of pixels in the image domain is often modeled by the *Potts model* (Baxter, 1990). The model prescribes a zero penalty for adjacent pixel having the same label and a constant penalty if the adjacent pixels have different labels. This prior model reflects a natural assumption on the segmentation to be locally homogeneous. Homogeneous patches have higher probability to become a part of the solution.

On the other hand, the Potts model cannot incorporate any stronger assumption on the *shape* of the patches, i.e. on the structure of the segmentation. In this paper, we make an explicit requirement for the shape of the patches to be segmented. Moreover, we are able to model some distant relations among segmented patches, namely their *alignment*. This is done by using auxiliary labels and by introducing a prior model of more complicated structure, in which pairwise probabilities between labels are typically asymmetric. The resulting MRF is thus non-Gibbsian. Similar structure prior models appeared in (Werner, 2005; Werner, 2007) where they demonstrate the functionality of their labeling solver.

We will apply the proposed models on the problem of segmentation of windowpanes or windows in façade images. Window arrays in orthographically rectified façade images are almost always a set of non-overlapping axis-parallel rectangles, and the windows are often fully aligned in rows and columns.

The obvious drawback of the proposed approach compared to similar segmentation with Potts model is that it leads to NP-hard problem. However, we will show that an approximate algorithm will give acceptable results and the segmentation quality of the proposed method is superior to Potts model.

The presented structure model constrains a configuration of labels and describes a language, i.e. a set of all configurations (words) which are allowed. Unlike languages described by a grammar (Zhu & Mumford, 2006; Schlesinger & Hlaváč, 2002; Průša & Hlaváč, 2007) this model is not generative. On the other hand, image segmentation in both labeling and grammatical formulations is a search for the most probable word in a language given (a noisy) observation, looking for the maximum aposteriori probability solution.

Of course, there are several different methods to detect windows in façade images. For instance, in (Mayer & Reznik, 2005; Dick *et al.*, 2004) they use a *parametric* model of windows. Changing the parameters (as width, aspect ratio, brightness, etc.) using Markov Chain Monte Carlo sampling they try to generate the image which is the most similar to the given image. In (Alegre & Dellaert, 2004), they use stochastic context-free grammars to represent a hierarchical regular structure of a façade. These approaches are very different from our simple formulation based on segmentation.

The rest of the paper is structured as follows: The proposed strong structure models are described in Sec. 2, where we also bring some theoretical results concerning equivalence and partial ordering of various labeling languages with respect to the inclusion relation. In Sec. 3 we give some details on the implementation of window segmentation. Experimental validation on both synthetic data and images of real façades is given in Sec. 4. Sec. 5 concludes the paper.

2 Strong Structure Models

The problem of segmentation under strong structural constraints is formulated in maximum aposteriori probability

sense:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in X^{|T|}} p(\mathbf{x} \mid \mathbf{I}), \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_{|T|})$ is a labeling (interpretation), and observation $\mathbf{I}$ is the input image. The T is a finite set of all pixels, i.e. locations, in the image. The X is a finite set of labels assignable to a pixel. We denote I_t the image intensity at pixel $t \in T$, and x_t a label $x \in X$ of pixel $t \in T$.

Using the Bayes law, the problem (1) is equivalent to

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in X^{|T|}} p(\mathbf{I} \mid \mathbf{x})p(\mathbf{x}), \tag{2}$$

where $p(\mathbf{I} \mid \mathbf{x})$ is the data term reflecting agreement of observation with an interpretation $\mathbf{x}$, and $p(\mathbf{x})$ the prior term reflecting the prior probability of $\mathbf{x}$.

Assuming independence, we can rewrite the task as

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in X^{|T|}} p(\mathbf{x}) \prod_t p(I_t \mid x_t). \tag{3}$$

The prior term $p(\mathbf{x})$ can be used to constrain the solution. We call this probability a *structure model*. Certain labelings are forbidden, i.e. they have zero probability $p(\mathbf{x}) = 0$. We call a *language* L a set of all allowed labelings such that

$$L = \{\mathbf{x} \in X^{|T|} : p(\mathbf{x}) > 0\} \tag{4}$$

Following a standard terminology, a given labeling $\mathbf{x}$ over the set of pixels T is called a 'word.'

2.1 Models Based on Consistent Labeling

We model the structural term $p(\mathbf{x})$ in (3) as a product of all pairwise pixel probabilities

$$p(\mathbf{x}) = \prod_{t,t'} p_{t,t'}(x_t, x_{t'}), \tag{5}$$

where t, t' is a pair of immediate 4-neighbors in $T \times T$. This gives us a Markov Random Field in (3). The $p_{t,t'}(x_t, x_{t'})$ is a probability of adjacent label co-occurrence. It represents the rules that generate a 2D language. In this paper, the $p_{t,t'}(x_t, x_{t'})$ is written as $p(x_t, x'_t)$ to shorten our notation.

Applying the logarithm to (3) and using (5) we get

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in X^{|T|}} \sum_{t,t'} g(x_t, x_{t'}) + \sum_t g(I_t \mid x_t), \tag{6}$$

where g stands for a logarithm of probability p. This is a max-sum labeling problem.

An illustration of the labeling problem is shown in Fig. 1. There is an image grid T. Pixels $t \in T$ are sketched as squares. Each of them contains a finite set of labels $x \in X$ creating nodes of an underlying graph. The labels between adjacent pixels are interconnected via edges. Each node is assigned a node quality $g(I_t \mid x_t)$, which reflects an agreement of the observation with the label. It is a part of the data term. Each edge is assigned an edge quality $g(x_t, x_{t'})$, which corresponds to pairwise label probabilities from (5). It is a part of the structure prior term of the language. The task of the max-sum labeling (6) is to select a single label per pixel to maximize the sum of node qualities and corresponding edge qualities over the entire image. A set of maximizing nodes and the corresponding edges are shown in red. Note that the graph of our problem is much larger than in Fig. 1, in the number of pixels (our images are up to 0.7Mpx) and the number of labels (we have up to 10 labels).

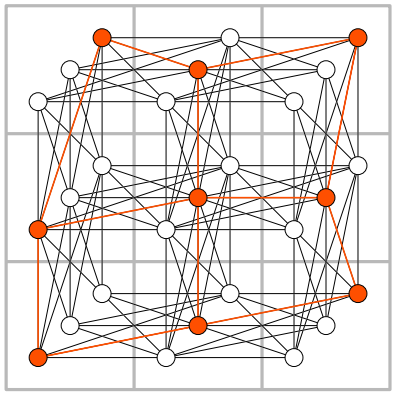


Figure 1: Illustration of the labeling problem on a 3×3 image. Squares represent image pixels, graph nodes correspond to labels (we have 3 labels per pixel in this example), and graph edges represent pairwise label compatibilities.

The problem (6) is NP-hard in general. There exist solvable sub-classes (Flach & Schlesinger, 2000), e.g. when the number of labels is two, or when the underlying graph does not contain cycles, or when $g(x_t, x'_t)$ is submodular (Kolmogorov & Zabih, 2004). Our task does not belong to any of the known solvable sub-classes and probably remains NP-complete. A brute-force approach to the problem is of complexity $\mathcal{O}(|X|^{|T|})$. However, there exist approximate algorithms which find a sub-optimal solution, e.g. (loopy) belief propagation (Pearl, 1988; Felzenszwalb & Huttenlocher, 2006), the TRW-S algorithm (Kolmogorov, 2006), max-sum diffusion (Kovalevsky & Koval, 1975; Flach, 1998), linear programming relaxation (Schlesinger, 1989; Werner, 2005; Werner, 2007), a recent method via dual decomposition (Komodakis *et al.*, 2007), and others.

We will distinguish two types of pairwise probabilities: horizontal $p_h(x, x')$ where x' is the right image neighbor of x, and vertical $p_v(x, x')$ where x' is the image neighbor below x. Probabilities $p_h(x, x')$, $p_v(x, x')$ are *stationary* (the same for all locations in the image), therefore the subscript t in x_t is omitted. We say neighboring labels x, x' satisfy a *structural constraint* if $p(x, x') > 0$. We will consider two bipartite graphs: horizontal compatibility graph G_h and vertical compatibility graph G_v. Edges $e(x, x')$ in these graphs represent compatibility (constraint satisfaction) and missing edges constraint violations (cases when $p(x, x') = 0$).

Fig. 3 shows a simple example of a structural model $p(\mathbf{x})$. Consider a language that contains only binary images of single pixel-wide horizontal stripes where a black stripe alternates with a white stripe. This language is modeled with two labels B (black) and W (white) and pairwise compatibility rules between neighboring pixel labels: Right to the pixel B, only another B is allowed, right to the pixel W, only another W is allowed (there is no change in the horizontal direction); down from the pixel B, it must be W, and down from W it must

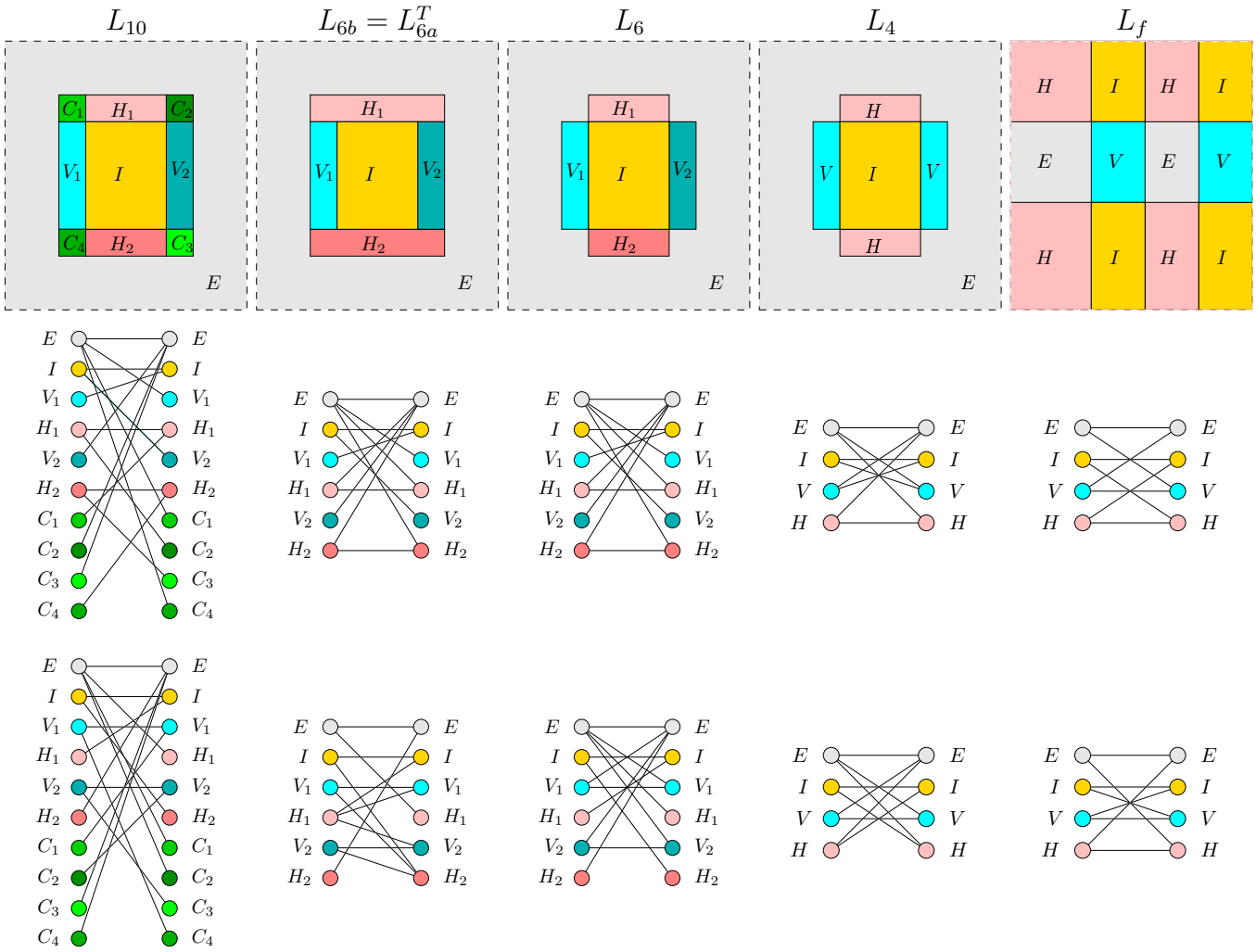


Figure 2: Various languages for axis-parallel non-overlapping rectangles induced by different labeling constraints. Top row shows all labels in an allowed configuration. The middle and bottom rows show horizontal and vertical pairwise compatibility graphs G_h, G_v, respectively.

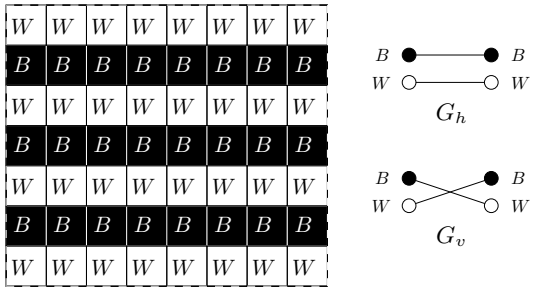


Figure 3: An example of the labeling model for image structure. Labeled image (left) and its horizontal pairwise compatibilities G_h and vertical pairwise compatibilities G_v (right).

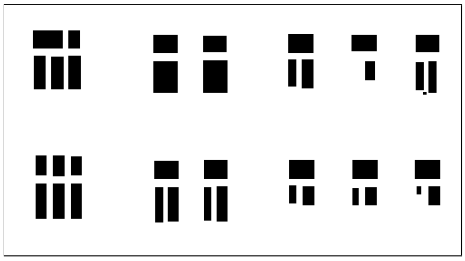


Figure 4: A word from the language of axis-parallel non-overlapping rectangles. (The boundary frame is not a part of the word.)

be B (labels must alternate in the vertical direction). These rules are expressed in pairwise compatibility graphs shown in Fig. 3.

In this paper we will be interested in binary image segmentations in which objects of interest are rectangular patches aligned with coordinate axes. No two patches are closer than one pixel. An example of a word from this language is shown in Fig. 4.

In Fig. 2 we define several representations for this class of languages by choosing pairwise compatibilities $e(x, x')$. Note that typically $e(x, x') \neq e(x', x)$, and in all cases $G_h \neq G_v$. It is easy to show that the only allowed words in languages L_{10}, L_{6b}, L_{6a}, L_6, L_4, L_f form a set of axis-parallel non-overlapping rectangles. Language L_{6a} is a transposed version of language L_{6b} with swapped horizontal and vertical pairwise compatibility graphs. The rectangles to be segmented are labeled I. Let there be an allowed configuration in which the I-region is not rectangular. In this case, any of the configuration in Fig. 8(a) with a label $x_i \neq I$ must be allowed. However, by the compatibilities of any of the presented languages, the only allowed labels are $x_i = I$, which is a contradiction.

In (Čech & Šára, 2008) we have used model L_{10} for constrained segmentation of windowpanes of orthographically rectified façade images. In subsequent sections we analyze relations among the languages and show that the strength of the model (inversely proportional to the size of the language) plays a key role in determining the quality of the segmentation results.

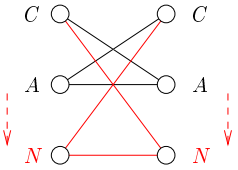


Figure 5: Equivalent expansion by label N from label A. The new label N and new compatibility edges are shown in red.

2.2 Equivalent and Nested Languages and the Ambiguity of Inference

In this section we study languages which are generated by different labeling formulations, i.e. different labels and/or different pairwise compatibilities. Our goal is to find the inclusion relations among the set of languages listed in Fig. 2.

A word is a labeled image, where the labels come from a label set X. We assume a language L is represented by either the set of all words or equivalently by the pair (X, E), where X is the label set and E is the set of pairwise label compatibilities. We then write $L = (X, E)$. As above, the pair (X, E) is represented as a pair of bipartite compatibility graphs G_h (horizontal) and G_v (vertical). Elements of X correspond to graph nodes and elements of E correspond to graph edges, as illustrated in Fig. 3.

We say languages $L_1 = (X, E_1)$ and $L_2 = (X, E_2)$ over the same label set X are *nested*, $L_1 \subseteq L_2$, if the set of all words from L_1 is a subset of the set of all words from L_2. We say languages L_1 and L_2 are *equivalent*, $L_1 = L_2$, if $L_1 \subseteq L_2$ and $L_2 \subseteq L_1$.

Lemma 2.1. *Let $L_1 = (X, E_1)$ and $L_2 = (X, E_2)$ be two languages such that $E_1 \subseteq E_2$. Then $L_1 \subseteq L_2$.*

Proof. Every word $w \in L_1$ belongs to L_2 since it satisfies its constraints. However L_2 can be larger since it has additional compatibilities from $E_2 \setminus E_1$. Therefore $L_1 \subseteq L_2$. □

Let X be a set of labels. A *binarization* is a mapping $B\colon X \to \{0, 1\}$. The union of all binarized words from L will be called a binarization of language L and denoted as $B(L)$. Loosely speaking, a binarization of a language flattens it into a language containing only binary words.

Given B_1, B_2, we say languages L_1 and L_2 are *B-nested* if $B_1(L_1) \subseteq B_2(L_2)$, i.e. the set of all binary words from L_1 is a subset of the set of all binary words from L_2. When $B_1 = B_2 = B$ we also write $L_1 \overset{\text{B}}{\subseteq} L_2$. *B-equivalence* is then defined in the same way as above, $B_1(L_1) = B_2(L_2)$ if $B_1(L_1) \subseteq B_2(L_2)$ and $B_2(L_2) \subseteq B_1(L_1)$. Clearly, nesting implies B-nesting: If $L_1 \subseteq L_2$ then $L_1 \overset{\text{B}}{\subseteq} L_2$.

Given a language $L = (X, E)$ we say language $\tilde{L} = (X \cup \{N\}, \tilde{E})$ is obtained by *equivalent expansion* from L by replicating label $A \in X$ to a new label $N \notin X$ if pairwise compatibilities in $\tilde{E}$ are $\tilde{e}(N, N) = e(A, A)$ and $\tilde{e}(N, x) = e(A, x)$, $\tilde{e}(x, N) = e(x, A)$ for all $x \in X \setminus \{A\}$. Equivalent expansion is illustrated in Fig. 5. The intuition is that the new label copies pairwise compatibility edges from the label it is expanded from.

Lemma 2.2. *Let language L_2 be obtained by equivalent expansion from language $L_1 = (X, E)$ by replicating its label*

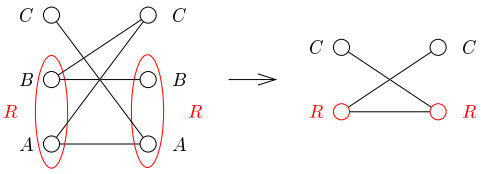


Figure 6: Reduction by unification of labels A and B to label R.

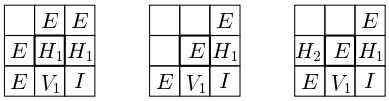


Figure 7: Illustrations for the proof that $L_{10} \overset{\text{B}}{=} L_{6b}$. The central pixel is t' in (a) and (b) and t in (c).

$A \in X$ to a new label $N \notin X$. Let B_1 be the binarization of L_1 and B_2 the binarization of L_2 such that $B_2(x) = B_1(x)$ for all $x \in X$ and $B_2(N) = B_1(A)$. Then L_1 and L_2 are B-equivalent, $B_1(L_1) = B_2(L_2)$.

Proof. The new label N does not bring any new constraints, since all its pairwise compatibilities are identically copied from pairwise compatibilities of the existing label A. This means the roles of N and A are identical, i.e. the labels are freely interchangeable. Since they have the same binarization, the set of binary words is always equal. □

Given two languages $L = (X \cup \{A, B\}, E)$ and $\tilde{L} = (X \cup \{R\}, \tilde{E})$, we say $\tilde{L}$ is a *reduction of L by unification* of labels A, B to label R if pairwise compatibilities in $\tilde{E}$ are such that $\tilde{e}(R, R)$ is present whenever $e(A, A)$ or $e(A, B)$ or $e(B, A)$ or $e(B, B)$ is present, which we write as $\tilde{e}(R, R) = e(A, A) \vee e(A, B) \vee e(B, A) \vee e(B, B)$, and if $\tilde{e}(R, x) = e(A, x) \vee e(B, x)$, $\tilde{e}(x, R) = e(x, A) \vee e(x, B)$ for all $x \in X \setminus \{A, B\}$. Reduction by unification is illustrated in Fig. 6. The intuition is that the labels are unified together with all edges of the bipartite graph and all parallel edges are merged.

Theorem 2.3. *Let $L_1 = (X \cup \{A, B\}, E_1)$ and $L_2 = (X \cup \{R\}, E_2)$ be two languages such that L_2 is a reduction of L_1 by unification of labels A and B to label R. Let B_1 be a binarization of L_1 and B_2 a binarization of L_2 such that $B_1(A) = B_1(B) = B_2(R)$ and $B_1(x) = B_2(x)$ for all $x \in X$. Then L_1 is B-nested in L_2, $B_1(L_1) \subseteq B_2(L_2)$.*

Proof. Let L_2 be as required. Let L_3 be obtained by equivalent expansion of L_2 by replicating label R to label N. To get the same label set as in L_1 we rename labels of L_3 by $R \mapsto A$, $N \mapsto B$. Then L_3 is B-equivalent to L_2 by Lemma 2.2, i.e. $B_1(L_3) = B_2(L_2)$. By the construction of equivalent expansion, we see that $E_1 \subseteq E_3$. Lemma 2.1 implies $L_1 \subseteq L_3$, hence $B_1(L_1) \subseteq B_1(L_3)$. It follows $B_1(L_1) \subseteq B_2(L_2)$. □

The above theory helps find the structure of the set of languages of non-overlapping rectangles L_{10}, L_{6a}, L_{6b}, L_6, L_4 shown in Fig. 2. We binarize them as $I \to 1$ and $X \setminus \{I\} \to 0$. We have already shown the only allowed words are members of the set of non-overlapping rectangles. The only small differences are in separations which are allowed between two neighboring rectangles, for instance L_4 allows 1 px separation while $L_{10}, L_{6a}, L_{6b}, L_6$ allow 2 px separation only.

By applying Theorem 2.3 we can show that $L_{10} \overset{\text{B}}{\subseteq} L_{6b}$ by successively unifying labels C_1, C_2 to H_1, and C_3, C_4 to H_2. Similarly, we can show $L_{10} \overset{\text{B}}{\subseteq} L_{6a}$, $L_{10} \overset{\text{B}}{\subseteq} L_6$, and $L_6 \overset{\text{B}}{\subseteq} L_4$. We now show that L_{10} and L_{6b} are B-equivalent.

Theorem 2.4. *Let $L_{10} = (X_{10}, E_{10})$, $L_{6b} = (X_{6b}, E_{6b})$. Let B binarize $I \to 1$ and $(X_{10} \cup X_{6b}) \setminus \{I\} \to 0$. Then $B(L_{10}) = B(L_{6b})$.*

Proof. Let us construct a language $\tilde{L} = (\tilde{X}, \tilde{E})$ which is (a super-set[1] of the) union of these languages, by unifying the label sets $\tilde{X} = X_{10} \cup X_{6b}$ and the edges $\tilde{E} = E_{10} \cup E_{6b}$. We say label $x \in \tilde{X}$ is *translatable at location t* to another label $y \in \tilde{X}$ if every word $w \in \tilde{L}$ in which x occurs at location t can be replaced by another word $w' \in \tilde{L}$ in which location t holds label y. Moreover, if $B(x) = B(y)$, we say the label x is *B-translatable* at t. Every translation of a word w that occurs at all translatable labels of w is called *equivalent* and every such translation that preserves binarization is *B-equivalent*.

We show that every word $w \in \tilde{L}$ is either in L_{10} or there is a B-equivalent translation that transforms w to another word $w' \in L_{10}$.

Let $w \in \tilde{L}$, $w \notin L_{10}$. Every such word must engage some compatibility edge $e(x, x')$ in $E_{6b} \setminus E_{10}$. By inspecting Fig. 2, we see these edges are (E, H_1), (E, H_2), (H_1, E), (H_2, E) in $G_h(L_{6b})$, and (V_1, H_2), (V_2, H_2), (H_1, V_1), (H_1, V_2) in $G_v(L_{6b})$.

Consider neighboring locations (t, t') in $w \in \tilde{L}$ with labels $x_t = E$, $x_{t'} = H_1$, which we will write as a label pair (E, H_1). We show that the label H_1 at t' is B-translatable to C_1. Consider the neighborhood of t' in w. Labels of this neighborhood are uniquely restricted in $\tilde{L}$ by labeling of t, t' to form a local configuration, in which t' is the central pixel, as shown in Fig. 7a. In other words, every $w \in \tilde{L}$ which is labeled (E, H_1) at (t, t') must have neighboring-location labels listed in Fig. 7a. If we now replace label H_1 with C_1 at t' in every such word the new word will still be in $\tilde{L}$ but will not engage consistency edge $\tilde{e}(E, H_1)$ at location (t, t'). Similar transformations are possible in all other edges from $E_{6b} \setminus E_{10}$. As a result, for every word $w \in \tilde{L}$ there is a translation $w' \in L_{10}$. Since H_1 and C_1 have the same binarization, the translation is B-equivalent.

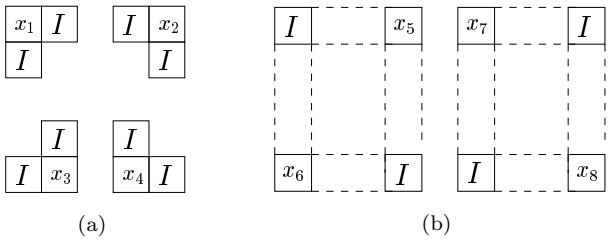
Hence, $B(\tilde{L}) \subseteq B(L_{10})$. We know that $L_{10} \subseteq \tilde{L}$, hence $B(L_{10}) = B(\tilde{L})$. By Theorem 2.3 we know that $B(L_{10}) \subseteq B(L_{6b})$. From the above and from $L_{6b} \subseteq \tilde{L}$ we have $B(L_{10}) \subseteq B(L_{6b}) \subseteq B(L_{10})$. It follows $B(L_{10}) = B(L_{6b})$. □

The B-equivalence of L_{6a} and L_{10} can be proved in the same way. Note an attempt to "prove" $B(L_{10}) = B(L_6)$ fails: Consider the edge (E, H_1) in $E_h(L_6) \setminus E_h(L_{10})$ (E_h are compatibility edges from G_h). The restriction of neighboring labels around t' becomes one as shown in Fig. 7b. But the label E at t is not translatable to C_1 since there are words in $\tilde{L}$ that are inconsistent with this label, e.g. words including local configuration shown in Fig. 7c.

To summarize, for languages introduced in Fig. 2 it holds:

$$L_{10} \overset{\text{B}}{=} L_{6b} \overset{\text{B}}{=} L_{6a} \overset{\text{B}}{\subseteq} L_6 \overset{\text{B}}{\subseteq} L_4\,. \tag{7}$$

[1] It is a super-set because we are unifying the language representations rather than the sets of words themselves.



(a) (b)

Figure 8: A Configuration of I-label discussed in the proof of validity of languages.

Figure 9: Annotation used to learn language models. Red rectangles (windowpanes) are used to learn probabilities for L_{10}, L_{6a}, L_{6b}, L_6, L_4 and dashed blue rectangles (windows) are used to learn L_f.

This result also demonstrates that the problem of inferring structural constraints from a training set of binary images need not have a single solution (consider L_{6a} and L_{6b}). As we shall see in a controlled performance study in Sec. 4, in case we decide to regularize language description inference from training data, minimum-size label set need not be the best choice if we are concerned with the speed of image parsing.

2.3 A More Constrained Language

So far, the structure model constrained the shape of the segmented region only. There was no model for region alignment, except for the non-overlap requirement. Therefore we asked whether it is possible to introduce such higher-level constraints and implement some long-distance relations using local compatibilities between pixels.

The answer is positive. We formulate the model for aligned rectangles, i.e. rectangles which are aligned both horizontally and vertically. Such model reflects a situation where the windows in a façade are aligned in a rectangular array. The spacings between the rows and columns of the array are left unconstrained.

The language for such model is shown in Fig. 2 as L_f. It uses four labels again. The language is binarized by $\{I\} \to 1$, $\{H, E, V\} \to 0$. Again, we can easily prove using the compatibility rules that the only allowed segments are rectangles, by showing that labels $x_{1..4}$ in Fig. 8(a) must be labeled I. In addition, we have to prove that all rectangles are both horizontally and vertically aligned. Let us assume the rectangles are not aligned. Then, any of the two configurations with labels $x_{5..8} \neq I$ in Fig. 8(b) representing misaligned configuration must occur. However, by looking at the rules of L_f, the only allowed configuration is $x_{5..8} = I$, which is a contradiction.

An interesting observation is that language L_4 representing non-overlapping rectangles only and language L_f representing aligned rectangles have the same number of labels. These languages differ in pairwise compatibilities only.

In Sec. 4 we demonstrate the performance of all the models in both controlled synthetic-data experiment with ground-truth and on real images.

3 Implementation

We applied the languages to the task of windowpane or window segmentation in orthographically rectified façade images. Language L_f of aligned rectangles is intended to segment windows, the other languages are supposed to segment individual windowpanes.

We use a very simple image model $p(\mathbf{I} \mid x)$. For each label separately, we learn a probability distribution of the pixel color $\mathbf{I} = (r, g, b)$ as red, green, and blue intensity channels

$$p(\mathbf{I} \mid x) = p(r, g, b \mid x), \tag{8}$$

where the $p(r, g, b \mid x)$ is assumed to be of Gaussian distribution. The mean value vector and covariance matrix Σ_I are estimated from annotated training images. We used a very modest set of training data. We manually annotated all windowpanes in the image shown in Fig. 9. This served as training data for the label I. The complement to the windowpanes in the image (a façade) was the training data for all the remaining labels. In fact, this model is rather naive since the color itself is not really a stable feature. Moreover, windowpanes are either dark or glossy, reflecting the color of the sky. The unimodal distribution does not fit well. The color of the façade is clearly not of a Gaussian distribution, moreover it varies among different buildings. However, as can be seen from our real image experiments, the method performs quite well with such a simple model. We hypothesize this is due to a strong structure modeling.

Probabilities $p(x, x')$ are estimated from labeled examples, as a relative frequency of individual co-occurrences $(x, x') \in X \times X$ in all neighbor pairs appearing in the labeled examples.

We kept both the image and the structure model parameters fixed throughout all the real image experiments.

As mentioned above, we used a general labeling solver to optimize (6) in all languages and in the Potts model.[2] We selected Werner's implementation (Werner, 2005; Werner, 2007) of linear programming relaxation-based max-sum solver originally proposed by Schlesinger (Schlesinger, 1989). It seems to give good results for our problem, unlike belief propagation (Pearl, 1988; Felzenszwalb & Huttenlocher, 2006) which often oscillates and max-sum diffusion (Flach, 1998; Kovalevsky & Koval, 1975) which is very slow.

4 Experiments

Synthetic-Data Experiment We use a synthetic gray-scale test image simulating a 'façade', see Fig. 10. It is of 100×

[2] A faster solution of the 2-labeling problem can be obtained by Max-Flow algorithm, which is polynomial and optimal for this problem (Boykov & Kolmogorov, 2004).

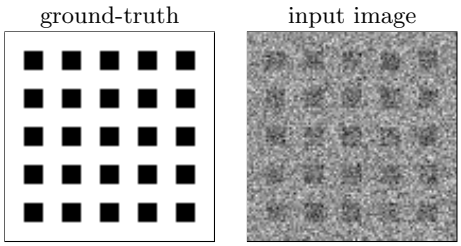


Figure 10: Synthetic image of unit contrast (ground-truth) and the image contaminated with additive Gaussian noise of $\sigma = 0.8$.

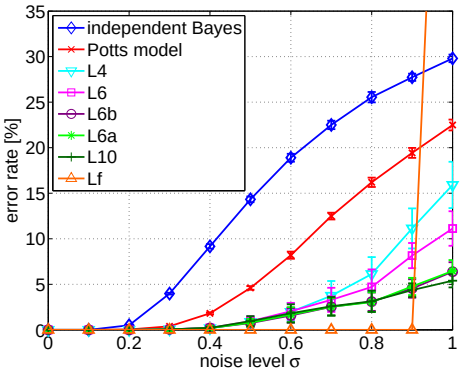


Figure 11: Labeling error rate as a function of noise level.

100 pixels, containing 25 dark rectangles of intensity $\mu_0 = 0$ in the light background of intensity $\mu_1 = 1$. We added i.i.d. Gaussian noise with increasing standard deviation σ to the image. Thus the statistical image model of the rectangles and background segments is $N(\mu_0, \sigma^2)$, $N(\mu_1, \sigma^2)$ respectively.

In a repeated experiment, we measured an error rate, i.e. percentage of pixels which were labeled incorrectly (mislabeled or undecided), as a function of noise level σ. We compare the performance of the MRF segmentation based on the proposed models of aligned rectangles L_f, free rectangles $L_{10}, L_{6a}, L_{6b}, L_6, L_4$, and homogeneous patches using Potts model, and the local Bayesian classifier

$$\forall t \in T,\ x_t^* = \arg\max_{\mathbf{x}_t \in X} p(I_t \mid x_t) p(x_t), \tag{9}$$

where $p(x)$ is a prior probability of the label. Thus the Bayesian decision is performed in each pixel independently (we call the method *Independent Bayes*).

The prior of the Bayesian classifier was estimated from ground-truth labeling. The parameters of the image models were set to $\mu_0 = 0$, $\mu_1 = 1$, $\sigma_I = 0.2$ and kept fixed throughout the experiment.

The results shown in Fig. 11 are averaged from 10 random trials with one-sigma error-bars. The local decision (Independent Bayes) has worse performance than any method modeling pixel neighborhood relations. The 2-label Potts model MRF segmentation outperforms the local method, which is a well known fact in segmentation literature. All shape modeling languages $L_{10}, L_{6a}, L_{6b}, L_6, L_4$ perform even better. Their modeling power, allowing only axis-parallel non-overlapping rectangles has a large impact on the result. The language L_f of aligned rectangles produces consistently error-free segmentation until the noise level of $\sigma = 1$ (where it does not decide at all). This model is much stronger, i.e. it is more informed than the others, therefore it performs the best. However, when the data does not support the model well, as in the case of very strong noise, the algorithm fails. See also the (binarized) segmentation maps for the noise level $\sigma = 0.8$ in Fig. 12.

We can also see there are small differences in performance among languages of free rectangles. Languages L_{10}, L_{6b} and L_{6a} are consistently the best, L_6 is worse and L_4 has the worst performance. The performance of the languages is decreasing with increasing language size, in the same order as the B-nesting relation (7). Although differences between the individual languages seem small, they differ significantly in experimental performance, especially at higher noise levels. Interestingly, languages of fewer labels do not necessarily perform better.

Language L_{6b} and its transposed version L_{6a} have identical performance up to (most likely) numerical accuracy. At the highest noise level a small difference between L_{10} and $L_{6b} = L_{6a}^T$ shows up. This is probably caused by the fact that the corner labels of L_{10} have additional corner-specific pairwise probabilities, e.g. $p_h(C_1, H_1)$, $p_v(C_1, V_1)$ in the upper-left corner, which is not the case of L_{6b}, where it has $p_h(H_1, H_1)$, $p_v(H_1, V_1)$ only. The $p_h(H_1, H_1)$ is not unique for the upper-left corner since it must play a role in the upper edge, too. In other words, despite languages L_{10} and $L_{6b} = L_{6a}^T$ model the same set of binary words, the probabilities of their identical binary words differ.

Besides the error rate, we also computed the percentage of undecided pixels, i.e. pixels where the approximate solver did not find a unique labeling, see Fig. 13. There are differences between the languages. We see the solver tends to produce more pixels with ambiguous labeling for languages with fewer labels. It produces significantly fewer undecided pixels in language L_{10} than in L_{6b} or L_{6a}, although they are all B-equivalent. Additional finer pairwise probabilities of L_{10} around corners probably cause a better 'shape' of the optimum of the discrete optimization task. This is an interesting property, since this means languages with fewer labels need not be the best choice for optimal performance.

We also measured CPU time spent by the solver as a function of noise level, see Fig. 14. The results are shown for the same data as above. The mixed Matlab/C implementation was run on a standard PC with a C2 2.4 GHz CPU. We can see that languages $L_{10}, L_6, L_{6b}, L_{6a}, L_4$ have similar time complexity. The L_4 is slightly faster for lower noise levels, while it is slower for higher noise levels. The language of aligned rectangles L_f is computationally most intensive, moreover, we see the CPU time grows rapidly with noise level. This is probably a manifestation of NP-hardness of the task.

Real Data We run the proposed method on several images of real façades. The images were from 0.1 to 0.7 Mpx in size. Both image model and structure model were learnt from a single façade image with manually annotated windowpanes, as described in Sec. 3.

A comparison of results for the proposed models L_f (aligned rectangles) and L_{10} (free rectangles), Potts model

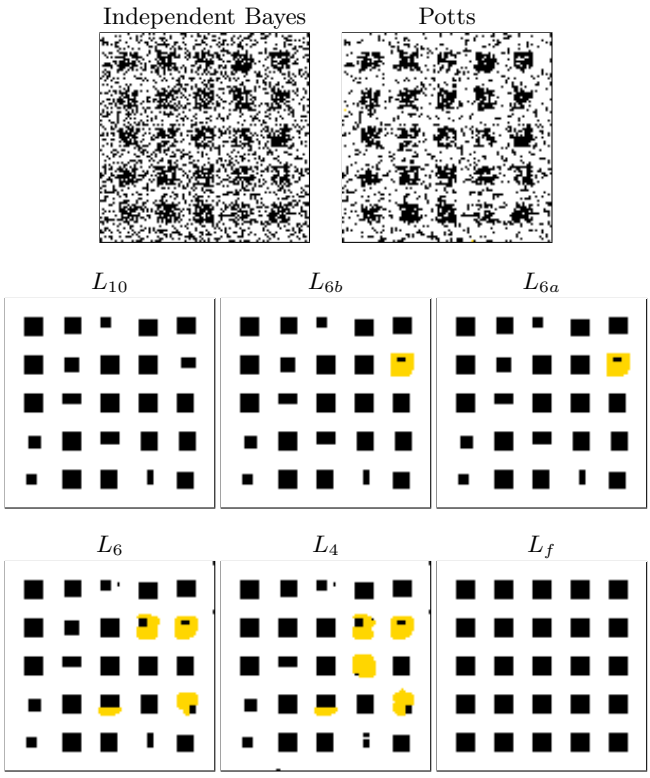


Figure 12: A typical result of the labeling for $\sigma = 0.8$ shown as segmentation maps. Labels belonging to a rectangle class are shown in black, labels of the background class in white. Gold color represents undecided regions, i.e. pixels where the solver did not find a unique labeling. The input image is shown in Fig. 10.

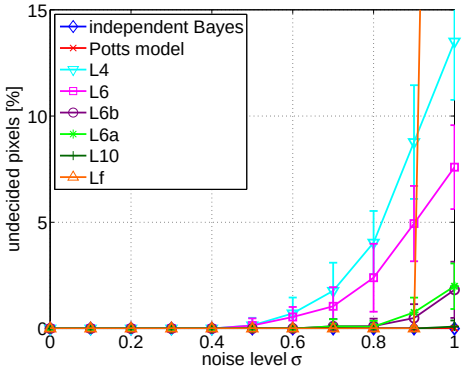


Figure 13: Percentage of undecided pixels as a function of noise level.

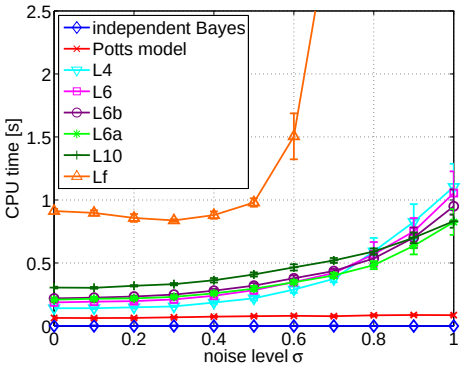


Figure 14: CPU time as a function of noise level.

(weak structure model) and the Independent Bayes model (purely local method) is shown in Fig. 16. We can see the strongest model L_f correctly finds all windows without any false positive detections. Model L_{10} forces the windowpanes to be rectangles and helps reject regions of the same appearance (pixel color) as true windowpanes, but violating the structure, e.g. railings, tree branches, shadows. The rejection cannot occur in the Potts model or in the local method.

Fig. 15 shows results of L_f and L_{10} overlaid over input images (top) and as color-coded labeling maps (bottom). We can see that for L_f we get a good window detection, the only false positives occur on pipes that fit the model both in appearance and structure. The detection is much worse in L_{10}: We see several false positives and false negatives. This also happens in other results, shown in Fig. 17. This again confirms that the stronger model produces higher-quality segmentations.

The reason for the observed segmentation errors is two-fold: (1) the actual image and structure model are locally violated, or (2) the solution we obtained from approximate algorithm is not the global optimum.

The CPU time needed to solve the task for the Potts model was a fraction of a second. The CPU time for the L_{10} model is less than 10 seconds per image. The timing for the L_f model ranges from about one minute to almost two hours for the image in Fig. 16. The runtime does not depend too much on the size of the image, rather it depends on scene complexity. For instance, the windows in Fig. 16 are not perfectly aligned because of a three-dimensional structure of the façade. This makes L_f an invalid model for this image. As a result, it takes too long for the solver to find a local optimum of (6). Even worse, the solution of L_f for the last image in Fig. 17 was not found after 7 millions iterations having taken several hours. In that case, the model is heavily violated. This is a limitation of any strong model that may not be suitable for data it is supposed to interpret.

5 Conclusion

We designed several structure (shape and alignment) models for binary segmentation using auxiliary pixel labels and constraining their pairwise compatibilities. We presented languages where segmented patches are axis-parallel non-overlapping rectangles, and a language where the segmented

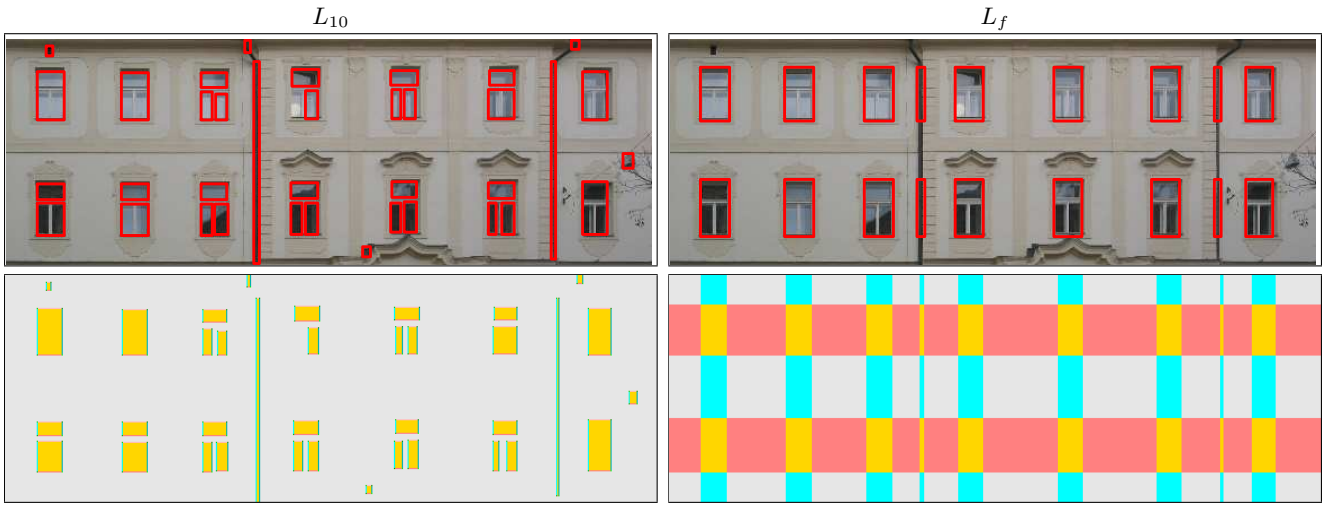


Figure 15: Results on real façade images. Detection results and color-coded labeling maps (words from languages).

Figure 16: Segmentation maps using models of decreasing strength.

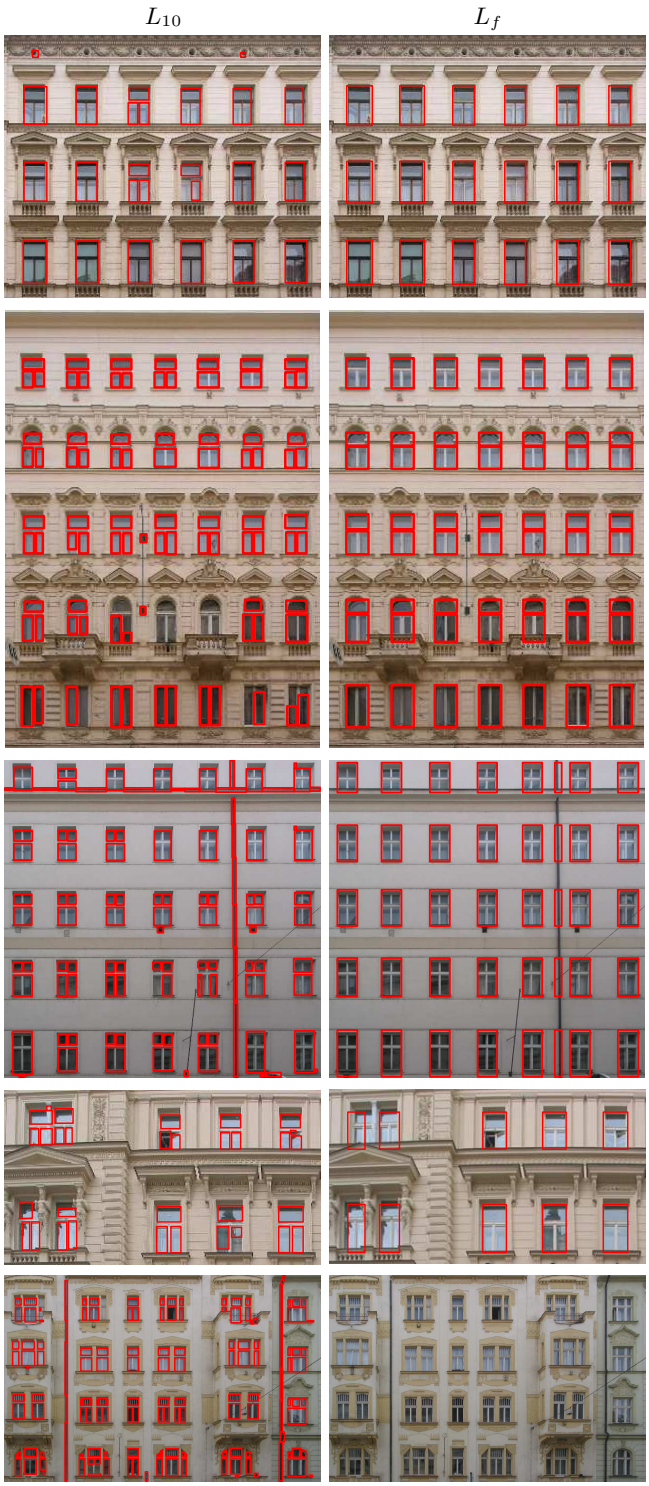


Figure 17: Other results on real façade images.

rectangles are aligned both horizontally and vertically with an arbitrary spacing between rows and columns.

Based on the notion of B-nesting of languages, we showed that incorporating a stronger structure model has a large impact on the quality of segmentation results, and that more constrained models (provably smaller languages) indeed perform better in cases where the model captures the real structure of the scene.

Although our task leads to NP-hard problem, we showed that solutions obtained from an approximate algorithm (Werner, 2007) are acceptable even when the data model is quite poor. The method fails in cases where the model used for constrained segmentation is heavily violated. We believe this is a fundamental limitation of any task posed as optimal labeling: it is not possible to 'recognize' the region of model validity and give up explaining the rest of the image. Any special 'turn-off' labels must model the shape of the valid region but the shape of the region is free-form and thus the structure of the 'turn-off' labels cannot be learned.

Acknowledgment This work has been supported in part by EC under project FP6-IST-027113 eTRIMS and in part by the Czech Academy of Sciences under project 1ET101210406.

References

Alegre, F., & Dellaert, F. 2004. A Probabilistic Approach to the Semantic Interpretation of Building Facades. Pages 1–12 of *International Workshop on Vision Techniques Applied to the Rehabilitation of City Centers*.

Baxter, R.J. 1990. *Exactly Solved Models in Statistical Mechanics*. Academic Press, New York.

Boykov, Y., & Kolmogorov, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. *IEEE Trans. on PAMI*, **26**(9):1124–1137.

Čech, J., & Šára, R. 2008. Windowpane Detection based on Maximum Aposteriori Probability Labeling. Pages 3–11 of *Image Analysis - From Theory to Applications*. Research Publishing Services.

Dick, A., Torr, P., & R., Cipolla. 2004. Modelling and Interpretation of Architecture from Several Images. *IJCV*, **60**(2):111–134.

Felzenszwalb, P. F., & Huttenlocher, D. P. 2006. Efficient Belief Propagation for Early Vision. *IJCV*, **70**(1):41–54.

Flach, B. 1998. *A Diffusion Algorithm for Decreasing Energy of Max-Sum Labeling Problem*. Fakultät Informatik, Technische Universität Dresden, Germany. Unpublished manuscript.

Flach, B., & Schlesinger, M. I. 2000. A Class of Solvable Consistent Labeling Problems. Pages 462–471 of *IAPR International Workshops on Advances in Pattern Recognition*.

Geman, S., & Geman, D. 1984. Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images. *IEEE Trans. on PAMI*, **6**(6):721–741.

Kato, Z., & Pong, T.-C. 2006. A Markov Random Field Image Segmentation Model for Color Textured Images. *Image and Vision Computing*, **24**(10):1103–1114.

Kolmogorov, V. 2006. Convergent Tree-reweighted Message Passing for Energy Minimization. *IEEE Trans. on PAMI*, **28**(10):1568–1583.

Kolmogorov, V., & Zabih, R. 2004. What energy functions can be minimized via graph cuts. *IEEE Trans. on PAMI*, **26**(2):147–159.

Komodakis, N., Paragios, N., & Tziritaz, G. 2007. MRF Optimization via Dual Decomposition: Message Passing Revisited. Pages 1–8 of *ICCV*.

Kovalevsky, V. A., & Koval, V. K. 1975. *A Diffusion Algorithm for Decreasing Energy of Max-Sum Labeling Problem*. Glushkov Institute of Cybernetics, Kiev, USSR. Unpublished manuscript.

Mayer, H., & Reznik, S. 2005. Bulding Facade Interpretation from Image Sequences. Pages 55–60 of *ISPRS Workshop CMRT*.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco.

Průša, D., & Hlaváč, V. 2007. Mathematical Formulae Recognition using 2D Grammars. Pages 849–853 of *ICDAR 2007: Proceedings of the 9th International Conference on Document Analysis and Recognition*, vol. 2. IEEE Computer Society.

Schlesinger, M. I. 1989. *Mathematical Tools of Image Processing*. Naukova Dumka, Kiev. In Russian.

Schlesinger, M. I., & Hlaváč, V. 2002. *Ten Lectures on Statistical and Structural Pattern Recognition*. Kluwer Academic Publishers. chapter 10.

Werner, T. 2005. *A Linear Programming Approach to Max-Sum Problem: A Review*. Tech. rept. CTU–CMP–2005–25. Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic. http://cmp.felk.cvut.cz/cmp/software/maxsum/.

Werner, T. 2007. A Linear Programming Approach to Max-sum Problem: A Review. *IEEE Trans. on PAMI*, **29**(7):1165–1179.

Wilson, R., & Li, C.-T. 2002. A Class of Discrete Multiresolution Random Fields and Its Application to Image segmentation. *IEEE Trans. on PAMI*, **25**(1):42–55.

Woods, J. W. 1972. Two-Dimensional Discrete Markovian Fields. *IEEE Trans. on Information Theory*, **18**(2):232–240.

Zhu, S. C., & Mumford, D. 2006. A Stochastic Grammar of Images. *Foundations and Trends in Computer Graphics and Computer Vision*, **2**(4):259–362.