A Layered Motion Representation with Occlusion and Compact Spatial Support

Allan D. Jepson[1] David J. Fleet[2] Michael J. Black[3]

[1] Department of Computer Science, University of Toronto, Toronto, Canada
[2] Xerox PARC, 3333 Coyote Hill Rd., Palo Alto, CA 94304, USA
[3] Department of Computer Science, Brown University, Providence, USA

Abstract. We describe a 2.5D layered representation for visual motion analysis that incorporates localized spatial support, occlusion, arbitrary numbers of moving regions, and model selection. A compositional model of the image motion provides a global interpretation of the image observations. The underlying representation is composed of a background and several foreground regions. Each region comprises a parametric shape model, a parametric motion model, and an appearance model. The representation also contains depth ordering so visibility and occlusion are rightly included in the estimation of the model parameters. Finally, because the number of objects, their positions, shapes and sizes, and their relative depths are all unknown, initial models are drawn from a proposal distribution, and then compared using a penalized likelihood model selection criterion. This allows us to automatically initialize new models, and to compare different depth orderings.

1 Introduction

One goal of visual motion analysis is to compute representations of image motion that allow one to infer the structure and identity of moving objects. Towards this end, layered image representations are promising; they provide a natural way of estimating different image regions that move with different velocities, and they are often effective for separating foreground objects from backgrounds [10, 23, 4, 25]. One weakness of current layered representations is that they assign pixels to layers independently of other neighboring pixels. In doing so the underlying generative model does not capture the typical coherence of objects in natural scenes. In this paper we describe an new layered representation in which the layers have specific opaque interiors along with soft boundaries, and thereby provide an explicit representation of their spatial coherence and extent.

Our 2.5D layered representation is motivated by a desire to find effective descriptions of images in terms of a relatively small number of simple moving parts. The representation is based on a composition of layered regions, which are individually called *polybones*. One novel aspect of the representation is that the polybones have compact spatial support and a probabilistic representation for their borders. This, together with a partial depth ordering among the polybones, provides an explicit representation of visibility and occlusions.

Along with this representation we also describe a method for parsing image motion to find global image descriptions in terms of an arbitrary number of layered, moving polybones. Since the number of objects, their positions, motions, shapes, sizes, and relative depths are all unknown, a complete search of this space is infeasible. Instead we employ a stochastic search in which new parses are drawn from a proposal distribution. The parameters of the individual polybones within each such proposal are refined using the EM-algorithm. Alternative parses are then compared using a penalized likelihood model selection criterion. This allows us to automatically explore alternative parses, and to select the most plausible ones.

2 Previous Work

Many current approaches to motion analysis over relatively long image sequences are formulated as model-based tracking problems. In most cases one has a significant amount of prior knowledge about the objects of interest. For example, one typically has knowledge of the number of objects, the appearance of objects, a model for object dynamics, and perhaps an initial guess about object position. This includes 3D people trackers which assume detailed kinematic models of shape and motion, and initialization is still often done manually [2, 3, 6, 18]. Recent success with curve-tracking of human shapes also relies on a user specified model of the desired curve [9, 14]. For even more complex objects under variable illuminants it has been common to learn models of object appearance from training sets of images prior to tracking [7, 1]. Whether one tracks blobs to detect activities like football plays [8], or specific classes of objects such as blood cells, satellites or hockey pucks, it is comon to constrain the problem with a suitable model of object appearance and dynamics, along with a relatively simple form of data association [13, 15].

In contrast to model-based methods, layered representations offer a promising approach to generic motion analysis, with relatively weak models of scenes and their dynamics [10, 17, 22, 25]. These approaches have led to efficient estimation techniques and effective means for comparing different models. In particular, they provide one with the ability to softly assign pixels to layers, and to robustly estimate the motion parameters of each layer. However, in most of these approaches the assignment of pixels to layers is done independently at each pixel, without an explicit constraint on spatial coherence (although see [24]).

Our goal here is to develop a compositional representation for image motion with a somewhat greater degree of generic expressiveness than existing layered models. Broadly speaking, we seek a representation that satisfies three criteria: 1) it captures the salient structure of the time-varying image in an expressive manner; 2) it allows us to generate and elaborate specific parses within the representation in a computationally efficient way; and 3) it allows us to compare different parses in order to select the most plausible ones.

Towards this end, like previous work in [20, 21], we assume a relatively simple parametric model for the spatial support of each layer. However, unlike the Gaussian model in [20], where the spatial support decays exponentially from the center of the object, we use a *polybone* in which support is unity over the

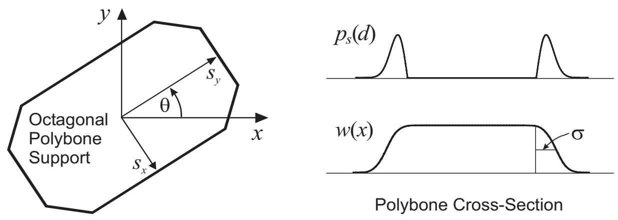

Fig. 1. (left) The spatial support of each polybone used in the experiments is a simple transform of a canonical octagonal shape. The allowed transforms include translation, rotation and independent scaling along two object-centered axes. (right) These plots depict boundary position density $p_s(d)$ and the occupancy probability $w(x)$. The occupancy probability is unity inside the polygon with a Gaussian-shaped soft shoulder.

interior of the object. The support smoothly decays to zero only in the vicinity of the spatial boundary to reflect our uncertainty about exact boundary position. This allows us to separate changes in object size and shape from our uncertainty about the boundary location, and it allows us to differentiate the associated likelihood function with respect to object shape and position. Most importantly, it allows us to explicitly express properties like visibility, occupancy, opacity and occlusions in a straightforward way.

One of the central issues in this research is whether or not the extraction and selection of a polybone description for image motion data is computationally feasible. Since the space of possible descriptions is large (owing to the unknown number of polybones, their unknown depth relationships and their unknown shape and motion parameters), we require efficient computational techniques to search this space and to compare the plausibility of various alternatives.

3 Layered Polybones

The polybone image model consists of a background layer and K depth-ordered foreground layers. Formally, the model $\mathbf{M}$ at time t can be written as

$$\mathbf{M} \;=\; (K(t),\, \mathbf{b}_0(t),\, ...,\, \mathbf{b}_K(t)) \;, \tag{1}$$

where $\mathbf{b}_k \equiv (\mathbf{a}_k, \mathbf{m}_k)$ is the vector of shape and pose parameters, $\mathbf{a}_k$, together with the motion parameters, $\mathbf{m}_k$, for the k^{th} polybone. By convention, the partial depth ordering is given by the order of the polybone indices. The background corresponds to $k = 0$, and the foremost polybone corresponds to $k = K$.

In the interests of a simple shape description, the interior of each polybone is defined by a closed convex polygon. These interior regions are assumed to be opaque, so anything behind a polybone interior is occluded. Given the simplicity of the polybone shape, we do not expect them to fit any particular region accurately. We therefore give each polybone a soft border to quantify our uncertainty in the true boundary location. More precisely, let p_s denote the probability density of the true boundary location, defined as a simple function of the distance, $d(\mathbf{x}; \mathbf{b}_k)$, from a given location $\mathbf{x}$ to the polygon specified by the parameters $\mathbf{b}_k$ (see Fig. 1(right)). Given this boundary density, the probability that $\mathbf{x}$ lies inside the true boundary is given by the occupancy probability,

$w(\mathbf{x}; \mathbf{b}_k) = p_s(d > d(\mathbf{x}; \mathbf{b}))$, that is, the cummulative probability that the true boundary is further than $\mathbf{x}$.

The occupancy probability, $w(\mathbf{x}; \mathbf{b}_k)$, provides our definition of spatial support, from which we define *visbility* and *occlusion*. As depicted in Figure 1(right), we model p_s so that the occupancy probability, $w(\mathbf{x}; \mathbf{b}_k)$, is unity in the interior of the polygon, and decays outside the polygon with the shape of a half-Gaussian function of the distance from the polygon interior. For convenience, the standard deviation of the half-Gaussian, $\sigma_{s,k}$, is taken to be a constant.[4]

With this definition of occupancy, the visibility of the j^{th} polybone at a pixel $\mathbf{x}$ is the product of the probabilities that closer layers do not occupy that pixel. That is, the visibility probability is

$$v_k(\mathbf{x}) \;=\; \prod_{j=k+1}^{K} (1 - w(\mathbf{x};\, \mathbf{b}_j)) \;=\; (1 - w(\mathbf{x};\, \mathbf{b}_{k+1}))\; v_{k+1}(\mathbf{x}) \; . \qquad (2)$$

All pixels in the foremost layer K are defined to be visible, so $v_K(\mathbf{x}) = 1$. Transparency could also be modelled by replacing $(1 - w(\mathbf{x};\, \mathbf{b}_j))$ in (2) by $(1 - \mu_j w(\mathbf{x};\, \mathbf{b}_j))$, where $\mu_j \in [0, 1]$ denotes the opacity of the j^{th} polybone (cf. [19, 5]). Previous layered models correspond to the special case of this in which $\mu_j = 0$ and each polybone covers the entire image, so $w(\mathbf{x}; \mathbf{b}_k) \equiv 1$.

In our current implementation we restrict the polybone shape to be a simple transform of a canonical octagonal boundary (see Fig. 1(left)). The shape and pose of the interior polygon is parameterized with respect to its local coordinate frame, with its scale in the horizontal and vertical directions $\mathbf{s} = (s_x, s_y)$, its orientation θ, and the image position of the polygon origin, $\mathbf{c} = (c_x, c_y)$ (see Fig. 1). Together with the soft boundary parameter, σ_s, these parameters define the shape and pose of a polybone:

$$\mathbf{a}_k \;=\; (\mathbf{s}_k, \theta_k, \mathbf{c}_k, \sigma_{s,k}) \; . \qquad (3)$$

Finally, note that our simple descriptions for shape and pose were selected, in part, to simplify the exposition in this paper and to facilitate the parameter estimation problem discussed below. It would be straightforward to include more complex polygonal or spline based shape descriptions in the representation (although local minima in the optimization may present more of a problem).

In addition to shape and pose, the polybone parameters also specify the motion within the layer. Here, we use a motion mixture model to represent the displacement field from one time to the next. Accordingly, the parameters associated with the k^{th} polybone, denoted by $\mathbf{m}_k$, specify an image warp $\mathbf{w}(\mathbf{x}; \mathbf{m}_k(t))$ from pixels in frame $t+1$ to pixels in frame t. In the current implementation we use similarity displacement fields, where $\mathbf{m}_k$ specifies translation, rotation and uniform scaling between frames.

[4] We truncate polybone shoulders to zero after a distance of $2.5\sigma_s$, and $\sigma_s = 4$ pixels was used in all the examples.

4 Model Likelihood

The likelihood of a given model $\mathbf{M}_t$ is defined in terms of how well it accounts for the motion between frames t and $t+1$. As is common in optical flow estimation, we formulate a motion likelihood function from a simple data conservation assumption. That is, let $d(\mathbf{x}, t)$ denote image data at pixel $\mathbf{x}$ and frame t. The warp parameters for the k^{th} polybone specify that the corresponding point in the subsequent frame is at $(\mathbf{x}\prime, t+1) = (\mathbf{w}(\mathbf{x}; \mathbf{m}_k(t)), t+1)$. The similarity of the data at these two points is typically measured in terms of a probability distribution for the difference, $\delta d_k(\mathbf{x}, t) = d(\mathbf{w}(\mathbf{x}; \mathbf{m}_k(t)), t+1) - d(\mathbf{x}, t)$.

The distribution for the deviation δd is often taken to be a Gaussian distribution, say $p_1(\delta d)$, having mean 0 and standard deviation σ_m. Alternatively, in order to accomodate outliers, a mixture of such a Gaussian distribution with a broad ourlier distribution, $p_0(\delta d)$, is used. Indeed, mixture distributions have been found to enable robust motion estimation in the face of outliers and unmodelled surfaces [10]. Using this latter approach, we then define the likelihood associated with a single data observation $\delta d_k(\mathbf{x}, t)$ to be

$$p_k(\delta d_k(\mathbf{x}, t)) = (1 - \pi_{0,k}) p_1(\delta d_k(\mathbf{x}, t)) + \pi_{0,k} p_0(\delta d_k(\mathbf{x}, t)), \tag{4}$$

where $\pi_{0,k} \in [0, 1]$ is the outlier mixing proportion. The additional parameters required to specify this mixture model, namely σ_m and π_0, are included in the motion parameter vector $\mathbf{m}_k(t)$ for each polybone. Note that, as with the shape and pose parameterization, we have chosen simple forms for the parametric motion model and data likelihood both to simply the exposition and to facilitate parameter estimation.

This likelihood for the k^{th} polybone at a pixel $\mathbf{x}$ can be combined with the other polybones in the model $\mathbf{M}_t$ by incorporating each polybone's visibility, $v_k(\mathbf{x})$, and occupancy probability, $w(\mathbf{x}, \mathbf{b}_k(t))$. It is straightforward to show that the likelihood of the entire polybone model at location $\mathbf{x}$ and frame t is given by

$$p(\delta d(\mathbf{x}, t) \,|\, \mathbf{M}_t) \;=\; \sum_{k=0}^{K} v_k(\mathbf{x})\, w(\mathbf{x}; \mathbf{b}_k)\, p_k(\delta d_k(\mathbf{x}, t)) \;. \tag{5}$$

Finally, given independent data samples, the log likelihood of the polybone model $\mathbf{M}_t$ over the whole image is

$$\log p(\mathbf{D}_t \,|\, \mathbf{M}_t) \;=\; \sum_{\mathbf{x}} \log p(\delta d(\mathbf{x}, t) \,|\, \mathbf{M}_t). \tag{6}$$

Note that the use of $\mathbf{D}_t$ here involves some abuse of notation, since the image data at both frames t and $t+1$ are required to compute the deviations $\delta d(\mathbf{x}, t)$; moreover the model itself is required to determine corresponding points.

5 Penalized Likelihood

In this section we derive the objective function we use for optimizing polybone models and for comparing alternative models. The objective function is motivated by the standard Markov chain formulation for the posterior probability of the model $\mathbf{M}_t$, given all the data up to time t (denoted by $\mathcal{D}_t$). In particular, the posterior probability is proportional to

$$\mathcal{U}(\mathbf{M}_t) = \log p(\mathbf{D}_t \mid \mathbf{M}_t) + \log p(\mathbf{M}_t \mid \mathcal{D}_{t-1}), \tag{7}$$

where the first term is the log likelihood introduced in (6), and the second term is the log of the conditional distribution of the model $\mathbf{M}_t$ given all the previous data. The latter distribution can be expanded as

$$\mathbf{p}(\mathbf{M}_t \mid \mathcal{D}_{t-1}) = \int_{\tilde{\mathbf{M}}_{t-1}} p(\mathbf{M}_t \mid \tilde{\mathbf{M}}_{t-1}) p(\tilde{\mathbf{M}}_{t-1} \mid \mathcal{D}_{t-1}). \tag{8}$$

Given the complexity of the space of models we are considering, a detailed approximation of this term is beyond the scope of this paper. Instead, we use the general form in equations (7) and (8) to motivate a simpler penalized likelihood formulation for the objective function, namely

$$\mathcal{O}(\mathbf{M}_t) = \log p(\mathbf{D}_t \mid \mathbf{M}_t) + q(\mathbf{M}_t, \mathbf{M}_{t-1}). \tag{9}$$

The last term in (9) is called the penalty term, and it is meant to provide a rough approximation for the log of the conditional probability distribution in (8). More to the point, we simply choose $q(\mathbf{M}_t, \mathbf{M}_{t-1})$ to control the selection of which models maximize the objective function $\mathcal{O}(\mathbf{M}_t)$.

The penalty term serves two purposes. First, in cases where the data is absent, ambiguous, or noisy, the log likelihood term can be expected to be insensitive to particular variations in the model $\mathbf{M}_t$. In these situations the penalty term can provide a bias towards particular parameter values, for example, towards small and smoothly deforming polybones.

The second purpose of the penalty function is to control the complexity of the optimal model(s). Without a penalty term in (9) we should expect that strictly more complex models, such as models with an extra polybone, could achieve higher log likelihoods. In particular, the extra degrees of freedom in a model with an extra polybone could, for example, be used to fit additional structure or noise not accounted for by the simpler model. Thus, the maximum of the log likelihood term in (9) is monotonically increasing in the number of polybones. However, beyond a certain point, the extra polybone would primarily fit noise in the data set, and the increase in the log likelihood could be expected to be marginal. To control this we use a the penalty term in (9) that decreases with model complexity, thereby limiting the acceptance of more complex models to those which produce a large enough gain in the log likelihood. The penalty term we use in the experiments is provided in the Appendix.

6 Parameter Estimation

Suppose $\mathbf{M}_t^0$ is an initial guess for the model parameters in the form given in equation (1). In order to find local optima of (9) we use a form of gradient

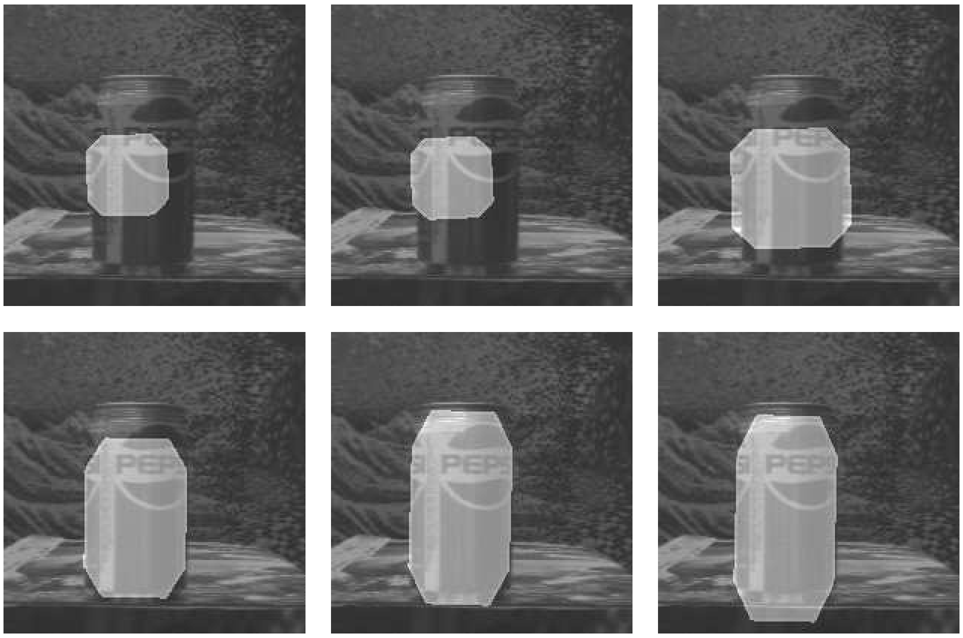

Fig. 2. Growth of a single foreground polybone. The background polybone (not shown) covers the entire image.

ascent. The gradient of the penalty term is straight-forward, while that of the log likelihood term is conveniently computed by exploiting the layered structure of the model. In particular, we first write the likelihood term $p(D(\mathbf{x}) \,|\, \mathbf{M})$ in a form which isolates the dependence on the parameters for the k^{th} polybone, and separates the dependencies on polybones nearer than the k^{th} from those further from the k^{th}.

To do this splitting note that the contribution to $p(D(\mathbf{x}) \,|\, \mathbf{M})$ due to only those polybones that are closer to the camera than the k^{th} bone is, from (2) and (5),

$$\begin{aligned} n_k(\mathbf{x}) &= \sum_{j=k+1}^{K} v_j(\mathbf{x})\, w(\mathbf{x}; \mathbf{b}_j)\, p(D(\mathbf{x})|\mathbf{b}_j) \\ &= v_{k+1}(\mathbf{x})\, w(\mathbf{x}; \mathbf{b}_{k+1})\, p(D(\mathbf{x}) \,|\, \mathbf{b}_{k+1}) + n_{k+1}(\mathbf{x}) \; . \end{aligned} \tag{10}$$

We refer to $n_k(\mathbf{x})$ as the 'near term' for the k^{th} polybone. Equations (10) and (2) provide recurrence relations (decreasing in k) for computing the near terms and visibilities $v_k(\mathbf{x})$, starting with $n_K(\mathbf{x}) = 0$ and $v_K(\mathbf{x}) = 1$. Similarly, we collect the polybones that are further from the camera than the k^{th} polybone

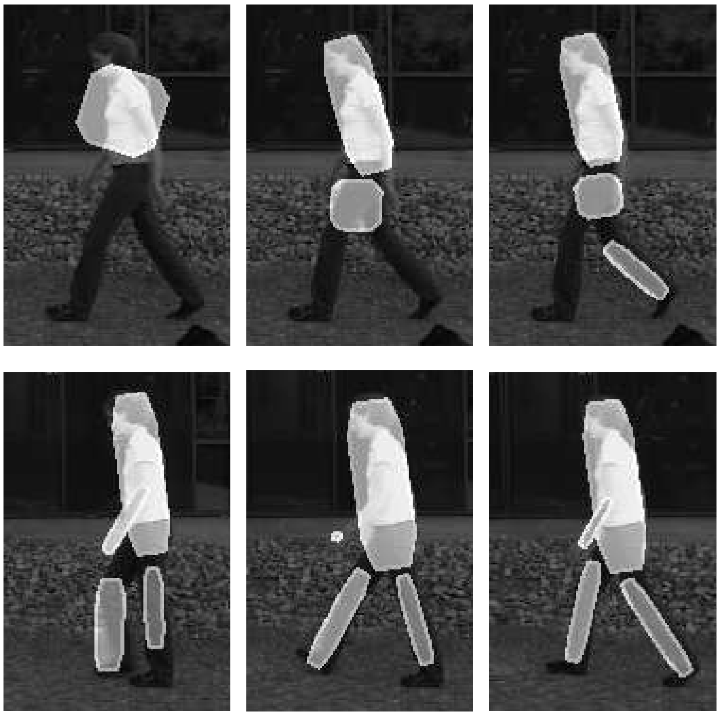

Fig. 3. The development of the optimal known model. The results for (top) the first three frames, namely 50, 51, 52, and (bottom) frames 60, 64, 68 of the sequence.

into the 'far term',

$$
\begin{aligned}
f_k(\mathbf{x}) &= \sum_{j=0}^{k-1} w(x;\mathbf{b}_j) \left[\prod_{l=j+1}^{k-1} (1 - w(\mathbf{x};\mathbf{b}_l)) \right] p(D(\mathbf{x})|\mathbf{b}_j) \\
&= w(\mathbf{x};\mathbf{b}_{k-1})\, p(D(\mathbf{x}) \,|\, \mathbf{b}_{k-1}) + (1 - w(\mathbf{x};\mathbf{b}_{k-1}))\, f_{k-1}(\mathbf{x}) \; . \qquad (11)
\end{aligned}
$$

Here we use the convention that $\sum_{j=n}^{m} q_j = 0$ and $\prod_{j=n}^{m} q_j = 1$ whenever $n > m$. Notice that (11) gives a recurrence relation for f_k, increasing in k, and starting with $f_0(\mathbf{x}) = 0$.

It now follows that, for each $k \in \{0, \ldots, K\}$, the data likelihood satisfies

$$
\begin{aligned}
p(D(\mathbf{x}) \,|\, \mathbf{M}) \; &= n_k(\mathbf{x}) + v_k(\mathbf{x})\, w(\mathbf{x};\mathbf{b}_k)\, p(D(\mathbf{x}) \,|\, \mathbf{b}_k) \\
&\quad + v_k(\mathbf{x})(1 - w(\mathbf{x};\mathbf{b}_k)) f_k(\mathbf{x}) \; . \qquad (12)
\end{aligned}
$$

Moreover, it also follows that $n_k(\mathbf{x})$, $v_k(\mathbf{x})$, and $f_k(\mathbf{x})$ do not depend on the parameters for the k^{th} polybone, $\mathbf{b}_k$. That is, the dependence on $\mathbf{b}_k$ has been isolated in the two terms $w(\mathbf{x};\mathbf{b}_k)$ and $p(D(\mathbf{x}) \,|\, \mathbf{b}_k)$ in (12), and so the gradient of with respect to $\mathbf{b}_k$ can be easily derived.

The gradient of $\mathcal{O}(\mathbf{M})$ is provided by the gradient of $\log p(\mathbf{D};\mathbf{M})$, which is evaluated as described above, along with the gradient of the penalty term,

$q(\mathbf{M}, \mathbf{M}_{t-1})$. In order to optimize $\mathcal{O}(\mathbf{M})$ we have found several variations beyond pure gradient ascent to be effective. In particular, for a given model $\mathbf{M}$ we use a front-to-back iteration through the recurrence relations (2) and (10) to compute the visibilities $v_k(\mathbf{x})$ and the near polybone likelihoods $n_k(\mathbf{x})$ (from the nearest polybone at $k = K$ to the furthest at $k = 0$), without changing the model. Then, from the furthest polybone to the nearest, we update the k^{th} polybone's parameters, namely $\mathbf{b}_k$, while holding the other polybones fixed. Once $\mathbf{b}_k$ has been updated, we use the recurrence relation (11) to compute the corresponding far term $f_{k+1}(\mathbf{x})$. We then proceed with updating the parameters $\mathbf{b}_{k+1}$, for the next nearest polybone. Together, this process of updating all the polybones is referred to as one back-to-front sweep.

Several sub-steps are used to update each $\mathbf{b}_k$ during a back-to-front sweep. First we update the 'internal' parameters of the k^{th} polybone, namely the motion parameters. This has the same structure as the EM-algorithm in fitting motion mixture models [10], except that in the polybone context the near and far terms contribute to the data ownership computation. The M-step of this EM-algorithm produces a linear problem for the update in the motion parameters, which is solved directly (without using gradient ascent). The mixing coefficients and the variance of the inlier process are also updated using the EM-algorithm. Once these internal parameters have been updated, the pose parameters are updated using a line search along the gradient direction in the pose variables.[5] Finally given the new pose, the internal parameters are re-estimated, completing the update for $\mathbf{b}_k$.

One final refinement involves the gradient ascent in the pose parameters, where we use a line-search along the fixed gradient direction. Since the initial guesses for the pose parameters are often far from the global optimum (see Section 7), we have found it useful to constrain the initial ascent to help avoid some local maxima. In particular, we found that unconstrained hill-climbing from a small initial guess often resulted in a long skinny polybone wedged in a local maximum. To avoid this behaviour we initially constrain the two scaling parameters s_x and s_y to be equal, and just update the mean position (c_x, c_y), angle θ, and this overall scale. Once we have detected a local maximum in these reduced parameters, we allow the relative scales s_x and s_y to evolve to different values.

7 Model Search

While this hill-climbing process is capable of refining rough initial guesses, the number of local maxima of the objective function is expected to be extremely large. Local maxima occur for different polybone placements, sizes, orientations, and depth orderings. Unlike tracking problems where one may know the number of objects, one cannot enumerate and compare all possible model configurations

[5] Before this line-search, the angle parameter θ_k is first rescaled by the radius of the k^{th} polybone to provide a more uniform curvature in the objective function.

Fig. 4. The optimal known models for frames (top) 0, 2, 4, (second row) 10, 20, 40, (third) 42, 44, 46, (fourth) 48, 50, 60 and (bottom) 70, 80, 90 of the sequence.

(cf. [16]). As a consequence the method by which we search the space of polybone models is critical.

Here we use a search strategy that is roughly based on the cascade search developed in [12]. The general idea is that it is useful to keep suboptimal models which have small numbers of polybones in a list of known intermediate states. The search spaces for simpler models are expected to have fewer local maxima, and therefore be easier to search. More complex polybone models are proposed by elaborating these simpler ones. This elaboration process is iterated, generating increasingly more complex models. Revisions in the simpler models may therefore cause distant parts of the search space for more complex models to be explored. This process creates, in a sense, a 'garden web' of paths from simpler models to progressively more complex ones. The hypothesis is that optimal model(s) can often be found on this web.

In this paper, the suboptimal intermediate states that we retain in our search are the best models we have found so far having particular numbers of polybones, that is

$$\mathcal{M}(t) = (\mathcal{M}_0(t), \mathcal{M}_1(t), \ldots, \mathcal{M}_{\bar{K}}(t)) \ . \tag{13}$$

Here $\mathcal{M}_N(t)$ is a list of the best models at frame t having exactly N foreground polybones and one background polybone, and $\bar{K}$ is a constant specifying the maximum number of foreground polybones to use. The sub-list $\mathcal{M}_N(t)$ is sorted in decreasing order of the objective function $\mathcal{O}(\mathbf{M})$, and is pruned to have at most L models (in the experiments, we used $L = 1$).

The general strategy of the search is as follows. Given the partitioned list $\mathcal{M}(t-1)$ of models for frame $t-1$, and an empty list $\mathcal{M}(t)$ for a new frame t, we use *temporal proposals* to generate seed models (denoted by $\mathbf{S}_t$) for frame t. These temporal proposals use the assumed dynamics of models $\mathbf{M}_{t-1} \in \mathcal{M}(t-1)$, as suggested by the conditional distribution $p(\mathbf{M}_t \,|\, \mathbf{M}_{t-1})$. The generated seed models are then used as initial guesses for the hill-climbing procedure described in Sec. 6. The models returned by the hill-climbing procedure are inserted into $\mathcal{M}(t)$ for the new frame and, if necessary, the sub-lists $\mathcal{M}_N(t)$ are pruned to keep only the best L models with N foreground polybones.

In addition to the temporal proposals, there are *revision proposals* which help explore the space of models. These are similar to the temporal proposals discussed above, except that they operate on models $\mathbf{M}_t$ at the current time, instead of at the previous time. Otherwise, they are used the same way as for temporal proposals. That is, given a model $\mathbf{M}_t \in \mathcal{M}(t)$ they generate a seed model $\mathbf{S}_t$ which provides an initial guess for the hill-climbing. The resulting model $\tilde{M}_t$ is inserted back into the partitioned list $\mathcal{M}(t)$. Broadly speaking, useful revisions include *birth and death proposals*, which change the number of polybones, and *depth ordering proposals* which switch the depth orderings of the polybone layers.

Finally, in order to limit the number of complex polybone models considered, we find the optimal model $\mathbf{M}^*_t \in \mathcal{M}(t)$ (i.e. with the maximum value of the objective function $\mathcal{O}(\mathbf{M})$) and then prune all the models with more polybones than $\mathbf{M}_t^*$. The temporal predictions for the next frame are based on only the remaining models within $\mathcal{M}(t)$.

At the beginning of a sequence, say frame t_0, each sub-list in $\mathcal{M}(t_0)$ is initialized to be empty, and one seed model $\mathbf{S}_{t_0}$ is proposed consisting of only a background polybone covering the entire image and an initial guess for the motion parameters. Here we consider simple background motions, and the initial guess of zero motion is sufficient. A parameterized flow model is then fit using the EM-algorithm described in Section 6.[6] This provides the initial model $\mathbf{M}_{t_0}$ which is inserted into $\mathcal{M}(t_0)$. The birth and death proposals, and the revision proposals are then used to further elaborate $\mathcal{M}_0(t_0)$. The revision proposals are then executed, and the models for subsequent frames are obtained as described above.

In our current implementation we only consider two kinds of proposals, namely *temporal prediction* proposals and *birth* proposals. Given a model $\mathbf{M}_{t-1} \in \mathcal{M}_N(t-1)$, the temporal prediction proposal provides an initial guess, $\mathbf{S}_t$, for the parameters of the corresponding model in the subsequent frame. Here $\mathbf{S}_t$ is generated from $\mathbf{M}_{t-1}$ by convecting each polybone (other than the background model) in $\mathbf{M}_{t-1}$) according to the flow for that polybone. The initial guess for the flow in each polybone is obtained from a constant velocity prediction. Notice that temporal prediction proposals do not change the number of polybones in the model, nor their relative depths, but rather they use a simple dynamical model to predict where each polybone will be found in the subsequent frame.

In order to change the number of polybones or find models with different depth relations, the only revision proposal we currently consider is a birth proposal. For a given $\mathbf{M}_t \in \mathcal{M}_N(t)$, the birth proposal computes an outlier map, which provides the probability that the data at each image location $\mathbf{x}$ is considered to be an outlier according to all the visible polybones within $\mathbf{M}_t$ at that location. This map is then blurred and downsampled to reduce the influence of isolated outliers. The center location for a new polybone is selected by randomly sampling from this downsampled outlier map, with the probability of selecting any individual center location being proportional to the value of the outlier map at that location. Given this selected location, the initial size of the new polybone is taken to be fixed (we used 16×16), the initial angle is randomly selected from a uniform distribution, the initial motion is taken to be zero, and the relative depth of the new polybone is randomly selected from the range 1 to $N+1$ (i.e. it is inserted in front of the background bone, but otherwise at a random position in the depth ordering). As a result this birth proposal produces a seed model $\mathbf{S}_t$ which has exactly one more polybone.

[6] The camera was stationary in all sequences other than the Pepsi sequence (in Figure 2), so only the standard deviation of the motion constraints and the outlier mixing coefficient needed to be fit for the background in these sequences. For the Pepsi sequence a translational flow was fit in the background polybone.

Fig. 5. The optimal models found for the PETS2000 sequence, (top) frames 802, 842, and 882, (middle) 1002, 1042, 1202. The the car, three pedestrians, and bushes blowing in the wind (middle-right) are detected. (bottom) Composite images formed from all the polybones of the optimal models in every fourth frame, for frames (bottom-left) 750 to 850, (bottom-middle) 850 to 1070, and (bottom-right) 1070 to 1290. Note that the car and the pedistrians are consistently detected.

8 Examples

The results of the overall process are shown in Fig. 2, where we have limited the maximum number of foreground polybones to one in order to demonstrate the sampling and growth of a single polybone. The image sequence is formed by moving the camera horizontally to the right, so the can is moving horizontally to the left, faster than the background. Given the first two frames the background flow was fit, a birth revision proposal sampled the outliers from this flow, and generated an initial guess for a foreground polybone. Finally the hill-climbing procedure generated the polybone model shown in Fig. 2 (top-left). This polybone grows in subsequent frames to cover the region of the can. The top of the can has been slightly underestimated since the horizontal structure is consistent with both the foreground and background flow, and the penalty function introduces a bias towards smaller polybones. Conversely, the bottom of the can has been overestimated due to the consistency of the motion data on the table end with the motion of the can itself. In particular, the end of the table is moving more like the foreground polybone than the background one, and therefore the foreground polybone has been extended to account for this data as well.

A more complex example is shown in Fig. 3, where we we allow at most four foreground polybones. Notice that in the first few frames a new polybone is proposed to account for previously unexplained motion data. By the 10th frame that was processed the polybones efficiently cover the moving figure. Notice that the polybone covering the arm is correctly interpreted to be in front of the torso when it is moving differently from the torso (see Fig. 3 bottom left and right). Also, at the end of the arm swing (see Fig. 3 bottom middle) the arm is moving with approximately the same speed as the torso. Therefore the polybone covering the torso can also explain the motion of the arm in this region. The size prior causes the polybone on the arm to shrink around only the unexplained region of the hand.

A similar example of the search process is provided in Figure 4. In this sequence the subject is walking towards the camera, resulting in a relatively slow image motion. This makes the motion segmentation more difficult than in Figuree 3. To alleviate this we processed every second frame. The top row in Figure 4 shows the initial proposal and development of a model with two foreground polybones. This two component model persisted until frame 40, or so, when the subject began raising their right arm. A third foreground polybone, and then a fourth, are proposed for modeling the motion of the arm (frames 40-50). At the end of the sequence the subject is almost stationary, and the model disolves back to the background model.

Despite the limitations of the simple basic process we used, these results indicate that: 1) the search for such simple models of image flow can be tractable; and 2) the best models produced and selected by such a search process can be of practical interest. In particular, for this second point, many of the image decompositions exhibited in Figures 3 and 4 appear to be suitable starting points for the initialization of a simple human figure model [18, 3].

The results on a common test setquence are shown in Fig. 5.[7] The same configuration is used as for the previous examples except, due to the slow motion of the people (especially when they are most distant and heading roughly towards the camera), we processed every fourth frame of the sequence. Shortly after the car appears in the field of view, the system has selected four polybones to cover the car (three of which can be seen in Figure 5 (top-left), the fourth is a tiny one on the roof). But by frame 822 (five times steps later) the system has found a presumably better model using just two polybones to cover the car. These two polybones persist until the car is almost out of view, at which point just a single polybone is deemed optimal. The reason for the persistence of two polybones instead of just one is that the simple spatial form of a single polybone does not provide a sufficiently accurate model of the shape of the car, and also that the similarity motion model does not accurately capture the deformation over the whole region. An important area for future work is to provide a means to elaborate the motion and shape models in this type of situation.

[7] This sequence is available from the First IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS2000, March, 2000. We selected frames 750 to 1290 from the original sequence as the most interesting range.

Three pedestrians are also detected in this sequence (see Fig. 5 (middle and bottom)), indicating the flexibility of the representation. Composite images formed from three successive subranges of the Pets sequence are shown in Fig. 5(bottom). All of the extracted foreground polybones for the most plausible model have been displayed in one of these three images (recall that only every fourth frame was processed). These composite images show that the car is consistently extracted in the most plausbile model. The leftmost person is initially only sporatically identified (see Fig. 5 bottom-left), but is then consistently located in subsequent frames when the image motion for that person is larger. The other two people are consistently detected (see Fig. 5 bottom middle and right).

9 Conclusions

We have introduced a compositional model for image motion which explicitly represents the spatial extent and relative depths of multiple moving image regions. Each region comprises a parametric shape model, and a parametric motion model. The relative depth ordering of the regions allows visibility and occlusion relationships to be properly included in the model, and then used during the estimation of the model parameters.

This modelling framework was selected to satisfy two constraints. First, it must be sufficiently expressive to be able to provide at least a preliminary description of the dominant image structure present in typical video sequences. Secondly, a tractable means of automatically estimating the model from image data is essential. We believe that our reported results demonstrate that both of these constraints are satisfied by our polybone models together with the preliminary local search technique.

References

1. M. J. Black and A. D. Jepson. EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.
2. C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. Computer Vision and Pattern Recognition, CVPR-98*, pages 8–15, Santa Barbara, June 1998.
3. T. Cham and J.M. Rehg. A multiple hypothesis approach to figure tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern R ecognition*, volume II, pages 239–245, Fort Collins, 1998.
4. T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE Pattern Analysis and Machine Intelligence*, 17(5):474–487, May 1995.
5. J.S. de Bonet and P. Viola. Roxels: Responsibility weighted 3d volume reconstruction. In *Proceedings IEEE International Conference on Computer Vision*, volume I, pages 418–425, Corfu, September 1999.
6. J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR00*, volume II, pages 126–133, Hilton Head, 2000.

7. G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and llumination. *IEEE Trans. PAMI*, 27(10):1025–1039, 1998.
8. S.S. Intille and A.F. Bobick. Recognizing planned, multi-person action. *Computer Vision and Image Understanding*, 81(3):1077–3142, 2001.
9. M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):2–28, 1998.
10. A. Jepson and M. J. Black. Mixture models for optical flow computation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–761, New York, June 1993.
11. A.D. Jepson and R. Mann. Qualitative probabilities for image interpretation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume II, pages 1123–1130, Corfu, Greece, September 1999.
12. J. Listgarten. Exploring qualitative probabilities for image understanding. Master's thesis, Department of Computer Science, University of Toronto, October 2000.
13. J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proceedings of the IEEE International Conference on Computer Vision*, volume I, pages 572–578, Corfu, Greece, September 1999.
14. J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand t racking. In *Proceedings of the European Conference on Computer Vision*, volume II, pages 3–19, Dublin, 2000.
15. C. Rasmussen and G.D. Hager. Joint probabilistic techniques for tracking multi-part objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 16–21, Santa Barbara, June 1998.
16. C. Rasmussen and G.D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Pattern Analysis and Machine Intelligence*, 23(6):560–576, June 2001.
17. H. S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Pattern Analysis and Machine Intelligence*, 18(8):814–831, 1996.
18. H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Proc. European Conference on Computer Vision, ECCV-00*, volume II, pages 702–718. Springer-Verlag, Dublin, June 2000.
19. R. Szeliski and P. Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61, August 1999.
20. H. Tao, H.S. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 134–141, Hilton Head, June 2000.
21. P.H.S. Torr, A.R. Dick, and R. Cipolla. Layer extraction with a bayesian model of shapes. In *Proceedings of the European Conference on Computer Vision*, volume II, pages 273–289, Dublin, 2000.
22. N. Vasconcelos and A. Lippman. Empirical bayesian motion segmentation. *IEEE Pattern Analysis and Machine Intelligence*, 23(2):217–221, February 2001.
23. J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.
24. Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 520–526, Puerto Rico, June 1997.
25. Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 321–326, San Francisco, June 1996.