

Not Far Away from Home: A Relational Distance-based Approach to Understand Images of Houses

Laura Antanas, Martijn van Otterlo, José Oramas M., Tinne Tuytelaars, Luc De Raedt

Katholieke Universiteit Leuven, Belgium

Abstract. Augmenting vision systems with high-level knowledge and reasoning can improve lower-level vision processes, such as object detection, with richer and more structured information. In this paper we tackle the problem of delimiting conceptual elements of street views based on *spatial relations* between lower-level components, e.g. the element ‘house’ is composed of windows and a door in a spatial arrangement. We use structured data: each concept can be seen as a graph representing spatial relations between components, e.g. in terms of right, up, close. We employ distances between logical interpretations to match parts of images with known examples and describe experimental results.

1 Introduction

Interpreting visual scenes is a hard task, and the field of computer vision has developed many techniques over the past decades for *segmentation*, *classification* and *recognition* of *objects* and *scenes*. Many of those techniques use a plethora of low-to medium-level and local features such as *lines*, *blobs*, *regions*, *interest points*, and many more [1, 2]. Most of these features are used in feature-based classifiers [3–6]. However, it is fairly intuitive for most people that visual scenes can best be described in terms of *hierarchical structures*, expressing the natural composition of scenes into *objects*, *parts* of objects and lower-level *substructures* [7, 8]. For example, a typical house consists of windows, one or more doors, possibly a chimney displayed in a particular *configuration*. A hierarchical aspect here is that the chimney itself, is composed of a specific arrangement of local features (e.g. “brick”-like patterns). Therefore, visual scenes are best described using high-level representations such as graphs, and more generally using *logical languages* [9]. The use of such formalisms in vision has always been desired, although the actual computational *use* of such languages for representation, inference and learning has been less often studied (but see [10, 11]).

In this paper, we start from logical and relational learning and investigate how logical generalization techniques can help to recognize and delineate substructures in an image. In order to do this, we propose a distance-based technique for image interpretation. In a more general framework, our aim is to employ a hierarchical representation of images where each image consists of several layers of information. The base layer is a set of features generated by a vision system, e.g. lines and local patterns. A subsequent layer consists of objects, e.g. windows and doors, and a higher level consists of *configurations* of objects. In this paper we focus on the delineation of known substructures at one particular layer – the house level. We assume access to manually labeled examples of houses. Each example is annotated with the locations and shapes of windows



Fig. 1: (a) Annotated/labeled image (Eindhoven). (b) Annotated image (Eindhoven).

and doors. We represent a house as a set of objects and a set of spatial relations defined on them (hence; a relational attribute graph). Each image substructure is *spatially embedded* in a 2D plane, and objects are related to each other with respect to this space.

Related to our work, several papers have explored structured models for building facades [12, 13], but as far as we know, none of these models is based on distances. In [14] a distance measure is employed for images of documents, and similar problems were tackled using inductive rule learning [15]. Our approach is different in that we use very recent general results on distance *metrics* for structured data to show how easily they can be used for computer vision tasks. In addition, we focus on a new problem; that of *delineating* houses in street view images. This can be very useful to enhance GOOGLE Street View images (e.g. with 3D reconstruction), or for home delivery robots to better guess the destination.

2 Setting

For our problem setting we work with street view images of houses. In the Netherlands many streets exist along which houses are built in one block as to minimize heat lost and to keep a uniform architecture (Fig. 1). They exhibit some variation in doors and windows appearance (e.g. windows having different frame colors), however often there is considerable *consistency* in the way these elements are *structured* at the house level. For example, the door is always on the left or right side of the house and a window is always above a door. The limited number of *configurations* define the concept *house*. In this work we identify such structures from real images. Utilizing this, we solve the *delineation* problem, where the goal is to distinguish individual houses in images that depict rows of adjacent houses, i.e. of repeated structures. We can use the same setup for other knowledge levels (e.g. going from lower-level features to the concept *window*), but in this paper we stop at the level of houses consisting of windows and doors.

Detecting house structures from images assumes access to manually labeled examples. Each house image is annotated with the bounding boxes and the labels of its elements in the image: doors and windows (Fig. 1(a)). This captures the inherent structure of the concept of a house. The configuration is extracted from these features by defining 2D *spatial relations* such as *right*, *above*, *left*, *below*, *close* and *touch* on the labeled bounding boxes. In this way, any structure can be expressed in terms of bounding boxes, their labels and spatial relations between them.

In order to map images to logical representations, we introduce some terminology. A logical *atom* is an expression of the form $p(t_1, \dots, t_n)$ where p/n is a *predicate symbol* and t_i are *terms*. We assume a functor-free language, hence terms are built from constants and variables. *Constants* are denoted in lower case and *variables* in upper case. *Ground* atoms do not contain variables and will be called *facts*. A *Herbrand interpretation* i assigns to each fact in the language a truth-value. We identify i with the set of facts $\{a_1, \dots, a_N\}$ to which it assigns *true*. A *substitution* $\theta = \{X_1/t_1, \dots, X_n/t_n\}$ is an assignment of terms t_1, \dots, t_n to variables X_1, \dots, X_n . Given i_1 and i_2 interpretations, i_1 θ -*subsumes* i_2 iff there is a substitution θ such that $i_1 \subseteq i_2$.

Now we can describe an image Z as follows. First, we obtain a set of *objects* $\{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_n\}$, by assigning to each bounding box¹ a constant \mathbf{o}_i . We derive a ground atom for each object j in the form $\text{part}(\mathbf{o}_j, \text{label})$, forming the set $\mathbf{O}(Z)$. Second, we use definitions of spatial relations between bounding boxes in our background knowledge (BK) to derive a set of spatial relations that hold among the objects in $\mathbf{O}(Z)$. The resulting set is denoted $\mathbf{R}(Z)$. An example of such an atom derived from the spatial relation *close right* is $\text{cRight}(\mathbf{o}_j, \mathbf{o}_k, \text{distance_value})$, which says that object \mathbf{o}_j is close², on the right of object \mathbf{o}_k . The term *distance_value* should be within a threshold for the relation to be true. Similarly, *cAbove* and *tRight* define *close above* and *touch right*, respectively. The BK can easily be extended, and enables to construct a logical representation of visual data:

Definition 1. A *visual interpretation* V of an image Z is the union of a set of object atoms $\mathbf{O}(Z)$ and the set of spatial relation atoms $\mathbf{R}(Z)$.

A visual interpretation can be seen as a graph; object atoms are attributed vertices and relation atoms are directed (attributed) edges between the vertices. New relations can be used to extend each visual interpretation, by defining them in the BK or adding new attributes to the existing ones. The key point about visual interpretations is that they are fully determined by the set of objects and the background knowledge. This implies that for one image, we can construct multiple visual interpretations by considering different subsets of the objects in the image, i.e. considering different subgraphs in the image. These graphs – in the context of our application in vision – will typically be connected.

We can now use the visual interpretations as an *instance space*, and in effect, a *concept* represents a set of visual interpretations. For example, some visual interpretations will belong to the concept 'house' whereas many others will not. For a particular image and its corresponding set of objects \mathbf{o} , the different possible instances will be the visual interpretations of the subsets $\mathbf{o}' \subseteq \mathbf{o}$. We denote ζ as the set of all labeled examples of a concept, called *prototypes* (Fig. 1(a)).

Example 1. A visual interpretation of the image in Fig. 1(b) is:

$$I_{\text{img}} = \{\text{part}(\mathbf{o}_1, \text{window}), \text{part}(\mathbf{o}_2, \text{door}), \text{part}(\mathbf{o}_3, \text{window}), \text{part}(\mathbf{o}_4, \text{window}), \\ \text{part}(\mathbf{o}_5, \text{door}), \text{part}(\mathbf{o}_6, \text{window}), \text{part}(\mathbf{o}_7, \text{window}), \text{part}(\mathbf{o}_8, \text{window}), \\ \text{part}(\mathbf{o}_9, \text{window}), \text{part}(\mathbf{o}_{10}, \text{window}), \text{cRight}(\mathbf{o}_2, \mathbf{o}_1, 60.0), \text{tRight}(\mathbf{o}_3, \mathbf{o}_2, 1.0), \\ \text{cRight}(\mathbf{o}_4, \mathbf{o}_3, 10.0), \text{cAbove}(\mathbf{o}_9, \mathbf{o}_3, 68.0), \text{cAbove}(\mathbf{o}_{10}, \mathbf{o}_1, 73.0), \text{cRight}(\mathbf{o}_9, \mathbf{o}_{10}, 70.0)\}$$

¹ Bounding box is a general term; in our experiments we employ polygon-like shapes.

² In practice we use approximate measures to correct for slight deviations stemming from noise.

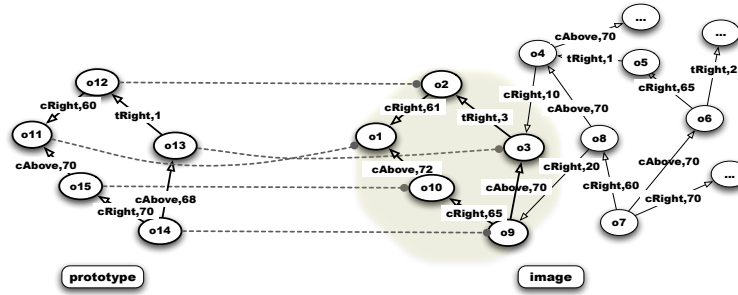


Fig. 2: Graph representations of a prototype and an image interpretation.

$cRight(o_8, o_9, 20.0)$, $cRight(o_8, o_4, 70.0)$, $cRight(o_7, o_8, 60.0)$, $cAbove(o_7, o_6, 70.0)$, $cRight(o_6, o_5, 65.0)$, $tRight(o_5, o_4, 1, 0)$. The prototype house in Fig. 1(a) is:
 $\zeta_i = \{\text{part}(o_{11}, \text{window}), \text{part}(o_{12}, \text{door}), \text{part}(o_{13}, \text{window}), \text{part}(o_{14}, \text{window}), \text{part}(o_{15}, \text{door}), cRight(o_{12}, o_{11}, 60.0), tRight(o_{13}, o_{12}, 1.0), cAbove(o_{14}, o_{13}, 68.0), cRight(o_{14}, o_{15}, 70.0), cAbove(o_{15}, o_{11}, 68.0)\}$.

Intuitively our goal is to look for known structures in a new image by trying to *embed* prototypes as well as possible in the image. In this direction we define the following:

Definition 2. A *matching* between two interpretations i_1 and i_2 , $m(i_1, i_2)$, is a mapping such that each atom $a_1 \in i_1$ corresponds to at most one atom $a_2 \in i_2$ and vice versa. To each matching we associate a *dissimilarity score* $d(i_1, i_2)$, which indicates how different the two interpretations are.

A possible matching between two interpretations (depicted as graphs) is shown in Fig. 2. The quality of the matchings is evaluated by the dissimilarity score. In the next section we will express this score in terms of a *distance metric* between interpretations. We formulate the delineation problem in the following, general, way:

Definition 3. The *delineation problem* is defined as: **given** a set of prototypes ζ , a visual interpretation V of image Z , a dissimilarity score d , **find** the set of matchings between sub-interpretations of V and any of the prototypes in ζ , **such that** all objects appearing in V are matched once, and the score d over the matchings is minimized.

In effect, solving the delineation problem will carve up the visual interpretation of an image into a set of known structures, i.e. individual houses in this paper.

3 Approach

We propose a possible scoring function d and show how to match prototypes in a new image Z . We combine *structure matching* and *distances* on interpretations. Our method consists of four steps. First, we define spatial BK, and the set of prototypes ζ . Second, we determine all candidate visual sub-interpretations of V . Third, we compute distances

between all our candidate structures and our prototypes. Fourth, we use the computed distances to find the best delineation. We will now explain the steps.

Step 1 Generate visual interpretations. We first extract image features from Z and generate a set of objects that together with BK forms a visual interpretation V . In addition, we have a set of labeled prototypes ζ generated in the same manner. We try to find groups of elements which are spatially close and we choose our BK relations accordingly, with relations such as `cRight` (more details in the experimental section).

Step 2 Generate matching candidates. Here we investigate parts of a visual interpretation that could be similar to a prototype. We only select sets of objects (and their corresponding relations) that are *connected*, resulting in the set M . Each element m of M is a possible candidate matching, and m must consist of at least two object atoms and a relation atom and at most all atoms in V . To be able to find the best delineation in case of noisy information, candidates with a small number of atoms are also needed. For example, if the image contains only a part of a (hypothetical) house, containing for example a door or window, they could be grouped with other elements or can be regarded as configurations on their own to best fit the image.

Step 3 Compute distances. To compute the quality of a matching we use a distance metric between two visual interpretations. It evaluates how well the two interpretations match structurally. Although other solutions exist ([16]), here we employ a recent result of [17] which shows that one can construct a metric for any partially ordered hypothesis space L (such as a subsumption lattice) under some mild assumptions. That is, d is a metric if it is defined in terms of the generality order on the hypothesis space, $|\cdot|$ an (anti-monotonic) and strict order preserving *size function* and $d(x, y) = |x| + |y| - 2 |m\text{gg}(x, y)|, \forall x, y \in L$. Here, `mgg` denotes the *minimally general generalization* of two hypotheses x and y . The result allows to derive distance metrics for different types of objects, including graphs, and therefore this result can be used to compute distances between interpretations. For example, one can choose a *graph isomorphism* as a partially ordered relation which induces a generality order on graphs, where the size function can be the number of vertices (see also [18]). Here the `mgg` corresponds to the *maximal common subgraph*, i.e. computing the distance between graphs g_1 and g_2 is equivalent to calculating the distance between their corresponding visual interpretations using `mgg`(g_1, g_2).

Compared to the *least general generalization* (`lgg`), which is obtained under θ -subsumption, the `mgg` represents computing the `lgg` under the assumption of *object identity* (OI) [19]. The `lgg` could also be used to find a *common part* between interpretations (resp. graphs) but it allows for different variables in the `lgg` to unify. This collapse of literals into one would violate the strictly ordering preserving condition for the size function. The `mgg` on the other hand is not unique (we can find multiple common parts) and is the result of exact *structure matching*, i.e. each constant in an interpretation (resp. each node in a graph) must be matched against a *different* constant (resp. node) in the other. Exact structure matching makes also more sense in our setting, since we want to find specific structures and do not want for example to collapse two windows into one. An illustration of `mgg` under OI-assumption is shown in Example 2.

Example 2. Let $i_1 = \{\text{cRight}(o_1, o_2, 2)\}$ and $i_2 = \{\text{cRight}(o_3, o_4, 2), \text{cRight}(o_5, o_4, 2)\}$. Under θ -subsumption $\text{m}\text{gg}_\theta = \{\text{cRight}(X_1, X_2, 2), \text{cRight}(X_3, X_2, 2)\}$ with $\theta_1 = \{X_1/o_1, X_2/o_2, X_3/o_1\}$,

Algorithm 1 Step 4 (delineate the image).

Require: prototypes ζ , distance function d , visual interpretation V and matchings M

- 1: compute $D_i = \sum_j \frac{d_{ij}}{|\zeta|}, \forall m_i \in M$
 - 2: rank $m_i \in M$ according to D_i , select the k -best, forming $M_k = \{(m_i, D_i)\}$
 - 3: let S be all subsets of M_k such that $\forall S' \in S$ all object atoms in V appear exactly once in S'
 - 4: rank all full solutions $S' \in S$ according to $d_s = \sum_{(m_i, D_i) \in S'} D_i$
 - 5: **return** the n -best solutions $S_n^* = \{S'\}$
-

$\theta_2 = \{X_1/o_3, X_2/o_4, X_3/o_5\}$. Under *OI*-subsumption there are two possible mgg's:
 $\text{mgg}_{OI}^0 = \{\text{cRight}(X_1, X_2, 2)\}$ with $\theta_1^0 = \{X_1/o_1, X_2/o_2\}$, $\theta_2^0 = \{X_1/o_3, X_2/o_4\}$ and
 $\text{mgg}_{OI}^1 = \{\text{cRight}(X_1, X_2, 2)\}$ with $\theta_1^1 = \{X_1/o_1, X_2/o_2\}$, $\theta_2^1 = \{X_1/o_5, X_2/o_4\}$.

Given the set $\text{mgg}_{\text{all}} = \{\text{mgg}(i_1, i_2)\}$, where i_1 and i_2 are interpretations we now define the distance between them in the sense of structural matching as:

$$d(i_1, i_2) = \min_{m \in \text{mgg}_{\text{all}}} (|i_1| + |i_2| - 2|m|) \quad (1)$$

where $|\cdot|$ is the number of atoms in the interpretation or in the mgg. In practice, we use a normalized metric $d(i_1, i_2) = \min_{m \in \text{mgg}_{\text{all}}} (1 - |m|/\max(|i_1|, |i_2|))$. Now we can calculate $d_{ij} = d(m_i, \zeta_j)$ which is the distance between matching m_i and prototype ζ_j .
Step 4 Delineate the image. We use d_{ij} to find the best delineation in Algorithm 1. D_i is the average of the distances from the candidate matching m_i to all the prototypes of a same concept³. It deals with the situation when among prototypes there are noisy examples⁴. We select the first k pairs (m_i, D_i) with the smallest distance, obtaining the set M_k . In a third step we take all subsets S of M_k such that the union of all object atoms in a subset $S' \in S$ is the set of object atoms in the image interpretation V and the union of all relation atoms in S' is included or equal to the set of relation atoms in V . Finally, we select from S the set $S_n^* = \{S'\}$, where S' is among the n solutions that best minimize the sum of distances d_s . Once the delineation at the house level is performed and the label for each concept instantiation is obtained we can use this information at a next layer (e.g. streets). A possible variation is to allow more relaxed versions of delineations where only some parts of the image are matched, or where matchings can overlap. Constraints can be specified to enforce that e.g. a tree cannot be part of a house.

4 Experimental setup and results

For our experiments we use street view images from Eindhoven. In our dataset there are two possible configurations depending on the position of the door (on the right or left side of the house, see Fig. 1(a)). The image dataset was collected using GOOGLE Street View, and we used the MATLAB toolbox for the LABELME image database [20] to annotate our images. For each image we annotated the windows and the doors. For the training images we annotated also the houses (Fig. 1(a)).

³ In our case the concept of house.

⁴ This can happen when perfect examples are not available, but variations from prototypes are.



Fig. 3: No complete occlusions: (a) Correct delineation. (b) Delineation obtained.



Fig. 4: Image with 5 occluded elements: (a) Correct delineation. (b) Delineation obtained.

Table 1: Delineation results. The accuracy increases as more top ranked solutions are considered.

n first solutions	1	2	3	4	5	6	7	8	9	10
Accuracy	0.73	0.76	0.83	0.9	0.93	0.93	0.93	0.93	0.93	0.96

The data is represented in XML format and then translated into PROLOG format. We use *close to the right* (`cRight`), *close above* (`cAbove`) and *touch to the right* (`tRight`) as spatial relations. Thresholds are used on the distance between house elements for `close` ($10 \leq \theta \leq \theta_{max}$) and `touch` ($\theta < 10$). θ_{max} is defined relatively to the size of the objects in the image. The amount k of best matchings is set to $\min(200 + 20\% \cdot |M|, 500)$. Choosing a k too small leads to finding no solution, while a high value can give large a search space, prevented by up-bounding k . However, the best solutions are likely to be found among the first ranked candidates. We use 2 noise-free house structures (Fig. 1(a)), one for each configuration, to delineate 30 new test images of houses with the same characteristics as the prototypes, but also different in appearance, size of house elements and distances between elements. Yet, they keep the same repeated structure of the houses as in the examples and also contain several occluded elements. We are able to delineate the houses for less occluded (Fig. 3) and for noisier images (Fig. 4). For all images we considered the first $1 \leq n \leq 10$ solutions, based on the distance value d_s . Table 1 shows the accuracy of correct delineated houses.

5 Conclusions

In this paper we have introduced a simple technique in which logic and distances between relational interpretations are used for the recognition of known structures in images. We have shown that our algorithm can identify substructures that form individual houses, effectively delineating a block of houses. Both the delineation problem, as

well as the logical decomposition utilizing distances are relatively novel aspects of our approach. Future work includes improving the efficiency of distance calculations, including attribute values in the distance, richer background knowledge bases, and more complex house structures. The most prominent direction is that of replicating our approach in a hierarchical fashion, thereby performing the interpretation process from low- to high-level. Another, straightforward direction is to employ first-order kernels [21] as our distance function.

References

1. Forsyth, D.A., Ponce, J.: *Computer Vision: A Modern Approach*. Prentice Hall (2003)
2. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision* **3**(3) (2007) 177–280
3. Li, L.J., Socher, R., Fei-Fei, L.: Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In: *IEEE Conference on Computer Vision and Pattern Recognition*. (2009)
4. Wang, G., Zhang, Y., Li, F.: Using dependent regions for object categorization in a generative framework. In: *Proceedings of CVPR06*. (2006) 1597–1604
5. Sudderth, E.B., Torralba, A., Freeman, W.T., Willsky, A.S.: Describing visual scenes using transformed objects and parts. *Int. J. on Computer Vision* **77**(1-3) (2008) 291–330
6. Bar-Hillel, A., Weinshall, D.: Efficient learning of relational object class models. *IJCV* **77**(1-3) (2008) 175–198
7. Witkin, A., Tenenbaum, J.: On the role of structure in vision. In Beck, J., Hope, B., Rosenfeld, A., eds.: *Human and Machine Vision*. Academic Press, New York (1983)
8. Pinz, A.J., Bischof, H., Kropatsch, W.G., Schweighofer, G., Haxhimusa, Y., Opelt, A., Ion, A.: Representations for cognitive vision: A review of appearance-based, spatio-temporal, and graph-based approaches. *Elec. Let. on Comp. Vision and Im. Analysis* **7**(2) (2008)
9. De Raedt, L.: *Logical and Relational Learning*. Springer (2008)
10. Needham, C.J., Santos, P.E., Magee, D.R., Devin, V.E., Hogg, D.C., Cohn, A.G.: Protocols from perceptual observations. *AI* **167**(1-2) (2005) 103–136
11. Tran, S.D., Davis, L.S.: Event modeling and recognition using Markov logic networks. In: *ECCV*. (2008) 610–623
12. Hartz, J., Neumann, B.: Learning a knowledge base of ontological concepts for high-level scene interpretation. In: *ICMLA07*. (2007) 436–443
13. Müller, P., Zeng, G., Wonka, P., Van Gool, L.J.: Image-based procedural modeling of facades. *ACM Transactions on Graphics* **26**(3) (2007) 85
14. Esposito, F., Malerba, D., Semeraro, G.: Classification in noisy environments using a distance measure between structural symbolic descriptions. *IEEE TPAMI* **14**(3) (1992) 390–402
15. Esposito, F., Ferilli, S., Basile, T.M.A., Mauro, N.D.: Discovering logical structures in digital documents. In: *Intelligent Information Systems*. (2004) 513–521
16. Ramon, J., Bruynooghe, M.: A framework for defining distances between first-order logic objects. In: *ILP '98*, London, UK, Springer-Verlag (1998) 271–280
17. De Raedt, L., Ramon, J.: Deriving distance metrics from generality relations. *Pattern Recognition Letters* **30**(3) (2009) 187–191
18. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters* **19**(3-4) (1998) 255–259
19. Khoshafian, S., Copeland, G.: Object identity. In: *1st ACM OOPSLA*. (1986) 406–416
20. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: A database and web-based tool for image annotation. *Int. J. of Computer Vision* **77**(1-3) (2008) 157–173
21. Gärtner, T.: A survey of kernels for structured data. *SIGKDD Explorations* **5** (2003) 49–58