Image and Vision Computing 28 (2010) 1022–1030

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Multiview segmentation and tracking of dynamic occluding layers

Ian Reid *, Keith Connor

Dept. of Engineering Science, University of Oxford, Oxford, UK

ARTICLE INFO

Article history:
Received 22 August 2007
Received in revised form 10 September 2009
Accepted 17 September 2009

Keywords:
Multiple views
Segmentation
Novel view synthesis

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 sequential estimation problem that is equivalent to tracking multiple objects in a given number views. Expectation–Maximisation is used to establish layer posterior probabilities for both occupancy and visibility, which are represented distinctly. 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. We demonstrate an application of this representation to dynamic novel view synthesis.

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 [1].

Layers have mainly found use in the representation of monocular video sequences, typically for applications concerned with video coding [2]. In previous work [3] 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 [1] 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. Subsequent approaches by Darrell and Pentland [4] and by Ayer and Sawhney [5] employ a probabilistic mixture model formulation to compute the maximum likelihood layer parameters by simultaneously computing segmentation and motion. In [6] model selection is also introduced, to determine the number of layers automatically.

A particular variant among previous approaches is whether or not occlusion is fully accounted for. The persistent representation

* Corresponding author.
E-mail address: ian@robots.ox.ac.uk (I. Reid).

0262-8856/$ - see front matter Crown Copyright © 2009 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.09.007

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. [8] 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 [9] 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, and the use of these constraints produces a more efficient algorithm; (ii) Frey and Jojic demonstrated only translational changes in layers. For this case, the number of possible poses is manageable (i.e., number of image positions), but this grows exponentially with the number of degrees of freedom of the transformation. So while their framework is not restricted to translation only, 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 [10] 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.

A common goal in monocular dynamic segmentation methods such as those above is to be able to remove unwanted foreground objects, or indeed to replace the background, in a process known as compositing commonly achieved by *blue screening* in television and film production. The corresponding work in computer vision has concentrated on segmentation of the image into two layers, foreground and background, but with an emphasis on extracting a high quality *alpha matte*, a map indicating the proportion of foreground present at each pixel in a scene [11–15]. Such an alpha matte can then be used for re-compositing, or creating novel views (though not necessarily physically plausible). Matting for multiple layers, automatically extracted, has recently been described in [16].

While the approaches above compute motion layers for a single view of a dynamic scene, less commonly layers are extracted from a binocular pair of views, in order to represent the static structure of the scene as a set of planar "layers". Examples of this include [17,18]. In these papers the transformation associated with a layer maps its location between spatially separated views, rather than its dynamic position. In contrast, our work considers both motion and structure.

Various work has recognised the virtues of combining segmentation with multiview structure estimation. Goldlucke [19] combines 3D structure estimation with background separation, and likewise Kolmogorov [20] combines binocular stereo with bi-layer (i.e., foreground/background) segmentation. These papers share a number of similarities with our work, but differ in that they are more tightly locked in to a multiview framework (e.g., in the former, a point is either background, or must have a disparity; there is no concept of a layer existing in only one view). While these efforts concentrated on the small baseline problem, more recently some authors have tackled the wide baseline problem in a combined segmentation and reconstruction framework [21,22]. Segmentation of foreground and background yields a set of silhouettes from disparate view which can be combined to yield the so-called visual hull, and which can subsequently be used to produce novel views by re-rendering the 3D reconstruction.

1.2. Roadmap

The remainder of the paper is structured as follows: we begin in Section 2 with a description of our generative layered model and then in Section 3 suggest a means of solving for the maximum *a posteriori* layer parameters via generalised Expectation–Maximisation. Section 4 discusses the algorithm that results and implementation details. In Section 5 we show segmentation results for one-view and two-view scenes, and show examples of novel view interpolation created using this segmentation. We conclude in Section 6.

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.

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 alignment 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); Fig. 1 illustrates the meaning of the layer parameters. The layered model at time t is denoted as $L_t = (L_t^0, L_t^1, \ldots, L_t^n)$, where

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

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

$$\Phi_t^i = \left\{\phi_t^{ij}\right\}, \quad j \in [1, \ldots, m] \tag{2}$$

If a layer is not within the field of view of a particular camera, the alignment parameters project the layer outside the image. Of course in this case there is no data to constrain the alignment directly – it will be indirectly constrained by observations in other images – analogous to if it were completely occluded.

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

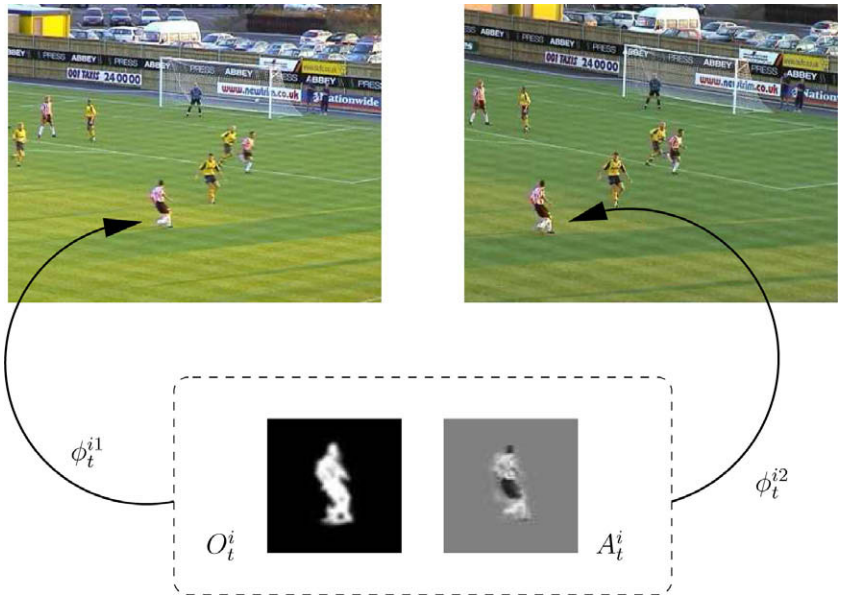

Fig. 1. The parameters that describe a layer are occupancy O_t^i (represented by a probabilistic map), appearance A_t^i (represented by an intensity map), and alignment ϕ_t^{ij} (a transformation relating the coordinate frame of the ith layer to the jth image). In the case shown there are two images, and so two alignments mapping from layer coordinates to image coordinates.

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

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:

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

Interpret the visibility $P\left(V_t^j(x)=i\right)$ 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

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

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 general layer model is illustrated by the network shown in Fig. 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 Fig. 2 can be factored as

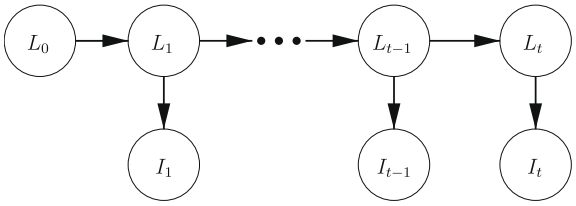

Fig. 2. A Bayesian network illustrates the problem of tracking the layered representation L_t given the observations (current images) I_t and takes the form of a hidden Markov model.

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

Rather than attempting to solve for the entire network, we adopt a recursive maximum *a posteriori* estimation approach, solving at time t for the current parameters, given the previous values by maximising $P(I_t|L_t)P(L_t|L_{t-1})$. Taking logs we obtain the equivalent maximisation of $F(L_t)$,

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

3.1. EM algorithm

The Expectation–Maximisation algorithm [23] 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

$$F(L_t) = \ln P(I_t|L_t) + \ln P(L_t|L_{t-1}) \tag{8}$$

$$= \ln\sum_V P(V, I_t|L_t) + \ln P(L_t|L_{t-1}) \tag{9}$$

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

$$\geqslant \sum_V Q(V)(\ln P(V, I_t|L_t) - \ln Q(V)) + \ln P(L_t|L_{t-1}) \tag{11}$$

where the bound arises directly from Jensen's inequality. Letting

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

be the lower bound, it can be shown that equality between F and f 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:

$$Q^{(k)}(V) = P(V|I_t, L_t^{(k-1)}) \quad (13)$$

M-step:

$$L_t^{(k)} = \arg\max_{L_t} \sum_V Q^{(k)}(V) \ln P(V, I_t|L_t) + \ln P(L_t|L_{t-1}) \quad (14)$$

Despite appearances, solving Eqs. (13) and (14) is much easier than solving Eq. (7) because the parameters of each layer can be solved for separately, as detailed below.

In the following the conditional dependence on the current layer parameters L_t is implicit. We assume that, conditioned on the hidden visibility variables and layer parameters, pixel intensities are independent. In the case that a layer keeps the same appearance but changes pose, the assumption is effectively that the per-pixel noise is independent. This is a common, expedient and justifiable assumption. A weakness, however, is that we do not model correlated changes, such as overall changes in lighting. The model can capture these at the level of individual pixels (by treating such changes as noise), but this is naturally a weaker representation of the phenomenon.

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

$$q^{ij}(x) = P\left(V_t^j(x) = i|I_t^j(x)\right) \quad (15)$$

$$\propto P\left(I_t^j(x)|V_t^j(x) = i\right)P\left(V_t^j(x) = i\right) \quad (16)$$

where the prior visibility is given by Eq. (5).

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

$$f(q, L_t) = \sum_{i=0}^{n}\sum_{j=1}^{m}\sum_x q^{ij}(x) \ln P\left(I_t^j(x)|V_t^j(x) = i\right) + q^{ij}(x) \ln P\left(V_t^j(x)\right) + \ln P\left(L_t^i|L_{t-1}^i\right) \quad (17)$$

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:

$$\sum_{i=0}^{n}\sum_{j=1}^{m}\sum_x q^{ij}\left(\phi_t^{ij}x\right) \ln P\left(I_t^j(\phi_t^{ij}x)|V_t^j(\phi_t^{ij}x) = i\right) + q^{ij}\left(\phi_t^{ij}x\right) \ln O_t^i(x) + \left(\sum_{k=0}^{i-1} q^{kj}(\phi_t^{ij}x)\right) \ln(1 - O_t^i(x)) + \ln P\left(\Phi_t^i|\Phi_{t-1}^i\right) + \ln P\left(O_t^i|O_{t-1}^i\right) + \ln P\left(A_t^i|A_{t-1}^i\right) \quad (18)$$

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 ith layer's alignment with the jth view, the expression to maximise is:

$$f\left(q, \phi_t^{ij}\right) = \sum_x -q^{ij}\left(\phi_t^{ij}x\right)\frac{\left(A_t^i(x) - I_t^j(\phi_t^{ij}x)\right)^2}{2\sigma_I^2} + q^{ij}\left(\phi_t^{ij}x\right) \ln O_t^i(x) + \left(\sum_{k=0}^{i-1} q^{kj}\left(\phi_t^{ij}x\right)\right) \ln\left(1 - O_t^i(x)\right) + \ln P\left(\phi_t^{ij}|\phi_{t-1}^{ij}\right) \quad (19)$$

In words, the optimum alignment for the ith layer with the jth image is found when (1) the appearance map agrees with the image data wherever the ith layer is visible (first term), (2) the occupancy map is large wherever the ith layer is visible (second term), (3) the occupancy map of the ith 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 [24], 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 alignment parameters.

3.3. Computing occupancy

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

$$f\left(q, O_t^i(x)\right) = \sum_{j=1}^{m} q^{ij}\left(\phi_t^{ij}x\right) \ln O_t^i(x) + \left(\sum_{k=0}^{i-1} q^{kj}\left(\phi_t^{ij}x\right)\right) \ln\left(1 - O_t^i(x)\right) + \ln P\left(O_t^i|O_{t-1}^i\right) \quad (20)$$

We model the prior occupancy as a beta distribution

$$P\left(O_t^i(x)\middle|O_{t-1}^i(x)\right) \propto {O_t^i}^{\alpha}\left(1 - O_t^i\right)^{\beta} \quad (21)$$

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.

$$O_t^i = \frac{\alpha + a}{1 + a + b}, \quad a = \sum_{j=1}^{m} q^{ij}\left(\phi_t^{ij}x\right), \quad b = \sum_{j=1}^{m}\sum_{k=0}^{i-1} q^{ij}\left(\phi_t^{ij}x\right) \quad (22)$$

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

$$f\left(q, A_t^i(x)\right) = \sum_{j=1}^{m} -q^{ij}\left(\phi_t^{ij}x\right)\frac{\left(A_t^i(x) - I_t^j(\phi_t^{ij}x)\right)^2}{2\sigma_I^2} - \frac{\left(A_t^i(x) - A_{t-1}^i(x)\right)^2}{2\sigma_A^2} \quad (23)$$

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

$$P\left(A_t^i(x)\middle|A_{t-1}^i(x)\right) \sim N\left(A_{t-1}^i(x), \sigma_A^2\right) \tag{24}$$

with mean given by the previous appearance and variance σ_A^2. The variance offers a control on how much we expect a layer's appearance to vary over time (useful in the case of non-rigid motion). Both parameters, σ_I and σ_A were set empirically in our experiments, but could potentially be learned.

We obtain a linear solution for the appearance

$$A_t^i(x) = \frac{\frac{1}{\sigma_I^2}\sum_{j=1}^m q^{ij}\left(\phi_t^{ij}x\right)I_t^j\left(\phi_t^{ij}x\right) + \frac{1}{\sigma_A^2}A_{t-1}^i(x)}{\frac{1}{\sigma_I^2}\sum_{j=1}^m q^{ij}\left(\phi_t^{ij}x\right) + \frac{1}{\sigma_A^2}} \tag{25}$$

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 posterior 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

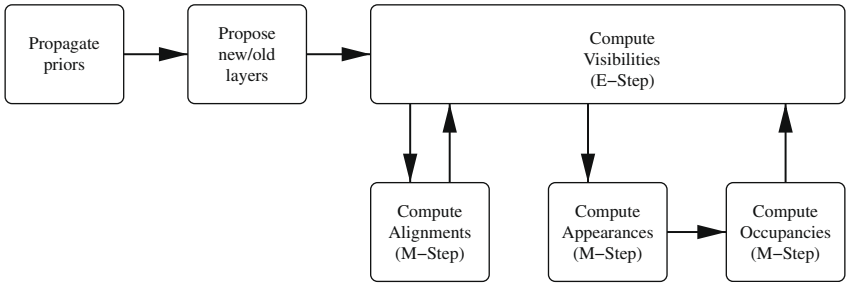

Fig. 3. The steps shown are performed as one cycle per frame. However, the Expectation and Maximisation steps may be iterated; we found that two or three iterations is usually sufficient for convergence.

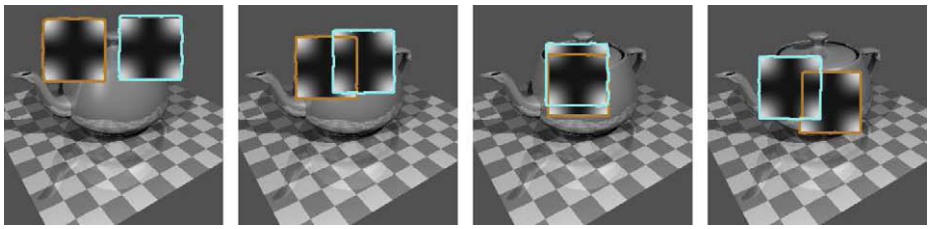

Fig. 4. Single view example: a synthetic sequence showing large occlusion and rigid motion.

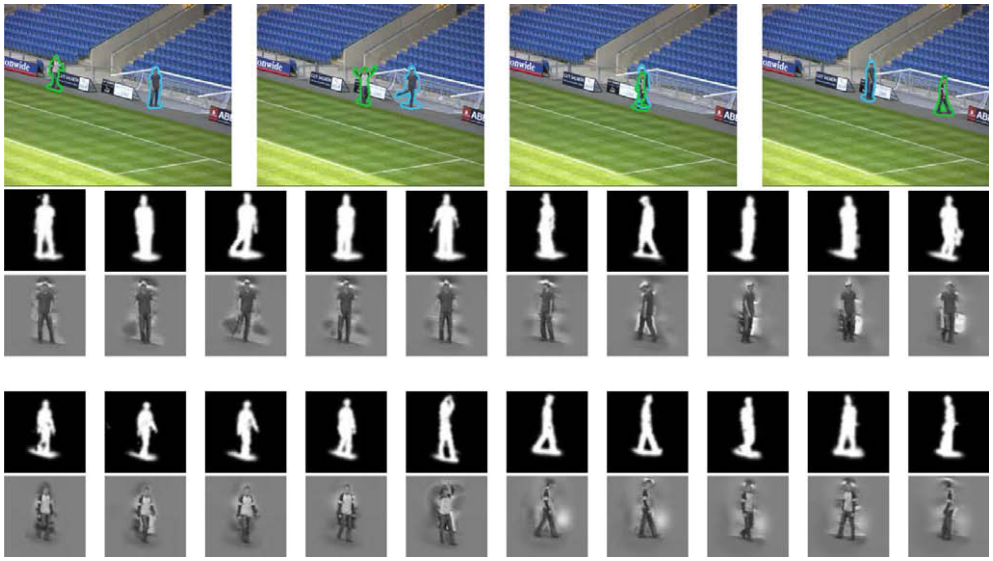

Fig. 5. Single view example: (top) segmentation showing large occlusion and non-rigid motion; (bottom) occupancy and appearance maps of the two foreground layers from the single view tracking example. Note the complete crossing of the individuals in the third frame at the top, which corresponds approximately to the occupancy and appearance images 5th and 6th from the left, which persist in spite of the near total occlusion.

each view we compute the evidence for the layered model from the following:

$$P\left(L_t|I_t^j(x)\right) = \sum_{i=0}^{n} P\left(I_t^j(x)|V_t^j(x)=i\right)P\left(V_t^j(x)=i\right)P(L_t) \quad (26)$$

and the evidence from an alternative and uninformative model M

$$P\left(M|I_t^j(x)\right) = P\left(I_t^j(x)|M\right)P(M) \quad (27)$$

We set the prior for the layered model as 0.99 and the prior for the alternative as 0.01. By flagging pixels where $P\left(L_t|I_t^j(x)\right) \leqslant P\left(M|I_t^j(x)\right)$ we obtain a mask for each image 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. Fig. 3 illustrates the full algorithm.

The algorithm begins with only one layer, the background. We assume that a good model of this has been learnt offline. In practice

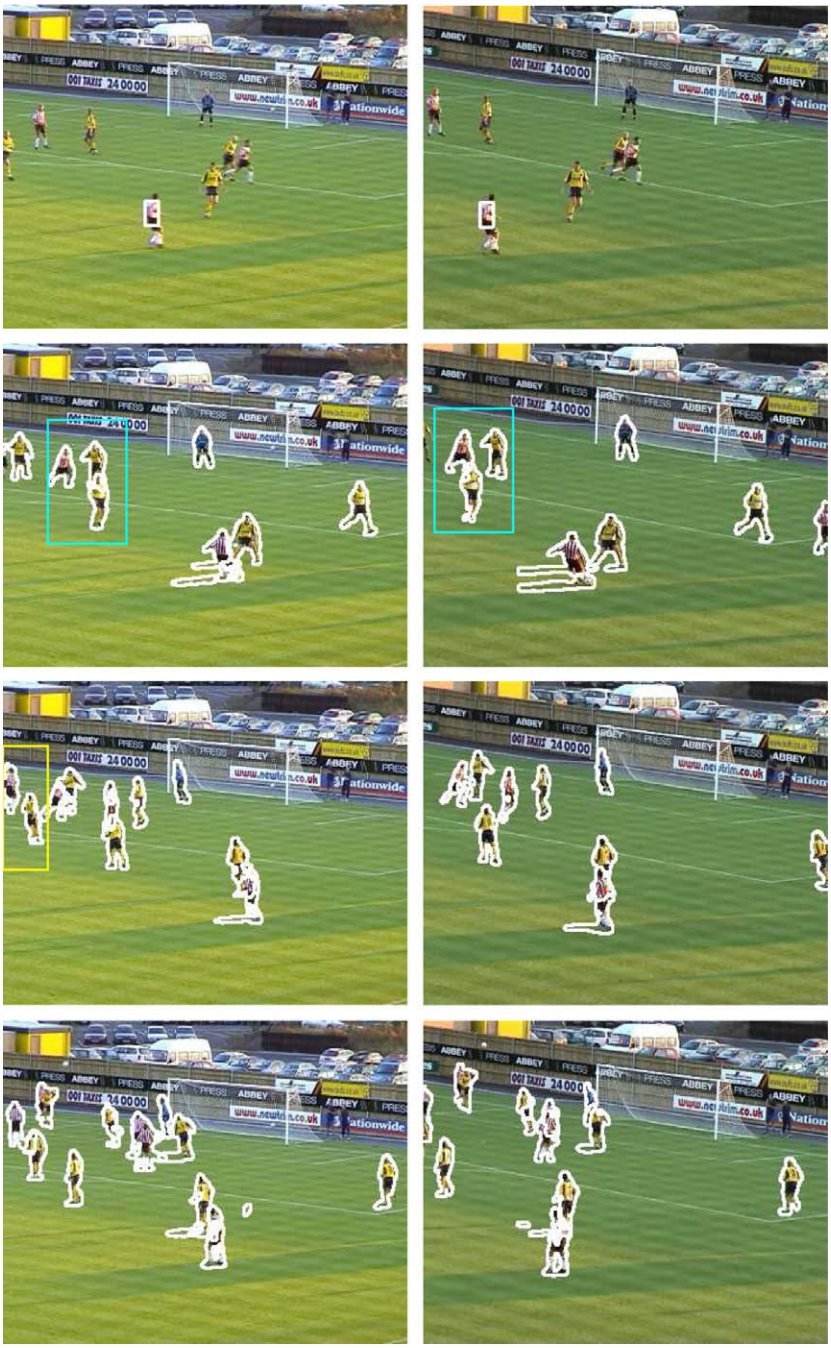

Fig. 6. Two view example: the left-hand column shows extracts from one sequence, while the right-hand column shows the same time instants from a different viewpoint. Note: (i) automatic creation of new layers from a single layer (top pair) to multiple layers; (ii) new layers being created as players enter one field of view (e.g., yellow box, third row); (iii) correct treatment of occlusion (e.g., cyan box, second row).

we achieve this by observation over an extended period, taking the most median value per pixel, relying on the assumption that the background is visible more frequently than any other objects. While not always true, it is reasonable in our main application area, and in any case a more sophisticated process could easily be incorporated. The background model is not updated over time.

At each new frame, Eqs. (26) and (27) are used to propose new single-view layers. Each single-view layer (either new or pre-existing) is then checked against the other views to determine if there is a correspondence between single-view layers in more than one view. If so, the two are merged to form a multiview layer, otherwise they persist as single view layers with undefined alignment for the other views.

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 monochrome 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.

In the first experiment the occlusion handling claim is tested using synthetic data in which two textured squares pass over each other on a textured background (Fig. 4). Although this example exhibits a large amount of occlusion, occupancy is accurately maintained and tracking continues after the occlusion. Often, in trackers which do not handle occlusion, this situation would not allow the exposure of the previously occluded object to be predicted by the model.

Fig. 5 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, it has been applied to two views of a football game in which new players are entering the scene in both views as time goes by (Fig. 6). 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 may be hidden in some views.

Our original motivation for developing a 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 visibility indicator variables, we can easily generate new views in real time, using visibility as an alpha matte, by varying the layer alignment parameters in a manner consistent with the novel viewpoint's epipolar geometry. The background in each view is pre-learned from the sequence by extracting the median colour; this expedient method works well under the assumptions that the camera is static, and that foreground objects move so that every background pixel is unoccluded for more time than it is occluded. Interpolated and extrapolated background layers are then generated using the method of [25]. Examples for the football sequence are shown in Fig. 7.

Another football sequence, captured with different cameras, is used to construct a sequence from a viewpoint midway between the cameras. The two background images for the cameras are shown in Fig. 8, and Fig. 9 shows a sequence of 32 frames rendered from the novel viewpoint.

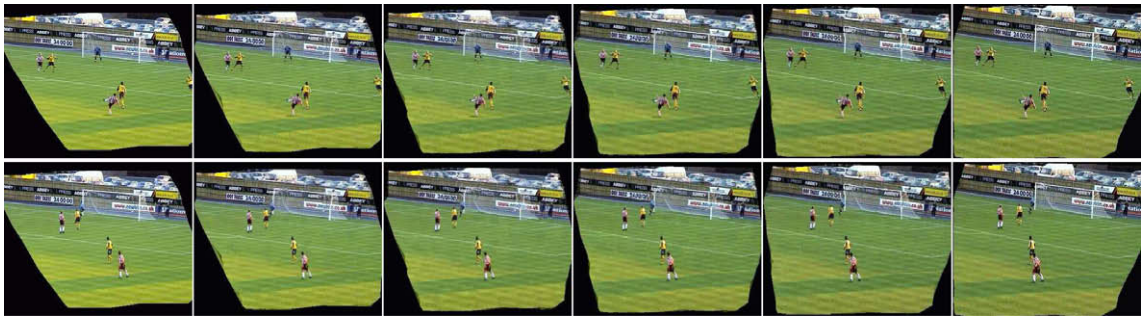

Fig. 7. Creating novel views: each row shows novel views that interpolate between the two cameras. (top) Frame 40 from a 100 frame sequence; (bottom) frame 70 from the same sequence.

Fig. 8. Two background images at the real viewpoints, extracted automatically from the sequences containing moving players.

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 a set of planar layers itself together with individual

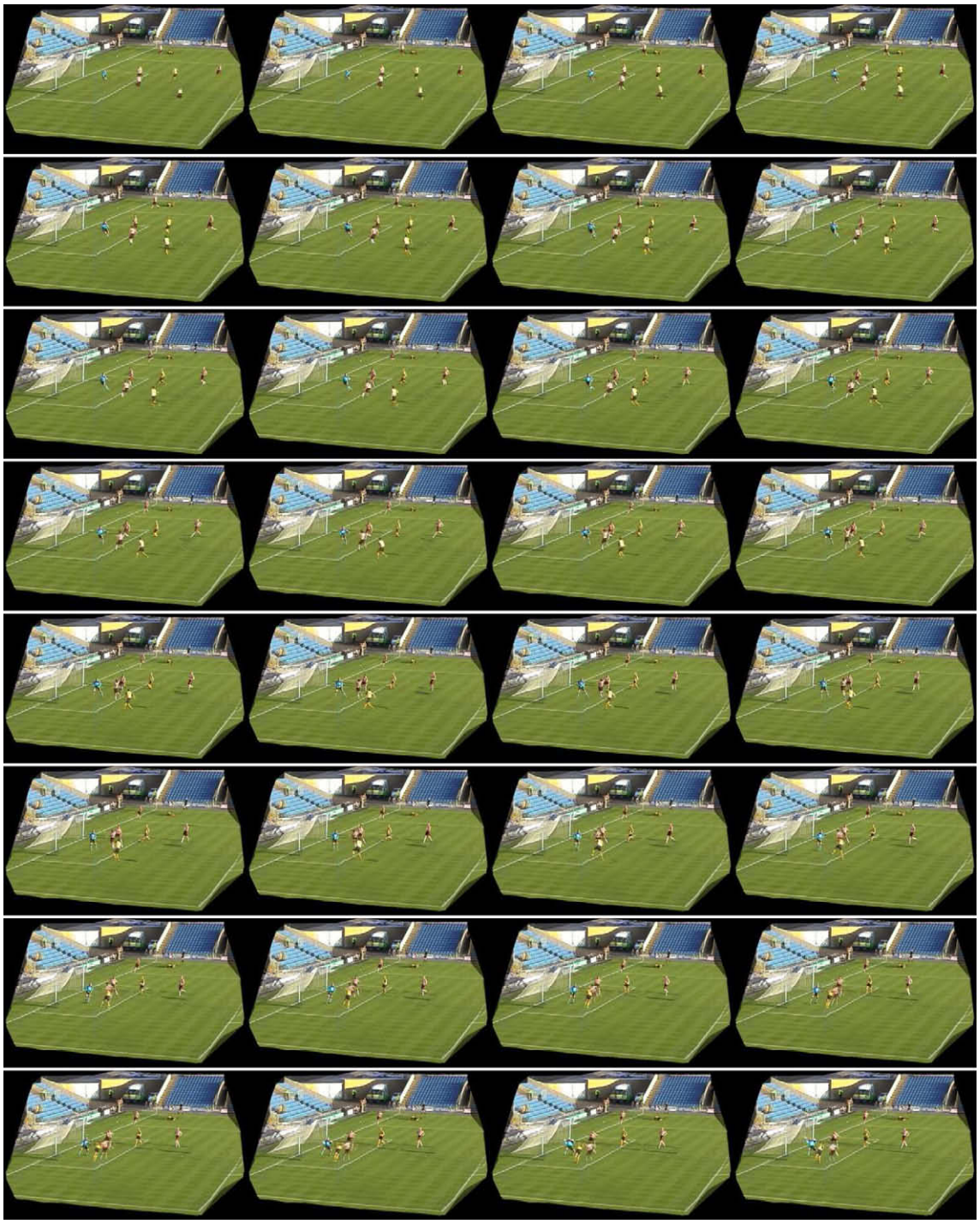

Fig. 9. A movie strip (left-to-right, top-to-bottom) showing a scene generated from a novel viewpoint mid-way between the two cameras. Note how the layers grow in the top row. Background rendering artifacts caused by the algorithm of [25] can be seen particularly at the top-left of the goal, while rendering errors caused by errors in the layered segmentation creep in towards the end of the sequence: e.g., note the players around the goal-keeper where the occupancy is incorrect causing ghosting and fuzzy edges to the players.

alignment parameters; this would then admit the possibility of parts of the background (e.g., the goal posts) occluding the foreground.

Acknowledgment

This work was supported by the European Framework 5 grant *EVENTS* [IST-1999-21125].

Appendix A. Supplementary data

Supplementary data, including video requences, associated with this article can be found, in the online version, at doi:10.1016/j.imavis.2009.09.007.

References

[1] J.Y.A. Wang, E.H. Adelson, Representing moving images with layers, IEEE Transactions on Image Processing: Image Sequence Compression 3 (5) (1994) 624–638 (special issue).
[2] F.C.N. Pereira, T. Ebrahimi, The MPEG-4 Book, Prentice-Hall, 2002.
[3] K. Connor, I. Reid, A multiple view layered representation for dynamic novel view synthesis, in: Proceedings of the British Machine Vision Conference, 2003.
[4] T. Darrell, A. Pentland, Cooperative robust estimation using layers of support, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (1995) 474–487.
[5] S. Ayer, H.S. Sawhney, Layered representation of motion video using robust maximum likelihood estimation of mixture models and MDL encoding, in: Proceedings of International Conference on Computer Vision, 1995, pp. 777–784.
[6] Y. Weiss, E.H. Adelson, A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1996, pp. 321–326.
[7] A.D. Jepson, D.J. Fleet, M.J. Black, A layered motion representation with occlusion and compact spatial support, in: Proceedings of European Conference on Computer Vision, 2002, pp. 692–706.
[8] H. Tao, H.S. Sawhney, R. Kumar, Object tracking with Bayesian estimation of dynamic layer representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (1) (2002) 75–89.
[9] N. Jojic, B. Frey, Learning flexible sprites in video layers, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001.
[10] Y. Zhou, H. Tao, A background layer model for object tracking through occlusion, in: Proceedings of International Conference on Computer Vision, 2003.
[11] Y.-Y. Chuang, A. Agarwala, B. Curless, D. Salesin, R. Szeliski, Video matting of complex scenes, in: Proceedings of Conference Computer Graphics and Interactive Techniques, ACM Press, 2002, pp. 243–248.
[12] Y. Wexler, A.W. Fitzgibbon, A. Zisserman, Bayesian estimation of layers from multiple images, in: Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, vol. 3, 2002, pp. 487–501.
[13] N.E. Apostoloff, A.W. Fitzgibbon, Bayesian video matting using learnt image priors, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2004.
[14] J. Xiao, M. Shah, Accurate motion layer segmentation and matting, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2005.
[15] P. Yin, A. Criminisi, J. Winn, I. Essa, Tree-based classifiers for bilayer video segmentation, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2007.
[16] D. Singaraju, R. Vidal, Interactive image matting for multiple layers, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2008.
[17] S. Baker, R. Szeliski, P. Anandan, A layered approach to stereo reconstruction, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998.
[18] P.H.S. Torr, R. Szeliski, P. Anandan, An integrated Bayesian approach to layer extraction from image sequences, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (3) (2001) 297–303.
[19] B. Goldlucke, M. Magnor, Joint 3d-reconstruction and background separation in multiple views using graph cuts, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2003, pp. 683–688.
[20] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, C. Rother, Probabilistic fusion of stereo with color and contrast for bi-layer segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (9) (2006) 1480–1492.
[21] P. Kohli, J. Rihan, M. Bray, P.H. Torr, Simultaneous segmentation and pose estimation of humans using dynamic graph cuts, International Journal of Computer Vision 79 (3) (2008) 285–298.
[22] J.-Y. Guillemaut, A. Hilton, J. Starck, J. Kilner, O. Grau, A Bayesian framework for simultaneous matting and 3d reconstruction, in: Proceedings of 3DIM, 2007, pp. 167–174.
[23] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39 (1977) 1–38.
[24] S. Baker, I. Matthews, Equivalence and efficiency of image alignment algorithms, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001.
[25] M. Lhuillier, L. Quan, Image interpolation by joint view triangulation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Fort Collines, USA, 1999.