Int J Comput Vis (2011) 93: 141–161
DOI 10.1007/s11263-010-0343-9

Predicate Logic Based Image Grammars for Complex Pattern Recognition

Vinay Shet · Maneesh Singh · Claus Bahlmann · Visvanathan Ramesh · Jan Neumann · Larry Davis

Received: 15 October 2009 / Accepted: 13 April 2010 / Published online: 28 September 2010

Abstract Predicate logic based reasoning approaches provide a means of formally specifying domain knowledge and manipulating symbolic information to explicitly reason about different concepts of interest. Extension of traditional binary predicate logics with the bilattice formalism permits the handling of uncertainty in reasoning, thereby facilitating their application to computer vision problems. In this paper, we propose using first order predicate logics, extended with a bilattice based uncertainty handling formalism, as a means of formally encoding pattern grammars, to parse a set of image features, and detect the presence of different patterns of interest. Detections from low level feature detectors are treated as logical facts and, in conjunction with logical rules, used to drive the reasoning. Positive and negative information from different sources, as well as uncertainties from detections, are integrated within the bilattice framework. We show that this approach can also generate proofs or justifications (in the form of parse trees) for each hypothesis it proposes thus permitting direct analysis of the final solution in linguistic form. Automated logical rule weight learning is an important aspect of the application of such systems in the computer vision domain. We propose a rule weight optimization method which casts the instantiated inference tree as a knowledge-based neural network, interprets rule uncertainties as link weights in the network, and applies a constrained, back-propagation algorithm to converge upon a set of rule weights that give optimal performance within the bilattice framework. Finally, we evaluate the proposed predicate logic based pattern grammar formulation via application to the problems of (a) detecting the presence of humans under partial occlusions and (b) detecting large complex man made structures as viewed in satellite imagery. We also evaluate the optimization approach on real as well as simulated data and show favorable results.

Keywords Stochastic image grammars · Logical reasoning · Human detection · Object detection and classification · Bilattice · Back propagation · Aerial image analysis

Application and experimental validation of the reasoning framework on aerial images has been funded by US Government contract # NBCHC080029. Aerial images provided by DigiGlobe.

J. Neumann contributed to the work presented in this paper while he was affiliated with Siemens Corporate Research.

V. Shet (✉) · M. Singh · C. Bahlmann · V. Ramesh
Siemens Corporate Research, a division of Siemens Corporation, 755 College Road East, Princeton, NJ 08540, USA
e-mail: vinay.shet@siemens.com

M. Singh
e-mail: maneesh.singh@siemens.com

C. Bahlmann
e-mail: claus.bahlmann@siemens.com

V. Ramesh
e-mail: visvanathan.ramesh@siemens.com

J. Neumann
Streamsage/Comcast, 1110 Vermont Ave NW, Washington, DC 20005, USA
e-mail: jan_neumann@cable.comcast.com

L. Davis
University of Maryland, Department of Computer Science, College Park, MD 20742, USA
e-mail: lsd@umiacs.umd.edu

1 Introduction

Reliably detecting patterns in visual data has been the primary goal of computer vision research for several years.

Such patterns could be strictly spatial in nature, like static images of pedestrians, bicycles, airplanes etc., or they could be spatio-temporal in nature, like patterns of human or vehicular activity over time. Complex patterns tend to be compositional and hierarchical—a human can be thought to be composed of head, torso and limbs—a head to be composed of hair and face—a face to be composed of eyes, nose, mouth. Such patterns also tend to be challenging to detect, robustly as a whole, due to high degree of variability in shape, appearance, partial occlusions, articulation, and image noise among other factors. While the computer vision community has made significant headway in designing fairly robust low level, local feature detectors, such feature detectors only serve to detect parts of the larger pattern to be detected. Combining the detections of parts into a context sensitive, constraint satisfying set of pattern hypotheses is a non-trivial task. The key questions we need to answer are how to represent knowledge of what the pattern looks like in a hierarchical, compositional manner and how this knowledge can be exploited to effectively search for the presence of the patterns of interest?

Predicate logic based reasoning approaches provide a means of formally specifying domain knowledge and manipulating symbolic information to explicitly reason about the presence of different patterns of interest. Such logic programs help easily model hierarchical, compositional patterns to combine contextual information with the detection of low level parts via conjunctions, disjunctions and different kinds of negations. First order predicate logic separates out the name, property or type of a logical construct from its associated parameters and further, via the use of existential and universal quantifiers, allows for enumeration over its parameters.

This provides for a powerful language that can be used to specify pattern grammars to parse a set of image features to detect the presence of the pattern of interest. Such pattern grammars encode constraints about the presence/absence of predefined parts in the image, geometric relations over their parameters, interactions between these parts and scene context, and search for solutions best satisfying those constraints. Additionally, it is straightforward to generate proofs or justifications, in the form of parse trees, for the final solution thus permitting direct analysis of the final solution in a linguistic form.

While formal reasoning approaches have long been used in automated theorem proving, constraint satisfaction and computational artificial intelligence, historically, their use in the field of computer vision has remained limited. In addition to the ability to specify constraints and search for patterns satisfying those constraints, it is important to evaluate the quality of the solution as a function of the observation and model uncertainty. One of the primary inhibiting factors to a successful integration of computer vision and first order predicate logic has been the design of an appropriate interface between the binary-valued logic and probabilistic vision output. Bilattices, algebraic structures introduced by Ginsberg (1988), provide a means to design exactly such an interface to model uncertainties for logical reasoning.

Unlike traditional logics, predicate logics extended using the bilattice-based uncertainty handling formalism, associate uncertainties with both logical rules (denoting degree of confidence in domain knowledge) and observed logical facts (denoting degree of confidence in observation). These uncertainties are taken from, and semantically interpreted within, a set structured as a bilattice. Modeling uncertainties in the bilattice facilitates independent representation of both positive and negative constraints about a proposition and furthermore provides tolerance for contradictory data inherent in many real-world applications. Performing inference in such a framework is also, typically, computationally efficient.

The predicate logic based approach extended using the bilattice formalism can therefore be used to encode pattern grammars to detect whether or not a specific pattern exists in an image, where in the image the pattern exists (via instantiated parameters of the predicates), why the system thinks the pattern exists (via proofs) and finally how strongly it thinks the pattern exists (final inferred uncertainty). Due to these characteristics, bilattice based logical reasoning frameworks appear to be promising candidates for use in time-sensitive, resource-bound, computer vision applications. In our previous work (Shet et al. 2006, 2007), we have shown the applicability of such a formalism in computer vision problems such as activity recognition, identity maintenance and human detection. Arieli et al. (2006) have applied such frameworks in machine learning for preference modeling applications. Theoretical aspects of these frameworks have been studied by Arieli et al. (2005), Ginsberg (1988), Fitting (1990)

1.1 Application Domain

Detecting specific object patterns in static images is a difficult problem. This difficulty arises due to wide variability in appearance of the pattern, possible articulation, deformation, view point changes, illumination conditions, shadows and reflections, among other factors. While detectors can be trained to handle some of these variations and detect object patterns individually, as a whole, their performance degrades significantly when the pattern visually deviates from this predefined template. While such deviations can potentially be caused by all the variations listed above, the two most significant causes of such deviations are (a) partial occlusions of the pattern, by other patterns either of the same or different class, and (b) pattern deformations, either due to object articulation, or in case of man made objects due to different functional requirements.

Part based detectors are typically better suited to handle such deviations in such patterns because they can, in principle, be used to detect local object components, which can then be assembled together to detect the whole object. However, the process of going from a set of component detections to a set of scene consistent, context sensitive, pattern hypotheses is far from trivial. Since part based detectors only learn part of the information from the whole pattern, they are typically less reliable and tend to generate large numbers of false positives. Occlusions and local image noise characteristics also lead to missed detections. It is therefore important to not only exploit contextual, scene geometry and specific object constraints to weed out false positives, but also be able to explain as many valid missing object parts as possible to correctly detect all patterns.

In this paper we explore two object classes (a) pedestrians, viewed in a surveillance setting, potentially under partial occlusions and (b) large, complex, deformable, man made structures as viewed in aerial satellite images.

Consider Fig. 1. It shows a number of humans that are occluded by the scene boundary as well as by each other. Ideally, a human detection system should be able to reason about whether a hypothesis is a human or not by aggregating information provided by different sources, both visual and non-visual. For example, in Fig. 1, the system should reason that it is likely that individual 1 is human because two independent sources, the head detector and the torso detector report that it is a human. The absence of legs indicates it is possibly not a human, however this absence can be justified due to their occlusion by the image boundary. Furthermore, hypothesis 1 is consistent with the scene geometry and lies on the ground plane. Since the evidence for it being human exceeds evidence against, the system should decide that it is indeed a human. Similar reasoning applies to individual 4, only its legs are occluded by human 2. Evidence against A and B (inconsistent with scene geometry and not on the ground plane respectively) exceeds evidence in favor of them being human and therefore A and B should be rejected as being valid hypotheses.

Figure 2, shows examples of a large man made object as viewed from a satellite. These objects, surface to air missile (SAM) sites, are highly variable in shape and generally very hard to discern from background clutter. However, the key signatures of these objects include the functional arrangement of its constituent missile launchers, contextual information such as the geographical and topological structure of its neighboring regions and the general arrangement of physical access structures around it. We need to be able to capture this information and encode it as constraints that support or refute the given hypothesis.

1.2 Overview

This paper proposes a predicate logic based approach that reasons and detects object patterns in the manner outlined above. In this framework, knowledge about contextual cues, scene geometry and object pattern constraints is encoded in the form of rules in a logic programming language and applied to the output of low level component, parts based detectors. Positive and negative information from different rules, as well as uncertainties from detections are integrated

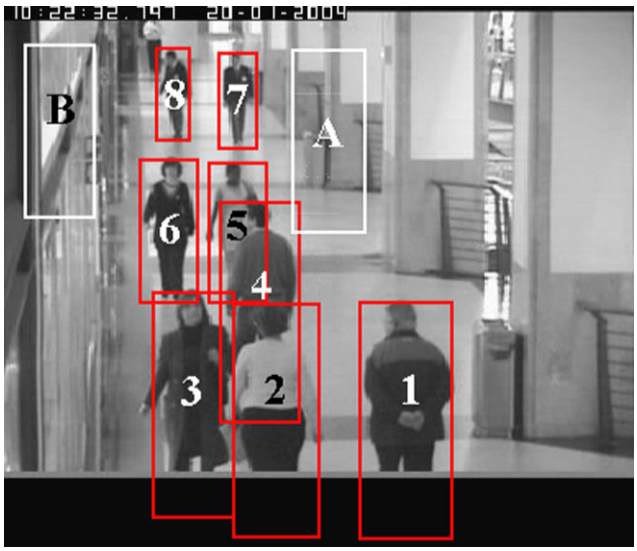


Fig. 1 Figure showing valid human detections and a few representative false positives

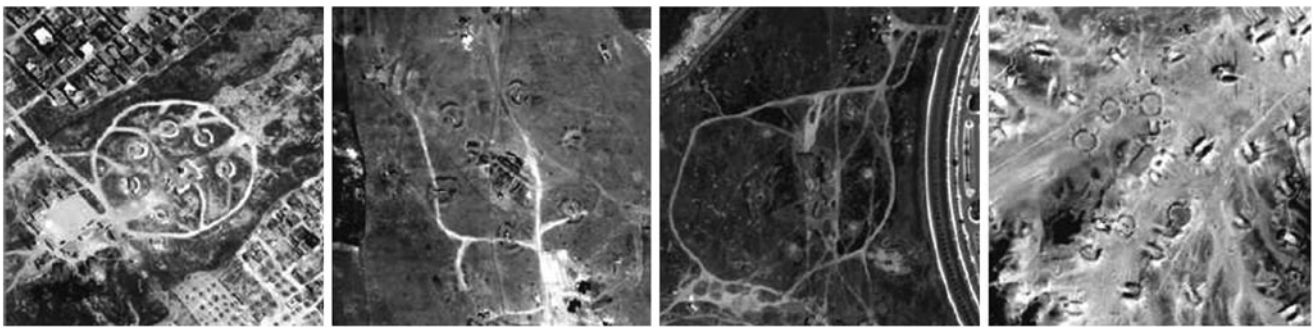


Fig. 2 Figure showing some examples of surface to air missile (SAM) sites in aerial imagery

within an uncertainty handling formalism known as the bilattice framework. This approach can also generate proofs or justifications for each hypothesis it proposes. These justifications (or lack thereof) are further employed by the system to explain and validate, or reject potential hypotheses. This allows the system to explicitly reason about complex interactions between object patterns and handle occlusions, deformations and other variabilities. Proofs generated by this approach are also available to the end user as an explanation of why the system thinks a particular hypothesis is actually a pattern of interest.

The rest of the paper is organized as follows: we first review past work on pattern grammar formalisms and statistical relational learning in Sect. 2. We then describe the use of predicate logic based pattern grammars to the problem of detecting complex object patterns in static images. We further motivate and describe the use of the bilattice framework to handle uncertainties inherent in such pattern detection problems (Sect. 3). We then discuss two applications of this framework: (a) detection of partially occluded humans in static images and (b) detection of man made objects in aerial imagery (Sect. 4). We evaluate the human detection system on the 'USC pedestrian set B' (Wu and Nevatia 2005), USC's subset of the CAVIAR dataset (CAVIAR 2003) (This dataset will henceforth be referred to in this paper as the USC-CAVIAR dataset). We also evaluate it on a dataset we collected on our own. In this paper, we refer to this dataset as Dataset-A. We evaluate the aerial object detection system on a specific type of man made object—surface to air missile site (Sect. 5). Automatically optimizing parameters associated with the specified knowledge base is an important aspect of such a system. In Sect. 6, we describe an approach that interprets the instantiated proof tree as a knowledge based artificial neural network and performs backpropagation based rule weight optimization. We report results of the learning methodology on the problem of human detection on real and simulated data (Sect. 7). We conclude in Sect. 8.

The bilattice based logical reasoning framework along with its application to the problem of human detection has been previously published in Shet et al. (2007). This paper extends the work reported in Shet et al. (2007) by introducing a rule weight optimization approach and further by applying the reasoning framework to complex spatial objects. A short summary of this paper also appears in Shet et al. (2009) as an extended abstract. Part of Shet et al. (2007), including some of the results reported, are being reproduced in this paper for a more self contained presentation.

2 Background

Computer vision approaches can broadly be characterized by the amount of model knowledge they exploit. Model free approaches assume no prior knowledge about the structure of the world and attempt to characterize patterns directly from data. These approaches typically utilize statistical (often discriminative) classifiers whose parameters are optimized via different learning methods. Support vector machines (Vapnik 1995), boosting (Schapire and Singer 1999), artificial neural networks (Rumelhart et al. 1986; LeCun et al. 1998a; Hinton et al. 2006), or regularization networks (Poggio and Girosi 1990), are examples of such approaches. In computer vision, some of these approaches operate by performing classification directly on image pixels (LeCun et al. 1998a), while others perform classification on extracted feature vectors (Viola and Jones 2001; Csurka et al. 2004). Such approaches typically require large volumes of training data to adequately capture all pattern variations.

Model based approaches on the other hand exploit some form of knowledge of the world to compactly describe the pattern of interest. Since the model already captures what the pattern and, to some extent, its variations should look like, the amount of data required for optimization is typically less than that for model free approaches. Models are often formalized in a generative Bayesian paradigm, based on/motivated from physical or geometric principles and are represented by associated probability distributions.

In this work we will primarily focus on model based approaches. In general, there exist three aspects of model based approaches one needs to consider: (1) Knowledge representation, (2) Learning, and (3) Inference. In the knowledge representation step, different variables that influence the final decision are identified and dependencies between these variables are made explicit. The learning step next, numerically captures the nature of the dependencies between different variables and how they influence each other. Finally, in the inference step, real world observations and their likelihoods are combined with the knowledge structure as well as the learnt relationships between variables to arrive at a likelihood of the final decision. Variations in the type of knowledge representation, methodology/approximations of inference, and the type of learning approach give rise to different flavors of such approaches (Binford and Levitt 2003; Mann 1995; Ponce et al. 1989; Ramesh 1995; Zhu and Mumford 2006).

In the remainder of this section, we will review stochastic image grammars, a class of model based approaches that attempt to encode a grammar for visual patterns. We then review statistical relational learning based approaches that combine first order logic with probabilistic reasoning. Finally, we contrast the proposed predicate logic based pattern grammar formulation with these two classes of approaches.

2.1 Stochastic Image Grammars

Stochastic image grammar based approaches formally model visual patterns as compositions of their constituent parts. Based on this model, these approaches then attempt to parse an image (or its extracted features) to detect the presence of that pattern of interest. Due to the nature of compositionality in images, such models typically tend to be hierarchical (i.e., trees, or DAGs in case of shared parts), with each level capturing a representation of the pattern at a particular granularity. Typical challenges associated with such approaches are (1) the formulation of the pattern grammar syntax, (2) learning of these pattern grammars from data (both structure and parameters), and (3) inference.

Various grammar approaches have been proposed in recent literature that attempt to tackle different aspects of the challenges described above. Zhu and Mumford (2006), for instance, use an AND-OR graph representation that models objects as a hierarchy of conjunctions and disjunctions of parts along with spatial and functional relations between nodes. In order to account for computational cost, they employ data driven probabilistic sampling methods to perform inference.

Fidler and Leonardis (2007) propose a framework for learning hierarchical, compositional representation of multiple class objects. They demonstrate that due to the deep hierarchical nature, several intermediate nodes in the tree get shared across multiple object classes. The growth in size of the hierarchy, and hence computational complexity, is highly sub-linear as the number of modeled classes is increased. This is one of the primary advantages of hierarchical approaches.

Todorovic and Ahuja (2008) also address the issue of designing models that share intermediate nodes across multiple object classes. In the context of categories that share certain parts, they aim at learning the underlying part taxonomy, relevances, likelihoods, and priors of those parts. They propose an inference approach where recognition is achieved by maximizing the match of the query sample with the taxonomy.

Jin and Geman (2006) propose a "composition machine" approach, which performs depth-first search on a restricted representation and corrects its results using re-ranking. Zhu et al. (2008) propose a recursive compositional model to represent shape and visual appearance of objects at different scales. Coarse-to-fine modeling is exploited by Kokkinos and Yuille (2009). Here, the authors exploit the hierarchical object representation to efficiently compute a coarse solution which is then used to guide search at a finer level.

Wang et al. (2006) propose the concept of "spatial random trees" (SRT) as an instance of an image grammar. SRTs provide polynomial-complexity exact inference algorithms, and come with maximum-a-posteriori estimation of both the tree structure and the tree states given an image. The concept of hierarchical compositionality for image grammars has been exploited by several other researchers, as well (Geman and Johnson 2003; Lin et al. 2007a; Tu and Zhu 2002).

Knowledge representations explored within the stochastic image grammar community have been primarily geared toward capturing hierarchical, compositional, models pertaining to visual patterns. The machine learning community, on the other hand, has focused on designing knowledge representation frameworks for general AI tasks. One such class of approaches viz. statistical relational learning combines a logic based representation with probabilistic reasoning and inference mechanisms, and thus is of high relevance to this paper.

2.2 Statistical Relational Learning

Statistical relational learning (SRL) (Kersting and De Raedt 2001; Cussens 1999; Friedman et al. 1999; Sato and Kameya 1997; Taskar et al. 2002) approaches model world knowledge using a first order logic. This allows SRL approaches to specify statistics over a set of relations as opposed to between a set of ground entities. Knowledge based model construction (Kersting and De Raedt 2001) for instance is a combination of logic programming and Bayesian networks. The logic program specifies a template for the pattern, which when instantiated with ground observations generates a Bayesian network. Stochastic logic programs (Cussens 1999) are a combination of logic programming and log-linear models and are a special case of knowledge based model construction. Probabilistic relational models (Friedman et al. 1999) combine frame based systems and Bayesian networks. One of the applications of SRL based approaches in the computer vision domain is documented by Fern (2005) where logical constraints are used to infer hidden states of relational processes. This is applied to classifying specific events in video sequences.

It is typical in SRL based approaches to employ a constrained subset of full first order predicate logic, called Horn clauses. Horn clauses are clauses with at most one positive literal. The reason this constrained language is sufficient for these approaches is because first order logical rules only serve as a template to instantiate the structure of the propositional graphical model (Markov network, Bayesian network). The distributions over the variables of these graphical models are typically estimated and maintained external to the graphical model. It is in these conditional distributions that the specific nature of influence between different variables of the graphical model is captured.

2.3 Contrast to Proposed Approach

Similar to the image grammar approaches reviewed above, the proposed predicate logic based approach attempts to

parse object patterns by modeling and specifying pattern grammars. This grammar specification is encoded as rules in a first order logic programming language and parsing of object pattern corresponds to searching through the feature space for solutions that best satisfy these logical constraints. In contrast to the statistical relational learning based approaches, the specific nature of influence between different variables is not captured externally in conditional probability tables, but rather, directly (and weakly) encoded in the rule specification itself. Finally, the use of the bilattice formalism permits exploitation of the full expressive power of first order predicate logical language via the use of existential and universal quantifiers, conjunctions, disjunctions, definite negations, negations by default etc.

Bayesian systems assume completeness of the world model. The proposed framework relaxes such assumptions. This incompleteness of information requires explicit handing of inconsistency (along the "degree of information" axis). The practical benefit that arises out of this is the ease of model specification and training to learn the model (Lesser complexity implies lesser training data) and similarly less complex (and faster) inference.

3 Reasoning Framework

Logic programming systems employ two kinds of formulae, facts and rules, to perform logical inference. Rules are of the form "$A \leftarrow A_0, A_1, \ldots, A_m$" where each A_i is called an atom and ',' represents logical conjunction. Each atom is of the form $p(t_1, t_2, \ldots, t_n)$, where t_i is a term, and p is a predicate symbol of arity n. Terms could either be variables (denoted by upper case letters) or constant symbols (denoted by lower case letters). The left hand side of the rule is referred to as the head and the right hand side is the body. Rules are interpreted as "if body then head". Facts are logical rules of the form "$A \leftarrow$" (henceforth denoted by just "A") and correspond to the input to the inference process. Finally, '$\neg$' represents negation such that $A = \neg\neg A$.

3.1 Logic Based Reasoning

To perform the kind of reasoning outlined in Sect. 1.1, one has to specify rules that allow the system to take input from the low level detectors and explicitly infer whether or not there exists a specific pattern at a particular location. For instance, for the human detection problem, if we were to employ a head, torso and legs detector, then a possible rule would be:

$$\begin{aligned} human(X, Y, S) \quad \longleftarrow \quad & head(X_h, Y_h, S_h), \\ & torso(X_t, Y_t, S_t), \\ & legs(X_l, Y_l, S_l), \\ & geometry\_constraint(X_h, Y_h, S_h, X_t, Y_t, S_t, X_l, Y_l, S_l), \\ & compute\_center(X_h, Y_h, S_h, X_t, Y_t, S_t, X_l, Y_l, S_l, X, Y, S). \end{aligned}$$

This rule captures the information that if the head, torso and legs detectors were to independently report a detection at some location and scale (by asserting facts $head(X_h, Y_h, S_h)$, $torso(X_t, Y_t, S_t)$, $legs(X_l, Y_l, S_l)$ respectively), and these coordinates respected certain geometric constraints, then one could conclude that there exists a human at that location and scale. A logic programming system would search the input facts to find all combinations that satisfy the rule and report the presence of humans at those locations. Note that this rule will only detect humans that are visible in their entirety. Similar rules can be specified for situations when one or more of the detections are missing due to occlusions or other reasons. There are, however, some problems with a system built on such rule specifications:

1. Traditional logics treat such rules as binary valued and definite, meaning that every time the body of the rule is true, the head of the rule will have to be true. For a real world system, we need to be able to assign some uncertainty values to the rules that capture its reliability.

2. Traditional logics treat facts as binary. We would like to take as input, along with the detection, the uncertainty of the detection and integrate it into the reasoning framework.

3. Traditional logic programming has no support for explicit negation in the head. There is no easy way of specifying a rule like:

$$\neg human(X, Y, S) \leftarrow \neg scene\_consistent(X, Y, S)$$

and integrating it with positive evidence. Such a rule says a hypothesis is not human if it is inconsistent with scene geometry.

4. Such a system will not be scalable. We would have to specify one rule for every situation we foresee. If we would like to include in our reasoning the output from another detector, say a hair detector to detect the presence of hair and consequently a head, we would have to re-engineer all our rules to account for new situations. We would like a framework that allows us to directly include new information without much re-engineering.

5. Finally, traditional logic programming does not have support for integration of evidence from multiple sources, nor is it able to handle data that is contradictory in nature.

3.2 Bilattice Theory

Over the last several decades, in the symbolic AI community, several different approaches have been introduced that handle different aspects of the limitations discussed above. Bilattices are algebraic structures introduced by Ginsberg (1988) as a uniform framework within which a number of

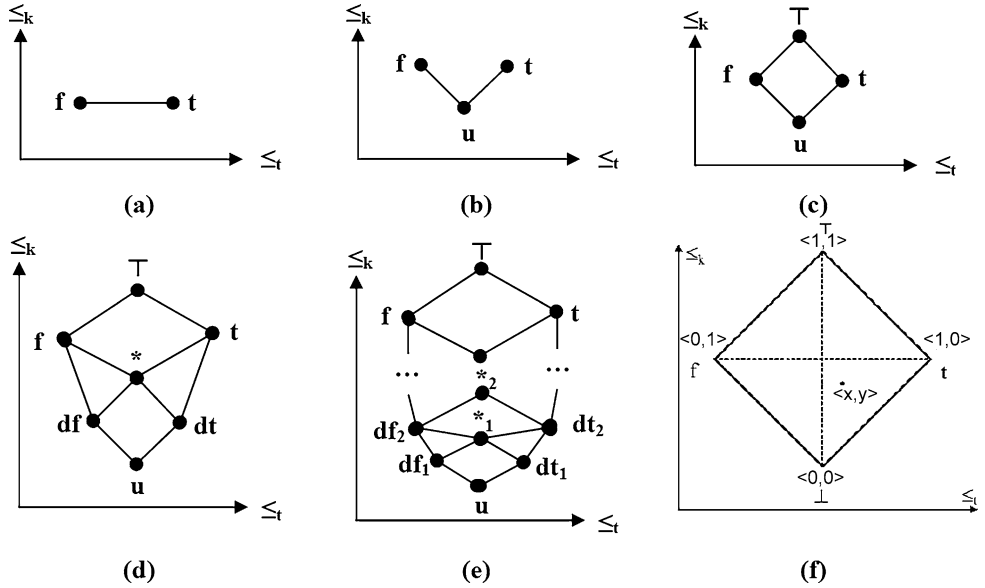


Fig. 3 The choice of different lattices that compose the bilattice gives rise to different logics. (**a**) Bilattice for two valued logics (trivial bilattice) with only `true` and `false` nodes, (**b**) bilattice for three valued logic with additional node for `unknown`, (**c**) bilattice for four valued logics with additional node for `contradiction`, (**d**) bilattice for default logics (Ginsberg 1988; Shet et al. 2006) where `dt`, `df` and `*` represent true by default, false by default and contradiction between dt-df respectively, (**e**) bilattice for prioritized default logics (Ginsberg 1988; Shet et al. 2006) with different levels of defaults and (**e**) bilattice for continuous valued logic (Arieli et al. 2005; Shet et al. 2007) as described in this paper

these approaches can be modeled. Ginsberg used the bilattice formalism to model first order logic, assumption based truth maintenance systems, and formal systems such as default logics and circumscription. Figure 3 shows examples of different bilattices and the types of logic they can be used to model. Figure 3(a) for instance models classical two valued logic, Fig. 3(b) models three valued logics, Fig. 3(c) models Belnap's four valued logics, Fig. 3(d) and (e) model traditional and prioritized default logics, and Fig. 3(f) models continuous valued logics.

In our application, the reasoning system is looked upon as a passive rational agent capable of reasoning under uncertainty. Uncertainties assigned to the rules that guide reasoning, as well as detection uncertainties reported by the low level detectors, are taken from a set structured as a bilattice. These uncertainty measures are ordered along two axes, one along the source's[1] degree of information and the other along the agent's degree of belief. As we will see, this structure allows us to address all of the issues raised in the previous section and provides a uniform framework which not only permits us to encode multiple rules for the same proposition, but also allows inference in the presence of contradictory information from different sources.

[1] A single rule applied to its set of corresponding facts is referred to as a source here. There can be multiple rules deriving the same proposition (both positive and negative forms of it) and therefore we have multiple sources of information.

All of the bilattices shown in Fig. 3, are generated via differing choices of their constituent lattices. While bilattices depicted in Fig. 3(a), (d) and (e) have been employed to address certain problems in computer vision (Shet et al. 2005, 2006), in this paper we focus on the continuous valued logic as modeled by the bilattice shown in Fig. 3(f).

Definition 1 (Lattice) A lattice is a set L equipped with a partial ordering $\leq$ over its elements, a greatest lower bound (glb) and a lowest upper bound (lub) and is denoted as $\mathscr{L} = (L, \leq)$ where glb and lub are operations from $L \times L \rightarrow L$ that are idempotent, commutative and associative. Such a lattice is said to be *complete*, *iff* for every nonempty subset M of L, there exists a unique lub and glb.

Definition 2 (Bilattice (Ginsberg 1988)) A bilattice is a triple $\mathscr{B} = (B, \leq_t, \leq_k)$, where B is a nonempty set containing at least two elements and $(B, \leq_t)$, $(B, \leq_k)$ are complete lattices.

Informally a bilattice is a set, B, of uncertainty measures composed of two complete lattices $(B, \leq_t)$ and $(B, \leq_k)$ each of which is associated with a partial order $\leq_t$ and $\leq_k$ respectively. The $\leq_t$ partial order (agent's degree of belief) indicates how true or false a particular value is, with f being the minimal and t being the maximal while the $\leq_k$ partial order indicates how much is known about a particular proposition. The minimal element here is $\perp$ (completely unknown) while

the maximal element is $\top$ (representing a contradictory state of knowledge where a proposition is both true and false). The glb and the lub operators on the $\leq_t$ partial order are $\wedge$ and $\vee$ and correspond to the usual logical notions of conjunction and disjunction, respectively. The glb and the lub operators on the $\leq_k$ partial order are $\otimes$ and $\oplus$, respectively, where $\oplus$ corresponds to the combination of evidence from different sources or lines of reasoning while $\otimes$ corresponds to the consensus operator. A bilattice is also equipped with a negation operator $\neg$ that inverts the sense of the $\leq_t$ partial order while leaving the $\leq_k$ partial order intact and a conflation operator – which inverts the sense of the $\leq_k$ partial order while leaving the $\leq_t$ partial order intact.

The intuition is that every piece of knowledge, be it a rule or an observation from the real world, provides different degrees of information. An agent that has to reason about the state of the world based on this input, will have to translate the source's degree of information, to its own degree of belief. Ideally, the more information a source provides, the more strongly an agent is likely to believe it (i.e. closer to the extremities of the t-axis). The only exception to this rule being the case of contradictory information. When two sources contradict each other, it will cause the agent's degree of belief to decrease despite the increase in information content. It is this decoupling of the sources and the ability of the agent to reason independently along the truth axis that helps us address the issues raised in the previous section. It is important to note that the line joining $\bot$ and $\top$ represents the line of indifference. If the final uncertainty value associated with a hypothesis lies along this line, it means that the `degree of belief for` and `degree of belief against` it cancel each other out and the agent cannot say whether the hypothesis is true or false. Ideally the final uncertainty values should be either f or t, but noise in observation as well as less than completely reliable rules ensure that this is almost never the case. The horizontal line joining t and f is the line of consistency. For any point along this line, the `degree of belief for` will be exactly equal to (`1-degree of belief against`) and thus the final answer will be exactly consistent.

Definition 3 (Rectangular Bilattice (Fitting 1990; Arieli et al. 2006)) Let $\mathscr{L} = (L, \leq_L)$ and $\mathscr{R} = (R, \leq_R)$ be two complete lattices. A rectangular bilattice is a structure $\mathscr{L} \odot \mathscr{R} = (L \times R, \leq_t, \leq_k)$, where for every $x_1, x_2 \in \mathscr{L}$ and $y_1, y_2 \in \mathscr{R}$,

1. $\langle x_1, y_1\rangle \leq_t \langle x_2, y_2\rangle \Leftrightarrow x_1 \leq_L x_2$ and $y_1 \geq_R y_2$,
2. $\langle x_1, y_1\rangle \leq_k \langle x_2, y_2\rangle \Leftrightarrow x_1 \leq_L x_2$ and $y_1 \leq_R y_2$

An element $\langle x_1, y_1\rangle$ of the rectangular bilattice $\mathscr{L} \odot \mathscr{R}$ may be interpreted such that x_1 represents the amount of belief for some assertion while y_1 represents the amount of belief against it. If we denote the glb and lub operations of

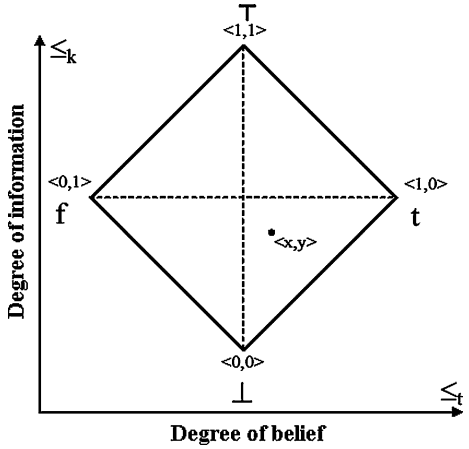


Fig. 4 The bilattice square $\mathscr{B} = ([0, 1]^2, \leq_t, \leq_k)$. Every element of this bilattice is of the form $\langle evidence\_for, evidence\_against\rangle$

complete lattices $\mathscr{L} = (L, \leq_L)$, and $\mathscr{R} = (R, \leq_R)$ by $\wedge_L$ and $\vee_L$, and $\wedge_R$ and $\vee_R$ respectively, we can define the glb and lub operations along each axis of the bilattice $\mathscr{L} \odot \mathscr{R}$ as follows (Arieli et al. 2006; Fitting 1990):

$$\begin{aligned}
\langle x_1, y_1\rangle \wedge \langle x_2, y_2\rangle &= \langle x_1 \wedge_L x_2, y_1 \vee_R y_2\rangle,\\
\langle x_1, y_1\rangle \vee \langle x_2, y_2\rangle &= \langle x_1 \vee_L x_2, y_1 \wedge_R y_2\rangle,\\
\langle x_1, y_1\rangle \otimes \langle x_2, y_2\rangle &= \langle x_1 \wedge_L x_2, y_1 \wedge_R y_2\rangle,\\
\langle x_1, y_1\rangle \oplus \langle x_2, y_2\rangle &= \langle x_1 \vee_L x_2, y_1 \vee_R y_2\rangle
\end{aligned} \tag{1}$$

Of interest to us in our application is a particular class of rectangular bilattices where $\mathscr{L}$ and $\mathscr{R}$ coincide. These structures are called *squares* (Arieli et al. 2005) and $\mathscr{L} \odot \mathscr{L}$ is abbreviated as $\mathscr{L}^2$. Since detection likelihoods reported by the low level detectors are typically normalized to lie in the [0, 1] interval, the underlying lattice that we are interested in is $\mathscr{L} = ([0, 1], \leq)$.[2] The bilattice that is formed by $\mathscr{L}^2$ is depicted in Fig. 4. Each element in this bilattice is a tuple with the first element encoding evidence for a proposition and the second encoding evidence against. In this bilattice, the element f (false) is denoted by the element $\langle 0, 1\rangle$ indicating, no evidence for but full evidence against, similarly element t is denoted by $\langle 1, 0\rangle$, element $\bot$ by $\langle 0, 0\rangle$ indicating no information at all and $\top$ is denoted by $\langle 1, 1\rangle$. To fully define glb and lub operators along both the axes of the bilattice as listed in (1), we need to define the glb and lub operators for the underlying lattice $([0, 1], \leq)$. A popular choice for such operators are triangular-norms and triangular-conorms. Triangular norms and conorms were introduced by Schweizer and Sklar (1963) to model the distances in probabilistic metric spaces and are considered to be

[2]Note that with this choice of the lattice, $\leq$ becomes a complete ordering, meaning all members of the lattice are comparable. Definition 3 therefore needs to be modified such that $\langle x_1, y_1\rangle \leq_t \langle x_2, y_2\rangle \Leftrightarrow x_1 - y_1 \leq x_2 - y_2$ and $\langle x_1, y_1\rangle \leq_k \langle x_2, y_2\rangle \Leftrightarrow x_1 + y_1 \leq x_2 + y_2$.

generalizations of classical two valued operators. Triangular norms are used to model the glb operator and the triangular conorm to model the lub operator within each lattice.

Definition 4 (Triangular norm) A mapping

$$\mathscr{T} : [0,1] \times [0,1] \to [0,1]$$

is a triangular norm (t-norm) iff $\mathscr{T}$ satisfies the following properties:

- Symmetry: $\mathscr{T}(a,b) = \mathscr{T}(b,a), \forall a,b \in [0,1]$.
- Associativity: $\mathscr{T}(a,\mathscr{T}(b,c)) = \mathscr{T}(\mathscr{T}(a,b),c), \forall a,b,c \in [0,1]$.
- Monotonicity: $\mathscr{T}(a,b) \leq \mathscr{T}(a',b') if\, a \leq a'$ and $b \leq b'$.
- One identity: $\mathscr{T}(a,1) = a, \forall a \in [0,1]$.

Definition 5 (Triangular conorm) A mapping

$$\mathscr{S} : [0,1] \times [0,1] \to [0,1]$$

is a triangular conorm (t-conorm) iff $\mathscr{S}$ satisfies the following properties:

- Symmetry: $\mathscr{S}(a,b) = \mathscr{S}(b,a), \forall a,b \in [0,1]$.
- Associativity: $\mathscr{S}(a,\mathscr{S}(b,c)) = \mathscr{S}(\mathscr{S}(a,b),c), \forall a,b,c \in [0,1]$.
- Monotonicity: $\mathscr{S}(a,b) \leq \mathscr{S}(a',b')$ if $a \leq a'$ and $b \leq b'$.
- Zero identity: $\mathscr{S}(a,0) = a, \forall a \in [0,1]$.

If $\mathscr{T}$ is a t-norm, then the equality $\mathscr{S}(a,b) = 1 - \mathscr{T}(1-a, 1-b)$ defines a t-conorm and we say $\mathscr{S}$ is derived from $\mathscr{T}$. There are number of possible t-norms and t-conorms one can choose. In our application, for the underlying lattice, $\mathscr{L} = ([0,1], \leq)$, we choose the t-norm such that $\mathscr{T}(a,b) \equiv a \wedge_L b = ab$ and consequently choose the t-conorm as $\mathscr{S}(a,b) \equiv a \vee_L b = a + b - ab$. Based on this, the glb and lub operators for each axis of the bilattice B can then be defined as per (1).

3.3 Inference

Inference in bilattice based reasoning frameworks is performed by computing the closure over the truth assignment.

Definition 6 (Truth Assignment) Given a declarative language L, a truth assignment is a function $\phi : L \to B$ where B is a bilattice on truth values or uncertainty measures.

Definition 7 (Closure) Let $\mathscr{K}$ be the knowledge base and ϕ be a truth assignment, labeling every formula $k \in \mathscr{K}$, then the closure over $k \in \mathscr{K}$, denoted $cl(\phi)$ is the truth assignment that labels information entailed by $\mathscr{K}$.

For example, if ϕ labels sentences $\{p, q \leftarrow p\} \in \mathscr{K}$ as $\langle 1,0\rangle$ (true); i.e. $\phi(p) = \langle 1,0\rangle$ and $\phi(q \leftarrow p) = \langle 1,0\rangle$, then $cl(\phi)$ should also label q as $\langle 1,0\rangle$ as it is information entailed by $\mathscr{K}$. Entailment is denoted by the symbol '$\models$' ($\mathscr{K} \models q$).

Let $S_q^+ \subset L$ be the collection of minimal subsets of sentences in $\mathscr{K}$ entailing q. For each minimal subset $U \in S_q^+$, the uncertainty measure to be assigned to the conjunction of elements of U is

$$\bigwedge_{p \in U} cl(\phi)(p) \tag{2}$$

This term represents the conjunction of the closure of the elements of U.[3] It is important to note that this term is merely a contribution to the final uncertainty measure of q and not the final uncertainty measure itself. The reason it is merely a contribution is because there could be other sets of sentences in S_q^+ that entail q representing different lines of reasoning (or, in our case, different rules and supporting facts). The contributions of these sets of sentences need to be combined using the $\oplus$ operator along the information ($\leq_k$) axis. Also, if the expression in (2) evaluates to false, then its contribution to the value of q should be $\langle 0,0\rangle$ (unknown) and not $\langle 0,1\rangle$ (false). These arguments suggest that the closure over ϕ of q is

$$cl(\phi)(q) = \bigoplus_{U \in S_q^+} \bot \vee \left[\bigwedge_{p \in U} cl(\phi)(p) \right] \tag{3}$$

where $\bot$ is $\langle 0,0\rangle$. This is however, only part of the information. We also need to take into account the sets of sentences entailing $\neg q$. Let S_q^- be collections of minimal subsets in $\mathscr{K}$ entailing $\neg q$. Aggregating information from S_q^- yields the following expression

$$cl(\phi)(q) = \bigoplus_{U \in S_q^+} \bot \vee \left[\bigwedge_{p \in U} cl(\phi)(p) \right] \oplus \neg \bigoplus_{U \in S_q^-} \bot \vee \left[\bigwedge_{p \in U} cl(\phi)(p) \right]. \tag{4}$$

For more details see Ginsberg (1988).

Table 1 shows an example, using a simplified logic program, illustrating the process of computing the closure as defined above by combining evidence from three sources. In this example, the final uncertainty value computed is $\langle 0.4944, 0.72\rangle$. This indicates that evidence against the hypothesis at $(25, 95)$ at scale 0.9 exceeds evidence in favor of and, depending on the final threshold for detection, this hypothesis is likely to be rejected.

[3] Recall that $\wedge$ and $\vee$ are glb and lub operators along the $\leq_t$ ordering and $\otimes$ and $\oplus$ along $\leq_k$ axis. The symbols $\bigwedge, \bigvee, \bigotimes, \bigoplus$ are their infinitary counterparts such that $\bigoplus_{p \in S} p = p_1 \oplus p_2 \oplus \cdots$, and so on.

Table 1 Example showing inference using closure within a $([0,1]^2, \leq_t, \leq_k)$ bilattice for a simplified set of rules for the human detection problem

Assume the following set of rules and facts:	
Rules	Facts
$\phi(human(X,Y,S) \leftarrow head(X,Y,S)) = \langle 0.40, 0.60\rangle$	$\phi(head(25,95,0.9)) = \langle 0.90, 0.10\rangle$
$\phi(human(X,Y,S) \leftarrow torso(X,Y,S)) = \langle 0.30, 0.70\rangle$	$\phi(torso(25,95,0.9)) = \langle 0.70, 0.30\rangle$
$\phi(\neg human(X,Y,S) \leftarrow \neg scene\_consistent(X,Y,S)) = \langle 0.90, 0.10\rangle$	$\phi(\neg scene\_consistent(25,95,0.9)) = \langle 0.80, 0.20\rangle$

Inference is performed as follows:

$$cl(\phi)(human(25,95,0.9)) = \langle 0,0\rangle \vee [\langle 0.4,0.6\rangle \wedge \langle 0.9,0.1\rangle] \oplus \langle 0,0\rangle \vee [\langle 0.3,0.7\rangle \wedge \langle 0.7,0.3\rangle] \oplus \neg(\langle 0,0\rangle \vee [\langle 0.9,0.1\rangle \wedge \langle 0.8,0.2\rangle])$$
$$= \langle 0.36,0\rangle \oplus \langle 0.21,0\rangle \oplus \neg\langle 0.72,0\rangle = \langle 0.4944,0\rangle \oplus \langle 0,0.72\rangle = \langle 0.4944,0.72\rangle$$

3.4 Negation

Systems such as this typically employ different kinds of negation. One kind of negation that has already been mentioned earlier is $\neg$. This negation flips the bilattice along the $\leq_t$ axis while leaving the ordering along the $\leq_k$ axis unchanged. Another important kind of negation is negation by failure to prove, denoted by *not*. $not(A)$ succeeds if A fails. This operator flips the bilattice along both the $\leq_t$ axis as well as the $\leq_k$ axis. Recall that, in Sect. 3.2, '$-$' was defined as the conflation operator that flips the bilattice along the $\leq_k$ axis. Therefore, $\phi(not(A)) = \neg - \phi(A)$. In other words, if A evaluates to $\langle 0,0\rangle$, then $not(A)$ will evaluate to $\langle 1,1\rangle$. This operator is important when we want to detect the absence of a particular object part for a hypothesis.

3.5 Generating Proofs

As mentioned earlier, in addition to using the explanatory ability of logical rules, we can also provide these explanations to the user as justification of why the system believes that a given hypothesis is a pattern of interest. The system provides a straightforward technique to generate proofs from its inference tree. Since all of the bilattice based reasoning is encoded as meta-logical rules in a logic programming language, it is easy to add predicates that succeed when the rule fires and propagate character strings through the inference tree up to the root where they are aggregated and displayed. Such proofs can either be dumps of the logic program itself or be English text. In our implementation, we output the logic program as the proof tree.

4 Pattern Grammars

We can now use this framework to define a knowledge base to detect different patterns of interest. We begin by defining a number of predicates and their associated parameters pertinent to the problem at hand. For instance, for the human detection problem, we can define atoms such as $human(X,Y,S)$,[4] $head(X,Y,S)$, $torso(X,Y,S)$ etc. We also define relational and geometric predicates such as $above(X_1,Y_1,S_1,X_2,Y_2,S_2)$, $smaller(X_1,Y_1,S_1,X_2,Y_2,S_2)$, $sceneconsistent(X,Y,S)$.[5]

The next step involves specification of the pattern grammar, as logical rules, over these defined atoms. Such rules would capture different aspects of the pattern to be recognized such as those shown in Fig. 5. Rules in such systems can be learnt automatically; however, such approaches are typically computationally very expensive. We manually encode the rules while automatically learning the uncertainties associated with them as described in Sect. 6.

A desirable property of any reasoning framework is scalability. We may expect scalability in vision systems as different objects or pattern classes are hierarchically composed of constituent patterns that share features like textures, edges etc. and as objects inhabit the same optical world and are imaged by similar optical sensors. We see scalability as a design principle wherein the model description is modular, hierarchical and compositional, reflecting the above understanding of the world. The proposed framework results in scalable systems if models are appropriately described as such.

With this goal in mind, we lay out the following design principle for object pattern grammar specification. We partition rule specification into three broad categories: object composition model based, object embodiment model based and object context model based.

Composition model Rules encoding these models capture a hierarchical representation of the object pattern as a composition of its constituent part detections. These parts might by themselves be composed of sub-parts. Rules in this category try to support or refute the presence of a pattern based on the presence or absence of its constituent parts.

[4] Meaning there exists a human at location (X,Y) and scale S in the image.

[5] Meaning the hypothesis at (X,Y) and scale S is consistent with the scene geometry and conforms, within bounds, to the expected size of an object at the location.

$$
\begin{aligned}
\phi(human(X,Y,S) &\leftarrow head(X,Y,S),\\
&\quad sceneconsistent(X,Y,S)) = \langle \alpha_1, \beta_1 \rangle\\
\phi(human(X,Y,S) &\leftarrow not(torso(X,Y,S),\\
&\quad torso\_occluded(X,Y,S,X_o,Y_o,S_o),\\
&\quad human(X_o,Y_o,S_o),\\
&\quad Y_o > Y) = \langle \alpha_2, \beta_2 \rangle.\\
\phi(\neg human(X,Y,S) &\leftarrow not(onground plane(X,Y,S))) = \langle \alpha_3, \beta_3 \rangle\\
\phi(\neg human(X,Y,S) &\leftarrow not(head(X,Y,S))) = \langle \alpha_4, \beta_4 \rangle
\end{aligned}
$$

Fig. 5 A sample subset of rules for human detection

Embodiment model These rules model knowledge about the object pattern's geometric layout and their embodiment in 3D projective spaces.

Context model These rules attempt to model the surrounding context within which the pattern of interest is embedded. These rules would for example model interactions between a given object and other objects or other scene structures.

As mentioned above, such an object oriented organization of the knowledge representation derives from an implicit understanding of our physical world as composed of objects. Specification and conceptual layering of rules in this manner induces a natural hierarchy in such a pattern specification. By enforcing that the specified rules are well structured, categorized into the above categories and follow general principles of composability, we ensure the scalability of our system.

It is important to note that there would typically exist multiple rules that derive the same proposition. These multiple rules are interpreted in logic programming as disjunctions (i.e. rule 1 is true or rule 2 is true etc.). Writing rules in this manner makes each rule independently 'vote' for the proposition to be inferred. This disjunctive specification results in a scalable solution where the absence of a single observation does not completely preempt the final output, but merely reduces its final confidence value. As can be seen from the subset of rules in Fig. 5, the inference tree formed would be comprised of conjunctions, disjunctions and different kinds of negations.

4.1 Human Detection

Human detection in images is a hard problem. There are a number of approaches in computer vision literature that detect humans both as whole as well as a collection of parts. Leibe et al. (2005) employs an iterative method combining local and global cues via a probabilistic segmentation, Gavrila (2000), Gavrila and Philomin (1999) uses edge templates to recognize full body patterns, Papageorgiou et al. (1998) uses SVM detectors, and Felzenszwalb (2001) uses shape models. A popular detector used in such systems is a cascade of detectors trained using AdaBoost as proposed by Viola and Jones (2001). Dalal and Triggs (2005) use an SVM based classifier based on the histogram of oriented gradients. This was further extended by Zhu et al. (2006) to detect whole humans using a cascade of histograms of oriented gradients. Part based representations have also been used to detect humans. Wu and Nevatia (2005) use edgelet features and learn nested cascade detectors for each of several body parts and detect the whole human using an iterative probabilistic formulation.

Our human body part detectors are inspired by Zhu et al. (2006). Similar to their approach we train a cascade of SVM-classifiers on histograms of gradient orientations. Instead of the hard threshold function suggested in their paper, we apply a sigmoid function to the output of each SVM. These softly thresholded functions are combined using a boosting algorithm (Freund and Schapire 1997). After each boosting round, we calibrate the probability of the partial classifier based on an evaluation set, and set cascade decision thresholds based on the sequential likelihood ratio test similar to Sochman and Matas (2005). To train the parts-based detector, we restrict the location of the windows used during the feature computation to the areas corresponding to the different body parts (head/shoulder, torso, legs).

The pattern grammar for the human detection problem is formulated as per the broad categories listed in the previous section. Component based rules hypothesize that a human is present at a particular location if one or more of the body part detectors described above detects a body part there. In other words, if a head is detected at some location, we say there exists a human there. There are positive rules, one each for the head, torso, legs and full-body based detectors as well as negative rules that fire in the absence of these detections.

Geometry based rules validate or reject human hypotheses based on geometric and scene information. This information is entered a priori in the system at setup time. We employ information about expected height of people and regions of expected foot location. The expected image height rule is based on ground plane information and anthropometry. Fixing a Gaussian at an adult human's expected physical height allows us to generate scene consistency likelihoods for a particular hypothesis given its location and size. The expected foot location region is a region demarcated in the image outside of which no valid feet can occur and therefore serves to eliminate false positives.

Context based rules are the most important rules for a system that has to handle occlusions. The idea here is that if the system does not detect a particular body part, then it must be able to explain its absence for the hypothesis to be considered valid. If it fails to explain a missing body part, then it is construed as evidence against the hypothesis being a human. Absence of body parts is detected using logic programming's 'negation as failure' operator (*not*). $not(A)$ succeeds when A evaluates to $\langle 0, 0 \rangle$ as described in Sect. 3.4.

A valid explanation for missing body part could either be due to occlusions by static objects or due to occlusions by other humans.

Explaining missed detections due to occlusions by static objects is straightforward. At setup, all static occlusions are marked. Image boundaries are also treated as occlusions and marked as shown in Fig. 1 (black area at the bottom of the figure). For a given hypothesis, the fraction of overlap of the missing body part with the static occlusion is computed and reported as the uncertainty of occlusion. The process is similar for occlusions by other human hypotheses, with the only difference being that, in addition to the degree of occlusion, we also take into account the degree of confidence of the hypothesis that is responsible for the occlusion, as illustrated in the second rule in Fig. 5.

This rule will check to see if $human(X, Y, S)$'s torso is occluded by $human(X_o, Y_o, S_o)$ under condition that $Y_o > Y$, meaning the occluded human is behind the 'occluder'. It is important to note that this would induce a scene geometry constrained, hierarchy in the parse graph, since whether or not a given hypothesis is a human depends on whether or not a hypothesis in front of it was inferred as being a valid pattern of interest. There exist similar rules for other components and also rules deriving $\neg human$ in the absence of explanations for missing parts.

4.2 Aerial Object Detection

Typical objects of interest in aerial images are buildings, roads, factories, rivers, harbors, airfields, golf courses, etc. We focus on the detection of surface-to-air missile (SAM) sites. The two primary classes of features we employ are geometric and contextual. Geometric features extracted are straight lines, circles, corners etc. In case of SAM sites, the primary discriminating feature is typically the arrangement of individual missile launchers that compose the SAM site. Circle feature detectors can be used to detect individual launchers as shown in Fig. 6 (a) while, line features can help detect neighboring structures such as a road network.

For contextual features, we attempt to discriminate terrain textures in aerial scenes, such as, "Forest", "Desert", "Road", "Urban", "Maritime", and "Agricultural" on a coarse level. Terrain textures, such as oceans, forest, urban, agricultural areas, contain repetitions of fundamental microstructures such as waves, trees, houses and streets, agricultural produce, respectively. Such configurations have been studied in literature as *texture* (with a *texton* being the micro structure) and identified as a significant feature for perception and identification, both in psychophysics (Julesz 1981) and computer vision (Leung and Malik 2001). Walker and Malik (2004) report that texture provides a strong cue for the identification of natural scenes in the human visual system. Our context features are inspired by well developed texture classification techniques; see Leung and Malik (2001),

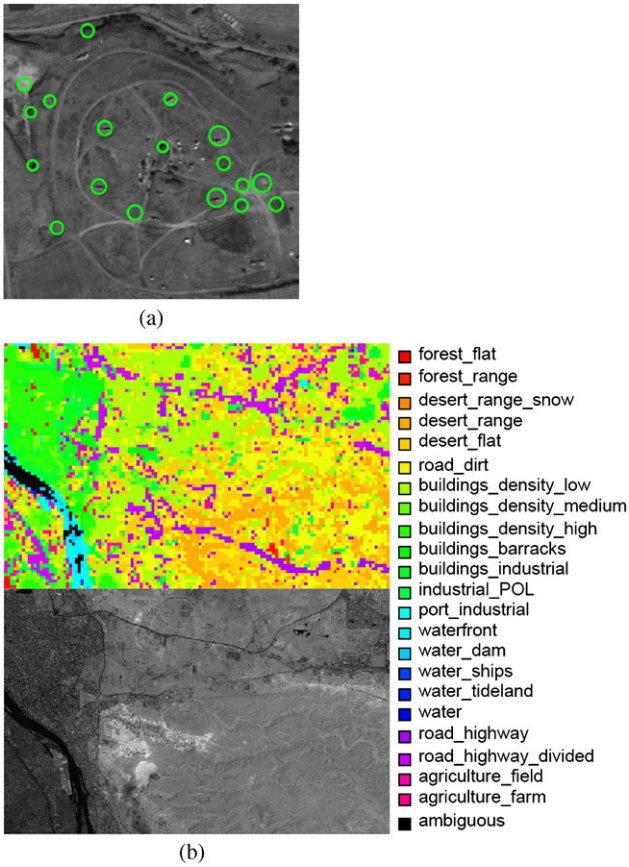


Fig. 6 Sample features detected in aerial images

Varma and Zisserman (2005). Figure 6(b) shows different regions of the image color labeled with the corresponding detected textures. The spatial, geometric and contextual constraints that need to be satisfied for an object to be classified as a SAM site are encoded as logical rules, again broadly falling in the categories listed above.

4.3 Implementation Details

A predicate logic based reasoning framework can be efficiently implemented in a logic programming language like Prolog. Distributions of Prolog like SWI-Prolog, allow for the straightforward integration of C++ with an embedded Prolog reasoning engine. Predefined rules can be inserted into the Prolog engine's knowledge base at set up time by the C++ module, along with information about scene geometry and other constraints. At runtime, the C++ module can apply the detectors on the given image, preprocess the feature detector output if needed, syntactically structure this output as logical facts, and finally insert it into the Prolog knowledge base. These detections then serve as initial hypotheses upon which the query can be performed. Since rules contain unbounded variables and observed facts contain constants as parameters, querying for a proposition in Prolog implies

finding a suitable binding of the rule variables to the constants of the supporting facts. If no such binding is found, the corresponding rule does not fire.

It is important to note that complexity of general inference in predicate logics can be combinatorial. In practice, however, variable interdependencies between different atoms of a rule restrict the search space significantly. Specifically, in the pattern grammar formulation described in this paper, there exists significant reuse of the variables between atoms both within and across different rules. Additionally, Prolog can be set up to index facts based on specific variables further reducing complexity of variable binding.

5 Evaluation I

In this section, we first describe some qualitative results on the human detection problem using the USC-CAVIAR dataset and show how our system reasons and resolves difficult scenarios. We subsequently present quantitative results on the USC-CAVIAR dataset as well as on problem of detecting SAM sites from aerial imagery. Please note that for both the problems, we obtain rule-weights using the positive predictive value (PPV) approach as described in Sect. 6.1.

5.1 Human Detection

We have evaluated below the performance of the bilattice based logical reasoning approach on the problem of human detection on static images.

5.1.1 Qualitative Results

Table 2 lists the proof for Human 4 from Fig. 1. For Human 4, the head and torso are visible while the legs are missing due to occlusion by human 2. In Table 2, variables starting with $\_G\cdots$ are non-unified variables in Prolog, meaning that legs cannot be found and therefore the variables of the predicate legs cannot be instantiated. It can be seen that evidence in favor of the hypothesis exceeds that against.

Figure 7(a), shows a sample image from the USC_CAVIAR dataset and shows the detection results overlaid. Figure 7(b) plots the uncertainty value for each hypothesis point in the bilattice space. The red circles on the right are the accepted detections and correspond to the bounding boxes in (a), while the gray circles in the left half of the bilattice are hypotheses rejected by the reasoning framework (not displayed in (a)).

5.1.2 Quantitative Results

We applied our framework to the set of static images taken from USC-CAVIAR (Wu and Nevatia 2005) dataset. This dataset, a subset of the original CAVIAR (CAVIAR 2003) data, contains 54 frames with 271 humans of which 75 humans are partially occluded by other humans and 18 humans

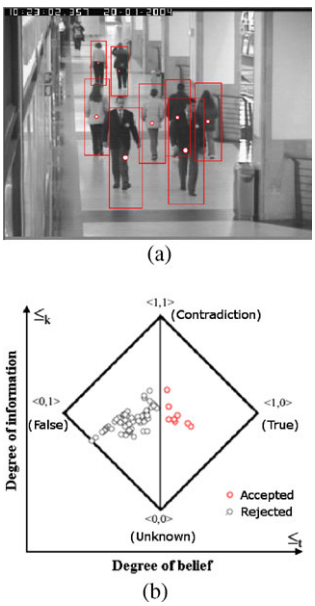


Fig. 7 (**a**) Figure showing a sample image from the USC_CAVIAR dataset with detection results overlaid and (**b**) Computed uncertainty value (for all human hypotheses in left image) plotted in the bilattice space

Table 2 Proof for human marked as '4' in Fig. 1

Total:	human(154,177,1.25)	$\langle 0.359727, 0.103261\rangle$
+ve evidence:	head(154, 177, 1.25)	$\langle 0.94481, 0.05519\rangle$
	torso(156.25, 178.75, 1.25)	$\langle 0.97871, 0.02129\rangle$
	on_ground_plane(154, 177, 1.25)	$\langle 1, 0\rangle$
	scene_consistent(154, 177, 1.25)	$\langle 0.999339, 0.000661\rangle$
	not((legs(_G7093,_G7094, _G7095), legs_body_consistent(154, 177, 1.25,_G7093, _G7094, _G7095)))	$\langle 1, 1\rangle$
	is_part_occluded(134.0, 177.0, 174.0, 237.0)	$\langle 0.260579, 0.739421\rangle$
−ve evidence:	¬scene_consistent(154, 177, 1.25)	$\langle 0.000661, 0.999339\rangle$
	not((legs(_G7260, _G7261, _G7262), legs_body_consistent(154, 177, 1.25,_G7260, _G7261, _ G7262)))	$\langle 1, 1\rangle$

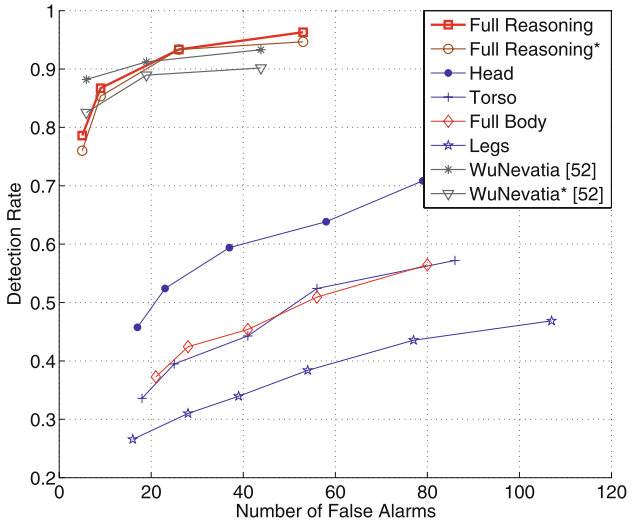


Fig. 8 ROC curves for evaluation on the USC-CAVIAR dataset. *Full Reasoning** is ROC curve for 75 humans occluded by other humans. Results of Wu and Nevatia (2005) on the same dataset are copied from their original paper. *WuNevatia** is ROC curve for the 75 humans occluded by other humans

Fig. 9 An image from Dataset-A

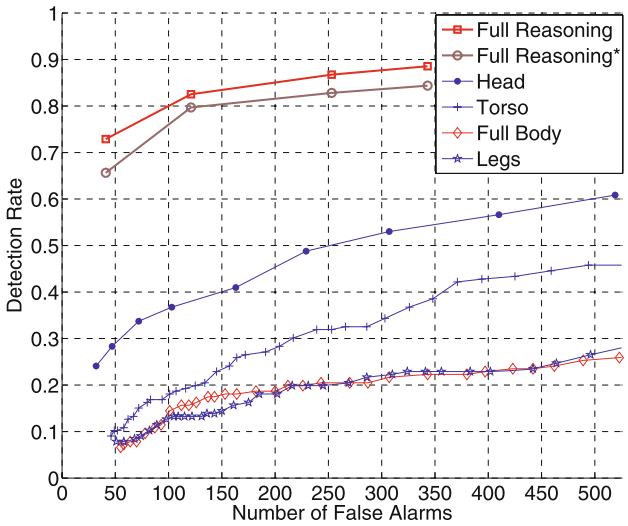


Fig. 10 ROC curves for evaluation on Dataset-A. *Full Reasoning** is ROC curve for 126 occluded humans

are occluded by the scene boundary. This data is not part of our training set. We have trained our parts based detector on the MIT pedestrian dataset (Papageorgiou et al. 1998). For training purposes, the size of the human was 32×96 centered and embedded within an image of size 64×128. We used 924 positive images and 6384 negative images for training. The number of layers used in full-body, head, torso and leg detectors were 12, 20, 20, and 7 respectively. Figure 8 shows the ROC curves for our parts based detectors as well as for the full reasoning system. "Full Reasoning*", in Fig. 8, is the ROC curve on the 75 occluded humans. ROC curves for part based detectors represent detections that have no prior knowledge about scene geometry or other anthropometric constraints. It can be seen that performing high level reasoning over low level part based detections, especially in presence of occlusions, greatly increases overall performance. We have also compared the performance of our system with the results reported by Wu and Nevatia (2005) on the same dataset. We have taken results reported in their original paper and plotted them in Fig. 8. As can be seen, results from both systems are comparable. The results in Fig. 8 were first reported in Shet et al. (2007). Since then Lin et al. (2007b) and Wu and Nevatia (2007) published new results on this datasets that show some improvements in the overall ROC curve. All the reported results however are comparable to each other.

We also applied our framework on another set of images taken from a dataset we collected on our own (in this paper we refer to it as Dataset-A). This dataset contains 58 images (see Fig. 9) of 166 humans, walking along a corridor, 126 of whom are occluded 30% or more, 64 by the image boundary and 62 by each other. Dataset-A is significantly harder than the USC-CAVIAR dataset due to heavier occlusions (44 humans are occluded 70% or more), perspective distortions (causing humans to appear tilted), and due to the fact that many humans appear in profile view. Figure 10 shows the ROC curves for this dataset. It can be seen that the low level detectors as well as the full body detector perform worse here than on the USC-CAVIAR data, however, even in such a case, the proposed logical reasoning approach gives a big improvement in performance.

5.2 Aerial Object Detection

We have evaluated the bilattice based logical reasoning approach on the problem of detecting SAM sites in aerial imagery. As can be seen from Fig. 2, these objects are highly variable in shape and are hard to detect even for humans.

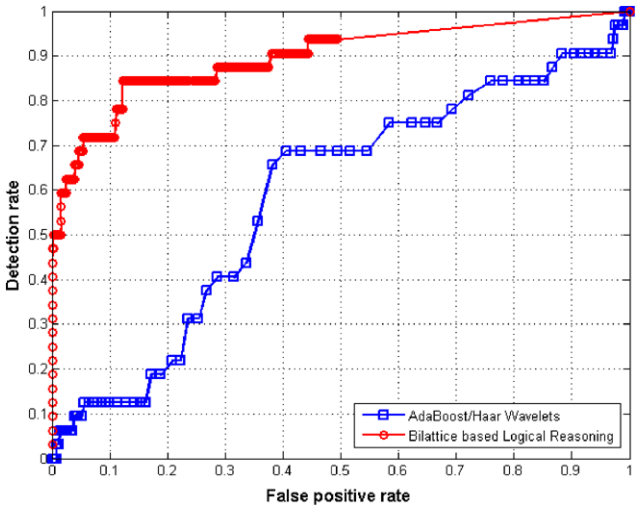


Fig. 11 ROC Curves for SAM site detection problem

However, the defining characteristic of such an object is the arrangement of its constituent missile launchers which arises from functional requirements. Additionally, there are a number of contextual cues that can be exploited such as geographical and topological makeup of the neighboring regions. We created a dataset of SAM sites containing 33 positive examples and 869 negative examples sampled from a 400 Km^2 physical region surrounding the positive example. Figure 11 shows the ROC curve obtained for this data. Figure 11 also plots the ROC curve on that data for a AdaBoost with Haar wavelets approach (Viola and Jones 2001). The AdaBoost based approach was trained on a separate training set of 869 negative images and 32 positive images.

As can be seen from Fig. 11, there is a marked improvement in performance using the pattern grammar based approach over a purely data driven approach. It is important to note however, that even for relatively simple, well constrained objects a purely data driven approach such as AdaBoost would need a lot of data to adequately generalize. In datasets such SAM sites, it is usually hard to acquire the required amounts of annotated data for such an approach to effectively learn. Add to that the high variability in the shape of the object and even more data would be needed to adequately generalize. In the case of the pattern grammar based approach, since knowledge of the object structure and surrounding context is directly specified, we would expect the results to be better than any purely data driven technique.

6 Rule Weight Learning

Although theoretical aspects of bilattices and the nature of semantics they give rise to in logic programs have been extensively studied in literature (Arieli et al. 2005; Fitting 1990; Ginsberg 1988), little work exists on automated learning procedures, which are of grave importance to computer vision applications. Learning in such systems implies: (a) Learning the structure of the rules, (b) Learning rule weights. While there exists literature for learning rule structure from data, such approaches tend to be computationally prohibitive and require large amounts of data. In this paper, we assume that the rule structure is given to us and focus instead on learning and optimizing rule weights within the bilattice framework.

6.1 Positive Predictive Value Based Learning

A common technique for rule weight learning is to use the positive predictive value (PPV) of the rule as its weight. Given training data in the form of observed facts, ground truth annotations, and a rule of the form $A \leftarrow B_1, B_2, \ldots, B_n$, a confidence value of $\langle \mathscr{F}(A|B_1, B_2, \ldots, B_n), \mathscr{F}(\neg A|B_1, B_2, \ldots, B_n)\rangle$ is computed. $\mathscr{F}(A|B_1, B_2, \ldots, B_n)$ is the fraction of times A coincides with the ground truth when $B_1, B_2, \ldots, B_n$ is true. As the name suggests, this value computes a measure of the fraction of the time, a rule that has fired, is expected to be correct with respect to ground truth. This measure is learnt individually for each rule. Typically a multiplicative constant is also employed to scale down all rule weights, if needed, to prevent saturation of the final uncertainty value of the inferred proposition, when multiple rules are combined. The results reported in the previous section of this paper, Sect. 5, have been generated using rule weights learnt using the PPV.

There are a number of issues with using the rule's PPV as its weight.

(1) The PPV depends on the ground truth annotations w.r.t. inferred variables for the observed facts. Often, however, ground truth is only known for rules that infer the output node. Deeper nodes (i.e., input or hidden nodes) usually lack this information, and hence, defy PPV based weight adaptation.
(2) Joint optimization of rules is not possible. Each rule weight is learnt individually, ignoring possible support or opposition of adjacent rules.
(3) Uncertainty values of the final inferred proposition can saturate to the maximal contradictory state of the bilattice, especially when multiple rules are combined, again because each rule weight is learnt individually. To handle this typically an appropriate multiplicative constant needs to be chosen.
(4) An inherently frequentist interpretation of the rules weights may not be optimal, due to the fact that the pattern grammar formulation itself may not be complete and may contain contradictions.

6.2 Knowledge Based Artificial Neural Networks

In this section, we present a rule weight learning method that attempts to address these issues. This approach (a) casts the instantiated inference tree from the logic program as a knowledge-based neural network, (b) interprets uncertainties associated with logical rules as link weights in this neural network and (c) applies a constrained, modified back-propagation algorithm to converge upon a set of rule weights that give optimal performance. The back-propagation algorithm has been modified to allow computation of local gradients over the bilattice specific inference operation.

The issues raised above are handled in the following manner:

(1) Similar to the error back-propagation algorithm with multi-layer perceptrons, ground truth is only required for rules that infer the output variable. As will be shown, the algorithm then "back-propagates" the computed error (of ground truth versus observed activation) to the deeper nodes of the rule hierarchy.
(2) Due to the choice of t-norm and t-conorm for the bilattice and the formulation of the weight optimization as a gradient descent algorithm, optimization of individual rule weights is tempered by the contributions of adjacent rules that have fired for a given hypothesis.
(3) Further in the gradient descent formulation, it is straightforward to include a regularization term in the error expression that penalizes extremely large or extremely small rule weights thus obviating the need for an external multiplicative scaling constant.
(4) Due to the fact that the KB may be incomplete or inconsistent, a gradient descent based approach might converge upon a set of rule weights that provide a favorable performance as compared to a PPV based measure.

Traditionally, artificial neural networks (ANNs) are modeled as black boxes. Given a set of input and output variables, and training data, a network is created in which the input nodes correspond to the input variables and the output nodes correspond to the output variables. Depending on the nature of the problem to be solved and a priori assumptions, a number of nodes are introduced between the input and output nodes that are termed hidden nodes. Each link connecting two nodes is assigned a link weight. Learning in an ANN implies optimizing link weights to minimize the mean squared error between the network predicted output and ground truth, given input data. In such networks, the intermediate hidden nodes don't necessarily have to be meaningful entities.

In knowledge based ANNs (KBANN) (Towell et al. 1990; Mahoney and Mooney 1993), unlike traditional ANNs, all nodes, hidden or not, have a semantically relevant interpretation. This semantic interpretability arises out of careful construction of the KBANN. In our case, we construct the KBANN from the rule base of the logic program. Each node of the KBANN therefore directly corresponds to each instantiated atom of the logic program while links weights correspond to rules weights. Given a logic program, optimizing the rule weights thus is a two step process. Step 1 is to use the rules and facts to create a KBANN and step 2 is to use a modified version of the standard backpropagation algorithm (Rumelhart et al. 1986) to optimize the link weights of the KBANN, thus in turn optimizing the rule weights in the original logic program.

6.2.1 *Building the KBANN*

The first step in the learning algorithm is to convert the rule base to a representation of a knowledge-based artificial neural network. Consider a set of rules, such as those depicted in Fig. 5. Given a set of training data, in the form of observed logical facts and associated ground truth, the first step is to generate a grounded, propositional, representation for each of the rules. Below is one such set of propositional rule representation.

$$\begin{aligned} \phi(j \leftarrow o_{11}, o_{12}, o_{13}) &= w^{+}_{j1} \\ \phi(j \leftarrow o_{21}, o_{22}) &= w^{+}_{j2} \\ \phi(\neg j \leftarrow o_{31}, o_{32}) &= w^{-}_{j3} \end{aligned} \tag{5}$$

where each term, j, o_{11}, o_{12}, etc., represent grounded atoms such as $human(23, 47, 0.4)$, $head(43, 55, 0.9)$, etc. The weights associated with these propositional rules corresponds to the *evidence_for* component of the original rules.[6] This grounded, propositional, rules representation can now be directly used to construct the artificial neural network. In such a network, observed features (logical facts) become the input nodes, while propositions corresponding to the rule heads become output nodes and are placed at the top of the network. Rule weights become link weights in the network.

Figure 13 shows the KBANN derived from the set of grounded, propositional rules from (5). It is important to note that conjuncts within a single rule need to first pass through a conjunction node before reaching the consequent node where along with the weights they would get combined with contributions from other rules in a disjunction.

[6] Recall that for a given rule, only the evidence_for component of the uncertainty attached to the rule is relevant. The evidence_against component of the rule weight gets discarded during inference due to the disjunction with $\langle 0, 0 \rangle$ (see (4)). Given a proposition, j, to be reasoned about, positive rules will contribute evidence supporting j, while negative rules will contribute evidence refuting it. The evidence_for component of the negative rule will contribute to the evidence_against component of the proposition to be reasoned about due to the negation. Please refer to the example in Table 1 for more details.

In Fig. 13, the links connecting the conjuncts to the product node are depicted using solid lines. This indicates that this weight is unadjustable and is always set to unity. Only the weights corresponding to the links depicted in dotted lines are adjustable as they correspond to the rule weights.

6.2.2 Computing Gradients

The approach proposed in this paper is inspired by the back propagation algorithm from neural networks, specifically, knowledge based artificial neural networks (KBANN) introduced by Towell et al. (1990) and applied by Mahoney and Mooney (1993).

Consider a simple ANN as shown in Fig. 12. In traditional back propagation, the output of an output node is

$$d_j = \sigma(z_j) = \frac{2}{1+e^{-\lambda(z_j)}} - 1 \tag{6}$$

where σ is the sigmoid function and where

$$z_j = \phi(j) = \sum_i w_{ji}\sigma(\phi(o_i)) \tag{7}$$

The error at the output node is

$$E = \frac{1}{2}\sum_j (t_j - d_j)^2 \tag{8}$$

where t_j is the ground truth for node j. Based on this measure of error, the change of a particular link weight is set to be proportional to the rate of change of error with respect to that link weight. Thus

$$\Delta w_{ji} \propto -\frac{\partial E}{\partial w_{ji}} \tag{9}$$

Using standard backpropagation calculus, the change in link weight can be computed to be

$$\Delta w_{ji} = \eta\delta_j\sigma(\phi(o_j)) \tag{10}$$

where

$$\delta_j = (t_j - d_j)\frac{\partial\sigma(z_j)}{\partial z_j} \tag{11}$$

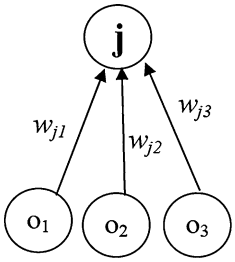
if j is an output node and

$$\delta_j = \frac{\partial\sigma(z_j)}{\partial z_j}\sum_{k\in DS(j)}\delta_k w_{kj} \tag{12}$$

if j is a non-output node, where $DS(j)$ is the set of nodes downstream from j.

We now need to extend these equations to the KBANN depicted in Fig. 13. This involves computing gradients over the bilattice specific inference operation. Recall that in the bilattice based logical reasoning approach, inference is performed by computing the closure over a logic program using (4). This equation can be simplified as

Fig. 12 A simple ANN



$$\begin{aligned} z_j &= \phi(j) \\ &= \bigoplus_i^{+ve} w_{ji}^+ \wedge \left[\bigwedge_l \phi(o_{il})\right] \oplus \neg\bigoplus_i^{-ve} w_{ji}^- \wedge \left[\bigwedge_l \phi(o_{il})\right] \end{aligned} \tag{13}$$

Note that this equation represents a general form of the closure operation before a commitment has been made on the underlying lattice structure and its corresponding glb and lub operators. Once the choice of the underlying lattice and corresponding operators has been made, in conjunction with (8), (9) and (13), it should be possible to compute the rate of change of each of the rule weights.

Consistent with Sect. 3.2, for the rest of this section, we choose the underlying lattice to be $\mathscr{L} = ([0,1], \leq)$ and choose the t-norm to be $\mathscr{T}(a,b) \equiv a \wedge_L b = ab$ and t-conorm as $\mathscr{S}(a,b) \equiv a \vee_L b = a + b - ab$. As defined in Sect. 3.2, the glb and lub operators for each axis of the bilattice B can then be defined as per (1). Plugging these operator instantiations in the (13), we can further simplify it to

$$z_j = \biguplus_i^{+ve} w_{ji}^+ \prod_l \phi(o_{il}) - \biguplus_i^{-ve} w_{ji}^- \prod_l \phi(o_{il}) \tag{14}$$

where $a \uplus b = a + b - ab$.

Note that, unlike the traditional output equation for back propagation (7), this formulation is slightly more complex due to the combination of observation nodes via the conjunction (product) node and then further combination of outputs of multiple rules via disjunction (probabilistic sum). A key point to note is that the probabilistic sum of weights, $\biguplus_i w_i$, can be easily differentiated, with respect to given weight w_k, as follows:

$$\frac{\partial \biguplus_i w_i}{\partial w_k} = 1 - \biguplus_{i\neq k} w_i \tag{15}$$

Using (14) and (15), we can compute the gradients to be

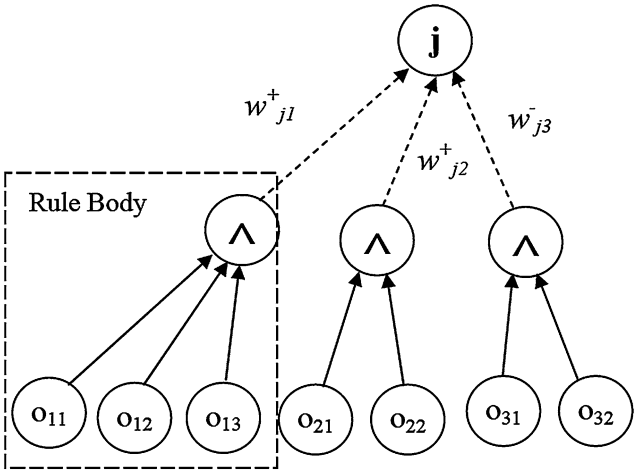


Fig. 13 Example of a knowledge based artificial neural network representing rules depicted in (5)

$$\frac{\partial E}{\partial w^+_{ji}} = -(t_j - d_j)\left[\prod_l \phi(o_{il})\right]\left[1 - \biguplus_{m\neq i} w^+_{jm} \prod_l \phi(o_{ml})\right] \tag{16}$$

$$\frac{\partial E}{\partial w^-_{ji}} = (t_j - d_j)\left[\prod_l \phi(o_{il})\right]\left[1 - \biguplus_{m\neq i} w^-_{jm} \prod_l \phi(o_{ml})\right] \tag{17}$$

We can now compute the rate of change of each rule weight as follows

$$\begin{aligned} \Delta w^+_{ji} &= \eta\delta_j\left[\prod_l \phi(o_{il})\right]\left[1 - \biguplus_{k\neq m} w^+_{jm} \prod_l \phi(o_{ml})\right] \\ \Delta w^-_{ji} &= -\eta\delta_j\left[\prod_l \phi(o_{il})\right]\left[1 - \biguplus_{k\neq m} w^-_{jm} \prod_l \phi(o_{ml})\right] \end{aligned} \tag{18}$$

where

$$\delta_j = t_j - d_j \tag{19}$$

if j is an output node and

$$\delta_j = \sum_{m\in DS(j)} \delta_m w_{mj} \prod_{l\neq j} \phi(o_{jl})\left[1 - \biguplus_{k\neq j} w_{mk} \prod_l \phi(o_{kl})\right] \tag{20}$$

if j is a non-output node, where $DS(j)$ is the set of nodes downstream from j.

Once we analytically compute the gradient there are a number of techniques we can adopt to perform the actual optimization. In this work, we choose to perform online weight update, where for each data point we computed the gradient and used it to instantaneously modify the rule weight. This is in contrast to a batch approach where the cumulative gradient of a batch of data points is used to update the weights. We believe an online approach such as the one adopted is better suited for applications with limited access to annotated data as has been suggested in LeCun et al. (1998b).

7 Evaluation II

In this section, we evaluate the proposed rule weight optimization algorithm for the bilattice based logical reasoning framework. For the purpose, we have chosen the human detection problem and not the SAM site detection problem. Our decision was influenced by the amount of available training data. There are practical difficulties in accessing a large amount of SAM site data due to its sensitive nature. This results in difficulty in setting up training, testing and validation subsets of any meaningful sizes.

Given the fact that the final accuracy of the overall framework is a function of the performance of the low level detectors in addition to how well the optimization algorithm optimizes rule weights, we attempt to isolate the performance of the optimization algorithm in two ways: (1) By fixing the rule set and performing multiple runs. For each run, the weights are randomly initialized and performance is measured both with the random initialization as well as after optimization. (2) By measuring performance improvements on simulated data. Working with simulated data allows us to model varying degrees of low level detector noise and evaluate performance of the optimization algorithm as a function of the detector noise.

7.1 Experimental Methodology

The experimental methodology we adopt is the repeated random sub-sampling based two-fold cross validation. We randomly split the data into two sets, training and testing, for the training set, we randomly initialize the rule weights, we then perform the proposed optimization with the random weights as a starting point and finally measure performance for the optimized weights on the testing dataset. To isolate the performance improvement attained by the optimization algorithm, we also measure the performance on the testing set with the initial random weights. This procedure is repeated multiple times, each time selecting different random training and testing subsets from the full dataset and each time, initializing the rule weights to different random values. Performance for a given set of rule weights is measured as the area under the ROC (AUROC) curve for the problem of human detection.

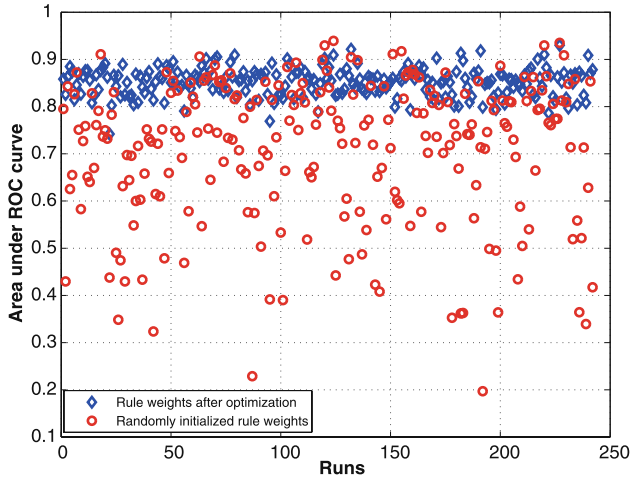


Fig. 14 Plot showing area under ROC curve for random initialization of rule weights and trained rules for multiple runs on the USC_CAVIAR dataset

Table 3 Table showing average increase in AUROC and reduction in variance after optimization over random initialization on the USC_CAVIAR

Average AUROC for randomly initialized rule weights	0.7084
Average AUROC for optimized rule weights	0.8519
% change in AUROC	20.2659 %
Variance of AUROC for randomly initialized rule weights	0.0230
Variance of AUROC for optimized rule weights	0.000903
% change in variance	−96.0695 %
AUROC for Positive Predictive value based rule weight initialization	0.8073

7.2 Pedestrian Dataset

We applied our framework to the set of static images taken from USC-CAVIAR dataset. Figure 14 displays the results of the each of the 242 randomly initialized runs on the USC_CAVIAR dataset. The red circles represent the AUROC for a random weight initialization, say $\mathbf{w}_0$, while the blue diamonds directly above the red circle represents the AUROC for the optimized rule weights $\mathbf{w}$ with $\mathbf{w}_0$ as the initial starting point.

As can be seen from both the graphs, optimizing the rule weights using the proposed approach significantly improves the overall results as well as significantly reduces the variance of the results over multiple runs as compared to a purely random initialization. This trend is numerically presented in Table 3. It can be seen that the proposed optimization approach increases the average AUROC by about 20% while reducing the average variance by 96%. We also compare in Table 3, the AUROC results for rule weight obtained in a frequentist manner by computing the positive predictive value. As can be seen, the proposed optimization approach also outperforms in this case.

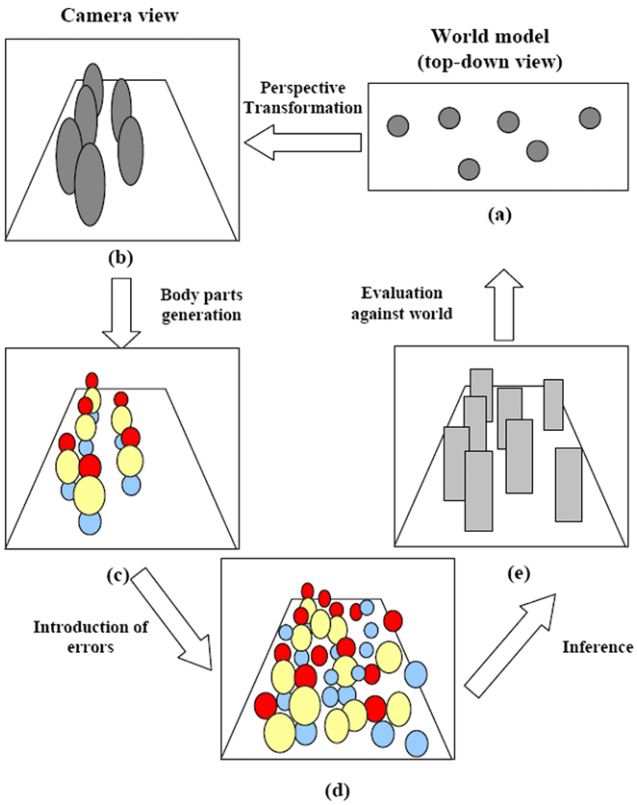


Fig. 15 Figure showing the approach adopted to generate simulated data

7.3 Simulated Data

We also evaluate the optimization algorithm on simulated data. Figure 15 depicts the approach adopted by us to generate the simulated data.

We first start by building a randomly initialized globally consistent world model of humans standing on the ground plane 15(a). We then transform this world model into camera coordinates to render the humans from a simulated camera's field of view 15(b). We then generate body part responses respecting any inter-human/human-scene occlusions 15(c). These responses represent the ideal, noise free detector response. We then introduce noise into these responses that results in the introduction of false positives and missed detections as well as a reduction in separability between the positive and negative class. The detector response is modeled using an exponential distribution conditioned to lie between [0, 1] for the negative class. For the positive class the distribution is mirrored around 0.5. This exponential distribution is characterized by parameter λ. The higher the λ the lower the false positives and missed detections and better the separability, while the converse is true for a small λ.

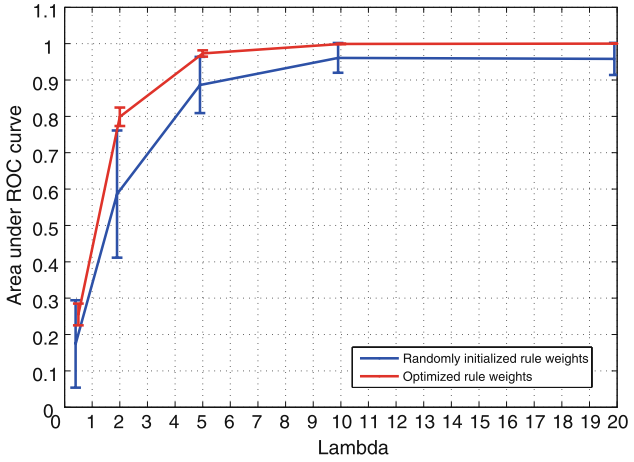


Fig. 16 Plot showing mean AUROC and associated variance on simulated data for varying values of λ

Varying λ allows us to represent a range of simulated detector performance over which we can evaluate the optimization algorithm as well as the overall bilattice based logical reasoning approach. For each λ, we executed multiple runs randomizing both over the training/testing dataset as well as initial starting rule weights. The results of each of these runs is shown in Fig. 16. As can be seen from the results, as expected as the amount of detector error increases, it gets harder to separate out the two classes and therefore overall AUROC is low. As the low level detectors are made stronger, AUROC improves significantly. In all these cases, applying the proposed rule weight optimization algorithm is clearly advantageous.

8 Conclusions

In this paper, we presented a predicate logic based reasoning approach that provides a means of formally specifying domain knowledge and manipulating symbolic information to explicitly reason about the presence of different patterns of interest. Such logic programs help easily model hierarchical, compositional patterns to combine contextual information with the detection of low level parts via conjunctions, disjunctions and different kinds of negations. First order predicate logic separates out the name, property or type of a logical construct from its associated parameters and further, via the use of existential and universal quantifiers, allows for enumeration over its parameters. This provides for a powerful language that can be used to specify pattern grammars to parse a set of image features to detect the presence of the pattern of interest. In order to admit stochastic definitions of visual patterns and to reason in the presence of uncertainty in facts (observations), we used the bilattice formalism as proposed by Ginsberg (1988). We believe that the framework presented in this paper is uniquely suited for high level reasoning in vision applications as it provides a means to (a) formally specify (stochastic) domain knowledge; (b) handle uncertainty in observations; (c) reconcile contradictory evidence (d) perform layered (hierarchical) inference; and, (d) explicitly generate justification for accepting/rejecting a pattern hypothesis.

We made several contributions in this paper: We proposed using of first order predicate logics, extended with a bilattice based uncertainty handling formalism, as a means of formally encoding pattern grammars, to parse a set of image features, and detect the presence of different patterns of interest. We then proposed a rule weight optimization method which casts the instantiated inference tree as a knowledge-based neural network, interprets rule uncertainties as link weights in the network, and applies a constrained, back-propagation algorithm to converge upon a set of rule weights that give optimal performance within the bilattice framework. Finally, we evaluated the proposed predicate logic based pattern grammar formulation via application to the problems of (a) detecting the presence of humans under partial occlusions and (b) detecting large complex man made structures as viewed in satellite imagery. We also evaluated the optimization approach on real as well as simulated data and showed favorable results.

Acknowledgements We wish to thank the reviewers of IJCV and the First International Stochastic Image Grammars workshop 2009, for their comments and notes which helped better shape this paper.

References

CAVIAR Dataset (2003). http://homepages.inf.ed.ac.uk/rbf/caviar/.

Arieli, O., Cornelis, C., & Deschrijver, G. (2006). Preference modeling by rectangular bilattices. In *Proc. 3rd international conference on modeling decisions for artificial intelligence (MDAI'06) (3885)* (pp. 22–33).

Arieli, O., Cornelis, C., Deschrijver, G., & Kerre, E. (2005). Bilattice-based squares and triangles. In *Symbolic and quantitative approaches to reasoning with uncertainty* (pp. 563–575).

Binford, T. O., & Levitt, T. S. (2003). Evidential reasoning for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(7).

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV* (pp. 1–22).

Cussens, J. (1999). Loglinear models for first-order probabilistic reasoning. In *Proceedings of the fifteenth conference on uncertainty in artificial intelligence*.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR05* (pp. I: 886–893).

Felzenszwalb, P. (2001). Learning models for object recognition. In *CVPR01* (pp. I:1056–1062).

Fern, A. (2005). A simple-transition model for structured sequences. In *International joint conference on artificial intelligence*.

Fidler, S., & Leonardis, A. (2007). Towards scalable representations of object categories: Learning a hierarchy of parts. In *Proc. IEEE conf. computer vision pattern recognition (CVPR)*.

Fitting, M. C. (1990). Bilattices in logic programming. In *20th international symposium on multiple-valued logic*, Charlotte (pp. 238–247). Los Alamitos: IEEE CS Press.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*, 119–139.

Friedman, N., Getoor, L., Koller, D., & Pfefier, A. (1999). Learning probabilistic relational models. In *Proceedings of the sixteenth international joint conference on artificial intelligence*.

Gavrila, D. (2000). Pedestrian detection from a moving vehicle. In *ECCV00* (pp. II: 37–49).

Gavrila, D., & Philomin, V. (1999). Real-time object detection for smart vehicles. In *ICCV99* (pp. 87–93).

Geman, S., & Johnson, M. (2003). Probability and statistics in computational linguistics, a brief review. In *Mathematical foundations of speech and language processing* (pp. 1–26). Berlin: Springer.

Ginsberg, M. L. (1988). Multivalued logics: Uniform approach to inference in artificial intelligence. *Computational Intelligence*.

Hinton, G. E., Osindero, S., Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*, 1527–1554.

Jin, Y., & Geman, S. (2006). Context and hierarchy in a probabilistic image model. In *CVPR* (pp. 2145–2152).

Julesz, B. (1981). Textons, the elements of texture perception and their interactions. *Nature*, *290*, 91–97.

Kersting, K., & De Raedt, L. (2001). Towards combining inductive logic programming with Bayesian networks. In *Proceedings of the eleventh international conference on inductive logic programming*.

Kokkinos, I., & Yuille, A. (2009). HOP: Hierarchical object parsing. In *Proc. IEEE conf. computer vision pattern recognition (CVPR)*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998a). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2325.

LeCun, Y., Bottou, G.O., Muller, K. (1998b). *Efficient backprop. Neural networks: Tricks of the trade*. Berlin: Springer.

Leibe, B., Seemann, E., & Schiele, B. (2005). Pedestrian detection in crowded scenes. In *CVPR05* (pp. I: 878–885).

Leung, T., & Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, *43*.

Lin, L., Peng, S., Porway, J., Zhu, S., & Wang, Y. (2007a). An empirical study of object category recognition: Sequential testing with generalized samples. In *ICCV07* (pp. 1–8).

Lin, Z., Davis, L., Doermann, D., & DeMenthon, D. (2007b). Hierarchical part-template matching for human detection and segmentation (pp. 1–8).

Mahoney, J. J., & Mooney, R. J. (1993). Combining neural and symbolic learning to revise probabilistic rule bases. In Hanson, S. J., Cowan, J. D., & Giles, C. L. (Eds.) *Advances in neural information processing systems* (Vol. 5, pp. 107–114). San Mateo: Morgan Kaufmann.

Mann, W. B. (1995). *Three dimensional object interpretation of monocular gray-scale images*. Ph.D. thesis, Department of Electrical Engineering, Stanford University.

Papageorgiou, C., Evgeniou, T., & Poggio, T. (1998). A trainable pedestrian detection system. In *Intelligent Vehicles* (pp. 241–246).

Poggio, T., & Girosi, F. (1990). Regularization algorithms that are equivalent to multilayer networks. *Science*, 978–982.

Ponce, J., Chelberg, D., & Mann, W. (1989). Invariant properties of straight homogeneous generalized cylinders and their contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(9), 951–966.

Ramesh, V. *Performance characterization of image understanding algorithms*. Ph.D. thesis, University of Washington, Seattle.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation (pp. 318–362).

Sato, T., & Kameya, Y. (1997). Prism: A symbolic statistical modeling language. In *Proceedings of the fifteenth international joint conference on artificial intelligence*.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, *37*(3), 297–336.

Schweizer, B., & Sklar, A. (1963). Associative functions and abstract semigroups. Publ. Math. Debrecen.

Shet, V., Harwood, D., & Davis, L. (2005). Vidmap: video monitoring of activity with prolog. In *IEEE AVSS* (pp. 224–229).

Shet, V., Harwood, D., & Davis, L. (2006). Multivalued default logic for identity maintenance in visual surveillance. In *ECCV* (pp. IV: 119–132).

Shet, V., Neumann, J., Ramesh, V., & Davis, L. (2007). Bilattice-based logical reasoning for human detection. In *CVPR*.

Shet, V., Singh, M., Bahlmann, C., & Ramesh, V. (2009). Predicate logics based image grammars for complex pattern recognition. In *First international workshop on stochastic image grammars*.

Sochman, J., & Matas, J. (2005). Waldboost: Learning for time constrained sequential detection. In *CVPR05* (pp. II: 150–156).

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. In *Proceedings of the eighteenth conference on uncertainty in artificial intelligence*.

Todorovic, S., & Ahuja, N. (2008). Learning subcategory relevances for category recognition. In *CVPR08*.

Towell, G. G., Shavlik, J. W., & Noordewier, M. O. (1990). Refinement of approximate domain theories by knowledge-based neural networks. In *Proceedings of the eighth national conference on artificial intelligence* (pp. 861–866).

Tu, Z., & Zhu, S. (2002). Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 657–673.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

Varma, M., & Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis*.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE conference on computer vision and pattern recognition (CVPR'01)*.

Viola, P., & Jones, M. J. (2001). *Robust real-time object detection* (Tech. Rep. CRL 2001/01). Cambridge Research Laboratory.

Walker, L. L., & Malik, J. (2004). When is scene recognition just texture recognition. *Vision Research*, *44*, 2301–2311.

Wang, W., Pollak, I., Wong, T., Bouman, C., Harper, M. P., Member, S., Member, S., & Siskind, J. M. (2006). Hierarchical stochastic image grammars for classification and segmentation. *IEEE Transactions on Image Processing*, *15*, 3033–3052.

Wu, B., & Nevatia, R. (2005). Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In *ICCV*, Beijing.

Wu, B., & Nevatia, R. (2007). Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, *75*(2), 247–266.

Zhu, L., Lin, C., Huang, H., Chen, Y., & Yuille, A. (2008). Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *Computer vision—ECCV*.

Zhu, Q., Yeh, M., Cheng, K., & Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *CVPR06* (pp. II: 1491–1498).

Zhu, S. C., & Mumford, D. (2006). A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, *2*(4), 259–362.