

Max Margin AND/OR Graph Learning for Efficient Articulated Object Parsing

Long (Leo) Zhu · Yuanhao Chen · Chenxi Lin · Alan Yuille

the date of receipt and acceptance should be inserted later

Abstract In this paper we formulate a novel AND/OR graph representation for parsing articulated objects into parts and recovering their poses. The AND/OR graph allows us to handle an enormous variety of articulated poses with a compact graphical model. We develop a novel inference algorithm, compositional inference, that uses a bottom-up compositional process for proposing configurations for the object. The strategy of surround suppression is applied to ensure that the inference time is polynomial in the size of input data. We present a novel structure-learning method, Max Margin AND/OR Graph (MM-AOG), to learn the parameters of the AND/OR graph model discriminatively. Max-margin learning is a generalization of the training algorithm for support vector machines (SVMs). The parameters are optimized globally, i.e. the weights of the appearance model for individual nodes and the relative importance of spatial relationships between nodes are learnt simultaneously. The kernel trick can be used to handle high dimensional features and to enable complex similarity measures to discriminate between object configurations. We applied our approach –

the AND/OR graph representation, compositional inference and max-margin learning – to the tasks of detecting, segmenting and parsing horses and human body. We demonstrate that the inference algorithm is fast and analyze its computational complexity empirically. To evaluate max margin learning, we perform comparison experiments on the horse and human baseball datasets, showing significant improvements over state of the art methods on benchmarked datasets.

Keywords

Hierarchy, Shape Representation, Object Parsing, Segmentation, Structure Learning, Max Margin.

1 Introduction

Most problems in machine intelligence can be formulated as probabilistic inference using probabilistic models defined on structured knowledge representations. Important examples include stochastic grammars [1] and, in particular, AND/OR graphs [2],[3],[4]. In practice, the nature of the representations is constrained by the types of inference algorithms which are available. For example, probabilistic context free grammars for natural language processing have a natural one-dimensional structure which makes it practical to use dynamic programming (DP) for inference [1]. But the complexity of vision, and its two-dimensional nature, makes it hard to see how to adapt these techniques directly.

Vision is a pre-eminent machine intelligence problem. The difficulty of vision arises from the complexity and ambiguity of natural images (notoriously models designed using synthetic stimuli almost never scale-up to work on realistic images). The importance, and difficulty, of visual perception can be appreciated by realizing that the visual cortex is estimated to be roughly half the size of the entire cortex. It has been argued that vision can be formulated in terms

Long (Leo) Zhu
Department of Statistics
University of California at Los Angeles
Los Angeles, CA 90095
E-mail: lzhu@stat.ucla.edu

Yuanhao Chen
University of Science and Technology of China
Hefei, Anhui 230026 P.R.China
E-mail: yhchen4@ustc.edu

Chenxi Lin
Microsoft Research Asia
E-mail: chenxi.lin@microsoft.com

Alan Yuille
Department of Statistics, Psychology and Computer Science
University of California at Los Angeles
Los Angeles, CA 90095
E-mail: yuille@stat.ucla.edu

of probabilistic inference on structured representations [5]. But how can we design structured representations and perform learning and inference for realistic visual tasks? In the computer vision community, the effectiveness of probabilistic/machine learning approaches was first demonstrated by the success of techniques such as AdaBoost for tasks such as face [6] and text detection [7]. But AdaBoost, and related regression techniques, are limited by their lack of hidden variables to encode the structure of objects. This limitation reduces the set of problems to which they can be applied and the tasks that they can address. By contrast hierarchical models, such as AND-OR graphs [3, 4], offer a far richer more deeply structured representation for objects and scenes but are only useful provided efficient inference and learning algorithms can be found.

In this paper, we address the problem of detecting, segmenting and parsing articulated deformable objects, such as horses and human body, in cluttered backgrounds. Parsing articulated object like human body (i.e. pose estimation of body parts) in static image has recently received a lot of attention. Such problems arise in many applications including human action analysis, human body tracking, and video analysis. But the major difficulties of parsing the human body, which arise from the *large appearance variations* (e.g. different clothes) and *enormous number of poses*, have not been fully solved. There are three aspects to addressing these problems. Firstly, what representation is capable of modeling the large variation of both shape and appearance? Secondly, how can we learn a probabilistic model defined on this representation? Thirdly, if we have a probabilistic model, how can we perform inference efficiently (i.e. rapidly search over all the possible configurations of the object in order to estimate poses for novel images). These three aspects are clearly related to each other. Intuitively, the greater the representational power, the bigger the computational complexity of learning and inference. Most works in the literature, e.g. [8–10], focus on only one or two aspects, and not on all of them (see section (2.2) for a review of the literature). In particular, the representations used have been comparatively simple. Moreover, those attempts which do use complex representations tend to specify their parameters by hand and do not learn them from training data.

In this paper, we *represent* the different poses of the human body and horse by the form of AND/OR graphs proposed by Chen et al. for modeling deformable articulated objects [10], see figure (1). The design of this graph uses the principle of *summarization*, so that lower level nodes in the graph only pass on summary statistics (as an abstraction) to the higher level nodes. More precisely, the nodes of the AND/OR graph specify the position, orientation and scale of sub-configurations of the object (together with an index variable which specifies which sub-configurations of the object are present). The probability distribution defined on this

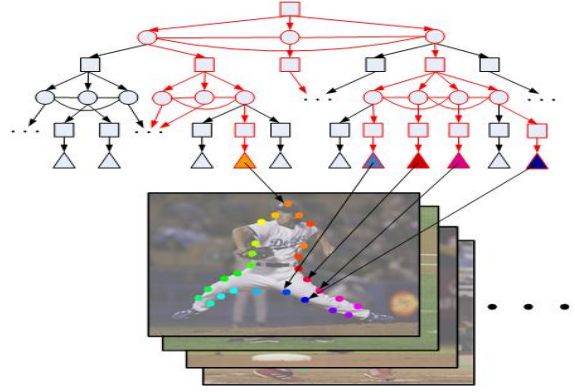


Fig. 1 The AND/OR representation allows us to model enormous poses of the object. A parse tree which is an instantiation of the AND/OR graph represents a specific pose of the human body. The nodes and edges in red indicate one parse tree. In this paper, there are 98 poses which can be modeled by the parse trees of the whole AND/OR graph.

representation is built using local potentials and hence obeys the Markov condition. It is designed to be invariant to the position, pose, and size of the object. The advantages of this AND/OR graph (see figure (1)) is that it can represent an enormous number of different poses (98 for human body and 40 for horse in this paper) in a compact form (i.e. only a small number of nodes are required), enforce (probabilistic) spatial relations on the configuration, and use many image features as input (to address the large appearance variations).

We next describe an algorithm for performing inference over this representation. This is a challenging task since the space of possible configurations is enormous. But the use of the summarization principle in our design of the AND/OR graph enables us to use (pruned) dynamic programming for inference. More precisely, our algorithm uses a bottom-up process that makes proposals for the possible configurations of the object. The bottom-up process is based on the principle of *compositionality*, where we combine proposals for sub-configurations together to form proposals for bigger configurations, hence we refer to this as *compositional inference*. To avoid a combinational explosion of proposals, we prune out proposals in two ways: (i) removing proposals whose goodness of fit is poor, and (ii) performing *surround suppression* to represent *local clusters* of proposals by a single *max-proposal*. Surround suppression ensures that the computational complexity of the inference algorithm is polynomial in the size of image (input data).

Finally, we *learn* the model parameters, which specify the geometry and the appearance, by a novel extension of the max-margin algorithm for structure learning [11–13]. This learning is global in the sense that we learn all the parameters simultaneously (by an algorithm that is guaranteed to find the global minimum) rather than learning local subsets of the parameters independently. Max-margin learning has been shown to be more effective than standard maximum

likelihood estimation when the overall goal is classification (e.g. into different poses). It also has some technical advantages such as: (i) avoiding the computation of the partition function of the distribution, and (ii) the use of the kernel trick to extend the class of features.

In summary, our paper makes contributions to both machine learning and computer vision. The contribution to machine learning is to extend max-margin learning to AND/OR graphs (max-margin has previously been applied to simpler models, see section (2)). The contribution to computer vision is the combination of the AND/OR *representation*, the max-margin *learning*, and compositional *inference* [10] to model articulated object (horse and human body) parsing. Moreover, our results, see section (6), show that our approach significantly outperforms the state of the art on benchmarked datasets.

2 Background

2.1 Object Representation

Detection, segmentation and parsing are all challenging problems. Most computer vision systems only address one of these tasks. There has been influential work on detection [14] and on the related problem of registration [15],[16]. Work on segmentation includes [17], [18], [19], [20], [21], [22], and [23]. Much of this work is formulated, or can be reformulated, in terms of probabilistic inference. But the representations are fixed graph structures defined at a single scale. This restricted choice of representation enables the use of standard inference algorithms (e.g. the hungarian algorithm, belief propagation) but it puts limitations on the types of tasks that can be addressed (e.g. it makes parsing impossible), the number of different object configurations that can be addressed, and on the overall performance of the systems.

In the broader context of machine learning, there has been a growing use of probabilistic models defined over variable graph structures. Important examples include stochastic grammars which are particularly effective for natural language processing [1]. In particular, vision researchers have advocated the use of probability models defined over AND/OR graphs [3],[4] where the OR nodes enable the graph to have multiple topological structures. Similar AND/OR graphs have been used in other machine learning problems [2].

But the representational power of AND/OR graphs comes at the price of increased computational demands for performing inference (and learning). For one dimensional problems, such as natural language processing, this can be handled by dynamic programming. But computation becomes considerably harder for vision problems and it is not clear how to efficiently search over the large number of configurations of an AND/OR graph. The inference problem sim-

plifies significantly if the OR nodes are restricted to lie at certain levels of the graph (e.g. [24], [25]), but these simplifications are not suited to the problem we are addressing.

2.2 Human Body Parsing

There has been considerable recent interest in human body parsing. Sigal and Black [26] address the occlusion problem by enhancing appearance models. Triggs and his colleagues [27] learn more complex models for individual parts by SVM and combine them by an extra classifier. Mori [9] use super-pixels to reduce the search space and thus speed up the inference. Ren et al. [28] present a framework to integrate multiple pairwise constraints between parts, but their models of body parts are independently trained. Ramanan [29] proposes a tree structured CRF to learn a model for parsing human body. Lee and Cohen [30] and Zhang et al. [31] use MCMC for inference. In summary, these methods involve representations of limited complexity (i.e. with less varieties of pose than AND/OR graphs). If learning is involved, it is local but not global (i.e. the parameters are not learnt simultaneously) [28,26,27,9]. Moreover, the performance evaluation is performed by the bullseye criterion: outputting a list of poses and taking credit if the groundtruth result is in this list [9,31,8].

The most related work is by Srinivasan and Shi [8] who introduced a grammar for dealing with the large number of different poses. Their model was manually defined, but they also introduced some learning in a more recent paper [32]. Their results are the state of the art, so we make comparisons to them in section (6).

By contrast, our model uses the AND/OR graph in the form of Chen et al. [10] which combines a grammatical component (for generating multiple poses) with a markov random field (MRF) component which represents spatial relationships between components of the model (see [4,3] for different types of AND/OR graph models). We perform global learning of the model parameters (both geometric and appearance) by max-margin learning. Finally, our inference algorithm outputs a only a single pose estimate which, as we show in section (6), is better than any of the results in the list output by Srinivasan and Shi [8] (and their output list is better than that provided by other algorithms [9]).

2.3 Max Margin Structure Learning

The first example of max-margin structure learning was proposed by Altun et al. [11] to learn Hidden Markov Models (HMMs) discriminatively. This extended the max margin criterion, used in binary classification [33] and multi-class classification [34], to learning structures where the output can be a sequence of binary vectors (hence an extension

of multi-class classification to cases where the number of classes is 2^n , where n is the length of the sequence). We note that there have been highly successful examples in computer vision of max-margin applied to binary classification, see SVM-based face detection [35].

Taskar et al. [12] generalized max margin structure learning to general markov random fields (MRF's), referred to as max margin markov network (M^3). Taskar et al. [13] also extended this approach to probabilistic context-free grammar (PCFG) for language parsing. But max-margin learning has not, until now, been extended to learning AND/OR graph models which can be thought of as combinations of PCFG's with MRF's.

This literature on max-margin structure learning shows that it is highly competitive with conventional maximum likelihood learning methods as used, for example, to learn conditional random fields (CRF's) [36]. In particular, max-margin structure learning avoids the need to estimate the partition function of the probability distribution (which is major technical difficulty of maximum likelihood estimation). Max-margin structure learning essentially learns the parameters of the model so that the groundtruth states are those with least energy (or highest probability) and states which are close to groundtruth also have low energy (or high probability). See section (5) for details.

3 The AND/OR Graph Representation

3.1 The Topological Structure of the AND/OR Graph

The structure of an AND/OR graph is represented by a graph $G = (V, E)$ where V and E denote the set of vertices and edges respectively. The vertex set V contains three types of nodes, "OR", "AND" and "LEAF" nodes which are depicted in figure (1) by circles, rectangles and triangles respectively. See two examples in figures (2) and (3). These nodes have attributes including position, scale, and orientation. The edge set E contains vertical edges defining the topological structure and horizontal edges defining spatial constraints on the node attributes. For each node $\nu \in V$, the set of its child nodes is defined by T_ν . Hence $\{T_\nu\}$ denotes all possible vertical edges of the AND/OR graph (the presence of OR nodes means that not all child nodes will appear in a parse, see next subsection). The horizontal edges are defined on triplets (μ, ρ, τ) of the children of AND nodes. The structure of the AND/OR graph is represented by the set of nodes and the edge set $\{(\nu, T_\nu, (\mu, \rho, \tau))\}$.

The directed (vertical) edges connect nodes at successive levels of the tree. They connect: (a) the AND nodes to the OR nodes, (b) the OR nodes to the AND nodes, and (c) the AND nodes to the LEAF nodes. The LEAF nodes correspond directly to points in the image. Connection types (a) and (c) have fixed parent-child relationships, but type

(b) has switchable parent-child relationship (i.e. the parent is connected to only one of its children, and this connection can switch). The horizontal edges only appear relating the children of the AND nodes. They correspond to Markov Random Fields (MRF's) and define spatial constraints on the node attributes (implemented by potentials). These constraints are defined to be invariant to translation, rotation, and scaling of the attributes of the children.

The AND/OR graph we use in this paper to represent the poses of human body and horse is shown in figures (2) and (3). In figure (2), the top node shows all the 98 possible configurations (i.e. parse trees of the human body). These configurations are obtained by AND-ing sub-configurations such as the torso, the left leg, and the right leg of the body (see circular nodes in the second row). Each of these sub-configurations has different *aspects* as illustrated by the AND nodes (rectangles in the third row). These sub-configurations, in turn, are composed by AND-ing more elementary configurations (see fourth row) which can have different aspects (see fifth row). The overall structure of this representation was hand-specified by the authors. Future work will attempt to learn it from examples.

3.2 The Representational Power of the AND/OR Graph Representation

The representational power of AND/OR graph is given by the number of topological configurations of the graph which we call parse trees and which correspond to different poses. Each parse tree corresponds to a specification of which AND nodes are selected by the OR nodes (i.e. each OR node is required to select a unique child). Hence the number of different parse trees is bounded above by W^{C^h} , where C is the maximum number of children of AND nodes (in this paper we restrict $C \leq 4$), W denotes the maximum number of possible children of OR nodes, and h is the number of levels containing OR nodes with more than one child node. The total number of parameters associated with the potential functions, which are defined on the edges of an AND/OR graph, is bounded above by MW^C where M is the number of AND nodes connecting to OR nodes. Hence the AND/OR graph can represent an exponentially large number of articulated poses, corresponding to different topologies, but with a compact form of polynomial size. This property of the AND/OR graph representation is very desirable for learning because it requires few training images to achieve good generalization. In the experiments reported in this paper we have $M = 35, C = 4, W = 3, h = 4$. There are 98 poses modeled by the AND/OR graph.

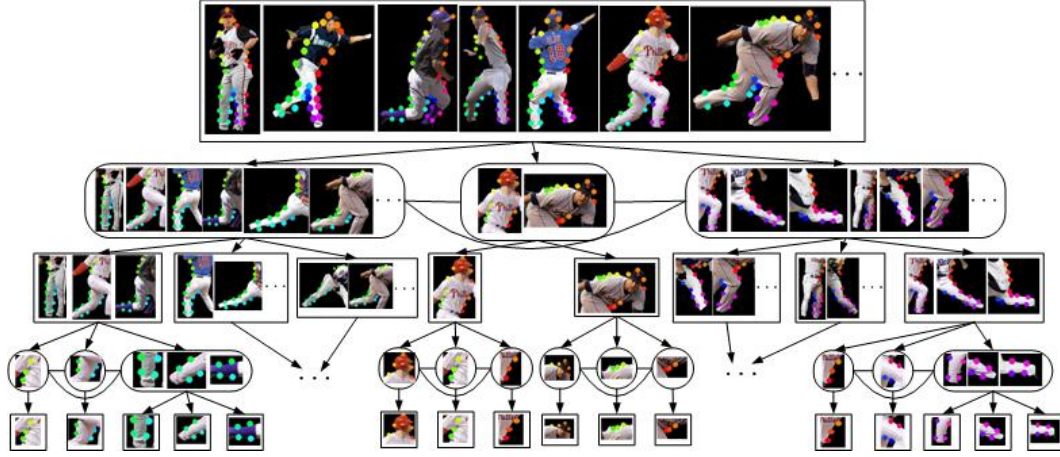


Fig. 2 The AND/OR graph is an efficient way to represent different appearances of an object. The graph was hand specified.. The bottom level of the graph indicates points along the boundary of human body. The higher levels indicate combinations of elementary configurations. The graph that we used contains eight levels (three lower levels are not depicted here due to lack of space). Color points distinguish different body parts. The arms are not modeled in this paper (or in related work in the literature).

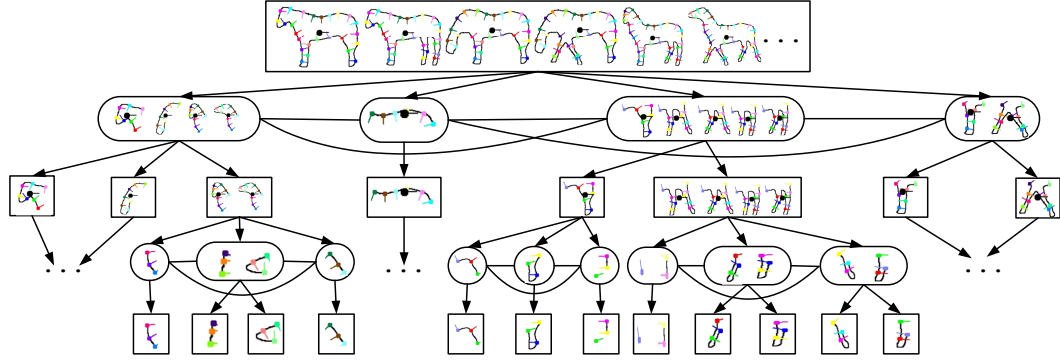


Fig. 3 The AND/OR graph for horses. There are 40 poses allowed in this paper. The first (typical) pose in the top node will be used for single hierarchy by fixing the child of all OR nodes.

3.3 The State Variables

A configuration (parse tree) of the AND/OR graph is an assignment of state variables $y = \{z_\nu, t_\nu\}$ with the state variable $z_\nu = (z_\nu^x, z_\nu^y, z_\nu^\theta, z_\nu^s)$ to each node ν , where (z_ν^x, z_ν^y) , z_ν^θ and z_ν^s denote image position, orientation, and scale respectively. The $t = \{t_\nu\}$ variable defines the specific topology of the parse tree, where t_ν denotes the children of node ν . More precisely, t_ν defines the vertical parent-child relations by indexing the children of node ν . t_ν is fixed and $t_\nu = T_\nu$ if ν is an AND node (because the node is always connected to all its children – recall that T_ν is the set of child nodes of ν), but t_ν is a variable for an OR node ν (to enable sub-configurations to switch their appearances and shapes), see figure (2). We use the notation Λ_ν to denote the state y_ν at node ν , together with the states of all the descendent nodes of ν (i.e. the children of ν , their children, and so on). The input to the graph is the image $x = \{x_\nu\}$ defined on the image lattice (at the lowest level of the graph).

We define $V^{LEAF}(t)$, $V^{AND}(t)$, $V^{OR}(t)$ to be the set of LEAF, AND, and OR nodes which are active for a specific

choice of the topology t of a parse tree. These sets can be computed recursively from the root node, see figure (2). The AND nodes in the second row (i.e. the second highest level of the graph) are always activated and so are the OR nodes in the third row. The AND nodes activated in the fourth row, and their OR node children in the fifth row, are determined by the t variables assigned to their parent OR nodes. This process repeats till we reach the lowest level of the graph.

A novel feature of this AND/OR representation is that the node variables are the same at all levels of the hierarchy. We call this the *summarization principle* which make use of the compositionality. It means that the state of an AND node will be a simple deterministic function of the state variables of the children (see section (3.4)). This differs from other AND/OR graphs [3],[4] where the node variables at different levels of the graph are typically at different levels of abstraction. The use of the summarization principle helps us to define a successful inference algorithm.

3.4 The Potential Functions for the AND/OR Graph

The conditional distribution on the states and the data is given by:

$$P(y|x; w) = \frac{1}{Z(x; w)} \exp \langle w, \Psi(x, y) \rangle. \quad (1)$$

where x is the input image, y is the parse tree, and $Z(x, w)$ is the partition function. $P(y|x; w)$ is a (conditional) exponential model which is defined by an inner product $\langle w, \Psi(x, y) \rangle$ between features $\Psi(x, y)$ and model parameters w (to be learnt). The features $\Psi(x, y)$ are of three types: (i) appearance features $\Psi^D(x, y)$, (ii) horizontal spatial relationship features $\Psi^H(y)$, and (iii) vertical relationship features $\Psi^V(y)$. Note that only the appearance features depend on the data x (the other features are like prior distributions). The inner product $\langle w, \Psi(x, y) \rangle$ can be decomposed into three energy terms:

$$\langle w, \Psi(x, y) \rangle = -E^D(x, y) - E^H(y) - E^V(y) \quad (2)$$

The data term $E^D(x, y)$ is given by:

$$E^D(x, y) = \sum_{\nu \in V^{LEAF}(t)} w_{\nu}^D \Psi_{\nu}^D(x, y) \quad (3)$$

where the appearance features $\Psi_{\nu}^D(x, y), \forall \nu \in V^{LEAF}(t)$ are data dependent and model the appearance of the object. They relate the appearance of the active leaf nodes to properties of the local image. More formally, y in $\Psi_{\nu}^D(x, y)$ refers to $z_{\nu} = (z^x, z^y, z^s, z^{\theta})$ for the active nodes $\nu \in V^{LEAF}$. $\Psi_{\nu}^D(x, z_{\nu})$ represent the local image features including the grey intensity, the gradient, canny edge map, the responses of Gabor filters at different scales and orientations, and related features. We use a total of 101 features of this type (i.e. the vector $\Psi^D(x, y)$ has 101 dimensions). But not all these features will be used (the max-margin learning will typically set some of the parameters w_{ν}^D to be zero).

The next two terms in r.h.s of equation (2) make use of the hierarchical structure. The horizontal component $E^H(y)$ of the hierarchical shape prior is used to impose the horizontal connections at a range of scales and defined by

$$E^H(y) = \sum_{\nu \in V^{AND}(t)} \sum_{(\mu, \rho, \tau) \in t_{\nu}} w_{\nu, \mu, \rho, \tau}^H \Psi_{\nu, \mu, \rho, \tau}^H(y) \quad (4)$$

where the horizontal spatial relationship features $\Psi^H(y)$ specify the horizontal relationships (which correspond to geometric constraints at a range of scales). They are defined by $\Psi_{\nu, \mu, \rho, \tau}^H(y) = g(z_{\mu}, z_{\rho}, z_{\tau}), \forall \nu \in V^{AND}(t)$ where $g(\cdot, \cdot, \cdot)$ is a logarithm of Gaussian distribution defined on the *invariant shape vector* $l(z_{\mu}, z_{\rho}, z_{\tau})$ [25] constructed from triple child nodes $(z_{\mu}, z_{\rho}, z_{\tau})$ of node ν . See figure (4). This shape vector models the shape deformation and depends only on variables of the triple, such as the internal angles, which are

invariant to the translation, rotation, and scaling of the triple. This type of feature is defined over all triples formed by the child nodes of each parent, see figures (4). The parameters of the Gaussians are estimated from the labeled training data (this is local learning, but max-margin will learn their parameters w^H globally).

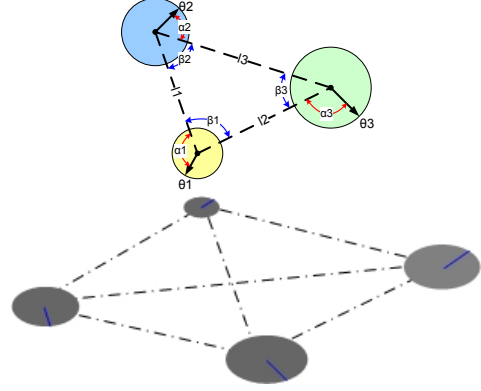


Fig. 4 Representation based on oriented triplet. The first panel demonstrates the invariant shape vector constructed from triple nodes. This shape vector depends only on variables of the triple, such as the internal angles, the ratio of length, which are invariant to the translation, rotation, and scaling of the triple. The second panel gives the cliques for the four children of a node. In this example, four triplets are computed. Each circle corresponds to one node in the hierarchy which has a descriptor (indicated by blue line) of position, orientation and scale. The potentials of the cliques are Gaussians defined over features extracted from triple nodes, such as internal angles of the triangle and relative angles between the feature orientation and the orientations of the three edges of the triangle. These extracted features are invariant to scale and rotation.

The vertical component $E^V(y)$ is used to hold the structure together by relating the state of the parent nodes to the state of its children. $E^V(y)$ is divided into three vertical energy terms denoted by $E^{V,A}(y)$, $E^{V,B}(z)$ and $E^{V,C}(z)$ which correspond to type(A), type(B) and type(C) vertical connections respectively. Hence we have

$$E^V(y) = E^{V,A}(y) + E^{V,B}(y) + E^{V,C}(y) \quad (5)$$

$E^{V,A}(y)$ specifies the coupling from the AND node to the OR node. This coupling is deterministic – the state of the parent node is determined precisely by the states of the child nodes. This is defined by:

$$E^{V,A}(y) = \sum_{\nu \in V^{AND}(t)} w_{\nu}^{V,A} \Psi_{\nu}^{V,A}(y) \quad (6)$$

where $\Psi_{\nu}^{V,A}(y) = h(z_{\nu}, \{z_{\mu} \text{ s.t. } \mu \in t_{\nu}\}), \forall \nu \in V^{AND}(t)$, where $h(\cdot, \cdot) = 0$ if the average orientations and positions of the child nodes are equal to the orientation and position of the parent node (i.e. the vertical constraints are “hard”). If they are not consistent, then $h(\cdot, \cdot) = \kappa$, where κ is a large negative number.

$E^{V,B}(y)$ accounts for the probability of the assignments of the connections from OR nodes to AND nodes. We define:

$$E^{V,B}(y) = \sum_{\nu \in V^{OR}(t)} w_{\nu}^{V,B} \Psi_{\nu}^{V,B}(y) \quad (7)$$

where $\Psi_{\nu}^{V,B}(y)$ is an indicator function which equals one while the node ν is active and zero otherwise. $w_{\nu}^{V,B}$ encodes the weights of the assignments determined by t_{ν} .

The energy term $E^{V,C}(y)$ defines the connection from the lowest AND nodes to the LEAF nodes. This is similar to the definition of $E^{V,A}(y)$, and $E^{V,C}(y)$ is given by

$$E^{V,C}(y) = \sum_{t_{\nu} \in V^{LEAF}(t)} w_{\nu}^{V,C} \Psi_{\nu}^{V,C}(y) \quad (8)$$

where $\Psi_{\nu}^{V,C}(y) = h(z_{\nu}; z_{t_{\nu}})$ where $h(.,.) = 0$ if the orientation and position of the child (LEAF) node is equal to the orientation and position of the parent (AND) node. If they are not consistent, then $h(.,.) = \kappa$.

Finally, we can compute the energy of the sub-tree for a particular node ν as root node. The sub-tree energy is useful when performing inference, see section (4). This is computed by summing over all the potential functions associating to the node ν and its descendants. This energy is defined by:

$$E_{\nu}(\Lambda_{\nu}) = E^D(x, y) + E^H(y) + E^V(y). \quad (9)$$

where $y \in \Lambda_{\nu}$ and $V^{LEAF}(t), V^{AND}(t), V^{OR}(t)$ in the summation of each term are defined in the set of the node ν and its descendants.

4 The Inference/Parsing Algorithm

We use the inference algorithm first reported in [10] to obtain the best parse tree y^* of an image x by computing $y^* = \arg \max_y \langle w, \Psi(x, y) \rangle$ where the inner product is defined in equation (2). This algorithm runs in polynomial time in terms of the size of input image and the number of levels of the AND/OR graph (no other algorithm has this level of inference performance on AND/OR graphs). This rapid inference is necessary to make max margin learning practical.

The algorithm has a bottom-up stage which makes proposals for the configuration of the AND/OR graph. This proceeds by combining proposals for sub-configurations to build proposals for larger configuration. For AND nodes, we combine proposals for the child nodes to form a proposal for the parent node. For OR nodes, we enumerate all proposals from all branches without composition. To prevent a combinatorial explosion we prune out weak proposals which have low fitness score ($\langle w, \Psi(x, y) \rangle$ evaluated for the configuration) and use clustering which selects a small set of *max-proposals* (each representing a cluster).

The pseudo-code for the algorithm is shown in figure 5. We use the following notation. Each node ν^l at level l has a set of *proposals* $\{P_{\nu,a}^l\}$ where a indexes the proposals (see table (2) for the typical number of proposals). There are also *max-proposals* $\{MP_{\nu,a}^l\}$, indexed by a , each associated with a local cluster $\{CL_{\nu,a}^l\}$ of proposals (see table (2) for the typical number of max-proposals). Let S denote the number of clusters. Each proposal, or max-proposal, is described by a state vector $\{y_{\nu,a}^l : a = 1, \dots, M_{\nu}^l\}$, the state vectors for it and its descendants $\{\Lambda_{\nu,a}^l : a = 1, \dots, M_{\nu}^l\}$, and an energy function score $\{E_{\nu}^l(\Lambda_{\nu,a}^l) = \langle w, \Psi(x, y) \rangle : a = 1, \dots, M_{\nu}^l\}$.

We obtain the proposals by a bottom-up strategy starting at level $l = 2$ (AND node) of the tree. For a node ν^2 we define windows $\{W_{\nu,a}^2\}$ in space, orientation, and scale. We exhaustively search for all configurations within this window which have a score (goodness of fit criterion) $E_{\nu}^2(\Lambda_{\nu,a}^2) < K_2$, where K_2 is a fixed threshold. For each window $W_{\nu,a}^2$, we select the configuration with largest score to be the proposal $MP_{\nu,a}^2$ and store the remaining proposals below threshold in the associated cluster $CL_{\nu,a}^2$. This window enforces *surround suppression* which performs clustering to keep the proposal with the maximum score in any local window. Surround suppression guarantees the number of the remaining proposals at each level is proportional to the size of image (input data). Note that the potential functions associated with the nodes at level l only rely on the position, orientation and scale at level l , but not on the states of its descendants at level $l - 1, l - 2, \dots, 1$. Therefore, different detailed configurations of subparts with the same global pose will have identical energies for the higher levels. This strategy essentially is (pruned) dynamic programming and ensures that we do not obtain too many proposals in the hierarchy and avoid a combinatorial explosion of proposals. We will analyze this property empirically in section 6. The procedure is repeated as we go up the hierarchy. Each parent node ν^{l+1} produces proposals $\{P_{\nu,a}^{l+1}\}$, and associated clusters $\{CL_{\nu,a}^{l+1}\}$, by combining the proposals from its children. All proposals are required to have scores $E_{\nu}^{l+1}(\Lambda_{\nu,a}^{l+1}) < K_{l+1}$, where K_l is a threshold.

Recall that the cluster is defined over image position, orientation and scale. Thus, the number S of proposals of each node at different levels is linearly proportional to the size of the image. It is straight forward to conclude that the complexity of our algorithm is bounded above by $M \times W^C \times S^C$. Recall that C is the maximum number of children of AND nodes (in this paper we restrict $C \leq 4$), W denotes the maximum number of possible children of OR nodes and M is the number of AND nodes connecting to OR nodes. This shows that the algorithm speed is polynomial in W and S (and hence in the image size). The complexity for our experiments is reported in section (6).

Input: $\{MP_{\nu,1}^1\}$. Output: $\{MP_{\nu,L}^L\}$. \oplus denotes the operation of combining two proposals.
 Loop : $l = 1$ to L , for each node ν at level l

- IF ν is an OR node
 1. Union: $\{MP_{\nu,b}^l\} = \bigcup_{\rho \in T_{\nu}, a=1, \dots, M_{\rho}^{l-1}} MP_{\rho,a}^{l-1}$
- IF ν is an AND node
 1. Composition: $\{P_{\nu,a}^l\} = \bigoplus_{\rho \in T_{\nu}, a=1, \dots, M_{\rho}^{l-1}} MP_{\rho,a}^{l-1}$
 2. Pruning: $\{P_{\nu,a}^l\} = \{P_{\nu,a}^l | E(A_{\nu,a}^l) > K_l\}$
 3. Local Maximum: $\{(MP_{\nu,a}^l, CL_{\nu,a}^l)\} = LocalMaximum(\{P_{\nu,a}^l\}, \epsilon_W)$ where ϵ_W is the size of the window W_{ν}^l defined in space, orientation, and scale.

Fig. 5 The inference algorithm. The operation LocalMaximum implements surround suppression.

In practice, the thresholds K_l are not explicitly defined. Instead, we keep top K proposals for each node whose energy scores are greater than those of any other proposals. K is empirically set to be 300 in our experiments. In very rare situations we may find no proposals for the state of one node of a triplet. In this case, we use the states of the other two nodes together with the horizontal potentials (geometrical relationship) to propose states for the node. A similar technique was used in [25].

5 Max Margin AND/OR Graph Learning

5.1 Primal and Dual Problems

The task of AND/OR graph learning is to estimate the parameters w from a set of training samples $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ drawn from some fixed, but unknown probability distribution. In this paper, x is the image and y is the configurations of the AND/OR graph.

We formulate this learning task in terms of the max-margin criterion which is designed to learn the parameters which are best for classification (i.e. to estimate y) rather than use the standard maximum likelihood criterion (see [33] for a justification for this strategy). But observe that the classification is over the set of values \mathcal{Y} , which is exponentially large, and hence differs greatly from simple binary classification. Effectively max-margin learning seeks to find values of the parameters w which ensure that the energies $\langle \Psi(x, y), w \rangle$ are smallest for the ground-truth states y and for states close to the ground-truth. A practical advantage of max-margin learning is that it gives a computationally tractable learning algorithm (which avoids the need to compute the partition function of the distribution).

The main idea of the max margin approach is to forego the probabilistic interpretation of equation (1). Instead we concentrate on the discriminative function:

$$F(x, y, w) = \langle \Psi(x, y), w \rangle. \quad (10)$$

We define the *margin* γ of the parameter w on example i as the difference between the true parse y_i and the best parse y^* :

$$\gamma_i = F(x_i, y_i, w) - \max_{y \neq y_i} F(x_i, y, w) \quad (11)$$

$$= \langle w, \Psi_{i,y_i} - \Psi_{i,y^*} \rangle \quad (12)$$

where $\Psi_{i,y_i} = \Psi(x_i, y_i)$ and $\Psi_{i,y} = \Psi(x_i, y)$.

Intuitively, the size of margin quantifies the confidence in rejecting the incorrect parse y using the function $F(x, y, w)$. Larger margins [33] leads to better generalization and prevents over-fitting.

The *goal* of max margin AND/OR graph learning is to maximize the margin:

$$\max_{\gamma} \quad (13)$$

$$\text{s.t.} \quad \langle w, \Psi_{i,y_i} - \Psi_{i,y} \rangle \geq \gamma L_{i,y}, \forall y; \quad \|w\|^2 \leq 1; \quad (14)$$

where $L_{i,y} = L(y_i, y)$ is a loss function (note there are an exponential number $|\mathcal{Y}|$ of constraints in equation (14)). The purpose of the loss function is to give partial credit to states which differ from the groundtruth by only small amounts (i.e. it will encourage the energy to be small for states near the groundtruth).

The loss function is defined as follows:

$$L(y_i, y) = \sum_{\nu \in V^{AND}} \Delta(z_{\nu}^i, z_{\nu}) + \sum_{\nu \in V^{LEAF}} \Delta(z_{\nu}^i, z_{\nu}) \quad (15)$$

where $\Delta(z_{\nu}^i, z_{\nu}) = 1$ if $dist(z_{\nu}^i, z_{\nu}) \geq \sigma$. Otherwise, we have $\Delta(z_{\nu}^i, z_{\nu}) = 0$. $dist(., .)$ is a measure of the spatial distance between two image points and σ is a threshold. Note that the summations are defined over the active nodes. This loss function, which measures the distance/cost between two parse trees, is calculated by summing over individual parts. This ensures that the computational complexity of the loss function is linear in the size of the LEAF and AND nodes of the hierarchy.

By standard manipulation, the optimization can be reformulated as minimizing the constrained quadratic cost function of the weights:

$$\min_w \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad (16)$$

$$\text{s.t.} \quad \langle w, \Psi_{i,y_i} - \Psi_{i,y} \rangle \geq L_{i,y} - \xi_i, \forall y; \quad (17)$$

where C is a fixed penalty parameter which balances the trade-off between margin size and outliers. Outliers are training samples which are only correctly classified after using a slack variable ξ_i to “move them” to the correct side of the margin. The constraints are imposed by introducing Lagrange parameters $\alpha_{i,y}$ (one α for each constraint).

The solution to this minimization can be found by differentiation and expressed in form:

$$w^* = C \sum_{i,y} \alpha_{i,y}^* (\Psi_{i,y_i} - \Psi_{i,y}), \quad (18)$$

where the α^* are obtained by maximizing the dual function:

$$\max_{\alpha} \sum_{i,y} \alpha_{i,y} L_{i,y} - \frac{1}{2} C \sum_{i,j} \sum_{y,z} \alpha_{i,y} \alpha_{j,z} \langle \Psi_{i,y_i} - \Psi_{i,y}, \Psi_{j,y_j} - \Psi_{j,z} \rangle \quad (19)$$

$$\text{s.t.} \quad \sum_y \alpha_{i,y} = 1, \forall i; \quad \alpha_{i,y} \geq 0, \forall i, y; \quad (20)$$

Observe that the solution will only depend on the training samples (x_i, y_i) for which $\alpha_{i,y} \neq 0$. These are the so-called *support vectors*. They correspond to training samples that either lie directly on the margin or are outliers (that need to use slack variables). The concept of support vectors is important for the optimization algorithm that we will use to estimate the α^* (see next subsection).

It follows from equations (18,19), that the solution only depends on the data by means of the inner product $\Psi \cdot \Psi'$ of the potentials. This enables us to use the kernel trick [37] which replaces the inner product by a kernel $K(\cdot, \cdot)$ (interpreted as using features in higher dimensional spaces). In this paper, the kernels $K(\cdot, \cdot)$ take two forms, the linear kernel, $K(\Psi, \Psi') = \Psi \cdot \Psi'$ for image features Ψ^D and the radial basis function (RBF) kernel, $K(\Psi, \Psi') = \exp(-r\|\Psi - \Psi'\|^2)$ for shape features Ψ^H where r is a parameter of RBF.

5.2 Optimization of the Dual

The main difficulty with optimizing the dual, see equation (19), is the exponential number of constraints (and hence the exponential number of $\{\alpha_{i,y}\}$ to solve for). We risk having to enumerate all the parse trees $y \in \mathcal{Y}$ which is almost impractical for an AND/OR graph. Fortunately, in practice only a small number of support vectors will be needed (equivalently, only a small number of the $\{\alpha_{i,y}\}$ will be non-zero). This motivates the working set algorithm [11,38] to optimize the objective function in equation (19). The algorithm aims at finding a small set of *active constraints* that ensure a sufficiently accurate solution. More precisely, it sequentially creates a nested working set of successively tighter relaxations using a cutting plane method. It is shown [11, 38] that the remaining (exponentially many) constraints are guaranteed to be violated by no more than ϵ , without needing to explicitly add them to the optimization problem. The pseudocode of the algorithm is given in figure (6). Note that the inference algorithm is performed at the first step of each loop. Therefore, the efficiency of the training algorithm

Loop over i

1. $y^* = \arg \max_y H(x_i, y)$ where $H(x_i, y) = \langle w, \Psi_{i,y} \rangle + L(y_i, y)$.
2. if $H(x_i, y^*; \alpha) - \max_{y \in S_i} H(x_i, y; \alpha) > \epsilon$
 $S_i \leftarrow S_i \cup y^*$
 $\alpha_s \leftarrow \text{optimize dual over } S, S = S \cup S_i$

Fig. 6 Working Set Optimization

highly depends on the computational complexity of the inference algorithm (recall that we show in section (4) that the complexity of the inference algorithm is polynomial in the size of the AND/OR graph and the size of the input image). Thus, the efficiency of inference makes the learning practical. The second step is to create the working set sequentially and then estimate the parameter α on the working set. The optimization over the working set is performed by Sequential Minimal Optimization (SMO) [39]. This involves incrementally satisfying the Karush-Kuhn-Tucker (KKT) conditions which are used to enforce the constraints. The pseudocode of the SMO algorithm is depicted in figure (7). This procedure consists of two step. The first step selects a pair of data points not satisfying the KKT conditions. The pseudocode of pair selection is shown in figure (8). Two KKT conditions are defined by:

$$\alpha_{i,y} = 0 \Rightarrow H(x_i, y) \leq H(x_i, y^*) + \epsilon; \quad (KKT1)$$

$$\alpha_{i,y} > 0 \Rightarrow H(x_i, y) \geq H(x_i, y^*) - \epsilon; \quad (KKT2)$$

where $H(x_i, y) = \langle w, \Psi_{i,y} \rangle + L(y_i, y)$, $y^* = \arg \max_y H(x_i, y)$ and ϵ is a tolerance parameter.

The second step is a local ascent step which attempts to update the parameters given the selected pair. The updating equations are defined as:

$$\begin{aligned} \alpha_{x_i, y'}^{new} &= \alpha_{x_i, y'} + \delta \\ \alpha_{x_i, y''}^{new} &= \alpha_{x_i, y''} - \delta \end{aligned} \quad (21)$$

The dual optimization problem in equation (19) becomes a simple problem on δ :

$$\max_{\delta} [H(x_i, y') - H(x_i, y'')] \delta - \frac{1}{2} C \|\Psi_{i, y'} - \Psi_{i, y''}\|^2 \delta^2 \quad (22)$$

$$\text{s.t.} \quad \alpha_{x_i, y'} + \delta \geq 0, \alpha_{x_i, y''} - \delta \geq 0 \quad (23)$$

Equivalently we have :

$$\max_{\delta} a\delta - \frac{b}{2} \delta^2 \quad (24)$$

$$\text{s.t.} \quad c \leq \delta \leq d \quad (25)$$

where $a = H(x_i, y') - H(x_i, y'')$, $b = C \|\Psi_{i, y'} - \Psi_{i, y''}\|^2$, $c = -\alpha_{x_i, y'}$, $d = \alpha_{x_i, y''}$.

Hence, the analytical solution for two data points can be easily obtained by

$$\delta^* = \max(c, \min(d, a/b)). \quad (26)$$

Given a training set S and parameter α
Repeat
1. select a pair of data points (y', y'') not satisfying the KKT conditions.
2. solve optimization problem on (y', y'')
Until all pairs satisfy the KKT conditions.

Fig. 7 Sequential Minimal Optimization (SMO)

1. $Violation = False$
2. For each $x^i, y', y'' \in S_i$
(a) If $H(x^i, y') > H(x^i, y'') + \epsilon$ and $\alpha_{x_i, y'} = 0$ (KKT 1)
 $Violation = TRUE$; Goto step 3.
(b) If $H(x^i, y') < H(x^i, y'') - \epsilon$ and $\alpha_{x_i, y'} > 0$ (KKT 2)
 $Violation = TRUE$; Goto step 3.
3. Return y', y'', v

Fig. 8 Pair Selection in SMO

Up to now, we have the solutions for the updating equations in (21). More details can be found in [39] and [12].

6 Experiments

In this section, we first study the performance of the AND/OR graph with max-margin learning on the horse dataset [40] and analyze the computational complexity of the inference. Next we apply the AND/OR graph for parsing the human body which has more poses and compare its performance with alternative methods.

6.1 Datasets and Implementation Details.

Datasets. We performed the experimental evaluations on two datasets, i.e. the Weizmann Horse Dataset [40] and Mori’s Human Baseball dataset [9]. There are many results reported for comparisons ([23, 20, 21, 17, 22] for horse segmentation and [8, 41, 9] for human body parsing). The Weizmann horse dataset is designed to evaluate segmentation, so the groundtruth only gives the regions of the object and the background. To supplement this groundtruth, we required students to manually parse the images by locating the positions of active leaf nodes (about 24 to 36 nodes) of the AND/OR graph in the images. These parse trees are used as ground truth to evaluate the ability of the AND/OR graph to parse the horses. In the experiment of human body parsing, Srinivasan and Shi [8] only used 5 joint nodes (head-torso, torso-left thigh, torso-right thigh, left thigh-left lower leg, right thigh-right lower leg) per image. In our case, there are 27 nodes along the boundary of human body per image used to give more detailed parsing than those [8, 41, 9]. Therefore, we also asked students to label the parts of the human body as ground truth (i.e. to identify different parts of the human). There are 328 horse images in [40] of which 100 images are

used for testing. The AND/OR model has 40 possible configurations to cover horse poses. For human body parsing, we used 48 human baseball images in Mori’s dataset [9] as the testing set. Some examples of the dataset are shown in figures (9) and (10) (The parsing and segmentations results are obtained by our method). The AND/OR graph for human body is capable of modeling 98 poses. In figure (10), observe that the dataset contains a large variance of poses of human body and the appearance of clothes changes a lot from image to image. We created a training dataset by collecting 156 human baseball images from the internet and got students to manually label the parse tree for each image.

Parameter Settings. The AND/OR graph learnt by max-margin was used to obtain the parse y (i.e. to locate the body parts). We used max-margin on the training dataset to learn the parameters of the max-margin model. During learning, we set $C = 0.1$ in equation (19), used the radial basis function kernel with $r = 0.1$, set the parameter in the loss function equation (15) to be $\sigma = 12$, and set $\epsilon = 0.01$ in figure (6). Our strategy to obtain segmentation, which is inspired by Grab-Cut [42], is to obtain the parse by the inference algorithm on the AND/OR graph and then segment object by graph-cut using the feature statistics inside the boundary as initializations (note that, unlike our approach, Grab-Cut requires initialization by a human).

The Criterion for Parsing. The *average position error* [8] is used as the measure of the quality of parsing. The position error means the distance at pixel level between the positions of groundtruth and the parsing result. The smaller the position error, the better the quality of the parsing. For horse, there are 24 to 36 leaf nodes used to cover the boundary of a horse. In the experiment of human body parsing, Srinivasan and Shi [8] only used 5 joint nodes (head-torso, torso-left thigh, torso-right thigh, left thigh-left lower leg, right thigh-right lower leg) per image. In our case, there are 27 nodes along the boundary of human body per image used to give more detailed parsing.

The Criteria for Segmentation. Two evaluation criteria are used to measure the performances of segmentation. We use *segmentation accuracy* to quantify the proportion of the correct pixel labels (object or non-object). For performance comparisons of human body parsing, we use the segmentation measure, “*overlap score*” named by [8], to quantify the performance of segmentation of human body. The overlap score is defined by $\frac{area(P \cap G)}{area(P \cup G)}$, where P is the area which the algorithm outputs as the segmentation and G is the area of ground-truth. The bigger the overlap score, the better the segmentation.

The Criterion for Detection. We rate detection as a success if the area of intersection of the detected object region and the true object region is greater than half the area of the union of these regions.

6.2 Performance of the AND/OR graph on the Horse dataset

Results. In table (1) we compare the performances between the AND/OR graph with 40 configurations and a simple hierarchical model with a fixed configuration (i.e. we fix the states of the OR nodes). This configuration (the first one in the top node in figure (3)) was chosen to be the typical pose that most frequently occurred. Column 3 gives the parsing accuracy – the average position error of leaf node of the AND/OR graph is 10 pixels. Column 4 quantifies the segmentation accuracy. Column 5 quantifies the detection rate. Column 6 lists the training time. The last column shows the average time of inference taken for one image. A computer with 4 GB memory and 2.4 GHz CPU was used for training and testing. The time costs are 150 minutes for learning a hierarchy and 180 minutes for AND/OR graph. For a new image, the testing (inference) time is 20 seconds for the hierarchy model and 27 second for the AND/OR model. The AND/OR graph outperforms the simple hierarchical model in the tasks of parsing, detection and segmentation with only 30% more computational cost. In figure (9), we show the parse and segmentation results obtained by the single hierarchy model and the AND/OR graph model. The states of the leaf nodes of parse tree indicate the positions of the points along the boundary which are represented as colored dots. In different images, the same color corresponds to the same object parts. Both models learnt by max-margin learning are able to deal with large shape deformation and appearance variations. See the top four examples which contains the white, black and textured body with cluttered background. The hierarchical model was only capable of reliably locating the main body but the AND/OR graph is able to reliably capture more details such as the legs and heads (despite their variability under different poses). See the last four examples in figure (9) where the legs and heads appear at different poses. The hierarchical model succeeds in segmentation in most cases even though its parse results are not accurate. In the last example, the incorrect parsing, where the hierarchical model locates the head at a wrong position with similar appearance, results in wrong segmentation. In this case, the AND/OR graph model performs well on both parsing and segmentation tasks.

Comparisons. In table (1), we compare the segmentation performance of our approach with other successful methods. Note that the object cut method [17] was reported on only 5 images. Levin and Weiss [21] make the strong assumption that the position of the object is given (other methods do not make this assumption) and not report how many images they tested on. Overall, Cour and Shi’s method [20] was the best one evaluated on large dataset. But their result is obtained by manually selecting the best among top 10 results (other methods output a single result). By contrast, our

Table 2 Complexity Analysis. This table shows the numbers of proposals and time costs at different levels.

L	Nodes	Aspects	Max-Proposals	Proposals	Time
8	1	12	11.1	2058.8	1.206s
6	8	1.5	30.6	268.9	1.338s
4	27	1	285.1	1541.5	1.631s
2	68	1	172.2	1180.7	0.351s

approach outputs a single parse only but yields a higher pixel accuracy of 95.2%. Hence we conclude that our approach outperforms those alternatives which have been evaluated on this dataset. Note no other papers report parsing performance on this dataset since most (if not all) methods do not estimate the positions of different parts of the horse (or even represent them).

6.3 Computational Complexity Analysis

Table (2) shows the complexity properties of the algorithm. We described the AND levels only (the model has 8 levels). The computation for the OR-nodes is almost instantaneous (you just need to list the proposals from all its children AND nodes) so we do not include it. Column 2 gives the number of nodes at each level. Column 3 states the average number of *aspects*¹ of the AND nodes at each level. Column 4 states the average number of max-proposals for each node. Column 5 gives the average number of proposals. Column 6 gives the time. Observe that the number of proposals increases by an order of magnitude from level 6 to level 8. This is mostly due to the similar increase in the number of aspects (the more the number of aspects, the more the number of proposals needed to cover them). But surround suppression is capable of reducing the number of proposals greatly (compare the numbers of Max-proposals and proposals in Table (2)).

6.4 Human Body Parsing

Parsing Results. We illustrate our parsing and segmentation results of human body in figure (10). The dotted points indicate the positions of the leaf nodes of parse tree which lie along the boundary of human body. The same parts in different images share the same color. For example, yellow and red points correspond to the left and right shoulder respectively. Light blue and dark blue points correspond to the left and right legs respectively. Observe that the variation of poses are extremely large, but our AND/OR graph is capable

¹ Here is the definition of *aspects*. Let AND node ν have children OR nodes $\{ \rho_i : i \in t_\nu \}$. This gives a set of grandchildren AND nodes $\bigcup_{i \in t_\nu} t_{\rho_i}$. The aspect of ν is $\prod_{i \in t_\nu} |t_{\rho_i}|$. The aspect of an AND node is an important concept because when passing proposals up to an AND node we must take into account the number of aspects of this node. We can, in theory, have proposals for all possible aspects.



Fig. 9 This figure is best viewed in color. Columns from (a) to (d) show the parsing and segmentation results obtained by hierarchy and AND/OR graph models respectively. The colored dots correspond to the leaf nodes of the object.

Table 1 Performance for parsing, segmentation and detection. The table compares the results for the hierarchical model (without OR nodes), AND/OR graph and other alternative methods.

Models	Testing Size	Parsing	Segementation Accuracy	Detection	Training Time	Testing Time
Hierarchical Model	100	15.6	94.5%	99%	150 m	20s
AND/OR Graph	50	9.7	95.2%	100%	180 m	27s
Ren et. al[23]	172	—	91%	—	—	—
Borenstein [19]	328	—	93.0%	—	—	—
LOCUS [22]	200	—	93.1%	—	—	—
Cour [20]	328	—	94.2%	—	—	—
Levin [21]	N/A	—	95.0%	—	—	—
OBJ CUT [17]	5	—	96.0%	—	—	—

of covering the articulated poses of body parts and segmenting the body nicely. The time cost of training the AND/OR graph is 20 hours. The inference takes 2 minutes for image with size 640×480 .

Performance Comparisons. We compare the performance obtained by our approach to those reported by Srinivasan and Shi [8], which are the best results achieved so far on this dataset (e.g. better than Mori et al. 's [9]). Firstly, we compare the average position errors in figure (11). Observe that our best parse gives performance slightly better than the best (manually selected) of the top 10 parses output by [8] and significantly better than the best (manually selected) of their top three parses. Secondly, we compare the average overlap scores in figure (11). The difference of performance measured by overlap score is more significant. Observe that our result is significantly better than the best (manually selected) of their top 10 parses.

Convergence Analysis. We study the convergence behavior of max-margin AND/OR graph learning in figure (12). The left figure shows the convergence curve in terms of objective function defined in equation (19). There is a big jump before iteration 200. The right figure plots the convergence curves of the average position error on the training and testing data. One can see that the trends of two curves are very similar.

7 Discussion

We formulated a novel AND/OR graph representation capable of describing the different configurations of deformable articulated objects. The representation makes use of the summarization principle which distinguishes it from other type of AND/OR graph [3]. We developed a novel compositional inference algorithm for proposing configurations. Surround suppression ensures that the inference time is polynomial in the size of image. We demonstrated that the algorithm was fast and effective as evaluated by performance measures on two public datasets. We learn the parameters of the AND/OR graph in a globally optimal way by extending max-margin structure learning technique developed in machine learning. Advantages of our approach include (i) the

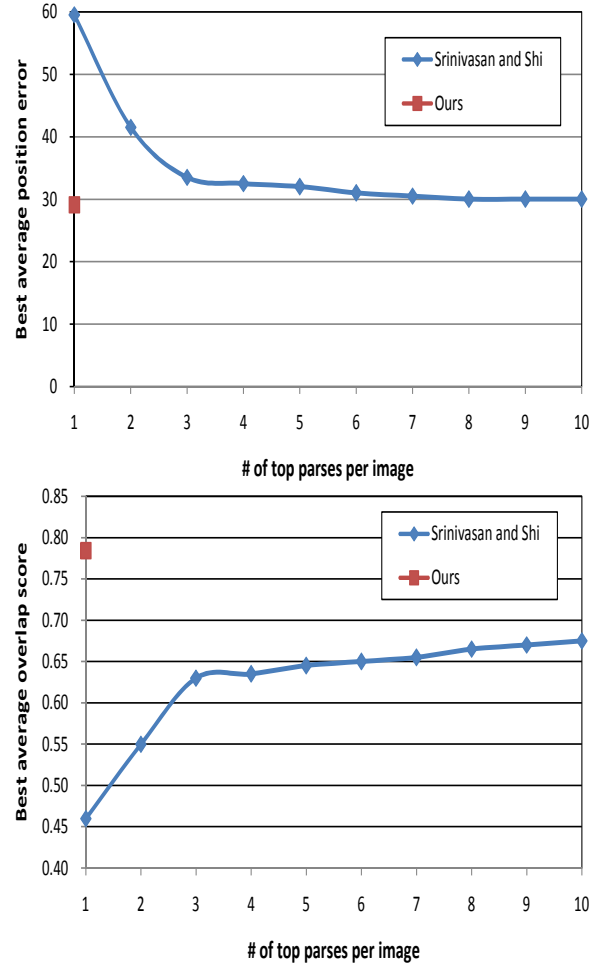


Fig. 11 We compare our results with that of Srinivasan and Shi [8]. The performance of parsing (position error) and segmentation (overlap score) are shown in the top and bottom figures respectively. Note that [8] select the best one (manually) of the top parses.

ability to model the enormous number of poses that occur for articulated objects such as humans, (ii) the discriminative power provided by max-margin learning (by contrast to MLE), and (iii) the use of the kernel trick to make use of high-dimensional features. We gave detailed experiments on the Weizmann horse and human baseball datasets, showing significant improvements over the state-of-the-art methods.



Fig. 10 The first column shows the parse results of the human body. Color points indicate the positions of body parts. The same color points in different images correspond to the same parts. The second column show the segmentations of human body. The next four columns show extra examples.

We are currently working on improving the inference speed of our algorithm by using a cascade strategy. We are also extending the model to represent humans in more details.

Acknowledgements

We gratefully acknowledge support from the National Science Foundation with NSF grant number 0413214 and from the W.M. Keck Foundation.

References

1. C. Manning and H. Schuetze, *Foundations of statistical natural language processing*. Cambridge, Mass, USA: MIT Press, 1999.
2. R. Dechter and R. Mateescu, "And/or search spaces for graphical models," *Artif. Intell.*, vol. 171, no. 2-3, pp. 73–106, 2007.
3. H. Chen, Z. Xu, Z. Liu, and S. C. Zhu, "Composite templates for cloth modeling and sketching," in *CVPR (1)*, 2006, pp. 943–950.
4. Y. Jin and S. Geman, "Context and hierarchy in a probabilistic image model," in *CVPR (2)*, 2006, pp. 2145–2152.
5. S. Zhu and D. Mumford, "A stochastic grammar of images," vol. 2, no. 4, pp. 259–362, 2006.
6. P. A. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
7. X. Chen and A. Yuille, "A time-efficient cascade for real-time object detection: with applications for the visually impaired," in *CVPR*, 2005.
8. P. Srinivasan and J. Shi, "Bottom-up recognition and parsing of the human body," in *CVPR*, 2007.
9. G. Mori, "Guiding model search using segmentation," in *ICCV*, 2005, pp. 1417–1423.
10. Y. Chen, L. Zhu, C. Lin, A. L. Yuille, and H. Zhang, "Rapid inference on a novel and/or graph for object detection, segmentation and parsing," in *NIPS*, 2007.
11. Y. Altun, I. Tschantzaris, and T. Hofmann, "Hidden markov support vector machines," in *ICML*, 2003, pp. 3–10.
12. B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *NIPS*, 2003.
13. B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning, "Max-margin parsing," in *EMNLP*, 2004.
14. J. M. Coughlan and S. J. Ferreira, "Finding deformable shapes using loopy belief propagation," in *ECCV (3)*, 2002, pp. 453–468.
15. H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in *CVPR*, 2000, pp. 2044–2051.
16. S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.
17. M. P. Kumar, P. H. S. Torr, and A. Zisserman, "Obj cut," in *CVPR (1)*, 2005, pp. 18–25.
18. B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *ECCV'04 Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 2004, pp. 17–32.
19. E. Borenstein and J. Malik, "Shape guided object segmentation," in *CVPR (1)*, 2006, pp. 969–976.

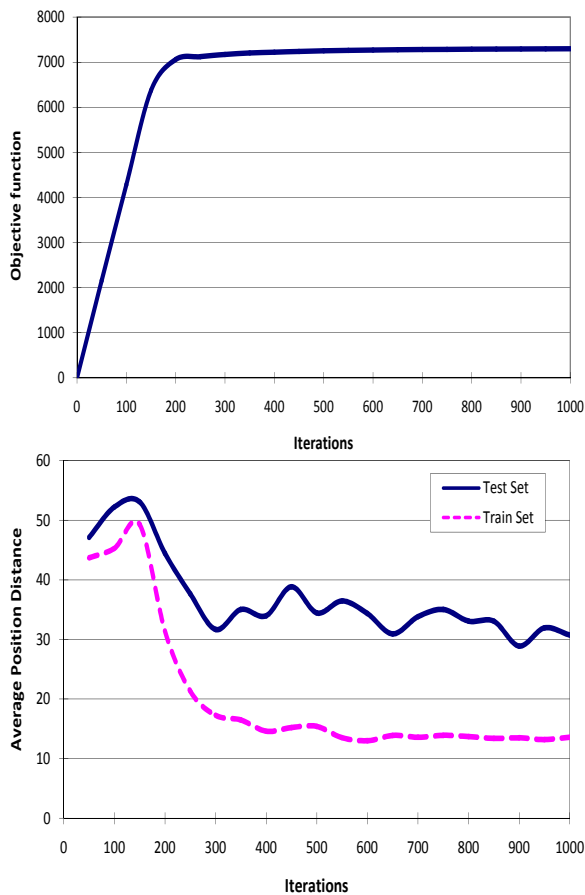


Fig. 12 Convergence Analysis. We study the behavior of max margin training. The first panel shows the convergence curve in terms of the objective function defined in equation (19). The second panel shows the converge curves of the average position error evaluated on training and testing set.

20. T. Cour and J. Shi, "Recognizing objects by piecing together the segmentation puzzle," in *CVPR*, 2007.
21. A. Levin and Y. Weiss, "Learning to combine bottom-up and top-down segmentation," in *ECCV (4)*, 2006, pp. 581–594.
22. J. M. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in *ICCV*, 2005, pp. 756–763.
23. X. Ren, C. Fowlkes, and J. Malik, "Cue integration for figure/ground labeling," in *NIPS*, 2005.
24. M. Meila and M. I. Jordan, "Learning with mixtures of trees," *Journal of Machine Learning Research*, vol. 1, pp. 1–48, 2000.
25. L. Zhu, Y. Chen, and A. L. Yuille, "Unsupervised learning of a probabilistic grammar for object detection and parsing," in *NIPS*, 2006, pp. 1617–1624.
26. L. Sigal and M. J. Black, "Measure locally, reason globally: Occlusion-sensitive articulated pose estimation," in *CVPR (2)*, 2006, pp. 2041–2048.
27. R. Ronfard, C. Schmid, and B. Triggs, "Learning to parse pictures of people," in *ECCV (4)*, 2002, pp. 700–714.
28. X. Ren, A. C. Berg, and J. Malik, "Recovering human body configurations using pairwise constraints between parts," in *ICCV*, 2005, pp. 824–831.
29. D. Ramanan, "Learning to parse images of articulated bodies," in *NIPS*, 2006, pp. 1129–1136.
30. M. W. Lee and I. Cohen, "Proposal maps driven mcmc for estimating human body pose in static images," in *CVPR (2)*, 2004, pp. 334–341.

31. J. Zhang, J. Luo, R. T. Collins, and Y. Liu, "Body localization in still images using hierarchical models and hybrid search," in *CVPR (2)*, 2006, pp. 1536–1543.
32. P. Srinivasan and J. Shi, "Bottom-up recognition and parsing of the human body," in *EMMCVPR*, 2007, pp. 153–168.
33. V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.
34. K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2001.
35. E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *CVPR*, 1997, pp. 130–136.
36. J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001, pp. 282–289.
37. N. Cristianini and J. Shawe-Taylor, *An introduction to support Vector Machines: and other kernel-based learning methods*. New York, NY, USA: Cambridge University Press, 2000.
38. I. Tsochanaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *ICML*, 2004.
39. J. C. Platt, "Using analytic qp and sparseness to speed training of support vector machines," in *NIPS*, 1998, pp. 557–563.
40. E. Borenstein and S. Ullman, "Class-specific, top-down segmentation," in *ECCV (2)*, 2002, pp. 109–124.
41. G. Mori, X. Ren, A. A. Efros, and J. Malik, "Recovering human body configurations: Combining segmentation and recognition," in *CVPR (2)*, 2004, pp. 326–333.
42. C. Rother, V. Kolmogorov, and A. Blake, "'grabcut': interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.