There are Plenty of Places like Home: Using Relational Representations in Hierarchies for Distance-based Image Understanding

Laura Antanas

Katholieke Universiteit Leuven, Belgium

Martijn van Otterlo

Radboud University Nijmegen, The Netherlands

José Oramas M.

Katholieke Universiteit Leuven, Belgium

Tinne Tuytelaars

Katholieke Universiteit Leuven, Belgium

Luc De Raedt

Katholieke Universiteit Leuven, Belgium

Abstract

Understanding images in terms of logical and hierarchical structures is crucial for many semantic tasks, including image retrieval, scene understanding and robotic vision. This paper combines robust feature extraction, qualitative spatial relations, relational instance-based learning and compositional hierarchies in one framework. For each layer in the hierarchy, qualitative spatial structures in images are detected, classified and then employed one layer up the hierarchy to obtain higher-level semantic structures. We apply a four-layer hierarchy to street view images and subsequently detect corners, windows, doors, and individual houses.

Keywords: relational representations, relational instance-based learning, hierarchical image understanding

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*, *recognition* and *retrieval* of *images*, *objects* and *scenes*, e.g., [1, 2]. Many of those techniques use a plethora of local low to medium-level features such as *geometric primitives*, *patches*, *point clouds* and *invariant features* [3]. However, for high-level tasks involving complex objects and scenes such features are potentially not enough. It is more intuitive to understand and describe images in terms of hierarchical *structural* or *graph-like* representations, which reflect their natural composition into *objects*, *parts* of objects and lower-level *substructures* [4]. Man-made (vs. natural) scenes especially exhibit considerable structure that can be captured using qualitative spatial relations. For example, a typical house consists of aligned elements such as: a roof, some windows, one or more doors and possibly a chimney. A hierarchical aspect is that a window and a chimney themselves are composed of particular *configurations* of local features (e.g., corners with a certain appearance arranged in a rectangular-like way and 'brick'-like patterns of a certain shape, respectively).

This view on image representation builds on very early ideas that compositional hierarchies and relational constraints between image parts are key components of an image understanding system [5, 6, 7, 8, 9, 10]. However, then, different from today, low-and mid-level vision procedures were too immature to support such ambitious representations and goals. In this paper, we renew the idea that visual scenes are best described using high-level representational devices such as graphs, and even more generally using *logical languages* [11]. The clear advantage of these rich symbolic representations is that they can, for example, abstract spatial relations between scene components away from exact locations and generalize over similar situations, independent of the metric details. In this paper we describe a novel, model-free relational distance-based technique for hierarchical image understanding. It considers the structural aspect of a scene and employs recent relational learning developments. Instead of using a formal model of the distribution of scenes (e.g., in the form of a full grammar), we start from a set of annotated examples of objects in the scene. Yet, our framework preserves some desired properties of grammars, that is, it employs structured input features and outputs a structured explanation of the image layer-wise in the hierarchy. The base layer relies on local feature descriptors. A subsequent layer

consists of objects and higher layers consist of *configurations* of objects. Spatial logical representations are used to generalize over configurations with different number of components. We explicitly focus on the *recognition* of known substructures in street view images (i.e., windows, doors and houses), however, our approach can be used for other domains as well.

Our contribution is a new framework in which *spatial configurations* and *relational distance functions* are used throughout all layers of a hierarchy, in a unified way, to recognize known objects. Many computer vision algorithms use probabilistic classifiers, distance functions and kernels for object detection. Yet, these techniques seem less well equipped for detecting higher-level concepts that consist of qualitative spatial configurations of objects. In these cases *relational* generalization techniques [11] are required. Thus far, most work in computer vision has focused on fixed compositional structures [12] or constellation models [13]. The use of relational knowledge, in general, and compositional systems together is limited [14, 15, 16]. This paper is a contribution to this line of research by showing how recent results in relational distance metrics [17] can be used as a generalization technique to help recognize higher-level structures in images.

We assume manually labeled examples of object categories to be available throughout all layers in the hierarchy (i.e., houses, windows and doors). Each house is annotated with the locations and shapes of its constituent windows and doors. Thus, we do not assume that a full domain model, like a grammar, is available. We only assume descriptions of images at the different semantic layers of the hierarchy. Each example is represented as a set of parts and a set of general qualitative spatial relations that hold among them (hence; a relational attribute graph). Each image substructure is *spatially embedded* in a 2D plane, and parts are related to each other with respect to this space. A strong point of our framework is that distance functions at each layer of the hierarchy, either in terms of low-level features or high-level relational spatial composites, can easily be replaced by alternatives. This paper extends our previous work by evaluating the proposed framework on an augmented dataset of 164 street view images.

2. Related Work

Several papers have applied computer vision techniques to house facades. In [18, 19] structure models of meaningful facade concepts are learned from examples. In [20] the authors tackle the house delineation problem by generating vertical separating lines on the facade and using a dissimilarity measure between these features. Finally [21, 22] assume having the structure of a building facade and then estimate the parameters of the model. Different from these, our work uses distances between logical interpretations to detect known structures in an instance-based fashion.

In other domains, e.g., document analysis, distance-based techniques have been used in a relational setting [23], yet they do not address the intrinsically noisy nature of vision-based interpretation of images of houses. In fact, most papers that do address such problems perform interpretation through complete image grammars [16, 18, 24, 25, 26]. These have been well-studied in the literature [25], but need considerably more bias (or learning procedures) to supply (or learn) the grammar rules. This in contrast to our model-free approach, which is based on a comparison to annotated examples. The use of rich logical formalisms in the state-of-the-art in computer vision is limited [27].

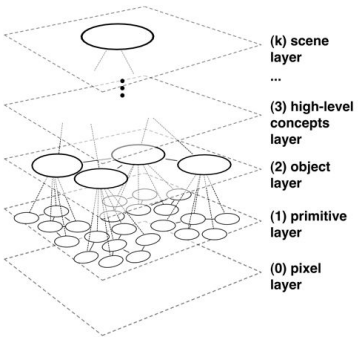


Figure 1: A typical hierarchy with $k+1$ layers. A layer i is a set of classified entities (circles) arranged in spatial configurations. Each configuration generates a classified entity at the next higher-level $i+1$ in the hierarchy.

Closely related are graph matching and graph kernel-based techniques for image understanding [28, 29]. However, different from these, our framework builds on recent general results on distance metrics for logical interpretations [17]. In this sense, we pursue a current interest in using relational learning techniques for complex vision tasks [30]. Other relevant work includes approaches based on relational object models [31] or probabilistic relational learning [32].

Hierarchical representations for image understanding have been exploited in both older and more recent works [14, 15, 33]. A clear advantage is the use of different levels which reduce the semantic gap between concepts to be detected (e.g. corners and houses in the house facade domain). Our work builds on this idea, however, it is novel in that it also employs relational representations at each layer.

3. The Hierarchical Framework

In our hierarchical framework an image Z is described at several layers $0,\ldots,k$ in the hierarchy, with 0 the base layer and k the top layer (Figure 1). At each layer, the description consists of a set of classified regions of interest or parts C_i as well as the spatial relationships among them. The classes denote the concepts the parts belong to. The task then is to use the description of an image at layer i to obtain and classify the parts C_{i+1} at the next higher layer $i+1$ in the hierarchy. We call this the *semantic segmentation* task. Images annotated at all layers are available as training data.

In our case, the *base layer* consists of the image itself, with the pixels as parts. In the *primitive layer* the parts are *local patterns*, e.g., a corner. The *object layer* is then built from *spatial configurations* of such local patterns, forming regions of

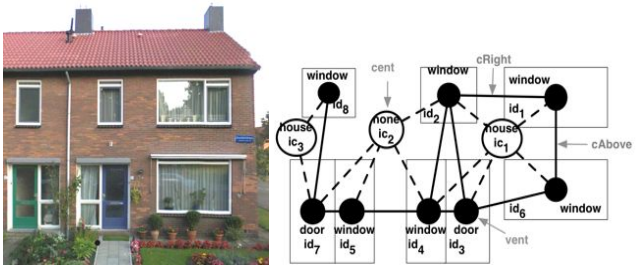
interest belonging to concepts such as *door* and *window*. These are then used at the next layer, i.e., the *house layer*, to find higher-level parts representing *houses*. We stop at the *scene layer* which groups houses into streets. Each layer consists of parts and the classes they belong to, and it is formed by making use of spatial configurations of parts from the previous lower-level layer. This *hierarchical image understanding* framework propagates the detected parts in a *bottom up* manner through each layer. Information flow is similar at all layers; first, the parts C_{i-1} of the previous layer are *detected*, then current-layer parts are generated using *configurations* of C_{i-1} and finally the *best ones* C_i are further employed at the next layer.

4. Layer-wise Representation and Function

We describe in more detail how an image Z is represented at one layer in the hierarchy. We assume knowledge about the layer identity and access to automatically detected and extracted regions of interest in the image at this layer, together with their descriptions. Based on these assumptions we define a *language* consisting of visual entities, spatial relations between visual entities, composite entities and membership relations between a visual entity and a composite entity. The language can differ from one layer to another, depending on the properties of the parts at each layer.

A *visual entity* $\texttt{vent}(id, attr1, \ldots)$ represents a part or entity in the image at the current layer i, e.g., a corner or a window with *id* as its unique object identifier. Attributes of a visual entity are its position, i.e., the coordinates of its bounding box, and its class label. *Spatial relations* impose a structure on visual entities (e.g. spatial neighborhood) and are defined using a logical background theory (a set of Prolog rules as in relational learning [11]). As an example consider the spatial relation $\texttt{cRight}(id_1, id_2, dist)$ (close aligned horizontally to the right) with an attribute for the Euclidean distance *dist* between visual entities id_1 and id_2. A *composite entity* $\texttt{cent}(ic, attr1, \ldots)$ is a candidate visual entity with identifier *ic* at layer $i+1$; it represents a set of visual entities at layer i and the relations that hold among them; thus it implicitly groups a set of visual entities into a composite entity using *membership relations* $\texttt{partof}(id, ic)$. All visual entities, composite entities, spatial and membership relations for image Z at one layer are denoted V_Z, C_Z, S_Z and M_Z, respectively. We define a *visual interpretation* I_Z of image Z as their union.

For any composite entity c we denote V_c as the set of visual entities grouped by c, S_c as the set of spatial relations representing the projection of S_Z on V_c and M_c as the set of membership relationships between the elements of V_c and c itself. We further denote I_c as the subset of I_Z that contains V_c, S_c, M_c and c itself. Finally, VS_c consists of V_c and S_c. An example of a visual interpretation at the house layer is given in Figure 2(b). Some elements of C_Z capture the inherent structure of the concept *house*; the rest belongs to the class *none*. It is convenient to visualize interpretations as graphs in which the entities correspond to vertices and the relations to directed labeled edges. A composite entity then represents the *subgraph* VS_c.



(a) An image Z (left) and its graphical representation (right). Each visual entity corresponds to a detected door/window (black circle) with its spatial location l_i (white rectangle). Each composite entity (white circle) is a possible house defined by a subgroup of visual entities. The spatial and membership relations are marked by the continuous and interrupted lines, respectively.

$I_Z = \{\texttt{vent}(id_1, l_1, win), \texttt{vent}(id_2, l_2, win), \texttt{vent}(id_3, l_3, door),$
$\texttt{vent}(id_4, l_4, win), \texttt{vent}(id_5, l_5, win), \texttt{vent}(id_6, l_6, win),$
$\texttt{vent}(id_7, l_7, door), \texttt{vent}(id_8, l_8, win), \texttt{cRight}(id_1, id_2, d_1),$
$\texttt{cAbove}(id_2, id_3, d_2), \texttt{cRight}(id_3, id_4, d_3), \texttt{cRight}(id_4, id_5, d_4), \ldots,$
$\texttt{cent}(ic_1, l_9, house), \texttt{cent}(ic_2, l_{10}, none), \texttt{cent}(ic_3, l_{11}, house),$
$\texttt{partOf}(id1, ic_1), \texttt{partOf}(id_2, ic_1), \texttt{partOf}(id_2, ic_2),$
$\texttt{partOf}(id_5, ic_2), \ldots\}.$

(b) Visual interpretation at the house layer for Z. Spatial relations are `cRight` (close to the right) and `cAbove` (close above).

Figure 2: Image representation at one layer.

Our goal is to recognize visual structures in a new image at each layer. We approach it in three steps. The first step is the generation of a set of composite entities C, from which only relevant ones become visual entities at the next layer. The *generation* of meaningful new entities is a novel task in the relational learning context. It can be seen as a dual to *predicate invention* [34]. There the goal is to determine new and useful predicates. Here the task is to *invent new entities*.

The second step is *classification*. For each of the candidate composite entities in C, we need to determine the concept they belong to (if any). This can be cast into a concept-learning problem. For each target class (such as *house*, *window* and *door*) we have examples in our training set. Consider, for instance, the concept of a house. In Figure 2(a) the composite entity ic_1 forms a positive example, while ic_2 is a negative example. For each composite entity c in the training set that forms a positive example of a concept, we use the visual interpretation VS_c as *prototype*. Such a prototype is shown as a graph in Figure 4 on the left, where it is matched with a part of an image interpretation. Its corresponding visual interpretation is presented in Figure 3. The composite entity classification task is solved using an instance-based learning approach. We use a relational distance measure to find the relevant matches of candidate composite entities with prototypes.

Each composite entity in C is classified in a local manner, by taking into account the entity to be classified and the set of prototypes, but no context. This may give unintuitive results at the global level. For instance, it could be that two entities with significant overlap are both classified as houses. Thus, we perform a third *selection* step in which contextual constraints are considered. Using global optimization we find the best subset of

$VS_c = \{\texttt{cent}(ic_1, l_9, house),$
$\texttt{vent}(id1, l1, win), \texttt{vent}(id2, l2, door),$
$\texttt{vent}(id3, l3, win), \texttt{vent}(id4, l4, win),$
$\texttt{vent}(id6, l6, win), \texttt{cRight}(id1, id2, d1),$
$\texttt{cAbove}(id2, id3, d2), \texttt{cRight}(id3, id4, d3),$
$\texttt{cAbove}(id2, id4, d4), \texttt{cRight}(id6, id3, d5),$
$\texttt{cAbove}(id1, id6, d6), \texttt{partOf}(id_1, ic_1),$
$\texttt{partOf}(id_2, ic_1), \texttt{partOf}(id_3, ic_1),$
$\texttt{partOf}(id_4, ic_1), \texttt{partOf}(id_6, ic_1)\}.$

Figure 3: An example of an instance in the house facade domain at the house layer. The target attribute is the class of the composite entity, i.e., *house* in this case.

classified entities C, from which we then derive our detections.

5. Layer-wise Semantic Segmentation

Given an image Z, we want to obtain a semantic segmentation at one layer by trying to best *embed* prototypes in Z. To this end, we first generate composite entities, we then classify them and we select the best ones to obtain class detections (see Algorithm 1).

I **Composite Entity Generation** (GENERATE). We generate the set of composite entities C for an image Z using a *language bias*, common in relational learning. As the number of all composite entities C_Z is exponentially large in the size of V_Z, we impose an upper bound on the number of composite entities considered. The bound, calculated image-wise, is exponentially proportional to the size of V_Z, such that $|C_Z| = |V_Z|^a$. The parameter a is constant and is established experimentally on the training data. Each composite entity maps a local configuration of visual entities, induced by the `close` relation, which is thresholded on the image characteristics. To each of these subgraphs a composite entity c is created and connected to all its visual entities using membership relations. The result is the subset C of *candidate composite entities*.

The candidate generation is done recursively for every image. It starts with a less strict threshold on the close relation and it automatically decreases this threshold at each iteration until the constraint on the upper bound of the size of C is met. To find the best semantic segmentation in case of noisy information, composite entities representing different numbers of visual entities are needed. For example, if the image contains some parts of a (hypothetical) house, they can be regarded as configurations on their own (e.g. the partial house ic_3 in Figure 2(a)). The minimum and the maximum number of visual entities represented by composite entities are given by the prototypes sizes. The upper bound is given for the set of all candidate composite entities having different sizes.

An example of a composite entity rule at the object layer, grouping three visual entities into potential windows/doors, is described in Example 1. Rules grouping more visual entities are defined in a similar way. At the house layer, the composite entities are similarly generated, without the contour segment constraints.

***Example* 1.** *A composite entity rule grouping three visual entities that satisfy a square-like spatial constraint is defined as:*

$\texttt{cent}(Id, BB) \leftarrow \texttt{sprl}(A,B), \texttt{sprl}(B,C), \texttt{edge}(E_{ab}, A),$
$\texttt{edge}(E_{ab}, B), \texttt{edge}(E_{bc}, B), \texttt{edge}(E_{bc}, C), \texttt{getid}([A,B,C], Id),$
$\texttt{getProp}([A,B,C], BB, Ar), min_{ar} < Ar < max_{ar}.$

where `sprl`/2 *returns a pair of visual entities that satisfy any spatial relation considered;* `edge`/2 *holds if there is a detected contour segment in the raw image attached to a visual entity and thus, two visual entities may be part of a composite entity if they are approximatively linked by the same contour segment;* `getid`/2 *associates a unique identifier to the newly generated candidate;* `getProp` *calculates properties of the composite entity, i.e., its bounding box (or location) and aspect ratio, given the set of visual entities. The aspect ratio of the candidate is checked to avoid unlikely long or tall candidates.*

II **Composite Entity Classification** (CLASSIFY). At all layers, except the primitive one, a k-nearest-neighbor approach based on a distance measure between two composite entities is used for composite entity classification. Each composite entity is represented by its corresponding visual interpretation I. A *matching* between any two interpretations I_1 and I_2, is a mapping such that each entity in I_1 is mapped to at most one entity in I_2. In terms of the graph representation, this corresponds to mapping the vertices from I_1 to those of I_2, as indicated in Figure 4. We define a *distance function* $d(I_1, I_2)$ that measures the quality of the mapping with two components. One characterizes the *structure* similarity, the other the *appearance*. Our choice is justified by the fact that both aspects may have impact on the matching score.

II-A **Classification: Structure.** To evaluate how well two logical interpretations match structurally, we must calculate their generalization (common part). We employ a recent result of [17] on metrics. It targets the *minimally general generalizations* of two interpretations, but applies to different types of objects, including graphs. We choose the object identity (OI)-subsumption order [35], which, for graphs, corresponds to *subgraph isomorphism*. The minimally general generalization (mgg) then is the *maximal common subgraph*. This means that vertices in the subgraph can be mapped to *at most one* vertex in the supergraph, imposing an exact structure matching, and thus the mgg is not necessarily unique [11]. Example 2 illustrates the mgg under OI-subsumption.

***Example* 2.** *Let* $I_1 = \{\texttt{cRight}(\texttt{o}_1, \texttt{o}_2, 2)\}$ *and* $I_2 = \{\texttt{cRight}(\texttt{o}_3, \texttt{o}_4, 2), \texttt{cRight}(\texttt{o}_5, \texttt{o}_4, 2)\}$.
The mgg*s are:*
$\text{mgg}^1_{OI}(I_1, I_2) = \{\texttt{cRight}(\texttt{X}_1, \texttt{X}_2, 2)\}$ *with substitutions* $\theta^1_1 = \{\texttt{X}_1/\texttt{o}_1, \texttt{X}_2/\texttt{o}_2\}$, $\theta^1_2 = \{\texttt{X}_1/\texttt{o}_3, \texttt{X}_2/\texttt{o}_4\}$ *and*
$\text{mgg}^2_{OI}(I_1, I_2) = \{\texttt{cRight}(\texttt{X}_1, \texttt{X}_2, 2)\}$ *with substitutions* $\theta^2_1 = \{\texttt{X}_1/\texttt{o}_1, \texttt{X}_2/\texttt{o}_2\}$, $\theta^2_2 = \{\texttt{X}_1/\texttt{o}_5, \texttt{X}_2/\texttt{o}_4\}$.

Consequently, the mgg for two interpretations I_1 and I_2 results in a set $\text{mgg}_{\text{all}} = \{\text{mgg}(I_1, I_2)\}$. Using one mgg from the set, the distance between two interpretations I_1 and I_2 is equivalent to:

$$d_s = |I_1| + |I_2| - 2|mgg(I_1, I_2)|,$$

where $|\cdot|$ is the number of the vertices in the interpretation. From this, it is straightforward to derive a *normalized struc-*

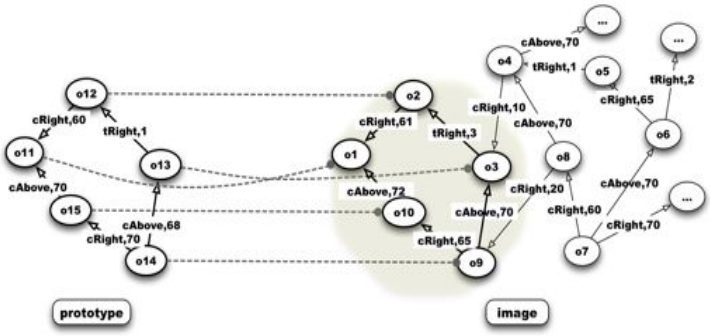


Figure 4: Graph representations of a prototype (left) and an image interpretation (right).

tural distance $d_{ns}(I_1, I_2)$. Similar distance measures are defined in [36, 37, 38].

II-B **Classification: Appearance.** In addition to structural similarities, properties of entities (e.g., color) are important. If *mgg* represents the maximal common *structure* between two interpretations I_1 and I_2, then $mgg\theta_1$ and $mgg\theta_2$ are specialized maximal common parts of *mgg* that correspond to I_1 and I_2, respectively. The substitutions θ_1 and θ_2 specify the mapping between different entities. Indeed, if $V/e_1 \in \theta_1$ and $V/e_2 \in \theta_2$ then e_1 is mapped onto e_2. We can now define a *normalized appearance distance* between the two interpretations I_1 and I_2 as:

$$d_{na}(I_1, I_2) = \frac{1}{|mgg|} \times \sum_{a \in mgg} d_0(a\theta_1, a\theta_2),$$

where a is an atom in *mgg*. Since *mgg* gives the common structure of the two interpretations, in order to compute $d_{na}(I_1, I_2)$ we start from *mgg* and specialize each atom $a \in mgg$, such that $a\theta_1$ and $a\theta_2$ are ground atoms with the same predicate symbol a. Let S denote the set of all symbols, then the distance $d_0 : S \times S \rightarrow [0,1]$ is a normalized distance measure defined for our particular application in the following way. Let t_i, s_i be attributes, then:

$$d_0(a(t_1, ...t_n), a(s_1, ..., s_n)) = \frac{1}{n} \times \sum_{i=1}^{n} d_0(t_i, s_i),$$

For *discrete* attributes we employ the hamming distance $d_0(t_1, t_2) = 1$ if $t_1 = t_2$, otherwise 0. For *numerical* attributes in the range $[min, max]$:

$$d_0(t_1, t_2) = \frac{abs(t_1 - t_2)}{max - min},$$

The structural and appearance-based aspects of the distance measure are *combined* into a global measure using a (normalized) weighted average:

$$d(I_1, I_2) = \min_{m \in \text{mgg}_{\text{all}}} (w_s \times d_{ns}(I_1, I_2) + w_a \times d_{na}(I_1, I_2)),$$

where $w_s + w_a = 1$. These weights can be supplied or learned. Because the mgg of interpretations I_1 and I_2 is not unique, a minimum over mgg_{all} is required.

We employ a *k-nearest neighbor classifier* (KNN). Given the set of composite entities C and the set of prototypes ζ, the algorithm evaluates the quality of each composite entity by computing the distance to the prototypes and classifies it based on the majority vote of its neighbors. The algorithm returns the set C_{ev} of triplets (y, d_ζ, c), where y is the class of $c \in C$ and d_ζ is the mean distance from c to the elements of the subset $\zeta^y \subseteq \zeta$ describing only concepts of class y.

III **Composite Entity Selection** (SELECT). In the selection step we first *rank* the set of composite entities of interest C according to their distances to the nearest prototypes in ζ. Then we *filter* them to obtain a reduced set C^* of C by imposing a threshold Th on the number of candidates. This is optional, but recommended as a large space of composite entities C can be generated. From this reduced set, we then select those that *together* explain best (most of) the visual features at that layer.

To this end, we formulate the composite entity selection problem as a *maximum weighted independence set problem*(WISP). Let $G = (V, E, W)$ be an undirected graph, where V, E and W are the set of vertices and edges and a vertex weighting function, respectively. An *independent set* is a set $S \subseteq V$ such that $\forall e \in E$ the two end vertices of e do not belong to S simultaneously. Then WISP is then formulated as: given an input graph $G = (V, E, W)$, find the independence set S of vertices in V such that the value $W(S)$ is maximal. To formalize our problem as a WISP, we do the following mapping:

- V becomes the set of composite entities C^*;
- $E = \{e(c_1, c_2) | c_1, c_2 \in C^*, V(c_1) \cap V(c_2) \neq \varnothing\}$ is the set of constraints between composite entities. An edge is added between two vertices if they share at least one visual entity. Thus, the solution must contain only composite entities that do not share any visual entities;
- $W : V \rightarrow \mathbb{N}$ is $W_c = \sigma(1 - d_\zeta(c, \zeta)), \forall c \in C^*$, where σ is a function which proportionally amplifies higher scores to ensure the selection of best scored composite entities. The function that we want to maximize by finding S is then $W(S) = \sum_{c \in S} W_c$.

The solution to the WISP problem is given by the function OPTIMIZE. This is an NP-hard optimization problem since our selection problem deals with general graphs. Both exact and approximation algorithms exist [39]. If the size of C^* is lower than 150 vertices we employ the exact *Cliquer* optimizer[1]. It is designed for the maximum clique problem, but it is equivalent to the WISP computed on the complement graph [40]. Otherwise, we use the approximate *QUALEX-MS* optimizer[2] [39]. Other approximation methods are also known to work in polynomial time [41]. However these are adequate for particular (i.e., planar) graphs, while our selection problem deals with general graphs.

[1] http://users.tkk.fi/pat/cliquer.html.
[2] http://www.stasbusygin.org.

Algorithm 1 Segments Z (visual entities V, prototypes ζ)

function SEMANTICSEGMENTATION(V,ζ)
  $C \leftarrow$ GENERATE (V)
  $C_{ev} \leftarrow$ CLASSIFY(C,ζ)
  Detections $\leftarrow$ SELECT(C_{ev})
  return *Detections*
end function

function CLASSIFY(C,ζ)
  return for each $c \in C$ a triple $\langle y,d,c \rangle$ where y,d are the *class*, respectively *distance* w.r.t. the prototypes in ζ according to a k-NN classifier.
end function

function SELECT(C_{ev})
  RANK candidates $(y,d,c) \in C_{ev}$ w.r.t. d
  FILTER candidates $C^* = \{(y,d,c) \in C_{ev} | \#C^* \leq \text{Th}\}$
  $\{(S,Qual)\} \leftarrow$ OPTIMIZE(C^*)
  $S^* = \arg\max_{Qual} \{(S,Qual)\}$
  Detections $\leftarrow$ PREDICT a bounding box for each $c \in S^*$
  return *Detections*$^* \leftarrow$ apply NMS on *Detections*
end function

The end goal of our framework is to *predict bounding boxes* of detected objects. We use the subgraph VS_c of the composite entity and map the bounding boxes of the visual entities V_c (i.e., vectors of 2D locations) to the bounding box corners of the object c. Also, the KNN classifier or the selection step may give multiple spatially overlapping detections for each instance of an object. Solving the WISP ensures that detections do not share any visual entities, however their bounding boxes can still overlap. After applying the bounding box prediction, we use a greedy procedure on the score to eliminate repeated detections via *non-maximum suppression* (NMS), similar to [12].

The algorithm follows the same principle for all layers of the hierarchy. However there are differences at each layer with respect to i) the interpretations generated (both size and structure) and ii) the distance function which is tuned for each layer.

6. Application and Experimental Evaluation

Dataset of 164 street view images We apply our method to 2D street view images of rows of houses (Figure 5). They commonly display a rich structure (and variety), yet are often quite consistent in terms of structure in a row of houses. We have annotated[3] 164 images of rows of house facades from different countries. A number of 20 images were collected by ourselves, the rest from Google Street View. All images show near-frontal views of the houses and no further rectification was performed. Each image has a resolution of 600x800 pixels. On these images, windows, doors and houses were manually annotated. We use the *close to the right* (`cRight`), *close above* (`cAbove`) and *touch to the right* (`tRight`) spatial relations as illustrated in Example 3. An Euclidian distance threshold is used for the `close` relation defined relatively to the size of the objects. The background knowledge can easily be extended with new relations, to enable even richer relational representations of visual data.

[3] Using the LABELME toolbox [42].

Figure 5: Images of houses in Eindhoven; an annotated training image is on the left; a testing image is on the right.

***Example* 3.** *The background knowledge for spatial relation* `cRight`*:*

$closeto(A,B,Dist) \leftarrow bbox(A,BB_1),\ bbox(B,BB_2),\ A \neq B,$
$distance(BB_1,BB_2,Dist),\ Dist < threshold.$

$cRight(A,B,Dist) \leftarrow bbox(A,BB_1),\ bbox(B,BB_2),$
$right(BB_1,BB_2), closeto(A,B,Dist).$

where bbox is the bounding box of a visual entity.

We make use of three layers in a four-layer hierarchy: *primitive*, *object* and *house* layers (Figure 6).

The *primitive layer* takes as input image pixels and groups them in corner-like features with local descriptors. We employ the KAS feature detector [43] to detect interest points formed by chains of 2 connected, roughly straight contour segments. The detector was run on the images at half their original size. We solve the classification problem by attaching a class label from the set $Y = \{cType00, cType01, cType10, cType11\}$ to each corner-like candidate. These labels represent top-right, top-left, bottom-right and bottom-left corners and are established based on the orientation of the segments composing the 2AS feature. The selection is done in two steps. Firstly, we only keep square-like corners with an angle $(90-\delta)^\circ < \alpha < (90+\delta)^\circ$. Secondly, we describe the corners with HOG descriptors [44] and train a binary classifier on these descriptors to discard irrelevant corners found on other structures than buildings (e.g., vegetation or cars). We use object layer annotations of windows and doors for training.

At the *object layer* visual entities are sparse, previously detected, corners. Each corner has a local HOG descriptor as an attribute. We employ a variation of the original HOG descriptor, with 16 orientation bins, window size of 128x128 pixels and a block size of 8x8 cells. Additionally, we use the squared root of the histogram, which showed improved results [45]. Instead of the raw descriptor we train a classifier to map each vector to a discrete attribute, either a *window* or a *door* label. Another attribute is the *corner type* (e.g., $cType00$). Based on our spatial theory, attributes representing the Euclidean distance between bounding boxes of spatially related visual entities, also contribute to the appearance-based distance. Composite entities represent possible *doors* or *windows* and are defined by subgraphs consisting of 3 up to 4 visual entities.

At the *house layer* visual entities are doors and windows found at the object layer, and composite entities represent possible *houses*. Again we employ our spatial theory to find potential composite entities, and derive attributes for the spatial relations between visual entities. Attributes of visual entities at this layer are the labels *door* and *window*. Composite entities are defined by subgraphs consisting of 2 up to 6 visual entities, estimated from the training data.

Experimental Evaluation Experiments were done in two phases. First, we performed experiments at single layers independently. More precisely, we used as input for the learning task at one single layer the annotated (for the house layer) or segmented (for the object layer), training data at that layer and then employed our method to compute the output. In this way, it is possible to get an appreciation of how difficult the learning problem is and what are the limitations of the data at each layer. Second, we performed experiments in the full hierarchical setting, that is, the inputs are image pixels and the outputs are at the house layer. This allows us to estimate how good the hierarchical approach works.

Because we have a *detection* problem we measure performance in terms of the number of true and false detections in the test set. In our setting positive predictions are all composite entities returned by the selection function. We evaluate the performance using the PASCAL VOC criterion [46]. It compares the detected bounding box BB_d to the ground-truth BB_t. If $area(BB_d \cap BB_t)/area(BB_d \cup BB_t) > 0.5$, then BB_d is a true positive (TP), otherwise it is a false positive (FP). *Precision P* is TP divided by the number of positive predictions, while *recall R* is TP divided by the number of ground-truths. The *F1* score is a measure of accuracy and is the harmonic mean of precision and recall: $F1 = 2PR/(P+R)$.

The problem of detection is often posed as a classification task, namely distinguishing in the image the class of interest with some score. Such a classifier can be turned into a detector by sliding it across the image and thresholding the scores to obtain a precision-recall curve. Differently, our formulation builds on top of a kNN classifier by selecting interesting (already scored) candidates which together find the best semantic segmentation of the image. Since they are together part of the solution, they are all predicted positive instances (except the spatially overlapping ones solved by the final NMS step). As a result, there is no obvious threshold that can be varied to trade-off precision vs. recall and instead of a precision-recall curve, the performance is measured as a precision-recall point. Since we are interested in measuring the impact of structure on our detection problem, we vary the parameter w_s of our model and show its influence on precision and recall values.

We have as parameters k (in the KNN) and the relative weights w_s and w_a (structure vs. appearance for classification). We experiment with different values of k to evaluate the influence of the structure parameter w_s on precision/recall values[4].

Results at single layer – houses At the house layer, we first test our approach directly on the ground-truth annotations, that is, on manually annotated objects such as windows and doors. We vary w_s from 0 to 1 as in Figure 7. We stress that w_s is not a threshold to trade precision for recall, but we use it to show the influence of using structure on the performance. We observe that if k is large enough ($k \geq 200$), more structure increases precision/recall values. We notice that the approach is not very sensitive to a precise value of w_s when $w_s > 0.2$. For $k = 300$ we obtain optimal values R=0.83, P=0.8 and F1=0.81 when $w_s = 0.4$. We observe that the appearance component $w_a = 0.6$ has also an influence in obtaining optimal values of precision and recall. We also note that, due to the selection procedure, precision and recall are highly coupled. For small values of k recall and precision are much lower for any w_s. That is explained by the fact that, given the structural variability at the house layer, a comparison with enough prototypes is needed.

Results at single layer – objects At the object layer the experiments are performed with available detected 2AS from the primitive layer (not annotations). They show that the variation of the structure still has an influence, though it is more limited. This can be explained by the fact that windows and doors have mostly the same structure. However, at the object layer the structure still has an indirect influence, as it is needed for computing appearance-based aspects. We ran experiments with different values for k and w_s. The results are shown in Figures 8 and 9 for classes *door* and *window*, respectively. The maximal values R=0.43, P=0.41 and F1=0.42 for class *door* and R=0.52,

Figure 6: Data flow in the four-layer hierarchy of the facades domain. Input layers: pixels, corner primitives and object entities. Corresponding output layers: corner primitives, object entities and house entities, respectively.

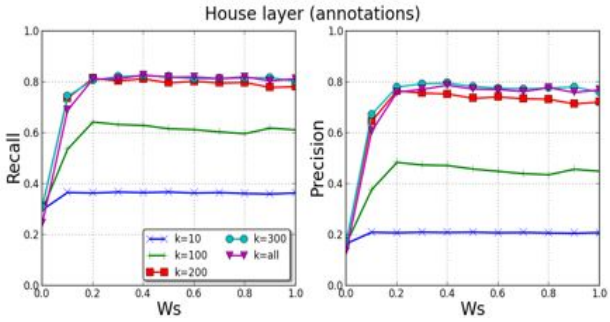


Figure 7: The influence of structure on R/P for different k.

[4]We choose w_s as the free parameter; $w_a = 1 - w_s$.

P=0.43 and and F1=0.47 for class *window* are obtained for parameters $k = 75$, $w_s = 0.4$, $w_a = 0.6$, and $k = 150$, $w_s = 0.2$, $w_a = 0.8$, respectively. However, results for other k values are close. At this layer, we do not evaluate against the entire set of training instances, but a sample of these (33%) according to the distribution of classes. A NMS step with 50% overlap was applied after the selection step.

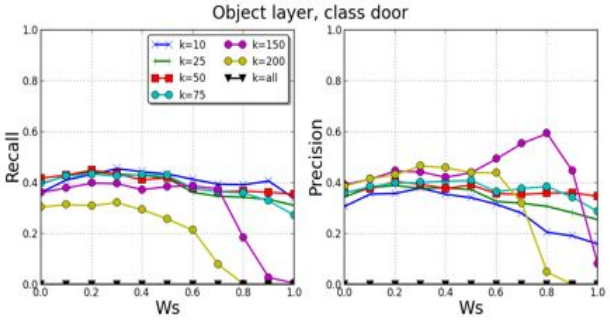


Figure 8: The influence of structure on R/P for different k.

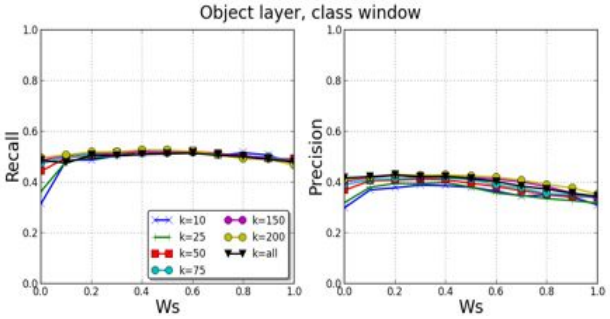


Figure 9: The influence of structure on R/P for different k.

Results at single layer – corners To asses the category accuracy of the parts that the object layer builds on, we also report results at the primitive layer. For the first classification step, establishing whether a corner is relevant or not, we obtain R=0.92, P= 0.85 and F1=0.88. The second classifier, distinguishing between window and door corners, gives as results R=0.74, P= 0.82 and F1=0.78.

Results with hierarchy – houses We evaluate detection results at the house layer using the full hierarchy. From the raw image we first detect the 2AS primitives. These are then employed further as input to detect windows and doors. At this point there are 2 possible ways to proceed. One option is to use as input for the house layer the windows and doors obtained after the selection step at the object layer. However, this gives less good results as a high enough recall is required from the object layer to obtain rich enough visual interpretations. Alternatively, instead of the full selection step, we consider the ranked composite entities on which we directly apply NMS. In this way, the full selection is replaced by a less selective mechanism, which keeps the top ranked candidates and improves the recall at the object layer (even with a NMS overlap of 0%). The selected candidates become visual entities at the house layer. This improves the results, as shown in Figure 10, to obtain for 0% NMS overlap, $k = 300$ and $w_s = 0.3$, R=0.53 and P=0.61.

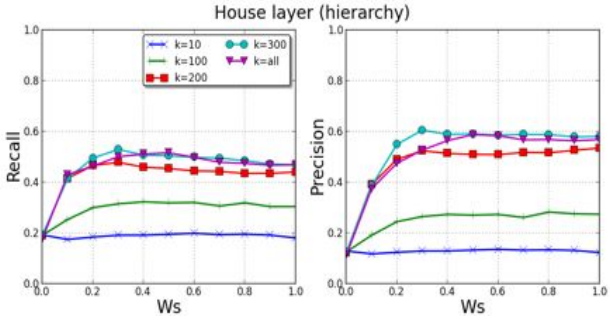


Figure 10: The influence of structure on R/P for different k.

One interesting aspect is how the number of top ranked candidates allowed at the next layer can influence the performance of the hierarchy in terms of recall/precision. We make this analysis by varying the degree of the NMS overlap across the dataset. The higher the overlap, the larger the number of candidates propagated at the house layer, image-wise. Table 1 shows this analysis. An interesting research question is how to computationally deal with this trade-off, because when the number of candidates propagated from one layer is too high, this will give computational difficulties at the next layer.

Propagated candidates (NMS overlap degree)	R	P	$F1$
0%	0.53	0.61	0.57
2%	0.57	0.65	0.61

Table 1: The influence of the number of candidates propagated at the next layer on hierarchy results.

In all experiments we perform a 5-fold cross-validation on the dataset with fixed folds. In practice, we set Th = 300 candidates for the single layer experiments. We are able to delineate houses and to separate them from neighboring houses, even when attached. Some qualitative results are presented in Figure 11 – 13. The higher we get in the hierarchy and therefore richer in the semantics, the more relevant the structural aspect becomes.

Comparison to other approaches/baselines The goal of this work is not to compete with powerful detectors, often building on dense feature representations, but rather to evaluate how structure can be flexibly exploited in detection problems in general. We show that even when starting from relatively sparse cues (Figure 6, primitive layer), detection and delineation of complex objects is feasible, thanks to the use of structure. Moreover, rather than just detecting bounding boxes of objects, our method can return a semantic hierarchical interpretation of the scene, decomposing each object into its constituents parts. For reference, we compare our method with several approaches to assess the difficulty of the problem.

As a first baseline, we use the generic object detector (Objectness) proposed in [47] and the objectness measure to quan-

tify how likely it is for an image window to contain a house[5]. We run the detector with 100 window samples. As a second baseline we combine the objectness measure with a separate classifier trained[6] for the class *house* on HOG feature descriptors [44] (Obj+HOG). The objectness classifier is used as a prior distribution to sample relevant hypotheses in the image, while the HOG classifier is used to re-score them. We first sample 100 house candidates in each image and then employ the specialized classifier to improve the predictions.

We additionally compare to two well known approaches in computer vision. One is the boosting approach[7] [48] which trains an ensemble of weak detectors for the class *house*. Each weak detector uses template matching with a localized patch in object centered coordinates. Individual houses can be more effectively detected using a template matching approach than a texture-based one, since houses in the same row have the same texture and most street scenes greatly vary in texture across the dataset. We use different numbers of weak classifiers (Boosting30–120) as shown in Table 2.

The second approach is the deformable part-based models (DPM) [12], a system that can represent highly variable objects using mixtures of multiscale deformable part models. Each model is a hierarchical star-structured model defined by a root filter (first layer) plus a set of parts filters with spring-like connections between the root and the parts (second layer). The score of a star models at a particular position and scale within an image is the score of the root filter at the given location plus the sum over parts of the maximum, over placements of that part, of the part filter score on its location minus a deformation cost measuring the deviation of the part from its ideal location relative to the root. To discriminatively train this model using object bounding boxes, a latent SVM is used. Results are reported for the standard DPM setting with one component (belonging to the front pose of the house) containing 8 parts. We use as positive examples the house bounding boxes and, although our approach does not use explicit negative examples, we provide as weak" negatives background samples of fixed size from the annotated bounding box surroundings.

Table 2 shows comparison results. The F1 values in the table are the maximum F1 scores over all precision-recall points in the obtained PR curves. Although the baseline detectors and the boosting approach perform reasonably well for the house detection problem, none of these detectors incorporates a fine-grained decomposition of a house, in the form of *structured output* which explains the image in the same way as our framework. DPM is an exception, as the trained model can be visualized in terms of its parts and displacements to the root. Still, these parts do not have an explicit meaning. Our results are comparable to DPM, however we start from sparse features and thus,less rich appearance cues. We only use as features the corners estimated from 2AS and HOG descriptors on their (reduced) neighborhood. Still, our result has room for improvement. One straightforward way is to apply the KAS feature detector on the original image size to increase the recall measure. Our method outperforms the baselines and the boosting approach.

[5]We use Version 1.5, available at `http://www.vision.ee.ethz.ch/~calvin/software.html`.

[6]Using the LIBSVM library available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[7]Available at `http://people.csail.mit.edu/torralba/shortCourseRLOC/boosting/boosting.html`.

Figure 11: Segmentation of images with partial occlusions on annotations at the house layer.

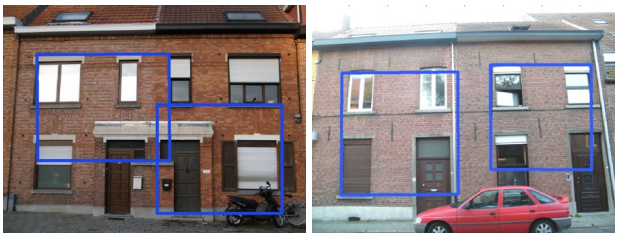


Figure 12: Segmentation of images with partial occlusions at the house layer using the hierarchy.

Figure 13: Segmentation of images at the object layer. Door detections are marked in green.

Method	R	P	$F1$
Objectness	0.21	0.08	0.12
Obj+HOG	0.35	0.10	0.16
Boosting30	0.53	0.55	0.51
Boosting60	0.52	0.51	0.53
Boosting120	0.51	0.54	0.49
DPM	0.62	0.61	0.62
Hierarchy (*house*)	0.57	0.65	0.61

Table 2: Comparison to baselines for class *house*.

Results on 60 images with hierarchy – houses In previous work [49], we performed the same experiments on a subset of 60 images of the dataset considered in this paper. For comparison, the results are summarized in Table 3. We emphasize that similar results are obtained in both settings for similar values of the parameters w_s/w_a. This shows that the importance of structure is roughly the same even when larger datasets are

considered. Our method generalizes well across larger datasets of house facades, independently of the appearance variability.

Setting	w_s	R	P	$F1$
Single layer (*house*)	0.4	0.92	0.90	0.91
Single layer (*door*)	0.3	0.42	0.47	0.44
Single layer (*window*)	0.5	0.61	0.35	0.45
Hierarchy (*house*)	0.4	0.61	0.65	0.63

Table 3: Summary of results on a subset of 60 images of the 164 dataset considered in this paper.

To summarize, we have shown that our framework gives promising results for the detection tasks at each individual layer and using the full hierarchy. A challenging aspect is the propagation of candidates up through the hierarchy. The recall obtained at one layer directly influences the performance at the next layer. If the number of allowed candidates is high enough which means that we do not just propagate the single best solution, but a larger number of candidate solutions we enable the higher layer to select from more candidates and achieve better performance. This balance between generating many candidates and propagating a suitable number of relevant candidates is an empirical question.

7. Conclusions

We have presented a novel general framework for hierarchical image understanding, incorporating distance-based classifications, relational, spatial knowledge representation and robust visual feature recognition. The experiments show i) the interplay between structural and appearance-based aspects in the recognition task and ii) good detection results both at single layers and full hierarchy. This work explores a new relational scheme for solving computer vision tasks and we believe that there is still room for improvement. Three strong points of the approach are that i) we do not assume availability of a full model of the domain (e.g., a grammar) but only a set of annotated examples, which is more natural and easier to obtain, ii) the framework can easily be extended by adding relational/spatial background knowledge, or replacing the classifiers by other similarity functions or kernels and iii) the approach incorporates a fine-grained decomposition of a house in the form of structured output which explains the image, as opposed to existing detectors.

Acknowledgments. Laura Antanas is supported by the European Commission under contract FP7-248258-First-MM.

References

[1] L.-J. Li, R. Socher, L. Fei-Fei, Towards total scene understanding: Classification, annotation and segmentation in an automatic framework, in: CVPR, IEEE, 2009, pp. 2036–2043.

[2] E. B. Sudderth, A. Torralba, W. T. Freeman, A. S. Willsky, Describing visual scenes using transformed objects and parts, IJCV 77 (2008) 291–330.

[3] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: A survey, Foundations and Trends in Computer Graphics and Vision 3 (2007) 177–280.

[4] A. J. Pinz, H. Bischof, W. G. Kropatsch, G. Schweighofer, Y. Haxhimusa, A. Opelt, A. Ion, Representations for cognitive vision: A review of appearance-based, spatio-temporal, and graph-based approaches, Electronic Letters on Computer Vision and Image Analysis 7 (2009) 35–61.

[5] A. Guzmán, Decomposition of a visual scene into three-dimensional bodies, in: Fall Joint Computer Conference, AFIPS '68 (Fall, part I), ACM, New York, NY, USA, 1968, pp. 291–304.

[6] T. Kanade, Model representations and control structures in image understanding, in: IJCAI, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1977, pp. 1074–1082.

[7] R. Haralick, Pictorial data analysis, NATO ASI series: Computer and system sciences, Springer-Verlag, 1983.

[8] A. Hanson, E. Riseman, Visions: A computer system for interpreting scenes, in: CVS78, pp. 303–333.

[9] T. Matsuyama, V. Hwang, Sigma: a framework for image understanding integration of bottom-up and top-down analyses, in: IJCAI, pp. 908–915.

[10] H. Bunke, A. Sanfeliu, Syntactic and Structural Pattern Recognition: Theory and Applications, World Scientific Pub Co Inc, 1990.

[11] L. De Raedt, Logical and Relational Learning, Springer, 2008.

[12] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, TPAMI 32 (2010) 1627 –1645.

[13] R. Fergus, P. Perona, A. Zisserman, Weakly supervised scale-invariant learning of models for visual recognition, IJCV 71 (2007) 273–303.

[14] L. Zhu, Y. Chen, Y. Lin, C. Lin, A. Yuille, Recursive segmentation and recognition templates for image parsing, TPAMI 34 (2012) 359–371.

[15] Y. Jin, S. Geman, Context and hierarchy in a probabilistic image model, in: CVPR, IEEE, 2006, pp. 2145–2152.

[16] R. B. Girshick, P. F. Felzenszwalb, D. A. McAllester, Object detection with grammar models, in: NIPS, 2011, pp. 442–450.

[17] L. De Raedt, J. Ramon, Deriving distance metrics from generality relations, Pattern Recognition Letters 30 (2009) 187–191.

[18] J. Hartz, B. Neumann, Learning a knowledge base of ontological concepts for high-level scene interpretation, in: ICMLA, IEEE, 2007, pp. 436–443.

[19] J. Hartz, Learning probabilistic structure graphs for classification and detection of object structures, in: ICMLA, IEEE, 2009, pp. 5–11.

[20] P. Zhao, T. Fang, J. Xiao, H. Zhang, Q. Zhao, L. Quan, Rectilinear parsing of architecture in urban environment, in: CVPR, IEEE, 2010, pp. 342–349.

[21] P. Müller, G. Zeng, P. Wonka, L. J. Van Gool, Image-based procedural modeling of facades, ACM Transactions on Graphics 26 (2007) 85.

[22] P. Koutsourakis, L. Simon, O. Teboul, G. Tziritas, N. Paragios, Single view reconstruction using shape grammars for urban environments, in: ICCV, IEEE, 2009, pp. 1795–1802.

[23] F. Esposito, D. Malerba, G. Semeraro, Classification in noisy environments using a distance measure between structural symbolic descriptions, PAMI 14 (1992) 390–402.

[24] M. A. Lippow, L. P. Kaelbling, T. Lozano-Perez, Learning grammatical models for object recognition, 2008.

[25] S.-C. Zhu, D. Mumford, A stochastic grammar of images, Found. Trends. Comput. Graph. Vis. 2 (2006) 259–362.

[26] K. Terzic, L. Hotz, J. Sochman, Interpreting structures in man-made scenes - combining low-level and high-level structure sources, in: ICAART, pp. 357–364.

[27] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.

[28] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, A. J. Smola, Learning graph matching, TPAMI 31 (2009) 1048–1058.

[29] Z. Harchaoui, F. Bach, Image classification with segmentation graph kernels, in: CVPR, IEEE, 2007, pp. 1–8.

[30] M. Petrou, The tower of knowledge: a novel architecture for organizing knowledge combining logic and probability, 2008.

[31] A. Bar-Hillel, D. Weinshall, Efficient learning of relational object class models, IJCV 77 (2008) 175–198.

[32] K. S. R. Dubba, A. G. Cohn, D. C. Hogg, Event model learning from complex videos using ilp, in: H. Coelho, R. Studer, M. Wooldridge (Eds.), ECAI, volume 215 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2010, pp. 93–98.

[33] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, 1983.
[34] S. Muggleton, W. L. Buntine, Machine invention of first order predicates by inverting resolution, in: J. E. Laird (Ed.), ML, Morgan Kaufmann, 1988, pp. 339–352.
[35] S. Ferilli, N. D. Mauro, T. M. A. Basile, F. Esposito, A complete subsumption algorithm, in: A. Cappelli, F. Turini (Eds.), AI*IA, volume 2829 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 1–13.
[36] S.-H. Nienhuys-Cheng, Distance between herbrand interpretations: A measure for approximations to a target concept, in: N. Lavrac, S. Dzeroski (Eds.), ILP, volume 1297 of *Lecture Notes in Computer Science*, Springer, 1997, pp. 213–226.
[37] T. Horváth, S. Wrobel, U. Bohnebeck, Relational instance-based learning with lists and terms, ML 43 (2001) 53–80.
[38] M. Kirsten, S. Wrobel, T. Horváth, Distance based approaches to relational learning and clustering, Relational Data Mining (2000) 213–230.
[39] S. Busygin, A new trust region technique for the maximum weight clique problem, Discrete Appl. Math. 154 (2006) 2080–2096.
[40] P. R. J. Östergrd, A fast algorithm for the maximum clique problem, Discrete Appl. Math. 120 (2002) 197–207.
[41] V. Lozin, M. Milanic, On the maximum independent set problem in subclasses of planar graphs, Journal of Graph Algorithms and Applications 14 (2010) 269–286.
[42] B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, LabelMe: A database and web-based tool for image annotation, IJCV 77 (2008) 157–173.
[43] V. Ferrari, L. Fevrier, F. Jurie, , C. Schmid, Groups of adjacent contour segments for object detection, TPAMI (2008) 36–51.
[44] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, IEEE, 2005, pp. 886–893.
[45] R. Arandjelovic, A. Zisserman, Three things everyone should know to improve object retrieval, in: CVPR, IEEE, 2012, pp. 2911–2918.
[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2008, 2008.
[47] T. Deselaers, V. Ferrari, Global and efficient self-similarity for object classification and detection, in: CVPR, IEEE, 2010, pp. 1633–1640.
[48] A. Torralba, K. P. Murphy, W. T. Freeman, Sharing features: Efficient boosting procedures for multiclass object detection, in: CVPR, IEEE, 2004, pp. 762–769.
[49] L. Antanas, M. van Otterlo, J. O. M., T. Tuytelaars, L. D. Raedt, A relational distance-based framework for hierarchical image understanding, in: P. L. Carmona, J. S. Sánchez, A. L. N. Fred (Eds.), ICPRAM (2), SciTePress, 2012, pp. 206–218.