A Relational Kernel-Based Framework for Hierarchical Image Understanding

Laura Antanas, Paolo Frasconi, Fabrizio Costa,
Tinne Tuytelaars, and Luc De Raedt

Katholieke Universiteit Leuven, Belgium

Abstract. While relational representations have been popular in early work on syntactic and structural pattern recognition, they are rarely used in contemporary approaches to computer vision due to their pure symbolic nature. The recent progress and successes in combining statistical learning principles with relational representations motivates us to reinvestigate the use of such representations. More specifically, we show that statistical relational learning can be successfully used for hierarchical image understanding. We employ kLog, a new logical and relational language for learning with kernels to detect objects at different levels in the hierarchy. The key advantage of kLog is that both appearance features and rich, contextual dependencies between parts in a scene can be integrated in a principled and interpretable way to obtain a qualitative representation of the problem. At each layer, qualitative spatial structures of parts 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 successfully detect corners, windows, doors, and individual houses.

1 Introduction

Understanding images by recognizing its constituent objects is a challenging task and it could be solved, in principle, using computer vision techniques that employ low- to medium-level features, such as geometric primitives, patches, or invariant features [1]. Although helpful for the recognition process, these features do not suffice for higher-level tasks dealing with more complex patterns. In this case, it is more intuitive to describe visual scenes in terms of *structural hierarchical* (or *graph-like*) representations that build on visual image parts. They reflect the natural composition of scenes into objects and parts of objects. In particular, man-made (vs. natural) scenes 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 itself is composed of rectangular-like corner configurations with a certain appearance.

This view on hierarchical image representation was embraced by early ideas that hierarchical structure and relations are key components of an image understanding system [2]. A key advantage of using relational representations [3] is their capability of exploiting contextual knowledge in images via symbolic relations. In addition, they abstract spatial information away from exact locations making it independent of metric details. Although popular in early work on syntactic or structural pattern recognition [4],

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 171–180, 2012.

relational approaches have been rarely used to solve computer vision problems (except [5, 6]). One reason is that low-and mid-level vision features were not always as mature as today to support such ambitious representations. Another reason is the limitation of pure relational approaches in handling noisy data. Yet, when combined with statistical techniques, they show robustness to noise [3, 7]. Motivated by our previous results on using distances between logical interpretations to hierarchically detect structures in images [5], we solve the same problem using kLog, a general purpose relational language for kernel-based learning. The resulting approach is more principled, as it is grounded in a statistical learning framework, is computationally more tractable and provides improved results. Our earlier approach relied on more expensive logical matching and generalization operations and was more tailored towards this particular application.

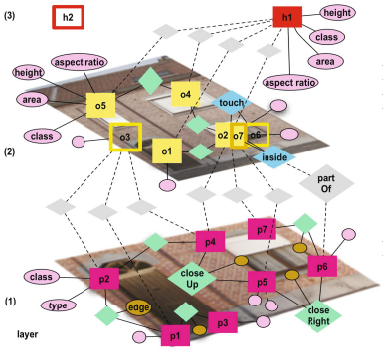
kLog [8] is a new statistical relational learning framework, which builds on ideas from statistics to address uncertainty, while incorporating a relational representation of the domain. Images are described in terms of automatically extracted semantic parts and relationships between them, thus as relational databases or (hyper)-graphs. Domain knowledge can easily be incorporated using logical rules. The novelty of kLog is that, starting from existing visual features, it can take relational contextual features into account in a principled and natural way. Furthermore, its declarative approach offers a flexible and interpretable way to consider both appearance and spatial information in an image. Finally, kLog transforms the relational databases into graph-based representations and uses graph kernels to extract the feature space. Thus, our contribution is a new approach to hierarchical image understanding, in which *spatial configurations* of scenes are combined with *kernel-based learning* for structured data to recognize objects throughout all layers of a hierarchy, in a unified way.

The goal of this paper is to understand images by recognizing objects at different layers of a hierarchy. The base layer relies on local interest points and their descriptors. A subsequent layer consists of objects, while higher layers consist of configurations of objects. We focus on the recognition of structures in street view images, yet, our approach can be used for other domains as well. We learn to recognize objects from a set of manually labeled examples of object categories, i.e., houses, windows and doors. Each house is annotated with the locations and shapes of its constituent windows and doors. The approach is evaluated on a dataset of 60 street view images.

2 Related Work

Thus far, most work in computer vision has focused on fixed compositional structures [9] or constellation models [10]. Recently, more attention was devoted to using high-level relational representations for image understanding or object recognition [11–13]. Yet, most of this work is restricted to a model-based approach and perform interpretation through image grammars. These have been well-studied [14], but need considerably more input from the user in terms of a set of grammar rules. This in contrast to our approach, which is based on learning from annotated examples and which uses domain knowledge to specify only basic qualitative spatial relations between image parts.

Several papers have addressed the problem of understanding images of house facades. In [15], structure models of meaningful facade concepts are learned from examples, while in [16], the authors tackle the house delineation problem by generating



$x = \{\texttt{part}(p_1, botL, door), \texttt{part}(p_2, topL, door),$
$\texttt{part}(p_5, botL, win), \texttt{part}(p_6, botR, win),$
$\texttt{part}(p_7, topR, win), \texttt{cUp}(p_2, p_1, d3, edge),$
$\texttt{cRight}(p_3, p_1, d2, edge), \texttt{cRight}(p_6, p_5, d3, edge),$
$\texttt{cRight}(p_4, p_2, d5, noedge), \ldots,$
$\texttt{cand}(o_1, thin, size3, h1), \texttt{cand}(o_5, thin, size2, h1),$
$\texttt{cand}(o_3, squared, size2, h2), \ldots, \texttt{partOf}(p_5, o_2),$
$\texttt{partOf}(p_6, o_2), \texttt{partOf}(p_2, o_1), \ldots$
$\texttt{inside}(o_7, o_2), \texttt{touch}(o_6, o_2), \ldots\}.$
$y = \{\texttt{class}(o_1, door), \texttt{class}(o_5, window),$
$\texttt{class}(o_3, none), \ldots\}.$

Fig. 1. A hierarchical description of a house image. Parts are squares (purple, yellow, red); relations are diamonds (green/blue – spatial/functional constraints, grey –memberships); properties are circles (pink). Parts not belonging to a class of interest are empty squares. A visual interpretation $i = (x, y)$ is on the right; x specifies the input features, while y is the learning target.

vertical separating lines on the facade and using a dissimilarity measure between these features. Finally, the works in [17, 18] assume having the structure or grammar of a building facade and estimate the parameters of the model. Closely related are graph matching and other kernel-based techniques for image understanding [19]. Different from these, our work combines the best of both worlds by using a kernel-based approach to learn from logical interpretations. The paper extends our recent results in [20] with more complex relationships and, thus, a richer feature space.

3 Hierarchical Image Understanding

In our hierarchical framework an image is described at several layers $(0), \ldots, (k)$ in a hierarchy, with 0 the base layer and k the top layer. Figure 1 shows the hierarchical structure of a partial house facade. At each layer, the image consists of a set of parts, their properties and (spatial) relationships among them. The task then is to use this information at layer i to generate and classify candidate parts at the next higher layer $i + 1$ in the hierarchy. Thus, at each layer, parts belonging to classes of interest are detected and employed at the next layer to detect higher-level concepts. As training data, annotated images are available at all layers.

In the house facade problem, 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 or an edge. The *object layer* is built from *spatial configurations* of such local patterns, forming higher-level parts that are *doors* and *windows*. These are then used at the next layer, i.e., the *house layer*, to find even higher-level parts representing *houses*. Each layer consists of parts and classes they belong to, and it is formed by making use of spatial configurations of parts from the previous lower-level layer. The hierarchical framework propagates the detected parts using a pipeline through each layer.

4 Object Detection at One Layer

Next, we describe how an image is relationally represented at one layer in the hierarchy and how our object detection problem is formalized and modeled with kLog. It is a domain specific language embedded in Prolog which allows to specify, in a declarative way, logical and relational learning problems. Figure 2 illustrates the information flow in kLog. We use the object layer as running example. Here, image parts are extracted from raw images via the primitive layer and using low-to medium-level features detectors, as described below. At the house layer the relational representation is built in a similar way using as parts the detections from the object layer.

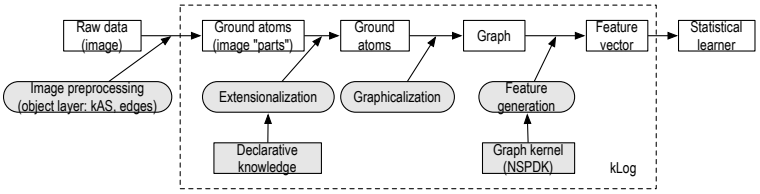


Fig. 2. From images to feature vectors in kLog

4.1 From Images to Primitive Parts

The primitive layer takes as input image pixels and groups them in corner-like features representing image parts at the object layer together with their properties. We employ the KAS detector [21] to detect corners formed by chains of 2 connected, roughly straight contour segments. Because we can get many detections we only keep square-like corners with an angle of $\approx 90^\circ$. Also, we train a binary classifier to discard irrelevant corners found on other structures than buildings (e.g., car), using the HOG descriptor [22] on the corners. We use the training annotations of windows and doors.

Each corner-like part can be one of the types in the set $\{topR, topL, botR, botL\}$ representing top-right, top-left, bottom-right and bottom-left corners, respectively. The corner type is given by the orientation of the segments composing the 2AS. We use the HOG descriptor[1] to characterize the appearance of each corner. Yet, instead of the raw descriptor we train another classifier to map each HOG to a discrete attribute, either a *window* or a *door* label. A final characteristic of a part is its estimated bounding box.

4.2 Data and Problem Modeling

We represent this information at a higher level using the classic entity/relationship (E/R) data model, a paradigm frequently used in database theory [23]. The E/R model for our problem, with some further assumptions required by kLog, is shown in Figure 3(a). It provides an abstract representation of the examples, i.e. class of interest candidate instance in this case. The elements of an E/R model are *entity sets* (in Figure 3(a) depicted

[1] A variation of the HOG descriptor with 16 orientation bins, window size of 128x128 pixels and a block size of 8x8 cells showed improved results.

as rectangles), *relationships* linking entity sets (depicted as diamonds) and *attributes* that describe objects and relationships (depicted as ovals). In kLog, the database scheme is directly derived from the E/R model, and contains two kinds of relations: those introducing entity sets (*E-relations*) and those introducing relationships (*R-relations*). As in database theory, they correspond to tuples (or facts) in the database.

In our problem, E-relations are parts of the image and candidate objects of interest. Each entity has properties and a unique identifier (underlined ovals). They can be visualized as relational facts, in Figure 1 (right). The tuple $\texttt{part}(p_1, botL, door)$ specifies a part entity, where p_1 is the identifier and the other arguments are properties extracted by the previous layer in the hierarchy. As already indicated, they are the corner type and category. The tuple $\texttt{cand}(o_1, thin, size3)$ represents a possible object of interest. It has identifier o_1 and properties describing its discretized aspect ratio and size. These are estimated from the extracted bounding box of the candidate. R-relations are linked to the entities that participate in the relationships. In our problem, we have spatial relationships amongst parts and, respectively, amongst candidates, as well as membership relations between parts and candidates. Spatial R-relations are derived from the spatial localization of the entities, i.e., bounding boxes, and extension. An example is the relationship $\texttt{cRight}(p_3, p_1, d2, edge)$, which indicates that part entities p_3 and p_1 are spatially close to each other and aligned on the X axis with p_3 to the right of p_1. It has as properties the discretized Euclidian distance between the bounding boxes and a property indicating if the two part entities are linked by a detected contour segment.

A key advantage of kLog is that it supports *extensional* as well as *intensional* relations. Extensional relations are explicitly listed sets of given relations, whereas intensional relations are defined implicitly using logical rules. In other words, intensional relations are derived from other intensional or extensional relations given a set of rules and they represent domain-related feature construction.

Declarative Feature Construction. Intensional relations are $\texttt{cUp}/4$, $\texttt{cRight}/4$, $\texttt{inside}/2$ and $\texttt{touch}/2$, derived using notions of spatial theory, $\texttt{cand}/4$ and $\texttt{partOf}/2$. As an example, the spatial relation $\texttt{cRight}/4$ is defined as a logical rule in the following way:

$\texttt{cRight}(A, B, D, Edge) \leftarrow \texttt{part}(A, \_, \_), \texttt{part}(B, \_, \_), \texttt{edge}(Edge, A),$
$\texttt{edge}(Edge, A), \texttt{right}(A, B), \texttt{close}(A, B, D).$
where $\texttt{close}(A, B, D) \leftarrow \texttt{bb}(A, BB_1), \texttt{bb}(B, BB_2), \texttt{dist}(BB_1, BB_2, D), D < th.$
and $\texttt{right}(A, B)$ is similarly defined based the bounding boxes BB_i of the part entities. In words, A is to the right of B if the min and max X coordinates of BB_1 are smaller than the minimum and the maximum X coordinates of BB_2, respectively, and if A is not too much above or below (in a fuzzy way) of B. The R-relation $\texttt{cUp}/2$ is defined in a similar way. The atom $\texttt{edge}(Edge, A)$ is true if the entity A belongs to a contour segment $Edge$.

The intensional E-relation $\texttt{cand}/4$ defines possible objects of a class of interest at one layer, i.e., doors/windows at the object layer. It is defined using the rule:

$\texttt{cand}(Id, Ar, A, H) \leftarrow \texttt{sprl}(A, B), \texttt{sprl}(B, C), \texttt{edge}(E_{ab}, A), \texttt{edge}(E_{ab}, B,),$
$\texttt{edge}(E_{bc}, B), \texttt{edge}(E_{bc}, C), \texttt{getid}([A, B, C], Id), \texttt{getprop}([A, B, C], Ar, A, H).$
where $\texttt{sprl}/2$ brings the pairs of parts that satisfy any of the spatial relations $\{\texttt{cRight}, \texttt{cUp}, \texttt{cDown}, \texttt{cLeft}\}$; $\texttt{getid}/2$ associates a unique identifier to the newly generated

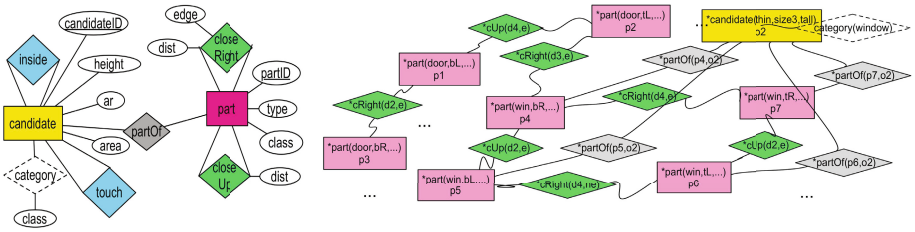


Fig. 3. a) E/R modeling of the object detection problem. Rectangles denote entity vertices, diamonds denote relationships, and circles denote properties. b) Part of the graphicalized interpretation of the image.

candidate (based on the combination of parts) and `getprop` calculates the discretized properties of the candidate relation, i.e., aspect ratio, area and height, based on the bounding box of the candidate, given the set of parts. Each candidate relation groups the three parts that satisfy a square-like spatial constraint. The membership relation `partOf`$/2$ indicates that a part belongs to a candidate.

Other intensional relations are `touch`$/2$, indicating if two candidate entities are spatially touching and `inside`$/2$, which holds if one candidate is spatially inside the other. The grounding of intensional relations is computed using Prolog's deduction mechanism and represents the extensionalization step in kLog's information flow. In the setting established above, each image is an instance of a relational database or an *interpretation*. An interpretation of an image at the object layer is exemplified in Figure 1.

Problem Definition. kLog learns from interpretations, a well-established setting in relational learning [3]. We are given a training set of n independent interpretations $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ sampled identically from some unknown but fixed distribution; x_i is a set of input ground atoms and y_i a set of output target ground atoms. In our problem and in Figure 1 the target is the unary relation `category`$/1$. The goal is to learn a mapping $h : X \rightarrow Y$, from the inputs X to the outputs Y. During prediction, we are given a partial interpretation of an image consisting of ground atoms x, and are required to complete the interpretation using h to predict the output atoms y.

4.3 Graphicalization and Feature Generation

Next, each interpretation x is converted into a bipartite graph G that has a vertex for each ground relation. Vertices correspond to grounded atoms, either E-relations or R-relations, but identifiers are removed. Edges connect E-relations and R-relations: there is an undirected edge $\{e, r\}$ if the entity identifier in e appears as an argument in r (see Figure 3(b)). Thus, edges connect vertices that share identifiers in the tuples. Role information (i.e., the position of an entity in a relationship) is retained as an edge annotation. The graph can be seen as the result of unrolling (or grounding) the E/R diagram for a particular image. There is no loss of information associated with this step.

Once interpretations are represented as graphs, any graph kernel in conjunction with a statistical learner can be used to solve the classification problem in the supervised

setting. The kLog implementation uses a variant of the fast neighborhood subgraph pairwise distance kernel (NSPDK) [24]. It has two advantages: i) it allows fast computations with respect to the graph size, as the graphicalization step can yield large graphs; ii) it is a general purpose kernel with a flexible bias, allowing us to integrate multiple heterogeneous features and context knowledge through the way it is defined.

NSPDK belongs to the large family of decomposition kernels [25] that count the number of common parts between two objects. Parts in this case are pairs of subgraphs defined as follows. Given a graph $G = (V, E)$ and a radius $r \in \mathbb{N}$, we denote by $N_r^v(G)$ the subgraph of G rooted in v and induced by the set of vertices $V_r^v \doteq \{x \in V : d^\star(x, v) \leq r\}$, where $d^\star(x, v)$ is the shortest-path distance between x and v. For a given distance $d \in \mathbb{N}$, the *neighborhood-pair* relation is then defined as $R_{r,d} = \{(N_r^v(G), N_r^u(G), G) : d^\star(u, v) = d\}$. The kernel between two graphs is then the decomposition kernel defined by relations $R_{r,d}$ for $r = 0, \ldots, R$ and $d = 0, \ldots, D$:,

$$K(G, G') = \sum_{r=0}^{R} \sum_{d=0}^{D} \sum_{\substack{A,\, B \,:\, R_{r,d}(A,\, B,\, G) \\ A',\, B' \,:\, R_{r,d}(A',\, B',\, G')}} \kappa((A, B), (A', B')). \tag{1}$$

Several choices are possible for κ. In our experiments we used an exact matching kernel where $\kappa((A, B), (A', B')) = 1$ iff (A, B) and (A', B') are pairs of isomorphic graphs, but also a soft matching kernel (see [8] for details). The maximum radius R and the maximum distance D are kernel hyperparameters. kLog provides a flexible architecture in which only the specification language is fixed. The actual features are determined by the choice of the graph kernel but also by the definition of intensional relations.

5 Summary of Experiments

We experimented on a dataset containing 60 street view images of rows of houses [5]. They commonly display a rich structure (and variety), yet, same row houses are quite consistent in terms of structure. All images show near-frontal views of the houses and no further rectification was performed. On these images, windows, doors and houses were manually annotated. We used three layers in the hierarchy: *primitive*, *object* and *house* layers. We experimented with kLog at the object and house layers, since these provide the most structure. The primitive layer serves as a preprocessing step. We measure performance in terms of precision P, recall R and F1 score and use the PASCAL VOC criterion[2] to compare the positive predicted candidate's bounding box to the ground-truth. If the overlap is larger than 50%, it is a true positive, otherwise a false positive.

Primitive Layer. To asses the accuracy of the parts categories 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 $F1 = 0.85$. For the second classification steps distinguishing between window and door corners, $F1 = 0.64$.

[2] Available at `http://pascallin.ecs.soton.ac.uk/challenges/VOC/`

Method	R	P	$F1$
RD hierarchy [5]	0.61	0.65	0.63
Boosting60	0.54	0.49	0.51
Boosting120	0.57	0.48	0.52
kLog	0.74	0.64	0.68

Method	R	P	$F1$
RD window	0.61	0.35	0.44
RD door	0.42	0.47	0.44
kLog window	0.60	0.55	0.57
kLog door	0.51	0.42	0.50

Fig. 4. kLog performance compared to baselines; classes *house* (left), *door* and *window* (right). For the feature boosting detector we use a different number of weak classifiers (Boosting60/120).

Object Layer. The experiments at the object layer are performed starting from sparse, previously detected, 2AS at the primitive layer that belong to windows or doors. We used the following features: part entity relation `part`, spatial relationships between parts `cRight`, `cUp`, candidate entity relation `cand`, membership relationship `partOf` and other spatial/functional relationships between candidate entities (such as `inside` and `touch`). At this layer, similarly, we solve the problem in two steps. First, we establish whether a candidate is relevant or not and then we distinguish between windows and doors. We vary the parameters of the kernel r and d to assess the impact of contextual features on the performance of detecting windows and doors. We obtain the best result, $F1 = 0.57$ for class *window* and $F1 = 0.50$ for class *door*, when $r = 2, d = 4$.

House Layer. Candidates classified as *window* or *door* become parts at the house layer. We used a variation of the same relations (e.g., the absence of property $edge$). Again, we vary the parameters r and d to assess the impact of contextual features on the performance of detecting houses and obtain $P = 0.64$, $R = 0.74$, $F1 = 0.68$.

Many alternative statistical learners can be used on the feature vectors created by kLog. In our experiments, we used a standard implementation of support vector machines [26], which was integrated via a wrapper in kLog, together with a linear kernel. We performed 5-fold cross-validation on the dataset with fixed folds. The cost c of the SVM was chosen via internal 5-fold cross-validation on the training set, for each split.

Comparison to Baselines. Our aim is not to compete with strong detectors using dense features, but to evaluate how structure and contextual knowledge can be flexibly exploited in our problem. We show that even if we start from sparse cues, the detection problem is solvable with good results thanks to the use of relational representations and kLog's flexible language and kernel. One baseline is the feature boosting approach with template matching [27]. We train an ensemble of weak detectors for the class *house*. 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 street scenes greatly vary in texture across the dataset. A second baseline is our relational distance-based approach (RD) [5]. It uses the same sparse features and data splits. Figure 4 shows results for comparison. The baselines perform well for our detection problem, however, by incorporating more structural context, kLog improves results. Also, in [5] we employed an extra candidate selection step, which resulted in higher precision. This step is not performed in the experiments with kLog.

6 Conclusions

We presented a new statistical relational learning approach to hierarchically understand images of houses. To this end, we employ kLog, a framework for logical and relational learning with kernels. The declarative, relational representation used by kLog allows a flexible exploitation of the structural and contextual knowledge in visual scenes. We show that even if we start from sparse cues, our problem is solvable with good results thanks to the use of relational representations and kLog's flexible language and kernel. This work explores a new relational scheme for solving computer vision problems. This result can be improved using a collective classification setting, in which target predictions are also considered during training and testing. Additionally, hierarchical features could be used as top-down feedback. For example, a detected house can constraint the number of doors composing the house, and thus, improve door detection results.

Acknowledgements. Laura Antanas is supported by the grant agreement First-MM-248258.

References

1. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: A survey. Foundations and Trends in Computer Graphics and Vision 3(3), 177–280 (2007)
2. Hanson, A., Riseman, E.: Visions: A computer system for interpreting scenes. In: CVS, pp. 303–333 (1978)
3. De Raedt, L.: Logical and Relational Learning. Springer (2008)
4. Fu, K.: Syntactic methods in pattern recognition, vol. 112. Elsevier Science (1974)
5. Antanas, L., van Otterlo, M., Tuytelaars, T., Raedt, L.D., Oramas Mogrovejo, J.: A relational distance-based framework for hierarchical image understanding. In: ICPRAM, vol. (2), pp. 206–218 (2012)
6. Pearce, A.R., Caelli, T., Bischof, W.F.: Learning relational structures: Applications in computer vision. Applied Intelligence 4, 257–268 (1994)
7. Getoor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of relational structure. In: ICML, pp. 170–177 (2001)
8. Frasconi, P., Costa, F., Raedt, L.D., Grave, K.D.: klog: A language for logical and relational learning with kernels. CoRR (2012)
9. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE TPAMI 32(9), 1627–1645 (2010)
10. Fergus, R., Perona, P., Zisserman, A.: Weakly supervised scale-invariant learning of models for visual recognition. IJCV 71(3), 273–303 (2007)
11. Han, F., Zhu, S.: Bottom-up/top-down image parsing with attribute grammar. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(1), 59–73 (2009)
12. Zhu, L., Chen, Y., Lin, Y., Lin, C., Yuille, A.: Recursive segmentation and recognition templates for image parsing. IEEE TPAMI 34(2), 359–371 (2012)
13. Girshick, R., Felzenszwalb, P., McAllester, D.: Object detection with grammar models. IEEE TPAMI 33(12) (2011)
14. Zhu, S.C., Mumford, D.: A stochastic grammar of images. Found. Trends. Comput. Graph. Vis. 2(4), 259–362 (2006)
15. Hartz, J.: Learning probabilistic structure graphs for classification and detection of object structures. In: ICMLA, pp. 5–11 (2009)

16. Zhao, P., Fang, T., Xiao, J., Zhang, H., Zhao, Q., Quan, L.: Rectilinear parsing of architecture in urban environment. In: CVPR, pp. 342–349 (2010)
17. Koutsourakis, P., Simon, L., Teboul, O., Tziritas, G., Paragios, N.: Single view reconstruction using shape grammars for urban environments. In: ICCV, pp. 1795–1802 (2009)
18. Terzic, K., Hotz, L., Sochman, J.: Interpreting structures in man-made scenes - combining low-level and high-level structure sources. In: ICAART, pp. 357–364 (2010)
19. Tuytelaars, T., Fritz, M., Saenko, K., Darrell, T.: The nbnn kernel. In: ICCV, pp. 1824–1831 (2011)
20. Antanas, L., Frasconi, P., Tuytelaars, T., De Raedt, L.: Employing relational languages for image understanding. In: IEEE Workshop on Kernels and Distances for Computer Vision, pp. 1–2 (2011)
21. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. TPAMI, 36–51 (2008)
22. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
23. Garcia-Molina, H., Ullman, J.D., Widom, J.: Database Systems: The Complete Book, 2nd edn. Prentice Hall Press, Upper Saddle River (2008)
24. Costa, F., Grave, K.D.: Fast neighborhood subgraph pairwise distance kernel. In: ICML, pp. 255–262 (2010)
25. Haussler, D.: Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California at Santa Cruz (1999)
26. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874 (2008)
27. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing features: Efficient boosting procedures for multiclass object detection. In: CVPR, pp. 762–769 (2004)