

Hidden Markov Models for Images

Daniel DeMenthon and David Doermann
Language and Media Processing Laboratory
University of Maryland
College Park, MD 20742-3275
daniel@cfar.umd.edu

Marc Vuilleumier Stüeckelberg
CUI - University of Geneva
Computer Science Department
24, rue Général Dufour
CH-1211 Geneva 4
Marc.VuilleumierStuckelberg@cui.unige.ch

Abstract

In this paper we investigate how speech recognition techniques can be extended to image processing. We describe a method for learning statistical models of images using a second-order hidden Markov mesh model. First, an image can be segmented in a way that best matches its statistical model by an approach related to the dynamic programming used for segmenting Markov chains. Second, given an image segmentation, a statistical model (3D state transition matrix and observation distributions within states) can be estimated. These two steps are repeated until convergence to provide both a segmentation and a statistical model of the image. We also describe a semi-Markov modeling technique in which the distributions of widths and heights of segmented regions are modeled explicitly by gamma distributions, in a way related to explicit duration modeling in HMMs. Finally, we propose a statistical distance measure between images based on the similarity of their statistical models, for classification and retrieval tasks.

1. Introduction

In this paper we address the problem of constructing statistical models of images using Hidden Markov modeling techniques developed in speech recognition. Contextual information in the image is encoded by a 3D array of transition probabilities. A labeling of the image pixels is produced by a global optimization over the whole image using a dynamic programming approach which closely follows the Viterbi algorithm commonly used to segment Markov chains. A segmental k means technique is used to learn the parameters of the statistical model from the image. The primary objective of this work is the computation of a distance measure between images that is less sensitive to camera pan

than techniques based on image feature vectors, yet able to encode image characteristics more specifically than current popular techniques. The most important application of such a distance measure is for indexing and retrieval in image and video databases.

2. Related work

Markov mesh random field models were introduced by Abend, Harley and Kanal [1] in the sixties. They form a causal (or unilateral) subclass of the Markov random field (MRF) pioneered by Besag [2] in the seventies. Some of the problems addressed in the study of MRFs are the labeling of the pixels and the computing of joint likelihoods of the labels [9]. In the case of non causal Markov fields, iterative solutions are available, based on the Gibbs distribution, but the computational cost is generally high [6]. In this paper we focus on the simplest Markov mesh model, a second-order model, i.e. a model in which the label likelihoods depend on the labels of only two past pixel neighbors when the pixels are observed by a raster scan. This model has been studied extensively by Devijver [5], who focused mainly on “real time” labeling, in the sense that only the pixels seen prior to the current pixel while scanning the image are used.

A rather different tack for applying Hidden Markov models to images can be found in the literature, the so-called “pseudo 2-D” HMM approach [10]. Rows of the image are segmented by a left-right version of 1-D HMM. Then patterns of the segmented rows can be grouped into vertical segments containing similar rows by applying an HMM in which the states (called superstates) correspond to different families of segmentation patterns within rows. This technique is being applied in many domains, for example in optical character recognition [10] and face identification [13]. The

method proposed in this paper produces a more descriptive model of image statistics, in the sense that 2-D local mesh context is described. Therefore its domain of application potentially overlaps that of pseudo 2-D HMMs.

3. Modeling components

We consider a rectangular $U \times V$ image. To each pixel (u, v) are associated two quantities, an observable vector $\mathbf{o}_{u,v}$ and a hidden state $q_{u,v}$ which can take the discrete values $\{1, 2, \dots, N\}$. A typical observation vector $\mathbf{o}_{u,v}$ for a pixel could be the list of DCT coefficients for the “pixel” (block) of a JPEG image, the result of a local filter applied over the pixel’s neighborhood, or the set of color components for a color image pixel, the focus of this paper. The hidden state can be viewed as a label for the pixel. The total number of states N for the model is *specified*. Which state best describes the data for each pixel is determined by the modeling process. The labeling of each pixel by a state provides a quantization of the observation vectors, and a segmentation of the image into regions with similar pixels. The relationship between observation vectors and states is part of the model and is expressed statistically. The modeling process will compute the probability distributions $P(\mathbf{o}_{u,v}|q_{u,v})$.

We also define:

$R_{u,v}$ the rectangle of image pixels defined by diagonal corner pixels $(0, 0)$ and (u, v) ;

$\mathbf{x}_{u,v}$ the vector of combined state $q_{u,v}$ and observation $\mathbf{o}_{u,v}$ at pixel (u, v) : $\mathbf{x}_{u,v} = (q_{u,v}, \mathbf{o}_{u,v})$;

$Q_{u,v}$ and $\mathbf{O}_{u,v}$ the configurations of states and observations on rectangles $R_{u,v}$

$\mathbf{X}_{u,v}$ the rectangular configuration of joint states and observations on rectangles $R_{u,v}$.

We simplify pixel indices when only relative pixel positions are important, by omitting indices of the observed pixel, and using index e for its east pixel, index n for its north pixel, and index ne for its north-east pixel.

Conditional probabilities for states at neighbor pixels are written $P(q|q_e, q_n)$, which is shorthand for the full expression $P(q = k|q_e = i, q_n = j)$, where i, j and k are the specific instances, or labels, taken by the pixel states.

4. Markov mesh properties

We assume the configuration of states in the image to be driven by a homogeneous second-order Markov random mesh model. The defining property is [1]

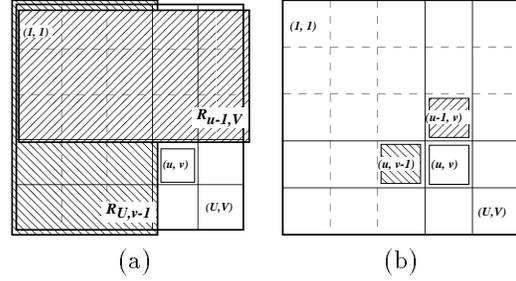


Figure 1. With a second-order causal Markov mesh model applied to an image $I_{U,V}$, the probability of seeing a state i at (u, v) conditional to the state configurations for pixels in rectangles $R_{U,v-1}$, $R_{u-1,V}$ (a) is equal to that probability conditional to the states of the east and north neighbors (b).

$$P(q_{u,v}|Q_{U,v-1} \cup Q_{u-1,V}) = P(q_{u,v}|q_{u,v-1}, q_{u-1,v}) \quad (1)$$

In other words, the probability of having a given state at pixel (u, v) conditional to the states in all the pixels on the rows above row u and all the remaining pixels to the left of column v is simply equal to the probability of having that state at pixel (u, v) conditional to the state of the pixel above (u, v) and to the state of the pixel to the left of (u, v) . This defining property is illustrated in Fig. 1. Kanal and coworkers showed [1] that assuming this property, the probability of seeing the configuration of states $Q_{u,v}$ over the rectangle $R_{u,v}$ is equal to the product of the probabilities of each pixel in the rectangle, each probability being conditional to the states of the east and north neighbor pixels.

$$P(Q_{u,v}) = \prod_{u'=1}^u \prod_{v'=1}^v P(q_{u',v'}|q_{u',v'-1}, q_{u'-1,v'}) \quad (2)$$

with the boundary conditions defined above. This factorization is what makes this Markov model attractive. It is similar in form to the factorization of the joint probability in a 1D Markov chain using first order dependencies between states.

5. Recursive expression for joint state probabilities

Our goal here is to implement for a Markov mesh a dynamic programming approach similar to the Viterbi algorithm used for Markov chains, in order to compute

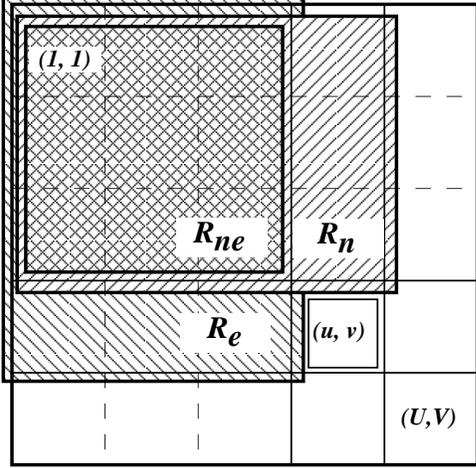


Figure 2. The joint probability for the configuration of pixel states over a rectangle R with a corner at pixel (u, v) can be expressed simply in relation to the probabilities for the configurations over rectangles R_e , R_n and R_{ne} with respective corners east, north and north-east of pixel (u, v) , and to the conditional probability $P(q|q_e, q_n)$ of seeing a state q at pixel (u, v) .

the most probable configurations of pixel states over a whole image. To do this, we need a recursive expression for the joint probability of pixel states and observations over a rectangle:

The joint probability for the configuration of states Q over a rectangle R defined by diagonal pixels $(0, 0)$ and (u, v) can be expressed in relation to the probabilities for the configurations Q_e , Q_n and Q_{ne} , and to the conditional probability of seeing the state q at pixel (u, v) . The expression is:

$$P(Q) = \frac{P(Q_e)P(Q_n)}{P(Q_{ne})}P(q|q_e, q_n) \quad (3)$$

Proof: The two probabilities in the numerator of the right hand side correspond to two rectangles R_e and R_n that overlap (Fig. 2). The probability of the denominator is for a rectangle R_{ne} that is exactly equal to the rectangle of overlap, therefore the factors duplicated in the numerator are eliminated by simplification with the factors of the denominator. This accounts for all the probabilities over rectangle R , except for the probability corresponding to pixel (u, v) . Once this factor is included as shown in Eq. 3 we have all the factors required to yield $P(Q)$.

With the same simplifying assumptions made in hidden Markov models for Markov chains, the observation

\mathbf{o} at pixel (u, v) is assumed independent of all the states other than state q at that pixel, and independent of all the other observations at other pixels. Therefore a second factorization follows:

$$P(\mathbf{O}_{u,v}|Q_{u,v}) = \prod_{u'=1}^u \prod_{v'=1}^v P(\mathbf{o}_{u',v'}|q_{u',v'}) \quad (4)$$

Consider now the event \mathbf{X} defined by the joint occurrence of specific states q and observations \mathbf{o} at pixels of the rectangle R : By combining the factorizations of Eqns 2 and 4, it is straightforward [4] to express \mathbf{X} as the product of individual $P(\mathbf{x})$ at each pixel of the rectangle:

$$P(\mathbf{X}_{u,v}) = \prod_{u'=1}^u \prod_{v'=1}^v P(\mathbf{x}_{u',v'}|q_{u'-1,v'}, q_{u',v'-1}) \quad (5)$$

Therefore, we can write the same type of decomposition for $P(\mathbf{X})$ as we wrote in Eq. 3 for $P(Q_{u,v})$:

$$P(\mathbf{X}) = \frac{P(\mathbf{X}_e)P(\mathbf{X}_n)}{P(\mathbf{X}_{ne})}P(\mathbf{x}|q_e, q_n) \quad (6)$$

6. Dynamic Programming Algorithm

In this section we follow closely (for a second-order Markov model and a Markov mesh) the presentation made by Rabiner for the Viterbi algorithm applied to a first-order model of a Markov chain [12]. We are looking for the single best state configuration $Q_{U,V}$ over the whole image lattice for the given observation map (image) $\mathbf{O}_{u,v}$, i.e. the state configuration that maximizes the probability $P(Q_{u,v}|\mathbf{O}_{u,v})$, or equivalently maximizes the probability $P(\mathbf{O}_{u,v}, Q_{u,v})$, which we called $P(\mathbf{X}_{u,v})$. This is a global optimization problem, which we approach as a dynamic programming problem, using the decomposition of Eq. 6. We define the quantity

$$\delta_{u,v}(q_{u,v}) = \max_{Q_{u,v}-q_{u,v}} P(\mathbf{X}_{u,v}) \quad (7)$$

where the notation $Q_{u,v} - q_{u,v}$ represents the configuration of states for all pixels of the rectangle $R_{u,v}$, except pixel (u, v) . Therefore, when maximized over $Q_{u,v} - q_{u,v}$, the expression remains a function of $q_{u,v}$.

Once we are able to compute such an expression for any (u, v) , then the maximum joint probability for the whole image configuration $Q_{U,V}$ is simply:

$$\max_{Q_{U,V}} P(Q_{U,V}, \mathbf{O}_{U,V}) = \max_{q_{U,V}} \delta_{U,V}(q_{U,V}) \quad (8)$$

We apply the recursion expression of Eq. 6 in the definition 7 of $\delta(q)$ and obtain a recursive equation for $\delta(q)$ by the following steps:

7. Modeling of observation probabilities

$$\begin{aligned} \delta(q) &= \max_{Q-q} \left[\frac{P(\mathbf{X}_e)P(\mathbf{X}_n)}{P(\mathbf{X}_{ne})} P(\mathbf{X}|q_e, q_n) \right] \\ &\approx \frac{\max_{\{q_e, q_n\}} [\delta_\epsilon(q_e)\delta_n(q_n)P(\mathbf{x}|q_e, q_n)]}{\delta_{ne}(q_{ne}^*)} \\ \delta(q) &= \max_{q_e, q_n} [\delta_\epsilon(q_e)\delta_n(q_n)P(q|q_e, q_n)] \frac{P(\mathbf{o}|q)}{\delta_{ne}(q_{ne}^*)} \quad (9) \end{aligned}$$

where $1 \leq q \leq N$ and q_{ne}^* is the state that corresponds to the best score for $\delta_{ne}(k)$. We made use of properties such as

$$\max_{Q-\{q\}} P(\mathbf{X}_e) = \max_{q_e} [\max_{Q_e-\{q_e\}} P(\mathbf{X}_e)] = \max_{q_e} \delta_\epsilon(q_e)$$

This expression is an *approximation*, in the sense that we have assumed that the maximum of the expression is obtained when the common part $P(\mathbf{X}_{ne})$ between $P(\mathbf{X}_e)$ and $P(\mathbf{X}_n)$ is maximum. This common part also appears at the denominator, and its maximum is $\delta_{ne}(q_{ne}^*)$. The approximation assumes that the optimization of δ at the present pixel should not involve the diagonal north-east pixel, which is in line with the second-order Markov mesh simplification assuming that the states at these pixels are independent.

We don't know the index q_{ne}^* yet at the time we do the calculation. However the arguments q_e and q_n that maximize Eq. 9 for each q are independent of the value of $\delta_{ne}(q_{ne}^*)$. Therefore, we use the north-east state that maximizes δ_{ne} at the first iteration, and q_{ne}^* from the map of best states when it becomes available at the next iterations. Other equivalent methods are described in [4].

To retrieve the best state map when the bottom right corner is reached, we need to keep track of the states q_e and q_n that maximized Eq. 9. We store these values in 2D arrays $\psi_{u,v}^H(q_{u,v})$ and $\psi_{u,v}^V(q_{u,v})$ respectively, where the letters H and V are a reminder that these arrays are used to backtrack along horizontal and vertical traces of the paths. Because there are multiple ways of backtracking from the bottom right corner of the image to any pixels, we select for each pixel the state that can be accessed by the largest numbers of backtracking paths. Details are provided in [4].

Experiments [4] confirm that the state configurations obtained by this method have a probability close to the maximum obtained by an exhaustive search (which is achievable only for very small images), and are much better than those obtained by "real-time" methods described in [5].

The observation probabilities provide a model of the variability of the observations at the pixels that have been labeled by the same states. For example, state labeling tends to segment color images into regions of similar color, and the modeling of the color components describes the variations of colors within each type of region. If the components of the observation vector are selected to be approximately independent, the distribution of each component can be independently modeled with a histogram (non-parametric discrete modeling of observations), with a normal distribution, or with a mixture of Gaussians (continuous observation densities). We have used normal distributions to model each of the (R, G, B) color components, and taken the observation probability of a pixel as the product of the probabilities of observations of the color components.

8. Training

The problem of learning the model of a given image is solved by putting the Viterbi algorithm and the probability estimations in an iteration loop. The iteration starts with random transition probabilities in the state transition matrix, while the Gaussians of the observation component probabilities are initially selected to overlap and cover the range of possible observations. The Viterbi algorithm provides a state segmentation. Then the quantities $P(q|q_e, q_n)$ and $P(\mathbf{o}|q)$ are computed by scanning the image, and (1) tallying and normalizing the different types of state transitions between pixels to obtain the state transition matrix, and (2) averaging the values of observation components and their squares for all pixels with the same state labels to estimate the mean and variance of observation components for pixels of identical states. As Viterbi segmentations and probability estimations are repeated alternately, the joint probability of the configuration of observations and states increases and the system consistently converges to a stable solution. Typically, little qualitative improvement can be observed after 10 iterations. The results are a statistical model of the image, a segmentation of the image, and an evaluation of the probability of observing the image given the model. This technique is called a segmental k -means procedure [12]. It is popular in speech recognition as a much faster model training alternative to the Baum-Welch procedure, with comparable quality of results.



Figure 3. Original 24 bit 64×64 Lenna picture (a), and 10-color picture segmented by proposed Viterbi and segmental k -means approach using 10 states (b).

9. Color segmentation

Once a labeling is obtained, we can display the segmented image with the most likely color for each labeled pixel. For each state label, the most likely color is the color whose components are the most likely, because we have assumed that the components are independent. The distribution of each component within each state is modeled by a normal distribution. The most likely color component in each state is the mean of its normal distribution in this state. Fig. 3 shows a 24 bit 64×64 Lenna image (a), and a segmentation obtained using 10 states (b). Note that the number of states is the only specified parameter of this segmentation.

10. Distances between Images

In videos, successive frames of the same scene are typically views of the same objects whose images are shifted in the frames due to the camera’s panning action, and we would like to assess that despite very different local frame pixel values after large camera pans, such frames are still similar. Statistical models of images characterize images, and allow computations of distances, yet are relatively insensitive to translation. Since histograms contain no information about the context of neighbor pixels, researchers have turned to more complex statistical descriptions, such as correlograms [11] with improved resulting image retrieval performance. However they still resort to euclidian distances for evaluating image similarities. It seems more natural to compare images using statistical distance measures between their statistical models.

A distance measure can be computed as follows. We construct a probabilistic model λ_1 of one image I_1 , and to find the distance to a new image I_2 we

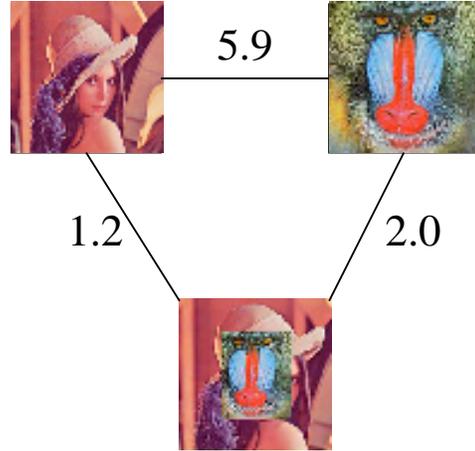


Figure 4. Distance measures between HMM models for 2 images and a composite image.

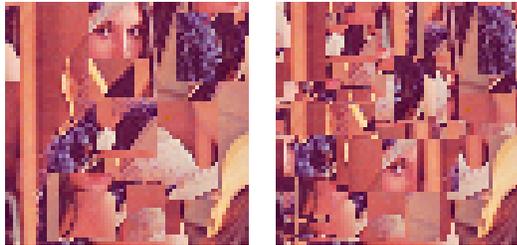
ask: what is the probability that the pattern of pixels on the new image I_2 could have been produced by the probabilistic model λ_1 of the first image? Such a probability is generally quite small and of the form $P(I_2|\lambda_1) = 10^{-D}$, with D larger for more different images. Therefore D , which is the negative log of the probability, $-\log P(I_2|\lambda_1)$, is an intuitive distance measure. To factor out the dependency of the total probability of the observation sequence on the number of observation, we normalize this measure by dividing by the number of pixels. In addition, this measure applied between an image and itself is not zero with this definition, so in most applications a measure shifted by the image distance to itself is preferable. Hence we obtain

$$D(I_1, I_2) = \frac{1}{N_2} \log P(I_2|\lambda_1) - \frac{1}{N_1} \log P(I_1|\lambda_1) \quad (10)$$

This distance measure requires the calculation of the model λ_1 of the first image only, but it is not symmetrical. When time to compute a model for the second image is available, one will prefer to compute a symmetrical distance measure D_s as the half sum of $D(I_1, I_2)$ and $D(I_2, I_1)$.

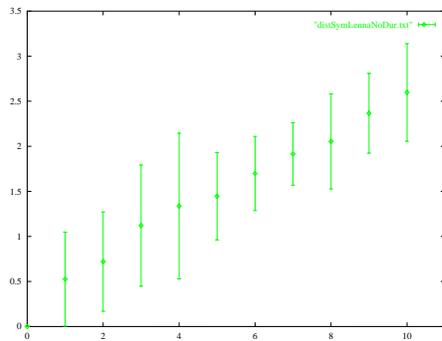
An example of distance calculation is shown in Fig. 4. We used the Lenna and Mandrill images to generate a composite third image in which a piece of the Lenna image is replaced by a piece of the Mandrill image. We obtain an “image triangle” and compute distances between the images using the symmetric distance measure.

The parts of the composite image that contain parts of Lenna’s image are well predicted by Lenna’s model,



(a)

(b)



(c)

Figure 5. Distances between Lenna picture and increasingly scrambled images obtained by swappings of square regions (top images are examples of 5 and 10 swappings). Abscissas show number of swapping, and ordinates show distances to original image. Distances are averages over 10 trials, and error bars are \pm standard deviations. normalsize

so they contribute to producing a larger joint probability and a smaller distance between Lenna and the composite image. We find a distance of 1.2. Similarly, the distance between the Mandrill image and the composite image is small, 2.0. On the other hand the Lenna and Mandrill image don't have much in common, and we find a distance of 5.9. Note that the triangle inequality is not satisfied in this type of example, therefore this distance measure is not a metric. Several studies have shown that human judgement of image similarity also violates metric axioms. We are developing an image retrieval system by applying techniques proposed for non-metric distances by Jacobs, Weinshall and Gdalyahu [8].

Another verification of the merit of this method can be obtained by taking an image, and scrambling it by swapping two pieces of the image. Example of images after 5 and 10 scramblings are shown in Fig. 5. As scrambling increases, the distance from the original image to the scrambled image increases as the image gets more scrambled, as shown in the plot of Fig. 5. This is an intuitive result, which would be difficult to obtain with distance measures which rely on feature matchings.

11. Semi-Markov Model with Explicit State Dimension Probabilities

In speech recognition, the probability of staying in the same state obtained by standard Markov modeling often lacks flexibility. Explicitly modeling "state durations" in analytic form was shown to improve performance [12]. The model then does not follow a Markov model as long as the model stays in the same state, and is called a semi-Markov model.

For a Markov mesh modeling applied to images, limitations of the standard model can also be observed. The probability of staying in the same state k can be shown [4] to decrease monotonically with increasing region size, either horizontally or vertically. Therefore small blobs are modeled as being more probable than larger blobs. Instead, for a model that is trained on an image with redish regions that are mostly 10 pixel wide and 20 pixel high, we would like the model of probability distribution of "state dimensions" to have a maximum at around 10 pixels for the horizontal dimension of contiguous red pixels, and around 20 pixels for the vertical dimension.

For modeling durations, the Gamma distribution was found to produce a good fit to observed duration histograms [3]. It assigns zero probabilities to negative values of the variable, which is very desirable for modeling durations or dimensions. This distribution is of

the form

$$p(x) = \frac{\alpha^p}{\Gamma(p)} e^{-\alpha x} x^{p-1} \quad (11)$$

The two free parameters of the distribution are α and p . For this application, they are positive. The parameter α controls the vertical scaling of the distribution, while p controls whether there is a peak or not and where the peak is located. For $p < 1$, a monotonically decreasing distribution is modeled, otherwise a peak occurs at $\frac{p-1}{\alpha}$. The curve shape is less skewed and more Gaussian-like as the peak is located further from the origin. The two parameters of the Gamma distribution can be simply estimated from the mean \bar{x} and variance \bar{x}^2 of the data x by

$$\alpha = \frac{\bar{x}}{\bar{x}^2}, \quad p = \frac{\bar{x}^2}{x^2}$$

The cumulative probability corresponding to the Gamma distribution is called an incomplete Gamma function. Its complement to 1, $P_k(d)$, represents, for a state k , the probability that the region dimension D will be more than a given dimension d . If we were in a state $q_{u,v-1} = k$ at the previous pixel of the same row and have been in this state for a distance of $d-1$ pixels, the probability of still being in the state k at pixel (u, v) is the probability that the region's horizontal dimension D will be larger than d , given the fact that it is already larger than $d-1$. Therefore

$$P_{\text{stay}}(k, d) = \frac{P_k(d)}{P_k(d-1)} \quad (12)$$

The only other possibility is that we leave state k when we go to the next pixel, therefore the expression for the probability of leaving state k is the complement to 1 of the previous probability.

We can apply this technique to model region shapes in many ways. In our current implementation, we model both the horizontal distance (in pixels) from the top left pixel of a state region to its right edges, and the vertical distance from the top left pixel to its bottom edges. To train this model within the segmental k -means framework described above, as we scan an image that has been segmented into state regions by the dynamic programming algorithm described above, we compute horizontal and vertical distances for each pixel using distance information passed from neighbors and use them to build up the sums used to obtain distance mean and variance in regions. From such means and variances, we can compute gamma distributions for these quantities, and compute the probability of staying in a state or leaving it as described above.

After we have obtained by segmental k -means training the transition matrix a_{ijk} that tabulates the probability of having the current pixel (u, v) of being in state k when the east pixel $(u, v-1)$ is in state j and the north pixel $(u-1, v)$ is in state j , we compute a new transition matrix a'_{ijk} that accounts for the modeling of dimensions of state regions. If k is equal to i or j (or both if i and j are equal) then the current pixel is still in the same state and we ignore the inadequate exponential distribution that the transition matrix would provide for the probability of staying in the same state, replacing it by a combination of probabilities of seeing the observed region dimensions in the horizontal and vertical directions. If however the state k of the current pixel is different from both i and j , then the information provided by the transition matrix is adequate, but must be combined with the probabilities of leaving the regions of the neighbor pixels in a consistent way so that the probabilities in the third dimension of the resulting matrix a_{ijk} add up to 1 (since for given east and north pixel states, the current pixel must be in one of the state $k = 1, \dots, N$.) The full expressions for computing transition matrix terms that account for state dimension modeling are provided in [4].

12. Comparison of Field Realizations

The improvement of image modeling provided by the explicit modeling of region dimensions can be verified by comparing realizations of Markov fields obtained by the plain Markov model and by the semi Markov model. We obtain these realizations by generating random colors from top left to bottom right along a raster scan. States are produced to follow the transition probabilities a'_{ijk} of the model trained on images, with decisions to leave regions produced by Gamma deviates. States are mapped to colors whose components are the means of the normal distributions that model the observations. In Fig. 6, we show realizations for models trained with three images, an image with ellipses of different colors and aspect ratios, the Lenna image, and the Mandrill image (left column). The fields of the middle column are obtained with the plain Markov model, and the fields of the right column are obtained with the semi-Markov model approach. Note that the region sizes, aspect ratios and contiguities are much closer to those of the regions of the original images with the semi-Markov model. A drawback of causal modeling in general is that context can be only verified from previously generated pixels in the raster scan, so there are strong diagonal directional effects for large regions. For a comparison of realizations of MRF models and causal Markov mesh models for mostly binary images,

refer to [7].

13. Conclusion

The main contribution of this paper is the description of a training method for acquiring statistical models of images using a hidden second order Markov mesh model. This model encodes the contiguities of image regions of similar properties, and the distributions of the pixel attributes within each region. Distances between images can be computed as distances between statistical models of these images. We showed using composite images that these distances are very sensitive to common image content without the need for feature matching. Computation time for the Viterbi algorithm is proportional to $(U \times V \times (N^3))$, where U and V are the height and width of the image, N is the specified number of states. Applications being investigated are classification in image data bases, image restoration in MPEG video streams, robust cut and transition detection in videos, and change detection in streams from static cameras. Our research on object recognition applies techniques that mimic keyword spotting in speech recognition, and preliminary results seem particularly promising.

References

- [1] Abend, K., Harley, T.J, and Kanal, L.N., "Classification of Binary Random Patterns", IEEE Trans. Information Theory, IT-11, pp. 538-544, 1965.
- [2] Besag, J., "Spatial Interaction and the Statistical Analysis of Lattice Systems (with Discussion)", J. Royal Statistical Society, series B, vol. 34, pp. 75-83, 1972.
- [3] Burshtein, D., "Robust Parametric Modeling of Durations in Hidden Markov Models", IEEE, 1995.
- [4] ***, "Dynamic Programming and Segmental K-Means Learning for Hidden Markov Modeling of Images", Technical Report, December 1999, [authors have been omitted to keep present paper anonymous].
- [5] Devijver, P.A., "Probabilistic Labeling in a Hidden Second Order Markov Mesh", Pattern Recognition in Practice, II, E.S. Gelsema and L.N. Kanal eds., North-Holland, Amsterdam, 1986.
- [6] Geman, S., and Geman, D., "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. PAMI-6, no. 6, pp. 721-741, 1984.
- [7] Gray, A.J., Kay, J.W., and Titterington, D.M., "An Empirical Study of the Simulation of Various Models Used for Images", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. PAMI-16, no. 5, pp. 507-513, 1994.

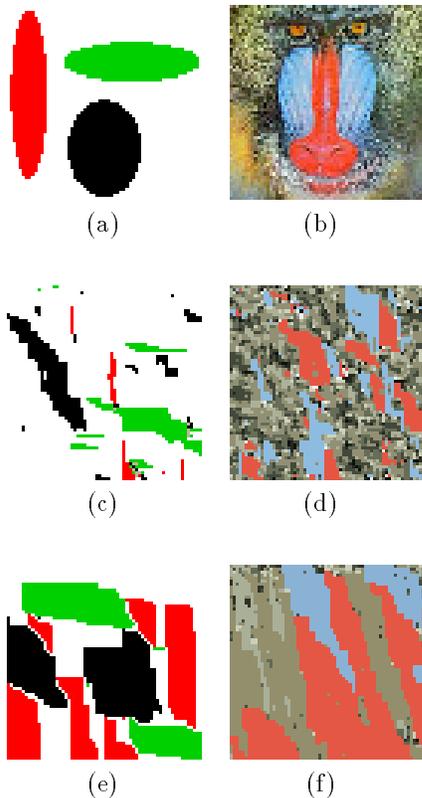


Figure 6. Synthetic images obtained from Markov mesh models trained on images of the top row. Middle row: Standard HMM. Bottom row: HMM with state dimension modeling.

- [8] Jacobs, D., Weinshall, D., and Gdayahu, Y., “Condensing Image Databases when Retrieval is based on Non-Metric Distances”, Proc. 6th International Conference on Computer Vision, 1998.
- [9] Kashyap, R.L., and Chellappa, R., “Estimation and Choice of Neighbors in Spatial Interaction Models of Images”, IEEE Trans. Information Theory, vol. TI-29, pp. 60–72, 1983.
- [10] Kuo, S-S. and Agazzi, O.E., “Keyword Spotting in Poorly Printed Documents using Pseudo 2-D Hidden Markov Models”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 16, no, 8, pp. 842–848, 1994.
- [11] Huang, S., Kumar, S.R., Mitra, M., Zhu. W.J., and Zabih, R., “Image Indexing using Color Histograms”, Proc. Computer Vision and Pattern Recognition, pp. 762–768, 1997.
- [12] Rabiner, L.R., and Juang, B.-H., “Fundamentals of Speech Processing”, Prentice Hall, pp. 321–389, 1993.
- [13] Samaria, F., and Young, S., “HMM-based architecture for Face Identification”, Image and Vision Computing, vol. 12, no. 8, 1994.