# Recursive estimation of generative models of video

Nemanja Petrovic
Google Inc.

Aleksandar Ivanovic
University of Illinois Urbana

Nebojsa Jojic
Microsoft Research

Sumit Basu
Microsoft Research

Thomas Huang
University of Illinois Urbana

## Abstract

*In this paper we present a generative model and learning procedure for unsupervised video clustering into scenes. The work addresses two important problems: realistic modeling of the sources of variability in the video and fast transformation invariant frame clustering. We suggest a solution to the problem of computationally intensive learning in this model by combining the recursive model estimation, fast inference, and on-line learning. Thus, we achieve real time frame clustering performance. Novel aspects of this method include an algorithm for the clustering of Gaussian mixtures, and the fast computation of the KL divergence between two mixtures of Gaussians. The efficiency and the performance of clustering and KL approximation methods are demonstrated. We also present novel video browsing tool based on the visualization of the variables in the generative model.*

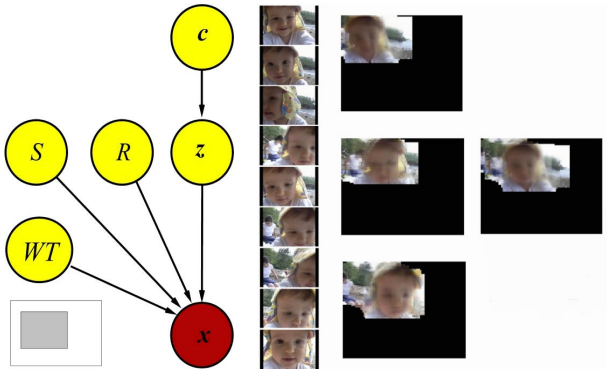


Figure 1. Left: Scene generative model. Pair $c - \mathbf{z}$ is a Gaussian mixture. Observation $\mathbf{x}$ is obtained by scaling by $Z$, rotating by $R$, transforming and cropping the latent image $\mathbf{z}$ by translation indexed by $T$. $W$ is the fixed cropping window that models frame $\mathbf{x}$ as the small part of the video scene $\mathbf{z}$. The effect of the composition $WT$ is illustrated as the shaded rectangle that indicates the position of $\mathbf{x}$ in $\mathbf{z}$. Right: Nine typical frames from the video are initially clustered into three clusters using only translation invariant clustering. Three so obtained distributions are clustered into a single cluster using translation, scale and rotation invariant distribution clustering (cf. Sec. 3).

## 1. Introduction

The amount of video data available to an average consumer has already become overwhelming. Still, there is a lack of efficient general-purpose tools for navigating this vast amount of information. We suggest that a successful video browsing and summarization system has to accomplish two goals. First, it shall correctly model the sources of vast information content in the video. Second, it shall provide the user with an intuitive and fast video navigation interface that is *compatible*, if not even jointly optimized with the analysis algorithm. As a solution of the first problem we propose the clustering of related but non-sequential frames into scenes. Clustering is based on the generative model (Fig. 1) that builds on the model for translation invariant clustering [14]. Learning in the generative model with multiple discrete variables faces considerable computational challenges. We utilize the properties of video signal to develop provably convergent recursive clustering algorithm. To make the model more realistic, it assumes the video frame to be a portion of the video scene, which is reminiscent of the panoramic scene representations. As a solution of the second problem we propose a "video navigation" tool based on the visualization of the variables in the generative model, which ideally reflects all frames in the video. The navigation tool serves as a visually meaningful index into the video.

Video clustering and summarization is one of the most difficult problems in the automatic video understanding. It aims to produce short, yet representative, synopsis of the video by extracting pertinent information or highlights that would enable the viewer to quickly grasp the general story or navigate to the specific segment. Two main approaches to video summarization include static summarizations (including shots, mosaics and storyboards), and dynamic summarizations (video skimming). Numerous shot and key-frame detection approaches are based on extracting and tracking low level features over the time and detecting their abrupt

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

changes. But, long lists of shots result in another information flood rather than abstraction, while key frames only are not sufficient for a user to judge the relevance of the content. Dynamic video summarization, often referred to as skimming, consist of collecting representative or desirable sub-clips from the video. The navigation tool we introduce in this paper has the distinctive feature that it includes both static and dynamic summary of the video.

Several shot-independent approaches for summarization have been proposed recently, like the recursive key-frame clustering [9, 10]. While promising, they lack robust similarity measure and with the number of clusters above the certain levels, visually different shots start to merge.

There have been many interesting approaches to video browsing and summarization based on mosaicking. In this paper we introduce *probabilistic* mosaic representation of video that includes the invariance with respect to camera motion, change in the scale and rotation. It assumes the video scene to be much larger that the viewing field of the camera. We should emphasize that our goal is not the building of perfect mosaics. Rather, it is the constructing of robust similarity measure that yields to the high likelihood of the frame under the generative model.

Similar mosaicking representations [4, 5, 6, 7] were used before, some of them based of generative models [8], but were overly constrained with respect to the camera motions, requirements that target scenes must be approximately planar and should not contain moving objects. For example, [5] studied mosaics in the context of video browsing and a synoptic view of a scene. Video summaries work [6] uses mosaics for summarization, but ignores the foreground motion and relies on the background invariance for the scene clustering together with ad-hoc scene similarity measure. Our work is similar to the mosaicking work [7] on mosaic-based representations of video sequences. There are a few important differences. Our method, having video clustering and compact video presentation as the ultimate goals, has a notion of variability of appearance (moving objects in the scene and blemishes in the background are allowed and treated as the noise), automatically estimates the number of parameters (eg. number of different scenes), explains the cause of variability of the data, and recognizes the scene that already appeared. Also, other methods were used in the highly regimented cases (eg. aerial surveillance, "sitcoms") where our is intended for the general class of unconstrained home videos.

Realistic graphical (generative) models may sometimes face serious computational challenges. Similarly, naive learning in this model is infeasible. Clustering of one hour of video does not allow visiting each datum more than once. This constraint suggests "one pass" over-clustering of the frames, followed by iterative cluster grouping. Each of this operations correspond to the re-estimation of the parameters in the model. We derive the algorithm for recursive estimation of this model based on the EM algorithm, thus inheriting its convergence and optimality properties. Fast inference methods and video navigation tool are the features of this work.

## 2. Model

The video analysis algorithm we present is based on a generative model Figure 1 (left) that assumes the video scenes are generated by a set of normalized scenes that are subjected to geometrical transforms and noise [3]. The appearance of the scene is modeled by a Gaussian appearance map. The probability density of the vector of pixel values $\mathbf{z}$ for the latent image corresponding to the cluster $c$ is

$$p(\mathbf{z}|c) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_c, \boldsymbol{\Phi}_c), \tag{1}$$

where $\boldsymbol{\mu}_c$ is the mean of the latent image $\mathbf{z}$, and $\boldsymbol{\Phi}_c$ is a diagonal covariance matrix that specifies the variability of each pixel in the latent image. The variability $\boldsymbol{\Phi}_c$ is necessary to capture various causes not captured by the variability in scene class and transformation, eg. slight blemishes in appearance or changes in lighting. We do not model the full covariance matrix as there is never enough data to estimate it form the data. It is possible, however to use a subspace modeling technique to capture some correlations in this matrix. The observable image is modeled as a cropped region of the latent scene image. Before cropping, the latent image undergoes a transformation, composed of a zoom, rotation and translation. The motivation for this is that out of the global camera motion types, zoom and pan are the most frequent, while rotations are fairly rare. Leaving out more complex motion, such as the ones produced by perspective effects, several dominant motion vector fields, nonuniform motion, etc., speeds up the algorithm (real time in our implementation), but makes the above defined variance maps a crucial part of the model, as they can capture the extra variability, although in a cruder manner. In addition, the nonuniform variance map has to capture some other causes of variability we left out, such as small illumination changes, variable contrast, etc.

The probability density of the observable vector of pixel values $\mathbf{x}$ for the image corresponding to the zoom $\mathbf{Z}$, translation $\mathbf{T}$, rotation $\mathbf{R}$, latent image $\mathbf{z}$ and fixed cropping transform $\mathbf{W}$ is

$$p(\mathbf{x}|\mathbf{T}, \mathbf{Z}, \mathbf{R}, \mathbf{z}) = \delta(\mathbf{x} - \mathbf{WTZRz}) \tag{2}$$

where $\mathbf{T}$, $\mathbf{R}$ and $\mathbf{Z}$ come from a finite set of possible transformations. Similar affine generative model in conjunction with Bayesian inference was proposed in [13]. We consider only a few different levels of zoom and rotation. The computational burden of searching over all integer translations

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

is relieved by the use of Fast Fourier Transform (FFT) for performing computations in the Fourier domain (Sec. 4).

While we can view this model as the model where the composition $\mathbf{WTZR}$ is treated as a novel transformation, it is an imperative to keep these transformations separate in order to derive an efficient inference algorithm based on the FFTs, which is several orders of magnitude faster than the algorithm based on testing all possible transformations jointly.

The joint likelihood of a single video frame $\mathbf{x}$ and latent image $\mathbf{z}$, given $c$ and $\mathbf{T}$ is

$$p(\mathbf{x}, \mathbf{z}|c, \mathbf{T}, \mathbf{Z}, \mathbf{R}) = \delta(\mathbf{x} - \mathbf{WTZRz})\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_c, \boldsymbol{\Phi}_c) \quad (3)$$

Note that the distribution over $\mathbf{z}$ can be integrated out in the closed form

$$p(\mathbf{x}|c,\mathbf{T},\mathbf{Z},\mathbf{R}) = \mathcal{N}(\mathbf{x}; \mathbf{WTZ}\boldsymbol{\mu}_c \mathbf{WTZ}\boldsymbol{\Phi}_c\mathbf{R}'\mathbf{Z}'\mathbf{T}'\mathbf{W}') \quad (4)$$

Under the assumption that each frame is independently generated in this fashion, the joint distribution over all variables is

$$p(\{\mathbf{x}, c_t, \mathbf{R}_t, \mathbf{Z}_t, \mathbf{T}_t\}_{t=1}^T) = \prod_t p(\mathbf{x}_t|c_t, \mathbf{R}_t, \mathbf{Z}_t, \mathbf{T}_t) \; p(c_t)p(\mathbf{T}_t, \mathbf{Z}_t) \quad (5)$$

The model is parameterized by scene means $\boldsymbol{\mu}_c$, pixel variances stored on the diagonal of $\boldsymbol{\Phi}_c$ and scene probabilities $\pi_c = p(c_t = c)$, and as such providing a summary of what is common in the video. The hidden variables $c_t, \mathbf{R}_t, \mathbf{Z}_t, \mathbf{T}_t$, describe the main causes of variability in the video, and as such vary from frame to frame. The prior distribution over $\mathbf{R}, \mathbf{Z}, \mathbf{T}$ is assumed uniform.

## 3. Recursive model estimation

It is possible to derive the EM algorithm in the closed form (Sec. 4) for the proposed model. However, the number of scenes (components in the mixture) is unknown. Also, the exhaustive computation of the posterior probabilities over transformations and classes is intractable. We use the variant of incremental EM algorithm [15, 16] to quickly cluster the frames into the large number of clusters using, at this stage, translation and cropping-invariant model only. We dynamically update the number of classes, by adding a new class whenever the model cannot explain the new data well.

Given that a large number of frames $\mathbf{x}_t$ have been clustered (summarized) in this manner using a mixture model $p(\mathbf{x}) = \sum_c p(\mathbf{x}|c)p(c)$ with $C$ clusters (components), each described by the prior $p(c) = \pi_c$, mean $\boldsymbol{\mu}_c$ and a diagonal covariance matrix $\boldsymbol{\Phi}_c$, we want to estimate another mixture model $p^1$ defined by a smaller number of clusters $S$ with parameters $\pi_s, \boldsymbol{\mu}_s, \boldsymbol{\Phi}_s$ on the same data. We will formally derive the re-estimation algorithm using a Gaussian mixture model as an example, with the understanding that the same derivation is carried out for the more complex models that includes transformations. Assuming that $p$ summarizes the data well, we can replace the real data $\{\mathbf{x}_i\}$ with the similar ("virtual") data $\{\mathbf{y}_i\}, i = 1 \dots N$ randomly generated from the obtained mixture model, and estimate the parameters of the model $p^1$ using the virtual data $\{\mathbf{y}_t\}$. When the number of virtual data ($N$) grows infinitely, the distribution converges in probability to the original data distribution. We can fit the simpler distribution $p^1$ to $\{\mathbf{y}_t\}$ *without* actually generating them, but rather by working only with the expectations under the model $p$. We optimize the expectation of the likelihood of the generated data, $\frac{1}{N}\sum_i \log p^1(\mathbf{y}_i)$ for large $N$, where $\mathbf{y}_i$ is sampled from $p(\mathbf{y})$ (in our example the mixture model with parameters $\{\pi_c, \boldsymbol{\mu}_c, \boldsymbol{\Phi}_c\}$).

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^N \log p^1(\mathbf{y}_i) \to E[\log p^1(\mathbf{y})] = \\
&= \int_{\mathbf{y}} p(\mathbf{y}) \log p^1(\mathbf{y}) = \int_{\mathbf{y}} \left[\sum_c p(\mathbf{y}|c)p(c)\right] \log p^1(\mathbf{y}) \\
&= \sum_c p(c) \int_{\mathbf{y}} p(\mathbf{y}|c) \log p^1(\mathbf{y}) \\
&\geq \sum_c p(c) \int_{\mathbf{y}} p(\mathbf{y}|c) \sum_s q_c(s) \log \frac{p^1(\mathbf{y}|s)p^1(s)}{q_c(s)} = -EF, \quad (6)
\end{aligned}$$

where the inequality follows by the same convexity argument as in the case of the standard free energy [15]. Such reparametrized model $p^1$ can be recursively reparametrized, giving the hierarchy of models of the decreasing complexity. By doing this we resort to the original data exactly once and avoid costly re-processing of hundreds of thousands of video frames. The new bound on the free energy $EF$ would be tight if $q_c(s)$ were exactly equal to the posterior, i.e., $q_c(s) = p^1(s|\mathbf{y})$. However, we assume that the posterior is the same for all $\mathbf{y}$ once the class $c$ is chosen, and we emphasize this with the notation $q_c(s)$. Under this assumption the bound further simplifies into

$$\begin{aligned}
-EF &= \sum_c p(c) \Bigg\{ \sum_s q_c(s) \left[ \int_{\mathbf{y}} p(\mathbf{y}|c) \log p^1(\mathbf{y}|s) \right] + \\
&\qquad \sum_s q_c(s)[\log p^1(s) - \log q_c(s)] \Bigg\} \\
&= \sum_c p(c) \sum_s q_c(s) \Bigg[ -\frac{1}{2}(\boldsymbol{\mu}_s - \boldsymbol{\mu}_c)^T \boldsymbol{\Phi}_s^{-1} (\boldsymbol{\mu}_s - \boldsymbol{\mu}_c) \\
&\qquad -\frac{1}{2} tr(\boldsymbol{\Phi}_s^{-1}\boldsymbol{\Phi}_c) - \frac{1}{2}\log|2\pi\boldsymbol{\Phi}_s| \Bigg] \\
&\qquad + \sum_c p(c) \sum_s q_c(s)[\log p^1(s) - \log q_c(s)] \quad (7)
\end{aligned}$$

Minimizing the free energy under the usual constraints, e.g., $\sum_s q_c(s) = 1$ yields an iteration of an EM algorithm that reparameterizes the model, e.g., for the plain mixture model,

$$q_c(s) \propto p^1(s) e^{-\frac{1}{2}(\boldsymbol{\mu}_s - \boldsymbol{\mu}_c)^T \boldsymbol{\Phi}_s^{-1} (\boldsymbol{\mu}_s - \boldsymbol{\mu}_c) - \frac{1}{2} tr(\boldsymbol{\Phi}_s^{-1} \boldsymbol{\Phi}_c) - \frac{1}{2} \log |2\pi \boldsymbol{\Phi}_s|} \tag{8}$$

$$\begin{aligned} \boldsymbol{\mu}_s &= \frac{\sum_c p(c) q_c(s) \boldsymbol{\mu}_c}{\sum_c p(c) q_c(s)} \\ \boldsymbol{\Phi}_s &= \frac{\sum_c p(c) q_c(s) [(\boldsymbol{\mu}_s - \boldsymbol{\mu}_c)(\boldsymbol{\mu}_s - \boldsymbol{\mu}_c)^T + \boldsymbol{\Phi}_c]}{\sum_c p(c) q_c(s)} \\ \pi_s &= p^1(s) = \frac{\sum_c p(c) q_c(s)}{\sum_c p(c)} = \sum_c p(c) q_c(s) \end{aligned} \tag{9}$$

Similar reparametrization algorithm was intuitively proposed in [1] for data clustering in the presence of uncertainties. The idea of recursive density estimation is reminiscent of [2]. The EM algorithm above will converge to a local maximum and the quality of the results will depend on the validity of the assumption that the posterior $q(s)$ is shared among all virtual data samples from the same class $c$. When model $p$ captures the original data with lots of narrow models $p(\mathbf{y}|c)$, and $S \ll C$, the approximation is reasonable and reduces the computation by a factor of $T/C$ in comparison with retraining directly on the original data. The result of recursive model estimation is a hierarchy of models which can be elegantly presented to the user through an appropriate user interface shown in the video submission. Figure 1 (right) illustrates recursive clustering of three distributions into a single hyper-cluster, using both translation and scale invariant clustering.

### 3.1. Computing the fast approximate KL divergence between two mixtures

The optimization in Eq.(6) can be alternatively seen as the minimization of the KL divergence between distributions $p$ and $p^1$. Thus, we can use the bound on the variational free energy for the re-estimation problem to obtain tight upper bound on the KL divergence between two mixture of Gaussians (MoGs) – a problem not tractable in the closed form. Recently, efficient and accurate computation of the KL divergence between the mixtures has attracted a lot of attention [17, 18].

As the ground truth for the computation of the KL divergence, we will use Monte Carlo simulation with large number of particles as

$$KL(f||g) = \int f \log \frac{f}{g} \approx \frac{1}{n} \sum_{t=1}^{n} \log \frac{f(x_t)}{g(x_t)} \tag{10}$$

While this method is asymptotically exact, it is painfully slow. In [17] authors proposed a couple of approximations on KL divergence based on counting the influence only of nearest components in two mixtures ("weak interactions"). They demonstrated that their approximation is better than previous one published in [18]. The conclusion of their work is that KL divergence based on unscented transformation [19] (also known as the "quadratic approximation") gives excellent results, with the slight computational overhead. This method is based on the approximate computation of the expectation of some function $h$ under $d$ dimensional Gaussian $f$ with the mean $\mu$ and covariance matrix $\Sigma$ as

$$\int f(x) h(x) dx \approx \frac{1}{2d} \sum_{k=1}^{2d} h(x_k) \tag{11}$$

where the set of $2d$ "sigma points" $x_k$is defined as

$$\begin{aligned} x_k &= \mu + (\sqrt{d\Sigma})_k k = 1, \ldots, d \\ x_{d+k} &= \mu - (\sqrt{d\Sigma})_k k = 1, \ldots, d \end{aligned} \tag{12}$$

We will use this method as the current art to compare against the variational method.

Given two mixtures $p$ and $p^1$ the KL divergence can be separated into two terms

$$\begin{aligned} KL(p, p^1) = H(p) - \int_y p(y) \log p^1(y) = \\ \int_y p(y) \log p(y) - \int_y p(y) \log p^1(y) \end{aligned} \tag{13}$$

We note that optimization we performed in Eq.(6) is the variational maximization of the lower bound of $\int_y p(y) p^1(y)$. By substituting the $S \times C$ matrix $q$ (readily computed in Eq.(8)) into Eq.(7) the *upper* bound for $-\int_y p(y) p^1(y)$ follows. In the same manner, the *lower* bound on entropy $H(p)$ of the Gaussian mixture $p$ with parameters $(\pi_c, \boldsymbol{\mu}_c, \boldsymbol{\Phi}_c)$ can be approximated as

$$\sum_c \{\pi_c (-\frac{1}{2} \log \det(2\pi \boldsymbol{\Phi}_c)) + \log(\pi_c)\} < H(p) \tag{14}$$

The summation of the lower and upper bound of two terms in the KL divergence need not lead to the unambiguous conclusion on the nature of the approximation. Empirically, we found that the entropy term negligibly contributes to the KL divergence.

## 4. Inference of classes and transformations. Learning the scenes in the model

**Inference (posterior optimization).** In this section we will omit the rotation $\mathbf{R}$ by treating it as an identity transformation in order to keep the derivations simple. For our

model it is possible to derive exact EM algorithm that optimizes the free energy [15] of the form

$$F = \sum_t \sum_{c_t, \mathbf{Z}_t, \mathbf{T}_t} q(c_t, \mathbf{Z}_t, \mathbf{T}_t) \log \frac{p(\mathbf{x}_t | c_t, \mathbf{Z}_t, \mathbf{T}_t) \pi_{c_t}}{q(c_t, \mathbf{Z}_t, \mathbf{T}_t)} \tag{15}$$

For given parameters, we can optimize the free energy with respect to the posterior $q$. We express the posterior as $q(c_t, \mathbf{Z}_t) q(\mathbf{T}_t | c_t, \mathbf{Z}_t)$ and optimize $F$ under the normalization constraints $\sum_{c_t, \mathbf{Z}_t} q(c_t, \mathbf{Z}_t) = 1$ and $\sum_{\mathbf{T}_t} q(\mathbf{T}_t | c_t, \mathbf{Z}_t) = 1$, which results in the same result as applying the Bayes rule,

$$q(\mathbf{T}_t | c_t, \mathbf{Z}_t) \propto p(\mathbf{x}_t | c_t, \mathbf{Z}_t, \mathbf{T}_t), q(c_t, \mathbf{Z}_t) \propto p(c_t) e^{-q(c_t | \mathbf{Z}_t, \mathbf{T}_t) \log p(\mathbf{x}_t | c_t, \mathbf{Z}_t, \mathbf{T}_t)} \tag{16}$$

**Parameter optimization.** Finding the derivatives of $F$ with respect to the cluster mean $c_t = k$ we get

$$\sum_{t=1}^{T} \sum_{\{\mathbf{T}_t, c_t\}} q(\{c_t, \mathbf{T}_t\}) (\mathbf{W}\mathbf{T}_t\mathbf{Z}_t)' (\mathbf{W}\mathbf{T}_t\mathbf{Z}_t\mathbf{\Phi}_k\mathbf{Z}_t'\mathbf{T}_t'\mathbf{W}')^{-1} \times (\mathbf{x}_t - \mathbf{W}\mathbf{T}_t\mathbf{Z}_t\boldsymbol{\mu}_k) = 0 \tag{17}$$

It can be shown that

$$\mathbf{T}'\mathbf{W}'\mathbf{W}\mathbf{T}\mathbf{Z}\mathbf{\Phi}_c^{-1}\mathbf{Z}'\mathbf{T}'\mathbf{W}'\mathbf{x}_t = \mathbf{Z}\mathbf{\Phi}_c^{-1}\mathbf{Z}'\mathrm{diag}(\mathbf{T}'\mathbf{X}_t)$$
$$\mathbf{T}'\mathbf{W}'\mathbf{W}\mathbf{T}\mathbf{Z}\mathbf{\Phi}_c^{-1}\mathbf{Z}'\mathbf{T}'\mathbf{W}'\mathbf{W}\mathbf{T} = \mathbf{Z}\mathbf{\Phi}_c^{-1}\mathbf{Z}'\mathrm{diag}(\mathbf{T}'\mathbf{m}) \tag{18}$$

where $\mathbf{m} \triangleq diag(\mathbf{W}'\mathbf{W})$, is the binary mask that shows the position of observable image within latent image (upper left corner), and where $\mathbf{X} \triangleq \mathbf{W}'\mathbf{x}$ is frame $\mathbf{x}$ zero padded to the resolution of the latent image. Thus, assuming that the zoom has small effect on inverse variances, i.e., $(\mathbf{Z}\mathbf{\Phi}\mathbf{Z}')^{-1} \approx (\mathbf{Z}\mathbf{\Phi}^{-1}\mathbf{Z}')$ [1] we obtain simple update rules, e.g. for $\tilde{\boldsymbol{\mu}}_k$

$$\frac{\sum_{t=1}^{T} \sum_{\mathbf{Z}_t} q(c_t = k, \mathbf{Z}_t) \mathbf{Z}_t^{-1} \sum_{\mathbf{T}_t} q(\mathbf{T}_t | c_t = k, \mathbf{Z}_t) (\mathbf{T}'\mathbf{X}_t)}{\sum_{t=1}^{T} \sum_{\mathbf{Z}_t} q(c_t = k, \mathbf{Z}_t) \mathbf{Z}_t^{-1} \sum_{\mathbf{T}_t} q(\mathbf{T}_t | c_t = k, \mathbf{Z}_t) (\mathbf{T}'\mathbf{m})} \tag{19}$$

where $\mathbf{Z}^{-1}$ is the pseudoinverse of matrix $\mathbf{Z}$, or the inverse zoom. In a similar fashion, we obtain the derivatives of $F$ with respect to other two types of model parameters $\mathbf{\Phi}_k$ and $\pi_k$, and derive the update equations. It may seem at first that zero padding of the original frame to the size of the latent image constitutes the unjustified manipulation of the data. But, taking into account that zero is neutral element for the summation of the sufficient statistics in (19), it is actually the mathematical convenience to treat all variables as being of the same dimensionality (resolution). The intuition behind (19) is that the mean latent (panoramic) image is the weighted sum of the properly shifted and scaled frames, normalized with the "counter" that keeps track how many times each pixel was visited.

**Speeding up inference and parameter optimization using FFTs.** Inference and update equations (16) and (17) involve either testing all possible transformations or summations over all possible transformations $\mathbf{T}$. If all possible integer shifts are considered (which is desirable since one can handle arbitrary interframe shifts), then these operations can be efficiently performed in the Fourier domain by identifying them as either convolutions or correlations. For example, (19) can be efficiently computed using two dimensional FFT [11] as

$$\sum_{\mathbf{T}} q(\mathbf{T}|c, Z)(\mathbf{T}'\mathbf{X}) = \mathrm{IFFT2}[\mathrm{conj}(\mathrm{FFT2}(q(\mathbf{T}))) \circ (\mathrm{FFT2}(\mathbf{X}))] \tag{20}$$

where $\circ$ denotes point wise multiplication, and "conj" denotes complex conjugate. This is done for each combination of the class and scale variables, and a similar convolution of the transformation posterior is also applied to the mask $\mathbf{m}$. Similarly, FFTs are used for inference to compute Mahalanobis distance in (4). This reduces computational complexity of both inference and parameter update from $N^2$ to $NlogN$ ($N$ – number of pixels), allows us to analyze video frames of higher resolution, and demonstrate the benefits of keeping translation variable $\mathbf{T}$ and separate from cropping $\mathbf{W}$ and zoom $\mathbf{Z}$ in the model. The computation is still proportional to the number of classes, as well as the number of zoom levels we search and sum over in the E and M steps, but the number of these configuration is typically much smaller than the number of possible shifts in the image.

**On-line learning.** The batch EM learning suffers from two drawbacks: the need to preset the number of classes C, and the need to iterate. The structure of realistic video allows development of more efficient algorithms. Frames in video typically come in bursts of a single class which means that the algorithm does not need to test all classes against all frames all the time. We use an on-line variant of the EM algorithm with the incremental estimation of sufficient statistics [15, 16]. The reestimation update equations (Eq. (9)) are reformulated in the same manner.

In order to dynamically learn the number of scenes in the on-line EM algorithm, we introduce the threshold on the log-likelihood such that whenever the log-likelihood falls under the threshold a new scene is introduced. The sensitivity due to the choice of the threshold is overcome by setting high threshold that guarantees the likelihood of the dataset remains high, but which may lead to over-clustering. The problem of merging large number of clusters – still much smaller than the number of frames – is addressed in Section

[1] To avoid degenerate solutions, the likelihood is scaled with the number of pixel increase that the zoom causes.

3. When the number of clusters is reduced to the order of 100, we apply the full learning using the batch EM algorithm with number of clusters determined by the MDL criterion. Taking into account that camera/object shifts are by far the most common transformations in the video, another speed-up is to perform translation-only invariant clustering in the first pass (by setting $\mathbf{Z}$,$\mathbf{R}$ to identity matrices). This approach reduces most of the variability in the data with little computational cost. The overall performance of our clustering is 35 frames per second on 3GHz PC.

## 5. Experiments

**Computing the KL divergence between two mixtures.** We tested the performance of computing the upper bound on the KL divergence between two Gaussians in the setup similar to [17]. The mean of each Gaussian in the five dimensional space is randomly chosen according to $\mathcal{N}(0,1)$, whereas the covariance matrix is sampled from Wishart distribution. In order to avoid almost singular covariance matrices that may arise in the random sampling, condition number is set to be at least 20. Covariance matrix is premultiplied with a small number $\epsilon$ that accounts for the width of the Gaussians. Higher values of $\epsilon$ correspond to higher overlap between the blobs. We tested four methods: Monte Carlo simulation with 10,000 particles (MC10000) assumed to be the golden standard; Monte Carlo simulation with 100 particles (MC100); method based on unscented transform; and, our method (variational). We repeated each simulation 100 times and averaged the results. We present the results in Table 1 and Fig. 2. The best results were obtained via the unscented approximation, followed by our method, and the MC100. The bottom row of Table 1 indicates the relative processing time needed to compute KL divergence for each method. While approximate, our method is by far the fastest of the proposed methods. In Fig. 2 we illustrate the accuracy of the computation of the KL divergence for different values of the parameter $\epsilon$ and the computational complexity of our method for different dimensionality of the space. As demonstrated, our method especially scales well in the the high dimensional space.

**Video clustering and navigation.** We tested our system on an 18 minutes long home video. In the first pass of on-line learning, the video is summarized in 290 clusters, many of them repeating. We reestimate this model until we end up with roughly three dozen of classes. In all but the first pass we search over all configurations of the zoom, rotation and class variables. The complexity of the learning drops as we go higher and higher in the hierarchy (due to the smaller number of clusters), and so we do not need to be careful about the exact selection of the sequence of thresholds or numbers of classes - we simply train a large number of models, as the user can choose any one of them quickly at the browsing time.

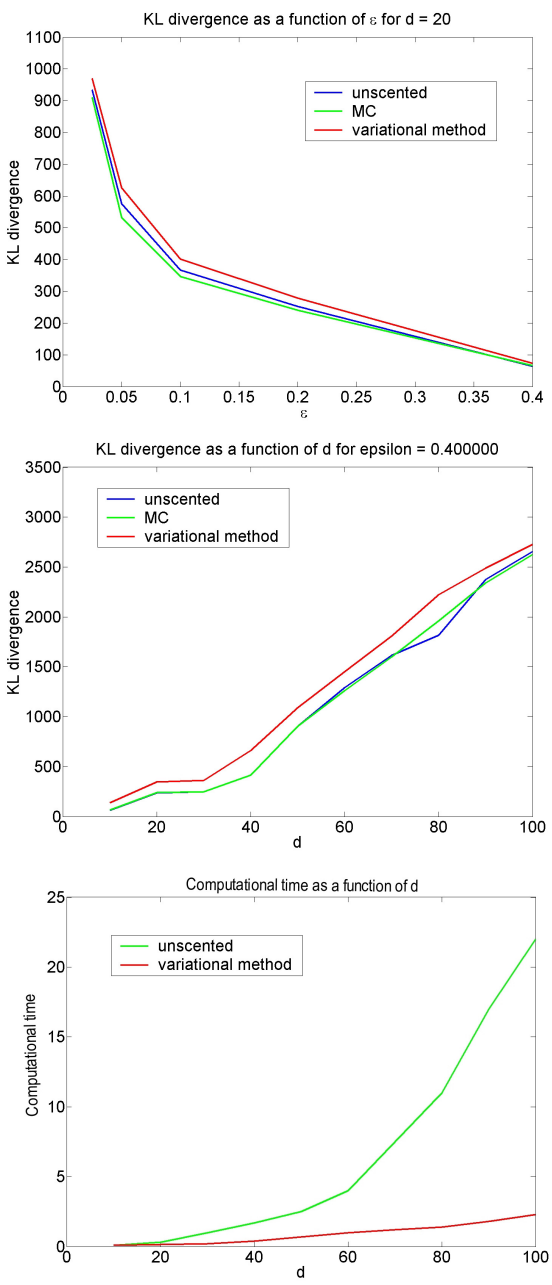


Figure 2. Computing KL divergence between two mixtures (12 and 8 components). Comparison of Monte-Carlo simulations, unscented transform-based method and the method in this work. Top: KL divergence is calculated between two MoGs in the 20-dimensional space. Horizontal axis depicts the regularization parameter epsilon. Our method, as anticipated, is upper bound to the true KL divergence. Middle: KL divergence as a function of the dimensionality of mixtures. Regularization parameter is fixed to 0.4. Bottom: Computational time in seconds for our method and unscented-based method as a function of space dimensionality. Monte Carlo simulations are too slow to scale.

Fig. 3 (top) illustrates the time line and scene partitioning using shot detectors and our method. On the left we hand-

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

| $\epsilon$ | unscented | MC10000 | variational | MC100 |
|---|---|---|---|---|
| 0.025 | 92.97 | 92.75 | 96.15 | 127.75 |
| 0.050 | 92.86 | 92.29 | 104.84 | 95.58 |
| 0.100 | 20.74 | 20.98 | 25.21 | 17.76 |
| 0.200 | 18.39 | 18.72 | 20.75 | 19.98 |
| 0.400 | 49.24 | 45.32 | 55.24 | 34.41 |
| Time | 0.125 | 5.219 | 0.016 | 0.062 |

Table 1. Values of the KL-divergence for four different methods and different values of regularization parameter $\epsilon$. MC10000 is taken as the golden standard. Variational method is a reasonable approximation and it is by far the fastest. As expected, for larger $\epsilon$ all methods make large errors.

label ground truth (also indicating the repeating scenes). In the middle we show shot detection results using commercial shot detector largely on the color histogram (from Microsoft MovieMaker). On the right we show the clustering results using our method, and indicate the cases where our method over-segmented. The clustering fails in the cases when scene variability is not explained by the model. Some of the issues can be tackled by using the higher number of scale levels in the model and increasing the scene size with respect to the frame size.

But, the real benefit of this approach is in the novel video navigation and browsing tool. Supplemental video material (Fig. 3 bottom and http://www.ifp.uiuc.edu/~nemanja/Video1.wmv) demonstrates the usefulness of this method. Cluster means are visualized as the thumbnail images that represent the index into the video. For each pixel in each frame in the video there is a mapping into the thumbnail image. The user browses the video by moving the mouse pointer over the active panel. Instantly, frames within the cluster that are located in the proximity of the cursor are retrieved and marked in green on the time-line at the bottom of the interface. The user can further double-click at each thumbnail images and it will decompose into the "child" clusters that were merged together in the re-estimation procedure. The browsing may then be continued seamlessly at the different level. Informally tested on 165 users, the system proved to be very useful for the users to rapidly grasp the content of the video not seen before.

## 6. Conclusions

In this work we presented a generative model for video that proved useful for unsupervised video clustering. We specifically addressed the problem of intractability of naive learning for large scale problems by introducing the number of algorithmic techniques for rapid learning and inference. Our video analysis requires no hand-set parameters. The burden of selecting the optimal number of scenes – itself highly subjective task – is shifted to the user to choose among the hierarchy of the models. We believe that this model and the accompanying intuitive user-interface will prove useful for quick and seamless video retrieval. Additionally, we will further explore the benefits of the proposed method for the rapid computation of the KL divergence, especially in the high dimensional space and for the massive data-sets.

## Acknowledgments

This work was supported in part by Advanced Research and Development Activities (ARDA) under Contract MDA904-03-C-1787.

## References

[1] T.F. Cootes, C.J.Taylor. "Statistical Models of Appearance for Computer Vision", *Technical Report University of Manchester*, 1998.

[2] A. Gray, A. Moore. "Rapid Evaluation of Multiple Density Models", In *Aritificial Intelligence & Statistics*, 2003.

[3] D. Mumford. "Pattern theory: a unifying perspective", In *Perception as Bayesian Inference*, Cambridge University Press, 1996.

[4] R. Jones, D. DeMenthon, and D. Doermann, "Building mosaics from video using MPEG motion vectors" In *ACM Multimedia*, 1999.

[5] P. Anandan, M. Irani, M. Kumar, and J. Bergen. "Video as an image data source: Efficient representations and applications", In *Proceedings of IEEE ICIP*, pp. 318–321, 1995.

[6] A. Aner, J. R. Kender. "Video summaries through mosaic-based shot and scene clustering", In *ECCV*, 2002.

[7] M. Irani, P. Anandan, S. Hsu. "Mosaic based representations of video sequences and their applications", In *ICCV*, pages 605–611, June 1995.

[8] M. Brown, D. Lowe, "Recognising Panoramas", *ICCV03*.

[9] M. Yeung, B. Yeo, B. Liu. "Segmentation of video by clustering and graph analysis", In *CVIU*, 71:1, July 1998.

[10] A. Girgensohn, J. Boreczky. "Time-constrained keyframe selection technique", In *Proc. IEEE Multimedia Computing and Systems*, 1999.

[11] B.J. Frey and N. Jojic. "Fast, large-scale transformation-invariant clustering", In *NIPS 14*, Cambridge, MA: MIT Press, 2002.

[12] C. M. Bishop. "Variational learning in graphical models and neural networks" In *Proceedings 8th ICANN*, 1998.

[13] J. Winn, A. Blake. "Generative Affine Localisation and Tracking" In *NIPS*, 2003.

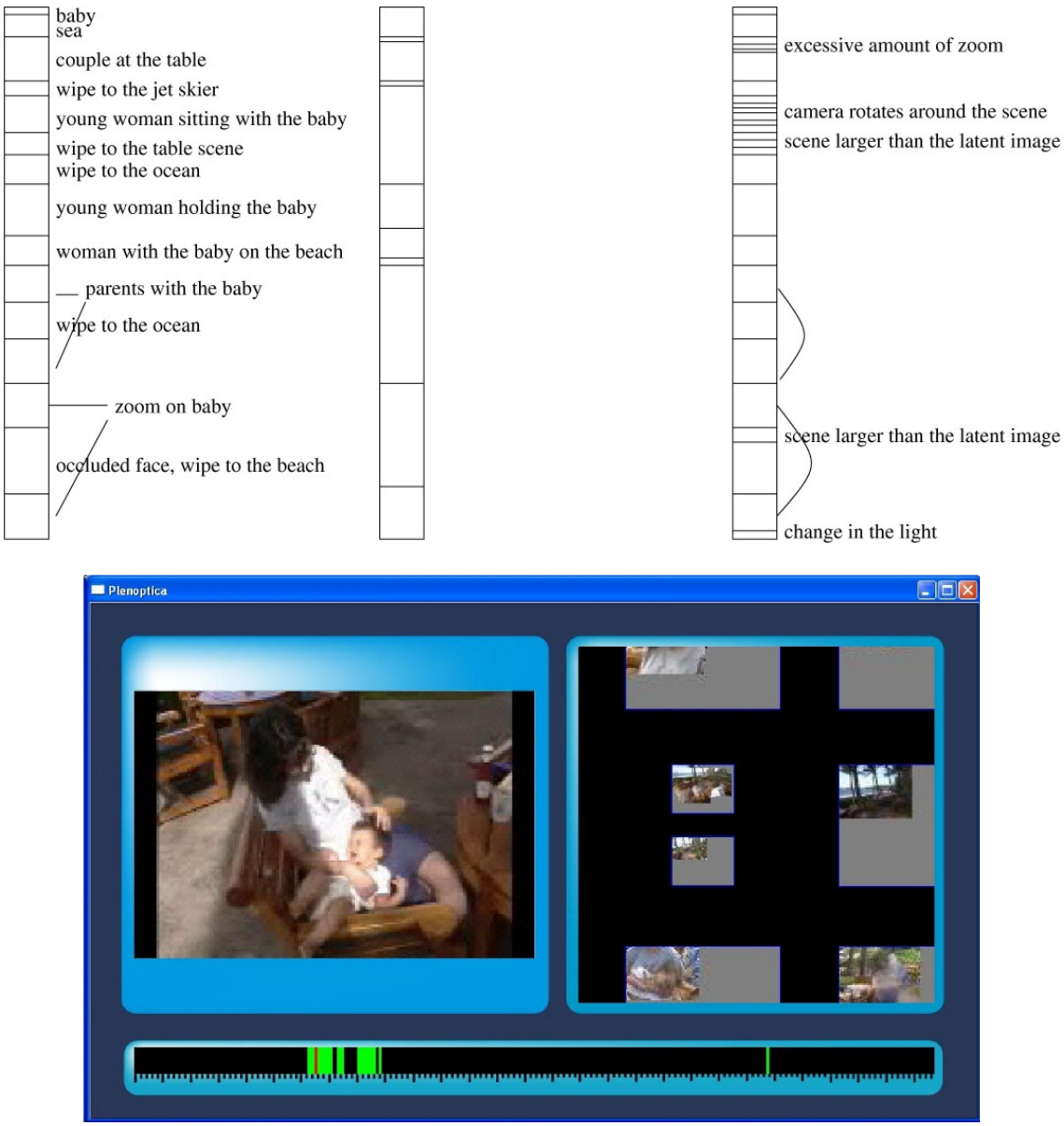


Figure 3. Top: Comparison of ground truth, shot detection and our method. The time line with ground truth segmentation (left) illustrates positions and labels of 15 scenes in the video. Some of the scenes are repeating. Shot detection algorithm based on color histogram (center) is sensitive to sudden swipes and changes in light. In the same time, scene changes are undetected if there are due to slow swipes of the camera. Our approach (right) correctly detects scenes in most cases. We label the situations when it over-segments. Bottom: A snapshot of video navigation and browsing tool. See the demo at http://www.ifp.uiuc.edu/˜nemanja/Video1.wmv

[14] B.J. Frey, N. Jojic. Transformation-invariant clustering using the EM algorithm. *PAMI*, 25(1), Jan 2003.

[15] R. M. Neal, G. E. Hinton. "A new view of the EM algorithm that justifies incremental, sparse and other variants", In *Learning in Graphical Models*, page 355-368. Norwell MA: Kluwer Academic Publishers, 1998.

[16] S. J. Nowlan. "Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures", *Ph.D. thesis*, Carnegie Mellon University, 1991.

[17] J. Goldberger, S. Gordon, H. Greenspan. "Efficient Image Similarity Measure based on Approximations of KL-Divergence Between Two Gaussian Mixtures", *ICCV03* .

[18] N. Vasconcelos. "On the complexity of probabilistic image retrieval", In *ICCV* , 2001.

[19] S. Julier, J. K. Uhlmann. "A general method for approximating non-linear transformations of probability distributions", *Technical report,* RRG, University of Oxford, 1996.