Learning Layered Motion Segmentations of Video

M. Pawan Kumar P.H.S. Torr
Dept. of Computing
Oxford Brookes University
{pkmudigonda,philiptorr}@brookes.ac.uk
http://cms.brookes.ac.uk/computervision

A. Zisserman
Dept. of Engineering Science
University of Oxford
az@robots.ox.ac.uk
http://www.robots.ox.ac.uk/˜vgg

Abstract

We present an unsupervised approach for learning a generative layered representation of a scene from a video for motion segmentation. The learnt model is a composition of layers, which consist of one or more segments*. Included in the model are the effects of image projection, lighting, and motion blur. The two main contributions of our method are: (i) A novel algorithm for obtaining the initial estimate of the model using efficient loopy belief propagation; (ii) Using $\alpha\beta$-swap and α-expansion algorithms, which guarantee a strong local minima, for refining the initial estimate. Results are presented on several classes of objects with different types of camera motion. We compare our method with the state of the art and demonstrate significant improvements.*

1. Introduction

We present an approach for learning a generative layered representation from a video for motion segmentation. Our method is applicable to any video containing piecewise parametric motion, e.g. piecewise homography, without any restrictions on camera motion. It also accounts for the effects of occlusion, lighting and motion blur. For example, Fig. 1 shows one such sequence where a layered representation can be learnt and used to segment the walking person from the static background.

Many different approaches for motion segmentation have been reported in the literature. Important issues are: (i) whether the methods model occlusion; (ii) whether they model spatial continuity; (iii) whether they apply to multiple frames (i.e. a video sequence instead of a pair of images); (iv) whether they are independent of keyframes for initialization. For instance, the approaches described in [2, 4] are examples of methods which do not model occlusion. Thus, they tend to over count the data when learning the model.

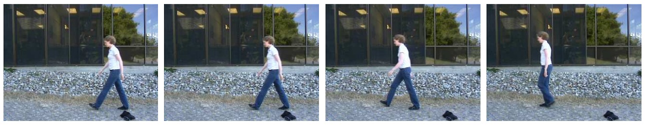

Figure 1: *Four intermediate frames of a 40 frame video sequence of a person walking sideways where the camera is static. Given the sequence, the generative model which best describes the person and the background is learnt in an unsupervised manner. Note that the arm always partially occludes the torso.*

Amongst the methods which do model occlusion are those which use a layered representation [14]. One such approach, described in [16], divides a scene into (almost) planar regions for occlusion reasoning. Torr *et al.* [12] extend this representation by allowing for parallax disparity. However, these methods rely on a keyframe for the initial estimation. Other approaches [6, 17] overcome this problem by using layered flexible sprites. A flexible sprite is a 2D appearance map and matte (mask) of an object which is allowed to deform from frame to frame according to pure translation. Winn *et al.* [19] extend the model to handle affine deformations. These methods do not model spatial continuity which leads to non-contiguous segmentation when the foreground and background are similar in appearance (see Fig. 6(c)). Moreover, they do not model changes in appearance due to lighting and motion blur. Most of these approaches, namely those described in [4, 6, 12, 14, 16], use either EM or variational methods for learning the parameters of the model which makes them prone to local minima.

Wills *et al.* [18] noted the importance of spatial continuity when learning the regions in a layered representation. Given an initial estimate, they learn the shape of the regions using the powerful α-expansion algorithm [3] which guarantees a strong local minima. However, their method does not deal with multiple frames. In our earlier work [7], we describe a similar motion segmentation approach to [18] for a video sequence. Like [10], this automatically learns a generative model of an object. However, the method depends on a keyframe to obtain an initial estimate of the model. This has the disadvantage that points not visible in the keyframe are not included in the model, which leads to incomplete segmentation.

We present a model which does not suffer from the problems mentioned above, i.e. (i) it models occlusion; (ii) it models spatial continuity; (iii) it handles multiple frames; (iv) it is learnt independent of keyframes. An initial estimate of the model is obtained using efficient loopy belief propagation [5]. Given this estimate, the shape of the segments, along with the layering, is learnt by minimizing an objec-

$\mathbf{D}$	Data (RGB values of all pixels in every frame of a video).
n_F	Number of frames.
n_P	Number of segments p_i including the background.
l_i	Layer number of segment p_i.
$\mathbf{\Theta}_M^i$	Matte for segment p_i.
$\mathbf{\Theta}_A^i$	Appearance parameter for segment p_i.
$\mathbf{\Theta}_{Ti}^j$	Transformation $\{x, y, s_x, s_y, \phi\}$ of segment p_i to frame j.
$\mathbf{\Theta}_{Li}^j$	Lighting parameters $\{\mathbf{a}_i^j, \mathbf{b}_i^j\}$ of segment p_i to frame j.
$\mathbf{\Theta}$	Model parameters $\{n_P, \mathbf{\Theta}_M, \mathbf{\Theta}_A, l_i; \mathbf{\Theta}_T, \mathbf{\Theta}_L\}$.

Table 1: *Parameters of the layered representation.*

tive function using $\alpha\beta$-swap and α-expansion algorithms [3]. We present results on several classes of objects with different types of camera motion and compare them with the state of the art.

In the next section, we describe the layered representation. In section 3, we present a four stage approach to learn the parameters of the layered representation from a video. Such a model is particularly suited for applications like motion segmentation. Results are presented in section 4.

2. Layered representation

This section introduces the generative model for layered representation which describes the scene as a composition of layers. Any frame of a video can be generated from our model by assigning appropriate values to its parameters (see Fig. 2). It also provides the likelihood of that instance. The parameters of the model, summarized in table 1, can be divided into two sets: (i) those that describe the *latent image*, and (ii) those that describe how to generate the frames using the latent image.

The latent image consists of a set of *segments*, which are 2D patterns (specified by their shape and appearance) along with their layering. The layering determines the occlusion ordering. Thus, each layer contains a number of non-overlapping segments. The shape of a segment p_i is modelled as a binary matte $\mathbf{\Theta}_{Mi}$, of size equal to the frame of the video, such that $\mathbf{\Theta}_{Mi}(\mathbf{x}) = 1$ if $\mathbf{x} \in p_i$ and $\mathbf{\Theta}_{Mi}(\mathbf{x}) = 0$ otherwise.

The appearance $\mathbf{\Theta}_{Ai}(\mathbf{x})$ is the RGB value of points $\mathbf{x} \in p_i$. In order to model the layers, we assign a layer number l_i to each segment p_i such that segments belonging to the same layer share a common layer number. Furthermore, each segment p_i can partially or completely occlude segment p_k, if and only if $l_i > l_k$. In summary, the latent image is defined by the mattes $\mathbf{\Theta}_M$, the appearance $\mathbf{\Theta}_A$ and the layer numbers l_i of the n_P segments.

When generating frame j, we start from a latent image and map each point $\mathbf{x} \in p_i$ to $\mathbf{x}'$ using transformation $\mathbf{\Theta}_{Ti}^j$. This implies that points belonging to the same segment move rigidly together. The generated frame is then obtained by compositing the transformed segments in descending order of their layer numbers. For this paper, each transformation has five parameters: rotation, translation and anisotropic scale factors. The model accounts for the effects of lighting conditions on the appearance of a segment p_i using parameter $\mathbf{\Theta}_{Li}^j = \{\mathbf{a}_i^j, \mathbf{b}_i^j\}$. The change in appearance of the segment p_i in frame j due to lighting conditions is modelled as $\mathbf{c}(\mathbf{x}') = \mathtt{diag}(\mathbf{a}_i^j) \cdot \mathbf{\Theta}_{Ai}(\mathbf{x}) + \mathbf{b}_i^j$. The motion of segment p_i from frame $j-1$ to frame j, denoted by $\mathbf{m}_i^j$, can determined using the transformation parameters $\mathbf{\Theta}_{Ti}^{j-1}$ and $\mathbf{\Theta}_{Ti}^j$. This allows us to take into account the change in appearance due to motion blur as $\mathbf{c}(\mathbf{x}') = \int_0^T \mathbf{c}(\mathbf{x}' - \mathbf{m}_i^j(t))dt$, where T is the total exposure time when capturing the frame.

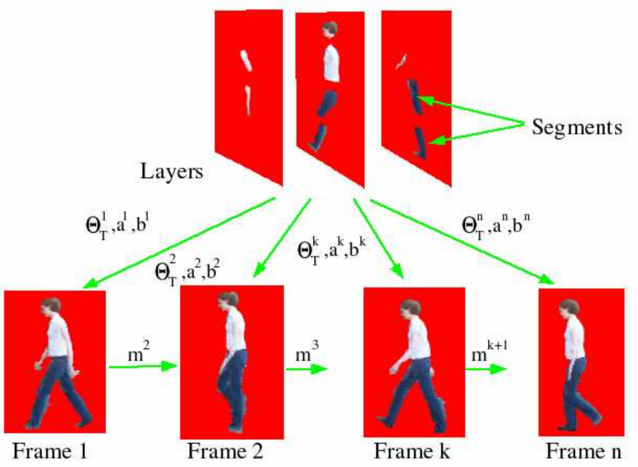

Figure 2: *The top row shows the various layers of a human model, the latent image in this case. Each layer consists of one of more segments whose appearance is shown. The shape of each segment is represented by a binary matte (not shown in the image). Any frame j can be generated using this representation by assigning appropriate values to its parameters. Note that the background is not shown.*

Posterior of the model: We represent the set of all parameters of the layered representation as $\mathbf{\Theta} = \{n_P, \mathbf{\Theta}_M, \mathbf{\Theta}_A, l_i; \mathbf{\Theta}_T, \mathbf{\Theta}_L\}$, where n_P is the total number of segments. Given data $\mathbf{D}$, i.e. the n_F frames of a video, the posterior probability of the model is given by

$$\Pr(\mathbf{\Theta}|\mathbf{D}) = \frac{\Pr(\mathbf{D}|\mathbf{\Theta})\Pr(\mathbf{\Theta})}{\Pr(\mathbf{D})} = \frac{1}{Z_{\mathbf{\Theta}}} \exp(-\Psi(\mathbf{\Theta}|\mathbf{D})). \tag{1}$$

The energy $\Psi(\mathbf{\Theta}|\mathbf{D})$ has the form

$$\Psi(\mathbf{\Theta}|\mathbf{D}) = \sum_{i=1}^{n_P} \sum_{\mathbf{x} \in p_i} \left(\mathcal{A}_i(\mathbf{x}) + \lambda_1 \sum_{\mathbf{y}} \left(-\mathcal{B}_i(\mathbf{x},\mathbf{y}) + \lambda_2 \mathcal{P}_i(\mathbf{x},\mathbf{y}) \right) \right), \tag{2}$$

where $\mathbf{x}$ and $\mathbf{y}$ are neighbouring points. The energy has two components: (i) the data log likelihood term which consists of the appearance term $\mathcal{A}_i(\mathbf{x})$ and the contrast term $\mathcal{B}_i(\mathbf{x},\mathbf{y})$, and (ii) the prior $\mathcal{P}_i(\mathbf{x},\mathbf{y})$ which encourages spatial continuity. The relative weight of the contrast and prior terms to , which encourages boundaries between two neighbouring segments to lie on edges in the frames, is given by λ_1. The parameter λ_2 is the weight given to spatial continuity. We use $\lambda_1 = \lambda_2 = 1$ in our experiments.

Let $\mathcal{I}_i^j(\mathbf{x})$ be the observed RGB values of point $\mathbf{x}' = \mathbf{\Theta}_{Ti}^j(\mathbf{x})$ in frame j and $\mathbf{c}_i^j(\mathbf{x}')$ be the generated RGB values. Here $\mathbf{\Theta}_{Ti}^j(\mathbf{x})$ is the projection of $\mathbf{x} \in p_i$ to frame j. The appearance term is given by

$$\mathcal{A}_i(\mathbf{x}) = \sum_{j=1}^{j=n_F} -\log(\Pr(\mathcal{I}_i^j(\mathbf{x})|\mathbf{x} \in p_i). \tag{3}$$

The likelihood of $\mathcal{I}_i^j(\mathbf{x})$ consists of two factors: (i) consistency of texture which is the conditional probability of $\mathcal{I}_i^j(\mathbf{x})$ given $\mathbf{x} \in p_i$ and is computed using histogram $\mathcal{H}_i$, and (ii) consistency of motion which measures how well the generated RGB values $\mathbf{c}_i^j(\mathbf{x}')$ match the observed values $\mathcal{I}_i^j(\mathbf{x})$. Thus,

$$\Pr(\mathcal{I}_i^j(\mathbf{x})|\mathbf{x} \in p_i) \propto \Pr(\mathcal{I}_i^j(\mathbf{x})|\mathcal{H}_i)\exp(-\mu(\mathbf{c}_i^j(\mathbf{x}') - \mathcal{I}_i^j(\mathbf{x}))^2), \tag{4}$$

where μ is some normalization constant. We use $\mu = 1$ in our experiments.

The contrast term pushes the projection of the boundary between parts to lie on image edges and has the form

$$\mathcal{B}_i(\mathbf{x},\mathbf{y}) = \begin{cases} \gamma_i(\mathbf{x},\mathbf{y}) & \text{if} \quad \mathbf{x} \in p_i, \mathbf{y} \notin p_i \\ 0 & \text{if} \quad \mathbf{x} \in p_i, \mathbf{y} \in p_i. \end{cases} \tag{5}$$

For this paper, we use

$$\gamma_i(\mathbf{x},\mathbf{y}) = \exp\left(\frac{-g_i^2(\mathbf{x},\mathbf{y})}{2\sigma^2}\right) \cdot \frac{1}{\mathtt{dist}(\mathbf{x},\mathbf{y})}, \tag{6}$$

where

$$g_i(\mathbf{x},\mathbf{y}) = \frac{1}{n_F}\sum_{j=1}^{n_F} |\mathcal{I}_i^j(\mathbf{x}) - \mathcal{I}_i^j(\mathbf{y})|. \tag{7}$$

Thus, $g_i(\mathbf{x},\mathbf{y})$ measures the difference between the RGB values $\mathcal{I}_i^j(\mathbf{x})$ and $\mathcal{I}_i^j(\mathbf{y})$ throughout the video sequence. The term $\mathtt{dist}(\mathbf{x},\mathbf{y})$, i.e. the euclidean distance between $\mathbf{x}$ and $\mathbf{y}$, gives more weight to the 4-neighbourhood of $\mathbf{x}$ than the rest of the 8-neighbourhood. The value of σ in equation (6) determines how the energy $\Psi(\mathbf{\Theta}|\mathbf{D})$ is penalized since the penalty is high when $g_i(\mathbf{x},\mathbf{y}) < \sigma$ and small when $g_i(\mathbf{x},\mathbf{y}) > \sigma$. Thus σ should be sufficiently large to allow for the variation in RGB values within a segment. In our experiments, we use $\sigma = 5$.

The prior is specified by an Ising model such that it encourages spatial continuity, i.e.

$$\mathcal{P}_i(\mathbf{x},\mathbf{y}) = \begin{cases} T & \text{if} \quad \mathbf{x} \in p_i, \mathbf{y} \notin p_i \\ 0 & \text{if} \quad \mathbf{x} \in p_i, \mathbf{y} \in p_i. \end{cases} \tag{8}$$

In the next section, we describe a four stage approach to calculate the parameters $\mathbf{\Theta}$ of the layered representation of an object, given data $\mathbf{D}$, by minimizing the energy $\Psi(\mathbf{\Theta}|\mathbf{D})$ (i.e. maximizing $\Pr(\mathbf{\Theta}|\mathbf{D})$. The method described is applicable to any scene with piecewise parametric motion.

3. Learning layered segmentation

Given a video, our objective is to estimate the parameters $\mathbf{\Theta}$, i.e. the latent image and the transformations, of the layered representation. We obtain these parameters in four stages. In the first stage, an initial estimate of the parameters is found. In the remaining stages, we alternate between holding some parameters constant and optimizing the rest as illustrated in table 2.

1. An initial estimate of the parameters $\mathbf{\Theta}$ is obtained by finding rigidly moving components between every pair of frames and combining them (§ 3.1).
2. The parameters $\mathbf{\Theta}_T$,$\mathbf{\Theta}_A$ and $\mathbf{\Theta}_L$ are kept constant and the mattes $\mathbf{\Theta}_M$ are optimized using $\alpha\beta$-swap and α-expansion algorithms. The layer numbers l_i are obtained (§ 3.2).
3. Using the refined values of $\mathbf{\Theta}_M$, the new appearance parameters $\mathbf{\Theta}_A$ are obtained (§ 3.3).
4. Finally, the transformation parameters $\mathbf{\Theta}_T$ and lighting parameters $\mathbf{\Theta}_L$ are re-estimated, keeping $\mathbf{\Theta}_M$ and $\mathbf{\Theta}_A$ unchanged (§ 3.4).

Table 2: *Estimating the parameters of the layered representation.*

3.1. Initial estimation of parameters

In this section, we describe a method to get an initial estimate of the parameters $\mathbf{\Theta}$ (excluding the layer numbers l_i) of the layered representation by computing the image motion. The method is robust to changes in appearance due to lighting and motion blur. The initial estimate is obtained using loopy belief propagation (LBP) and then refined using graph cuts. We develop a novel, efficient algorithm to determine rigidly moving components between every pair of consecutive frames which are then combined to get the initial estimate. This avoids the problem of finding only those segments which are present in one keyframe of the video.

In order to identify points that move rigidly together from frame j to $j+1$ in the given video $\mathbf{D}$, we need to determine the transformation that maps each point $\mathbf{x}$ in frame j to its position in frame $j+1$ (i.e. the image motion). However, at this stage we are only interested in obtaining a coarse estimate of the parameters $\mathbf{\Theta}$. We can reduce the complexity of the problem by dividing frame j into uniform patches $\mathbf{f}_k$ of size $m \times m$ pixels and determining their transformations φ_k. We use $m = 3$ for all our experiments.

The initial estimate of parameters is obtained in four stages: (i) finding a set of putative transformations φ_k for each fragment in frame j; (ii) finding the most likely transformation for each fragment in frame j using LBP (MAP estimation); (iii) combining rigidly moving components to determine $\mathbf{\Theta}_{Mi}$; (iv) computing the remaining parameters i.e. $\mathbf{\Theta}_{Ai}$, $\mathbf{\Theta}_{Ti}^j$ and $\mathbf{\Theta}_{Di}^j$. As the size of the patches is only 3×3 and we restrict ourselves to consecutive frames, it is sufficient to use transformations defined by a scale ρ_k, rotation θ_k and translation $\mathbf{t}_k$, i.e. $\varphi_k = \{\rho_k, \theta_k, \mathbf{t}_k\}$.

Finding putative transformations: We define a Markov random field (MRF) over the patches of frame j such that each site $\mathbf{n}_k$ of the MRF represents a fragment $\mathbf{f}_k$. Each label s_k of site $\mathbf{n}_k$ corresponds to a putative transformation φ_k. The likelihood $\psi(s_k)$ of a label measures how well the fragment $\mathbf{f}_k$ matches frame $j+1$ after undergoing transformation φ_k. The neighbourhood $\mathcal{N}_k$ of each site $\mathbf{n}_k$ is defined as its 4-neighbourhood. The prior over the transformations φ_k is modelled using pairwise potentials $\psi(s_k, s_l)$. We specify the prior that neighbouring patches tend to move rigidly together. The joint probability of the MRF is

$$\Pr(\varphi) = \frac{1}{Z} \prod_k \psi(s_k) \prod_{\mathbf{n}_l \in \mathcal{N}_k} \psi(s_k, s_l) \tag{9}$$

where φ is the set of transformations $\{\varphi_k, \forall k\}$.

By taking advantage of the fact that large scaling, translations and rotations are not expected between consecutive frames, we restrict ourselves to a small number of putative transformations. Specifically, we vary scale ρ_k from 0.8 to 1.2 in steps of 0.2, rotation θ_k from -0.3 to 0.3 radians in steps of 0.15 and translations $\mathbf{t}_k$ in x and y directions from -5 to 5 pixels and -10 to 10 pixels respectively in steps of 1. Thus, the total number of transformations is 3465.

The likelihood of fragment $\mathbf{f}_k$ undergoing transformation φ_k is modelled as $\psi(s_k) \propto \exp(\mathcal{L}(\mathbf{f}_k, \varphi_k))$, where $\mathcal{L}(\mathbf{f}_k, \varphi_k)$ is the normalized cross-correlation obtained using an $n \times n$ window around the fragment $\mathbf{f}_k$, after undergoing transformation φ_k, with frame $j+1$. When calculating $\mathcal{L}(\mathbf{f}_k, \varphi_k)$ in this manner, the $n \times n$ window is subjected to different degrees of motion blurring according to the motion specified by φ_k, and the best match score is chosen. This, along with the use of normalized cross-correlation, makes the likelihood estimation robust to lighting changes and motion blur. In all our experiments, we used $n = 5$. Since the appearance of a fragment does not change drastically between consecutive frames, normalized cross-correlation provides reliable match scores. Unlike [7], we do not discard the transformations resulting in a low match score. However, it will be seen later that this does not significantly increase the amount of time required for finding the MAP estimate of the transformations.

We want to assign the pairwise potentials such that neighbouring patches $\mathbf{f}_k$ and $\mathbf{f}_l$ which do not move rigidly together are penalized. However, we would be willing to take the penalty when determining the MAP estimate if it results in better match scores. Furthermore, we expect two patches separated by an edge to be more likely to move non-rigidly since they might belong to different segments. Thus, we define the pairwise potentials by a Potts model such that

$$\psi(s_k, s_l) = \begin{cases} 1 & \text{if rigid motion,} \\ \exp(-\zeta \nabla(\mathbf{f}_k, \mathbf{f}_l)) & \text{otherwise,} \end{cases} \tag{10}$$

where $\nabla(\mathbf{f}_k, \mathbf{f}_l)$ is the sum of the gradients of the neighbouring pixels $\mathbf{x} \in \mathbf{f}_k$ and $\mathbf{y} \in \mathbf{f}_l$, i.e. along the boundary shared by $\mathbf{f}_k$ and $\mathbf{f}_l$. We use $\zeta = 1$.

To handle occlusion, an additional label s_o is introduced for each site $\mathbf{n}_k$ which represents the fragment $\mathbf{f}_k$ being occluded in frame $j+1$. The corresponding likelihoods and pairwise potentials $\psi(s_o), \psi(s_k, s_o), \psi(s_o, s_k)$ and $\psi(s_o, s_o)$ are modelled as constants for all k. In our experiments, we used the values $0.1, 0.5, 0.5$ and 0.8 respectively.

MAP estimation: The MAP estimate of the transformation for each fragment is found by maximizing equation (9). We use loopy belief propagation (LBP) to find the posterior probability of a fragment $\mathbf{f}_j$ undergoing transformation φ_j. LBP is a message passing algorithm similar to the one proposed by Pearl [9] for graphical models with no loops. We describe the algorithm briefly [15].

The message that site $\mathbf{n}_k$ sends to its neighbour $\mathbf{n}_l$ at iteration t is given by

$$m^t_{kl}(s_k) = \sum_{s_k} \left(\psi(s_k, s_l)\psi(s_k) \prod_{\mathbf{n}_d \in \mathcal{N}_k \backslash \mathbf{n}_l} m^{t-1}_{dk}(s_k) \right). \tag{11}$$

All messages are initialized to 1, i.e. $m^0_{kl}(s_k) = 1$, for all k and l. The belief (posterior) of a fragment $\mathbf{f}_k$ undergoing transformation φ_k after T iterations is given by

$$b(s_k) = \psi(s_k) \prod_{\mathbf{n}_l \in \mathcal{N}_k} m^T_{lk}(s_k)\,. \tag{12}$$

The termination criterion is that the rate of change of all beliefs falls below a certain threshold. The label s^*_k that maximizes $b(s_k)$ is selected for each fragment thus, providing us a robust estimate of the image motion.

The time complexity of LBP is $O(nH^2)$, where n is the number of sites in the MRF and H is the number of labels per site, which makes it computationally infeasible for large H. However, since the pairwise potentials of the MRF are defined by a Potts model as shown in equation (10), the runtime of LBP can be reduced to $O(nH)$ using the method described in [5].

Another limitation of LBP is that it has memory requirements of $O(nH)$. To overcome this problem, we use a variation of the coarse to fine strategy suggested in [13]. This allows us to solve $O(\log(H)/\log(h))$ problems of h labels instead of one problem of H labels, where $h \ll H$. Thus, the memory requirements are reduced to $O(nh)$. The time complexity is reduced further from $O(nH)$ to $O(\log(H)nh/\log(h))$.

The basic idea of the coarse to fine strategy is to group together similar labels (differing slightly only in translation) to obtain h *representative* labels ϕ_k. We now define an MRF where each site $\mathbf{n}_k$ has h labels S_k such that $\psi(S_k) = \max_{\varphi_k \in \phi_k} \psi(s_k)$ and $\psi(S_k, S_l) = \max_{\varphi_k \in \phi_k, \varphi_l \in \phi_l} \psi(s_k, s_l)$. Using LBP on this MRF, we obtain the posterior for each representative transformation. We choose the best r representative transformations (unlike [13],

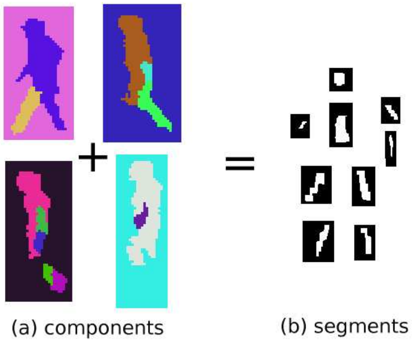

Figure 3: *Result of finding rigidly moving components between the four pairs of consecutive frames of the video shown in Fig. 1. Each component is shown in a different colour. For instance, for the top left image, one leg of the person moves differently from the rest of the body while the background remains static. The components are combined to get an initial estimate of the shape of the segments.*

which chooses only the best) with the highest posteriors for each site. These transformations are again divided into h representative transformations. Note that these h transformations are less coarse than the ones used previously. We repeat this process until we obtain the most likely transformation for each fragment $\mathbf{f}_k$. In our experiments, we use $h = 165$ and $r = 20$. LBP was found to converge within 20 iterations at each stage of the coarse to fine strategy.

Once the transformations for all the patches of frame j have been determined, we cluster the points moving rigidly together to obtain rigid components. Components with size less than 100 pixels are merged with surrounding components. We repeat this process for all pairs of consecutive frames of the video. The k^{th} component of frame j is represented as a set of points $\mathcal{C}_k^j$. Fig. 3 shows the result of our approach on four pairs of consecutive frames for the video shown in Fig. 1. Next, the rigid components need to be combined to get an initial estimate of the shape parameters of the segments p_i.

Combining rigid components: Given the set of all rigid components, we want to determine the number of segments p_i present in the scene and obtain an initial estimate of their shape $\mathbf{\Theta}_{Mi}$. To this end, we associate the components from one frame to the next using the transformations obtained above. This association is considered transitive, thereby establishing a correspondence of components throughout the video sequence.

Next, we cluster the components, based on appearance, using agglomerative clustering such that each cluster represents a segment of the scene. The similarity of two components is measured using normalized cross-correlation. Some components contain two or more segments, e.g. the leg component in the top left image of Fig. 3 contains two half limbs and the body component contains the head, torso and other half limbs. We rely on every segment of the scene being detected as an individual component in at least one frame. Empirically, this assumption is found to be true for a large class of scenes and camera motion. When clustering we simply let components containing more than one segment lie in a cluster representing one of these segments. For example, the body component in the top left image in Fig. 3 might lie in a cluster representing the torso while the leg component might belong to a cluster representing the upper half limb of that leg. However, the number of clusters would still be equal to the number of segments.

Once the clusters have been obtained, the smallest component of each cluster gives the shape $\mathbf{\Theta}_{Mi}$ of the segment p_i. This avoids using a component containing more than one segment to define the shape of a segment. However, this implies that the initial estimate will always be smaller than the ground truth and thus, needs to be expanded as described in § 3.2.

We need to account for the error introduced in the transformations when the patches are clustered to obtain the components. Thus, we measure the similarity of each component $\mathcal{C}_k^j$ in frame j with all the components of frame l that lie close to the component corresponding to $\mathcal{C}_k^j$ in frame l. The initial shape estimates of the segments, excluding the background, obtained in this manner are shown in the top row of Fig. 4. Note that all the segments of the person visible in the video have been obtained using our method.

Initial estimation of parameters: Once the mattes $\mathbf{\Theta}_{Mi}$ are found, we need to determine the initial estimate of the remaining parameters of the model. The transformation parameters $\mathbf{\Theta}_{Ti}^j$ are obtained using φ_k and the component clusters. The appearance parameter $\mathbf{\Theta}_{Ai}(\mathbf{x})$ is given by the mean of $\mathcal{I}_i^j(\mathbf{x})$ over all frames j. The lighting parameters $\mathbf{a}_i^j$ and $\mathbf{b}_i^j$ are calculated in a least squares manner using $\mathbf{\Theta}_{Ai}(\mathbf{x})$ and $\mathcal{I}_i^j(\mathbf{x})$, for all $\mathbf{x} \in p_i$. The motion parameters $\mathbf{m}_i^j$ are given by $\mathbf{\Theta}_{Ti}^j$ and $\mathbf{\Theta}_{Ti}^{j-1}$. This initial estimate of parameters is then refined by optimizing each parameter while keeping others unchanged. We start by optimizing the shape parameters $\mathbf{\Theta}_M$ as described in the next section.

3.2. Refining shape

In this section, we describe a method to refine the estimate of the shape parameters $\mathbf{\Theta}_M$ and determine the layer numbers l_i. Given an initial coarse estimate of the segments, we iteratively improve their shape using consistency of motion and texture over the entire video sequence. The refinement is carried out such that it minimizes the energy $\Psi(\mathbf{\Theta}|\mathbf{D})$ of the model.

The distribution of the RGB values obtained by projecting the segment into all frames is given by the histogram $\mathcal{H}_i$. This is required to compute the likelihood term in equation (2). The histograms $\mathcal{H}_i$ are obtained using the RGB values $\mathcal{I}_i^j(\mathbf{x})$. Given the mattes $\mathbf{\Theta}_{Mi}$ and the appearance parameters $\mathbf{\Theta}_{Ai}$, the energy of the model can be calculated using equation (2). Obviously, the optimum mattes $\mathbf{\Theta}_{Mi}^*$ are those which minimize $\Psi(\mathbf{\Theta}|\mathbf{D})$.

We take advantage of efficient algorithms for multi-way graph cuts which minimize an energy function over point labellings h of the form

$$\hat{\Psi} = \sum_{\mathbf{x} \in \mathbf{X}} D_{\mathbf{x}}(h_{\mathbf{x}}) + \sum_{\mathbf{x},\mathbf{y} \in \mathcal{N}} V_{\mathbf{x},\mathbf{y}}(h_{\mathbf{x}}, h_{\mathbf{y}}), \qquad (13)$$

under fairly broad constraints on D and V. Here $D_{\mathbf{x}}(h_{\mathbf{x}})$ is the cost for assigning the label $h_{\mathbf{x}}$ to point $\mathbf{x}$ and $V_{\mathbf{x},\mathbf{y}}(h_{\mathbf{x}}, h_{\mathbf{y}})$ is the cost for assigning labels $h_{\mathbf{x}}$ and $h_{\mathbf{y}}$ to the neighbouring points **x** and **y** respectively.

Specifically, we make use of two algorithms: $\alpha\beta$-swap and α-expansion [3]. The $\alpha\beta$-swap algorithm iterates over pairs of segments, p_α and p_β. At each iteration, it refines the mattes of p_α and p_β by swapping the values of $\mathbf{\Theta}_{M\alpha}(\mathbf{x})$ and $\mathbf{\Theta}_{M\beta}(\mathbf{x})$ for some points $\mathbf{x}$. The α-expansion algorithm iterates over segments p_α. At each iteration, it assigns $\mathbf{\Theta}_{M\alpha}(\mathbf{x}) = 1$ for some points $\mathbf{x}$. Note that α-expansion never reduces the number of points with label α.

In [7] we described an approach for refining the shape parameters of the LPS model where all the segments are restricted to lie in one *reference* frame. In that case, it was sufficient to refine one segment at a time using the α-expansion algorithm alone. Since in our layered representation this restriction no longer holds true, this method would lead to incorrect results as wrongly labelled points would never be relabelled. Hence, we extend our earlier approach using both α-expansion and $\alpha\beta$-swap algorithms.

We define the *limit* $\mathcal{L}_i$ of a segment p_i as the set of points **x** which lie within a distance of 25 from the current shape of p_i. Given segment p_i, let p_k be a segment such that the limit $\mathcal{L}_i$ of p_i overlaps with p_k in at least one frame j of the video. Such a segment p_k is said to be *surrounding* the segment p_i. The number of surrounding segments p_k is quite small for objects such as humans and animals which are restricted in motion. For example, the head segment of the person shown in Fig. 1 only overlaps with the torso segment and the background.

We iterate over segments and refine the shape of one segment p_i at a time. At each iteration, we perform an $\alpha\beta$-swap for p_i and each of its surrounding segments p_k. This relabels all the points which were wrongly labelled as belonging to p_i. We then perform an α-expansion algorithm to expand p_i to include those points $\mathbf{x}$ in its limit which move rigidly with p_i. During the iteration refining p_i, we consider three possibilities for p_i and its surrounding segment p_k: $l_i = l_k$, $l_i > l_k$ or $l_i < l_k$. If $l_i < l_k$, we assign $\Pr(\mathcal{I}_i^j(\mathbf{x})|\mathbf{x} \in p_i) = \texttt{const}$ for frames j where **x** is occluded by a point in p_k. We choose the option which results in the minimum value of $\Psi(\mathbf{\Theta}|\mathbf{D})$. We stop iterating when further reduction of $\Psi(\mathbf{\Theta}|\mathbf{D})$ is not possible. This provides us with a refined estimate of $\mathbf{\Theta}_M$ along with the layer number l_i of the segments.

Fig. 4 shows the result of refining the shape parameters of the segments by the above method using the initial estimates. Note that even though the torso is partially occluded by the arm and the backleg is partially occluded by the front leg in every frame, their complete shape has been learnt using overlapping binary mattes. Next, the appearance parameters corresponding to the refined shape parameters are obtained.

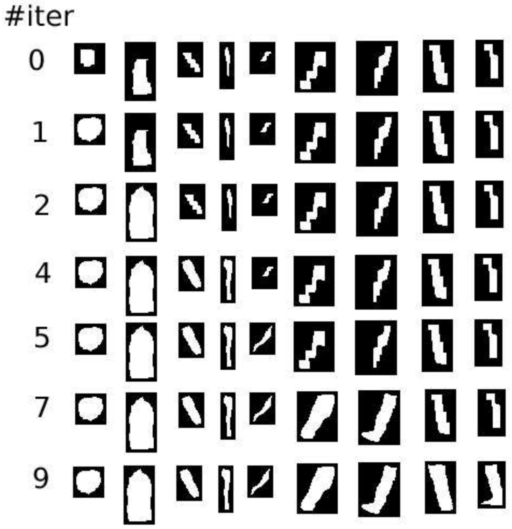

Figure 4: *The refined mattes of the layered representation of a person using multi-way graph cuts. The shape of the head is re-estimated after one iteration. The next iteration refines the torso segment. Subsequent iterations refine the half limbs one at a time. Note that the size of the mattes is equal to that of a frame of the video but smaller mattes are shown here for clarity.*

3.3. Updating appearance

Once the mattes $\mathbf{\Theta}_{Mi}$ of the segments are obtained, the appearance of a point $\mathbf{x} \in p_i$, i.e. $\mathbf{\Theta}_{Ai}(\mathbf{x})$ is calculated as the mean of $\mathcal{I}_i^j(\mathbf{x})$ over all frames j. The refined shape and appearance parameters help in obtaining a better estimate for the transformations as described in the next section.

3.4. Refining the transformations

Finally, the transformation parameters $\mathbf{\Theta}_T$ and the lighting parameters $\mathbf{\Theta}_L$ are refined by searching over putative transformations around the initial estimate, for all segments at each frame j. For each putative transformation, parameters $\{\mathbf{a}_i^j, \mathbf{b}_i^j\}$ are calculated in a least squares manner. The parameters which result in the smallest SSD are chosen. When refining the transformation, we searched for putative transformations by considering translations of upto 5 pixels, scales between 0.9 and 1.1 and rotations between -0.15 and 0.15 radians around the initial estimate. In the next section, we demonstrate the application of the learnt model for segmentation.

4. Results

We now present results for motion segmentation using the learnt layered representation of the scene. The method is applied to different types of object classes (such as jeep, humans and cows), foreground motion (pure translation, piecewise similarity transforms) and camera motion (static and panning) with static backgrounds. We use the same parameter values in all our experiments.

Our assumption that segments are always mapped using only simple geometric transformations is not always true. This would result in gaps between segments in the generated frame. In order to deal with this, we relabel points around the boundary of segments. This relabelling is performed by using the α-expansion algorithm. The cost $D_{\mathbf{x}}(h_x)$ of assigning point **x** around the boundary of p_i to p_i is the inverse log likelihood of its observed RGB values in that frame given by the histogram $\mathcal{H}_i$. The cost $V_{\mathbf{x},\mathbf{y}}(h_x, h_y)$ of assigning two different labels h_x and h_y to neighbouring points **x** and **y** is directly proportional to $\mathcal{B}_i(\mathbf{x}, \mathbf{y})$ for that frame.

Fig. 5 shows the segmentations obtained by generating frames using the learnt representation by projecting all segments other than those belonging to layer 0. Fig. 5(a) and 5(b) show the result of our approach on simple scenarios where each layer of the scene consists of segments which are undergoing pure translation. Despite having a lot of flexibility in the putative transformations by allowing for various rotations and scales, the initial estimation recovers the correct transformations, i.e. those containing only translation. Note that the transparent windshield of the jeep is (correctly) not recovered in the M.A.S.H. sequence as the background layer can be seen through it. For the sequence shown in Fig. 5(b) the method proves robust to changes in lighting condition. Not surprisingly, it learns the correct layering for the segments corresponding to the two people.

Fig. 5(c) and 5(d) show the motion segmentation obtained for two videos, each of a person walking. In both cases, the body is divided into the correct number of segments (head, torso and seven visible half limbs). Our method recovers well from occlusion in these cases. For such videos, the feet of a person are problematic as they tend to move non-rigidly with the leg in some frames. Note that the grass in Fig. 5(d) has similar intensity to the person's trousers. Thus, recovering the correct transformation of the legs is difficult.

Fig. 5(e) and 5(f) are the segmentations of a cow walking. Again, the body of the cow is divided into the correct number of segments (head, torso and eight half limbs). The cow in Fig. 5(e) undergoes a slight out of plane rotation in some frames, which causes some bits of grass to be pulled into the segmentation. The video shown in Fig. 5(f) is taken from a poor quality analog camera. However, our algorithm proves robust enough to obtain the correct segmentation. Note that when relabelling the points around the boundary of segments some parts of the background, which are similar in appearance to the cow, get included in the segmentation.

Timing: The initial estimation takes approximately 5 minutes for every pair of frames: 3 minutes for computing the likelihood of the transformations and 2 minutes for MAP estimation using LBP. The shape parameters of the segments are refined by minimizing the energy $\Psi(\mathbf{\Theta}|\mathbf{D})$ as described

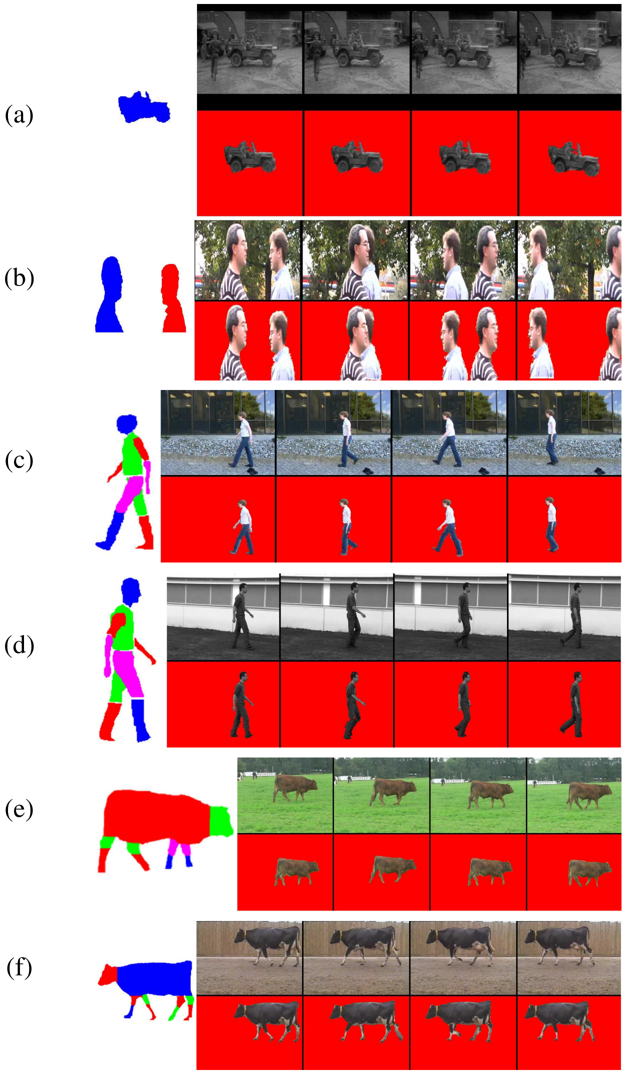

Figure 5: **Motion segmentation results.** *In each case, the left image shows the various segments obtained in different colours. The top row shows the original video sequence while the segmentation results are shown in the bottom row. (a): A 10 frame video sequence taken from 'M.A.S.H.'. The video contains a jeep undergoing translation against a static background while the camera pans to track the jeep. (b): A 40 frame sequence taken from a still camera (courtesy Nebojsa Jojic [6]). The scene contains two people undergoing pure translation in front of a static background. The results show that the layering is learnt correctly. (c): A 40 frame sequence taken from a still camera (courtesy Hedvig Sidenbladh [11]). The scene consists of a person walking against a static background. The correct layering of various segments of the person is learnt. (d): A 57 frame sequence taken from a panning camera of a person walking against a static background (courtesy Ankur Agarwal [1]). Again, the correct layering of the segments is learnt. (e): A 44 frame sequence of a cow walking taken from a panning camera. All the segments, along with their layering, are learnt. (f): A 30 frame sequence of a cow walking against a static (homogeneous) background (courtesy Derek Magee [8]). The video is taken from a still analog camera which introduces a lot of noise.*

in § 3.2. The graph cut algorithms used have, in practice, a time complexity which is linear in the number of points in the binary matte $\mathbf{\Theta}_{Mi}$. It takes less than 1 minute to refine the shape of each segment. Most of the time is taken up in calculating the various terms which define the energy $\Psi(\mathbf{\Theta}|\mathbf{D})$ as shown in equation (2). The algorithm converged after at most 2 iterations through each segment. All timings provided are for a C++ implementation on a 2.4 GHz processor.

Ground truth comparison: The segmentation performance of our method was assessed using eight manually segmented frames (four each from the challenging sequences shown in Fig. 5(c) and 5(f)). Out of 80901 ground truth foreground pixels and 603131 ground truth background pixels in these frames, 79198 (97.89%) and 595054 (98.66%) were present in the generated frames respectively. Most errors were due to the assumption of piecewise parametric motion and due to similar foreground and background pixels.

Sensitivity of parameters: When determining rigidity of two transformations or clustering fragment to obtain components, we allow for the translations to vary by one pixel in x and y directions to account for errors introduced by discretization of putative transformations. Fig. 6(a) shows the effects of not allowing for slight variations in the translations. As expected, it oversegments the body of the person. However, allowing for more variation does not undersegment as different components move quite non-rigidly for a large class of scenes and camera motion. Fig. 6(b) and (c) shows the effects of setting λ_1 and λ_2 to zero, thereby not encouraging spatial continuity.

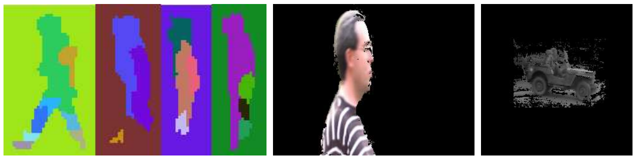
(a) (b) (c)

Figure 6: *(a) Result of finding rigidly moving components between four consecutive frames from the video shown in Fig. 1 without allowing for slight variation in translations (see text). (b)-(c) The appearance and shape of segments learnt without encouraging spatial continuity. While (b) indicates that the method works well for simple cases where the foreground and background differ significantly (e.g. see Fig. 5(b)), the result in (c) shows that the segmentation starts to include parts of the background if it is homogeneous (e.g. see Fig. 5(a)).*

5. Summary and Conclusions

The algorithm proposed in this paper achieves extremely good motion segmentation results. Why is this? We believe that the reasons are two fold. Incremental improvements in the Computer Vision field have now ensured that: (i) We can use an appropriate generative model which accounts for motion, changes in appearance, layering and spatial continuity. The model is not too strong so as to undersegment, and not too weak so as to oversegment; (ii) We have more powerful modern algorithmic methods such as LBP and graph cuts which avoid local minima better than previous approaches.

However, there is still more to do. As is standard in methods using layered representation, we have assumed that the visual aspects of the objects do not change throughout the video sequence. At the very least we need to extend the model to handle the varying visual aspects objects present in the scene, e.g. front, back and 3/4 views, in addition to the side views. The restriction of rigid motion within a segment can be relaxed using non-parametric motion models.

Acknowledgments. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

[1] A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In *ECCV*, pages III:54–65, 2004.

[2] M. Black and D. Fleet. Probabilistic detection and tracking of motion discontinuities. *IJCV*, 38:231–245, 2000.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.

[4] D. Cremers and S. Soatto. Variational space-time motion segmentation. In *ICCV*, pages II:886–892, 2003.

[5] P. Felzenszwalb and D. Huttenlocher. Fast algorithms for large state space HMMs with applications to web usage analysis. In *NIPS*, 2003.

[6] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *CVPR*, volume 1, pages 199–206, 2001.

[7] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered pictorial structures from video. In *ICVGIP*, pages 148–153, 2004.

[8] D. Magee and R. Boyle. Detecting lameness using re-sampling condensation and multi-stream cyclic hidden markov models. *IVC*, 20(8):581–594, June 2002.

[9] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kauffman, 1998.

[10] D. Ramanan and D. Forsyth. Using temporal coherence to build models of animals. In *ICCV*, pages 338–345, 2003.

[11] H. Sidenbladh and M. Black. Learning the statistics of people in images and video. *IJCV*, 54(1):181–207, September 2003.

[12] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated bayesian approach to layer extraction from image sequences. *IEEE PAMI*, 23(3):297–304, 2001.

[13] G. Vogiatzis, P. H. S. Torr, S. Seitz, and R. Cipolla. Reconstructing relief surfaces. In *BMVC*, pages 117–126, 2004.

[14] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. on IP*, 3(5):625–638, 1994.

[15] Y. Weiss. Belief propagation and revision in networks with loops. Technical Report AIM-1616, MIT, 1997.

[16] Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation. In *CVPR*, pages 321–326, 1996.

[17] C. Williams and M. Titsias. Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Computation*, 16(5):1039–1062, 2004.

[18] J. Wills, S. Agarwal, and S. Belongie. What went where. In *CVPR*, pages I:37–44, 2003.

[19] J. Winn and A. Blake. Generative affine localisation and tracking. In *NIPS*, 2004.