Multiview Segmentation and Tracking of Dynamic Occluding Layers

Ian Reid Keith Connor

Dept of Engineering Science
University of Oxford
Oxford, UK

Abstract

We present an algorithm for the layered segmentation of video data in multiple views. The approach is based on computing the parameters of a layered representation of the scene in which each layer is modelled by its motion, appearance and occupancy, where occupancy describes, probabilistically, the layer's spatial extent and not simply its segmentation in a particular view. The problem is formulated as the MAP estimation of all layer parameters conditioned on those at the previous time step; i.e. a sqeuential estimation problem that is equivalent to tracking multiple objects in a given number views. Expectation-Maximisation is used to establish layer occupancy and visibility (which are represented distinctly) posterior probabilities. Evidence from areas in each view which are described poorly under the model is used to propose new layers automatically. Since these potential new layers often occur at the fringes of images, the algorithm is able to segment and track these in a single view until such time as a suitable candidate match is discovered in the other views. The algorithm is shown to be very effective at segmenting and tracking non-rigid objects and can cope with extreme occlusion.

1 Introduction

The layered representation has become a popular means of representing and describing natural scenes in a compact way. The idea is that a video sequence may be represented by a small number of textured regions and their associated motions [12].

Layers have mainly found use in the representation of monocular video sequences, typically for applications concerned with video coding. In previous work [4] we described a layered representation suitable for multiple view descriptions of dynamic scenes in which occlusions occur. Our aim was to extract all relevant parameters from the layered model including segmentation, appearance, motion and correspondence information. The resulting representation has applications in, for example, video coding, but we were (and remain) motivated by the problem of novel view synthesis for dynamic scenes in which knowledge of occlusion boundaries can dramatically improve the speed and quality of novel rendered views.

In the current paper we reformulate the mathematical expression of the problem to deal not only with the binocular case, but also with the monocular case, obtaining in the process an algorithm that is potentially n-view (though our results to date only show a maximum of two views). We also make the important extension to our previous work that new layers are automatically proposed when the current generative model fails adequately to explain the current images.

Our algorithm is based fundamentally on the observation that in order to deal with occlusion, it is necessary to represent occupancy – i.e the spatial extent of each layer. Further, in order to estimate occupancy, visibility must be considered – i.e the visible subset of occupancy in a

particular view. The representation of both visibility and occupancy and the consideration of multiple views are the key features of our work, and distinguish it from the plethora of work that has gone before, much of which models only visibility, and most of which considers only a single viewpoint.

1.1 Related Work

The most common forms of layered model encountered in the literature are designed for the single view case. Early approaches were mostly bottom-up. Wang and Adelson [12], robustly compute affine motion parameters over an arbitrary grid of patches and proceed to cluster motion and re-evaluate both the number and extent of the layers. Then approaches by [5] [1], employ a probabilistic mixture model formulation to compute the maximum likelihood layer parameters by simultaneously computing segmentation and motion.

A particular variant among previous approaches is whether or not occlusion is fully accounted for. The persistent representation of a layer's occupancy in spite of occlusion is key for tracking and is exploited by Jepson *et al.* [7], where a strong shape model is employed. Tao *et al.* [10] model a layer's shape by a Gaussian spatial prior but this serves more as a segmentation (i.e. visibility) prior rather than an occupancy prior and thus does not explicitly consider occlusion.

Like us, Frey and Jojic [8] model occlusion through a layered generative model. Their method is designed to determine layers in a set of images in which there is no assumed temporal ordering. The placement of a layer in an image is modelled as a distribution over all possible locations, quantised to the resolution of the image grid. Although their approach is quite general, there are two reasons we do not pursue a similar approach here: (i) in many applications there are strong temporal constraints available from ordered image sequences. The use of these constraints produces a more efficient algorithm. (ii) Frey and Jojic demonstrated only translational changes in layers. Although their framework is not restricted to translation, there is a practical difficulty in computing the distribution over, say, all six affine degrees of freedom. In contrast by making a (fairly weak) assumption of temporal continuity, we can afford to represent alignments and their associated uncertainties analytically.

Zhou and Tao [13] describe an approach to modelling the background which may occlude foreground layers. This work is similar to ours in formulation but does not consider multiple views and in some respects may be regarded as a special case. In particular, their solution is via a method of axial iteration in which some parameters are held fixed while others are optimised. The solution method is therefore inefficient and will not reach a local optimum in the single pass used. Here however, we derive the exact EM algorithm for the generative model and obtain a much more efficient solution without needing to discretise the space.

Most previous approaches compute motion layers for a single view of a dynamic scene, while other less prolific work considers structural layers. The work of [2] and [11] consider two-views of a scene in order to extract 3D layers, where the transformations between views is due to structure rather that dynamic object motion. In contrast, our work considers both motion and structure.

2 Layered Model

In this section we describe the layered representation and consider a generative model; it is then shown how this suggests a solution via the EM-algorithm.

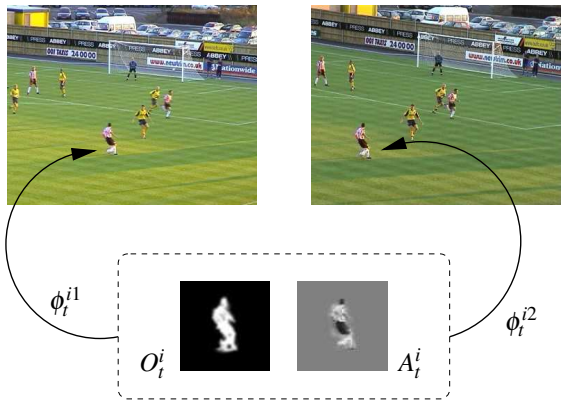

Figure 1: The parameters that describe a layer are occupancy O_t^i (represented by a probabilistic map), appearance A_t^i (represented by an intensity map), and alignment ϕ_t^{ij} (a transformation relating the coordinate frame of the ith layer to the jth image).

2.1 Parameters

Assume the layered model consists of $n+1$ depth ordered layers: the background layer and n foreground layers. Note that the ordering of layers is determined indirectly (via disparity) by their inter-viewpoint spatial aligment parameters. Each layer can be defined by its occupancy, appearance and alignment parameters. The first two properties correspond to the underlying object's shape and colour (the intrinsic parameters), whereas the alignment parameters relate the coordinate frame of the layer to each view (the extrinsic parameters); figure 1 illustrates the meaning of the layer parameters. The layered model at time t is denoted as $L_t = (L_t^0, L_t^1, ..., L_t^n)$, where

$$L_t^i = (O_t^i, A_t^i, \Phi_t^i) \tag{1}$$

are the parameters (occupancy, appearance, alignments) of the ith layer. Each layer has m alignments (one for each view)

$$\Phi_t^i = \{\phi_t^{ij}\} \qquad j \in [1, ..., m] \tag{2}$$

2.2 Model

Conceptually, an image is composed of a number of independent layers which, in general, may overlap and therefore occlude each other. The result is that the value of an image pixel is generated by the foremost layer at that point. The composition of layers involves two variables: which layer is the foremost and occupies a particular point (visibility), and what value does that layer generate at that point (appearance).

More formally, the generative model for an observed image in the jth view I_t^j is such that the intensity at pixel x is generated according to the realisation of a random variable described by the appearance model of the foremost layer at the point x. If we assume the existence of an indicator variable that states which layer is foremost (a visibility indicator), and further, we consider it to be a random variable we obtain a mixture model formulation. This is described by

$$P(I_t^j(x)) = \sum_{i=0}^{n} P(I_t^j(x)|V_t^j(x) = i)P(V_t^j(x) = i) \tag{3}$$

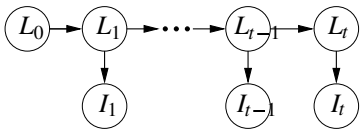

Figure 2: A Bayesian network illustrates the problem of tracking the layered representation L_t given the observations (current images) I_t and takes the form of a hidden Markov model.

in which the probability of the pixel value $I_t^j(x)$ given that the ith layer is visible constitutes the ith layer's appearance model. Here, the observed intensity is assumed to be distributed normally conditioned on the visibility and has mean given by the aligned appearance map:

$$P(I_t^j(x)|V_t^j(x) = i) \sim N(A_t^i(\phi_t^{ij^{-1}}x), \sigma_I^2) \tag{4}$$

Interpret the visibility $P(V_t^j(x) = i)$ as the probability that the ith layer is visible in the jth view at x. Then the visibility probability of the ith layer can be expressed in terms of the occupancy parameters of all layers

$$P(V_t^j(x) = i) = O_t^i(\phi_t^{ij^{-1}}x) \prod_{k=i+1}^{n} [1 - O_t^j(\phi_t^{kj^{-1}}x)] \tag{5}$$

that is, the probability that a particular layer is visible at x is given by the probability that it occupies x *and* that no closer layer occupies x.

The solution to mixture model problems typically involves attempting to *invert* the generative model given the generated data. If we know which layer is visible at each pixel, then our problem is partitioned into $n + 1$ simpler sub-problems which can be solved using ML or MAP parameter estimation for example. The problem is that we do not know the visibilities; they are hidden.

The EM-algorithm is a method which solves the hidden data problem by assuming an initial estimate for the parameters. We can produce initial estimates for the parameters from those obtained at the previous time step.

3 Estimating the Layer Parameters

The layer model is illustrated by the network shown in figure 2, where L_t represents the set of all layer parameters at time t and I_t represents the set of all images at time t. The joint probability of all nodes in figure 2 can be factored as

$$P(L_0) \prod_{\tau=1}^{t} P(I_\tau|L_\tau)P(L_\tau|L_{\tau-1}) \tag{6}$$

which, as a function of the layer parameters at time t is proportional to $P(I_t|L_t)P(L_t|L_{t-1})$. Therefore, the problem can be expressed as: find the parameters which maximise the function $F(L_t)$,

$$F(L_t) = \ln P(I_t|L_t) + \ln P(L_t|L_{t-1}) \tag{7}$$

3.1 EM algorithm

The Expectation-Maximisation algorithm [6] is applied in this section to solve a hidden data problem. Starting from the original cost function $F(L_t)$, we introduce the hidden visibility

variables V and a distribution $Q(V)$ over these variables to give

$$\begin{aligned} F(L_t) &= \ln P(I_t|L_t) + \ln P(L_t|L_{t-1}) && (8) \\ &= \ln \sum_V P(V, I_t|L_t) + \ln P(L_t|L_{t-1}) && (9) \end{aligned}$$

A lower bound is constructed using Jensen's inequality and it is the optimisation of this which is the EM algorithm.

$$F(Q, L_t) = \sum_V Q(V)(\ln P(V, I_t|L_t) - \ln Q(V)) + \ln P(L_t|L_{t-1}) \quad (10)$$

It can be shown that equality between the original and lower bound holds when $Q(V) = P(V|I_t, L_t)$, i.e when the $Q(V)$ is the posterior visibility distribution. Therefore, by assuming initial estimates for the parameters we can compute $Q(V)$ (the E-step). Next, we can maximise the lower bound $F(Q, L_t)$ given $Q(V)$ (the M-step). In summary, using k to represent the iteration number, we iterate the following steps until convergence:

E-step:

$$Q^{(k)}(V) = P(V|I_t, L_t^{(k-1)}) \quad (11)$$

M-step:

$$L_t^{(k)} = \arg\max_{L_t} \sum_V Q^{(k)}(V) \ln P(V, I_t|L_t) + \ln P(L_t|L_{t-1}) \quad (12)$$

Despite appearances, solving equations 11 and 12 is much easier than solving equation 7 because, as we will show, the parameters of each layer can be solved for independently.

In the following the dependence on the current layer parameters L_t is implicit. We assume that, conditioned on the hidden visibility variables and layer parameters, pixel values are independent. The E-step then involves computing the posterior visibility distribution over the layer index i for each pixel x of each view j denoted by $q^{ij}(x)$ and given by

$$\begin{aligned} q^{ij}(x) &= P(V_t^j(x) = i|I_t^j(x)) && (13) \\ &\propto P(I_t^j(x)|V_t^j(x) = i)P(V_t^j(x) = i) && (14) \end{aligned}$$

where the prior visibility is given by equation 5.

The M-step involves maximising the function $F(q, L_t)$

$$\begin{aligned} F(q, L_t) &= \sum_{i=0}^{n} \sum_{j=1}^{m} \sum_x q^{ij}(x) \ln P(I_t^j(x)|V_t^j(x) = i) \\ &+ q^{ij}(x) \ln P(V_t^j(x)) + \ln P(L_t^i|L_{t-1}^i) \quad (15) \end{aligned}$$

The final form of the cost function becomes the following, where here, the variable x is a position relative to the coordinate frame of the ith layer

$$\begin{aligned} &\sum_{i=0}^{n} \sum_{j=1}^{m} \sum_x q^{ij}(\phi_t^{ij} x) \ln P(I_t^j(\phi_t^{ij} x)|V_t^j(\phi_t^{ij} x) = i) \\ &+ q^{ij}(\phi_t^{ij} x) \ln O_t^i(x) + \left(\sum_{k=0}^{i-1} q^{kj}(\phi_t^{ij} x) \right) \ln(1 - O_t^i(x)) \\ &+ \ln P(\Phi_t^i|\Phi_{t-1}^i) + \ln P(O_t^i|O_{t-1}^i) + \ln P(A_t^i|A_{t-1}^i) \quad (16) \end{aligned}$$

It can be seen that the M-step may be performed by independently optimising each layer's parameters. Further, within each layer occupancy and appearance may be optimised independently of each other. However, the alignment parameters cannot be optimised independently

of the occupancy and appearance parameters. It is therefore necessary to perform an E-step between solving for the alignments and solving for the other parameters. This approach is a version of Generalised EM and is also guaranteed to converge.

3.2 Computing Alignment

In order to compute the alignment parameters we consider the cost function when all other parameters are fixed. Consider, the *i*th layer's alignment with the *j*th view, the expression to maximise is

$$\begin{aligned} F(q,\phi_t^{ij}) &= \sum_x -q^{ij}(\phi_t^{ij}x)\frac{(A_t^i(x)-I_t^j(\phi_t^{ij}x))^2}{2\sigma_I^2} + q^{ij}(\phi_t^{ij}x)\ln O_t^i(x) \\ &+ \left(\sum_{k=0}^{i-1} q^{kj}(\phi_t^{ij}x)\right)\ln(1-O_t^i(x)) + \ln P(\phi_t^{ij}|\phi_{t-1}^{ij}) \end{aligned} \tag{17}$$

In words, the optimum alignment for the *i*th layer with the *j*th image is found when (1) the appearance map agrees with the image data wherever the *i*th layer is visible (first term), (2) the occupancy map is large wherever the *i*th layer is visible (second term), (3) the occupancy map of the *i*th layer is small wherever any farther layers are visible (third term), and (4) the alignment agrees with the prior motion constraint (fourth term).

The solution is found by using a modified version of the probabilistic image alignment solution proposed in [3], the difference here being the addition of the extra term in the cost function (second term) and the weighting introduced by the posterior visibility. The result is a iterated linear solution for the alignments parameters.

3.3 Computing Occupancy

Now, taking the alignment parameters to be fixed we consider the occupancy parameters of the *i*th layer and the associated cost

$$\begin{aligned} F(q,O_t^i(x)) &= \sum_{j=1}^{m} q^{ij}(\phi_t^{ij}x)\ln O_t^i(x) \\ &+ \left(\sum_{k=0}^{i-1} q^{kj}(\phi_t^{ij}x)\right)\ln(1-O_t^i(x)) + \ln P(O_t^i|O_{t-1}^i) \end{aligned} \tag{18}$$

We model the prior occupancy as a beta distribution

$$P(O_t^i(x)|O_{t-1}^i(x)) \propto {O_t^i}^{\alpha}(1-O_t^i)^{\beta} \tag{19}$$

where, $\alpha = O_{t-1}^i$ and $\beta = 1-\alpha$. This is for two reasons: occupancy is limited to values between zero and one, and the other terms in the cost are in the form of the logarithm of a beta distribution.

Thus we obtain a linear solution for occupancy.

$$O_t^i = \frac{\alpha + a}{1+a+b}, \quad a = \sum_{j=1}^{m} q^{ij}(\phi_t^{ij}x), \quad b = \sum_{j=1}^{m}\sum_{k=0}^{i-1} q^{ij}(\phi_t^{ij}x) \tag{20}$$

This solution *makes sense* since large values of visibility or prior occupancy (numerator) tend to increase the occupancy and large values of farther layer's visibilities (denominator) tend to reduce the occupancy.

3.4 Computing Appearance

The appearance is computed by optimising the cost

$$F(q,A_t^i(x)) \quad = \quad \sum_{j=1}^{m} -q^{ij}(\phi_t^{ij}x)\frac{(A_t^i(x)-I_t^j(\phi_t^{ij}x))^2}{2\sigma_I^2} - \frac{(A_t^i(x)-A_{t-1}^i(x))^2}{2\sigma_A^2} \tag{21}$$

where we have assumed a constant appearance transition model and a prior on appearance given by the normal distribution

$$P(A_t^i(x)|A_{t-1}^i(x)) \sim N(A_{t-1}^i(x),\sigma_A^2) \tag{22}$$

with mean given by the previous appearance and variance σ_A^2. The variance offers a control on how much we expect a layers appearance to vary over time (useful in the case of non-rigid motion).

We obtain a linear solution for the appearance

$$A_t^i(x) = \frac{\frac{1}{\sigma_I^2}\sum_{j=1}^{m} q^{ij}(\phi_t^{ij}x)I_t^j(\phi_t^{ij}x) + \frac{1}{\sigma_A^2}A_{t-1}^i(x)}{\frac{1}{\sigma_I^2}\sum_{j=1}^{m} q^{ij}(\phi_t^{ij}x) + \frac{1}{\sigma_A^2}} \tag{23}$$

Thus, the appearance is updated during the M-step by a weighted blend of the prior appearance and the current images; the blending weights change each iteration and depend on the visibilities and alignments.

4 Algorithm and Implementation

At each time t the layer parameters are propagated from those computed at the previous time according to the mode of the prior distributions. This procedure acts much like a prediction and serves as the starting point of the EM algorithm. The next stage is to reconsider the order of the model, i.e does the model explain the data well and if not should there be additional layers. We take quite a simple approach to this which involves considering how well the model explains the data compared to a model which assumes a uniform data likelihood. More precisely, for each pixel in each view we compute the evidence for the layered model from the following

$$P(L_t|I_t^j(x)) = \sum_{i=0}^{n} P(I_t^j(x)|V_t^j(x)=i)P(V_t^j(x)=i)P(L_t) \tag{24}$$

and the evidence from an alternative and uninformative model M

$$P(M|I_t^j(x)) = P(I_t^j(x)|M)P(M) \tag{25}$$

We set the prior for the layered model as 0.99 and the prior for the alternative as 0.01. By flagging pixels where $P(L_t|I_t^j(x)) \leq P(M|I_t^j(x))$ we obtain a mask for each images of which pixels are poorly explained under the current model. By looking for locally dense clusters of unexplained pixels of a given minimum size a new layer is initialised by setting the occupancy to 0.8 inside the region initialised and taking the current image pixel values in that region as the appearance.

For layers that appear in two or more views the depth-ordering is easily obtained from the disparity; a new layer that exists only in one view is given a nominal depth value that is refined over time. Any layers which move outside the range of all views are deleted and new layers instantiated before solving for the new parameters. Figure 3 illustrates the full algorithm.

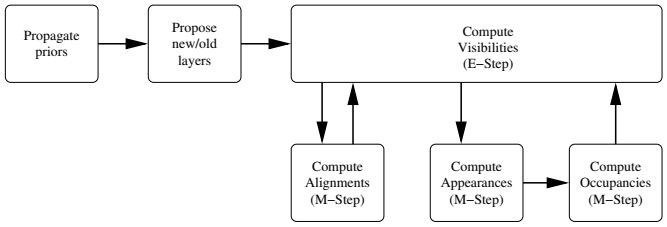

Figure 3: The steps shown are performed as one cycle per frame. However, the Expectation and Maximisation steps may be iterated; we found that two or three iterations is usually sufficient for convergence.

The algorithm begins with only one layer, the background, which has been learnt using robust statistics on samples of each pixel. As the algorithm iterates over time new layers are proposed, and spatial alignments in multiple views established.

In our implementation the alignment for the background is always the identity transformation (static cameras) but the framework is not restricted in this respect. To cope with pan-tilt-zoom parameters we could change ϕ_0 to be a four or five degree of freedom 2D homography. The alignment of all foreground layers is modelled using six degree of freedom affinities. The appearance model in our implementation is restricted to monchrome rather than full colour information.

5 Results

In this section we show results from applying the algorithm to various video data. Results are shown for tracking in one and two views. In the figures the boundaries drawn over the images indicate where a layer's occupancy passes through the value 0.5 and serve to show the layer's computed extent.

Figure 4 shows a sequence taken from a single viewpoint in which two people are wandering around and one passes in front of the other causing near total occlusion. Note the non-rigid motion of the arms and legs relative to the torso. The results show this is handled well. In addition the figure shows the progression of the occupancy and appearance maps of the two foreground layers.

To demonstrate the algorithm in a more demanding scenario, we applied it to two-views of a football game in which new players are entering the scene in both views as time goes by (figure 5). Although there are more parameters to solve for in two views than in one, there is better scope for direct layer measurement because even if part of a layer is occluded in one view it may be visible in another. The result is that the appearance and occupancy can be estimated even though an object is may be hidden in some views.

Our original motivation for developing motion segmentation and tracking algorithm was for novel view synthesis. The knowledge of occluding boundaries, and the temporal propagation of these, can lead to more efficient and better quality novel views. Given a precomputed layered segmentation and the corresponding occupancy and visibilty indicator variables, we can easily generate new views in real-time, using visibilty as an alpha-matte, by varying the layer alignment parameters in a manner consistent with the novel viewpoint's epipolar geometry. The pre-learned background is interpolated using the method of [9]. Examples for the football sequence are shown in figure 6.

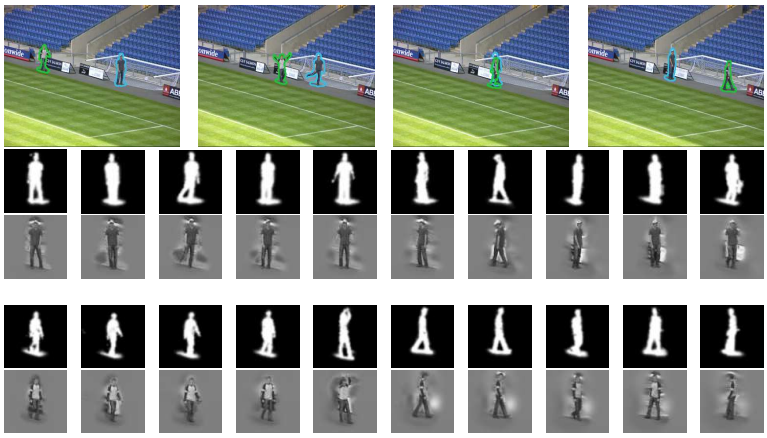

Figure 4: Single view example: (top) segmentation showing large occlusion and non-rigid motion; (bottom) occupancy and appearance maps of the two foreground layers from the single view tracking example. It can be seen that in spite of occlusion, occupancy and appearance persist.

Figure 5: Two view example: the top row shows extracts from one sequence, while the bottom row the same time-instants from a different viewpoint. Note: (i) automatic creation of new layers from a single layer (leftmost) to multiple layers; (ii) new layers being created as players enter one field of view (eg, yellow box, third column); (iii) correct treatment of occlusion (eg, cyan box, second column).

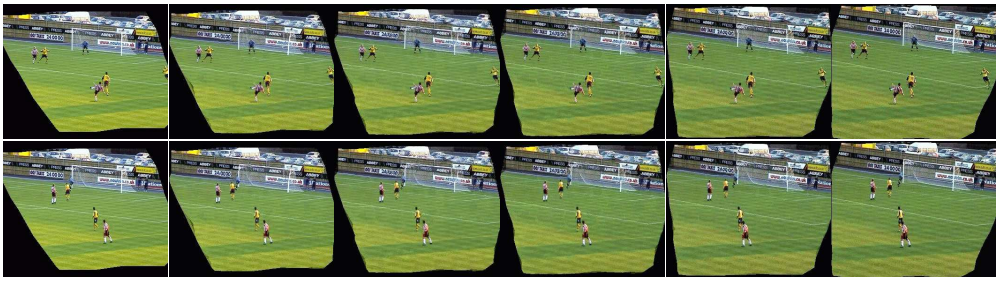

Figure 6: Creating novel views: each row shows novel views that interpolate between the two cameras. (top) frame 40 from a 100 frame sequence; (bottom) frame 70 from the same sequence.

6 Conclusion

We have presented a novel layered representation for multiple views of dynamic scenes, in which the single view problem is a special case. A MAP solution for sequentially estimating the parameters of the model was described with the facility of automatically initialising new layers. The result is a procedure which can track multiple moving objects over a number of views with a complete representation of the salient properties. In particular, the model maintains a persistent representation of occupancy in spite of occlusions and integrates measurements from each view according to visibility.

In principle the approach does not require a particular alignment parameterisation but in our implementation we assume affine alignment. Thus it admits planar like objects or relatively short baselines between views. One weakness of our current implementation is the restriction that the background is modelled as a single "special" layer, behind all others. In many scenes, there is in principle no reason why the background could not be modelled as as a set of planar layers itself together with individual alignment parameters; this would then admit the posibilty of parts of the background (eg the goal posts) occluding the foreground.

References

[1] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum likelihood estimation of mixture models and MDL encoding. In *Proc. of International Conference on Computer Vision*, pages 777–784, 1995.

[2] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, June 1998.

[3] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Dec 2001.

[4] K. Connor and I. Reid. A multiple view layered representation for dynamic novel view synthesis. In *in proc. British Machine Vision Conference*, 2003.

[5] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 17:474–487, 1995.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc.*, 39:1–38, 1977.

[7] A. D. Jepson, D. J. Fleet, and M. J. Black. A layered motion representation with occlusion and compact spatial support. In *Proc. of European Conference on Computer Vision*, pages 692–706, 2002.

[8] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[9] Maxime Lhuillier and Long Quan. Image interpolation by joint view triangulation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collines, USA, June 1999.

[10] H. Tao, H. S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):75–89, January 2002.

[11] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 983–990, 1999.

[12] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 3(5):625–638, Sep 1994.

[13] Yue Zhou and Hai Tao. A background layer model for object tracking through occlusion. In *Proc. of International Conference on Computer Vision*, October 2003.