LIPSUCKER

Categories:

1 Driving the model

Given our 3D finite element model we use the statistical 2D information provided by LAFTER, a real-time active-camera face and lips tracking system [8] to robustly track and recover the lips in 3D. The system is based on use of 2-D *blob features*, which are spatially-compact clusters of pixels that are similar in a statistical sense and in terms of low-level image properties.

The notion of "blobs" as a representation for image features has a long history in computer vision [9, 7, 2, 14], and has had many different mathematical definitions. In our usage it is a compact set of pixels that share a visual property that is not not shared by the surrounding pixels. This property could be color, texture, brightness, motion, shading, a combination of these, or any other salient spatio-temporal property derived from the signal (the image sequence). In our usage blobs are, therefore, a coarse, locally-adaptive encoding of the images' spatial and color/texture/motion/etc. properties. A prime motivation for our interest in blob representations is our discovery that they can be reliably detected and tracked even in complex, dynamic scenes, and that they can be extracted

in real-time without the need for special purpose hardware. These properties are particularly important in applications that require tracking people, and recently we have used 2-D blob tracking for real-time whole-body human interfaces [14] and real-time recognition of American Sign Language hand gestures [13]. In this method feature vectors at each pixel are formed by adding (x, y) spatial coordinates to the spectral (or textural) components of the imagery. These are then clustered so that image properties such as color and spatial similarity combine to form coherent connected regions, or "blobs," in which all the pixels have similar image properties.

This blob description method is, in fact, a special case of recent Minimum Description Length (MDL) algorithms [4, 3, 1]. By using Expectation Maximization (EM) methods to obtain Gaussian mixture models for the spatio-chrominance feature vector, very complex shapes and color patterns can be adaptively estimated from the image stream. In our system we use an incremental version of EM, which allows us to adaptively and continuously update the spatio-chromatic blob descriptions. Thus not only can we adapt to very different skin colors, etc., but also to changes in illumination.

1.1 Blobs: A Probabilistic Representation

We can represent shapes in both 2-D and 3-D by their low-order statistics. Clusters of 2-D points have 2-D spatial means and covariance matrices, which we shall denote $\bar{q}$ and C_q. The blob spatial statistics are described in terms of their second-order properties; for computational convenience we will interpret this as a Gaussian model. The Gaussian interpretation is not terribly significant, because we also keep a pixel-by-pixel *support map* showing the actual occupancy.

Like other representations used in computer vision and signal analysis, including superquadrics, modal analysis, and eigen-representations, blobs represent the global aspects of the shape and can be augmented with higher-order statistics to attain more detail if the data supports it. The reduction of degrees of freedom from individual pixels to blob parameters is a form of regularization which allows the ill-conditioned problem to be solved in a principled and stable way.

For both 2-D and 3-D blobs, there is a useful physical interpretation of the blob parameters in the image space. The mean represents the geometric center of the blob area (2-D) or volume (3-D). The covariance, being symmetric, can be diagonalized via an eigenvalue decomposition: $C = \Phi L \Phi^T$,where Φ is orthonormal and L is diagonal.

The diagonal L matrix represents the size of the blob along independent orthogonal object-centered axes and Φ is a rotation matrix that brings this object-centered basis in alignment with the coordinate basis of C.

This decomposition and physical interpretation is important for estimation, because the shape L can vary at a different rate than the rotation Φ. The parameters must be separated so they can be treated appropriately.

1.2 Maximum Likelihood Estimation

The blob features are modeled as a mixture of Gaussian distributions in the color (or texture, motion, etc.) space. The algorithm that is generally employed for learning the parameters of such a mixture model is the *Expectation-Maximization (EM)* algorithm of Dempster *et al* [5], [11].

In our system the input data vector d is the normalized R,G,B content of the pixels in the image. The color distribution of each of our blobs is modeled as a mixture of Gaussian Probability Distribution Functions (PDF's) that are iteratively estimated using EM. We can perform a maximum likelihood decision criterium after the clustering is done because human skin forms a dense manifold in *color space*. Two different clustering techniques, both derived from EM are employed: an off-line training process and an on-line adaptive learning process.

In order to determine the mixture parameters of each of the blobs, the unsupervised EM clustering algorithm is computed off-line on hundreds of images of the different classes to be modeled (in our case, face, lips and interior of the mouth). When a new frame is available the likelihood of each pixel is computed using the learned mixture model and compared to a likelihood threshold. Only those pixels whose likelihood is above the threshold are classified as belonging to the model.

1.3 Adaptive Modeling via EM

Even though general models make the system relatively user-independent, they are not as good as an adaptive, user-specific model would be. We therefore use adaptive statistical modeling of the blob features to narrow the general model, so that its parameters are closer to the specific users' characteristics.

The first element of our adaptive modeling is to update the model priors as soon as the user's face

and face features have been detected. The new distribution parameters are computed as follows:

$$\begin{aligned} \Sigma_{new} &= [\Sigma_{general}^{-1} + \Sigma_{user}^{-1}]^{-1} \\ \mu_{new} &= \Sigma_{new}[\Sigma_{general}^{-1} * \mu_{general} + \Sigma_{user}^{-1} * \mu_{user}] \end{aligned} \tag{1}$$

Equation 1 corresponds to a *model averaging* of the general and the learned models, in which the prior probability for each model is the same.

This update of priors occurs only at the beginning of the sequence, assuming that the blob features are not going to drastically change during run time. To obtain a fully adaptive system, however, one must also be able to handle second-to-second severe changes in illumination and user characteristics.

We therefore use an *on-line* Expectation-Maximization algorithm ([10]) to adaptively model the image characteristics, We model both the background and the face as a mixture of Gaussian distributions with mixing proportions π_i and K components:

$$p(x/\Theta) = \sum_{i}^{K} \pi_i \frac{e^{-1/2(x-\mu_i)^T \Sigma_i^{(-1)}(x-\mu_i)}}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \tag{2}$$

The unknown parameters of such a model are the sufficient statistics of each Gaussian distribution (μ_i, Σ_i), the mixing proportions π_i and the number of components of the mixture K.

The sufficient statistics are updated by computing an on-line version of the traditional EM update

rules. If the first n data points have already been computed, the parameters when data point $(n+1)$[1] is read are estimated as follows: First, the *responsibility* h_i^{n+1} for a new data point x^{n+1} is computed:

$$h_i^{n+1} = \frac{\pi_i^n p(x^{n+1}/\theta_i^n)}{\sum_j \pi_j^n p(x^{n+1}/\theta_j^n)} \tag{3}$$

This responsibility can be interpreted as the probability that a data point x^{n+1} was generated by component i. Once this responsibility is known, the sufficient statistics of the mixture components are updated, weighted by the responsibilities:

$$\pi_i^{n+1} = \pi_i^n + \frac{h_i^{n+1} - \pi_i^n}{n} \tag{4}$$

$$\mu_i^{n+1} = \mu_i^n + \frac{h_i^{n+1}}{n * w_i^n}(x^{n+1} - \mu_i^n) \tag{5}$$

$$\sigma_i^{2(n+1)} = \sigma_i^{2(n)} + \frac{h_i^{n+1}}{n * w_i^n}((x^{n+1} - \mu_i^n)^2 - \sigma_i^{2(n)}) \tag{6}$$

where σ_i is the standard deviation of component i and w_i^{n+1} is the *average* responsibility of component i per point: $w_i^{n+1} = w_i^n + \frac{h_i^n - w_i^n}{n}$. The main idea behind this update rules is to distribute the effect of each new observation to all the terms in proportion to their respective likelihoods.

A new component is added to the current mixture model if the most recent observation is not *sufficiently well explained* by the model. If the last observed data point has a very low likelihood with respect of each of the components of the mixture, i.e. if it is an outlier for all the components,

[1]Superscript n will refer in the following to the estimated parameters when n data points have already been processed

then a new component is added with mean the new data point and weight and covariance matrix specified by the user. The threshold in the likelihood can be fixed or could be stochastically chosen: the algorithm would randomly choose whether to add a component or not given an outlier. There is a maximum number of components for a given mixture as well.

The foreground models are initialized with the off-line unsupervised learned *a priori* mixture distributions described above. In this way, the algorithm quickly converges to a mixture model that can be directly related to the *a priori* models' classes. The background models are not initialized with an *a priori* distribution but learned on-line from the image.

1.4 MAP segmentation

Given these models, a MAP foreground-background decision rule is applied to compute *support maps* for each of the classes, that is, pixel-by-pixel maps showing the class membership of each model. Given several statistical blob models that could potentially describe some particular image data, the membership decision is made by searching for the model with the Maximum A Posteriori (MAP) probability.

Once the class memberships have been determined, the statistics of each class are then updated via the EM algorithm, as described above.

Our system exploits a coarse to fine representation to robustly and accurately find and track the mouth of the user.

Our approach to the face finding problem uses coarse color and size/shape information. This

approach has advantages over correlation or eigenspace methods, such as speed and rotation invariance under constant illumination conditions. Moreover, our own work [14], or that of Schiele *et al* or Hunke *et al* [12, 6] have shown that use of normalized or chromatic color information ($\tilde{r} = \frac{r}{r+g+b}, \tilde{g} = \frac{g}{r+g+b}$) can be reliably used for finding 'flesh areas' present in the scene despite wide variations in lighting.

By training the model on thousands of skin color samples, using off-line EM clustering, we have obtained a model that is valid for a broad spectrum of users (Indian, Asian, Caucasian, South American...).

As described above our system uses an adaptive EM algorithm as well. Both the foreground and background classes are learned incrementally from the data. As a trade-off between the adaptation process and speed, new models are updated only when there is a significant drop in the a posteriori match between model and data.

Two to three mixture components is the typical number required to accurately describe the face. Lip models are more densely distributed in the chromatic space, having a considerably smaller variance than skin models. One or two components are typically sufficient to accurately describe lips. Figures 1, 2 and 3 show the probability distributions of both lips ($P(lips|(\tilde{r}, \tilde{g}))$) and skin ($P(skin|(\tilde{r}, \tilde{g}))$) respectively. Both lips and skin pdfs need to be smoothed by convolving them with a smoothing kernel, such as a *hamming* window. However, due to the much higher locality of the lips pdf the convolution window for the lips pdf is much wider than that for the skin pdf.

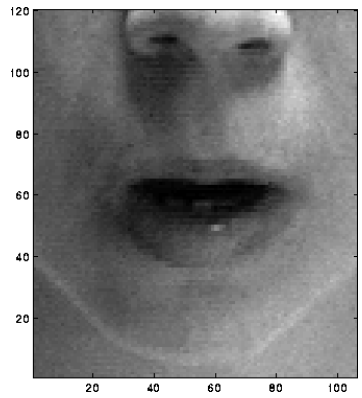

Figure 1: Original image.

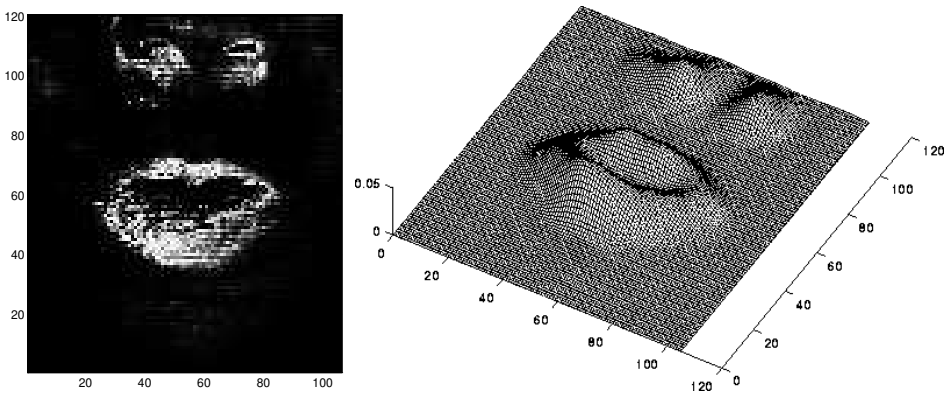

Figure 2: Lips pdf and lips smoothed pdf

1.5 Blob Growing

After initial application of the MAP decision criterion to the image, often isolated and spurious pixels are misclassified. Thus local pixel information needs to be merged into connected regions that correspond to each of the blobs.

The transition from local to global information is achieved by applying a connexity algorithm which grows the blob. The algorithm we use is an speed-optimized version of a traditional connectivity algorithm that considers for each pixel the values within a neighborhood of a certain radius

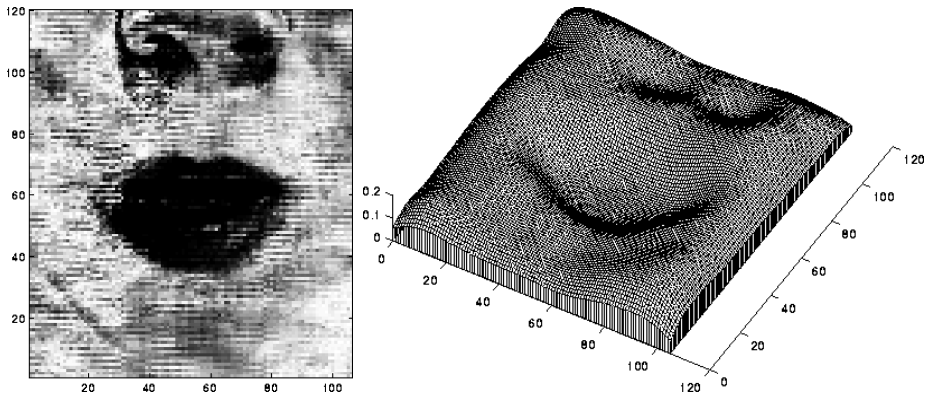

Figure 3: Skin pdf and skin smoothed pdf

(which can be varied at run-time) in order to determine whether this pixel belongs to the same connected region.

Finally, these blobs are then filtered to obtain the best candidate for being a face or a mouth. Color information alone is not robust enough for this purpose. The background, for instance, may contain skin colors that could be grown and erroneously considered as faces. Additional information is thus required. In the current system, geometric information, such as the size and shape of the object to be detected (faces) is combined with the color information to finally locate the face. In consequence, only those 'skin blobs' whose size and shape (ratio of aspect of its bounding box) are closest to the canonical face size and shape are considered. The result is shown in figure 4.

1.6 Active Camera Control

Because our system already maintains a Kalman filter estimate of the centroid and bounding box of each blob, it is a relatively simple matter to use these estimates to control the camera so that the

Figure 4: Face detection and growing

face of the user always appears in the center of the image and with the desired size. Our system uses an abstraction of the camera control parameters, so that different camera/motor systems (currently the Canon VCC1 and Sony EVI-D30) can be successfully used in a totally transparent way. In order to increase tracking performance, the camera pan-tilt-zoom control is done by an independent light-weight process (thread) which is started by the main program.

The current estimation of the position and size of the user's face provides a reference signal to a PD controller which determines the tilt, pan and zoom of the camera so that the target (face) has the desired size and is at the desired location.

2 Mouth Extraction and Tracking

Once the face location and shape parameters are known (center of the face, width, height and image rotation angle), we can use anthropometric statistics to define a bounding box within which the mouth must be located.

The mouth is modeled using the same principles as the face, i.e. through a second-order mixture

model that describes both its chromatic color and spatial distribution. However to obtain good performance we must also produce a more finely detailed model of the face region surrounding the mouth. The face model that is adequate for detection and tracking is not adequate for accurate mouth shape extraction.

Our system, therefore, acquires image patches from around the located mouth and builds a Gaussian mixture model. In the current implementation, skin samples of three different facial regions around the mouth are extracted during the initialization phase and their statistics are computed.

The robustness of the system is increased by computing at each time step the linearly predicted position of the center of the mouth. A confidence level on the prediction is also computed, depending on the prediction error. When the prediction is not available or its confidence level drops below a threshold, the mouth's position is reinitialized.

Rotation invariance is achieved by computing the face's image-plane rotation angle and rotating the region of interest with the negative of this angle. Therefore even though the user might turn the head the mouth always appears nearly horizontal

3 Statistical derivation of the forces

The gradient of the log likelihood of each of the classes given the chromatic color information provides a continuous force field in the chromatic color space. For each pixel in the image we can

compute the force induced by this statistically derived force field as the gradient:

$$f_{lips}(x,y)_x = \frac{\partial(log(P(lips|(\tilde{r},\tilde{g}))))}{\partial x} \quad (7)$$

$$f_{lips}(x,y)_y = \frac{\partial(log(P(lips|(\tilde{r},\tilde{g}))))}{\partial y} \quad (8)$$

for a lips pixel and an analogous expression for a skin pixel.

This force field allows us to avoid the *correspondence problem* between observation points in the image space and control points in the model. Instead of selecting some feature points in the image and tracking them, solving the correspondences frame per frame, we just have to compute the force applied to each of the control points of the 3D model. In consequence, traditional problems derived from the correspondeces, such as noise, instability, lack of robustness and occlusions are avoided in our framework.

3.1 Labeling of the points : different nature : different cost functions

References

[1] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV95*, pages 777–784, 1995.

[2] A. Bobick and R. Bolles. The representation space paradigm of concurrent evolving object descriptions. *PAMI*, 14(2):146–156, February 1992.

[3] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *PAMI*, 17(5):474–487, May 1995.

[4] T. Darrell, S. Sclaroff, and A. Pentland. Segmentation by minimal description. In *ICCV90*, pages 173–177, 1990.

[5] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via de *em* algorithm. *Journal of the Royal Statistical Society*, 39-B:1–38, 1977.

[6] H. Hunke. Locating and tracking of human faces with neural networks. Technical report, CMU, Pittsburgh PA, August 1994.

[7] R. Kauth, A. Pentland, and G. Thomas. Blob: An unsupervised clustering approach to spatial preprocessing of mss imagery. In *11th Int'l Symp. on Remote Sensing of the Environment*, Ann Harbor MI, 1977.

[8] N. Oliver, A. Pentland, and F. Bérard. Lafter: Lips and face real time tracker. In *IEEE International Conference on Computer Vision and Pattern Recognition. CVPR'97*, Puerto Rico, June 1997.

[9] A. Pentland. Classification by clustering. In *IEEE Symp. on Machine Processing and Remotely Sensed Data*, Purdue, IN, 1976.

[10] C. Priebe. Adaptive mixtures. *Journal of the American Statistical Association*, 89(427), 1994.

[11] R. Redner and H. Walker. Mixture densities, maximum likelihood and the *em* algorithm. *SIAM Review*, 26:195–239, 1984.

[12] B. Schiele and A. Waibel. Gaze tracking based on face color. In *International Workshop on Automatic Face and Gesture Recognition*, pages 344–349, 1995.

[13] T. Starner and A. Pentland. Real-time asl recognition from video using hmm's. Technical Report 375, MIT, Media Laboratory, MIT, Media Laboratory, Cambridge, MA 02139, 1996.

[14] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *Photonics East, SPIE*, volume 2615, Bellingham, WA, 1995.