Principled Fusion of High-level Model and Low-level Cues for Motion Segmentation

Arasanathan Thayananthan * Masahiro Iwasaki † Roberto Cipolla *

* University of Cambridge
Department of Engineering
Cambridge, CB2 1PZ, UK
{at315|cipolla}@eng.cam.ac.uk

† Panasonic Europe Ltd.
18a Sheraton House, Castle Park
Cambridge, CB3 0AX, UK
iwasaki.masahiro@jp.panasonic.com

Abstract

High-level generative models provide elegant descriptions of videos and are commonly used as the inference framework in many unsupervised motion segmentation schemes. However, approximate inference in these models often require ad-hoc initialization to avoid local minima issues. Low-level cues, obtained independently from the high-level model, can constrain the search space and reduce the chance of inference algorithms falling into a local minima. This paper introduces a novel principled fusion framework where, local hierarchical superpixels segmentation of images are used to capture local motion. The low-level cues such as local motion, on their own, not adequate to obtain full motion segmentation as occlusion needs to be handled globally. We fuse the low-level motion cues with the high-level model in a principled manner to surmount the shortcomings of using only the high-level model or low-level cues to perform motion segmentation. The fused model contains both continuous and discrete variables which forms a number of Markov Random fields. Variational approximation or belief propagation algorithms cannot be applied due to the complex interactions between the variables. Hence, approximate inference is performed using expectation propagation (EP) algorithm. The scheme is demonstrated by performing motion segmentation in two video sequences.

1. Introduction

Unsupervised motion segmentation of a video can be considered as a task of learning the appearances and motions of independently moving, constituent layers that are present in that video. The problem is difficult to solve because of complex object motions, occlusions and varying imaging conditions. Due to the unsupervised nature of the task, high-level generative models such as layer-representation [10, 2, 11], are used as the inference framework. The video can be reconstructed using the generative model if the hidden variables (e.g. appearances and motions of layers) are known. The motion segmentation task is concerned with the inverse process: inferring the hidden variables from the video data. Figure 1(a) describes a commonly used [2, 11] high-level model of a video with two foreground layers in front of a stationary background.

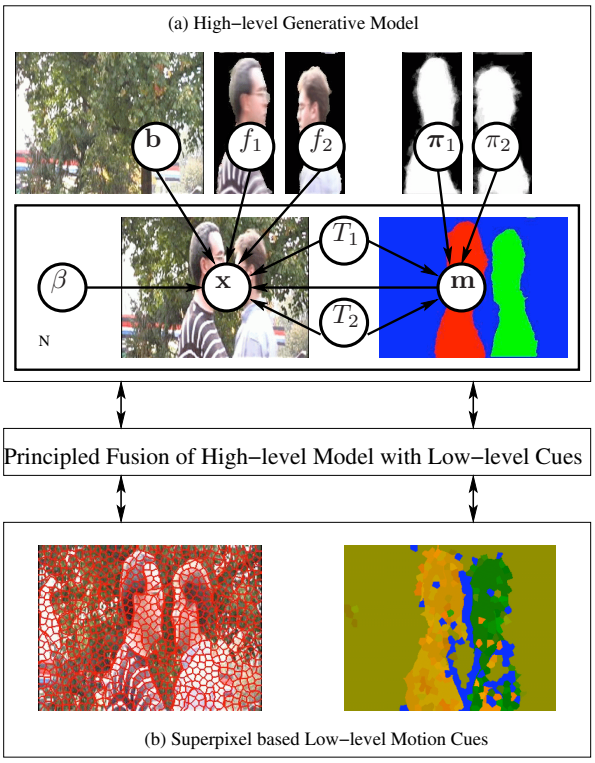


Figure 1. **Fusion of High-level model and low-level cues (a)** *An example of a layer-based generative model of video used in many motion segmentation schemes [2, 4, 11]* **(b)** *Superpixel segmentation of a video frame obtained using the algorithm proposed by Ren and Malik [5]and the corresponding local motion of the superpixels into the next frame. The motions with highest cross-correlations are shown for each superpixel. Similar colours indicate similar motions. Blue indicates the superpixels are occluded in the next frame.*

Most high-level motion segmentation schemes are hampered by local minima problems and harness various low-level cues, mostly through ad-hoc techniques to obtain good results. This paper describes a more principled way for fusing the high-level generative models with low-level cues, allowing interactions between the two levels. We use hierarchical superpixel representation of the video frames to estimate local motion (figure 1 (b)). This information is then

fused with the high-level model. Expectation propagation algorithm (EP) is used to perform approximate inference in this fused model. Next section reviews some relevant literature to set up the context for our proposed framework.

2. Related Work

Many motion segmentation schemes utilise different variations of expectation-maximisation (EM) algorithms to learn the hidden variables of the generative models in an iterative manner. The task is not straight forward; the search space is large even for simple motions such as translations and affine motions; EM type inference algorithms are hampered by local minima, and need good initialization to converge to the correct solution. The exact manner of initialization vary from scheme to scheme and often involve improvised domain-specific procedures. The common theme among these methods is that they use low-level image cues, independent of the generative model, to obtain an approximate estimation of the hidden variables and then proceed with the top-down inference. However, the quality of the convergence inevitably depends on the initialization and because of that most methods use elaborate, often ad-hoc, procedures to obtain a reasonably good initialization.

For example, Allan et al. [1] uses SIFT features to track and separate individual objects as the initialization procedure. Kumar et al. [4] first identify independently moving objects from frame to frame and then uses a complicated clustering procedure to learn the appearances of the individual objects. Once the approximate appearances and frame to frame transformations are estimated, they use the multiway graph-cut technique to refine the mask labels further. This allows them to first limit the search space of the hidden variables such as transformations of foreground objects and secondly to avoid the inference falling into a local minima. However, combining low-level information with the generative model in this sequential fashion has some drawbacks; low-level information is discarded after initialization and is not further used to guide the evolution of the inference in the generative model; due to the ad-hoc nature of the initialization procedure, it is difficult to generalise these schemes to include different types of low-level information.

On the other end of the spectrum lie the elegant, top-down probabilistic motion segmentation schemes, which completely avoids any ad-hoc initialization. Jojic and Frey [2] introduced a variational framework [12] that iteratively learns approximate factorized distributions over the hidden variables. They used this framework to segment fronto-parallel translation in videos. Winn and Blake [11] extended it to segment the affine motion of a single object in front of a static background in video sequences. They choose exponential forms for the likelihood and prior terms of the generative model and the partition function of the posterior distribution is equal to 1 by design. Additionally the graphical model is strictly maintained as a directed acyclic graph (DAG). This careful design allows them to derive an elegant variational scheme for approximating the posterior distribution. However, like other EM algorithms they are highly susceptible to local minima. Random initialization of these schemes invariably leads to wrong local minima even in simple video sequences with just two foreground objects (If there is only a single foreground object, the solution has a well defined global minima and hence EM type algorithms usually converges correctly from random initializations [11]). Instead of random initializations, using one of the explicit initialization schemes [1, 4] described above, will allow these models to converge to the correct solution. However, it will introduce the same disadvantages of the EM-type algorithms discussed above, in addition to loosing the elegance of the probabilistic methods.

It is desirable to have a framework, where low-level cues are integrated with the top-down generative model so that the initialization is handled implicitly in a principled manner, in contrast to any explicit initialization of the hidden variables. In addition, the continuous interaction of low-level cues and generative model, reduces the chances of inference falling into local-minima solution. Low-level cues introduce local constraints, while the generative model maintains consistency by providing a global context, which is essential to handle situations such as occlusions of objects. However, it is not easy to combine low-level bottom-up cues with variational frameworks in a principled manner due to the requirement to keep the graphical model as DAGs. In a later work [3] Jojic et al. propose the use of switching variables to choose from alternative likelihood and prior terms. Nonetheless, that framework is still limited to DAGs. Low-level cues are easily included through Markov Random Fields (MRF). But MRFs have complex intractable partition functions and cannot be handled in a principled manner by the variational schemes described in the above-mentioned papers [11, 2, 3].

The main contribution of this paper is a framework for unsupervised motion segmentation of videos, where low-level cues, in the form of MRFs, are fused with the top-down generative model in a principled, probabilistically meaningful manner. Pairwise potential functions link top-level hidden variables with low-level variables. A scheme based on expectation- propagation algorithm (EP) [6], is used to perform inference in the fused model. The other important contribution is the introduction of hierarchical superpixel representation of image frames as an effective tool for estimating local motion, which provides the low-level cues. The effectiveness of the framework is demonstrated by performing motion segmentation in two video sequences. The rest of the paper is organized as follows. Section 3 describes the formulation of the proposed framework; section 4 derives the EP-based approximate inference scheme. Section 5 details a number of experiments and we conclude in section 6.

3. Proposed Framework

In this section,we first briefly describe a commonly used top-down generative model framework used for motion seg-

mentation and illustrate the local-minima issues associated with the top-down approaches. Secondly we introduce a hierarchical, super-pixel based low-level motion segmentation framework. Thirdly we fuse these two frameworks together in a principled manner to avoid some of the shortcomings of these two frameworks.

3.1. Top-down Generative model

Given the input video frames $\{\mathbf{x}^t\}_1^N$ and the assumption that there are K objects or layers in the video, we are interested in learning hidden variables of the generative model: canonical appearances ($\{\mathbf{f}_k\}_{k=1}^K$) and the shapes ($\{\boldsymbol{\pi}_k\}_{k=1}^K$) of the layers; their transformations from the canonical frame to the individual frames ($\{T_k^t\}_{k=1,t=1}^{K,N}$);and the mask labels for each pixel at each frame $\mathbf{m}^t$ that indicates which layer each pixel belongs to. Figure 1 illustrates the generative model in the case of a video with three layers. Let H_g be the set of all hidden variables from the above generative model. Posterior distribution over H_g is described by $p(\mathbf{H}_g|\{\mathbf{x}^t\}) \propto p(\{\mathbf{x}^t\}|\mathbf{H}_0)p(\mathbf{H}_g)$. The generative likelihood term below, models video pixels as transformations of the canonical appearances with additive noise.

$$p(\mathbf{x}^t|\mathbf{H}_g) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}^t|T_k^t\mathbf{f}_k, \beta_k^t)^{\delta(\mathbf{m}^t=k)} \quad (1)$$

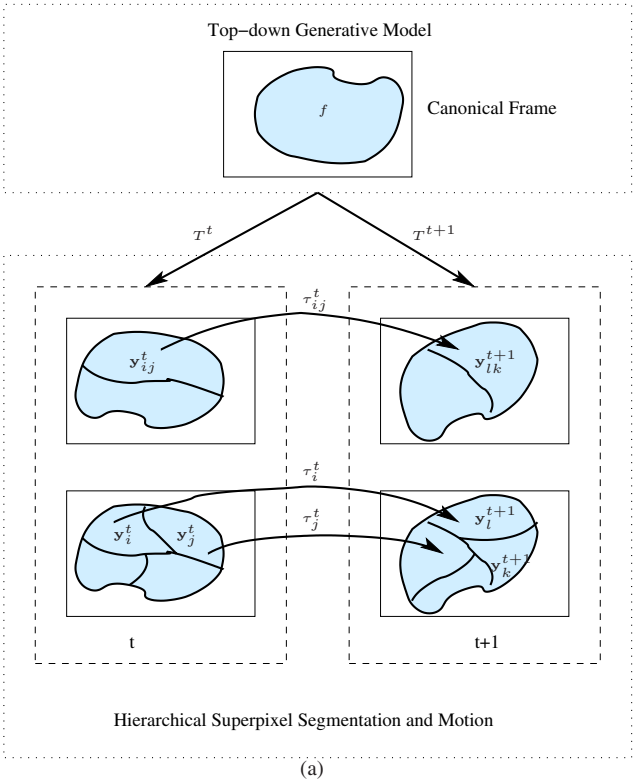
Essentially, the equation (1) states that if any of the pixels $\mathbf{x}^t$ belongs to k^{th} layer, there are explained as gaussian distributions with mean $T_k^t\mathbf{f}_k$ and precision β_k. Here the transformation T_k^t transforms the canonical appearance of the k^{th} layer into the t^{th} frame. Often the prior distributions over the hidden variables are used to model occlusion ordering [1, 11].

$$p(\mathbf{H}_g) = \left\{\prod_t p_o(\mathbf{m}^t|\{\boldsymbol{\pi}_k, T_k^t\})\right\} \prod_k p(\boldsymbol{\pi}_k)p(\beta_k)p(\mathbf{f}_k)) \quad (2)$$

with,

$$p_o = \prod_{k=1}^{K}\left\{(T_k^t\boldsymbol{\pi}_k)\prod_{j=1}^{k-1}(1-T_j^t\boldsymbol{\pi}_j)\right\}^{\delta(\mathbf{m}^t=k)} \quad (3)$$

In our case, equation (3) states that the K^{th} layer is not occluded by any other layers; $K-1^{th}$ layer is not occluded by any other layers except by the K^{th} layer, etc. The terms $p(\boldsymbol{\pi}_k)$,$p(\beta_k)$ and $p(\mathbf{f}_k)$ are called *non-informative* priors and complete a Bayesian setup for the generative model. Furthermore, they can be used to break symmetry during inference, through random initializations of its parameters. The transformations $\{T_k^t\}$ are limited to rotation and translations for the video sequence described in this paper. Translation are allowed to any whole pixel location (altogether 76800). The rotation are discretisized at 10° intervals from 0° to 360°. Hence altogether there 76800×36 candidates for each canonical transformation T^t.



(a)

Figure 2. **Fusion of high-level model with low-level cues (a)** *Illustrates how the superpixel motion τ constraints the transformations of the object from canonical frame into the video frames, T^t, T^{t+1}. For clarity, only a single foreground object is shown with some variables omitted from the model.*

We implemented the variational approximation scheme described in [11], which approximates the posterior distribution $p(\mathbf{H}_g|\{\mathbf{x}^t\})$ by a tractable, fully factorized parametric distribution $Q(\mathbf{H}_g)$, which is simply a product of the following distributions ; Gaussian distributions, $\{Q(\mathbf{f}_k)\}$; beta distributions, $\{Q(\boldsymbol{\pi}_k)\}$; Discrete distributions $\{Q(\mathbf{m}^t)\}$,$\{Q(\mathbf{T}_k^t)\}$; and gamma distributions $\{Q(\beta_k)\}$. The variational scheme obtains an optimal $Q(\mathbf{H}_g)$ by minimizing the *exclusive* KL divergence $KL(Q(\mathbf{H}_g)||p(\mathbf{H}_g|\{\mathbf{x}^t\}))$ with respect to the parameters of $Q(\mathbf{H}_g)$. The best results obtained (out of a number of randomly initialized runs) are shown in the second row of figure 4.

It is clear that the performing variational inference using the generative model alone cannot guarantee convergence to global minima even in simple videos with just two foreground objects. This is a common problems in top-down models with large number of hidden variables. The approximate posterior distribution $Q(\mathbf{H}_g)$ has many simplifying independence assumptions to make inference simpler (in our case, we assume all hidden variables are independent of each other). Often this leads to local minima convergence. In the next sub-section, we take a different approach and describe a bottom-up motion segmentation framework, based on detecting the motion of superpixels.

3.2. Super-pixel Motion Estimation

This subsection introduces a novel framework for clustering superpixels into a number of coherently moving objects, based on their low-level motion cues. Superpixels segmentation provide an over-segmentation of an image, by clustering pixels into contiguous groups based on color, texture and edge boundary. Superpixels are preferable to individual pixels or features for estimating local motion. Firstly, it provides a more meaningful segmentation improving the accuracy of estimated local motion; secondly small number of superpixels are adequate to represent an image, reducing the computational cost of estimating the local motions; and thirdly different levels of superpixel segmentation (see figure 3) can be introduced in a coarse-to-fine manner to improve the accuracy of the estimated motion. Superpixels have also been used in other vision tasks such as image segmentation [5], bottom-up human pose estimation [7] and finding object boundaries using motion cues [9].

We obtain superpixel segmentation for each frame independently, using publicly available code provided with [5]. Likelihood values for a number of candidate motions are calculated for each superpixel in each frame by matching it with superpixels in neighboring frames (note that there are no perfect matches, as superpixels' shape, size and orientation differ from frame to frame). Strong matches create strong *inter-frame links* between the superpixels from different (but neighboring) frames. Similarly we create strong *intra-frame links* between neighboring superpixels from the same frame, if their motions are similar. Given these graph-like links, our aim is to group the superpixels from all the frames into a number of layers and estimate the layers' local motion throughout the video. This leads to two intertwined MRFs on the graph structure, one over the layer labels and the other over the motion of the superpixels. The rest of the section details how Markov Random fields are formed from the superpixels' motions. For clarity we only describe the equations related to a single level superpixel segmentation and their motions into the next frame ($t+1$ th frame). MRFs arising from (1) different levels of superpixel segmentation and (2) motions into $t+2$ frame are analogous to the ones described below.

Given the superpixel segmentation at each frame, the pixel data contained within the i^{th} superpixel from frame t is denoted by $\mathbf{y}_i^t$; it's motion into the next frame is τ_i^t and it's class label $\mu_i^t \in 1..K$. These variables are illustrated in figure 2. To generate candidates for τ_i^t, for each superpixel in a frame we perform a fixed number of discrete transformations into the next *two* frames and calculate a corresponding cross-correlation value. Transformations include 7 rotations around the centre of the superpixel (at $-30^\circ - 20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ, 30^\circ$ d) and 41 translations in both x and y directions (from -20 to 20) as well as a single transformation with a fixed likelihood value accounting for the occlusion of the superpixel in the next frames. Hence altogether there are 11768 candidate transformations for each superpixel. However, we did not calculate cross-correlations for all possible motions for each superpixel. Instead a coarse-to-fine strategy was used to identify the candidate motions with highest cross-correlations. Candidate motions with low likelihoods were discarded to improve computational efficiency. Additional transformations such as scale change and shear can be easily included, but not needed for the video sequences used in our experiments. Let $\mathbf{H}_s = \{\boldsymbol{\mu}^t, \boldsymbol{\tau}^t\}$. A posterior over $\mathbf{H}_s$ is given by $p(\mathbf{H}_s|\{\mathbf{y}_t\}) \propto p(\{\mathbf{y}^t\}|\mathbf{H}_s)p(\mathbf{H}_s)$. The likelihood of superpixel $\mathbf{y}_j^t$ from frame t overlapping the superpixel $\mathbf{y}_k^{t+1}$ at frame $t+1$ through transformation τ_j^t is given by

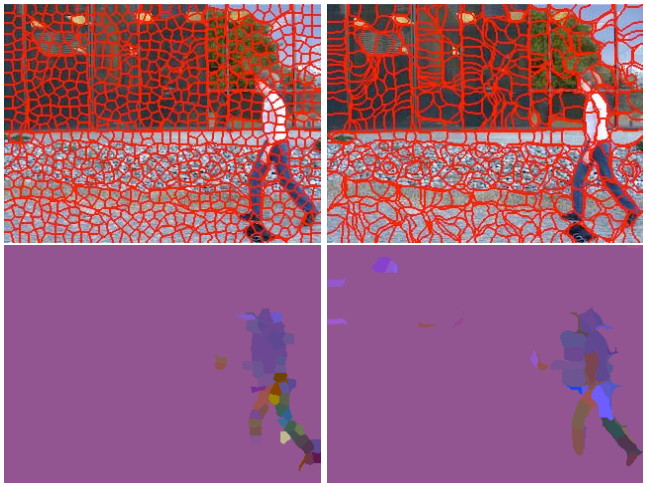


Figure 3. **Hierarchical Superpixel Motion** *We use two levels of superpixel segmentation for the walking sequence. Each frame is segmented into 200-300 superpixels on the first level and 800-900 superpixels in the second level. The images in the second row show superpixels' motion pattern into the next frame. Similar color indicates similar motions. Only the motion with the highest likelihood are shown for each superpixel (see text).*

$$p(\{\mathbf{y}^t\}_{t=1}^N|\mathbf{H}_s) = \prod_t \prod_{j,k} \prod_{\tau_j^t} p_l\left(\mathbf{y}_k^{t+1}, \mathbf{y}_j^t|\tau_j^t, \mu_k^{t+1}, \mu_j^t, \beta_\tau\right) \tag{4}$$

where,

$$p_l = \begin{cases} \exp\left\{\beta_\tau \Psi\left(\mathbf{y}_k^{t+1}, \tau_j^t \mathbf{y}_j^t\right)\right\} & \text{if} \quad \mu_k^{t+1} = \mu_j^t \\ \epsilon & \text{otherwise} \end{cases} \tag{5}$$

The function Ψ calculates the standard cross-correlation between super pixel data $\mathbf{y}_k^{t+1}$ and the transformed superpixel data $\tau_j^t \mathbf{y}_j^t$, using the pixels from common locations after the transformation. The precision value β_τ and the very-low probability ϵ was set to 10.0 and 10^{-10} in our experiments. The prior $p(\mathbf{H}_s)$ contains two type of potential functions. The first type of potential functions is formed from the fact that the neighboring superpixels from a frame should have the same motion if they belong to the same object.

$$p(\tau_i^t, \tau_j^t|\mu_i^t, \mu_j^t) = \begin{cases} \exp\left\{-\gamma_\tau \Phi(\tau_i^t, \tau_j^t)\right\} & \text{if} \quad \mu_i^t = \mu_j^t \\ \epsilon & \text{otherwise} \end{cases} \tag{6}$$

where the function Φ measures the difference between two transformations. It is difficult to compare two transformation when the actual motion involved is small; a small rotation and a translation of a superpixel can be easily approximated by another translation alone. Hence we compare the transformations by the differences in the superpixel locations $\mathbf{z}$ after the transformations. The second type of potential functions in $p(\mathbf{H}_s)$ is a data-driven Ising prior which requires that mask labels of neighboring superpixels should be the same unless there is strong edge between them is described by equation 7. The function $\Omega(\mathbf{y}_i^t, \mathbf{y}_j^t)$ provides the average edge strength along the border of the two superpixels.

$$p(\mu_i^t, \mu_j^t) = \begin{cases} 1 & \text{if} \quad \mu_i^t = \mu_j^t \\ \exp\left\{-\gamma_e \Omega(\mathbf{y}_i^t, \mathbf{y}_j^t)\right\} & \text{otherwise} \end{cases} \quad (7)$$

The hidden variables in these MRFs are all discrete and hence, we used the belief propagation algorithm to approximate $p(\mathbf{H}_s|\{\mathbf{y}_t\})$. The results are shown in the third row in figure 4. Since this low-level superpixel segmentation schemes lacks any global appearance and shape models, it cannot handle occlusion in a meaningful manner and the method fails.

3.3. Fusion of Top-down and bottom-up Schemes

We now describe a novel framework where generative framework (section 3.1) and the low-level superpixel motion segmentation scheme (section 3.2) are fused together to overcome the shortcomings described in the previous subsections. Let $\mathbf{H} = \{\mathbf{H}_s, \mathbf{H}_g\}$ be the set of all hidden variables. Posterior distribution over $\mathbf{H}$ is written as

$$p(\mathbf{H}|\{\mathbf{x}^t, \mathbf{y}^t\}) \propto p(\{\mathbf{x}^t\}|\mathbf{H}_0)p(\{\mathbf{y}^t\}|\mathbf{H}_s)p(\mathbf{H}_s|\mathbf{H}_g)p(\mathbf{H}_g) \quad (8)$$

Here, $p(\{\mathbf{x}^t\}_{t=1}^N|\mathbf{H}_g)$ and $p(\mathbf{H}_g)$ are the likelihood and prior terms of the generative model described in section 3.1. Similarly $p(\{\mathbf{y}^t\}_{t=1}^N|\mathbf{H}_s)$ is the likelihood term at the superpixel level described in section 3.2.The new term $p(\mathbf{H}_s|\mathbf{H}_g)$ contains potential functions described in $p(\mathbf{H}_s)$. In addition, it introduces the following fusion potential functions, which play a pivotal role by connecting the hidden variables of the generative model to the low-level superpixels. This allows a continuous interchange of information between the two levels throughout the evolution of the inference in the fused framework. For example, the motion of a superpixel (τ) between the frames acts as a strong prior on the canonical transformations of the objects to those frames (T_k^t, T_k^{t+1}) , (figure 2). This allows us to avoid the local minima problems described in section 3.1. The following potential functions encapsulates these constraints.

$$p(\tau_i^t|H_g) = \prod_{k=1}^{K} \left\{exp\left\{-\gamma\Phi\left(\tau_i^t, T_k^{t+1} - T_k^t\right)\right\}\right\}^{\delta(\mu_i^t=k)} \quad (9)$$

where, Φ is defined as

$$\Phi(\tau_i^t, T_k^{t+1} - T_k^t) = ||\tau_i^t \mathbf{z}_i^t - T_k^{t+1}(T_k^t)^{-1}\mathbf{z}_i^t|| \quad (10)$$

Mask label of a pixel ($\mathbf{m}^t$) and the mask label of it's parent superpixel (μ^t) should be same. Occlusion is modeled properly within the high-level model. Occlusion handling is now propagated to the superpixel level through the following constraints, which allow us to avoid the occlusion handling problems encountered in using only the super-pixel motion for motion segmentation (section 3.2).

$$p(\mu_i^t|m_j^t) = \begin{cases} 1 & \text{if} \quad \mu_i^t = m_j^t \\ \epsilon & \text{otherwise} \end{cases} \quad (11)$$

It is important to realize that once the generative model variables and the low-level variables are fused in the above manner, there is no need to differentiate between them from the inference algorithm's point of view. This is one of the main advantage of the proposed framework in that there is an abstraction between building the model and performing approximate inference. This allows one to easily add or remove functionalities and constraints within the model without worrying its effects on the approximate inference algorithm.

4. Approximate Inference in the fused framework

The combined posterior distribution given in eqn. (8) consists of both continuous and discrete variables. It is difficult to design a variational approximation scheme for this distribution due to the intractable partition function Z. Belief propagation (BP) algorithm can handle complex partition functions, but limited to discrete variables only. Instead, we use an approximation scheme based on expectation propagation (EP) algorithm [6], which generalises BP to both continuous and discrete variables.

The posterior distribution can be considered as a product of *factors*(e.g. Consider $p(a, b, c) = p_1(a, b)p_2(b, c)$. Here p_1 and p_2 are *factors* of p). EP approximates each of these factors using simpler distributions, $q_j(\mathbf{H}_j)$, and together they make up the global approximation, $Q(\mathbf{H})$. This is a crucial difference to variational approximation schemes, where there is usually only a single approximate distribution ($Q(\mathbf{H}_j)$) for each variable (see section 3.1).

$$p(\mathbf{H}|\{\mathbf{x}, \mathbf{y}\}) \propto \prod_j p_j(\mathbf{H}_j) \approx \prod_j q_j(\mathbf{H}_j) = Q(\mathbf{H}) \quad (12)$$

Here, $\mathbf{H}_j \subseteq \mathbf{H}$ is subset of hidden variables involved in the factor $p_j(\mathbf{H}_j)$. Furthermore, we use a *fully-factorized* approximation of the form

$$q_j(\mathbf{H}_j) = \prod_{\mathbf{h}_k \in \mathbf{H}_j} q_j(\mathbf{h}_k) \quad (13)$$

.

A global approximation for a particular variable $\mathbf{h}_l$ can be obtained by multiplying individual approximations of that distribution from each of the factors.

$$Q(\mathbf{h}_l) \propto \prod_j q_j(\mathbf{h}_l) \quad \text{if} \quad \mathbf{h}_l \in \mathbf{H}_j \tag{14}$$

EP algorithm proceeds by updating each $q_j(\mathbf{h}_l)$, in turn. As an example, consider the following approximation of a posterior factor from eqn. (1).

$$p_i(\mathbf{f}_1, T_1^t, \mathbf{m}^t) = \{\mathcal{N}(\mathbf{x}^t|T_1^t\mathbf{f}_1, \beta)\}^{\delta(\mathbf{m}^t=1)} \tag{15}$$

$$\approx q_i(\mathbf{f}_1)q_i(T_1^t)q_i(\mathbf{m}^t) \tag{16}$$

where, $q_i(\mathbf{f}_1)$ is a Gaussian distribution; $q_i(T_1^t)$ and $q_i(\mathbf{m}^t)$ are discrete distributions. An update for the approximate factor $q_i(\mathbf{f}_1)$ is obtained by minimising the following KL-divergence.

$$q_i^{\text{new}}(\mathbf{f}_1) = \arg\min_{q_i(\mathbf{f}_1)} \text{KL}\left(\phi_i(\mathbf{f}_1)Q^{\setminus i}(\mathbf{f}_1)||q_i(\mathbf{f}_1)Q^{\setminus i}(\mathbf{f}_1)\right) \tag{17}$$

with

$$\begin{aligned}\phi_i(\mathbf{f}_1) &= \sum_{T_1^t}\sum_{\mathbf{m}^t} \{p_i(\mathbf{f}_1, T_1^t, \mathbf{m}^t)\} Q^{\setminus i}(\mathbf{m}^t)Q^{\setminus i}(T_1^t) \\ &= \sum_{T_1^t} \left\{\mathcal{N}(\mathbf{f}_1|\,(T_1^t)^{-1}\,\mathbf{x}^t, \beta)\right\} Q^{\setminus i}(\mathbf{m}^t = 1)Q^{\setminus i}(T_1^t)\end{aligned}$$

Here, the notation $Q^{\setminus i}(\mathbf{f}_1)$ denotes the global approximate distribution of $\mathbf{f}_1$ without the contribution from $q_i(\mathbf{f}_1)$, i.e.

$$Q^{\setminus i}(\mathbf{f}_1) = \prod_{j\neq i} q_j(\mathbf{f}_1) \tag{18}$$

It is clear from the above expression that $\phi_i(\mathbf{f}_1)$ is a mixture of gaussian distribution. Both global approximation $Q(\mathbf{f}_1)$, and local approximation $q_i(\mathbf{f}_1)$, are single gaussian distributions. Minimising the *inclusive* KL-divergence in eqn. (17) is equivalent to matching the moments of the two distributions $\{\phi_i(\mathbf{f}_1)Q^{\setminus i}(\mathbf{f}_1)\}$ and $\{q_i(\mathbf{f}_1)Q^{\setminus i}(\mathbf{f}_1)\}$ by adjusting the parameters of the local distribution $q_i(\mathbf{f}_1)$. This procedure have a closed form solution and avoids any gradient descent-based optimisation of the parameters of $q_i(\mathbf{f}_1)$. Similar closed-form update schemes are also derived for $q_i(T_1^t)$ and $q_i(\mathbf{m}^t)$. Note that the noise precision β is treated as a parameter and an optimum value is obtained through gradient-based optimisation after each iterative update of $q_i(\mathbf{f}_1)$, $q_i(T_1^t)$ and $q_i(\mathbf{m}^t)$. We derive analogous update schemes for all the posterior factors from both the generative model and low-level MRFs, described in the previous section. Algorithm 1 provides a summary of the generic update scheme. Table 1 lists the the type of distributions used for each variable in the fused model. Note the notation $\mathbf{H}_j\setminus\mathbf{h}_k$ denotes "every variables in set $\mathbf{H}_j$ except $\mathbf{h}_k$". Like BP, EP algorithm can be easily implemented using message passing routines. In theory there is no need to differentiate between the variables while performing the appropriate inference. In practice, however, we use the information from the superpixel motions to limit the possible candidates for the canonical transformations of the generative model. This is done in order to reduce the computational effort needed to perform the EP inference.

$\mathbf{h}$	Description	$Q(\mathbf{h})$ & $q(\mathbf{h})$	$\phi(\mathbf{h})$
$\mathbf{f}$	layer appearance	gaussian	mix. of gauss.
T^t	layer transform	discrete	discrete
$\mathbf{m}^t$	pixel label	discrete	discrete
$\boldsymbol{\pi}$	shape	beta	mix. of beta
μ	superpixel label	discrete	discrete
τ	superpixel motion	discrete	discrete

Table 1. **Types of approximate distributions used with EP algorithm**

Algorithm 1 Expectation Propagation Update Scheme

1. Initialization

$q(\mathbf{h})$ are initialised with large variance for continuous variables and uniform distributions for discrete variables.

2. Repeat until convergence

a. For each *posterior factor* $p_j(\mathbf{H}_j)$ in $p(\mathbf{H}|\mathbf{x}^t, \mathbf{y}^t)$

For each $\mathbf{h}_k \in \mathbf{H}_j$

$$\begin{aligned}\phi_j(\mathbf{h}_k) &= \int_{\mathbf{H}_j\setminus\mathbf{h}_k} p_j(\mathbf{H}_j)Q^{\setminus j}(\mathbf{H}_j\setminus\mathbf{h}_k) \\ q_j^{new}(\mathbf{h}_k) &= \\ &\arg\min_{q_j(\mathbf{h}_k)} KL\left(\phi_j(\mathbf{h}_k)Q^{\setminus j}(\mathbf{H}_j)||q_j(\mathbf{h}_k)Q^{\setminus j}(\mathbf{h}_k)\right)\end{aligned}$$

b. For each $\mathbf{h}_l \in \mathbf{H}$

$$Q(\mathbf{h}_l) = \prod_j q_j(\mathbf{h}_l)$$

5. Experiments

Jojic-Frey video: The proposed scheme was first tested on 40 frames of Jojic-Frey video [2](15 Hz, resolution 320×240, **jojic-input1.avi**(videos are provided with the supplementary material). The results are shown in figure 4. Our implementation of the variational motion segmentation scheme proposed by [11], did not converge satisfactorily to the correct solution (second row), even after a number of runs from different random initializations and limiting the motion to translation only. This is no surprise as the variational inference on generative model alone , without the benefit of low-level information, can easily fall into local minima.

Similarly inference on the low-level MRFs alone, with out the generative model, produced comparatively poor results (third row). This is expected as it is difficult to handle occlusions without the generative model. The proposed framework, which combines the generative model with the low-level MRFs, converges correctly (fourth row, **jojic-mask1.avi**). Only a single level of superpixel segmentation was used for this sequence and the motions of the superpixels were estimated only to the next frame. We used the learned appearances ($\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{b}$) and the canonical transformations for each frame (T_1^t, T_2^t to synthesise the frames (fifth row, **jojic-synth1.avi**).

Temporal Super-resolution: Later, we sub-sampled the input video to create lower frame rate input video (3.75Hz, 10frames, **jojic-input4.avi**). We still managed to learn the appearances and the canonical transformations from just 10 frames. By interpolating the canonical transformations, we then synthesised 30 intermediate frames along with the 10 original frames to create 40 frames 15Hz video. Some of the synthesised frames are shown in the last row of figure 4 (**jojic-synth4.avi**).

Walking sequence: We also used the framework to perform motion segmentation of the walking sequence from [8](15Hz 35 frames, **walk-input.avi**). By allowing five objects with the model, we were able to extract five body parts (figure 6, which are performing independent rigid-body motion approximately throughout the video. We included two levels superpixels (figure 3 for this walking sequence. Furthermore, motion of the superpixel from one frame is estimated for the next two frames (instead of just the next frame as done for the previous sequence). The results are show in figure 5 (**walk-mask.avi**). We are able to identify the motion of meaningful body parts without using prior knowledge about the articulated structure of the human body. Note that we did not attempt to segment the motion of the hands, as they are very small and much finer superpixel representation is needed to segment their motions. This is left for future work. From the video it could be seen the estimated motion of the body parts is not smooth. This is mainly resulting from the smallest discritisation of the rotation part of canonical transformations to 10°. Allowing smaller discritisation values will smooth the motion, but will introduce additional computational cost.

Computation time: The cross-correlation values for the candidate motions of the superpixels are calculated efficiently using Fast Fourier Transform techniques (FFT) techniques. This takes approximately 20 minutes in a computer with 2.1GHz Duo Core Pentium processor and 2GB ram, for all 35 frames in the sequence. Performing the inference with expectation propagation algorithm approximately takes 4 hours, as we used a large number of iterations to guarantee full convergence, but in practice few iterations are adequate to obtain similar results (about 50 minutes). This compares well with the recent state-of-the-art algorithms which produces similar results such as Kumar et al [4](which takes more than 4 hours on the same sequences) . We are also looking into FFT-type techniques to speed up the calculations of the EP updates as fututre work. All our code was implemented in c++.

6. Conclusion

This paper has introduced a framework for fusing high-level generative model with low-level superpixel motion segmentation in a principled manner for performing unsupervised motion segmentation. Resulting model is complex and contains a number of MRFs involving both discrete and continuous variables. We derive a novel approximate inference scheme based on expectation propagation algorithm. The framework has several advantages over other schemes for motion segmentation . It separates the tasks of model building and performing appropriate inference, allowing flexibility for adding or removing functionalities and constraints within the model. It relies on continuous interaction between the high-level model and the low-level cues, for guiding the inference to a correct convergence, avoiding local minima solutions. The framework is demonstrated by obtaining similar results to state of the art in unsupervised motion segmentation.

Acknowledgments The authors would like to thank Dr. Takeo Azuma of Matsushita Electric Industrial Co., Ltd. for insightful discussions.

References

[1] M. Allan, M. K. Titsias, and C. K. I. Williams. Fast Learning of Sprites using Invariant Features. In *Proc. of the British Machine Vision Conference*,2005.

[2] N. Jojic, and B. Frey. Learning flexible sprites in video layers. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2001.

[3] N. Jojic, J. Winn, L. Zitnick. Escaping local minima through hierarchical model selection: Automatic object discovery, segmentation, and tracking in video In *Proc. Conf. Computer Vision and Pattern Recognition*, 2006.

[4] M. Pawan Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentations of video. In *Proc. International Conference on Computer Vision*, 2005.

[5] X.Ren, and J.Malik. Learning a classification model segmentation. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2003.

[6] T. Minka Expectation Propagation for approximate Bayesian inference In *Proc. Conf. Uncertainty in Artificial Intelligence*,2001.

[7] G. Mori. Guiding Model Search Using Segmentation In *Proc. IEEE International Conference on Computer Vision*, 2005.

[8] H. Sidenbladh, F. D. L. Torre, and M. J. Black. A framework for modeling the appearance of 3d articulated figures. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

[9] A. Stein, D. Hoeim, and M. Hebert. Learning to Find Object Boundaries Using Motion Cues. In *Proc. International Conference on Computer Vision* , 2007.

[10] J. Y. A. Wang, and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*,vol. 3, no. 5, pp. 625-638, 2004.

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

[12] J. Winn, and C.M. Bishop. Variational Message Passing. *Journal of Machine Learning Research*,vol. 6, pp. 661-694, 2005.

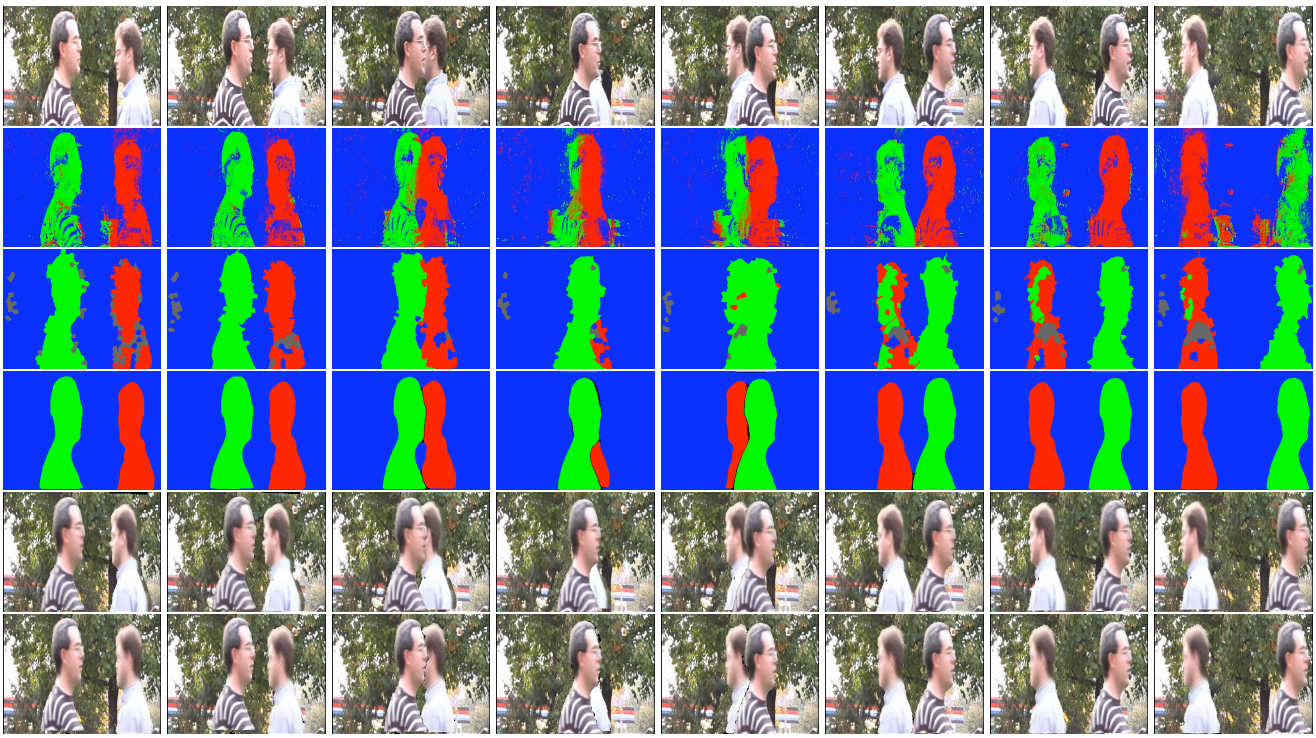


Figure 4. **Results from Jojic-Frey sequence First row:** Some of the input frames from the 40 frame sequence. **Second row:** Segmentation results from using just the generative model without low-level superpixels motion information. We used the variational scheme proposed in [11] to infer the hidden variables. **Third row:** Results from just using the low-level information without the generative model. Expectation propagation was used to perform the inference. **Fourth row:** Results from the proposed fused model using the expectation propagation inference scheme. **Fifth row:** Synthesised frames created using the the learned appearances and the canonical transformations. **Sixth row:** Temporal super resolution results (see text)

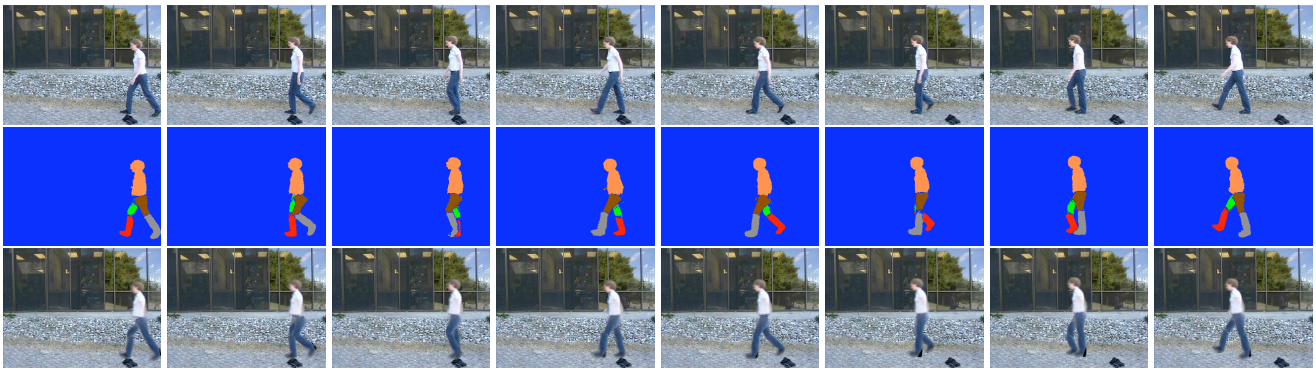


Figure 5. **Results from the walking sequence First row** Some of the input frames from the 35 frame sequence. **Second row** Segmentation results from the proposed fused model using the expectation propagation inference scheme (with five foreground objects). **Third row** Synthesised frames created using the the learned appearances and the canonical transformations.

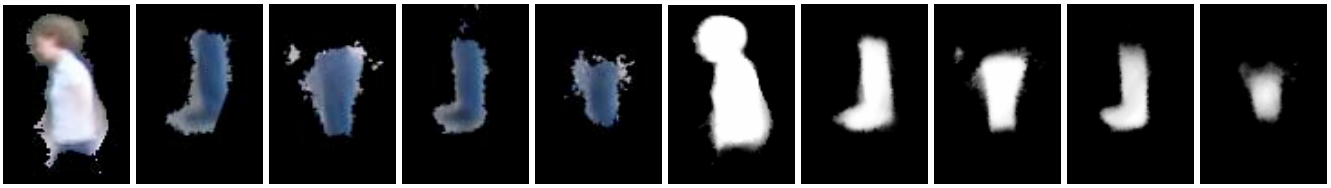


Figure 6. **Appearances and Shapes of the objects learned from the walking sequence**