Mixture-of-Parts Pictorial Structures for Objects with Variable Part Sets

Robin Hess and **Alan Fern** and **Eric Mortensen**
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331
{hess,afern,enm}@eecs.oregonstate.edu

Abstract

For many multi-part object classes, the set of parts can vary not only in location but also in type. For example, player formations in American football involve various subsets of player types, and the spatial constraints among players depend largely upon which subset of player types constitutes the formation. In this work, we study the problem of localizing and classifying the parts of such objects. Pictorial structures provide an efficient and robust mechanism for localizing object parts. Unfortunately, these models assume that each object instance involves the same set of parts, making it difficult to apply them directly in our setting. With this motivation, we introduce the mixture-of-parts pictorial structure (MoPPS) model, which is characterized by three components: a set of available parts, a set of constraints that specify legal part subsets, and a function that returns a pictorial structure for any legal part subset. MoPPS inference corresponds to jointly computing the most likely subset of parts and their positions. We propose a restricted, but useful, representation for MoPPS models that facilitates inference via branch-and-bound optimization, which we show is efficient in practice. Experiments in the challenging domain of American football show the effectiveness of the model and inference procedure.

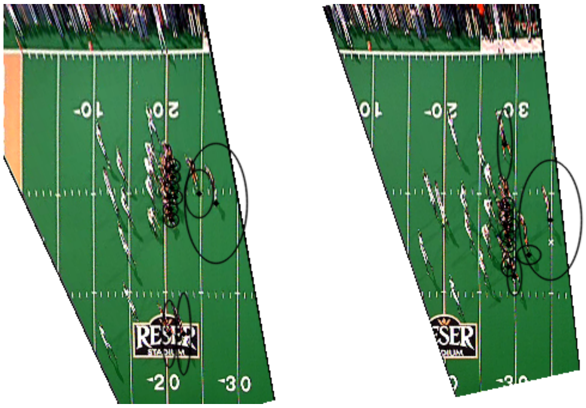
1. Introduction

Pictorial structures are graphical models for representing and localizing objects with multiple spatially related parts. These models represent an object as a set of parts with local appearance models for each part and deformable connections between parts that describe their ideal relative locations. Given a pictorial structure, object recognition/localization corresponds to jointly assigning locations to all parts that minimize the combined local, appearance-based cost of each part plus the deformation cost based on the connections between parts. For restricted—but useful—classes of pictorial structures, efficient algorithms for performing this minimization have been developed that make recognition quite reasonable in practice [3]. By jointly reasoning about part appearances and relative positions, pictorial structures can provide more robust inference than approaches that reason about object parts in isolation, as has been demonstrated for a number of multi-part object recognition problems [1, 8, 4].

A fundamental assumption underlying pictorial structures is that each object instance contains the same set of parts with the same set of deformation constraints among those parts. Unfortunately, this assumption does not hold for many multi-part object classes for which the set of parts can vary not only in location but also in type. Examples of this type of object class include furniture, such as chairs, which can have different types of arms, legs, backs, rockers, etc.; the human figure, along with accessories such as watches, hats, footwear, etc.; houses and other buildings; multi-agent sports scenes; car and airplane types; or any object class for which occlusion can be an issue.

As a specific example, consider our motivating application of recognizing player formations in American football, which can be viewed as multi-part objects, whose parts correspond to players. Each football formation involves various subsets of players, each having a distinct player type corresponding to his role (e.g. left flanker, fullback, center, etc.), and, importantly and as illustrated in Figure 1, the spatial constraints between players are determined largely by the particular subset of players in the formation. This aspect, combined with the fact that appearance is similar across players, makes the relative locations of players the most informative piece of evidence for formation recognition, and the pictorial structure model seems to provide a natural framework for exploiting this information. However, the rules of football enforce certain hard constraints on formations that restrict the number of certain types of players in the formation as well as their spatial configuration, and these factors make it very difficult to formulate a single pictorial structure to recognize all possible football formations. Moreover, because there are thousands of legal formations, formulating a pictorial structure model for each one is practically infeasible and would ignore the significant degree of common structure between similar formations.

In this work we study the problem of recognizing such



(a) (b)

Figure 1: The configuration of players in an American football formation can vary drastically depending on the subset of players in the formation. Above are depicted, mapped to an overhead view, two very different formations containing different subsets of players. Player locations are marked along with confidence ellipses at two standard deviations based on distributions of the relative locations of players. Because player appearances are nearly identical, this variation in structure provides the necessary leverage point for formation recognition.

multi-part objects with variable part sets, or, more specifically, the problem of localizing and classifying the parts of these objects. Our solution is an extension of classical pictorial structures called the mixture-of-parts pictorial structure (MoPPS) model. This model is characterized by three components: a set of available parts, a prior distribution on part subsets that assigns positive probability only to legal part sets, and a function that returns a pictorial structure for any legal part subset. Intuitively, a MoPPS model can be viewed as an implicit representation of a very large collection of pictorial structures that captures the possible variations of objects with variable part sets. Under a generative view of this model, a subset of parts is first drawn from the corresponding prior distribution, then, given the part subset, the corresponding pictorial structure is used to generate locations and appearances for each of the parts. Inference on a MoPPS model corresponds to jointly computing the most likely, or least cost, subset of parts and their locations.

In the absence of special structure, exact inference in MoPPS models is a hopelessly complex combinatorial optimization problem. Therefore, we describe a restricted but reasonable representation for MoPPS models that facilitates their easy specification as well as practically efficient inference. In particular, we represent MoPPS models in terms of a large pictorial tree structure involving all possible parts along with hard constraints on legal part subsets. This representation facilitates the computation of upper and lower cost bounds on part subsets that can be integrated into branch-and-bound style optimization.

To validate the MoPPS model and tree representation, we apply them to the challenging American football formation recognition problem. In a previous attempt to solve this problem, Intille used a knowledge base of purely hard constraints along with a SAT-like procedure for inference [7]. Unfortunately, Intille's method was quite brittle, required significant human pre-processing, and performed poorly enough to be deemed unacceptable for use in later stages of his football understanding system. In contrast, our results show that MoPPS models facilitate accurate recognition and localization in a reasonable time frame without human pre-processing. To our knowledge, there have been no other attempts—in the football domain or otherwise—to solve the recognition problem for objects with variable part sets.

2. Pictorial Structures

Under the classical pictorial structure model, a class of objects is represented as a graph with n vertices $V = \{v_1, \dots, v_n\}$ representing the parts of the object and a set of edges $E = \{(v_i, v_j) \mid v_i \text{ and } v_j \text{ are connected}\}$ representing the connections between parts. Associated with each object class is also a set of model parameters Θ which includes part appearance parameters $A = \{a_1, \dots, a_n\}$ and connection parameters $\Delta = \{\delta_{ij} \mid (v_i, v_j) \in E\}$ describing the ideal relative locations of connected parts. A particular instance of an object is represented as a set of locations of its parts $L = \{l_1, \dots, l_n\}$.

Given an image I and a set of object model parameters Θ, the posterior distribution over the set of part locations is

$$p(L \mid I, \Theta) = \alpha\, p(I \mid L, \Theta)\, p(L \mid \Theta), \tag{1}$$

where α is a normalizing term, $p(I|L,\Theta)$ measures the likelihood of the image data given a particular configuration of the object, and $p(L|\Theta)$ is the prior distribution over object configurations.

Locating a single object in an image corresponds to maximizing (1), and Felzenszwalb and Huttenlocher have shown that if E, $p(I|L,\Theta)$, and $p(L|\Theta)$ satisfy certain, reasonable conditions, then efficient algorithms exist to perform this maximization exactly [3]. Specifically, if the edges in E form a tree and $p(I \mid L, \Theta)$ can be factored as a product of individual part appearance models, then the posterior distribution takes the form

$$p(L \mid I, \Theta) = \alpha \prod_{i=1}^{n} p(I \mid l_i, a_i) \frac{\prod_{(v_i,v_j)\in E} p(l_i, l_j \mid \delta_{ij})}{\prod_{i=1}^{n} p(l_i \mid \Theta)^{\deg(v_i)-1}}, \tag{2}$$

where the $p(I \mid l_i, a_i)$ are individual part appearance models, $p(l_i, l_j|\delta_{ij})$ are priors over relative locations of connected parts, $p(l_i|\Theta)$ are priors over individual part locations, and $\deg(v_i)$ is the degree of vertex v_i.

Under this factorization, finding the optimal configuration L^* of an object corresponds to the following well known cost minimization problem:

$$L^* = \arg\min_{L} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right), \tag{3}$$

where $m_i(l_i) = -\log p(I|l_i, a_i) + (\deg(v_i) - 1) \log p(l_i|\Theta)$ is the local match cost for each part and $d_{ij} = -\log p(l_i, l_j|\delta_{ij})$ is the deformation cost between each pair of connected parts. If $p(l_i, l_j \mid \delta_{ij})$ is Gaussian, then (3) can be computed exactly in $O(hn)$ via distance transforms, where h is the number of possible part locations [3, 2].

3. Mixture-of-Parts Pictorial Structures

As discussed in the introduction, the classical pictorial structure model's assumption of a static part set undermines its ability to recognize some multi-object classes whose parts can vary not only in location but also in type. To overcome this limitation, we introduce in the next two subsections an extension of classical pictorial structures called the mixture-of-parts pictorial structure (MoPPS) model and a specific, restricted MoPPS model representation that facilitates practically efficient inference.

3.1. General MoPPS Model

The MoPPS model is a triple $M = \langle \mathcal{V}, p_v, f \rangle$ where $\mathcal{V}$ is a finite set of parts, p_v is a probability distribution over $2^{\mathcal{V}}$ (i.e. subsets of $\mathcal{V}$), and f is a function that assigns a pictorial structure model to each subset $V \in 2^{\mathcal{V}}$ with $p_v(V) > 0$ (later, we discuss a particular representation for p_v and f). We use Θ_V to denote the parameters of the pictorial structure assigned to part set V and take the vertices and edges of the structure to be implicit in the parameters. Intuitively, a MoPPS model can be viewed as generating image data by first drawing a part subset V according to p_v and generating the image according the generative process dictated by the pictorial structure parameterized by Θ_V.

In the case of American football, the set of parts $\mathcal{V}$ corresponds to all possible players, each of which has a specific role (e.g. fullback, left flanker, shotgun quarterback, etc.). The probability distribution p_v assigns non-zero probability only to those formations that contain exactly 11 parts, the number of players required in a formation, and that obey the formation constraints dictated by the rules of football (e.g. there must be 7 players on the line). Given a legal subset of players V, the corresponding pictorial structure Θ_V encodes the spatial constraints among the players in V along with local observation models for each player. Note that in this domain, the observation models for each player/part are identical since players have very similar appearances.

Given an image I and a MoPPS model $M = \langle \mathcal{V}, p_v, f \rangle$, we are interested in inferring the most likely part set V and the locations L of those parts. The joint posterior distribution over V and L is given by

$$p(L, V \mid I, M) \; = \; \alpha \, p(I \mid L, \Theta_V) \, p(L \mid \Theta_V) \, p_v(V), \tag{4}$$

where α is a normalizing term, $p(I|L, \Theta_V)$ measures the image data likelihood under the pictorial structure model for V, and $p(L|\Theta_V)$ is the corresponding prior distribution over part locations. Note that under this model the marginal probability of the image data can be viewed as a mixture distribution of pictorial structure components, with one per legal subset of parts; hence the name MoPPS.

Let $C(L \mid I, V) = -\log(p(I \mid L, \Theta_V) \, p(L \mid \Theta_V))$ denote the cost assigned to locations L for parts V by pictorial structure Θ_V. We can then write our objective of finding the most likely locations and parts as computing

$$(L^*, V^*) = \arg \min_{(L,V)} C(L \mid I, V) - \log p_v(V). \tag{5}$$

Assuming all pictorial structures Θ_V allow for efficient minimization of $C(L \mid I, V)$, e.g. by assuming tree structures and Gaussian edge potentials, then the primary complexity in this minimization problem is the potentially exponentially large set of part subsets that must be considered. An exhaustive enumeration of these will typically not be tractable. However, if one does not make any assumptions about the MoPPS model then in the worst case exhaustive search is the best we can do (it is straightforward to show NP-completeness). To achieve practically efficient inference, therefore, we developed the MoPPS tree representation for a restricted class of MoPPS models. We present this representation in the next subsection. To simplify the discussion, we will assume for the remainder of the paper that p_v is a uniform distribution over all legal sets of parts. There are straightforward ways in which the inference procedure we describe later can incorporate non-uniform priors.

3.2. The MoPPS Tree Representation

A MoPPS tree representation is a triple $\langle \mathcal{V}, \Theta, T \rangle$, where $\mathcal{V}$ is again a finite set of available parts, Θ is a tree-structured pictorial structure (the *global pictorial structure*) over the entire set of parts, and T is a boolean function that maps each part subset V to either **true** or **false** depending, respectively, on whether or not it is a legal part subset. We will denote by $\Theta|_V$ the projection of Θ onto V, which is just the subgraph of Θ induced by the part set V. Given a MoPPS tree representation the corresponding MoPPS model is given by $\langle \mathcal{V}, p_v, \Theta|_V \rangle$, where p_v is uniform over subsets V with $T(V) =$ **true**.

This representation can be viewed as compactly specifying $f(V) = \Theta|_V$ using the global pictorial structure by returning the projection of part set V onto this structure for any legal V. The set of pictorial structures allowed by this model is constrained so that the pictorial structures returned for any two part sets V and V' must be consistent for parts in $V \cap V'$. Furthermore the pictorial structures $\Theta|_V$ will all be tree structured. An important property of this representation utilized in the inference procedure described in the next section is the monotonicity of the pictorial structure cost function. In particular, if $C^*(I, V) = \min_L C(L \mid I, V)$ is the minimum pictorial structure cost for part set V, then for any part subsets (legal or illegal) V and V', if $V \subseteq V'$ then $C^*(I,V) \leq C^*(I,V')$.

Clearly MoPPS trees cover only a subclass of possible MoPPS models. Intuitively, they are unable to represent object classes for which the spatial relationships between parts are not pairwise independent. Also, MoPPS trees cannot represent models in which one legal part set is a subset of another because, due to the monotonicity property of MoPPS trees, the larger part set will always achieve a higher cost and so will never be selected as the best solution. However, despite these restrictions, MoPPS trees are rich enough to represent interesting object classes, as we demonstrate in Section 5, and they provide structure that can be leveraged to help achieve practically efficient inference. Extending to allow for richer subclasses while maintaining practical inference is an interesting direction for future work.

4. MoPPS Inference

Given a MoPPS model M represented as a MoPPS tree $\langle \mathcal{V}, \Theta, T \rangle$ we wish to solve the minimization problem defined in (5). Note that that if we know V^*, then we can efficiently compute L^* via the pictorial structure $\Theta|_{V^*}$. Thus, the fundamental problem here is to compute V^*. Under our assumption of a uniform p_v we can formulate the optimization problem as

$$V^* = \arg \min_{\{V:T(V)\}} C^*(I,V). \tag{6}$$

In other words, we simply wish to find a legal part set with minimum pictorial structure cost among other legal sets.

Our approach to solving this optimization problem is to cast it in the framework of branch-and-bound search (BBS) and to leverage the special structure of the MoPPS tree representation to efficiently compute informative upper and lower cost bounds as required by BBS.

4.1. Branch-and-bound search

Branch-and-bound search is a classical approach to combinatorial optimization that searches through a tree structure in which every node represents a subset of a space of combinatorial objects. Leaves of the BBS tree typically represent singleton sets or single combinatorial structures. As BBS proceeds, it continually expands new tree nodes and prunes any node from consideration whenever it can be proven that all structures it represents are suboptimal. Finding these nodes is done by computing both an upper and a lower bound on the cost of the combinatorial structures represented by each expanded node. A node can be pruned without sacrificing optimality if its lower bound is greater than any other node's upper bound.

In the case of MoPPS inference, the combinatorial objects of interest are legal part sets, and, hence, each node of the search tree represents collections of part sets. Each node is labeled by a set of parts V, indicating that the node represents all legal part sets V' that contain the parts in V. More formally, we assume that a search space $\langle V_0, s \rangle$ is available for a given MoPPS optimization problem, where V_0 represents the initial search node (V_0 is just a set of parts or possibly the empty set), and s is a successor function that for any node of the tree V returns $s(V) = \{V'_1, \ldots, V'_k\}$ where the V'_i are successor part sets of V and it is assumed that the space satisfies $V \subseteq V'_i$ for all successors.

For a particular application it is generally easy to hand-specify the search tree. However, it is also relatively easy to automatically compile such a search from a MoPPS tree representation. In particular, we need only assume the availability of a function T' that returns **true** for a part subset V iff it is a subset of some legal part set. Given the function T' one can automatically specify a space by setting $V_0 = \emptyset$ and then having $s(V)$ return the set of all part sets V' that result from adding one part to V and such that $T'(V')$ is **true**. We take this latter approach in our application.

The other basic elements that must be specified to cast MoPPS inference as BBS are methods for computing an informative upper bound $c_u(V)$ and lower bound $c_l(V)$ on the cost of the set of part sets represented by a search node V.

Given a search space and upper and lower bound functions, we use a best-first search strategy for BBS, which additionally requires an ordering relation $<_o$ with which to maintain a priority queue of encountered search nodes. Under this strategy, each search step removes the first node from the priority queue, expands it according to s, and adds its successors to the priority queue according to $<_o$. Search stops when all search nodes have been eliminated except a single leaf node representing the optimal solution. In our experiments we consider two ordering relations: $<_l$, which orders nodes according to their lower bound, and $<_u$, which orders according to the upper bound. The effect of using different ordering relations is that a better ordering will expand good leaf nodes earlier than a poor one.

Algorithm 1 gives pseudocode for best-first BBS.

4.2. Lower bound computation

An important property resulting from the subset relationship maintained by the successor function s is that any descendent V' of a search node V is a superset of V and hence, due to the monotonicity of the MoPPS tree representation, we have $C^*(I,V) \leq C^*(I,V')$. In particular, the cost of a node V will never be greater than that of any leaf node (i.e. legal part set) under V. This means that to compute a lower bound on the cost of any complete part set represented by V, i.e. the any of the leaf nodes under V, we need only to compute $C^*(I,V)$, which can be done efficiently using the pictorial structure $\Theta|_V$. Thus, one choice for the lower bound is to take $c_l(V) = C^*(I,V)$.

Algorithm 1 Best-first branch-and-bound MoPPS tree search

```
Input: $\langle V_0, s\rangle$ – Input search space
       $<_o$ – Ordering relation
       $c_l$ – Lower bound function
       $c_u$ – Upper bound function
       $I$ – Input image

$c^* \leftarrow \infty$                              // initialize minimum cost
$Q \leftarrow$ NIL
ENQUEUE$(Q, V_0, <_o)$                               // initialize priority queue
                                                        with initial search node
repeat
    $V \leftarrow$DEQUEUE$(Q)$                       // get best node on queue
    if $s(V) = \emptyset$ then
        if $C^*(I, V) < c^*$ then                    // check for minimum cost leaf node
            $c^* \leftarrow C^*(I, V)$
            $V^* \leftarrow V$
        end if
        if $\forall\, V'$ in $Q$,  $c^* \le c_l(V')$ then
            RETURN $V^*$
        end if
    else
        $\{V'_1, \ldots, V'_k\} \leftarrow s(V)$     // expand $V$
        for $i \leftarrow 1..k$ do
            ENQUEUE$(Q, V'_i, <_o)$
        end for
        PRUNE$(Q, c_l, c_u)$                         // prune dominated nodes in $Q$
    end if
until forever
```

This lower bound can be easily improved in cases where one can find out the minimum number of parts in any leaf node under V. This is straightforward in the football domain since each formation must contain exactly 11 players. In general, suppose that the minimum size leaf node has k additional parts beyond V, and let $C_v^* = \min_{v \notin V} C^*(I, \{v\})$ denote the minimum cost of any pictorial structure $\Theta|_{\{v\}}$, where v is a part that is not in V (note that each such cost is based only on the corresponding part's local match cost). It is straightforward to verify that in this case $c_l(V) = C^*(I, V) + k\, C_v^*$ is still a lower bound.

4.3. Upper bound computation

The main idea of our upper bound calculation is to quickly find a legal set of parts V_u that is a superset of the current node V and that we expect will have low (though perhaps not optimal) cost. If we can find such a set of parts, then we can use $C^*(I, V_u)$ as an upper bound on the cost of V. The key then is to quickly compute V_u, which we can do by leveraging the MoPPS tree representation.

In particular, prior to search, we use the global pictorial structure Θ to compute locations $\mathcal{L}$ for the entire set of parts $\mathcal{V}$. Then, to compute an upper bound on the cost of a node V during BBS, we select V_u as the minimum cost legal subset of $\mathcal{V}$ containing V with the location of each part in V_u fixed at the one specified in $\mathcal{L}$. That is, we select the V_u that minimizes $C(\mathcal{L}[V_u] \mid I, V_u)$ such that $V \subseteq V_u \subseteq \mathcal{V}$, $T(V_u) =$ **true**, and where $\mathcal{L}[V_u]$ is the set of locations in $\mathcal{L}$ for parts in V_u. We can then use $c_u(V) = C(\mathcal{L}[V_u] \mid I, V_u)$ as an upper bound on the cost of V. This upper bound may be tightened at the expense of an extra pictorial structure optimization by computing $c_u(V) = C^*(I, V_u)$.

The key to this upper bound is the fact that evaluating $C(\mathcal{L}[V_u] \mid I, V_u)$ for different subsets V_u is many orders of magnitude faster than computing V^*, which involves optimization over both locations and part sets. This permits for the search for the optimal V_u to be done via another branch-and-bound search or exhaustively, if computationally feasible. If exact optimization of V_u is still too costly, V_u may be approximated with a greedy, approximate hill-climbing search which at every step selects from the parts remaining in $\mathcal{V}$ the minimum cost part that does not make V_u an illegal part set. Such an approximation will typically yield a useful upper bound, though this will not always be the case. Ultimately, if a legal part set V_u has a low cost relative to $C(\mathcal{L} \mid I, \mathcal{V})$ (above) and $c_l(V)$ (below), it is likely that that V_u is a reasonably good part set.

5. Experiments in American Football

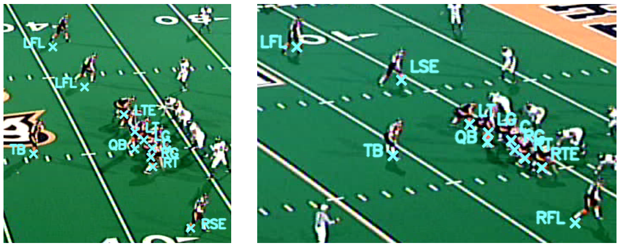
In this section, we demonstrate the capability of the MoPPS tree model by applying it to the challenging American football formation recognition problem. Our goals in this domain are to classify the players that constitute the formation as well as to determine their locations. This is an interesting problem, considering that all professional and most major college football teams employ crews of video scouts who spend many man hours each week using specialized software to manually label opponent video by formation and other factors to allow for content-based queries by coaches. Thus, a semi-automated system for this task would have commercial impact potential. Interestingly, the imagery we use in our experiments comes directly from the video used by the Oregon State University football team.

5.1. Domain Description

The dataset on which we tested the MoPPS tree model contains 25 images of the initial formations of American football plays.[1] Each formation consists of 11 players who may be one of 16 basic types. The rules of football impose certain restrictions on formations such as the requirement that there be at exactly seven players on the line of scrimmage (the imaginary line between offense and defense), the requirement that the rest of the players be at least one yard behind the line of scrimmage, and the requirement that there be a quarterback and five down linemen.

The images in our dataset depict several various formations, but as illustrated in Figure 2 (a) and (b), the differences between them are sometimes very subtle. However, because player appearances are very similar in our low-resolution imagery, these cannot be used as an indicator of

[1] An expanded version of this labeled dataset is available at `http://eecs.oregonstate.edu/football/formations/dataset/`.



(a) (b)

Figure 2: Some formations in American football differ only very subtly. The offensive formations depicted above (the orange and black players, with inferred locations and types overlaid) are two such ones. These formations differ by three players, but the differences between their spatial configurations are very slight and may be difficult even for an untrained human eye to detect. Still, as shown, the MoPPS tree model correctly locates and classifies all of the players in both images.

player identity. Instead, we must rely on the relative spatial configuration of the players, which is determined by the particular subset of players that constitutes the formation.

As discussed above, classical pictorial structures cannot cope with the variation in player types in the class of American football formations. Intille attempted to solve the football formation recognition problem [7], but his recognition system had many major shortcomings. For instance, whereas we attempt to jointly compute the most likely set of players and their locations, Intille's system took as input a set of manually specified player locations and attempted only to assign player type labels to those locations. To do this, Intille manually constructed a knowledge base of hard constraints, such as "near", "to the left of", and "bit of vertical space between", relating player locations—in itself an enormously time consuming and tedious task. He used this knowledge base to cast formation labeling as a SAT-like problem that was solved approximately using a number of search heuristics. In the end, the results of this system were poor, largely because of the strong numerical aspects of the problem, and could not be used in later stages of Intille's football understanding system.

5.2. MoPPS Model for Football Formations

For formation recognition, we use a MoPPS tree model with a total of 34 available parts corresponding to the 16 basic player types as well as several subtypes that capture different attributes of certain players (such as whether the quarterback is in shotgun formation or under center). These parts, subject to a set of hard constraints based on the rules of football, combine to form over 3200 legal formations.

Each image in our dataset can be automatically registered to an overhead view of the football field, as depicted in Figure 1, using the technique described in [5], allowing us to model the relative locations of players in 2D football field coordinates. Specifically, the connection parameters δ_{ij} are the mean and diagonal covariance of a Gaussian distribution over each player's ideal location in field coordinates relative to a "parent" player in the MoPPS tree. These parameters were manually set using a small set of training images.

Each player is treated as being identical in appearance, and the observation model $p(I|l_i, a_i)$ for players is a combination of two models: one, $p_b(I|l_i, a_i)$, based on background segmentation and another, $p_h(I|l_i, a_i)$, based on color histogramming.

To compute $p_b(I|l_i, a_i)$, we register a large collection of football video with the planar overhead field model and, for each pixel in the model, draw a set of samples uniformly from the set of all RGB values that register to that pixel. This sample set is used to compute a kernel density estimate of the field color distribution for the pixel under the (valid) assumption that the pixel exhibits it's true field color for all but a small fraction of the video frames in which it appears. This process is repeated for every pixel in the field model. Player likelihoods for the image I are computed using this model by projecting I into field model space and computing the probability of each pixel under its corresponding field color distribution. Pixels whose probability is below a manually specified threshold are considered foreground pixels, and all others are considered background pixels. The likelihood $p_b(I|l_i, a_i)$ is computed for each pixel in the original image space as the proportion of foreground pixels within a player-sized rectangular region anchored to that pixel at the bottom center (to put high likelihood at players' feet), and these likelihood values are projected back into field model space for compatibility with the structure model.

Because $p_b(I|l_i, a_i)$ does not differentiate between players on opposing teams, we also use an HSV histogram-based model $p_h(I|l_i, a_i)$ to help separate the players on the team of interest from the players on the other team. To compute $p_h(I|l_i, a_i)$, we use the method described by Pérez *et al.* in [9]. Specifically, we compute a reference histogram of HSV player color using a small set of manually segmented player regions. The likelihood $p_h(I|l_i, a_i)$ is computed at each pixel in the original image space based on the similarity between the reference histogram and the histogram defined by the player-sized rectangular region anchored to that pixel at the bottom center. These likelihoods are also projected back into field model space.

Unfortunately, the simple combination of these two models is imperfect because it can over count evidence. Some authors attempt to mitigate the effects of overcounting by applying a smoothing factor to the observation likelihood [3, 10]. However, we have found that this approach accentuates false peaks in the observation likelihood that are due to slight errors during registration with the field model. Instead, we apply a multiplicative reward term β_i to the observation likelihoods of players whose ideal locations make over counting the evidence associated with them unlikely. Thus, the final player likelihood $p(I|l_i, a_i)$ is the product of $p_b(I|l_i, a_i)$, $p_h(I|l_i, a_i)$, and β_i, and the appearance parameters a_i for each player are the background color model, the HSV histogram model, and β_i.

5.3. Search strategies considered

In our experiments, we consider two different variants of best-first BBS. The first of these, referred to below as LB BBS, uses ordering relation $<_l$ and the lower bound function described in Section 4.2. Because a best-first search ordered by $<_l$ must consider all nodes V with $c_l(V) < C^*(I, V^*)$, the first leaf node drawn from the priority queue in LB BBS necessarily corresponds to V^*, and no nodes before V^* can be pruned. For this reason, we simply use constant ∞ as an upper bound function for LB BBS to avoid the cost of a more expensive computation. The second variant of BBS, referred to below as UB BBS, uses ordering relation $<_u$ and the lower and upper bound functions described in Sections 4.2 and 4.3, respectively. For comparison, we also consider exhaustive search and greedy, approximate hill-climbing as described at the end of Section 4.3.

5.4. Results

Table 1 summarizes the quantifiable statistics of the search procedures we consider for MoPPS inference. We use two different metrics to quantify error in predicted location, both of which compare the set of player types V^* and associated locations L^* inferred by the MoPPS model to a corresponding set of hand-labeled ground-truth player types and locations. The first metric, $e_c(V^*, L^*)$, computes the mean pixel distance between the locations of correctly classified players and the corresponding ground-truth locations. The second metric, $e_a(V^*, L^*)$, associates ground-truth locations with incorrectly classified players by finding the minimum matching between the locations of incorrectly classified players and ground-truth locations not associated with correctly classified players and then computes the mean pixel distance between the locations of all players and their associated ground-truth locations.

A far more important measure of performance in the football formation recognition domain is the percentage of correctly classified players. This is because the huge number of possible formations precludes naming all of them, so every football team uses their own language to describe formations. Player type information, however, can be translated into any team's formation language. Thus, the ability to correctly recognize which players are on the field along with special attributes of some (e.g. whether the quarterback is in shotgun formation) is akin to the ability to correctly recognize entire formations. This information, therefore, would be used to index plays in a coach's database.

Because best-first BBS is an optimal search, the location error rates and player classification rate for LB BBS and UB BBS are the same as those achieved through exhaustive search. By both measures, MoPPS inference with these methods is very accurate.

In particular, they achieve a mean location error rate of 4.36 pixels for correctly classified players. Considering that six pixels in our field model are equal to one yard on the football field, this error rate is quite good. Moreover, many of the higher individual errors we observed can be attributed to the fact that the low resolution of our imagery led the MoPPS inference method to locate some players at their waist instead of at their feet.

Even more striking is the 98.55% correct classification rate these methods achieved, representing a total of four misclassifications out of a possible 275 players. In each of of these cases, the misclassified player was placed either on a false peak in the observation likelihood around one of the several logos on the field, all of which are composed of colors identical to the ones in the players uniforms, or on a second peak in the likelihood generated by a single player.

Of course, both LB BBS and UB BBS considerably outperform exhaustive search in terms of running time, with LB BBS beating UB BBS by about a factor of two. To obtain further insight on the running times of LB BBS and UB BBS, we measured their anytime behavior, which is plotted in Figure 3. Both search strategies perform similarly in terms of location error, achieving very low error rates with one minute of computation and near-optimal rates within two minutes. However, in terms of the percentage of correctly classified players, LB BBS consistently outperforms UB BBS for any given amount of computation time, again achieving near-optimal results within two minutes. This can be mostly attributed to the fact that UB BBS must spend additional processing time during the upper bound computation at every node searching for the optimal superset V_u and tightening the bound via pictorial structure minimization.

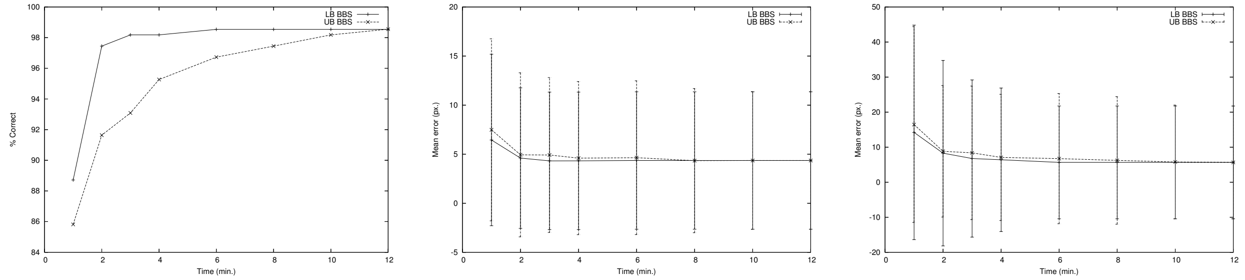
Overall, these results are very promising. While it is true that our dataset of 25 images directly represents only a small fraction of the over 3200 possible football formations, we are reassured by the fact that many other formations can be composed by combining correctly recognized pieces of our 25, suggesting the MoPPS model will likely work well for these other formations, too. In addition, informal evaluation on unlabeled images convinces us that the model is robust outside the test set.

6. Summary and Future Work

In this paper we introduced the mixture-of-parts pictorial structure (MoPPS) model for recognizing object classes whose parts can vary in both location and type. We formulated a restricted but reasonable tree-structured representation of the MoPPS model and described how practically efficient inference could be performed on that model to jointly compute the most likely set of parts and their locations. Finally, we demonstrated the effectiveness of the model and inference procedure through experiments in the challenging

Search	**Running time (min.)**			**Nodes Expanded**			**% Correct**	$e_c(V^*, L^*)$			$e_a(V^*, L^*)$		
Strategy	Mean	Min.	Max.	Mean	Min.	Max.	**Class.**	Mean	Std.	Max.	Mean	Std.	Max.
LB BBS	4.35	0.48	11.18	392	51	988	98.55 %	4.36	7.00	17.46	5.65	16.09	37.11
UB BBS	9.00	1.44	24.14	412	57	1148	98.55%	4.36	7.00	17.46	5.65	16.09	37.11
UB BBS, 1^{st} Leaf	2.45	1.90	5.23	52	36	110	92.00 %	4.72	7.96	27.20	9.36	23.27	66.07
Greedy	0.57	0.53	0.63	11	11	11	80.72 %	9.14	8.01	47.10	19.33	28.42	167.05
Exhaustive	41.69	41.40	41.92	3264	3264	3264	98.55 %	4.36	7.00	17.46	5.65	16.09	37.11

Table 1: This table summarizes the quantifiable statistics of various search strategies for MoPPS inference over the entire dataset of 25 images. Location error rates are in pixel units, six of which in our field model are equal to one yard on the football field. The optimal searches, LB BBS, UB BBS and exhaustive search yield excellent results in terms of both location error and classification rate. However, LB BBS provides a significant speedup over UB BBS, which is naturally much faster than exhaustive search. Halting UB BBS as soon as it encounters its first leaf node yields good results within a modest time frame, and greedy, approximate hill-climbing search yields reasonable results fairly quickly.



(a) % Correct Classifications

(b) Mean $e_c(V^*, L^*)$

(c) Mean $e_a(V^*, L^*)$

Figure 3: The plots above depict the anytime behavior of MoPPS inference with LB BBS and UB BBS over the entire dataset of 25 images in terms of (a) the percentage of correctly classified players and (b) & (c) the mean location error rates. For both strategies, a solution was computed using greedy, approximate hill-climbing search whenever a complete solution was not found in the alloted time. While both search strategies perform well in terms of location error, LB BBS clearly outperforms UB BBS in terms of the percentage of players it classifies correctly within a given amount of time. This is notable because classification accuracy the most important measure for our application.

American football formation recognition domain.

We believe the MoPPS model will be generally useful whenever detailed internal object structure is needed and not just object existence/location. An important direction for future work will be to evaluate the merits of this model in terms of its expressiveness and computational speed on other recognition domains, such as furniture, for example chairs, which can be composed of various types of legs, arms, backs, rockers, etc.; the human figure, along with accessories such as hats, watches, footwear, etc., of which exponentially many combinations are possible; specific types of cars, which can have exponentially many combinations of different parts like spoilers, rims, etc.; and similarly for specific types of airplanes. In addition, we believe the MoPPS paradigm is particularly well suited for coping with occlusion during general object recognition and localization and would like to explore its capacity in this regard.

Many other directions for future work exist including extending MoPPS to richer representations than trees, such as k-fans [1]; incorporating richer sets of hard or nearly-hard constraints and logic-based reasoning; developing a part set prior that incorporates the image data to permit more efficient inference; and developing proposal distributions for MCMC sampling methods to allow for probabilistic queries (e.g. "what is the probability there is a tight end?"). There are also several opportunities to incorporate learning into the MoPPS paradigm, which we discuss in detail in [6].

Acknowledgments

We would especially like to thank the coaches and staff of the Oregon State University football team for providing us with our video data as well as the reviewers of this paper for helpful comments and suggestions. We are also grateful to NSF for financial support under grant IIS-0307592.

References

[1] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. *Object Recognition by Combining Appearance and Geometry*, volume 4170/2006 of *LNCS*, pages 462–482. Springer, 2006.

[2] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical Report TR2004-1963, Cornell Computing and Information Science, 2004.

[3] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1), 2005.

[4] R. Fergus, P. Perona, and A. Zisserman. A sparse object category for efficient learning and exhaustive recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[5] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[6] R. Hess and A. Fern. Toward learning mixture-of-parts pictorial structures. In *The ICML 2007 Workshop on Constrained Optimization and Structured Output Spaces*, 2007.

[7] S. Intille. *Visual Recognition of Multi-Agent Action*. PhD thesis, MIT, 1999.

[8] X. Lan and D. Huttenlocher. Beyond trees: common-factor models for 2D human pose recovery. In *Proc. IEEE Int'l Conf. on Computer Vision*, 2005.

[9] P. Pérez, C. Hue, J. Vermaak, and M. Ganget. Color-based probabilistic tracking. In *Proc. European Conf. on Computer Vision*, 2002.

[10] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *Proc. IEEE International Conf. on Computer Vision*, 2003.