Learning Layered Pictorial Structures from Video

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

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

Abstract

We propose a new unsupervised learning method to obtain a layered pictorial structure (LPS) *representation of an articulated object from video sequences. It will be seen that this is related in turn to methods for learning* sprite based *representations of an image. The method we describe involves a new generative model for performing segmentation on a set of images. Included in this model are the effects of motion blur and occlusion. An initial estimate of the parameters of the model is obtained by dividing the scene into rigidly moving components. The estimate of the matte of each part is refined using a variation of the α-expansion graph cut algorithm. This method has the advantage of achieving a strong local minimum over labels. Results are demonstrated on animals for which an articulated* LPS *representation is naturally suited.*

1. Introduction

In order to manage the variability in appearance of objects, there is a broad agreement that object categories should be represented by a collection of spatially related parts each with its own appearance. This sort of approach dates back to the *pictorial structures* (PS) model of Fischler and Elschlager, introduced three decades ago [4]. Pictorial structures (and the related constellation of parts model) have met with great success in object recognition [1, 3, 9, 11], and so the question naturally arises, how might we learn this representation automatically?

Weber *et al.* [11] and Fergus *et al.* [3] both consider the problem of learning PS in unsegmented images e.g. given a large number of images that contain horses, but not a segmentation, the task is to learn a representation of a horse. The algorithms they propose require very large amounts of training data and are extremely slow which prompts the question might not video input data be better used for the task of learning representations? For example, Fig. 1 shows some frames of one such video which can be used to learn the representation of a zebra.

Ramanan *et al.* [9] present a method for learning pictorial structures from a video sequence by clustering segments obtained by searching for parallel lines. However, their model does not describe the object completely. Furthermore, the relative depth of the various parts of the model are also not determined. We propose a generative approach to estimating pictorial structures from video taking inspiration from the related *sprite* based approaches.

Figure 1. *Four intermediate frames of a 25 frame video sequence of a zebra running as the camera pans to follow it. Given the sequence, the generative model which best describes the zebra and the background is learnt in an unsupervised manner.*

Jojic and Frey [5] provide a generative Bayesian framework in which each image is explained as a layered composition of *sprites* moving under pure translation. A sprite is a 2D appearance map and matte (mask) of an object, learned using a variational algorithm. Each sprite is assigned to one of L depth layers which determine the occlusion ordering when composing the image. William and Titsias [12] use a greedy sequential approach whereby the model for each sprite is extracted in turn. However, using a greedy method restricts them from obtaining models which are optimized over all sprites. Both these methods do not consider the spatial continuity in labelling, which means the methods are unlikely to work unless the image sequences are very long and highly textured. Moreover, they do they take self-occlusion or changes in appearance due to lighting into account.

We present a new model which addresses the deficiencies in [5, 12], the *layered pictorial structure*, LPS, which generalizes and improves both the layered sprite models and PS. Unlike previous models, each part is also assigned a layer number which represents its relative depth. In con-

trast to the greedy method described in [12], all parts are learnt simultaneously.

We present a method to estimate the parameters of the model in an unsupervised manner. Using rough estimates of motion obtained by a motion segmentation algorithm [7], the shape parameters of the model, represented as a binary matte, are learnt by minimizing an objective function using the α-expansion algorithm to perform multi-way graph cuts. The parts obtained by this method describe the object completely. Unlike the greedy method proposed in [12], α-expansion method optimizes over all parts and guarantees that a strong local minimum (i.e. bounded by a constant factor of the global minimum) is found. Since our method works by refining parts instead of dealing with individual pixels in the scene, we can explicitly model the change in appearance of parts arising from lighting conditions and motion blur.

In the next section, we describe the LPS model in detail. In section 3, we present a four stage algorithm to estimate the shape, appearance, layer number and transformation parameters for all parts of the LPS. The learnt LPS model can be used for several applications such as recognition, pose estimation and point-and-click object removal. In section 4, we demonstrate the application of the model for segmentation.

2. Layered pictorial structures

This section describes the LPS model and its terminology. PS are compositions of *parts*, which are 2D patterns with a probabilistic model for their shape and appearance. In the LPS model, in addition to shape and appearance, each part is also assigned a layer number.

LPS is a generative model, i.e. any instance of the object and background can be generated from it by assigning appropriate values to its parameters as shown in Fig. 2. It also provides the likelihood of that instance.

A model reference frame describes the shape and appearance of the parts (top image in Fig. 2). The shape of a part p_i is represented by a binary matte $\mathbf{\Theta}^i_M$, such that

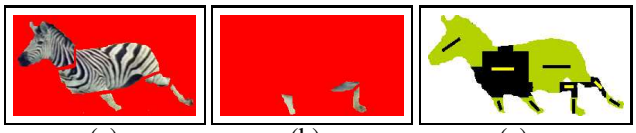
$$\begin{aligned} \mathbf{\Theta}^i_M(\mathbf{x}) &= 1, \ \texttt{if } \mathbf{x} \in p_i, \\ &= 0, \ \texttt{otherwise}. \end{aligned} \tag{1}$$

The appearance $\mathbf{\Theta}^i_A(\mathbf{x})$ is the RGB value of point $\mathbf{x}$ in the model reference frame. Instances of the object (e.g. frames of a video) along with their likelihoods are generated by applying a transformation to each part. The transformations $\mathbf{\Theta}^j_{Ti}$ generate frame j by mapping each point $\mathbf{x} \in p_i$ of the model reference frame onto point $\mathbf{x}' = \mathbf{\Theta}^j_{Ti}(\mathbf{x})$ in the frame as shown in Fig. 2. Each transformation is defined by a translation $\{x, y\}$, rotation ϕ and scales s_x and s_y in x and y direction respectively.

Parts are composited in descending order of their layer numbers to handle self-occlusion. The layer number l_i determines the relative depth of a part with respect to other parts. Several parts can have the same layer number (see Fig. 3). The part p_i can partially or completely occlude part p_j if and only if $l_i > l_j$.

$\mathbf{D}$	Data (RGB values of all pixels in every frame of a video).
n_F	Number of frames.
n_P	Number of parts p_i including the background.
l_i	Layer number of part p_i.
$\mathbf{\Theta}^i_M$	Matte for part p_i.
$\mathbf{c}(\mathbf{x})$	RGB value [r($\mathbf{x}$) g($\mathbf{x}$) b($\mathbf{x}$)] for point $\mathbf{x}$.
$\mathbf{\Theta}^i_A$	Appearance parameter for p_i i.e. $\mathbf{c}(\mathbf{x}) \ \forall \ \mathbf{x} \in$p$_i$.
$\mathbf{\Theta}^i_P$	Parameters $\{l_i, \mathbf{\Theta}^i_M, \mathbf{\Theta}^i_A\}$ of part p_i.
$\mathbf{\Theta}^j_{Ti}$	Transformation $\{x, y, s_x, s_y, \phi\}$ of part p_i in frame j.
$\mathbf{\Theta}^j_{Di}$	Lighting and motion parameters $\{\mathbf{a}^j_i, \mathbf{b}^j_i, \mathbf{m}^j_i\}$ of part p_i in frame j.
$\mathbf{\Theta}$	Model parameters $\{n_P, \mathbf{\Theta}_P; \mathbf{\Theta}_T, \mathbf{\Theta}_D\}$.

Table 1. *Parameters of the* LPS



(a) (b) (c)

Figure 3. *An articulated object is divided into multiple layers in the* LPS *model. Several parts can belong to the same layer. Parts of the model for a zebra belonging to layer 2 and layer 1 are shown in (a) and (b) respectively. Mattes of various parts are shown in (c). The lines indicate the relative position and orientation of parts.*

The effects of lighting conditions and *motion blur* on appearance are explicitly modelled using parameter $\mathbf{\Theta}^j_{Di} = \{\mathbf{a}^j_i, \mathbf{b}^j_i, \mathbf{m}^j_i\}$. The change in appearance of the part p_i in frame j due to lighting conditions is accounted for by an affine transform of the RGB values:

$$\mathbf{c}(\mathbf{x}') = \texttt{diag}(\mathbf{a}^j_i) \cdot \mathbf{c}(\mathbf{x}) + \mathbf{b}^j_i. \tag{2}$$

The parameter $\mathbf{m}^j_i$ is the time varying component of the motion which depends on the location of part p_i in the previous frame. Unlike previous approaches, this allows us to take into account the change in appearance due to motion blur as

$$\mathbf{c}(\mathbf{x}') = \int_0^T \mathbf{c}(\mathbf{x}' - \mathbf{m}^j_i(t))dt, \tag{3}$$

where T is the total exposure time when capturing the frame. The notation used to develop the model is summarized in table 1.

Model energy. Given data $\mathbf{D}$, which consists of the RGB values of all pixels in every frame of a video, let $\mathcal{I}^j_i(\mathbf{x})$ be the observed RGB values of point $\mathbf{x}' = \mathbf{\Theta}^j_{Ti}(\mathbf{x})$ in frame j. The likelihood of the model has two components: (i) appearance and (ii) boundary. The appearance component is a measure of the consistency of the observed RGB values $\mathcal{I}^j_i(\mathbf{x})$ with the generated RGB values $\mathbf{c}(\mathbf{x}')$ over the entire video sequence. The boundary component gives preference to an edge being present between two neighbouring points

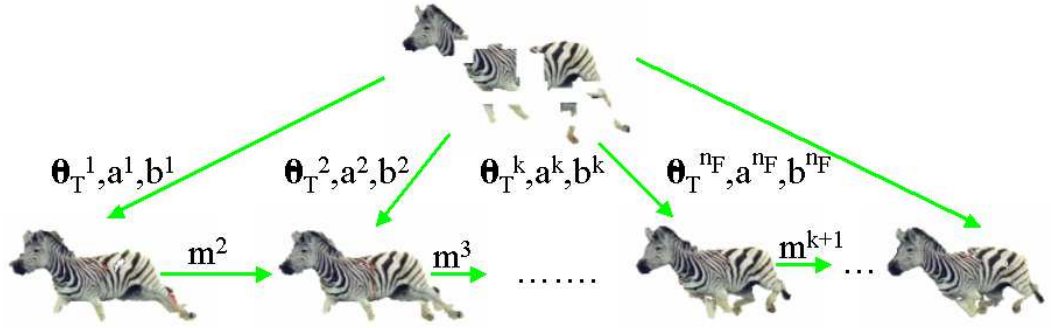


Figure 2. *The top row shows the model reference frame of the* LPS *model for a zebra. Any frame j can be generated using the* LPS *model by assigning appropriate values to the parameters. When generating the frame k, the motion parameters m_i^k are obtained using the transformation of the parts p_i in frame $k-1$. The background part for the generated frames is not shown.*

$\mathbf{x}$ and $\mathbf{y}$ which belong to different parts. This is particularly true if one point belongs to the background and the other belongs to some part of the object. Thus, we can define the energy of the model as:

$$\Psi(\mathbf{D}) = \sum_{i=1}^{n_P} \sum_{\mathbf{x} \in p_i} \left(\mathcal{A}(\mathbf{x}, p_i) + \lambda \sum_{\mathbf{y}} \mathcal{B}(\mathbf{x}, \mathbf{y}) \right), \quad (4)$$

where $\mathbf{x}$ and $\mathbf{y}$ are two neighbouring points which belong to different parts.

Appearance. For a given frame k, the appearance component for a point $\mathbf{x} \in p_i$, i.e.

$$\mathcal{A}_k(\mathbf{x}, p_i) = -\log(\Pr(\mathcal{I}_i^k(\mathbf{x}) | \mathbf{x} \in p_i)) \quad (5)$$

is the inverse log-likelihood of the observed data being generated by the parameters of p_i. It is assumed each part has a distinctive appearance, such that the intensities conform to some given distribution, which models the appearance of the texture of that part. For this paper we assume that the set of RGB values of the pixels for a given part p_i over all frames follows a Gaussian Mixture Model (GMM) $\mathcal{M}_i$.

If a point $\mathbf{x}$ belongs to part p_i, the error in generating that point using the parameters for p_i is small. Thus the likelihood of $\mathbf{x}$ is

$$\Pr(\mathcal{I}_i^k(\mathbf{x}) | \mathbf{x} \in p_i) \propto \mathcal{M}_i(\mathcal{I}_i^k(\mathbf{x})) \exp(-(\mathbf{c}(\mathbf{x}') - \mathcal{I}_i^k(\mathbf{x}))^2), \quad (6)$$

where $\mathbf{c}(\mathbf{x}')$ is the RGB value generated for point $\mathbf{x} \in p_i$ in frame k using the model reference frame as shown in equation (3). In homogeneous regions, motion vectors yield little discrimination between foreground and background. In such case, the texture model given by the GMMs provides better discrimination. For example, motion may not be determined locally for a uniformly brown horse. However, its appearance ('brown') distinguishes it from the background ('green grass'). Therefore, it is necessary to estimate the likelihood of the RGB values $\mathcal{I}_i^k(\mathbf{x})$ given by the GMM of part p_i.

For a video, the appearance component for $\mathbf{x} \in p_i$ is given by summing over all frames:

$$\mathcal{A}(\mathbf{x}, p_i) = \sum_{k=1}^{n_F} \mathcal{A}_k(\mathbf{x}, p_i). \quad (7)$$

The appearance component for the part is obtained by summing over $\mathbf{x} \in p_i$ as shown in equation (4).

Boundary. The boundary component is given by

$$\mathcal{B}(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-g_i^2(\mathbf{x}, \mathbf{y})}{2\sigma^2} \right) \cdot \frac{1}{\mathtt{dist}(\mathbf{x}, \mathbf{y})}, \quad (8)$$

where $\mathbf{x} \in p_i$ and $\mathbf{y} \notin p_i$ are two neighbouring points which belong to different parts. The neighbourhood of a point $\mathbf{x}$ is defined as its 8-neighbourhood. The total cost is obtained by summing along the boundary of all parts as shown in equation (4). For a given frame k, $g_i(\mathbf{x}, \mathbf{y})$ measures the difference in the RGB values of points $\mathcal{I}_i^k(\mathbf{x})$ and $\mathcal{I}_i^k(\mathbf{y})$ and $\mathtt{dist}(\mathbf{x}, \mathbf{y})$ gives more weightage to the 4-neighbourhood of $\mathbf{x}$. For a video, we define $g_i(\mathbf{x}, \mathbf{y})$ as the average difference in the RGB values $\mathcal{I}_i^k(\mathbf{x})$ and $\mathcal{I}_i^k(\mathbf{y})$ over all frames k, i.e.

$$g_i(\mathbf{x}, \mathbf{y}) = \frac{1}{n_F} \sum_{k=1}^{n_F} |\mathcal{I}_i^k(\mathbf{x}) - \mathcal{I}_i^k(\mathbf{y})|. \quad (9)$$

The term λ in equation (4) specifies the relative weight of the boundary part to the appearance part in the likelihood of the model. Using a high value for λ results in discontinuous parts for textured animals such as zebras as it encourages edges between points belonging to different parts. A lower value for λ would result in background being included in the parts belonging to the object. The value of σ in equation (8) determines how the energy $\Psi(\mathbf{D})$ is penalized since the penalty is high when $g_i(\mathbf{x}, \mathbf{y}) < \sigma$ and small when $g_i(\mathbf{x}, \mathbf{y}) > \sigma$.

In the next section, we describe a four stage approach to calculate the parameters $\mathbf{\Theta}$ of the LPS model of an object, given data $\mathbf{D}$, by minimizing the energy $\Psi(\mathbf{D})$. The method described is applicable to any articulated object category. We demonstrate the method for animals.

3. Learning layered pictorial structures

Given the video of an animal in motion, our objective is to estimate the parameters $\mathbf{\Theta}$ of the model for the animal and the background. 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. Following this method guarantees finding a strong local minimum.

1. An initial estimate of the parameters $\mathbf{\Theta}$ is obtained by dividing the scene into rigidly moving components (§ 3.1).
2. The parameters $\mathbf{\Theta}_T$,$\mathbf{\Theta}_A$ and $\mathbf{\Theta}_D$ are kept constant and $\mathbf{\Theta}_M$ is optimized using the α-expansion algorithm. The layer numbers l_i are also 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 parameters $\mathbf{\Theta}_T$ and $\mathbf{\Theta}_D$ are re-estimated , keeping $\mathbf{\Theta}_M$ and $\mathbf{\Theta}_A$ unchanged (§ 3.4).

Table 2: *Estimating the parameters of the* LPS *model*

3.1. Initial estimation of parameters

We obtain an initial estimate of parameters $\mathbf{\Theta} = \{n_P, \mathbf{\Theta}_P; \mathbf{\Theta}_T, \mathbf{\Theta}_D\}$ (excluding the layer numbers l_i) using the method described in [7]. Briefly, this approach consists of choosing one frame (e.g. the first) as a reference frame. The reference frame is divided into uniform rectangular fragments which are tracked throughout the video sequence. Tracking for a frame is performed by defining a Markov random field (MRF) such that the sites of the MRF correspond to the fragments in the reference frame. The neighbourhood of a fragment is defined as its 4-neighbourhood. Each putative transformation of a fragment is represented by a label on the corresponding node of the MRF. Its likelihood is proportional to the cross-correlation of the fragment, after undergoing the transformation, with the frame. Pairwise potentials are defined such that neighbouring fragments which do not move rigidly are penalized. MAP estimation over all fragments is obtained using loopy belief propagation [8]. Fig. 4 shows the reference frame and tracking results for the frames shown in Fig. 1.

The initial estimate of the parts of the model is obtained by clustering rigidly moving points. Fig. 5 (top) shows the initial mattes $\mathbf{\Theta}_M$ of the parts obtained. The RGB values of the corresponding points in the reference frame provide the appearance parameter $\mathbf{\Theta}_A$ and the motion parameters estimated above provide the initial estimate of $\mathbf{\Theta}_T$ and $\mathbf{m}_i^j$. The lighting parameters $\mathbf{a}_i^j$ and $\mathbf{b}_i^j$ are then computed in a least squares manner.

The initial estimate is then refined by optimizing one parameter at a time while keeping others unchanged. We start by optimizing the shape parameters $\mathbf{\Theta}_M$ as described in the next section.

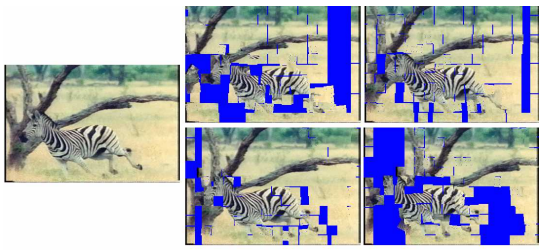


Figure 4. *Reference frame (left) and reconstructed frames corresponding to those in Fig. 1. Each fragment from the reference frame is mapped using the most likely transformation obtained. The pixels where no points from the reference frame are mapped are coloured in blue. Note that despite the errors due to discretization of the reference frame into fragments, the method provides roughly correct motion parameters.*

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 parts, 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 $\Psi(\mathbf{D})$ of the model.

The distribution of the RGB values obtained by projecting the part into all frames is given by the GMM $\mathcal{M}_i$. This is required to compute the appearance component in equation (4). The parameters of $\mathcal{M}_i$ are obtained using the EM algorithm. Given the mattes $\mathbf{\Theta}_M^i$, with the corresponding points in the reference frame defining the appearance parameters $\mathbf{\Theta}_A^i$, the energy of the model can be calculated using equation (4). Obviously, the optimum mattes $\mathbf{\Theta}_M^{i*}$ are those which minimize $\Psi(\mathbf{D})$.

We take advantage of the efficient α-expansion algorithm [6] for solving MRFs which is based on graph cuts. Specifically, it is possible to efficiently minimize an energy function over point labellings h of the form

$$\hat{\Psi} = \sum_{\mathbf{x}\in\mathbf{X}} D_{\mathbf{x}}(h_{\mathbf{x}}) + \sum_{\mathbf{x},\mathbf{y}\in\mathcal{N}} V_{\mathbf{x},\mathbf{y}}(h_{\mathbf{x}}, h_{\mathbf{y}}), \qquad (10)$$

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 $\mathbf{x}$ and $\mathbf{y}$ respectively.

In our case, each label h_i assigns a point $\mathbf{x}$ in the model reference frame to part p_i. Comparing the model energy $\Psi(\mathbf{D})$ in equation (4) with the energy function $\hat{\Psi}$ we get

$$D_{\mathbf{x}}(h_i) = \mathcal{A}(\mathbf{x}, p_i), \qquad (11)$$

and

$$V_{\mathbf{x},\mathbf{y}}(h_i, h_j) = \lambda\mathcal{B}(\mathbf{x}, \mathbf{y}), \qquad (12)$$

where $h_j \neq h_i$. For the sake of completeness, we define graph cuts and the α-expansion algorithm below.

A *cut* on a graph partitions its vertices into two disjoint sets. The cost of a cut is defined as the sum of the weights of the edges between vertices belonging to different sets. The problem of minimizing $\hat{\Psi}$ for two labels can be formulated as the problem of finding the cut with the minimum cost, which in turn can be efficiently solved by computing the maximum flow of the graph [2].

Unfortunately, minimizing $\hat{\Psi}$ for more than two labels is an NP-hard problem [6]. However, an iterative algorithm called α-expansion can be used to find a strong local minimum which lies within a constant factor of the global minimum. This constant factor is at least 2, and depends on V [2]. We define the *limit* $\mathcal{L}_i$ of a part p_i as the set of points $\mathbf{x}$ which lie within a distance of 25 from the bounding box of the current shape of p_i. Each iteration of the α-expansion algorithm refines a part p_i by solving the problem of assigning two labels h_i and $\neg h_i$ to points $\mathbf{x} \in \mathcal{L}_i$, where $\neg h_i$ is the union of all labels excluding h_i. In other words, at each iteration we do not allow a part to expand beyond its limit $\mathcal{L}_i$. For example, during an iteration where the head part is being refined, each point $\mathbf{x}$ can either retain its label, i.e. remain a point on the torso or leg parts, or get assigned the label corresponding to the head part.

Given part p_i, let p_j be a part such that the limit $\mathcal{L}_i$ of p_i overlaps with p_j in at least one frame k of the video. The number of such parts p_j is quite small for objects such as animals which are restricted in motion. For example, the head part of the zebra shown in Fig. 1 only overlaps with the torso part. During the iteration refining p_i, we consider three possibilities for p_i and p_j: $l_i = l_j$, $l_i > l_j$ or $l_i < l_j$. If $l_i < l_j$, we assign $\Pr(\mathcal{I}_i^k(\mathbf{x}) | \mathbf{x} \in p_i) = \texttt{const}$ for frames k where $\mathbf{x}$ is occluded by a point in p_j. We choose the option which results in the minimum value of Ψ. After iterating through each part once, the parameters of the GMM $\mathcal{M}_i$ are updated. We stop iterating when further reduction of Ψ is not possible. This provides us with a refined estimate of $\mathbf{\Theta}_M$ along with the layer number l_i of the parts.

Fig. 5 shows the result of refining the shape parameters of the parts by the above method using the initial estimates. Next, the appearance parameters corresponding to the refined shape parameters are obtained.

3.3. Refining appearance

Once the mattes $\mathbf{\Theta}_M^i$ of the parts are obtained, the corresponding pixels in the reference frame provide the final estimate of the appearance parameter $\mathbf{\Theta}_A^i$. The refined shape and appearance parameters help in obtaining a better estimate for the transformations as described in the next section.

3.4. Refining transformations

Finally, the transformation parameters $\mathbf{\Theta}_T$ and the lighting and motion blur parameters $\mathbf{\Theta}_D$ are refined by searching over transformations $\{x, y, s_x, s_y, \phi\}$ around the initial estimate, for all parts at each frame j. For each putative transformation, parameters $\{\mathbf{a}_i^j, \mathbf{b}_i^j\}$ are calculated in a least squares manner and the motion parameters $\mathbf{m}_j^i$ are found using the current transformation and $\mathbf{\Theta}_{Ti}^{j-1}$. The parameters which result in the smallest SSD are chosen. In the next section, we demonstrate the application of the learnt model for segmentation.

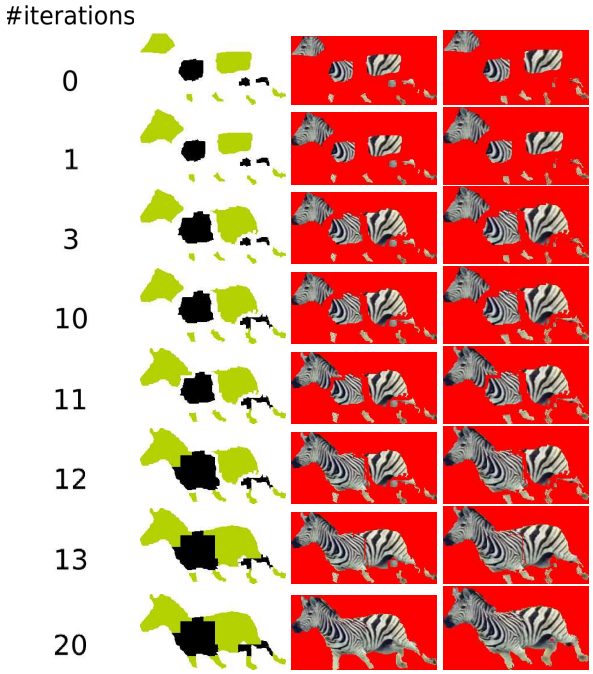


Figure 5. *The refined mattes of the* LPS *parts for a zebra obtained using multi-way graph cuts. The figure also shows two of the generated frames. The shape of the head is re-estimated after one iteration. The next two iterations refine the torso parts which expand out while the half-limbs remain unchanged. All parts (including background) are refined once after 10 iterations. At this point, the parameters of the* GMM*, $\mathcal{M}_i$, are re-estimated. The final estimates of the head and the two body parts are obtained after 11, 12 and 13 iterations respectively. The final matte of the parts after 20 iterations followed by merging smaller regions with their neighbours is shown in the last row. Note that even with a bad initial estimate, the α-expansion method results in a good local minimum.*

4. Results, Segmentation

We present an application of the learnt LPS model for segmentation. The parts along with their layer number and their transformations for each frame, are obtained by minimizing the energy $\Psi(\mathbf{D})$. In our experiments, λ and σ are assigned the values 1 and 5 respectively.

In practice, the α-expansion algorithm described in § 3.2 converged after 2 expansion moves for each part, resulting in a total of 20 expansion moves, for all the video sequences we used. When refining parameter $\mathbf{\Theta}_T$ as described in section 3.4, we searched for putative transformations by considering translations of upto 5 pixels, scales between 0.8 and 1.2 and rotations between -0.3 and 0.3 radians around the initial estimate.

To obtain segmentation of the video, the parameters $\mathbf{\Theta}$ are used to generate the frames as described in section 2. Our assumption that parts are always mapped onto the frame being generated using only simple geometric transformations $\{x, y, s_x, s_y, \phi\}$ is not always true. This would result in gaps between parts in the generated frame. In order to deal with this, we allow for slight parallax distortion [10] for each part by relabelling points around the boundary of parts. This relabelling is performed by using the α-expansion algorithm. The cost $D_{\mathbf{x}}(h_i)$ of assigning point $\mathbf{x}$ 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 GMM $\mathcal{M}_i$. The cost $V_{\mathbf{x},\mathbf{y}}(h_i, h_j)$ of assigning two different labels h_i and h_j to neighbouring points $\mathbf{x}$ and $\mathbf{y}$ is directly proportional to $\mathcal{B}(\mathbf{x}, \mathbf{y})$ for that frame.

Fig. 6 and 7 show the results of segmentation on zebra and rhinoceros videos which have 25 frames and 100 frames respectively. The third row in each figure shows the difference between the actual frame and the one generated using the LPS model. Most of the error in the zebra video is due to the misalignment of texture in the head and body parts. The performance of our method was measured using two manually segmented intermediate frames corresponding to those shown in Fig. 5. Out of 27268 ground truth foreground pixels in these frames, 26224 (96.17%) were present in the generated frames. The errors in the rhinoceros video are mainly due to the legs being partially occluded by grass. Results indicate that excellent segmentation is obtained using the LPS model.

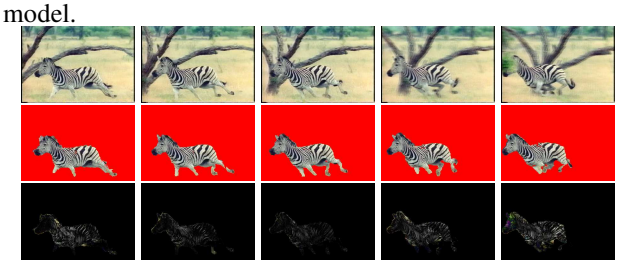


Figure 6. *Segmentation results I (Zebra sequence). The first row shows five frames of the original video sequence. The second row shows the result of segmenting the zebra from the corresponding frames. Each of these segmented frames is generated by the parameters $\mathbf{\Theta}$ of the learnt* LPS *as described in § 2. The difference in* RGB *values of the corresponding pixels is shown in the third row. Note that most of the errors are due to misalignment of texture in the head and body parts due to muscular expansion and contraction. Small parts moving non-rigidly, such as ears and mane, also result in some error.*

5. Summary and Conclusions

We have proposed a new model, LPS, which overcomes many deficiencies in previous models such as handling self-occlusion and changes in appearance due to lighting and motion blur. We also describe a method to learn the parameters of the LPS model for an object from a video in an unsupervised manner using multi-way graph cuts. The learnt model gives excellent results for segmentation of an object. The model can also be used for other applications such as pose estimation and object recognition.

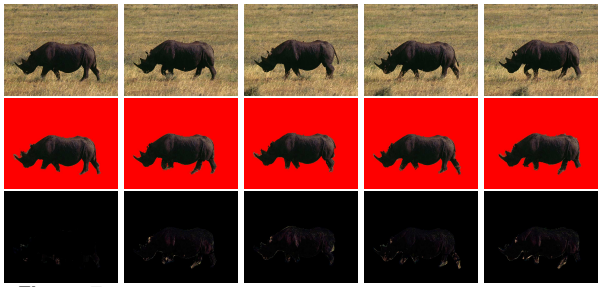


Figure 7. *Segmentation results II (Rhino sequence). Most of the errors are due to a part being partially occluded by the background.*

The method needs to be extended to include various visual aspects of an object, i.e. in addition to side views, it must also handle frontal, back and $3/4$ views. The model should also be improved to include parts of the scene not present in the reference frame.

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] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV02*, page IV: 113 ff., 2002.

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

[3] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR03*, pages II: 264–271, 2003.

[4] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *TC*, 22:67–92, January 1973.

[5] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *ICCV01*, volume 1, pages 199–206, 2001.

[6] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV02*, page III: 82 ff., 2002.

[7] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered pictorial structures from video. Technical report, Oxford Brookes University, 2004.

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

[9] D. Ramanan and D. Forsyth. Using temporal coherence to build models of animals. In *ICCV03*, pages 338–345, 2003.

[10] 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.

[11] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *CVPR00*, pages II: 101–108, 2000.

[12] 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.