14.2 Face Recognition

Kirby and Sirovich (1990)

Face image can be compressed:

$$\tilde{x} = m + \sum_{i=0}^{M-1} a_i \mu_i$$
Scatter or Co-variance matrix:
\[ C = \frac{1}{N} \sum (x_j - m)(x_j - m)^T \]

Eigen Decomposition:
\[ C = U A U^T \frac{1}{N} \sum_{j=0}^{N-1} \lambda_j u_j u_j^T \]

Any arbitrary vector \( x \) can be represented:
\[ a_i = (x - m)u_i \]

The distance of a projected face DIFS (Distance in face space)
\[ \bar{x} = m + \sum_{i=0}^{M-1} a_i u_i \quad DIFS = \|\bar{x} - m\| = \sqrt{\sum_{i=0}^{M-1} a_i^2} \]

The distance between two faces
\[ DIFS = \|\bar{x} - \bar{y}\| = \sqrt{\sum_{i=0}^{M-1} (a_i - b_i)^2} \]

We are not utilizing the eigen value information, compute Mahalonobis distance
\[ DIFS' = \|\bar{x} - m\|_{\lambda^{-1}} = \sqrt{\sum_{i=0}^{M-1} \frac{a_i^2}{\lambda_i^2}} \]

Pre-scale the eigen vectors by eigenvalues:
\[ \hat{U} = U A^{-1/2} \]

The difference between a novel image and prototype image:
\[ \|x - x_k\| = \|a - a_k\| \quad \text{Saving in computation} \]
Images Taken under Different Illuminations

A space that discriminates between different classes (people) and is less sensitive to within-class variations

Fisher Faces (LDA)

Total within class scatter matrix

\[ S_W = \sum_{k=0}^{K-1} \sum_{C_k} (x_i - m_k)(x_i - m_k)^T \]

Between class scatter matrix

\[ S_B = \sum_{k=0}^{K-1} N_k (m_k - m)(m_k - m)^T \]

\( m \) is global mean

Direction which results in the Largest ratio between the projected between-class and within-class variations

\[ u^* = \arg \max_u \frac{u^T S_B u}{u^T S_W u} \]

Generalized eigenvalue problem

\[ S_B u = \lambda S_W u \quad \Rightarrow \quad \lambda u = S_W^{-1} S_B u \]
Bayesian Technique

Difference images

\[ \Delta_{ij} = x_i - x_j \]

\[ \Sigma_i = \text{intraperson co - variance} \]

\[ \Sigma_e = \text{extraperso - variance} \]

Distance \( \Delta \) between a new face \( x \) and a stored training image \( i \);

Intraperson likelihood

\[ p_i(\Delta_i) = p_N(\Delta_i, \Sigma_i) = \frac{1}{\sqrt{2\pi\Sigma_i}} \exp\left(-\frac{\|\Delta_i\|_2^2}{2\Sigma_i}\right) \]

The Mahalonobis distance:

\[ \|\Delta_i\|_\Sigma_i^{-1}^2 = \Delta_i^T \Sigma_i^{-1} \Delta_i = \|a' - a_i'\|_2^2 \]

Bayesian likelihood of a new image \( x \) matching a training image \( i \)

\[ p(\Delta_i) = \frac{p_i(\Delta_i)l_i}{p_i(\Delta_i)l_i + p_e(\Delta)_l} \]
Principle Components of Difference images

Difference images

\[ \Delta_{ij} = x_i - x_j \]

\[ \Sigma_i = \text{intraperson co - variance} \]

\[ \Sigma_e = \text{extraperson co - variance} \]

Modular Eigen Space

Have separate eigen spaces for each part of the face
Use this to technique to detect parts of the face
View-based Eigen space

(a)

Have separate eigen space for each view

(b)

8.3 Spline Based-Motion
• Pixel-wise optical flow (one pixel)

• Global flow (parametric flow) (all pixels)
  \[ u(x, y) = a_1 x + a_2 y + b_1 \]
  \[ v(x, y) = a_3 x + a_4 y + b_2 \]
  Affine

  \[ \mathbf{u} = \frac{A\mathbf{x} + \mathbf{b}}{C^T\mathbf{x} + 1} \]
  Projective

Spline-based Motion

\[ u(x_i, y_i) = \sum_j \hat{u}_j B_j(x_i, y_i) \quad \text{or} \quad u_i = \sum_j \hat{u}_j w_{ij}, \]

For 640 by 480 image, with \( m = 8 \), \( 81 \times 61 \times 2 = 10,000 \) parameters
Splines

1. block: \( B(x, y) = 1 \) on \([0, 1]^2\)

2. linear: \( B(x, y) = \begin{cases} 
(1 - x - y) & \text{on } [0, 1]^2, \\
(x + 1) & \text{on } [-1, 0] \times [0, 1], \text{ and}
\end{cases} \)
\( (y + 1) \text{ on } [0, 1] \times [-1, 0]. \)

3. linear on sub-triangles: \( B(x, y) = \max(0, 1 - \max(|x|, |y|, |x + y|)) \)

4. bilinear: \( B(x, y) = (1 - |x|)(1 - |y|) \) on \([-1, 1]^2\)

5. bi quadratic: \( B(x, y) = B_2(x)B_2(y) \) on \([-1, 2]^2\), where \( B_2(x) \) is the quadratic B-spline
Levenberg-Marquardet

SSD Error

\[ E(\{u_i, v_i\}) = \sum_i [I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i)]^2. \]

\[ u(x_i, y_i) = \sum_j \hat{u}_j B_j(x_i, y_i) \quad \text{or} \quad u_i = \sum_j \hat{u}_j w_{ij}, \]

\[ e_i = I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i) \]

\[ g_j^u = \frac{\partial E}{\partial u_j} = 2 \sum_i e_i G_i^u w_{ij}; \]

\[ g_j^v = \frac{\partial E}{\partial v_j} = 2 \sum_i e_i G_i^v w_{ij}; \]

\[ (G_i^u, G_i^v) = \nabla I_1(x_i + u_i, y_i + v_i) \]

Levenberg-Marquardet

SSD Error

\[ E(\{u_i, v_i\}) = \sum_i [I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i)]^2. \]

\[ u(x_i, y_i) = \sum_j \hat{u}_j B_j(x_i, y_i) \quad \text{or} \quad u_i = \sum_j \hat{u}_j w_{ij}, \]

\[ a_{jk}^{uu} = 2 \sum_i \frac{\partial e_i}{\partial u_j} \frac{\partial e_i}{\partial u_k} = 2 \sum_i w_{ij} w_{ik} (G_i^u)^2 \]

\[ a_{jk}^{vv} = a_{jk}^{vv} = 2 \sum_i \frac{\partial e_i}{\partial v_j} \frac{\partial e_i}{\partial v_k} = 2 \sum_i w_{ij} w_{ik} G_i^v G_i^v \]

\[ a_{jk}^{uv} = 2 \sum_i \frac{\partial e_i}{\partial v_j} \frac{\partial e_i}{\partial v_k} = 2 \sum_i w_{ij} w_{ik} (G_i^v)^2. \]

\[ (A + \lambda \text{diag}(A)) \Delta u = -g, \]
Results

Rueckert et al

\[ \Gamma(x) = \sum_{m=0}^{3} \sum_{l=0}^{3} B_m(v) B_l(u) \Phi_{i+l,j+l+m} \quad x = [x, y]^T \]

\[ B_0(t) = (1 - t)^3/6, \]
\[ B_1(t) = (3t^3 - 6t^2 + 4)/6, \]
\[ B_2(t) = (-3t^3 + 3t^2 + 3t + 1)/6, \]
\[ B_3(t) = (t)^3/6. \]

\[ i = \lfloor x/s_x \rfloor - 1, \quad j = \lfloor y/s_y \rfloor - 1, \quad u = x/s_x - \text{size } s_x \times s_y \]
\[ \lfloor x/s_x \rfloor, \quad \text{and } \quad v = y/s_y - \lfloor y/s_y \rfloor \]

\[ E(\{u_i, v_i\}) = \sum_i [I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i)]^2. \]

\[ SSD(M, T) = \frac{\sum_w (M(w) - T(w))^2}{w \times h}, \quad \text{NMI}(M, T) = \frac{H(M) + H(T)}{H(M, T)}, \]

Use Quasi Newton method to solve non-linear optimization
Results