Capturing video structure with mixture of probabilistic index maps

A. Perina, M. Cristani, Vittorio Murino, N. Jojic

Capturing video structure with mixture of probabilistic index maps

A.Perina[1], M.Cristani[1], V.Murino[1], and N.Jojic[2]

[1] University of Verona,
Department of Computer Science,
Strada le Grazie 15, 37134 Verona, Italy
alessandro.perina@univr.it,marco.cristani@univr.it,
vittorio.murino@univr.it
[2] Microsoft Research
Redmond, WA
jojic@microsoft.com

Abstract. The ability to segment or separate foreground from background in video images is useful to a number of applications including video compression, human-computer interaction, and object tracking to name a few. In order to generate such segmentation in both a reliable and visually pleasing manner the fusion of both spatial and temporal information is required. This fusion typically requires to process a large amount of information thereby imposing a heavy computational cost and/or requiring substantial manual interaction. This heavy computational cost unfortunately limits its applicability. In this paper a generative model to solve this problem is proposed. The model has been designed with a particular emphasis on efficiency, but also provide visually pleasing results. The approach selects salient appearance poses of the foreground shared across the entire sequence in an unsupervised way, and uses them to better extract the foreground from the single frames. Results prove the validity of the approach.

1 Introduction

Detection of foreground is an important precursor for many image and video applications, such as tracking, identification, and surveillance. Although, there is often no prior information available about the foreground object to be detected, in many situations the background scene is available in all frames of the video. In general, scene modeling must be robust enough to accommodate pixel feature variation, pixel position uncertainty, and minimize or completely avoid pixel registration, in order to be performed in real-time.
The problem of dynamic background has been recently addressed in [1], where they propose an adaptive Kernel Density Estimation technique for modeling individual pixels, working well for a stationary camera but difficult to generalize to a moving camera.
The concept of layered representation of a scene was first introduced by Adelson

[2], making a very good case for modeling a video scene as a group of layers instead of single pixels. Many other work followed this trend, for example [3]. Within this approach, others try to learn the appearances of objects in videos [4] to better extract the foreground.
In this paper, we propose a general robust method to extract foreground objects from color video sequences with non-static background. The model is an extension of the probabilistic maps model proposed in [5]. To cope with various changes in the foreground pose and position, we cluster the frames into salient poses, we infer priors dependent on these poses and we use them to extract the foreground of the single frames. In practice, to extract foreground from a frame we use the information carried by the neighboring frames. It is worth to note that we think at the neighbor relation, not in terms of time, but in terms of structure. In this way, although possible, we do not use any "temporal window" to define this relation, but we define the neighboring frames through a clustering operation. To improve the performances we smoothed the parameters of the generative model as proposed in [6]. Experimental results show that the proposed algorithm works well while keeping really low computational requirements.
The rest of the paper is organized as follows. Section 2 describes the probabilistic index map model and our extension to video modeling. Section 3 details the inference algorithm. In Section 4 the smoothing operations are discussed and the problems, and the related solutions, arising from such modelization are reported. In Section 5, numerical and visual results are reported. Finally, in Section 6, the obtained results are commented and some considerations are drawn.

2 Palette Indexing and probabilistic index maps

One efficient representation of an image is the palette-based representation. Each f-th image is seen as a collection of indices s_{ij}^f (or an index map), one index per pixel per image, that points to a separate palette table containing the possible intensities that a pixel can assume (see Figure 1)B. This representation is heavily used in many images formats, as it drastically reduces the storage requirements. If a set of images shares the structure, the index map s_{ij} are dependent on the pixel coordinates i, j and this information is shared across the collection of images (Figure 1B). With this assumption we obtain a basic palette-invariant model which assumes a fixed spatial arrangement of the features, but the features themselves can arbitrarily change from one image to the next. In the following, we will refer to each region pointed by an index, as *image part*. For the example depicted in Figure 1 these are, the hairs (s=1), the skin (s=2), and so on.
The idea in [5] and [7] was to relax the hard assumption that indices that point the same locations across the F images composing a dataset should be equal (i.e., $s_{ij}^1 = s_{ij}^2 = \ldots s_{ij}^F = s_{ij}$), allowing that the same indices can vary but following the same distribution (i.e., $p(s_{ij}^1 = s) = p(s_{ij}^2 = s) = \ldots p(s_{ij}^F = s) = p_{ij}(s)$). The distributions $p_{ij}(s)$ represents priors over the index map and they are multinomials of dimensionality equal to S, describing the variability in different locations of the image; therefore $p_{ij}(s = k)$ represent the probability

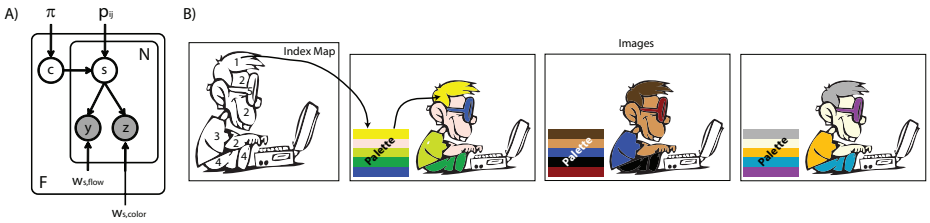

Fig. 1. A) The video-PIM Bayesian network B) An illustration of the index map as a palette-invariant representation.

that the pixel (i, j) points to the k-th entry of the palette. To summarize the model assume that each f-th image has its own value for s_{ij}^f, but the prior on that value $p_{ij}(s)$ is shared across the images. The immediate effect is that in this way we obtain consistent indices. For example, in Figure 1 the index $s = 1$ models in all the images the hair whose color can change due to the image palette. An illustration of the probabilistic version of $p_{ij}(s)$ can be found in Figure 3, on the top.

The overall distribution over the index maps $S = \left\{ s_{ij}^f \right\}$ is

$$p(\mathbf{S}) = \prod_{i,j,f} p_{ij}(s_{ij}^f) \tag{1}$$

and the joint probability distribution is

$$p(\mathbf{S}, \mathbf{Z}) = \prod_{i,j,f} p(z_{ij}^f | s_{ij}^f) \cdot p_{ij}(s_{ij}^f) \tag{2}$$

where z^f represents the f-th image.

For example, if a video shows a tree that moves slightly due to the wind, then the added level of variability in the index map p_{ij} helps to capture the flutter of the leaves. If the palette is not shared across the training set (i.e., each image has its own palette) the model can take into account for changes of illumination. In the case of a generic video flow it is not possible to think that the structural variability captured by the model, can be sufficient to model all the variability present in the sequence. This can be sufficient locally, only between near frames. To solve this problem, we propose the use of a mixture variable which cluster the frames. The centroids obtained clustering the frames model salient/particular configurations of a sequence, leaving the modeling of the transition frames to the probabilistic index map. The situation is shown in Figure 2, where we highlighted four salient configurations. It should be noted how the difference between the temporal clustering and the structural clustering is here evident: eventually the last frames have structure similar to the first frames and hence they are modeled by the same component.

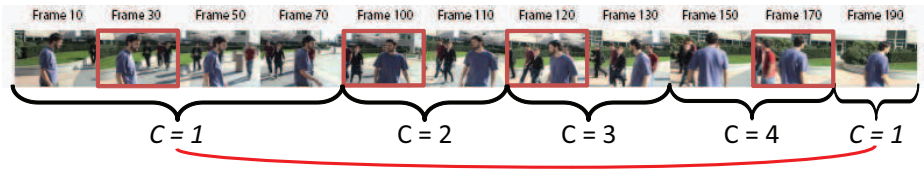

Fig. 2. Mixture centroids models frames with similar structure.

3 Model inference and learning

Consider a video sequence composed by F frames $\mathbf{Z} = \left\{z^f\right\}_{f=1}^{F}$, where $i = 1 \dots I$ and $j = 1 \dots J$ indexes pixels, and $f = 1 \dots F$ index a frame. We associate an intensity palette $\mathcal{C}^f$ to each image and an index map s_{ij} with each pixel sharing it across images (see Figure 3). Each palette $\mathcal{C}$ is a table composed by S colors, indexed by s. For example, $\mathcal{C}(s) = \mu_s$ could be equal to a RGB vector $[r, g, b]^T$. At this point, the color of a pixel becomes $z_{ij}^f = \mathcal{C}(s_{ij}^f)$.
In many image formats, it is assumed that each image has its own color table. In the case of a video sequence, if no relevant changes of illumination are present, it is possible to share the palette between the frames since no one can expect a large variation of the colors.
In general, we can think each color palette entry $\mathcal{C}(s)$ being modeled by an unimodal distribution, also called observation model; in this way we assign a probability to each pixel of having as specific intensity, the s-th color in the palette, i.e. $p(z|\mathcal{C}(s))$. For example in [5] a Gaussian distribution is used.
The problem with an unimodal distribution over the palette is that it can model image parts composed by just one color. Since the purpose of this paper is to extract the foreground from a video, we cannot assume that it is colored by a single color. Therefore the unimodal choice is not suitable.
Fortunately there are other kind of models that represent different colors in the same entry of the palette, like the histogram observation model, or the mixture of gaussian observation model. For efficiency reasons, we chose to employ an histogram-based model which have the following form:

$$p(z_{ij}^f = k|\mathcal{C}(s)) = \mathtt{w}_s(k) \tag{3}$$

In these kinds of models, each s-th entry of the palette is a multinomial distribution or a normalized histogram. The k-th bin of the histogram represent a color $B_k \rightarrow [r, g, b]^T$ and this is achieved by discretizing the measurements through a separate clustering of local measurements to create a codebook of colors. In this way $\mathtt{w}_s(k)$ represent the probability to find the k-th color in the s-th image part. An illustration of the histogram palette can found in Figure 3. This palette model was previously used in a related model, dubbed LOCUS [8], but only for the case of two image segments (foreground/background) while here we have to deal with multiple image parts.
The second additional solution to model a video stream is to introduce a mix-

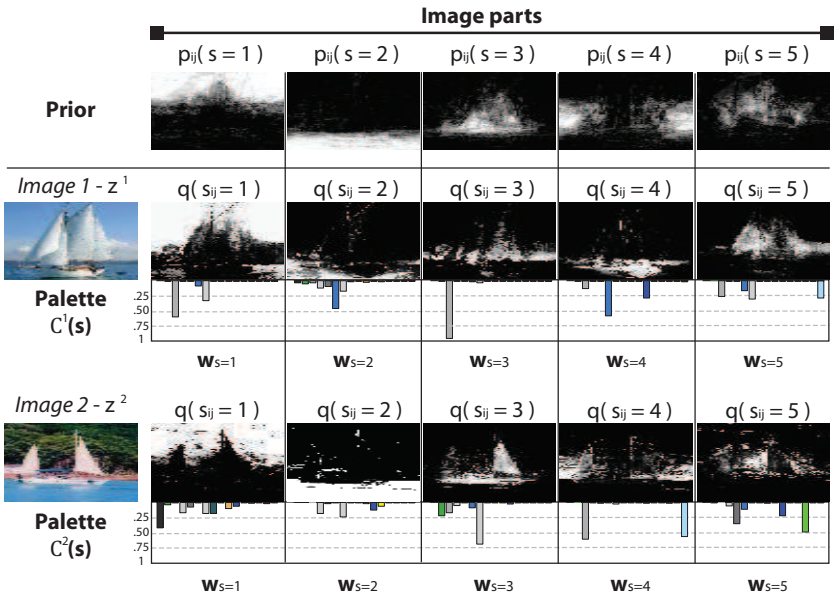

Fig. 3. Probabilistic index map. Top: the inferred priors p_{ij} for a dataset composed by 50 boat images. Bottom: two examples, with their posterior and the histogram palettes. The variables s_{ij}^f are represented in a probabilistic way $q(s_{ij}^f)$ because of the particular inference paradigm used (see Sec.3).

ture variable c, similar to what done in [5] for clustering purposes, to capture the variability in the structure of the foreground along time. As expected, the mixture variable affects the structure contained in the index variables $S = \{s_{ij}\}$ and the joint probability distribution becomes

$$p(\mathbf{c}, \mathbf{S}, \mathbf{Z}) = p(c^f) \cdot \prod_{i,j,f} p(z_{ij}^f | s_{ij}^f) \cdot p_{ij}(s_{ij}^f | c) \tag{4}$$

Here with each component we want to model the variability present in the video that can not accounted for by the probabilistic indices. For example in Figure 4 are shown the learnt priors for the test video stream. It is visible how the first component models the first 70 frames, where the boy (FG) taken from the side, while the second component models all the other poses. The higher the number of components are used, the more salient poses are learnt. Since these $p_{ij}(s)$ are used as prior for the segmentations of the single images, it is straightforward to understand how this improves the segmentation accuracy.
The model described so far is not sufficient to solve the foreground extraction problem since often a color present in the foreground appear also in the background. In this case, the probabilistic index map model would tend to merge the parts of background and foreground that share the appearance (color). To solve this problem we introduce in the model a second observation, the optical

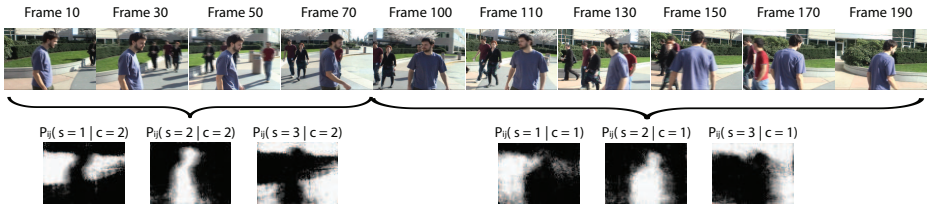

Fig. 4. The effect of the mixture variable: each cluster models a salient configuration/pose.

flow $\mathbf{Y} = \left\{y^f\right\}_{f=1}^{F}$ (OF). Due to its discrete nature, we use again the discrete histogram observation. With this novel observation plugged into the model, the final joint probability distribution becomes

$$p(\mathbf{c},\mathbf{S},\mathbf{Z},\mathbf{Y}) = p(c^f)\cdot\prod_{i,j,f} p(z_{ij}^f|s_{ij}^f,\mathcal{C}_{color}(s))\cdot p(y_{ij}^f|s_{ij}^f,\mathcal{C}_{flow}^f(s))\cdot p_{ij}(s_{ij}^f|c) \quad (5)$$

Differently from the color, the OF palettes are not shared across the images; in fact their meaning is just to keep separate the foreground and the background, which intuitively would have different flow distributions.
The OF and the color are tied at level of the index map variable s_{ij}, therefore the model search for configurations in which the intra-image OF and the inter-images color are consistent. The graphical model which describes equation 5 is shown in Figure 1A.

3.1 Free energy of a graphical model

The aim of the inference algorithm is to learn a distribution over the latent variable of the model, given the observed variables. In this case, the observed variables are the video frames and the optical flow $\mathbf{v} = \left\{ < z^f, y^f > \right\}_{f=1}^{F}$, while the hidden variables are the index map and the mixture variable $\mathbf{h} = \left\{ s^f, c^f \right\}_{f=1}^{F}$
A standard way to solve this problem is maximize the (log) likelihood of the observed data, obtained by integration of the hidden variables $\mathbf{h}$ for a given set of parameters θ, i.e., $\log p(z,y|\theta) = \int_h \log p(z,y,h|\theta)$. In our model, we treat the index variable s^f and the mixture variable c^f as hidden variables, and color maps $p_{ij}(s)$ and the palettes $\mathcal{C}_{color} = \mathtt{w}_s^{color}(k_1)$ and $\mathcal{C}_{flow}^f = \mathtt{w}_{f,s}^{flow}(k_2)$ as parameters. When this optimization problem is intractable, approximate methods such as variational inference must be used. The objective function to be optimized is the free energy [9] a quantity which bounds the negative log likelihood of the data, and is defined as

$$\mathbf{F} = \int_h q(\mathbf{h})\cdot\log q(\mathbf{h}) - \int_h q(\mathbf{h})\log p(\mathbf{v},\mathbf{h}|\theta) \quad (6)$$

where $q(\mathbf{h})$ are arbitrary distributions on the hidden variables, thought as approximate posterior distributions that can be used to compute a lower bound

of the log likelihood of the data. The variational inference consists in choosing a particular form for the q distribution as to make tractable the optimization of F using an iterative method. In this case, we can use both the fully factorized posterior $q(\mathbf{h}) = q(s) \cdot q(c)$ (known as mean-field form) or the full (correct) posterior distribution $q(\mathbf{h}) = q(c) \cdot q(s|c)$ (which often results intractable). For efficiency reasons we choose the mean-field form.
At this point, we can derive the inference (E step) and the parameter update rules (M step) for the model. In this model each observation has a separate index s_{ij}^f modeled in a probabilistic way (i.e., $q(s_{ij})^f$) due to the variational inference, but the prior $p_{ij}(s|c)$ is shared for each location among the frames (see Figure 3).
To obtain the inference procedure, derivatives of $\mathbf{F}$ w.r.t the q functions has to be calculated, paying attention that $\int_h q(h) = 1$. This normalization constraints can be easily taken into account by using the Lagrange multipliers. The inference update rules are reported below

$$q(c^f = c) \propto \pi_{c=c} \cdot e^{\sum_{ij} \sum_s q(s_{ij}^f = s) \cdot \log p_{ij}(s_{ij} = s | c = c)}$$

$$q(s_{ij}^f = s) \propto \mathtt{w}_{s_{ij}=s}^{color}(z_{ij}) \cdot \mathtt{w}_{f,s_{ij}=s}^{flow}(y_{ij}) \cdot e^{\sum_c q(c^f = c) \cdot \log p_{ij}(s_{ij} = s | c = c)}$$

$$\pi_{c=c} = \frac{1}{F} \sum q(c^f = c)$$

$$p_{ji}(s = s | c = c) = \frac{\sum_f q(s_{ij}^f = s) \cdot q(c^f = c)}{\sum_f q(c^f = c)}$$

$$\mathtt{w}_s^{color}(k) = \frac{\sum_f \sum_{ij} [z_{ij} = k] \cdot q(s_{ij}^f = s)}{I \cdot J \cdot F} \quad \mathtt{w}_{f,s}^{flow}(k) = \frac{\sum_{ij} [y_{ij} = k] \cdot q(s_{ij}^f = s)}{I \cdot J}$$

where $[x = k]$ is the indicator function, equal to one, if $x = k$, and zero otherwise.

4 Smoothing, Generalized EM and model selection

The EM algorithm has a greedy nature and usually converge to a local minima very fast [10]. In the case of video probabilistic index map, this problem usually leads to the disalignment of the index map priors across the mixture components. This means that a given probability map $s = i$ can be modeled by $s = j, \ \ i \neq j$ in another component (cluster). It is straightforward to understand how this problem has to be avoided, otherwise it would be impossible to find a mapping between S and FG since this would depend on the mixture variable C. The essence of the problem stays in the $q(c)$ distribution which becomes peaky in the first iterations of the EM, when the prior over the index maps p_{ij} is not well defined; in practice for each mixture component, the prior is inferred independently (see Figure 5A). To avoid this behavior, we propose the smoothing of the mixture distributions $q(c)$ for the first few iterations. The idea that yields to this solution is that since the input data are time-ordered video frames, it is not an hard assumption to think that the posterior distributions $q(c)$ of near frames would only slightly change between them [11, 12]. We will refer to this

as *temporal-smoothing*. In this way, we ensure that at least some frames have a balanced $q(c)$ distributions, that will contribute to all the priors. The immediate effect is the creation of a (temporary) link between the C mixture coefficients for the first iterations; as soon as the priors begin to become defined, we can release the smoothing and leave the posterior update normally, obtaining aligned index maps (see Figure 5B). This heuristics has proved to work in every test, it is not computationally expensive and is introduced and supported in [6], where the authors showed how smoothing parameters of a Bayesian network lead to better results.

Following the same intuition, we performed temporal-smoothing on $q(s)$ to en-

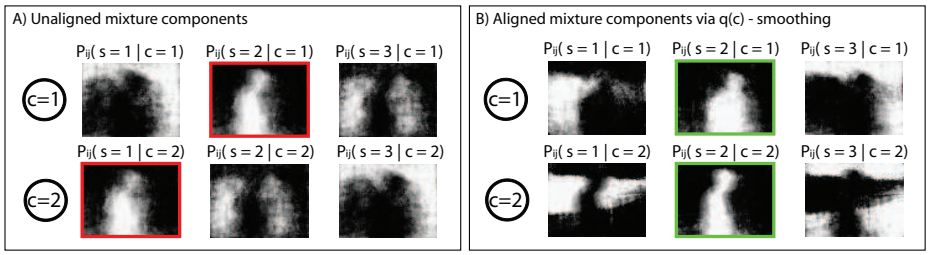

Fig. 5. A) Bad alignment of the index maps across the components. Frames whose $q(c^f)$ points at $c = 1$ have foreground modeled by S=2, while frames whose $q(c^f)$ point to $c = 2$ have foreground modeled by $S = 2$. B) Exact alignment of the index maps obtained smoothing the mixture distributions.

sure that near frames has similar posterior index maps.

Following the ideas in [6], to improve the quality of the segmentation results, we also performed *spatial-smoothing* on the pixels of $q(s_{ij}^f)$, to "prefer" that close pixels belong to the same image part, and on $p_{ij}(s)$ to soften the strength of the prior which, for its nature tend to be too peaky.

5 Experimental results and conclusions

The probabilistic index map technique has already been tested on video data in [13] where a video-surveillance sequence with significative changes of illumination (from daylight to night) and static background was used to show the invariance of the model to illumination changes and how well the model captures the scene structure.

Our goal here is to show how we can model a generic video using 1) histogram observation model, 2) the optical flow as second observation, 3) parameters smoothing and 4) the mixture components. To achieve this goal we took 220 frames of a video sequence used to test the hierarchical model selection idea of [13]. This sequence contains significant illumination changes, background clutter, significant

(and confusable) foreground and background motion, as well as dramatic changes in size and pose of the foreground object. The comparison was made based on a manual pixel-based segmentation of every 10th frame[3]. Given the groundtruth data, there is one out of possible labels for each pixel $l_{ij} \in \{0, 1\}$, where 0 refers to background (BG) and 1 refers to foreground (FG). After the learning phase, we have also S labels for each pixel based on the model $s_{ij} \in \{1, \ldots, S\}$. These labels are probabilistic, so we have $q(s_{ij} = s)$ rather than just a discrete s_{ij}. To create a correspondence between s and l, we need a mapping $\phi : s \rightarrow l$ to evaluate the segmentation (i.e., $\phi(s = i) = 1$). The overall segmentation accuracy (**OV**) would be akin to the probability that this mapping returns the exact label image, or the fraction of the pixels that are correctly labeled. In formulae:

$$\mathbf{OV}_k = \frac{1}{N} \sum_{ij} \sum_{s} [\phi_k(s_{ij}) = l_{ij}] \cdot q(s_{ij} = s) \tag{7}$$

where the index k addresses a particular mapping function.
Since we used a small value for S (2, 3 or 4), it is reasonable to consider as result $\mathbf{OV} = \max_k \mathbf{OV}_k$, that is to report results for the best mapping ϕ. The problem with this measure is that it is biased, since it depends on the relative size of the foreground and background. For example, if the foreground object is very small or very large, this value will be close to 100%. To overcome this problem we also computed the individual true positive/true negative rates for each of the labels l.

$$\mathbf{FF} = \frac{1}{\sum_{ij}[l_{ij} = 1]} \sum_{ij} \sum_{s} [\phi(s_{ij}) = 1] \cdot q(s_{ij} = s) \tag{8}$$

$$\mathbf{BB} = \frac{1}{\sum_{ij}[l_{ij} = 0]} \sum_{ij} \sum_{s} [\phi(s_{ij}) = 0] \cdot q(s_{ij} = s) \tag{9}$$

To extract the optical flow we used the well-known Lucas-Kanade method [14], the histogram size for the colors is set to B = 7, but higher values can be used as well with no relevant effect on the results. The results are presented in Table 1 in terms of various choice of S and K, where we defined K as the number of mixture component per frame (i.e., K = F / C). In Figure 7 some visual results are shown.

As expected, increasing the number of the mixture components we can model a larger number of salient poses and performance increases. The results show how the method is able to extract with good performances the background with low computational requirements. It should be noted that [13] reports better performances but is a much more complex hierarchical model composed by 7 hidden variables, some of them constrained and with multiple components specialized for video processing, including LOCUS model [8] which, on its own, significantly underperformed the full mix. In Figure 6 the learned priors $p_{ij}(s)$ for $C = 2, 3, 4$ and the frames with more similar structure are shown. It is easy to see how the

[3] The groundtruth is available upon request to the corresponding author.

Table 1. Accuracy of the proposed model foreground extraction. The results obtained in [13] are reported in the last row. In this case the values of K have no meaning.

	K = 100			K = 50			K = 25		
	FF	BB	OV	FF	BB	OV	FF	BB	OV
S=2	0.69	0.70	0.70	0.70	0.72	0.72	0.70	0.76	0.74
S=3	0.88	0.87	0.87	0.89	0.88	0.88	0.90	0.90	0.90
S=4	0.81	0.90	0.87	0.81	0.93	0.89	0.82	0.95	0.92
[13]							0.95	0.95	0.95

higher the C value, the more significant pose the detected poses. For example, if $C = 2$ the model learns the boy walking toward left and a mixture of all the other poses. Increasing C to 4, the model learn also the boy walking from behind.

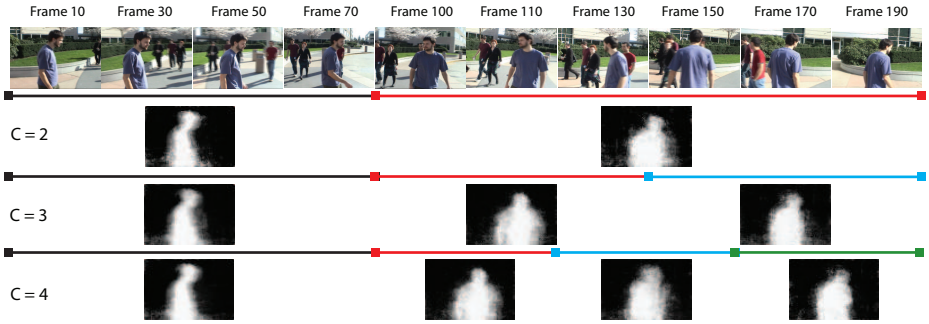

Fig. 6. Probabilistic index maps for the image part which models foreground and their position over the temporal axis.

6 Conclusions

This paper presents an application to foreground extraction using the mixture of probabilistic index map technique. The model is able to cluster the frames in salient poses calculating priors on the structure and using them to extract the foreground. Despite the smoothing operations used to align components and to improve the segmentations, the computational effort is really low. In particular, in [5] is shown how the probabilistic index map model can be used in real time. Taking the advantage of a new, more efficient, observation model it is straightforward to note how this method can also do it. Future effort will investigate how to mix these priors since the frames far from the structure captured by the prior, or with a unusual structure are penalized. This derives by the fact that the mixture variable does not carry information about the degree of membership

to a particular cluster as it is only a pointer to a discrete value. Actually, the degree of membership, is lost in calculating the posterior (normalization). This leads to the choice of a single component to calculate the segmentation and to the problems discussed above.
Other research efforts will be devoted to substitute the clustering operation by the splitting of the temporal axis in user-predefined slices taking advantage of the fact that close frames in a video most likely share a common structure.

References

1. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. cvpr **02** (2004) 302–309
2. Adelson, E.H.: Layered representation for vision and video. In: VSR '95: Proceedings of the IEEE Workshop on Representation of Visual Scenes, Washington, DC, USA, IEEE Computer Society (1995) 3
3. Patwardhan, K., Sapiro, G., Morellas, V.: Robust foreground detection in video using pixel layers. IEEE Trans. Pattern Anal. Mach. Intell. **30** (2008) 746–751
4. Tao, H., Sawhney, H.S., Kumar, R.: Dynamic layer representation with applications to tracking. cvpr **02** (2000) 2134
5. Jojic, N., Caspi, Y.: Capturing image structure with probabilistic index maps. In: CVPR (1). (2004) 212–219
6. Friedman N., G.D., M., G.: Bayesian networks classifiers. Machine Learning **29** (2007) 131–163
7. Jojic, N., Caspi, Y., Reyes-Gomez, M.: Probabilistic index maps for modeling natural signals. In: AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence, Arlington, Virginia, United States, AUAI Press (2004) 293–300
8. Winn, J., Jojic, N.: Locus: learning object classes with unsupervised segmentation. Volume 1. (2005) 756–763 Vol. 1
9. Jordan, M., Ghahramani, Z., Jaakkola, T., Saul, L.: An introduction to variational methods for graphical models. Machine Learning **37** (1999) 183–233
10. Reddy, C.K., Chiang, H.D., Rajaratnam, B.: Trust-tech-based expectation maximization for learning finite mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **30** (2008) 1146–1157
11. Ghahramani, Z.: Learning dynamic bayesian networks. In: Adaptive Processing of Sequences and Data Structures, International Summer School on Neural Networks, "E.R. Caianiello"-Tutorial Lectures, London, UK, Springer-Verlag (1998) 168–197
12. Frey, B., Jojic, N.: A comparison of algorithms for inference and learning in probabilistic graphical models. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 1392–1413
13. Jojic, N., Winn, J., Zitnick, L.: Escaping local minima through hierarchical model selection: Automatic object discovery, segmentation, and tracking in video. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, IEEE Computer Society (2006) 117–124
14. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision (ijcai). In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81). (1981) 674–679

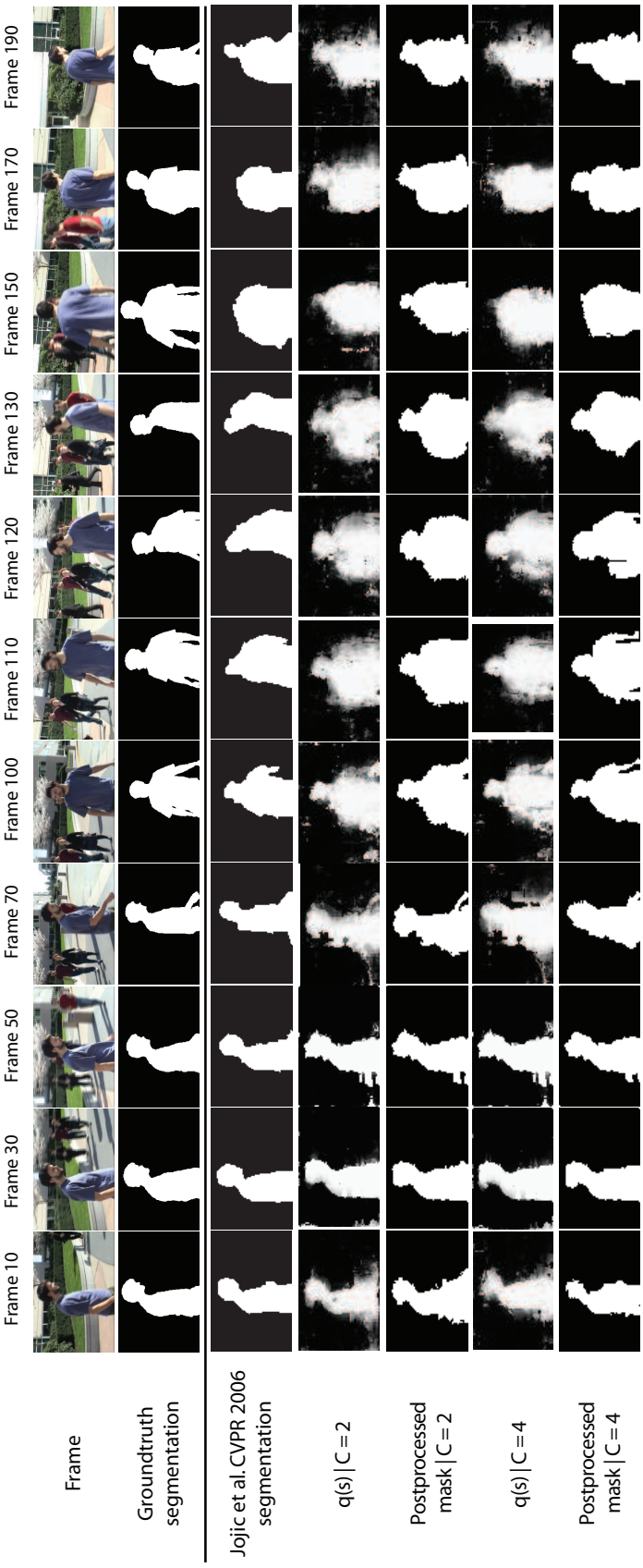

Fig. 7. Top: some segmented frames and their ground truth. Bottom: the segmentations obtained in [13], the probabilistic maps obtained by our method (for C=2 and C=4) and their correspondent postprocessed masks. Postprocessing operations include deterministic operations such as thresholding, and hole filling.