# Capturing image structure with probabilistic index maps

**Nebojsa Jojic**
Microsoft Research
www.research.microsoft.com/∼jojic

**Yaron Caspi**
The Hebrew University of Jerusalem
www.cs.huji.ac.il/∼caspiy

## Abstract

One of the major problems in modeling images for vision tasks is that images with very similar structure may locally have completely different appearance, e.g., images taken under different illumination conditions, or the images of pedestrians with different clothing. While there have been many successful attempts to address these problems in application-specific settings, we believe that underlying a large set of problems in vision is a representational deficiency of intensity-derived local measurements that are the basis of most efficient models. We argue that interesting structure in images is better captured when the image is defined as a matrix whose entries are discrete indices to a separate palette of possible intensities, colors or other features, much like the image representation often used to save on storage. In order to model the variability in images, we define an image class not by a single index map, but by a probability distribution over the index maps, which can be automatically estimated from the data, and which we call probabilistic index maps. The existing algorithms can be adapted to work with this representation, as we illustrate in this paper on the example of transformation-invariant clustering and background subtraction. Furthermore, the probabilistic index map representation leads to algorithms with computational costs proportional to either the size of the palette or the log of the size of the palette, making the cost of significantly increased invariance to non-structural changes quite bearable.

## 1 Introduction

An image is typically defined as a matrix whose entries are intensity levels, colors, or features, which are all fairly sensitive to changes in the environment illumination, let alone more interesting changes, such as the change in a persons clothes. We argue that for general image matching tasks, the interesting structure in images is better captured when the image is defined as a matrix whose entries are indices to a separate palette of possible intensities, colors or other features, much like the image representation often used to save on storage. An advantage of this representation is that the palette can be arbitrarily changed without changing the image structure. In order to model the variability in image structure, we define an image class not by a single index map, but by a probability distribution over the index maps, which can be automatically estimated from the data. Under this model, images of a certain class (e.g., head-and-shoulder photographs of people) will have similar index maps which are all likely under this probabilistic model, but their palettes can be completely different. With the cost proportional to the size of the palette, the existing algorithms can be adapted to work with this representation, as we illustrate in this paper on the example of transformation-invariant clustering.

In previous work, a very interesting step in the direction of color-invariance was made by Stauffer et al, who replace the image intensities with a self-similarity measure [4, 5]. They build a large ”co-occurrence matrix” with an entry for every pair of pixels. This statistic is computed from a labeled training set, and as far as we know their technique is only used in supervised algorithms. The major problem is the size of the matrix ($10^5 \times 10^5$ entries for a $256 \times 256$ image). Computational and storage problems have so far limited their experiments to tasks that use small images, e.g., pedestrian detection. Our representation is considerably more efficient and is easily used in unsupervised algorithms. It is also easily combined with other causes of variability in graphical models, e.g., the models developed by Jojic and Frey [1, 2]. However, as our experiments show, our new representation provides much greater color- and feature-invariance, which helped it outperform these appearance-based models in unsupervised transformation-invariant clustering tasks.

## 2 Palette indexing

One efficient representation of an image is as the collection of indices, one index per pixel, that points to a separate table of possible values the pixels can take. This representation is heavily used in image formats, as it drastically reduces the storage requirements. Although the goal is usually storage efficiency, this is achieved by exploring self-similarity in the image, at

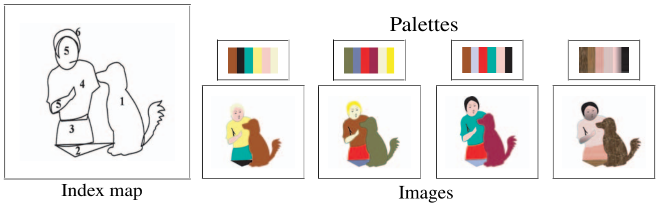


Figure 1: An illustration of the index map as a palette-invariant representation

least at the lowest level. Thus, such a representation is potentially useful in other image processing tasks including image understanding. The index table is typically a color table, or a palette, but there is no reason why it would not contain derived features, such as wavelet coefficients, for example. The palette can be shared across a collection of images, e.g. a video sequence, or a set of objects with similar structure but different colors.

In this section, we discuss possible assumptions on dependencies among variables in such a representation to point out the similarities and differences between standard techniques based on color/feature histograms and our models based on the idea of complete or partial palette invariance.

Consider a collection of $T$ $I \times J$ images $\mathbf{X} = \{X^t\}_{t=1}^T$, where each image is defined on the domain $\{(i,j)|i \in [1..I], j \in [1..J]\}$, and the individual pixels (or other measurements) are indexed by $t$, $i$, and $j$, i.e., $\mathbf{X}^t = \{\mathbf{x}_{ij}^t\}_{(1,1)}^{(I,J)}$. Despite the redundancy, we temporarily associate a separate color (or feature) palette $\mathcal{C}_{i,j}^t$ and an index $s_{ij}^t$ with each pixel. We will later make the images share the index maps but not the palettes (Fig. 1). Each palette $\mathcal{C}$ is a table of $S$ color or feature models, indexed by $s$. For example, $\mathcal{C}(s) = \boldsymbol{\mu}_s$ could be an $[r,g,b]^T$ vector for the $s$th color in the table, as customary. Then, the color of a pixel is $\mathbf{x}_{ij}^t = C(s_{ij}^t)$. The information stored in the palette can be extended to include:

- Extra features (both in $\mathcal{C}$ and $\mathbf{x}$)
  - Wavelet or Gabor coefficients
  - Edge descriptors
  - Texture measurements, etc.
- Description of uncertainty/variability:
  - Color variance parameters, e.g., the covariance matrix describing $E[\mathbf{x}_{ij}^t - \boldsymbol{\mu}_s][\mathbf{x}_{ij}^t - \boldsymbol{\mu}_s]^T$ in a Gaussian distribution
  - Other forms of error distributions, e.g., a hand-tuned robust distribution (a mixture of Gaussian and uniform, for example, which would discount large errors)
  - Hidden causes of variability (mixture variables, subspace coordinates (in case of high-dimensional observations $\mathbf{x}_{ij}^t$, or when the model spans the entire palette), etc.

In general, we can think of each color model $\mathcal{C}_{ij}^t(s)$ as the parameters of a distribution $p(\mathbf{x}|\mathcal{C}(s))$ over all possible measurements, $\mathbf{x}$. For example, a color table $\mathcal{C}(s)$ could be simply defined as the mean $\boldsymbol{\mu}_s$ and the covariance matrix $\boldsymbol{\Phi}_s$ of a Gaussian distribution over the observation $\mathbf{x}_{ij}^t$,

$$p(\mathbf{x}_{ij}^t|\mathcal{C}(s)) = \mathcal{N}(\mathbf{x}_{ij}^t; \boldsymbol{\mu}_s, \boldsymbol{\Phi}_s), \tag{1}$$

where $\mathbf{x}_{ij}^t$ could be a vector with the color coordinates in a suitable color space, or a vector of Gabor coefficients, or a vector of quantitative texture descriptors, vector of spatial and temporal derivatives, or any other vector describing an image location.

Obviously, there are many other ways to describe the distribution over image measurements, but the discussion that follows is largely independent of the choice of image features and the form of the probability model $p(\mathbf{x}_{ij}^t|\mathcal{C}_{ij}^t(s))$.

First, we show how some of the traditional image representations map into our notation.

**Color palette and image compression.**

In many image formats, it is assumed that each image has its own color table, i.e.,

$$\begin{array}{lcl}
\mathcal{C}_{11}^1 = \mathcal{C}_{12}^1 = ... = \mathcal{C}_{I,J-1}^1 = \mathcal{C}_{IJ}^1 & = & \mathcal{C}^1 = \{\boldsymbol{\mu}_s^1\}_{s=1}^S \\
\mathcal{C}_{11}^2 = \mathcal{C}_{12}^2 = ... = \mathcal{C}_{I,J-1}^2 = \mathcal{C}_{IJ}^2 & = & \mathcal{C}^2 = \{\boldsymbol{\mu}_s^2\}_{s=1}^S \\
 & ... & \\
\mathcal{C}_{11}^T = \mathcal{C}_{12}^T = ... = \mathcal{C}_{I,J-1}^T = \mathcal{C}_{IJ}^T & = & \mathcal{C}^T = \{\boldsymbol{\mu}_s^T\}_{s=1}^S.
\end{array}$$

This representation is useful when each image contains a relatively small number of colors, but sampled from a large portion of the color space. Then, a small number of colors, e.g., $S = 256$, are found that represent all the colors in the image most faithfully. Each entry in the palette is a 24-bit color, but in each location $i,j$ in the image, only the 8-bit index $s_{ij}$ is stored, yielding almost a three-fold compression, as the size of the palette is negligible in comparison with the size of the image. Usually, each image has a separate color palette, although the palettes can also be shared.

**Spatially-invariant color or feature distribution models** Lots of simple image understanding tools rely on color or feature histograms. Faced with the huge variability in the visual data, these algorithms typically assume that images or their portions are as similar as the distribution of colors present in them, and they ignore the spatial configuration of the colors. In our notation this idea can be expressed as the assumption that similar images share the same color model for all pixels

$$\mathcal{C}_{11}^1 = \mathcal{C}_{12}^1 = ... = \mathcal{C}_{I,J-1}^T = \mathcal{C}_{IJ}^T \quad = \quad \mathcal{C}. \tag{2}$$

## 3 Palette-invariant models

We can derive a new class of models that assume that indices $s$ *are* dependent on the coordinates $i, j$, but this information is shared across the collection of images. For example, if we assume that index $s_{ij}$ for each location in the image is shared across the entire collection

$$s_{ij}^1 = s_{ij}^2 = ... = s_{ij}^t = ... = s_{ij}^{T-1} = s_{ij}^T = s_{ij}, \tag{3}$$

and the palette $\mathcal{C}^t$ for each image is shared across all locations $i, j$,

$$\mathcal{C}_{11}^t = \mathcal{C}_{12}^t = ... = \mathcal{C}_{ij}^t = \mathcal{C}_{I,J-1}^t = \mathcal{C}_{IJ}^t = \mathcal{C}^t, \tag{4}$$

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. For example, Fig. 2 shows an index map that describes a whole class of objects. The index map was learned from 50 examples of car images, using the algorithm we will describe shortly. In the same figure, we show the inferred palettes for 8 detected car images outside our training set. One useful property of the palette-invariant model is that it equates the images taken under different overall levels of illumination. However, as shown in our example, the objects with different surface properties but similar spatial structure are also considered similar under this model. The basic palette-invariant model can be extended in several ways, but the most important concept that we would like to focus on in this paper is the introduction of the variability in the index map.

**Modeling uncertainty: probabilistic index maps (PIM) .** We can relax the hard assumption in (3) and allow the indices that model the same location in the image to vary, but follow the same distribution

$$p(s_{ij}^1 = s) = p(s_{ij}^2 = s) = ... = p(s_{ij}^T = s) = p_{ij}(s), \tag{5}$$

where location-dependent distributions $p_{ij}$ describe the variability in different locations of the image, and the overall distribution over the index maps $S = \{s_{ij}^t\}$ is

$$p(S) = \prod_{i,j,t} p_{ij}(s_{ij}^t), \tag{6}$$

and the joint probability distribution is

$$p(S, \mathbf{X}) = \prod_{i,j,t} p(\mathbf{x}_{ij}^t | s_{ij}^t) p_{ij}(s_{ij}^t) \tag{7}$$

For example, if the image collection $\mathbf{X}$ contains the frames from a video of a tree that moves slightly in the wind while the illumination conditions are varying due to the movement of the clouds, then added level of variability in the index map ($p_{ij}$) helps capture the flutter of the leaves, still allowing generalization of the image under varying illuminations. This variability is separate from the intra-image appearance variability captured in the individual palettes, and tends to model intra-class structural variability instead.

## 4 Inference and learning procedures for models using probabilistic index maps

Some of the models in the previous section are not only simple to express in terms of a joint probability distribution functions, but also the standard Bayesian inference and EM learning algorithms can be used to infer the index maps and learn the palettes. However, in general, even the simple models will in practice only be used as components of more complex models that capture other causes of variability in the images, such as large motion, multiple objects and their mutual occlusions, etc. In such models, the indexed-image representations discussed above can be valuable for modeling illumination variability or other significant appearance variability that is not due to structural changes.

In order to use these components in more elaborate generative models, we first derive the cost function and the associated inference and learning algorithm that will make these components more easily extensible.

### 4.1 Free energy of a graphical model

In the previous section, we expressed various models as the joint probability distributions over the data and the variables describing hidden causes of variability. A standard criterion to optimize when fitting such (graphical) models is the likelihood or the log likelihood of the observed data, obtained by summing or integrating over the hidden variables $\mathbf{h}$ for a given set of parameters $\boldsymbol{\theta}$, i.e., $\log p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{\mathbf{h}} \log p(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta})$. In our models, we will treat the index variables $s$ as

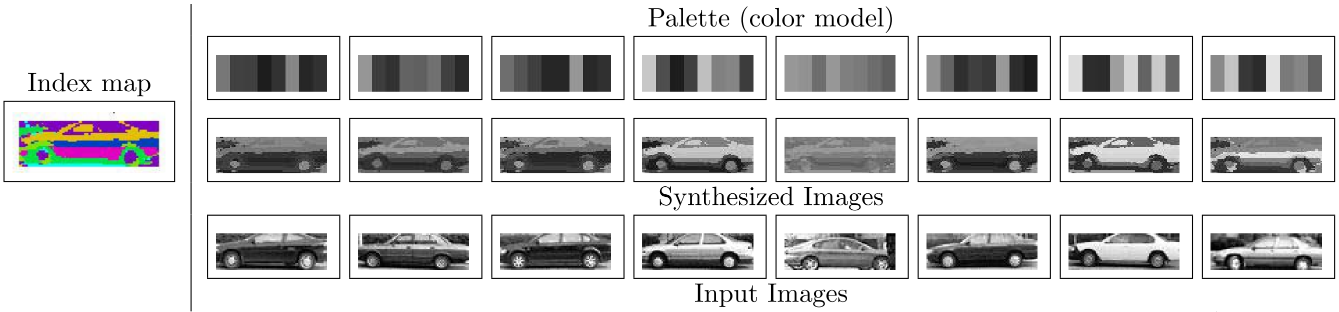


Figure 2: The palette-invariant generative model trained on car images uses the common index map (shown in color on the left) to model different images (bottom), by varying the palettes (top) to produce the corresponding synthesized images (middle).

hidden variables and color/features maps or palettes $\mathcal{C}$ as model parameters.

When this optimization is intractable, approximate methods must be used. The machine-learning community has recently started to converge to a unified view of various approximations. This view is based on an alternative cost, named *free energy* for its similarity with the quantity used in statistical physics. The free energy bounds the negative log likelihood of the data, and is defined as

$$\begin{aligned} F &= \sum_{\mathbf{h}} q(\mathbf{h}) \log \frac{q(\mathbf{h})}{p(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta})} = \\ &= \sum_{\mathbf{h}} q(\mathbf{h}) \log q(\mathbf{h}) - \sum_{\mathbf{h}} q(\mathbf{h}) \log p(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta}) \quad (8) \end{aligned}$$

where $q(\mathbf{h})$ is an arbitrary distribution. Making the substitution $q(\mathbf{h}) = p(\mathbf{h}|\mathbf{x}, \boldsymbol{\theta})$ yields $F = -\log p(\mathbf{x}|\boldsymbol{\theta})$. In addition, using Jensen's inequality, it can be $F \geq -\log p(\mathbf{x}|\boldsymbol{\theta})$ for *any* probability distribution $q(\mathbf{h})$—that is for any function $q(\mathbf{h})$ such that $\sum_{\mathbf{h}} q(\mathbf{h}) = 1$. Thus, $q$ is seen as an approximate posterior distribution, that can be used to compute a lower bound on the log likelihood of the data. In a class of inference algorithms called variational methods, $q$ functions are chosen so as to simplify the free energy into a tractable form that can be efficiently optimized using an iterative algorithm [3]. The variational EM algorithm iteratively decreases $F(q, \boldsymbol{\theta})$, with respect to the posterior distribution and the model parameters. Given that the free energy is bounded from bellow, the algorithm is guaranteed to converge to at least a local maximum.

## 4.2 Free energy of a *probabilistic* index map (PIM)

Here, we derive the inference (E step) and the parameter update rules (M step) for the PIM model (7) that uses a variable index map described by (5) and sample-independent color maps (4). In this model, each observation has a separate index $s_{ij}^t$ but the prior $p_{ij}(s)$ for each location is shared among the observed images. Using a factorized posterior $q(\{s_{ij}^t\}) = \prod_{i,j} q(s_{ij}^t)$, where $q(s_{ij}^t)$ are independent distributions, we can write the free energy as

$$\begin{aligned} F \;=\; & \sum_{i,j,t} \sum_{s_{ij}^t} q(s_{ij}^t) \log q(s_{ij}^t) - \\ & - \sum_{i,j,t} \sum_{s_{ij}^t} q(s_{ij}^t) \log p_{ij}(s_{ij}^t) - \\ & - \sum_{i,j,t} \sum_{s_{ij}^t} q(s_{ij}^t) \log p(\mathbf{x}_{ij}^t | s_{ij}^t, \mathcal{C}^t). \qquad (9) \end{aligned}$$

This can be derived from the general form of the free energy expression by observing that $\sum_s q(s_{ij}^t = s) = 1$ (see also the tutorial at the author's web page if needed). Optimizing for the $q$ distributions under the constraint that $\sum_{s_{ij}^t} q(s_{ij}^t) = 1$, we obtain

$$q(s_{ij}^t) \propto p_{ij}(s_{ij}^t) p(\mathbf{x}_{ij}^t | s_{ij}^t, \mathcal{C}^t), \qquad (10)$$

which is normalized to enforce $\sum_{s_{ij}^t} q(s_{ij}) = 1$. Note that this is the same result that would be obtained from the exact Bayes rule, as the true posterior in deed has the same factorized form. The parameters of this simple model are the $IJ$ multinomial distributions $p_{ij}(s_{ij})$ and the $T$ palettes $\mathcal{C}^t$, and they are estimated in the M step by keeping $q(s_{ij}^t)$ fixed an minimizing the free energy. Assuming a Gaussian model in each entry of the palette $\mathcal{C}^t(s) = \{\boldsymbol{\mu}_s^t, \boldsymbol{\Phi}_s^t\}$, we obtain the following update rules:

$$\begin{aligned} p_{ij}(s_{ij} = s) &= \frac{1}{T} \sum_t q(s_{ij}^t = s) \\ \boldsymbol{\mu}_s^t &= \frac{\sum_{ij} q(s_{ij}^t = s) \mathbf{x}_{ij}^t}{\sum_{ij} q(s_{ij}^t = s)} \\ \boldsymbol{\Phi}_s^t &= \frac{\sum_{ij} q(s_{ij}^t = s) [\mathbf{x}_{ij}^t - \boldsymbol{\mu}_s^t][\mathbf{x}_{ij}^t - \boldsymbol{\mu}_s^t]^T}{\sum_{ij} q(s_{ij}^t = s)}. \end{aligned}$$

## 5 Probabilistic index maps in complex graphical models

The factorized form of the posterior discussed in the previous section is actually exact when there are no additional hidden variables, but in general, factorization can be used as an approximation not only to make inference and learning more tractable for complex graphical models, but also to modularize the inference engine.

### 5.1 PIM mixtures

If the dataset consists of images of different objects, we can add the cluster variable $c$ and use the resulting mixture model to automatically cluster the objects. The cluster variable should affect the structure contained in the index variables $S = \{s_{ij}\}$ and the joint probability distribution is

$$P^t = p(c^t)p(S^t|c^t)p(\mathbf{X}^t|S^t,\mathcal{C}^t), \tag{11}$$

and we can tractably use both the fully factorized posterior $q = q(c)q(S)$ or the full (correct) posterior $q(c)q(S|c)$.

### 5.2 Spatial transformation invariance and alignment

We consider now the case when the images under consideration may have undergone unknown spatial transformations in addition to the palette changes. Such situations are a norm, rather than exception in realistic applications. For example, photographs of the same scene will not only be obtained with slightly different exposure settings, but the angle of view will change, video frames undergo transformations due to camera motion, and even a collection of object photographs will usually come unaligned.

The model that describes such images should include the transformation variable, $\mathbf{T}$ [1]. At a first glance, it may seem that estimating at the same time the transformation and the palette colors will be intractable. However, we show here that the complexity of aligning a map of indices $S = \{s_{ij}\}$ with an unknown palette to an image $X = \{x_{ij}\}$ is of the same complexity as aligning two images, the problem for which the computer vision community has proposed many tractable solutions, e.g., the FFT-based translational alignment and the multi-resolution affine alignment of Lucas and Kanade. To be more precise, cost of inference of both the color map for an image and the transformation will grow only linearly in the number of colors in the palette, as it can be reduced to multiple minimizations of variance in image segments, where each minimization is computationally of similar nature as aligning two images.

To make this point, we will consider the log-distribution over the image $\mathbf{X}$ given the transformation and the index map,

$$\begin{aligned}\log p(\mathbf{X}|S,\mathbf{T}) = \\ -\tfrac{1}{2}\textstyle\sum_{ij}\big[(\mathbf{x}_{ij}-\boldsymbol{\mu}_{s_{\mathbf{T}(ij)}})'\boldsymbol{\Phi}^{-1}_{s_{\mathbf{T}ij}}(\mathbf{x}_{ij}-\boldsymbol{\mu}_{s_{\mathbf{T}(ij)}}) \\ +\log|2\pi\boldsymbol{\Phi}_{s_{\mathbf{T}ij}}\big],\end{aligned} \tag{12}$$

where $\mathbf{T}(ij)$ are the coordinates into which $ij$ maps under $\mathbf{T}$. If not handled properly, this part of the generative model will be the main source of intractability, as maximizing it jointly over color distribution parameters $(\boldsymbol{\mu}_s, \boldsymbol{\Phi}_s)$ and transformations $\mathbf{T}$ will be required. To transform the above into a more tractable computation, we rearrange the summation so that we first sum over all pixels that map to color $s = 1$, then those that map to color $s = 2$, and so on:

$$\begin{aligned}\log p(\mathbf{X}|S,\mathbf{T}) = \\ -\tfrac{1}{2}\textstyle\sum_{k=1}^{S}\sum_{i,j|s_{\mathbf{T}(ij)}=k}\big[(\mathbf{x}_{ij}-\boldsymbol{\mu}_k)'\boldsymbol{\Phi}_k^{-1}(\mathbf{x}_{ij}-\boldsymbol{\mu}_k) \\ +\log|2\pi\boldsymbol{\Phi}_k|\big],\end{aligned} \tag{13}$$

Without the loss of generality, and for the sake of notational simplicity, we will focus on the case of a gray level (scalar) pixels [1], in which case we can write

$$\log p(\mathbf{X}|S,\mathbf{T}) = -\frac{1}{2}\sum_{k=1}^{S} d_k, \tag{14}$$

$$d_k = \mathbf{T}(S_k)'[\phi_k^{-1}(\mathbf{X}-\mu_k)^2 + \log 2\pi\phi_k]. \tag{15}$$

where we use $S_k$ to denote the binary image indicating for each pixel if it is assigned to palette entry $k$ or not, and $\mathbf{T}(S_k)$ is the transformed version of this binary image. These binary images and image $\mathbf{X}$ are represented as one-dimensional vectors of pixels (unwrapped images), so that the distance $d_k$ can be written as an inner product. Palette entry parameters $\mu_k, \phi_k$ are scalar, and the sum of a vector and a scalar is defined as adding the scalar to all elements of the vector, i.e., $\mathbf{X} + \mu = \mathbf{X} + \mu\mathbf{E}$, where $\mathbf{E}$ is the vector of ones.

The sum like the one above will be a part of every model that uses indexed images as well as transformations, and so we will first analyze the computational requirements of optimizing this sum alone, which is

[1] When the covariance matrix $\boldsymbol{\Phi}_k$ is diagonal, the Mahalanobis distance breaks into a sum of distances between scalars, and if $\boldsymbol{\Phi}_k$ is not diagonal, it can be diagonalized by SVD, so both cases can be reduced to the case of scalar observations $x_{ij}$.

equivalent to aligning the index map $S$ to the image $\mathbf{X}$ when palette parameters are unknown. For the transformation $\mathbf{T}$, the palette means and variances the maximize the sum are given by

$$\mu_k = \frac{\mathbf{T}(S_k)'\mathbf{X}}{\mathbf{T}(S_k)'\mathbf{E}}, \quad \phi_k = \frac{\mathbf{T}(S_k)'(\mathbf{X}-\mu_k)^2}{\mathbf{T}(S_k)'\mathbf{E}}. \tag{16}$$

The efficient optimization of $-\sum_k d_k$ should evaluate for each transformation under the consideration the optimal palette entry $\mu_k, \phi_k$ *before* evaluating the entire sum. All of the computations are inner products of vectors, just as traditional image alignment techniques are optimizing the inner product between an image and a transformed version of another $\mathbf{T}(\mathbf{Y})'\mathbf{X}$. Substituting the estimates for the mean and variance in each segment back into (15), we obtain

$$d_k = \log 2\pi\phi_k, \tag{17}$$

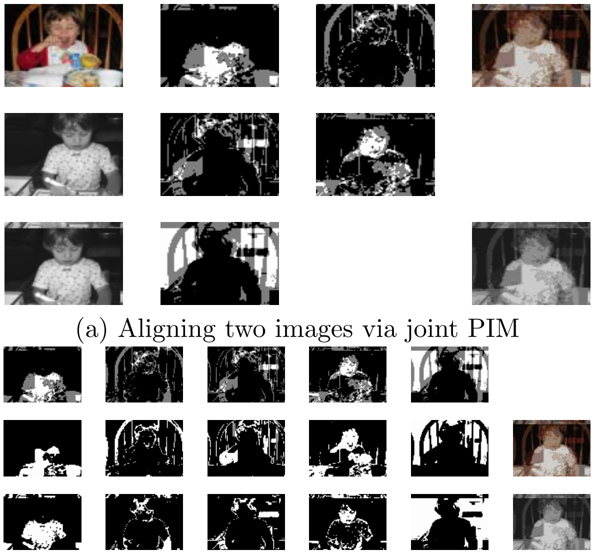
which is the log of the variance (16) in the image region defined by $s_{\mathbf{T}(ij)} = k$. In other words, aligning the index map to an image when the palette is unknown corresponds to minimizing the variances in the image segments defined by the pixel indices.

When $S$ is unknown (hidden variable) the free energy will involve expectations under $q(s_{ij})$ of the above distance, which will in the end have a similar form with $S_k$ now being a soft image of probabilities $q(s_{ij} = k)$.

As noted above, an efficient way of optimizing image distances is the one based on multiresolution. In our case, we would severely downsample $S_k$ and $\mathbf{X}$ and perform quick but rough alignment (search for $\mathbf{T}$), and then refine the search at higher and higher resolutions. In that case, the main increase in computational load in comparison to image-to-image registration would come from having to sum over all indices $k = 1, ..., S$. However, this optimization can be sped up following the same recipe, except that we would deal in addition with the palette multiresolution.

Another efficient way for alignment is based on fast Fourier transformation which can be computed in the time proportional to the log of the image resolution, leading to a representation in which convolution becomes pointwise multiplication. For instance, the following piece of Matlab code would compute the variances $\phi_k$ in (16) for *all* discrete shifts $\mathbf{T}$ in $O(\log(IJ))$ time:

```
fs=fft2(Sk); fx=fft2(X);
fx2=fft2(X.^2); fe=fft2(ones(size(X)));
D=ifft2(conj(fs)*fx-(conj(fs)*fx2./(conj(fs)*fe)).^2);
```



(a) Aligning two images via joint PIM

(b) Inferred individual index maps

Figure 3: Aligning two images with different colors and features. The first two images in the first column of (a) show two different images of a child taken on different days. One image is in color and the other one is black and white. The third image in the column shows the result of BW image alignment. The middle two columns show the probabilistic index map in terms of its components $p(s_{ij} = k)$ using a palette with only five entries. The last column shows the probabilistic index map in terms of the palettes inferred in two images. In (b), we contrast the common probabilistic index map $p(s_{ij})$ (top row) with the inferred index maps $q^t(s_{ij} = k)$ for individual images.

In this implementation, $\mathbf{X}$ and $Sk$ are 2D images (I×J) and the final result $D$ is also a matrix of the same dimensions, so that $i, j$-th entry contains the variance $\phi_k$ (whose log is the distance $d_k$) for the shift of $i$ pixels in one direction and $j$ in the other.

### 5.3 Transformed mixtures of probabilistic index maps

Adding both the mixing variables $c$ and transformation variables $\mathbf{T}$, we can construct a transformed mixture of PIMs (TMPIM), with the joint probability distribution for the $t$-th image

$$p(\mathbf{X}^t, S^t, c^t, \mathbf{T}^t) = p(\mathbf{X}^t|\mathbf{T}^t, S^t)p(S^t|c^t)p(T^t)p(c^t) \tag{18}$$

To minimize the free energy of this model, we use the following, this time approximate, form of the posterior q:

$$q^t = q(c^t)q(S^t|c^t)q(\mathbf{T}^t|c^t), \tag{19}$$

$$q(S^t|c^t) = \prod_{ij} q(s_{ij}^t|c^t) \tag{20}$$

Following the recipe from the previous sections, we derived the update rules leading to the optimization that iterates (a) optimization of the color or feature palette ($\{\boldsymbol{\mu}_k, \boldsymbol{\phi}_k\}$) for each image; (b) inference of the variational posterior for each image (posterior distribution over the segmentation map, transformation that aligns the image with the current guess at PIM $p(S|c)$, and the posterior distribution over the class $c$ for each image); and (c) re-estimation of the class PIMs $p(s|c)$ and the prior $p(c)$. All of these steps are performed by minimizing the free energy of the model, as described above, with the efficient treatment of transformations as described in the previous section.

The resulting algorithm is illustrated on a mini two-image dataset in Fig. 3. In order to align a color image of a child with another gray-level image, we train a single-class TMPIM model which brings the two images into alignment with respect to the shared probabilistic index map. An example of unsupervised clustering images with TMPIM is given in the next section.

## 6 Experimental results and conclusions

The probabilistic index map is a universal image representation that can find its way into many existing computer vision algorithms. To illustrate its benefits, we used the PIM representation in two typical computer vision tasks: background subtraction and transformation-invariant image clustering.

In Fig. 4 we show that PIM representation allows for background subtraction based on a single frame, rather than on tracking incremental changes in a continuous video stream. An 8-index PIM model is learned by minimizing its free energy on the small collection of background images (Section 4.2). Then, for each new test image, the color palette is inferred and the pixel-wise free energy $F_{ij}^t$ is estimated. The foreground detection is then given in terms of the bumps in the energy profile, shown in the last column in (c-f). To better illustrate what is happening "under the hood," the middle column shows the expected background image $\mathbf{B}^t$ using the inferred color palette for each test image,

$$E[b_{ij}^t] = \sum_k q(s_{ij}^t = k)\boldsymbol{\mu}_k. \tag{21}$$

Note, however, that the free energy also depends on the inferred variance for each palette entry.

It is interesting to compare the use of PIM representation with standard appearance models in more complex graphical models. Here, we use Frey and Jojic' transformed mixtures of Gaussians (TMG) for comparison, as this model captures variability in both appearance and transformation, and has been shown to be successful at unsupervised image clustering [1]. As we show in Fig. 5, PIM representation leads to superior illumination invariance at a low extra computation cost. Using the 200 images from the dataset published with the TMG algorithm (Fig. 4a in [1]), the transformed mixture of probabilistic index maps was able to automatically cluster the data in (a) into two clusters representing two different people with an error rate of only 2.5%. In contrast regular mixture of Gaussians and TMG had much poorer error rates of 40.5% and 26%, respectively. All three techniques were applied in a completely unsupervised fashion.[2]

In all experiments we report in this paper, we only used color or gray-level intensity as image features. This makes it easier to separate the benefits of the probabilistic image map representation from the wise choice of local image features. To use other features, we can form an extended feature vector that concatenates the color information with other local measurements. As the model allows for learning the covariance structures for various entries in the palette, (or even more complex probability distributions), the feature selection occurs automatically.

## References

[1] B. Frey and N. Jojic, "Transformation-invaraint clsutering using the EM algorithm ," IEEE Trans. PAMI, Jan. 03.

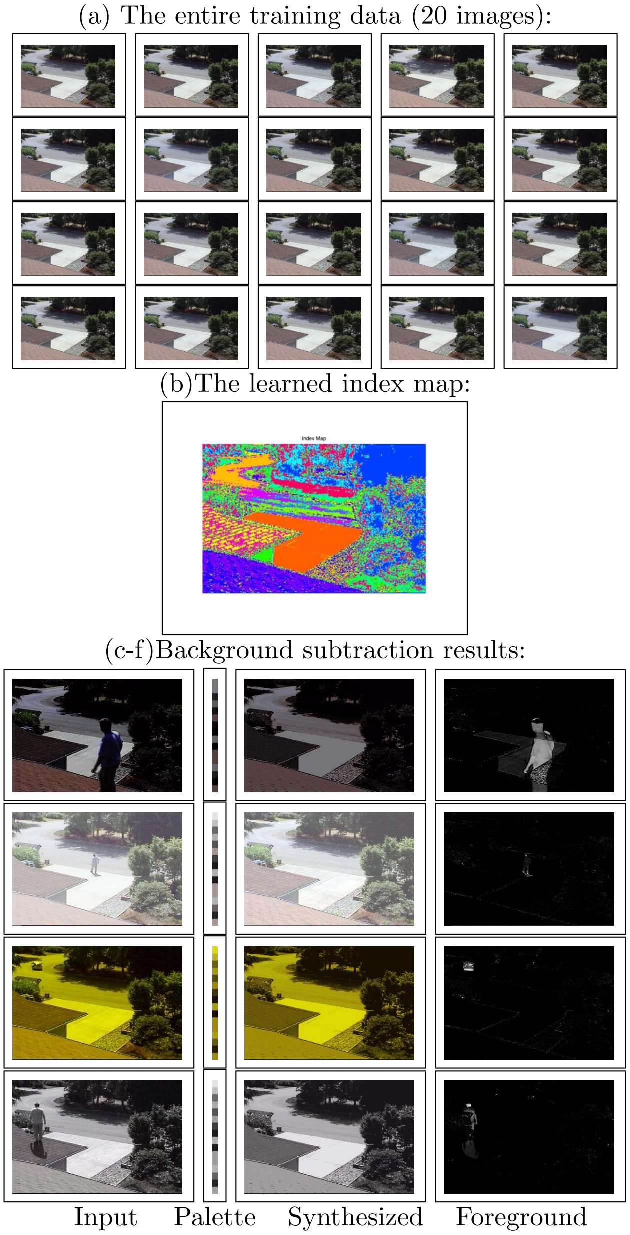
[2] N. Jojic and B. Frey, "Learning flexible sprites in video layers," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '01)

[3] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," in *Learning in Graphical Models*, M. I. Jordan, Ed. Kluwer Academic Publishers, Norwell MA., 1998.

[4] C. Stauffer and E. Grimson, "Similarity templates for detection and recognition," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '01)

[5] C. Stauffer and E. Miller and K. Tieu, "Transform-invariant image decomposition with similarity templates," Proc. of Neural Information Processing Systems (NIPS '01)

[2] For TMG to start separating the two faces into nonoverlapping classes, the complexity (number of classes) has to be increased beyond the equivalent complexity of the TMPIM model. Even then, it is not clear how the resulting multiple classes would be grouped into two identities.



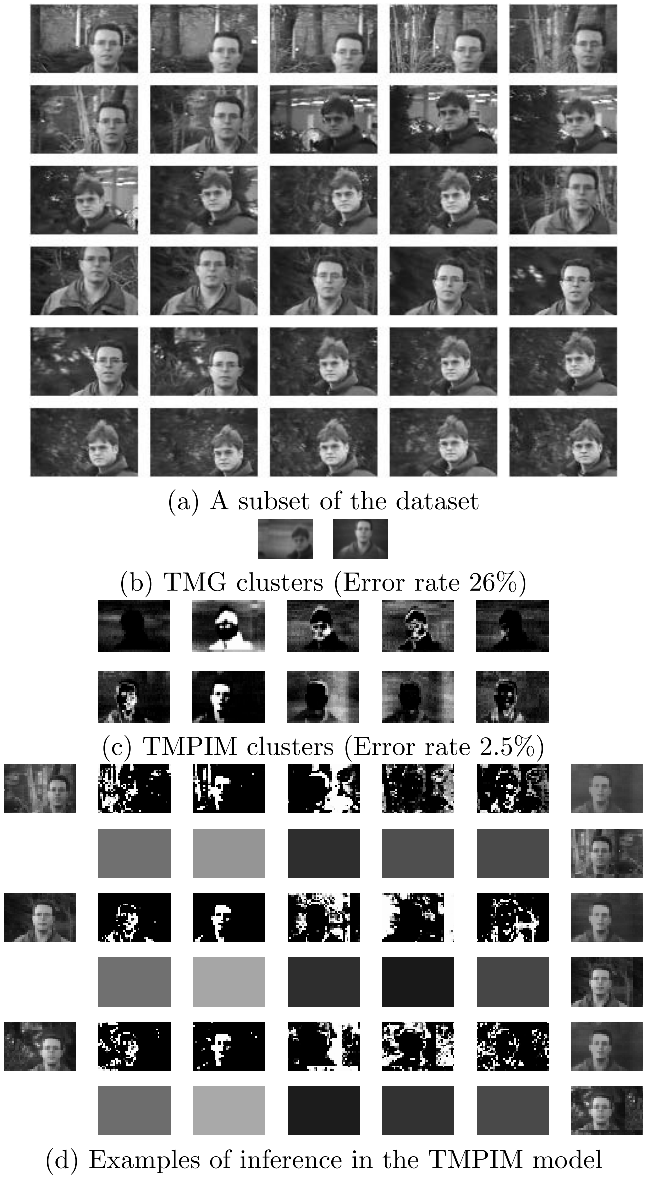
(a) The entire training data (20 images):

(b)The learned index map:

(c-f)Background subtraction results:

Input Palette Synthesized Foreground

Figure 4: Illumination-invariant background subtraction. The background model is trained using only the 20 images shown in (a). The learned index map is shown in (b). Rows (c)-(d) show the images with drastic illumination changes, the recomputed background to match the the new conditions and the result of the background subtraction. The situations PIM model can handle include low illumination (c), image saturation (d), color channel malfunction (e), or even a switch to a different set of measurements, such as IR, or as in (f), gray-level images. Note that in all cases the recovery from the illumination change is instantaneous, and that the color training data had no examples remotely similar in intensities to the test examples.



(a) A subset of the dataset

(b) TMG clusters (Error rate 26%)

(c) TMPIM clusters (Error rate 2.5%)

(d) Examples of inference in the TMPIM model

Figure 5: Unsupervised clustering using transformed mixtures of probabilistic index maps (TMPIM). TMPIM clusters are represented by two distributions $p(S|c)$, shown as probability maps for index k=1,...,5. TMPIM, with its clustering accuracy of 97.5%, compares favorably to the standard mixture of Gaussians model that had a clustering accuracy of only 59.5% and the TMG technique [1] with accuracy of 74%. In (b) we show inferred variational posterior $q(S|c)$, the palette means, the synthesized image and the aligned input for three images.