Bayesian Clustering of Optical Flow Fields

Jesse Hoey James J. Little
Department of Computer Science
University of British Columbia
Vancouver, BC, CANADA

Abstract

We present a method for unsupervised learning of classes of motions in video. We project optical flow fields to a complete, orthogonal, a-priori set of basis functions in a probabilistic fashion, which improves the estimation of the projections by incorporating uncertainties in the flows. We then cluster the projections using a mixture of feature-weighted Gaussians over optical flow fields. The resulting model extracts a concise probabilistic description of the major classes of optical flow present. The method is demonstrated on a video of a person's facial expressions.

1. Introduction

Unsupervised learning of categories of motions from video streams is an important open problem. It finds applications in video databases and surveillance, where the goal is to *discover* patterns of motion without having to label an extensive training set. It also finds applications in understanding human motions, where the goal is to recognize patterns of human action which are predictive of actions. For example, an agent observing a computer user would like to correlate the motions of a particular user's face and body with the ongoing context of interaction and the user's actions. Since parts of each user's motions are unique, the agent can benefit from discovering important categories of motion. While many approaches use dynamical models [5, 4], we attempt to cluster the instantaneous motions on a frame by frame basis. This paper describes a method for learning the salient classes of frame-to-frame motions in a video.

The first step is to estimate the way things are moving between frames. Optical flow provides such an estimate [2], but yields a very high dimensional signal. While clustering optical flow fields directly is possible, we prefer to first extract some low dimensional representation of the flow fields, making clustering easier. The problem is then one of finding an appropriate subspace of the flow field vector space in which the motions we are trying to categorize are sufficiently well separated. Further, we would like such a subspace to be data independent. The advantage of data independence is that the subspace can deal with arbitrary motions, and models can be learned for different types of motions. For example, while we learn classes of face motions in this paper, our system could be easily applied to gestures or gaits. Leaving any commitment to particular motions to higher level processing is an advantage in many cases.

Data independence implies an *a-priori* set of basis functions onto which the optical flow fields can be projected. In this work, we use the complete and orthogonal basis of Zernike polynomials. This basis set provides a rich and data independent description of optical flow fields which can be seen as an extension of the affine basis. However, simply computing optical flow and subsequently projecting to this basis does not make use of the uncertainties inherent in the estimation process. We describe an efficient Bayesian solution for directly computing probability distributions over basis coefficients from image gradients using brightness constancy. We show how the method's incorporation of the flow uncertainties improves the estimation of the parametrized flow by discounting image regions with less certain flow vectors.

Clustering the optical flow fields can then be accomplished using a mixture of Gaussians over the basis vector space. Our mixture model includes weights on the basis coefficients, which describe the effectiveness of each feature at achieving good clusters. Our probabilistic projections ensure that the uncertainties inherent in the calculation of optical flow are propagated to the cluster membership variables, leading to a more robust clustering. We show how to learn the parameters of the mixture model, including the feature weights, with the expectation-maximization (EM) algorithm. Learning the feature weights fuses well with the probabilistic projections of optical flows, as it allows the model to take the structured optical flow variances into account when building clusters.

This paper makes two main contributions. First, it describes an efficient method for representing flow fields in a compact representation, while preserving information about the variance inherent in the flow calculation, and composing it in a Bayesian manner with a mixture model's predictions over the flow field parametrization. Second, it describes a method for learning the parameters of a model for clustering optical flow fields with feature weighting.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

Much effort has recently been expended in classification of human motions in video [7, 5, 4, 10, 8]. These methods, however, usually include a supervised training phase. This requires extensive labelled training data, which may be difficult and costly to obtain. We wish to avoid such a step, and build an unsupervised system which *discovers* the salient patterns of motion in a video database. Our work also differs from other human motion analyses in that we do not build a dynamical model, but only cluster instantaneous motions. We use the feature weighting techniques of [6], with modifications to incorporate our probabilistic projections. Our probabilistic projections of flow fields can be seen as a method for constrained optical flow estimation, which have been extensively studied in the literature [11]. Work has also been done on categorizing motion vectors within a flow field [3], such as in the case of multiple motions. Our work instead approaches the problem of categorizing entire flow fields directly [8].

This paper is organized as follows. Section 2 describes the clustering model, including the Zernike basis and the probabilistic projections of optical flow fields. It also generalizes the method to a multi-scale version, and describes the feature weighting priors. Section 2.5 then describes how to learn the maximum *a-posteriori* parameters of the Gaussian mixture model with feature weighting from training data, using the expectation-maximization (EM) algorithm. Section 3 shows some results demonstrating the benefits of the probabilistic projections, and clustering results on images of a person's facial expressions.

2. Overview of model

Given a set of images, $I_1 \dots I_{N_t}$, we wish to find N_x clusters of the optical flow fields, v_t, between successive pairs of images, $\{I_t, I_{t+1}\}$. That is, we wish to assign each of the $N_t - 1$ flow fields one of N_x cluster labels $X_1 \dots X_{N_x}$, such that the optical flows with the same label are as similar as possible to each other, but as dissimilar as possible from the flow fields with any other label. For example, if describing motion over the human face, states of X may correspond to instantaneous motion fields during *smiling* (mouth expansion), *frowning* (contraction between the eyes), or *talking* (lip motion). Figure 1 shows our model represented as a Bayesian network. We consider the measurements to be the image spatial ($f_s = \{f_x, f_y\}$) and temporal (f_τ) derivatives, calculated using a centered difference method. [1] We can express classification of an image motion as the maximization of the probability distribution over the classes, X,

[1] These variables are fields over all N pixels in the image: f_τ is a $N \times 1$ column matrix, $f_s = [f_x f_y]$ is a $N \times 2N$ matrix (where f_x is a $N \times N$ matrix with the horizontal spatial derivative f_x along the diagonal, and similarly for f_y) and $v = [v_x v_y]'$ is a $2N \times 1$ matrix with the components of horizontal and vertical flow.

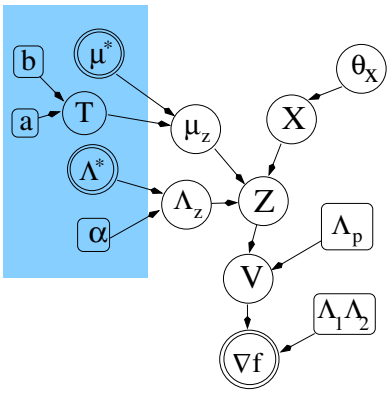


Figure 1: Bayesian network for the mixture of Gaussians over optical flow fields with feature weighting. Double circle nodes are observed, single circle nodes are unknown random variables, and boxes are fixed hyper-parameters. The shaded area contains the priors for feature weighting.

given the spatial and temporal derivatives,

$$P(X|\nabla f, \Theta) \propto P(\nabla f|X, \Theta)P(X|\Theta), \qquad (1)$$

where Θ are the parameters of the model, and $\nabla f = \{f_x, f_y, f_\tau\}$. Since we wish to classify optical flow fields, we expand the probability distribution over the classes, X, as

$$P(X|\nabla f, \Theta) = \int_v P(\nabla f|v, \Theta)P(v|X, \Theta)P(X|\Theta)$$

where we have assumed the image derivatives to be independent of the high level motion class given the optical flow. To simplify notation in the following, we assume dependence on the model parameters, Θ, for every term, and drop explicit reference to Θ.

There are three terms in the integration. The prior over classes, $P(X)$, is part of our model, parametrized with a multinomial $\Theta_{x,i} = P(X = i)$. The distribution over spatio-temporal derivatives conditioned on the flow, $P(\nabla f|v)$, is estimated in a gradient-based formulation using the brightness constancy assumption, $f_\tau + f_s v = 0$. The noise in the estimation is described with two zero-mean Gaussian variances, Λ_1 and Λ_2, which result from failures of the planarity assumption, and errors in the temporal derivative measurements [13]:

$$P(\nabla f|v) \propto \mathcal{N}(f_\tau; -f_s v, f_s \Lambda_1 f_s' + \Lambda_2), \qquad (2)$$

where $\Lambda_1 = \sigma_1 I_N$, $\Lambda_2 = \sigma_2 I_N$ (I_N is $N \times N$ identity).

We do not represent $P(v|X)$ directly in our model, but instead we parametrize this distribution using a probabilistic projection of v to the basis of Zernike polynomials. As we will show, this projection can be written as a distribution

over v, given the projection coefficients, z, $P(v|z)$. We then parametrize the distribution over z given X with a normal $P(z|X) = \mathcal{N}(z; \mu_{z,x}, \Lambda_{z,x})$. We are expecting flow fields to be normally distributed in the space of the basis function projections.

A more naïve approach avoids the integrations over z by first computing the mean optical flow field, μ_v, using a zero-mean prior, projecting this field to the Zernike basis, and taking the resulting feature vector, z, as the input data to a classification scheme using $P(z|x)$ [9]. That is, the naïve approach considers $P(X|\nabla f) = P(M\mu_v|X)P(X)$, where the columns of M are the basis functions. However, this approach ignores the variance information in the flow calculation, leading to less accurate results. For example, Figure 2 shows two frames from a video sequence of a person's face. There is significant motion upwards near and above the eyebrows and downwards along the sides of the jaw. The mean flow field, μ_v, calculated using the method

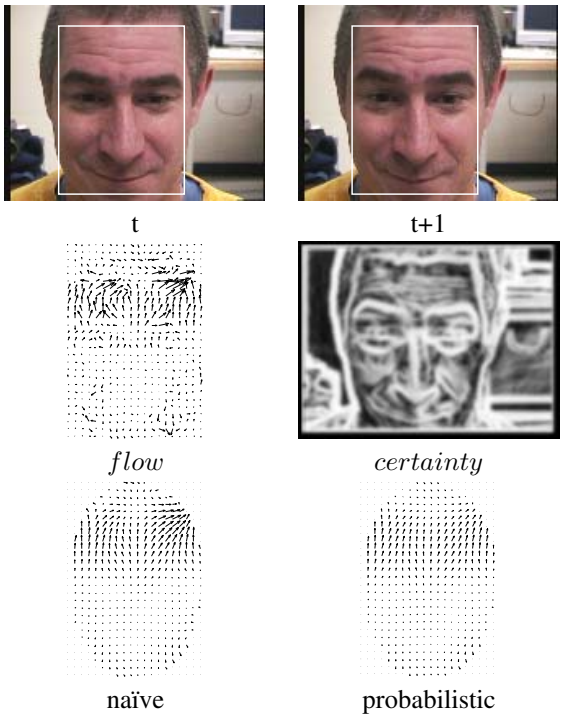


Figure 2: Top: two subsequent frames from a video sequence. Middle: variance on the horizontal (Λ_u) and vertical (Λ_v) optical flow fields. Bottom: naïve and probabilistic reconstructions from low dimensional basis

of Simoncelli [13], is shown in Figure 2, for the image region of the subject's face as indicated. The certainty of the flow vectors (the trace of the inverse flow variances), is also shown in Figure 2 (brighter means the flow estimates are more certain). Large variance flows are prevalent in regions with little contrast since we are using a gradient based optical flow calculation. The jaw, forehead and the background wall are examples. A projection of these flow fields to a low dimensional basis will suffer because of these regions, unless the flow variances are taken into account. The bottom row in Figure 2 shows reconstructions of the flow fields from projections to the Zernike basis. On the left, we see a simple projection (dot product), while on the right is the projection which takes the variances on the flow into account. We can see improvements around the jaw and forehead areas. We compare our probabilistic projection approach with the naïve approach further in Section 3.1.

In the remainder of this section, we describe optical flow and the Zernike basis, and we show how the integrations in Equation (2) can be performed analytically, leading to an efficient method for calculating $P(X|\nabla f)$, taking all variance information in the flow fields into account. We then show how to implement weights on the dimensions of the projections, and how our methods can be implemented in a multi-scale approach. Section 2.5 shows how the distribution $P(X|\nabla f)$ is used to learn the parameters of the model using the expectation-maximization algorithm.

Note that the model does not take violations of the brightness constancy assumption, such as occlusions, reflections, or transparent motions, into account. While this is an important problem, it typically leads to high spatial frequency violations of our assumptions. Since we are mainly interested in generating low dimensional representations of optical flow fields suitable for clustering, we do not consider this problem further here.

2.1. Zernike projections

Zernike polynomials are a complete and orthogonal set of complex polynomials defined on the unit disk [12], and can be used to represent the flow fields, v, over some image region, φ, to an arbitrary degree of accuracy. The lowest two orders of Zernike polynomials correspond to the standard affine basis. The next order polynomials correspond to extensions of the affine basis, roughly *yaw*, *pitch* and *roll*, as explored in [4]. Higher orders represent motions with higher spatial frequencies. The basis is orthogonal over the unit disk, such that each order can be used as an independent characterization of the flow, and each flow field has a unique decomposition in the basis. Zernike polynomials are expressed in polar coordinates as a radial function, $R_n^m(\rho)$, modulated by a complex exponential in the angle, ϕ:

$$U_n^m(\rho, \phi) = R_n^m(\rho)e^{im\phi} \tag{3}$$

The indices n,m control the spatial frequency of the basis functions. The orthogonality of the basis allows the decomposition of an arbitrary function on the unit disk, $F(\rho, \phi)$, in

terms of a unique combination of Zernike polynomials [12]:

$$F(\rho,\phi) \approx \sum_{m=0}^{M}\sum_{n=m}^{N}\left[A_n^m cos(m\phi) + B_n^m sin(m\phi)\right] R_n^m(\rho), \tag{4}$$

The coefficients, A_n^m and B_n^m, of the decomposition of the horizontal and vertical flow estimates, $u(x,y)$ and $v(x,y)$, are obtained using:

$$\begin{matrix} {}^{u}A_n^m \\ {}^{u}B_n^m \end{matrix} = \frac{\epsilon_m(n+1)}{\pi}\sum_x\sum_y u(x,y)R_n^m(\rho)\begin{matrix} cos(m\phi) \\ sin(m\phi) \end{matrix} \tag{5}$$

Equation (4) allows us to represent the optical flow fields to an arbitrary degree of accuracy as $v = Mz$, where

$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad M = \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} \quad z = \begin{bmatrix} z_x \\ z_y \end{bmatrix}. \tag{6}$$

The columns of P are the N_z basis vectors and z are the Zernike coefficients, A_n^m and B_n^m. As $N_z \rightarrow \infty$, the reconstruction error of v from z becomes arbitrarily small. In practice, M will be some subset of the Zernike basis vectors, the remaining variance in the flow fields being attributed to zero-mean Gaussian noise. Thus, we write $v = Mz + n_p$, where $n_p \propto \mathcal{N}(0,\Lambda_p)$, and so $P(v|z) = \mathcal{N}(v; Mz, \Lambda_p)$.

2.2. Estimating the likelihood

We can now write down the likelihood of the image derivatives given the high-level motion class (Equation 2) as

$$P(\nabla f|X) = \int_{v,z}\mathcal{N}(f_\tau; -f_s v, A)\mathcal{N}(v; Mz, \Lambda_p)\mathcal{N}(z; \mu_{z,x}, \Lambda_{z,x}) \tag{7}$$

where $A = f_s\Lambda_1 f_s' + \Lambda_2$. This distribution is implicitly conditioned on the image region, φ, since the derivatives are computed over this image region. We consider the image region to be specified in this paper. We can perform the integrations over v and z by successively completing the squares in v and z to obtain

$$P(\nabla f_\tau|x_t) = \frac{\sqrt{|\tilde{\Lambda}_{z,x}|}}{\sqrt{|A||\Lambda_{z,x}|}} e^{\frac{1}{2}(\tilde{\mu}'_{z,x}\tilde{\Lambda}^{-1}_{z,x}\tilde{\mu}_{z,x} - \mu'_{z,x}\Lambda^{-1}_{z,x}\mu_{z,x} - \epsilon)} \tag{8}$$

where

$$\begin{aligned} \Lambda_w &= (f_s'A^{-1}f_s + \Lambda_d^{-1})^{-1} \\ \tilde{\Lambda}_{z,x} &= (\Lambda_{z,x}^{-1} + M'(\Lambda_d + (f_s'A^{-1}f_s)^{-1})^{-1}M)^{-1} \\ \tilde{\mu}_{z,x} &= \tilde{\Lambda}_{z,x}(\Lambda_{z,x}^{-1}\mu_{z,x} - M'\Lambda_d^{-1}\Lambda_w w) \\ \epsilon &= f_\tau'A^{-1}f_\tau + w'\Lambda_w w \qquad w = f_s'A^{-1}f_\tau \end{aligned} \tag{9}$$

If we normalize this distribution over x, we can remove all terms which are independent of x, and obtain

$$\frac{P(f_\tau|xf_s)}{\sum_x P(f_\tau|xf_s)} = \frac{\sqrt{|\tilde{\Lambda}_{z,x}|}}{\sqrt{|\Lambda_{z,x}|}} e^{\frac{1}{2}(\tilde{\mu}'_{z,x}\tilde{\Lambda}^{-1}_{z,x}\tilde{\mu}_{z,x} - \mu'_{z,x}\Lambda^{-1}_{z,x}\mu_{z,x})}. \tag{10}$$

The mean, $\tilde{\mu}_{z,x}$, and covariance, $\tilde{\Lambda}_{z,x}$, are the parameters of the distribution of basis vector coefficients, z:

$$P(z|X\nabla f) \propto e^{-(z-\tilde{\mu}_{z,x})'\tilde{\Lambda}^{-1}_{z,x}(z-\tilde{\mu}_{z,x})},$$

Thus, the most likely Zernike vector, $\tilde{z}$, given the model can be computed as $\tilde{z} = \sum_{i=1}^{N_x}\tilde{\mu}_{z,i}\Theta_{x,i}$, and the most likely flow field, $\tilde{v}$, for a given state can be computed using

$$\tilde{v}_x = M\tilde{\mu}_{z,x} \tag{11}$$

2.3. Multi-scale implementation

The brightness constancy assumption fails if the velocity v is large enough to produce aliasing. Therefore, a multi-scale pyramid decomposition of the optical flow field must be used. This results in distribution over the flow vectors, $P(v|\nabla f) \sim \mathcal{N}(v; \mu_v, \Lambda_v)$, where $\Lambda_v = (f_s'A^{-1}f_s)^{-1}$ and $\mu_v = -\Lambda_v f_s'A^{-1}f_\tau$ [13]. Using these coarse-to-fine estimates, Equations 9 become

$$\begin{aligned} \tilde{\Lambda}_{z,x} &= (\Lambda_{z,x}^{-1} + M'(\Lambda_d + \Lambda_v)^{-1}M)^{-1} \\ \tilde{\mu}_{z,x} &= \tilde{\Lambda}_{z,x}(\Lambda_{z,x}^{-1}\mu_{z,x} + M'(\Lambda_d + \Lambda_v)^{-1}\mu_v) \end{aligned} \tag{12}$$

2.4. Feature weighting

In general, we will not know which basis coefficients are the most useful for our classification task: which basis vectors should be included in M, and which should be left out (as part of n_p). Further, *selecting* a relevant subset of the basis vectors for clustering can lead to significant computational savings. We build on the feature weighting techniques of [6], which characterize the relevance of basis vectors by examining how the cluster means, $\mu_{z,x}$, are distributed along each basis dimension, $k = 1 \ldots N_z$. Relevant dimensions will have well separated means (large inter-class distance along that dimension), while irrelevant dimensions will have means which are all similar to the mean of the data, μ^*. To implement these notions, we place a prior on the cluster means, $\mu_{z,x} \sim \mathcal{N}(\mu^*, T)$, where T is diagonal with elements $\tau_1^2 \ldots \tau_{N_z}^2$, and τ_k^2 is the feature weight for dimension k. τ_k^2 will be large if k is a dimension relevant to the clustering task, while $\tau_k^2 \rightarrow 0$ if the dimension is irrelevant. Feature selection occurs if we allow $\tau_k^2 = 0$ for some k. We place an inverse Gamma distribution on each τ_k^2,

$$P(\tau_k^2|a,b) \propto (\tau_k^2)^{-a-1}e^{-b/\tau_k^2}.$$

This prior allows some control over the magnitude of the learned feature weights, τ_k^2. Finally, an inverse-Wishart prior on the covariances stabilizes the cluster learning:

$$P(\Lambda_{z,x}|\alpha,\Lambda^*) \propto |\Lambda_{z,x}|^{-(\alpha+N_z+1)/2}e^{-\frac{1}{2}tr(\alpha\Lambda^*\Lambda_{z,x}^{-1})},$$

where Λ^* is the covariance of all the data, and α is a parameter which dictates the expected size of the clusters (the intra-class distance).

2.5. Clustering flow fields

We learn the parameters of the mixture of Gaussians from data using the expectation-maximization (EM) algorithm, which maximizes the expected log-posterior

$$\sum_{\mathbf{X}} \int_{\mathbf{Z}} P(\mathbf{XZ}|\nabla\mathbf{f}, \Theta') \log P(\nabla\mathbf{f}\mathbf{XZ}\Theta) \qquad (13)$$

over the model parameters, Θ, where Θ' are the current estimates. The complete set of parameters in the model is therefore $\Theta = \{\mu_z, \Lambda_z, \Theta_x, \tau_k, \Lambda_1, \Lambda_2, \Lambda_p, \alpha, a, b\}$. While $\{\mu_z, \Lambda_z, \Theta_x, \tau_k\}$ are learned from data, $\{\Lambda_1, \Lambda_2, \Lambda_p, \alpha, a, b\}$, are fixed. The bold face variables indicate sets of variables, $\mathbf{X} = \{X_1, ...X_{N_t}\}$, where N_t is the number of flow fields, and similarly for $\mathbf{Z}$ and $\nabla\mathbf{f}$. The EM algorithm alternates between "E" and "M" steps until the increase in the log-posterior becomes smaller than some convergence threshold. The expectation, or "E", step of the EM algorithm is the calculation of the posterior according to Equation (10), using the current model parameters, Θ'.

The "M" step is then to maximize Equation (13) over Θ, for which we can find analytical expressions by taking derivatives. The update equations differ from those for a standard mixture of Gaussians with feature weighting [6] because of the integrations over z. To derive the EM update equations, we only perform the integrations at the end. To update the output mean, we set the derivative with respect to the mean for state $X = i$, $\mu_{z,i}$, to zero. The derivative picks out the $X_k = i$ (written $X_{k,i}$) terms from the sum over i, leaving,

$$\sum_{k=1}^{N_t} \int_{z_k} P(X_{k,i} z_k | \nabla f_k \Theta') \frac{\partial}{\partial \mu_{z,i}} \log P(z_k | X_{k,i}) = 0. \qquad (14)$$

The derivative gives $-2\Lambda_{z,i}^{-1}(z_k - \mu_{z,i}) + T^{-1}(\mu_{z,i} - \mu^*)$, and so we can solve for $\mu_{z,i}$

$$\mu_{z,i} = (\xi_{\cdot,i}\Lambda_{z,i}^{-1} + T^{-1})^{-1} \left[\Lambda_{z,i}^{-1} \left(\sum_{k=1}^{N_t} \tilde{\mu}_{z,x} \xi_{k,i} \right) + T^{-1}\mu^* \right]$$

where $\xi_{k,i} = P(X_{k,i}|\nabla f \Theta')$ and $\xi_{\cdot,i} = \sum_{k=1}^{N_t} \xi_{k,i}$. Thus, the most likely mean for each state x is the weighted sum of the most likely values of z as given by Equation (9). Dimensions of the means, $\mu_{z,i}$, with small feature weights, τ_k^2, will be biased toward the data mean, μ^*, in that dimension. This is reasonable, because such dimensions are not relevant for clustering, and so should be the same for any cluster, X. The updates to the feature weights, τ_k, are

$$\tau_k^2 = \frac{b}{a + N_x/2 + 1} + \frac{1}{2a + N_x + 2} \sum_{i=1}^{N_z} (\mu_{z,i,k} - \mu_k^*)^2$$

where $\mu_{z,i,k}$, μ_k^* are the k^{th} dimensions of $\mu_{z,i}$ and μ^*, respectively. The updates to the feature weights show that those dimensions, k, with $\mu_{z,i,k}$ very different from the data mean, μ_k^*, across all states, will receive large values of τ_k^2, while those with $\mu_{z,i,k} \sim \mu_k^*$ will receive small values of τ_k^2. Intuitively, the dimensions along which the data is well separated (large inter-class distance) will be weighted more.

The updates to the covariance matrix, $\Lambda_{k,i}$, are found in a similar way, giving

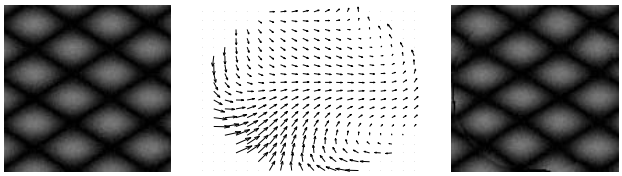
$$\Lambda_{z,i} = \frac{\sum_{k=1}^{N_t} (\tilde{\Lambda}_{z,x} + \tilde{\mu}_{z,x}\tilde{\mu}'_{z,x})\xi_{k,i} - \mu_{z,i}\mu'_{z,i}\xi_{\cdot,i} + \alpha\Lambda^*}{\xi_{\cdot,i} + \alpha + N_m + 1}.$$

The prior covariance is represented with $\alpha\Lambda^*$, which stabilizes the updates, avoiding matrix singularity problems in the inverses in Equation (10). The updates to the prior over x are given by $\Theta_x = \sum_{k=1}^{N_t} P(X_{k,i}|\nabla f) / \sum_{i=1}^{N_x} \xi_{\cdot,i}$.

3. Experiments

3.1. Probabilistic Projections

We performed two experiments to examine the advantages of using the probabilistic projection described by Equation (12) over the naïve projection given by $z = M\mu_v$. In the first, flow fields were reconstructed from 500 20 dimensional Zernike vectors, with all coefficients randomly generated in the interval $[-1, 1]$. The resulting flows (< 5 pixels/frame) were used to warp a synthetic 120×120 image using linear interpolation. The original image, shown in Figure 3(a), is a sine grating with added Gaussian noise $\sigma = 5$ greyscale values. An additional amount of Gaussian noise ($\sigma = 5$ again) was added after the warp. Figure 3 (b) and (c) show and example flow field and the corresponding warped image, respectively. Optical flow was



(a) original (b) flow field (c) warped

Figure 3: Synthetic images and flows.

projected to the Zernike basis using both naïve and probabilistic methods. The probabilistic projection used a single x state with a zero mean prior, $\mu_{z,x} = 0$, and a diagonal covariance $\Lambda_{z,x} = \sigma_{z,x} I$, with $\sigma_{z,x} = 0.01$. The coefficients were compared with the ground truth. The mean Euclidean distances were 1.08 ± 0.32 for the probabilistic projection, and 1.55 ± 0.37 for the naïve projection. The mean difference (naive-probabilistic) was 0.48 ± 0.13, showing that the probabilistic projection significantly outperforms the naïve method, as expected.

Our second experiment used the synthetic Yosemite flow-through sequence, constructed from an aerial image

and a depth map [1]. Ground truth over the ground region is used to evaluate the performance of the projection methods. There are 14 316x252 frames in the sequence, and the flow fields range up to 4 pixels/frame. The ground sections of two frames are shown in the top row of Figure 4, while the middle row shows the ground truth and the estimate of the flow field using the method of [13]. We pre-smoothed each image using separable Gaussian filters ($\sigma = 1.0$), and used a 3-level Gaussian pyramid. The noise parameters were set to $\sigma_1 = 0.08$, $\sigma_2 = 1.0$, $\sigma_p = 10.0$, $\sigma_0 = 0.5$ and $\sigma_d = 0.1$. The angular error [2] on the flow estimate is 8.4 ± 12.0. The first 10 Zernike coefficients were

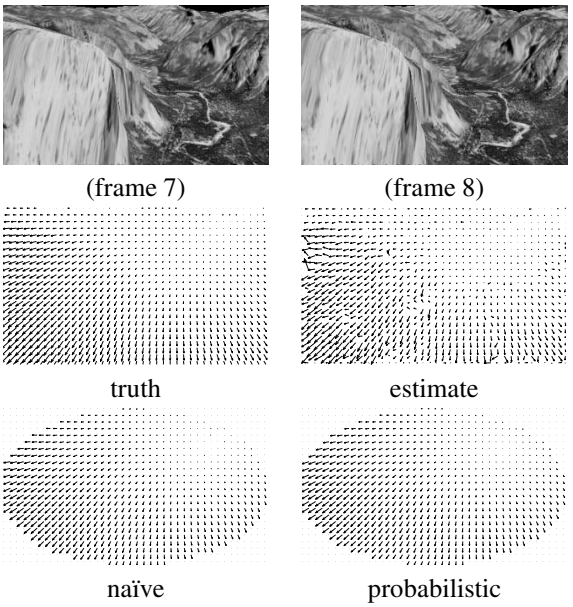


Figure 4: Top: two frames from the Yosemite sequence. Middle: Ground truth flow and estimate using [13]. Bottom: reconstructed flow fields.

estimated for all frames using both naïve and probabilistic projections, and were used to reconstruct flow fields over an elliptical region covering the rigid portion of the scene using Equation 4. We used a single zero-mean $\mu_z = 0.0$ model with diagonal covariance $\Lambda_z = 0.001$. The bottom row in Figure 4 shows the reconstructed flow fields for the two projections. The average angular errors over all frames for the reconstructions were 6.27 ± 6.36 for the naïve and 5.66 ± 6.22 for the probabilistic projections. Again, we see the advantage of the probabilistic projection.

3.2. Clustering

We clustered a set of 904 frames from a 3600 frame sequence of a person performing 4 different facial expressions. The subject was imitating an on-screen cartoon face

[2] The angular error, E, between ground truth v_c and estimate v_e is $E = \arccos(v_c \cdot v_e)$, where $v = \frac{1}{\sqrt{u^2+v^2+1}}(u, v, 1)^T$

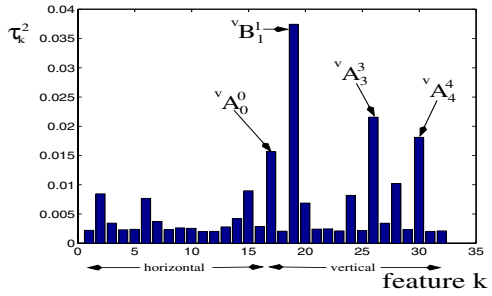


Figure 5: Feature weights learned for facial expression data.

which was displaying 4 prototypical expressions: happy, sad, surprised and angry. The person's face was tracked using a simple flow-based tracker. Our models are all applied over the tracked region in each image. It is the same dataset used in [8], to which we refer for further details. The video frames in this data set are not labeled, and so the analysis is qualitative: our methods *discover* clusters of optical flow fields, and we *interpret* these clusters, which can be related to established high-level concepts.

We automatically selected 904 frames which had significant motions in them by thresholding the mean magnitude of the optical flow. Applying our methods to all 3600 frames does not substantially change the result, since the flow fields from the other frames all fall close to the origin, and so are represented by one of the learned clusters. We used the first 16 Zernike coefficients for each horizontal and vertical flow, resulting in a 32-dimensional basis vector, z. The noise parameters were set to $\sigma_1 = 0.08$, $\sigma_2 = 1.0$, $\sigma_p = 10.0$, $\sigma_0 = 0.5$ and $\sigma_d = 0.1$. The feature weighting parameters were set to $a = 1$, $b = 0.01$ and $\alpha = 34$. The parameters $\mu_{z,x}$ and $\Lambda_{z,x}$ were initialized by choosing K data points randomly as the initial seeds for K-means clustering, and Gaussian distributions were fit to the resulting classes. The feature weights τ_k^2 were all initialized to 1. The results were relatively insensitive to the initialization.

We trained a model with 8 classes. Figure 5 shows the final values of the feature weights, τ_k^2. The first 16 dimensions are the Zernike coefficients corresponding to horizontal flow ($^uA_n^m, ^uB_n^m$ for $n < 5$), while the last 16 are those corresponding to vertical flow ($^vA_n^m, ^vB_n^m$ for $n < 5$), ordered by increasing n and m values. The feature weights are clearly favoring the vertical flows, because a major component of the facial expressions are raising and lowering of eyebrows. The four most relevant features are $\{^vB_1^1, ^vA_3^3, ^vA_4^4, ^vA_0^0\}$. There are six other moderately relevant features. The remaining 22 features are irrelevant.

Figure 6 shows the reconstructed Zernike vectors plotted along two of the relevant features ($^vB_1^1, ^vA_0^0$). The clusters are denoted by the shape and color of the data points. Reconstructed optical flow fields (using Equation 11) are

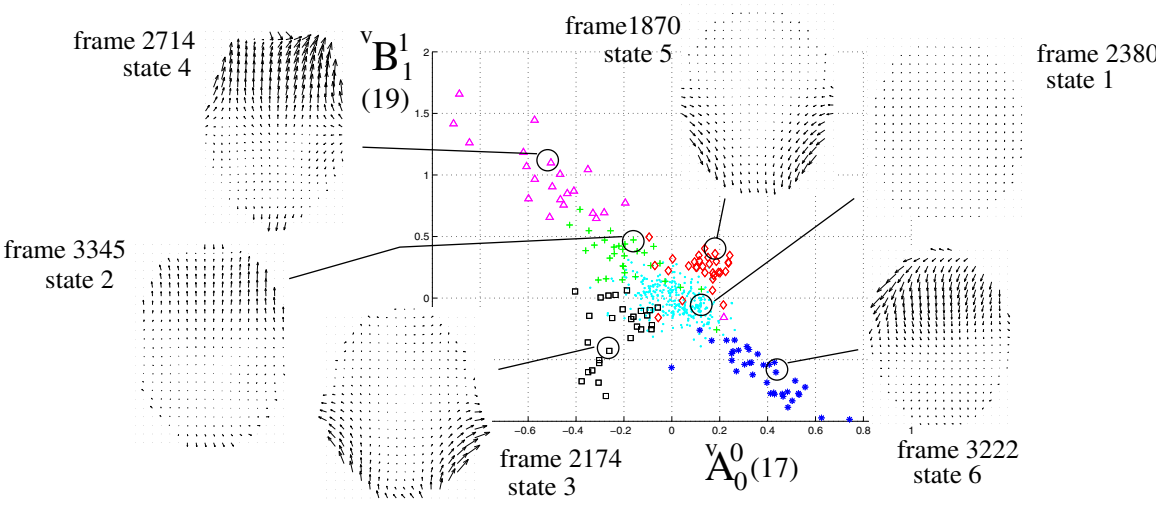


Figure 6: Clustering result for facial expressions along two most relevant features.

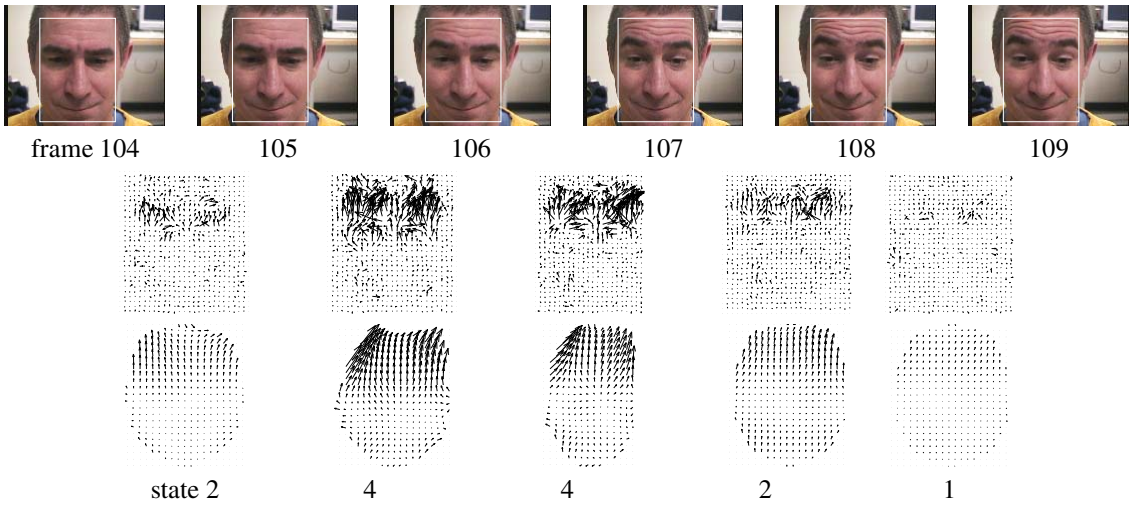


Figure 7: Eyebrow raising classified as states 2 and 4. The corresponding eyebrow lowering is shown in Figures 9 and 10.

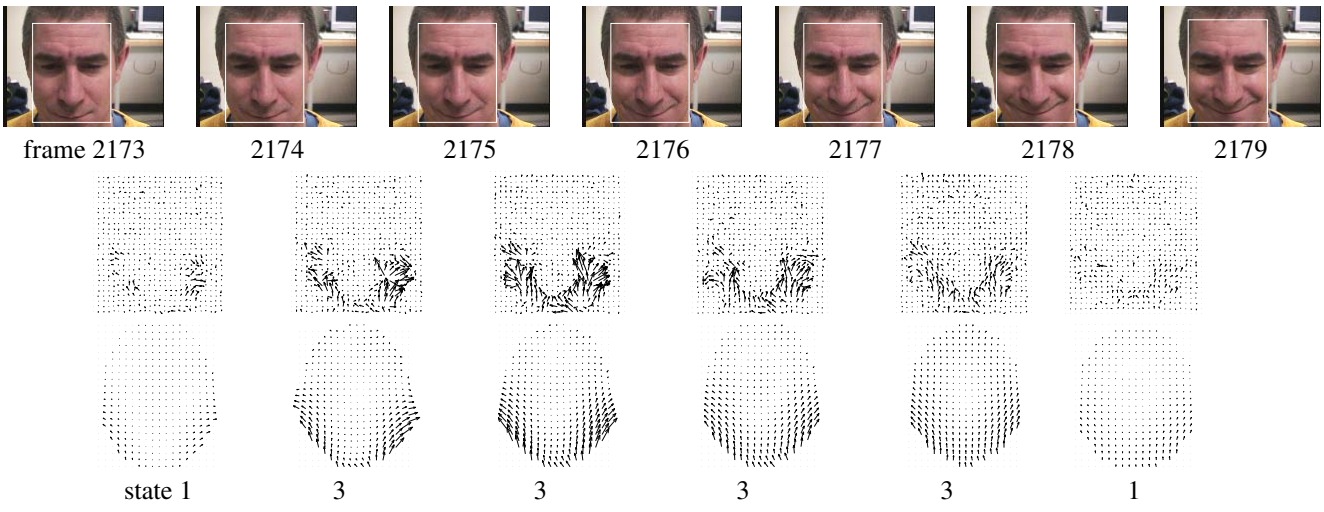


Figure 8: Smiling event classified as state 3, surrounded by no motion (state 1) events.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

shown for representative frames within each cluster. The classes are roughly (1) little or no motion, (2) eyebrows raising slowly, (3) jaw expanding, (4) eyebrows raising quickly, (5) jaw relaxing and (6) eyebrows lowering. The remaining two classes corresponded to translational motions (7) up and (8) down. However, these clusters only accounted for a small fraction of the data. We do not consider them further here. Figure 7 shows an example of a raising eyebrows event. The flow fields and the reconstructions from the model states are shown. The two central flow fields (105-106-107) are detected as state 4 (eyebrows raising rapidly), surrounded by more slowly raising eyebrow motions (state 2). Once the eyebrows reach their apex, the state returns to 1 (no motion) by frame 108.

Figure 8 shows a smiling event detected as state 3 from frame 2174-2178, surrounded by state 1 events (no motion). Figures 9 and 10 show the sequel to Figure 7, in which the subject's face returns to neutral. He begins by lowering his eyebrows (Figure 9, frames 115-117), which is classified as state 6, followed by a relaxation of his smile (Figure 10, frames 162-164), which is classified as state 5.

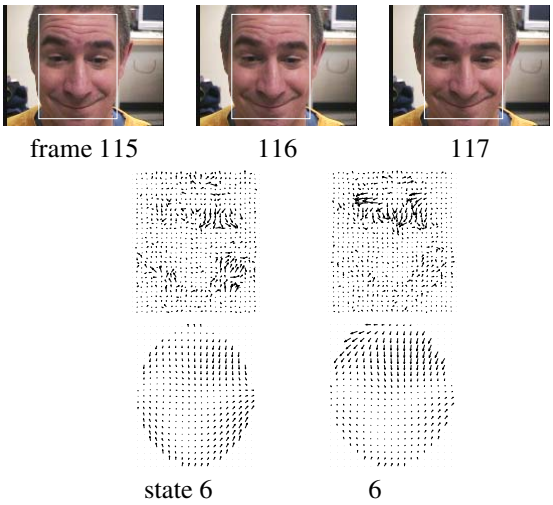


Figure 9: Eyebrow lowering event classified as state 6.

4. Conclusions

We have demonstrated a method for learning the parameters of a model for clustering flow fields with feature weighting. Our results show that the method can be used to discover the salient categories of instantaneous motions in video. We have tested our methods on other data sets, and found interpretable results. We are extending our methods to include a temporal dynamical process over the cluster variable, X, resulting in a feature weighted hidden Markov model (HMM), and higher level variables, leading to mixtures of HMMs.

Acknowledgements: Supported by the Institute for Robotics and Intelligent Systems (IRIS), a Canadian Network of Centres of Excellence, and by a Precarn scholarship. The authors thank Nando de Freitas, Don Murray and our anonymous reviewers.

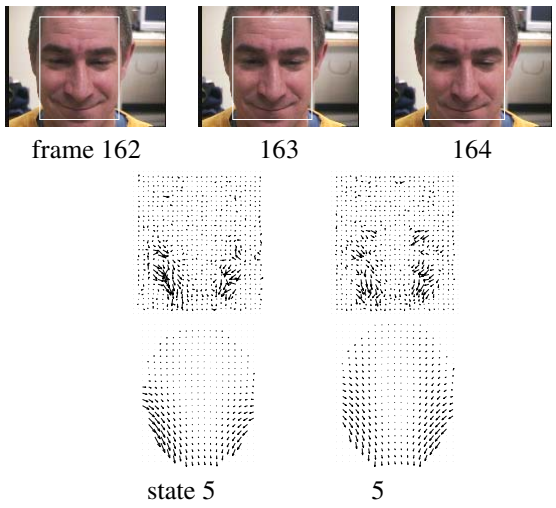


Figure 10: Smile returning to neutral classified as state 5.

References

[1] J. Barron, D. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994.

[2] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.

[3] M. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. *CVIU*, 63(1):75–104, January 1996.

[4] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motions. *IJCV*, 25(1):23–48, 1997.

[5] C. Bregler. Learning and recognising human dynamics in video sequences. In *IEEE CVPR* pages 568–574, 1997.

[6] P. Carbonetto, N. de Freitas, P. Gustafson, and N. Thompson. Bayesian feature weighting for unsupervised learning with application to object recognition. In C. M. Bishop and B. J. Frey, editors, *AI-STATS*, Key West, Florida, January 2003.

[7] I. A. Essa and A. O. Pentland. Facial expression recognition using a dynamic model and motion energy. In *ICCV*, pages 360–367, Cambridge, Mass., 1995.

[8] J. Hoey. Clustering contextual facial display sequences. In *Proceedings of IEEE International Conference on Face and Gesture*, Washington, DC, May 2002.

[9] J. Hoey and J. J. Little. Representation and recognition of complex human motion. In *Proc. IEEE CVPR*, Hilton Head, SC, June 2000.

[10] J. J. Little and J. Boyd. Recognizing people by their gait: the shape of motion. *Videre*, 1(2), Winter 1998.

[11] D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Proceedings ICCV'98*, pages 696–702, January 1998.

[12] A. Prata and W. Rusch. Algorithm for computation of Zernike polynomials expansion coefficients. *Applied Optics*, 28(4):749–754, February 1989.

[13] E. Simoncelli, E. Adelson, and D.J.Heeger. Probability distributions of optical flow. In *Proceedings of CVPR*, pages 310–315, Mauii, Hawaii, USA, 1991.