Robust Face Recognition using Deformable 2D Models

Anne Jorstad
AMSC Candidacy Prospectus
April 2009

Face Recognition is a widely studied problem within the field of Computer Vision, and has many real-world applications including security and human-computer interaction. Much work has been done to interpret face images across variations in head pose and scene lighting. The third somewhat less explored variable of face recognition is variation in facial expression. Changes in expression are generally less extreme than potential variations in pose, but they can still be problematic. The most successful algorithms that handle expression variation rely on comparisons between some form of morphable model of a face. In my research, I will attempt to develop a more robust mathematical model of the face than those commonly used, in order to more accurately measure moderate variations in expression and pose.

The face recognition problem I consider involves identifying an unknown individual pictured in a single 2D image of reasonably good quality, given a database of known 2D face images. This involves defining a metric between the new image and all images in the database, and identifying the unknown individual with the identity of the closest known image. Each face will be modeled as a deformable structure, and distances between faces can be described by an energy minimization problem depending on how closely a new face can be deformed to match a known face and how much deformation is required to achieve this similarity.

Before faces are compared, some image preprocessing is assumed. It is often a good idea to normalize the intensities of an image, for example so that the pixel intensities have zero mean and unit variance. Commercial face recognition systems can be relied upon to consistently locate a small number of facial feature points in images, such as the centers of the eyes and the center of the mouth. From this small number of points, the faces can be aligned to minimize the sum of squared distances between the feature points on an individual face and the average locations of these points. This effectively removes image rotation, translation and scaling from the problem.

Many traditional face recognition algorithms directly compare image intensities after performing only the above image transformations. In the simplest case, after two faces are aligned, the difference in intensities at each pixel location is computed, and these differences are summed, resulting in the overall image difference. This method is very sensitive to even slight variations in pose, lighting and expression. Two very common algorithms applied to face recognition that do not perform any image deformations are known as Eigenfaces and Fisherfaces [7]. These methods treat an $m \times n$ image of a face as an $mn \times 1$ vector, and face point correspondences are based purely on location in the image. The Eigenface method uses Principal Component Analysis (PCA) to extract a meaningful set of face image vectors that can act as a basis for the space of face images, maximizing the overall scatter of the data. The Fisherface method uses Linear Discriminant Analysis (LDA) to maximize the ratio of between-class scatter to within-class scatter, providing a basic classification scheme. These methods are able to weight some parts of an image more heavily than others, removing some sensitivity to lighting and expression variation. Such methods can be expanded using a Bayesian probability model to predict how a new face will fit in with existing data, such as in [10]. But all these methods assume a correspondence between pixels in different images that happen to be at the same coordinate location. For more general purpose face recognition, it is desirable to allow unknown face images to be meaningfully deformed to more closely match known faces.

One way to model the variability of real-world data is through statistical structures, as is done in [8]. The method of Active Appearance Models learns the allowable variation in shape from the training set, using PCA to extract the primary directions of this shape variation. An individual image consisting of feature points x with intensity values g can be modeled as

$$x = \bar{x} + Q_s c \tag{1}$$

$$g = \bar{g} + Q_g c, \tag{2}$$

where $\bar{x}$ and $\bar{g}$ are the mean shape values and intensity values, Q_s and Q_g are the matrices describing the modes of variation from the image data, and c is the parameter vector controlling the influence of each mode. To compare a novel image to the database, a synthetic images is generated using the allowable variations in shape and intensity. A coarse initial prediction is produced, then at each iteration the difference between the current guess and the novel image is calculated, and the model parameters are updated to provide a better fit. This method requires a dense correspondence between all feature points in all images. The method successfully captures the consistent principal variation in shape and intensity of the known dataset, but it cannot handle any new types of variability.

Anatomically-based models of the human face have been constructed to model realistic changes in facial expression, including the method of [12]. Here, the authors implement a 3D face model with anatomically-based facial tissue and muscle control. The model is able to reproduce expressions found in image sequences, and it can be used to generate synthetic images to closely match novel images, which can then be used for comparison. However, the system can only reproduce expressions from images when the salient features have been highlighted, and this extensive computational model has not been shown to be any more effective than much simpler strictly 2D algorithms.

One of the first successful 2D morphable models developed for handling facial expression variation is found in [3]. Here Lades et al. define the Dynamic Link Architecture, an extension of the classical Neural Networks scheme used in many Artificial Intelligence applications. A uniform 7×10 grid of nodes is placed over the face, where each node is a feature vector or "jet" defined by a Gabor wavelet convolution with the image over five scales. To match a new image to a model image, the best rigid alignment is found, then each node is displaced locally to find the location where it is the most similar to a corresponding model node, by maximizing

$$S_v(J^I, J^M) = \frac{\langle J^I, J^M \rangle}{\|J^I\| \|J^M\|} \tag{3}$$

for a each image jet J^I and model jet J^M. The relation to neighboring nodes is also considered, minimizing the distortion of node x_i relative to its neighbors x_j where $j \in V$ the vertex set and $(i, j) \in E$ the set of graph edges. With the distance between nodes defined as

$$\Delta_{ij} = x_j - x_i, \tag{4}$$

minimizing distortion is equivalent to minimizing

$$S_e(\Delta_{ij}^I, \Delta_{ij}^M) = \left(\Delta_{ij}^I - \Delta_{ij}^M\right)^2. \tag{5}$$

The total cost to be minimized at each node is then a combination of these two terms, weighted by distortion penalty λ:

$$C(x_i^I) = \lambda \sum_{(i,j) \in E} S_e(\Delta_{ij}^I, \Delta_{ij}^M) - \sum_{i \in V} S_v(J^I(x_i^I), J_i^M). \tag{6}$$

This method separates global position information from relational information, allowing local distortions. Small variations in expression and pose are successfully captured with this algorithm, but it breaks down when occlusions are introduced. This method considers both the similarity of a transformed image and the amount of transformation required to achieve this similarity.

Felzenszwalb and Huttenlocher set up a model similar to the Dynamic Link Architecture in [9] in order to detect faces in images, but instead of using a uniform grid of nodes, they consider connections between a smaller set of more meaningful "parts". For a face, the "parts" correspond to the center of the eyes, nose, and corners of the mouth. They are represented as 27-dimensional feature points constructed using Gaussian derivative filters of different orders, orientations and scales, and their relations to one

another are models as spring-like connections, allowing for variation in the relative locations of the parts. A face is found in an unknown image by determining the best fit of the pictorial structure model L to the image:

$$L^* = \underset{L}{\operatorname{argmin}} \left(\sum_{i \in V} m_i(\ell_i) + \sum_{(i,j) \in E} d_{ij}(\ell_i, \ell_j) \right), \tag{7}$$

where $m_i(\ell_i)$ is the degree of mismatch when a part v_i is placed at location ℓ_i, and $d_{ij}(\ell_i, \ell_j)$ is the degree of deformation of the model between parts v_i and v_j, measured by the Mahalanobis correlation distance

$$d_{ij}(\ell_i, \ell_j) = \left(T_{ij}(\ell_i) - T_{ji}(\ell_j)\right)^T \Sigma_{ij} \left(T_{ij}(\ell_i) - T_{ji}(\ell_j)\right). \tag{8}$$

Here $T_{ij}(\ell_i)$ is the transformed location from starting point ℓ_i, and Σ_{ij} is a diagonal weighting term. This algorithm uses a deformable learned model of a face to find the best location of a face in an unknown image. However, the algorithm is not suitable for distinguishing between individual faces, as it models all faces as a single structure, and not enough discriminative information is present to narrow this identification down any further. This method again considers both the similarity of a transformed image and the amount of transformation required to achieve this similarity.

For more robust algorithms, we would like to consider more than just a small subset of feature points, and we seek a fully dense correspondence between every point in each image. The most common automatic method for finding dense image correspondences is the optical flow algorithm. Traditional optical flow [6] estimates the motion between two images by determining the displacement of every pixel in the first image to the most similar pixel in the second image. It is assumed that both images are taken by a single moving camera at times t and $t + \delta t$ so that the visible scene is similar, and it is assumed that intensity is preserved between corresponding patches of the images, so that $I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t)$. For a 2D image I, using a first order Taylor expansion,

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t \tag{9}$$

$$0 = \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t \tag{10}$$

$$0 = \frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t}. \tag{11}$$

This is the fundamental equation of optical flow. The image derivatives $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are calculated using finite differences of the pixel values. A second constraint must be added in order to explicitly solve for v_x and v_y, and this is can be accomplished in many ways. It is common to minimize the overall norm of the vector field (Horn and Schunk), or to assume the flow is constant within each small region of the image (Lucas and Kanade). The vector $[v_x, v_y]^T$ defines the optical flow at each pixel, providing a dense corresponding from pixels in the first image to locations in the second. Optical flow is used successfully in many face recognition systems that attempt to handle variations including expression.

Beymer and Poggio use optical flow in [1], separating the information in an image into a texture vector storing intensity values of each pixel, and a corresponding shape vector storing the (x, y) displacement vectors of each pixel, as compared to a common reference image. A shape-free representation for each individual is generated by warping all faces to a common view where features line up exactly. In this representation, the only variations between images is texture, so direct texture differencing can be calculated to determine the similarity between images. In order to perform such a warp, the transformation from a new image to a trained pose must be known. For each pixel j, this transformation to a new pose r is determined by $\left(y_{p_j,r} - y_{p_j}\right)$, a vector defining the displacement from the pixel location in the given pose y_{p_j} to its location in the desired pose $y_{p_j,r}$. Given a new face image y_n, the synthetic image of this face in pose r is then determined by

$$y_{n,r} = y_n + \left(y_{p,r} - y_p\right), \tag{12}$$

warping the location of each pixel of the new image, as defined by the transformation learned from the training images. The dense pixel-by-pixel transformation $(y_{p,r} - y_p)$ is found using optical flow. This method effectively generates synthetic images by applying learned transformations to new images, going a step beyond the Dynamic Link Architecture in [3] which involves no trained decision making. The method has trouble with regions of the image which should be visible in the synthetic view but not the original view, and it is limited in the size of the deformations it can handle by the local nature of the optical flow algorithm. In this paper, 15 views are generated for each individual varying the pose of the head, and a novel image is compared to each view of each person to determine the best match, effectively handling moderate variations in pose. Although expression variation is not discussed in this paper, it could be treated in exactly the same manner given training images spanning known variations in expression. The final matching decision of this algorithm depends only on the texture values of a synthetic image after the warp, and it does not measure how much an image must be warped in order to closely match an image in the database.

In [2], Blanz and Vetter develop an algorithm similar to that of Beymer and Poggio, using a statistical 3D model instead of several 2D images. Laser scans are used to collect 3D models of many different faces, and a dense correspondence between these 3D training models is found using a 3D version of the optical flow algorithm. Once this correspondence is found, shape and texture information can be completely separated as in [1]. PCA is performed on the intensity information T, one value at each point, and on the shape information S, a 3D vector describing how a specific point differs from the model average of that point. This gives m significant eigenvectors representing the texture and shape information, and an individual face can be modeled as

$$s = \bar{s} + \sum_{i=1}^{m-1} \alpha_i S_i, \tag{13}$$

$$t = \bar{t} + \sum_{i=1}^{m-1} \beta_i T_i. \tag{14}$$

This means that a face is completely defined by the model coefficients $\vec{\alpha}$ and $\vec{\beta}$ that determine the influence of each principal component in S and T. To determine $\vec{\alpha}$ and $\vec{\beta}$ for a new face, a synthetic image of the model is generated that minimizes the sum of squared distances (SSD) between the intensity of each pixel in the new image and the corresponding pixel in the synthetic image. A probability model is setup to maximize

$$p\left(\vec{\alpha}, \vec{\beta}, \vec{\rho} \mid I_{input}, F\right), \tag{15}$$

where ρ is the parameter determining the pose of the face in the image, I_{input} is the new image, and F is a small set of feature points found on the face during preprocessing to determine the initial approximate alignment of the model (the corners of the eyes, center of the nose, and corners of the mouth). This probability is solved using Bayes rule and prior probabilities calculated from the training set. With $\vec{\alpha}$ and $\vec{\beta}$ determined, the full 3D model of the new face is determined, and the coefficients are compared to every known set of coefficients in the training set. Setting $\vec{c_k}$ to be the single vector of all model coefficients for face k, it was found that the covariance-based distance function

$$d_W = \frac{\langle c_1, C_W^{-1} c_2 \rangle}{\|c_1\|_W \|c_2\|_W}, \tag{16}$$

where C_W is the covariance matrix, provided the most accurate results. This algorithm does not compare pixel intensities after one image has been warped to correspond with another, as is done in [1]; instead it compares model coefficients. But similar to Beymer and Poggio, the amount of transformation required to transform an example to fit the model is not considered in the final model comparison. This method is often considered the state-of-the-art in face recognition, given good quality images in a semi-controlled environment.

Martínez has developed several algorithms using a morphable model to capture variation in facial expression. In [4], the optical flow is computed between a new image T and every image I_j in the training set

$$F_j = \text{OpticalFlow}(I_j, T), \tag{17}$$

and then the amount of change at each pixel is considered. Small changes are weighted heavily and large changes are weighted lightly, to emphasize invariants. Several weighting schemes are presented, with the most successful being the simple linear model

$$w_j = max(\|F_j\|) - \|F_j\|, \tag{18}$$

where $max(\|F_j\|)$ is the maximal magnitude of the flow over all pixels in F_j. This defines the weight of the optical flow at each pixel. The cost to transform I_j to T is then

$$C = \|W_j(I_j - T)\|, \tag{19}$$

and the best match to the novel image T is the training image I_j that minimizes this cost. This algorithm makes its final matching decision again based on the similarity of a transformed image, but this similarity is weighted by how much the image must be transformed at each pixel. The algorithm references the amount of deformation required to warp one image to the other, but it does not include a full measure of this deformation.

A successful face recognition system should be based on a cost function that measures both the similarity between images and the amount of transformation required to attain this similarity. The algorithms of [3] and [9] each incorporate both these costs into their final decision model, as can be seen in equations (6) and (7). The methods of [1], [2] and [4] explore measurements of the similarity of transformed images, with [4] also considering the amount of warping required for matching. To develop a more robust model, I will explore the following problem: Given two images $I(x)$ and $J(x)$, define a transformation v that warps $I(x)$ to $I(v^{-1}(x))$, an image similar to $J(x)$, and define a metric $\|\cdot\|_g$ that measures this transformation. Then the similarity between images can be defined by the following distance cost function:

$$d(I, J) = \min_v \|J(x) - I(v^{-1}(x))\|_{L_2} + \lambda\|v\|_g \tag{20}$$

for weighting constant λ that adjusts the relative importance of the image similarity $\|J(x) - I(v^{-1}(x))\|_{L_2}$ with the cost of the deformation $\|v\|_g$. For many applications, it makes sense to allow the relative influence to change with the data, and the most effective way to combine these values can be learned using a standard Support Vector Machine algorithm such as in [11]. A Support Vector Machine finds a binary classification model of a set of potentially very high dimensional data by finding the class boundary hyperplane that maximizes the margin of separation between pre-classified training points and the hyperplane. To pose face recognition as a binary classification problem, an unknown face is determined to be either similar or not similar to a known faces.

An image transformation and transformation metric are presented in a robust mathematical framework in [5]. Here, an image is represented as a continuous Riemannian manifold through the pixels. A Lie group can be defined on the image as diffeomorphisms of the manifold, which determine the possible transformations of the image. The corresponding Lie algebra is the vector space of infinitesimal steps in the direction of these transformations, that is, continuous vector fields deforming the manifold. The best transformation v can then be defined as a geodesic through these vector fields. From all possible transformations, the geodesic can be thought of as the one requiring the least cost. This is a potentially more robust algorithm than the variations of optical flow currently used to solve the dense correspondence problem, and is still fully automatic. This mathematical framework has been successfully applied to a small number of face images, defining a continuous morphing between faces in different poses. I propose to expand this algorithm to a full face recognition system handling moderate variations in expression and pose.

Primary Material

[1] D. Beymer, T. Poggio. "Face Recognition From One Example View." Proceedings of the Fifth International Conference on Computer Vision (ICCV), 1995.

[2] V. Blanz, T. Vetter. "Face Recognition Based on Fitting a 3D Morphable Model." IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 25, 2003.

[3] M. Lades, J. Vorbrüggen, Joachim Buhmann, Jörg Lange, Chrostoph v.d. Malsburg, Rolf Würtz. "Distortion Invariant Object Recognition in the Dynamic Link Architecture." IEEE Transactions on Computers, vol. 42, 1993.

[4] A. Martínez. "Recognizing Expression Variant Faces from a Single Sample Image per Class." IEEE Computer Vision and Pattern Recognition (CVPR), vol. 1, 2003.

[5] A. Trouvé, L. Younes. "Metamorhposes Through Lie Group Action." Foundations of Computational Mathematics, 2004.

Secondary Material

[6] J. Barron, D. Fleet, S. Beauchemin. "Performance of Optical Flow Techniques." International Journal of Computer Vision (IJCV), vol. 12, 1994.

[7] P. Belhumeur, J. Hespanha, D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection." IEEE Transactions on PAMI, vol. 19, 1997.

[8] T. Cootes, G. Edwards, C. Taylor. "Active Appearance Models." IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 1998.

[9] P. Felzenszwalb, D. Huttenlocher. "Pictorial Structures for Object Recognition." International Journal on Computer Vision (IJCV), vol. 61, 2005.

[10] B. Moghaddam, T. Jebara, A. Pentland. "Bayesian Modeling of Facial Similarity." Advances in Neural Information Processing Systems, vol. 11, 1999.

[11] P. Phillips. "Support Vector Machines Applied to Face Recognition." Advances in Neural Information Processing Systems, vol. 11, 1999.

[12] D. Terzopoulos, K. Waters. "Analysis and Synthesis of Facial Image Sequences Using Physical and Anatomical Models." IEEE Transactions on Pattern Matching Analysis and Machine Intelligence, vol. 15, 1993.

Course Material

Primary Mathematical Content:

MATH 632 (at Wisconsin-Madison) - Introduction to Stochastic Processes: Markov chains, theory and applications.
MATH 703 (at Wisconsin-Madison) - Methods of Applied Math 1: Linear algebra, analytical differential equations, fluid dynamics.
MATH 712 (at Wisconsin-Madison) - Methods of Computational Math 1: Finite difference methods, initial and boundary value problems, stability.
MATH 713 (at Wisconsin-Madison) - Methods of Computation Math 2: Spectral methods, finite element methods, mesh-free methods.

Area of Application:

ENEE 631 - Digital Image Processing: 2D signal processing, sampling, transforms, compression.
CMSC 733 - Computer Processing of Pictoral Information: Extracting meaningful information from a single image, multi-view geometry.
CMSC 828 - Approaches to Representing and Recognizing Objects: Mathematical and algorithmic techniques for representing and recognizing objects in images.
COMP SCI 730 (at Wisconsin-Madison) - Nonlinear Programming Algorithms: Description and convergence proofs for numerical nonlinear optimization algorithms.
COMP SCI 766 (at Wisconsin-Madison) - Computer Vision: Survey course presenting most main topics of computer vision.