# Joint design of data analysis algorithms and user interface for video applications

**Nebojsa Jojic**
Microsoft Research

**Sumit Basu**
Microsoft Research

**Nemanja Petrovic**
University of Illinois

**Brendan Frey**
University of Toronto

**Thomas Huang**
University of Illinois

## Abstract

The graphical modeling paradigm provides a way of representing data through hidden causes of variability which can be estimated from the data in an unsupervised manner. Recently a lot of research has been dedicated to finding efficient inference and learning engines for graphical models in general, as well as to finding various ways of using graphical models to perform recognition, classification, segmentation, and tracking tasks in video applications. Little research, however, has focused on another advantage of a graphical model - by discovering the structural elements in the data, it renders the data much easier to browse, manipulate, or interact with. In this paper, we present several ideas on how the user interface and the data analysis tools can be designed jointly starting from an appropriate data representation scheme and a generative model based on it. We base our approach on three basic principles:

- Compatibility of the graphical model's structure with our own perception of the world
- Simplicity in representation, leading to more efficient inference
- Providing intuitive interactivity on the level of hidden causes of variability

## 1 The graphical model should reflect the structure of the world

We think about the world in terms of scenes, objects, and motion. While these are often primarily visual objects, we associate with them other sensory stimuli, such as sounds and smells. The basic components of a scene sometimes interact in a structured way to form a distinct activity, or even a story about the objects. In our memory of events, the time axis is warped and sometimes stretched but most of the time compressed, and the key elements are the objects and their relationships. At the very basic level, our distant memories are mostly of scenes and short activities that give a sense of our past experience, rather than detailed and precise accounts of what had really happened.

Video is a recording of the world, which even though it can contain very rich visual and auditory information, is provided in a completely different form - a set of pixels, waveforms, and possibly short term motion vectors as part of the compression scheme. In order to have an intuitive interface with this data, we have to transform it into the form closer to what we store in our memory.

Graphical modeling approach to data analysis is compatible with this goal. We can easily describe the data formation process as a combining multiple objects and scene background to form a video frame. Such a model would have as hidden variables for each frame the positions and orientations of the objects, ordering of objects that defines the occlusions in the scene, and illumination characteristics of the scene. At the higher level, additional hidden variables could control how the higher level variables change through time, thus capturing the motion or illumination patterns. As parameters that apply to all frames, we can have descriptors of all objects that appear in all frames, meta properties of illumination sources, priors on motion patterns, etc.

In principle, given lots of data, e.g. an hour of vacation video, completely unsupervised parameter estimation (learning) would result in a summary of all objects, likely motion patterns, etc., while the inference result for each frame would consist of the information on presence/absence, position and orientation of each object, brightness of the frame, etc. A number of inference strategies have been developed in the machine learning community to help with this task.

## 2 The model should be simple enough to reduce the tractability issue in inference

Given the state of the computer graphics today, we can be tempted to use a very detailed, almost perfect model of the world with hidden variables that interact in a very nonlinear manner. For instance, we may be tempted to model the world in terms of the full 3-D structure of each object, and generation of each frame based on ray-tracing. There has even been some examples of amazingly successful fitting of detailed 3-D models to image data in specialized applications, e.g. [1]. However, in order to start only with the model structure, and no other prior knowledge on the shape, size and position of the objects, it is important to simplify the model as much as possible to allow for robust unsupervised learning.

Thus representations that keep the basic notion of the world structure and for which efficient inference and learning techniques can be developed are preferable. For instance, in Fig. 1, we illustrate a graphical model that treats objects as bitmaps with per-pixel defined noise, that are combined with the shape defined by a transparency map [2]. The transformation variable can take on a finite number of values, each defining at least a different translation in the image. The objects are generated in a number of layers, and the final image is the composition of these layers based on the transparency maps. In addition to the number of discrete variables that make the number of configurations large, the only other nonlinearity in the model is the sprite composition equation,

$$\begin{aligned} \mathbf{x} = \mathbf{T}_L\mathbf{m}_L * \mathbf{T}_L\mathbf{s}_L + \mathbf{T}_L\bar{\mathbf{m}}_L * \\ (\mathbf{T}_{L-1}\mathbf{m}_{L-1} * \mathbf{T}_{L-1}\mathbf{s}_{L-1} + \mathbf{T}_{L-1}\bar{\mathbf{m}}_{L-1} * \\ (\mathbf{T}_{L-2}\mathbf{m}_{L-2} * \mathbf{T}_{L-2}\mathbf{s}_{L-2} + \mathbf{T}_{L-2}\bar{\mathbf{m}}_{L-2} * \\ \ldots \\ (\mathbf{T}_1\mathbf{m}_1 * \mathbf{T}_1\mathbf{s}_1 + \mathbf{T}_1\bar{\mathbf{m}}_1 * \mathbf{s}_0))) \\ + noise. \end{aligned} \tag{1}$$

We have shown that due to the particular way they interact with the sprite appearances, the translation hidden variables can be efficiently dealt with in the FFT domain both in the E and the M step of the learning algorithm [3]. The product in (1) still renders the exact inference intractable, but it turns out that variational inference that treats the posterior as Gaussian and decoupled, making the process tractable and yet not too far from the expected result of exact inference.

## 3 The interaction should be based on visual manipulation of the inferred hidden variables

Having defined the model structure, we can start with an empty model and a video sequence and use an appropriate learning algorithm to fill in the blanks - e.g., the parameters defining average object appearance and shape in the sequence, and the hidden causes of variability in each frame. e.g., the transformation variables and the current segmentation of the objects, defined by the posterior distributions $p(\mathbf{T}_{i,t}|\mathbf{x}_t)$ and $p(\mathbf{s}_{i,t}|\mathbf{x}_t)$.

The interaction with the data can be done directly by interacting with the inferred parameters and hidden variables. For instance, if we are interested in all parts of the video in which a certain object was visible, we can click on the learned bitmap representing this object at the top of the screen. If we want to edit the sequence, changing the motion parameters, for example, we can directly edit the transformation variables and regenerate the sequence. We can also easily remove or insert objects via drag and drop interactions.

This high level of interactivity is enabled by a bidirectional mapping between the hidden causes of variability in the data and the pixels in the video. For each pixel in the video, the posterior distribution over transformation and segmentation masks gives us the information about which class of object it

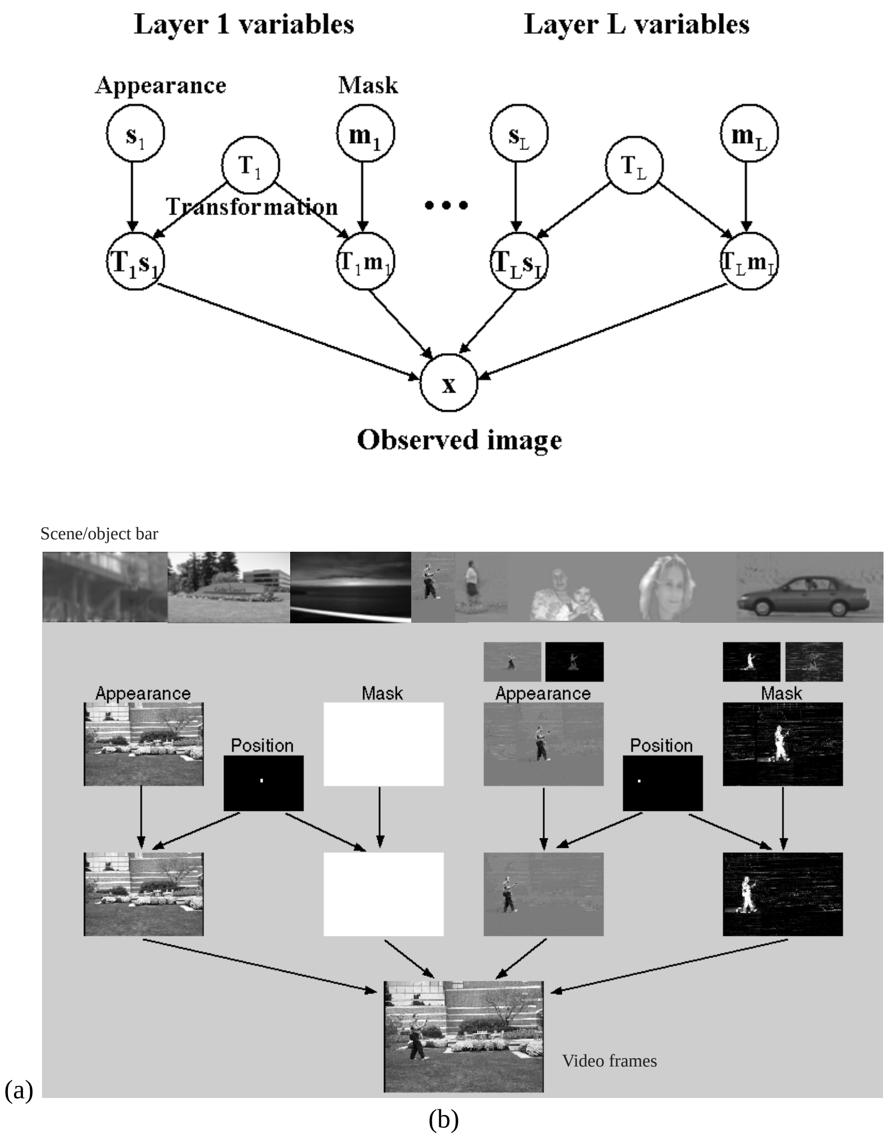


Figure 1: Flexible sprite graphical model (a) as the interface to the data (b). The probability model is mainly defined by the product of the prior distributions over classes and transformations with the Gaussian appearance and mask distributions $p(\mathbf{s}_\ell|c_\ell)$, $p(\mathbf{m}_\ell|c_\ell)$, as well as the Gaussian distribution over the observed frame with the mean equal to the composition of the layers (Eq. 1). As the generative model captures the basic elements of the user's own intuitive representation of the world, it can be revealed to the user for interaction (b). The top bar contains a subset of the learned classes $c$ of objects and scenes, illustrated by their mean appearance (even though each class is defined by means and variances for both appearance and shape). The graph bellow illustrates the hidden variables for a particular frame. The users can select one of the classes $c$ on top to retrieve all frames containing an object, or a combination of objects of interest, and they can manipulate the sprite graph in order to edit the video (stabilize, remove or insert objects, for example).

belongs too as well as to which pixel in this class the observed pixel is aligned to. Vice versa, for each pixel in the learned object bitmap, we know which frames and even which portions of these frames contained a version of it. In some cases, we manipulate the data by going directly for the hidden variable (e.g., to retrieve all frames with a certain object), while in other cases, a more intuitive interaction mode is to point at the objects in the video, e.g., with the goal of stabilizing the motion.

There are two very interesting features of the coalescent design of user interaction and data analysis, which will often affect the design criteria for the graphical model:

**Analysis aids interaction**

- *Interaction by direct manipulation of the variables in the graph.* While a graphical model seem a very technical way of representing the data, once we make sure that the structure follows the structure of the world as we perceive it, the variables in the graph will have the right meaning even to non-technical users, allowing interaction by direct manipulation of the variables in the graph, as discussed above (Fig. 1, Videos 1a-c).
- *Manipulating a visual representation that links the graph to the data.* Even more intuitive way of manipulating the data is to leave some of the variables hidden during the interaction, as they are in the human mind usually downplayed. For example, even though we are aware of the position of an object in the scene, the presence is of much higher importance than the position, and we do not need this variable separately emphasized from the object itself. In Fig. 2 and Video 2, we show an alternative user interface which uses the results of panoramic scene clustering and allows the user to browse by exploring both the scene index and the transformation variable in a more direct way. The panel on the right shows all scenes in the image, and by simply mousing over a particular pixel in any of the scenes, the system retrieves all frames that were not only part of that scene, but actually contained that particular scene element. This interface explores directly the bidirectional mapping between the scene classes and the pixels in the video, implicitely taking care of the transformation variability, which had to be inferred in order to make the browser work.

**Interaction as a remedy for the ambiguities in the analysis**

- *On-line learning based on user interaction.* While an initial representation can be used as a basis for initial interaction with video, the subsequent user's access to the various parts of the structure can be used to correct the inference results where necessary. For instance, in the UI in Fig. 2, even if the video did not initially provide strong enough cue to separate two objects from the background in a particualr scene, if the user tends to select always from two different locations in the panoramic scene, this information can be used to for a prior on object locations in a new round of re-training.
- *Overcomplete representations and user selection.* One issue with simplification in the model is that there arise many ambiguities in terms of the final representation. For instance, the number of scenes or object classes would have to be decided upon in advance, and it will affect the quality of the clustering result. We can turn to the user input in this case, but not simply by requiring the number of classes to be predefined, but rather by offering the user a hierarchy of clustering results to choose from. By simple use of zoom controls, e.g., the scroll button on the mouse, the user selects the level of the hierarchy, using coarser or finer levels to gain context of the scene in the video (Video 3 and Fig. 2 left). For this usage scenario, it turns out that a very efficient hierarchical learning scheme can provide the appropriate metadata for interaction.

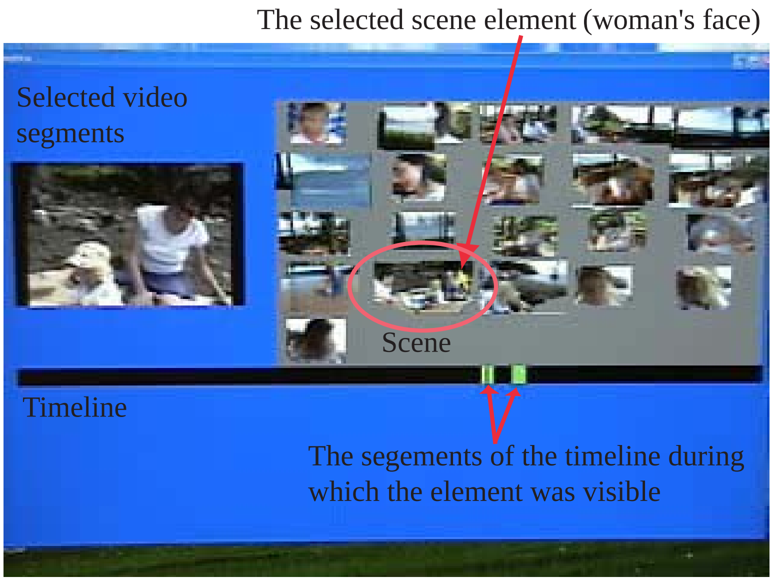
## References

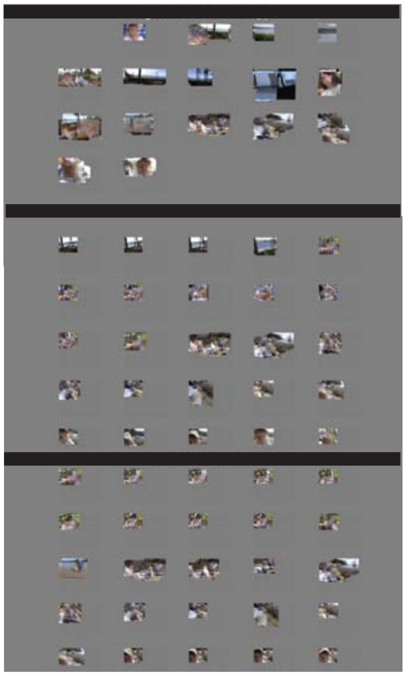
[1] Blanz, V. & Vetter, T. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH*, 1999.

[2] Jojic, N. & Frey, B.J. Flexible sprites. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[3] Frey, B.J. & Jojic, N. Fast, large-scale transformation-invariant clustering. In *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press 2002

[4] Petrovic, N., Jojic, N., Frey, B. & Huang, T. On-line learning of transformed hidden Markov models of video In *Proceedings of the AISTATS*, 2003.


The selected scene element (woman's face)
Selected video segments
Scene
Timeline
The segements of the timeline during which the element was visible



Hierarchy selection: The user holds the cursor over a scene (in the center of the panel in this case), and uses a zoom control to reveal a more detailed local clusters, or coarser, more global representations. We show only three of nine levels.

Figure 2: Video browser that maps the mouse actions into the latent space of classes and transformations. The background panel contains scenes estimated by hierarchical generative model estimation.