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] Palo Alto Research Center, 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. The representation provides a global interpretation of image motion in terms of several spatially localized foreground regions along with a background region. Each of these regions comprises a parametric shape model and a parametric motion 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 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. For intermediate-level visual analysis one particularly promising type of representation is based on the concept of layered image descriptions [4, 12, 26, 28]. Layered models provide a natural way to estimate motion when there are sevaral regions having different velocities. They have been shown to be effective for separating foreground objects from backgrounds. One weakness of existing layered representations is that they assign pixels to layers independently of pixels at neighboring locations. In doing so their underlying generative model does not manifest the constraint that most physical objects are spatially coherent and have boundaries, nor does it represent relative depths and occlusion.

In this paper we develop a new, 2.5D layered image representation. We are 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 called polybones, each of which has compact spatial support and a probabilistic representation for its borders. This representation of opaque spatial regions and soft boundaries, along with a partial depth ordering among the polybones, gives one an explicit representation of visibility and occlusion. As such, the resulting layered model corresponds to an underlying generative model that captures more of the salient properties of natural scenes than existing layered models.

Along with this 2.5D 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 (e.g., see Figure 1 (right)). Since the number of objects, their positions,

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2350, pp. 692–706, 2002.
© Springer-Verlag Berlin Heidelberg 2002

motions, shapes, sizes, and relative depths are all unknown, a complete search of the model space is infeasible. Instead we employ a stochastic search strategy 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 long image sequences are formulated as model-based tracking problems. In most cases we exploit prior knowledge about the objects of interest. For example, one often uses knowledge of the number of objects, their shapes, appearances, and dynamics, and perhaps an initial guess about object position. With 3D models one can take the effects of directional illumination into account, to anticipate shadows for instance [14]. Successful 3D people trackers typically assume detailed kinematic models of shape and motion, and initialization is still often done manually [2, 3, 7, 21]. Recent success with curve-based tracking of human shapes relies on a user defined model of the desired curve [11, 17]. For complex objects under variable illuminants, one could attempt to learn models of object appearance from a training set of images prior to tracking [1, 8]. Whether one tracks blobs to detect activities like football plays [9], or specific classes of objects such as blood cells, satellites or hockey pucks, it is common to constrain the problem with a suitable model of object appearance and dynamics, along with a relatively simple form of data association [16, 19].

To circumvent the need for such specific prior knowledge, one could rely on bottom-up, motion-based approachs to segmenting moving objects from their backgrounds, prior to tracking and identification [10, 18]. Layered image representations provide one such approach [12, 20, 25, 28]. With probabilistic mixture models and the EM (Expectation-Maximization) algorithm [6], efficient methods have been developed for determining the motion and the segmentation simultaneously. In particular, these methods give one the ability to softly assign pixels to layers, and to robustly estimate the motion parameters of each layer. One weakness in most of these methods, however, is that the assignment of pixels to layers is done independently at each pixel, without an explicit constraint on spatial coherence (although see [23, 27]). Such representations, while powerful, lack the expressiveness that would be useful in layered models, namely, the ability to explicitly represent coherence, opacity, region boundaries, and occlusion.

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 of the image motion 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 [23, 24], we assume a relatively simple parametric model for the spatial support of each layer. However, unlike the Gaussian model in [23], where the spatial support decays exponentially from the center of the object, we use a *polybone* in which support is unity over the interior of the object, and then smoothly

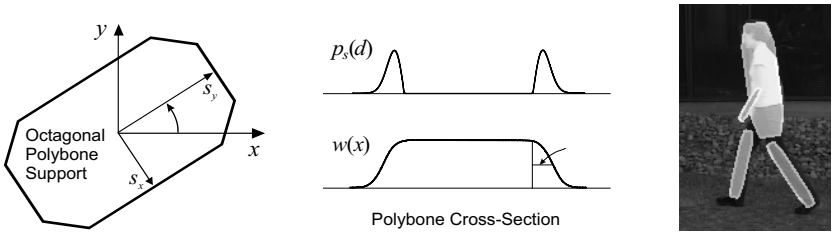

Fig. 1. (left) The spatial support of each polybone used in the experiments is a simple transform of a canonical octagonal shape. The allowed transforms include translation, rotation and independent scaling along two object-centered axes. (middle) These plots depict boundary position density $p_s(d)$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

decays to zero only in the vicinity of the spatial boundary. This representation embodies our uncertainty about exact boundary position, it 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 occlusion in a straightforward way.

One of the central issues in this research is whether or not the extraction and selection of a layered polybone description for image motion is computationally feasible. The space of possible descriptions is large owing to the unknown number of polybones, the unknown depth relations between the different polybones, and the dimension of the continuous parameter space for each polybone. We therefore require effective methods to search this space for plausible models.

3 Layered Polybones

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

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

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 of the layers 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, we define the probability density of the true boundary location, p_s, as a function of the distance, $d(\mathbf{x};\, \mathbf{b}_k)$, from a location $\mathbf{x}$ to the polygon specified by $\mathbf{b}_k$ (see Fig. 1(middle)). Given p_s, the probability that $\mathbf{x}$ lies inside of the true boundary is then expressed as $w(\mathbf{x};\mathbf{b}_k) = p_s(d > d(\mathbf{x};\mathbf{b}_k))$, the cummulative probability that the distance d, in the direction $\mathbf{x}$, is greater than $d(\mathbf{x};\mathbf{b}_k)$. This occupancy probability, $w(\mathbf{x};\mathbf{b}_k)$,

serves as our definition of spatial support from which we can formulate *visbility* and *occlusion*. As depicted in Fig. 1(middle), 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. For convenience, the standard deviation of the half-Gaussian, $\sigma_{s,k}$, is taken to be a constant. In practice we truncate the polybone shoulders to zero after a distance of $2.5\sigma_s$.

With these definitions, the visibility of the j^{th} polybone at a pixel $\mathbf{x}$ depends on the probabilities that closer layers do not occupy $\mathbf{x}$; i.e., the visibility probability is

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

where all pixels in the foremost layer are defined to be visible, so $v_K(\mathbf{x}) = 1$. It may be interesting to note that transparency could also be modeled 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. [5, 22]). 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 and the examples below we restrict the polybone shape to be a simple transformation of a canonical octagonal boundary (see Fig. 1(left)), and we let $\sigma_s = 4$ pixels. 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(left)). Together with the boundary uncertainty parameter, σ_s, these parameters define the shape and pose of a polybone:

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

This simple description for shape and pose was selected, in part, to simplify the exposition in this paper and to facilitate the parameter estimation. It would be straightforward to include more complex polygonal or spline based shape descriptions in the representation (although local extrema in the optimization may be more of a problem).

Finally, in addition to shape and pose, the polybone parameters also specify the motion within the layer. In particular, the motion parameters associated with the k^{th} polybone, denoted by $\mathbf{m}_k$, specify a parametric image warp, $\mathbf{w}(\mathbf{x}; \mathbf{m}_k(t))$, from pixels at time $t + 1$ to pixels at time t. In the current implementation we use similarity deformations, where $\mathbf{m}_k$ specifies translation, rotation and uniform scaling between frames.

4 Model Likelihood

The likelihood of a layered polybone model $\mathbf{M}_t$ depends on how well it accounts for the motion between frames t and $t + 1$. As is common in optical flow estimation, our motion likelihood function follows 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 points $(\mathbf{x}, t)$ map to points in the next frame given by $(\mathbf{x}', t + 1) = (\mathbf{w}(\mathbf{x}; \mathbf{m}_k(t)), t + 1)$. The similarity of the image data at these two points is typically measured in terms of a probability distribution for the difference

$$\delta d_k(\mathbf{x}, t) \;=\; d(\mathbf{w}(\mathbf{x}; \mathbf{m}_k(t)), t + 1) \;-\; d(\mathbf{x}, t) \;. \tag{4}$$

The distribution for the deviation δd is often taken to be a Gaussian density, say $p_1(\delta d)$, having mean 0 and standard deviation σ_m. To accommodate data outliers, a linear mixture of a Gaussian density and a broad outlier distribution, $p_0(\delta d)$, can be used. Such mixture models have been found to improve the robustness of motion estimation in the face of outliers and unmodelled surfaces [12, 13]. Using a mixture model, we then define the likelihood (i.e., the observation density) of a single data observation, $\delta d_k(\mathbf{x}, t)$, given the warp, $\mathbf{w}(\mathbf{x}; \mathbf{m}_k(t))$, to be

$$p_k(\delta d_k(\mathbf{x}, t)) \;=\; (1 - \pi_{0,k})\, p_1(\delta d_k(\mathbf{x}, t)) \;+\; \pi_{0,k}\, p_0(\delta d_k(\mathbf{x}, t)) \;, \tag{5}$$

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

The likelihood for the k^{th} polybone at a pixel $\mathbf{x}$ can be combined with the likelihoods for 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 layered polybone model at a single location $\mathbf{x}$ and frame t is given by

$$p(\{\delta d_k(\mathbf{x}, t)\}_{k=0}^{K} \mid \mathbf{M}_t) \;=\; \sum_{k=0}^{K} v_k(\mathbf{x})\; w(\mathbf{x}; \mathbf{b}_k)\; p_k(\delta d_k(\mathbf{x}, t)) \;. \tag{6}$$

Finally, given independent noise at different pixels, the log likelihood of the layered polybone model $\mathbf{M}_t$ over the entire image is

$$\log p(\mathbf{D}_t \mid \mathbf{M}_t) \;=\; \sum_{\mathbf{x}} \log p(\{\delta d_k(\mathbf{x}, t)\}_{k=0}^{K} \mid \mathbf{M}_t) \;. \tag{7}$$

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 $\delta d_k(\mathbf{x}, t)$; moreover, the model itself is required to determine corresponding points.

5 Penalized Likelihood

We now derive the objective function that is used to optimize the polybone parameters and to compare alternative models. The objective function is motivated by the standard Bayesian filtering equations 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, ignoring constant terms, the log posterior is given by

$$\mathcal{U}(\mathbf{M}_t) \;=\; \log p(\mathbf{D}_t \mid \mathbf{M}_t) + \log p(\mathbf{M}_t \mid \mathcal{D}_{t-1}) \;. \tag{8}$$

The last term above is the log of the conditional distribution over models $\mathbf{M}_t$ given all the previous data, which is typically expressed as

$$\mathbf{p}(\mathbf{M}_t \mid \mathcal{D}_{t-1}) \;=\; \int_{\tilde{\mathbf{M}}_{t-1}} p(\mathbf{M}_t \mid \tilde{\mathbf{M}}_{t-1}) p(\tilde{\mathbf{M}}_{t-1} \mid \mathcal{D}_{t-1}) \;, \tag{9}$$

given suitable independence and Markov assumptions. Given the complexity of the space of models we are considering, a detailed approximation of this integral is beyond the scope of this paper. Instead, we use the general form of (8) and (9) to motivate a simpler penalized likelihood formulation for the objective function, namely

$$\mathcal{O}(\mathbf{M}_t) \;=\; \log p(\mathbf{D}_t \,|\, \mathbf{M}_t) + q(\mathbf{M}_t, \mathbf{M}_{t-1}) \;. \tag{10}$$

The last term in (10), called the penalty term, is meant to provide a rough approximation for the log of the conditional probability distribution in (9).

The penalty term serves two purposes. First, when 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 provides a bias towards particular parameter values. In our current implementation we include two terms in $q(\mathbf{M}_t, \mathbf{M}_{t-1})$ that bias the models to smaller polybones and to smooth shape changes:

$$q_1(\mathbf{M}_t) = \sum_{k=1}^{K} \log[L_1(s_{x,k,t} - 1) L_1(s_{y,k,t} - 1)] \tag{11}$$

$$q_2(\mathbf{M}_t, \mathbf{M}_{t-1}) = \sum_{k} \log N(\mathbf{a}_{k,t} - \tilde{\mathbf{a}}_{k',t}; \Sigma_a) \tag{12}$$

Here q_1 provides the bias towards small polybones, with $L_1(s)$ equal to the one-sided Laplace density $\lambda_s e^{-\lambda_s s}$. The second term, q_2, provides a bias for smooth shape changes with a mean zero normal density evaluated at the temporal difference in shape parameters. Here, k' is the index of the polybone in $\mathbf{M}_{t-1}$ that corresponds to the k^{th} polybone in $\mathbf{M}_t$; if such a k' exists, then $\tilde{\mathbf{a}}_{k',t}$ denotes the pose of this polybone at time $t-1$ warped by the motion defined by $\mathbf{m}_{k',t-1}$. The sum in (12) is over all polybones in $\mathbf{M}_t$ that have corresponding polybones in $\mathbf{M}_{t-1}$.

The second purpose of the penalty function is to control model complexity. Without a penalty term the maximum of the log likelihood in (10) will be monotonically increasing in the number of polybones. However, beyond a certain point, the extra polybones primarily fit noise in the data set, and the corresponding increase in the log likelihood is marginal. The penalty term in (10) is used to ensure that the increase in the log likelihood obtained with a new polybone is sufficiently large to justify the new polybone. To derive this third term of $q(\mathbf{M}_t, \mathbf{M}_{t-1})$ we assume that each polybone parameter can be resolved to some accuracy, and that the likelihood does not vary significantly when parameters are varied within such resolution limits. As with conventional Bayesian model selection, the penalty function is given by the log volume of the resolvable set of models. In our current implementation, the third term in the penalty function is given by

$$q_3(\mathbf{M}_t) = \sum_{k=1}^{K} \log \left(\left[\frac{4\sigma_s^2}{n_x n_y} \right]^2 \; \frac{4\sigma_s}{\pi r} \; \frac{1}{10} \; \frac{2\sigma_{v,k}}{10} \; \frac{\log(2)}{\log(20)} \right) \;, \tag{13}$$

where the different factors in (13) correspond to the following resolutions: We assume the location and size parameters of any given polybone are resolved to $\pm\sigma_s$ over the image of size $n_x \times n_y$; the angle θ is resolved to $\frac{4\sigma_s}{r}$ where r is the radius of the polybone; the inlier mixing proportion used in the motion model is resolved to ± 0.05 out of the

range $[0,1]$; the inlier motion model has flow estimates that are resolved to within $\pm\sigma_{v,k}$ over a possible range of $[-5,5]$; and $\sigma_{v,k}$ is estimated from the inlier motion constraints to within a factor of 2 (i.e. $\pm\sqrt{2}\sigma_{v,k}$), with a uniform prior for $\sigma_{v,k}$ having minimum and maximum values of 0.1 and 2.0 pixels/frame.

6 Parameter Estimation

Suppose $\mathbf{M}_t^0$ is an initial guess for the parameters (1) of the layered model. In order to find local extrema of (10) we use a form of gradient ascent. The gradient of the penalty term is easy to compute, while that of the log likelihood is simpler to compute if we exploit the layered structure of the model. We do this by rewriting $p(D(\mathbf{x}) \mid \mathbf{M})$ in a form that isolates the parameters of each individual polybone.

To simplify the likelihood expression, first note from (2) and (6) that the contribution to $p(D(\mathbf{x}) \mid \mathbf{M})$ from only those polybones that are closer to the camera than the k^{th} bone can be expressed as

$$\begin{aligned} n_k(\mathbf{x}) &= \sum_{j=k+1}^{K} v_j(\mathbf{x})\, w(\mathbf{x};\mathbf{b}_j)\, p(D(\mathbf{x})|\mathbf{b}_j) \\ &= v_{k+1}(\mathbf{x})\, w(\mathbf{x};\mathbf{b}_{k+1})\, p(D(\mathbf{x}) \mid \mathbf{b}_{k+1}) \;+\; n_{k+1}(\mathbf{x}) \;. \end{aligned} \tag{14}$$

We refer to $n_k(\mathbf{x})$ as the *near term* for the k^{th} polybone. Equations (14) 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 into the 'far term',

$$\begin{aligned} f_k(\mathbf{x}) &= \sum_{j=0}^{k-1} w(x;\mathbf{b}_j) \left[\prod_{l=j+1}^{k-1} (1 - w(\mathbf{x};\mathbf{b}_l)) \right] p(D(\mathbf{x})|\mathbf{b}_j) \\ &= w(\mathbf{x};\mathbf{b}_{k-1})\, p(D(\mathbf{x}) \mid \mathbf{b}_{k-1}) + (1 - w(\mathbf{x};\mathbf{b}_{k-1}))\, f_{k-1}(\mathbf{x}) \;. \end{aligned} \tag{15}$$

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 (15) 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

$$\begin{aligned} p(D(\mathbf{x}) \mid \mathbf{M}) &= n_k(\mathbf{x}) \;+\; v_k(\mathbf{x})\, w(\mathbf{x};\mathbf{b}_k)\, p(D(\mathbf{x}) \mid \mathbf{b}_k) \\ &\quad + v_k(\mathbf{x})(1 - w(\mathbf{x};\mathbf{b}_k))\, f_k(\mathbf{x}) \;. \end{aligned} \tag{16}$$

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 $p(D(\mathbf{x}) \mid \mathbf{b}_k)$ in (16). This greatly simplifies the derivation and the computation of the gradient of the likelihood with respect to $\mathbf{b}_k$.

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 in (2) and (14). In doing so we 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 parameters. 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 in (15) 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 (motion) parameters of the k^{th} polybone. This has the same structure as the EM-algorithm in fitting motion mixture models [12], except that here the near and far terms contribute to the data ownership computation. The M-step of this EM-algorithm yields a linear solution for the motion parameter update. This 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.[4] 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 stuck at a local maximum. To avoid this behaviour we initially constrain the scaling parameters s_x and s_y to be equal, and just update the mean position (c_x, c_y), angle θ, and this uniform scale. Once we have detected a local maximum in these reduced parameters, we allow the individual scales s_x and s_y to evolve to different values. This behaviour is evident in the foreground polybone depicted in Fig. 2.

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 (cf. [19]). 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 [15]. 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 then proposed by elaborating these simpler ones.

[4] 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.

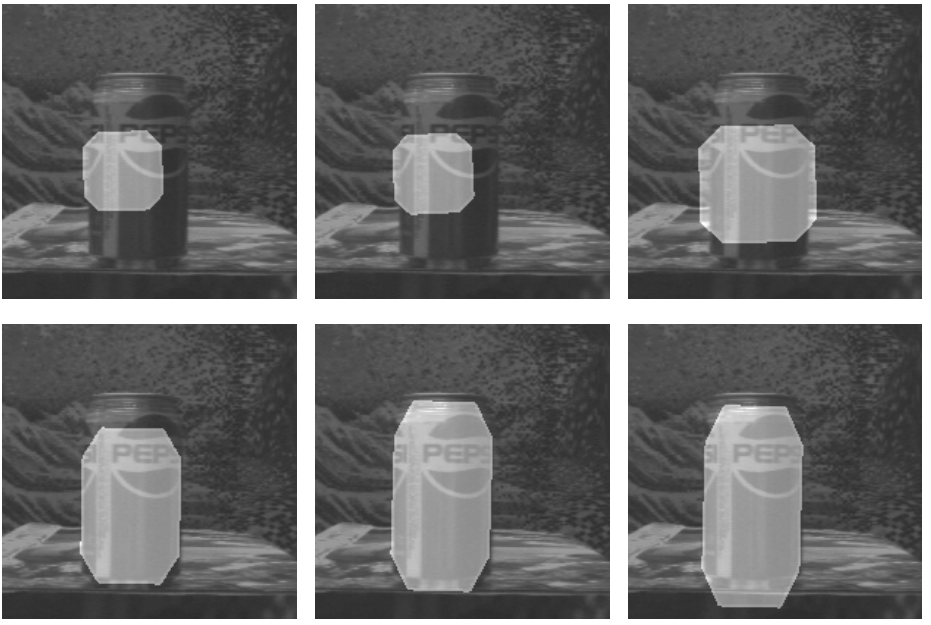

Fig. 2. Growth of a single foreground polybone in the first 6 frames (shown in lexicographic order) of a short sequence. The background polybone that is occluded by the foreground layer covers the entire image but is not shown. For the first three frames the two scale parameters, s_x and s_y, are constrained to be equal, afterwhich they are allow to vary independently (see text).

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. Our 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. We denote the collection of layered polybone models at frame t by

$$\mathcal{M}(t) = (\mathcal{M}_0(t), \mathcal{M}_1(t), \ldots, \mathcal{M}_{\bar{K}}(t)) \ , \tag{17}$$

where $\mathcal{M}_N(t)$ is a list of the best models found 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 $L = 1$).

To describe the general form of the search strategy, assume that we begin with a 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 then use *temporal proposals* to generate seed models (denoted by $\mathbf{S}_t$) for frame t. These temporal proposals arise from the assumed model dynamics suggested by $p(\mathbf{M}_t \mid \mathbf{M}_{t-1})$, for each model $\mathbf{M}_{t-1} \in \mathcal{M}(t-1)$. These seed models are used as initial guesses for the hill-climbing procedure described in Sec. 6. The models found by

the hill-climbing are then inserted into $\mathcal{M}(t)$, 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 temporal proposals, except that they operate on models $\mathbf{M}_t$ at the current time rather than at the previous time. That is, given a model $\mathbf{M}_t \in \mathcal{M}(t)$, a revision proposal generates a seed model $\mathbf{S}_t$ that provides an initial guess for hill-climbing. The resulting model $\tilde{M}_t$ is then 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 among 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 proposals for the next frame are obtained from only those models that remain in $\mathcal{M}(t)$.

Initially, given the first frame at time t_0, each sub-list in $\mathcal{M}(t_0)$ is taken to be empty. Then, given the second frame, one seed model S_{t_0} is proposed that consists of a background polybone with an initial guess for its motion parameters. The background polybone is always taken to cover the entire image. 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 Sec. 6.[5] This produces the initial model $\mathbf{M}_{t_0}$ that is inserted into $\mathcal{M}(t_0)$. Revision proposals are then used to further elaborate $\mathcal{M}_0(t_0)$, afterwhich the models for subsequent frames are obtained as described above.

Our current implementation uses two kinds of proposals, namely *temporal proposals* and *birth proposals*. Given a model $\mathbf{M}_{t-1} \in \mathcal{M}_N(t-1)$, the temporal proposal provides an initial guess, $\mathbf{S}_t$, for the parameters of the corresponding model in the next frame. Here $\mathbf{S}_t$ is generated from $\mathbf{M}_{t-1}$ by warping each polybone (other than the background model) in $\mathbf{M}_{t-1}$ according to the motion parameters for that polybone. The initial guess for the motion in each polybone is obtained from a constant velocity prediction. Notice that temporal proposals do not change the number of polybones in the model, nor their relative depths. 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, we currently rely solely on birth proposals. For a given $\mathbf{M}_t \in \mathcal{M}_N(t)$, the birth proposal computes a sparsely sampled outlier map that represents the probability that the data at each location $\mathbf{x}$ is owned by the outlier process, given 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 the new polybone is selected by randomly sampling from this downsampled outlier map. 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

[5] The camera was stationary in all sequences except that in Fig. 2, so only the standard deviation of the motion constraints and the outlier mixing coefficient needed to be fit for the background in these cases. For the Pepsi sequence a translational flow was fit in the background polybone.

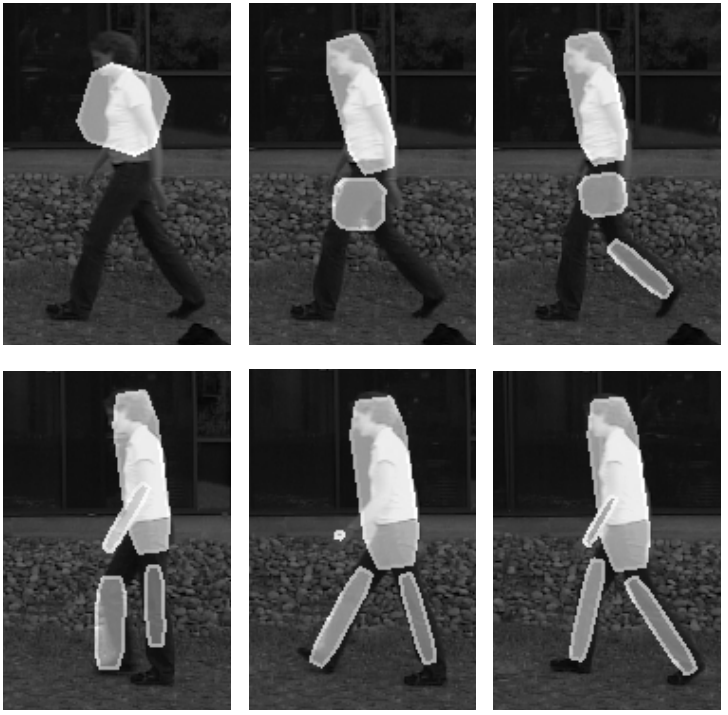

Fig. 3. The development of the optimal known model. The top row shows results for the first three frames. The bottom row shows results for frames 11, 15, and 19.

position in the depth ordering). Thus, the birth proposal produces a seed model $\mathbf{S}_t$ that has exactly one more polybone.

8 Examples

The results of the entire process are shown in Fig. 2. Here we 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 horizontal camera motion, so that the can is moving horizontally to the left faster than the background. Given the first two frames, the background motion was fit. An initial guess for a foreground polybone was generated by the birth process which, in this case, was sampled from the background motion outliers. The hill-climbing procedure then generated the polybone model shown in Fig. 2 (top-left). This polybone grows in subsequent frames to cover the can. The top of the can has been slightly underestimated since the horizontal structure is consistent with both the foreground and background motions, and the penalty function introduces a bias towards smaller polybones. Conversely, the bottom of the can was overestimated because the motion of the can and the table are consistent in this region. 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 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 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 depicted in Fig. 4. In this case the subject walks towards the camera, producing slow image velocities. This makes motion segmentation more difficult than in Fig. 3. To alleviate this we processed every second frame. The top row in Fig. 4 shows the initial proposal generated by the algorithm, and development of a model with two foreground polybones. The two component model persisted until about frame 40 when the subject began to raise their right arm. A third foreground polybone, and then a fourth, are proposed to model the arm motion (frames 40-50). At the end of the sequence the subject is almost stationary and the model dissolves into the background model. This disappearance of polybones demonstrates the preferance for simpler models, as quantified by $q_3(\mathbf{M}_t)$ in (13).

The results on a common test sequence are shown in Fig. 5.[6] 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 can be seen in Fig. 5 (top-left) and the fourth covers a tiny region 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 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.

Fig. 5 (middle and bottom) shows that three pedestrians are also detected in this sequence, indicating the flexibility of the representation. Composite images formed from three successive PETS subsequences 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 sporadically 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).

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

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.

9 Conclusions

We have introduced a compositional model for image motion that 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

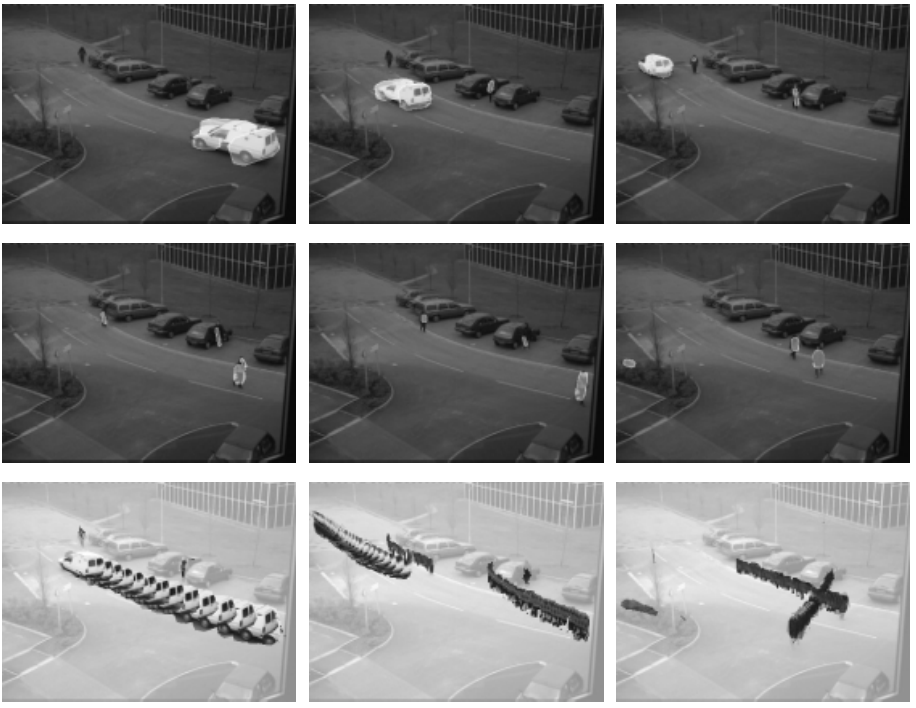

Fig. 5. The optimal models found for the PETS2000 sequence, (top) frames 802, 842, and 882, (middle) 1002, 1042, 1202. The the car, three pedestrians, and bushes blowing in the wind (middle-right) are detected. (bottom) Composite images formed from all the polybones of the optimal models in every fourth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

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 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. *IJCV*, 26:63–84, 1998.
2. C. Bregler and J. Malik. Tracking people with twists and exponential maps. *Proc. IEEE CVPR*, pp. 8–15, Santa Barbara, 1998.
3. T. Cham and J.M. Rehg. A multiple hypothesis approach to figure tracking. *Proc. IEEE CVPR*, vol. II, pp. 239–245, Fort Collins, 1998.
4. T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE PAMI*, 17:474–487, 1995.

5. J.S. de Bonet and P. Viola. Roxels: Responsibility weighted 3d volume reconstruction. *Proc. IEEE ICCV*, vol. I, pp. 418–425, Corfu, 1999.
6. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. B*, 39:1–38, 1977.
7. J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. *Proc. IEEE CVPR*, vol. II, pp. 126–133, Hilton Head, 2000.
8. G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and llumination. *IEEE PAMI*, 27:1025–1039, 1998.
9. S.S. Intille and A.F. Bobick. Recognizing planned, multi-person action. *CVIU*, 81:1077–3142, 2001.
10. M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *IJCV*, 12:5–16, 1994.
11. M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29:2–28, 1998.
12. A. Jepson and M. J. Black. Mixture models for optical flow computation. *Proc. IEEE CVPR*, pp. 760–761, New York, 1993.
13. A.D. Jepson, D.J. Fleet and T.F. El-Maraghi. Robust on-line appearance models for visual tracking. *Proc. IEEE CVPR*, Vol. 1, pp. 415–422, Kauai, 2001.
14. D. Koller, K. Daniilidis, T. Thorhallson, and H.-H. Nagel. Model-based object tracking in traffic scenes. *Proc. ECCV*, pp. 437–452. Springer-Verlag, Santa Marguerita, 1992.
15. J. Listgarten. Exploring qualitative probabilities for image understanding. MSc. Thesis, Dept. Computer Science, Univ. Toronto, October 2000.
16. J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *Proc IEEE ICCV*, vol. I, pp. 572–578, Corfu, 1999.
17. J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. *Proc. ECCV*, vol. II, pp. 3–19, Dublin, 2000.
18. F.G. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *CVGIP: Image Understanding*, 60:119–140, 1994.
19. C. Rasmussen and G.D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE PAMI*, 23:560–576, 2001.
20. H. S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE PAMI*, 18:814–831, 1996.
21. H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. *Proc. ECCV*, vol. II, pp. 702–718. Springer-Verlag, Dublin 2000.
22. R. Szeliski and P. Golland. Stereo matching with transparency and matting. *IJCV*, 32:45–61, 1999.
23. H. Tao, H.S. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. *Proc. IEEE CVPR*, vol. 2, pp. 134–141, Hilton Head, 2000.
24. P.H.S. Torr, A.R. Dick, and R. Cipolla. Layer extraction with a Bayesian model of shapes. *Proc. ECCV*, vol. II, pp. 273–289, Dublin, 2000.
25. N. Vasconcelos and A. Lippman. Empirical Bayesian motion segmentation. *IEEE PAMI*, 23:217–221, 2001.
26. J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Trans. Im. Proc.*, 3:625–638, 1994.
27. Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. *Proc. IEEE CVPR*, pp. 520–526, Puerto Rico, 1997.
28. Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. *Proc. IEEE CVPR*, pp. 321–326, San Francisco, 1996.