Face Recognition

Xiaojun Qi

Step 1: Face Detection

- Different Color Spaces
  - RGB
  - YCbCr
  - HSI
  - UCS (Perceptually uniform color system)
- Histogram-based Approaches
  - Skin color distribution model
  - Hair color distribution model
- Morphological Operations

Step 2: Feature Extraction

- Edge Detectors
- Wavelet and wavelet packets
- Discrete Cosine transformation
- Gabor filters
- Several Important Statistics: Moments
- Shape/color features
- Fuzzified features

Step 3: Face Recognition

- Classification Techniques
  - PCA (Principal Component Analysis)
  - LDA (Linear Discriminant Analysis)
  - ICA (Independent Component Analysis)
  - SVMs (Support Vector Machines)
  - NN (Neural Network)
- Statistical Modeling
  - Bayes classifier
- Distance/Similarity Measures
  - Euclidean Distance
  - Bhattacharyya Distance
  - Manhattan Distance

Face Detection: Commonly Used Techniques

- Finding Faces in images with controlled background
- Finding Faces by Color
- Finding Faces by Motion
- Finding Faces using the Mixture of the Above
- Finding Faces in Unconstrained Scenes

Face Detection Technique 1: Finding Faces in Images with Controlled Background

- Use images with a plain monocolour background, or use them with a predefined static background
- Remove the background will always give the face boundaries.
Face Detection Technique 2: Finding Face by Color

- The advantage: If you have access to color images, you might use the typical skin color to find face segments.

- The disadvantage: It doesn't work with all kinds of skin colors, and is not very robust under varying lighting conditions.

Basic Color Extraction for Face Detection

- Color provides a computationally efficient yet effective method which is robust under rotations in depth and partial occlusions. It can be combined with other methods such as motion and appearance-based face detection.

- Human skin forms a relatively tight cluster in color space even when different races are considered.

- Face color distributions are normally modeled as Gaussian mixtures.

Construct the Skin Color Model

- Gaussian density functions and a mixture of Gaussians are often used to model skin color. The parameters in a unimodal Gaussian distribution are often estimated using maximum-likelihood.

- The motivation for using a mixture of Gaussian distribution is based on the observation that the color histogram for the skin of people with different ethnic background does not form a unimodal distribution, but rather a multimodal distribution.

Gaussian Mixture Estimation

- A Gaussian mixture model is defined as:

\[ p(x) = \sum_{k=1}^{K} p(x \mid k)P(k) \]

- Where p(x|k), k=1, ..., K, are K Gaussian density functions. The parameters in such a model are the means, \( u_k \), covariance matrices, \( \Sigma_k \), and mixing parameters, P(k). These can be estimated from a data set using a Expectation-Maximization (EM) algorithm.

A Few Statistics Terms -- Review

- The **conditional probability** of an event B in relationship to an event A is the probability that event B occurs given that event A has already occurred. The notation for conditional probability is: \( P(B \mid A) \)

\[ P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)} = \frac{P(A \mid B)P(B)}{\sum P(A \mid B)P(B)} \]

\[ P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} = \frac{P(B \mid A)P(A)}{\sum P(B \mid A)P(A)} \]
In the n-dimensional case, the Gaussian density of the vectors in the jth pattern class has the form:

\[
p(x / \omega_j) = \frac{1}{\sqrt{2\pi \sigma_j^2}} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} \quad j = 1, 2
\]

where each density is specified completely by its mean vector \( m_j \) and covariance matrix \( C_j \), which are defined as

\[
m_j = E\{x\} \quad \text{and} \quad m_j = \frac{1}{N_j} \sum_{k \in \omega_j} x
\]

\[
C_j = E\{(x - m_j)(x - m_j)^T\} \quad \text{and} \quad C_j = \frac{1}{N_j} \sum_{k \in \omega_j} (x - m_j)(x - m_j)^T
\]

---

**Expectation-Maximization Algorithm**

- Given a data set \( X = \{x_1, \ldots, x_m, \ldots, x_M\} \), EM aims to find parameter values that maximize likelihood or, equivalently, minimize the negative log-likelihood function.

\[
e = -\ln L = -\sum_{m=1}^{M} \ln p(x_m) = -\sum_{m=1}^{M} \ln \left( \sum_{k=1}^{K} p(x_m | k)P(k) \right)
\]

- It provides an effective maximum-likelihood algorithm for learning a Gaussian mixture model.

---

2) Update the parameters to their new values where:

\[
\mu^{new}_k = \frac{\sum_{m=1}^{M} P^{old}(k | x_m)x_m}{\sum_{m=1}^{M} P^{old}(k | x_m)}
\]

\[
\sum_{i=1}^{new} = \sum_{m=1}^{M} P^{old}(k | x_m) \|x_m - \mu^{new}_k\| (x_m - \mu^{new}_k)^T
\]

\[
P^{new}(k) = \frac{1}{M} \sum_{m=1}^{M} P^{old}(k | x_m)
\]

3) Repeat steps (1) and (2) for a pre-determined number of iterations or until suitable convergence.

---

**Expectation-Maximization Algorithm**

- Let the initial parameter values to be labeled as old values. EM then performs an iterative approximation in order to try to find parameter values that maximize the likelihood by using the following update rules:

1) Evaluate the posterior probability for every mixture component \( k \)

\[
P(k | x) = \frac{p(x | k)P(k)}{p(x)}
\]
One Approach Based on the Known Skin Color Model: Step 1

1. Use the skin filter to find regions that are likely to contain human skin. This filter basically relies on either color or texture information or both.

One Example

1.1 The RGB image is transformed to log-opponent (IRgB) values and from these values the texture amplitude, hue, and saturation are computed. The conversion from RGB to log-opponent is calculated:

\[- I = \left\lfloor \frac{L(R) + L(B) + L(G)}{3} \right\rfloor \]
\[- R_g = L(R) - L(G) \]
\[- B_y = L(B) - \left[ \frac{L(G) + L(R)}{2} \right] \]

where the L(x) operation is defined as

\[L(x) = 105 \log_{10}(x+1).\]

One Example (Cont.)

1.2 The Rg and By matrices are then filtered with a windowing median filter of with sides of length 4*SCALE. The SCALE value is calculated as being the closest integer value to (height+width)/320.

The median filtering is the rate limiting step throughout the skin detection process.

1.3 Calculate hue and saturation using the filtered Rg and By:

\[\text{hue} = \left(\tan\left(\frac{R_g}{B_y}\right)\right)\]
\[\text{saturation} = \sqrt{R_g^2 + B_y^2}\]

One Example (Cont.)

1.4 Construct the texture map as follows:

- Median filter the original I matrix with a square window of length 8*SCALE
- Subtract the filtered image from the original I matrix
- Take the absolute value of the difference and median filter the result with a square window of length 12*SCALE to obtain a new matrix Inew

1.5 Find the possible skin regions which satisfy the following conditions:

\[- \text{Inew} < 4.5\]
\[- 150 < \text{hue} < 180\]
\[- 20 < \text{saturation} < 80\]

One Approach Based on the Known Skin Color Model: Step 2

2. Detect faces by using a combination of thresholding and mathematical morphology to find “holes” in the region.

- Histogram equalization is applied to “stretch” the image. This will help to make the dark and light regions fall into more predictable intensity ranges and compensate somewhat for effects of illumination in the image.
- Thresholding is used to remove the darkest and lightest pixels.
- Use morphological operations to find the holes.
- Make conclusions.

Face Detection Technique 3: Finding Face by Motion

- If you are able to use real-time video, you can use the fact that a face is almost always moving in reality. As a result, calculating the moving area can obtain the face.
- What if there are other objects moving in the background? Can you be able to find a face?
Simple Motion Approach

- Frame differencing is a common approach for motion analysis.
- The main problem about frame differencing is that for a uniformly colored object, frame differencing will only give response at the edge of the object, but not in its interior.
- However, since we are looking for high contrast features in the face, these features stand out very well under imaging differencing.

Motion Detection Approach: Frame Differencing

To detect and analyze movement in a video sequence, the following four steps are used:

1. Frame differencing
2. Thresholding
3. Noise removal
4. Add up pixels on each line in the motion image

Robust Motion Approach

- Motion estimation is best achieved at moving object contours where computations are likely to be most relevant and reliable.
- This can be effectively achieved by convolving the intensity history of each pixel in a Gaussian smoothed image frame $I(x,y,t)$ with the second-order temporal derivative of a Gaussian function $G(t)$.

Robust Motion Approach (Cont.)

- This yields a sequence given by:
\[ Z(x, y, t) = \frac{\partial^2 G(t) \circledast I(x, y, t)}{\partial t^2} \]

with the temporal Gaussian derivative filter given by:
\[ \frac{\partial^2 G(t)}{\partial t^2} = -\frac{2s^3}{\sqrt{\pi}} (1 - 2s^2 t^2) e^{-s^2 t^2} \]

where $s = \sqrt{3/(2m^2)}$ is a temporal smoothing constant controlled by the number of image frames taken for the temporal convolution.

Robust Motion Approach (Cont.)

- Moving object boundaries produce spatial zero-crossings in $Z(x,y,t)$ at image locations in the middle frame of the history used for the temporal convolution.
- Global illumination changes and even changes in the intensity level of static objects do not result in such zero-crossings.

Face Detection Technique 4:
Finding Face by Mixtures of Techniques

- Motion analysis can be further combined with heuristics or color information to get an initial guess at the location of a face.
- Scan the detected location at as many resolutions as our computer allows us to do in real time, and use the reconstruction error from an eigenface basis to determine the presence of a face.
Heuristics

- Use as much prior knowledge about the task as possible. For face detection, we know that the head is located on top of the body, and that a human normally walks upright and so on. The following heuristics capture some of this prior knowledge:
  1. If there is a large moving object in the image, there may be a human present
  2. If the movement in the upper part of the moving object is larger than a threshold, this may be the top of the human, thus a face.

The Final Step in Face Detection

- After applying all the above techniques, we have a fairly good indication of where there may be a human face present.
- However, we still need to determine if the object we have detected is a face especially when the images contain unconstrained scenes.

Face Detection Technique 5: Finding Face in Unconstrained Scenes

- Knowledge-Based Methods
  - Encode human knowledge of what constitutes a typical face (usually the relationship between facial features)

- Feature invariant Approaches
  - Aim to find structural features of a face that exist even when the pose, viewpoint, or lighting conditions vary.

Feature Extraction

- Edge Detectors
- Wavelet and wavelet packets
- Discrete Cosine transformation
- Wavelet Gabor filters
- Several Important Statistics: Moments
- Shape/color features
- Fuzzified features
Feature Extraction: 
Face Space

- The face space is usually computed by a principal components analysis or linear discriminant analysis (Fisher's Linear Discriminant) of the face database.
- Both analyses are classical methods for multivariate analysis.
- They can form eigenfaces or fisher faces.

Some examples of eigenfaces (the number indicates the principal component number, ordered according to eigenvalues)

Feature Extraction: 
Gabor Wavelet-based Features

- Gabor wavelet filters are multi-scale and selective to specific directional changes in the image.
- They can therefore be used to obtain invariance to scale change and to investigate the effect of locally oriented image features.
- In addition, they achieve a certain amount of localized normalization for illumination.

(b) The Gabor wavelet responses at 3 spatial frequencies and 4 orientations (0, 45, 90, 135).

Gabor Wavelet-based Features (Cont.)

It can be seen that at lower frequencies, faces are smoothed to a larger extent resulting in less sensitivity to small translations in the image-plane and greater correlation between nearby images in a sequence.

However, using excessively low frequencies results in loss of relevant spatial structure as can be seen in the first row of (c).
Gabor wavelets are complex, consisting of an odd and even kernel. The Gabor-filtered image can be decomposed into magnitude and phase components. (c) shows the magnitude responses for the same image.

In order to investigate the effect of the local intensity normalization performed by Gabor wavelets, the magnitude responses at the different orientations are superimposed. Figure (a) shows normalized intensity image. Figure (b) shows the superimposed magnitude image.

Face Recognition

- Classification Techniques
  - PCA (Principal Component Analysis)
  - LDA (Linear Discriminant Analysis)
  - ICA (Independent Component Analysis)
  - SVMs (Support Vector Machines)
  - NN (Neural Network)
- Statistical Modeling
  - Bayes classifier
- Distance/Similarity Measures
  - Euclidean Distance
  - Bhattacharyya Distance
  - Manhattan Distance

Improving Neural Network: Data Standardization

- Numerical values of features may be subject to different units or different dynamic ranges
- Scaling input – Data standardization
  - input pattern be shifted so that each feature has a zero mean and the same variance.
  - Standardization should be done once before the start of network training
  - Any test pattern must be subject to the same transformation before it is to be classified by the network

Improving Neural Network: Setting of Target Values

- For \( a = 1.716 \), the output units saturate at \( \pm 1.716 \)
- Target values should be near the saturating values
  - 1 for the target category
  - \(-1\) for the other categories
  - Ex. \( f(\text{net}) = \frac{2a}{1+e^{\text{net}}} - a \) \( \text{ net } = (-1, -1, 1, -1)^t \)

Improving Neural Network: Number of Hidden Units

- Number of hidden units
  - The number of hidden units, \( n_H \), governs the expressive power of the net.
    - Too many leads to overfitting
    - Too few has less expressive power
  - number of weights = \( n/10 \). For a 2-\( n_H \)-1 network \( 3(n_H + 1) \approx \frac{n}{10} \Rightarrow n_H \approx \frac{n}{30} - 1, \quad n : \text{number of patterns} \)
  - Start with a large number of hidden units and decay, prune, or eliminate weights

Cf. \( d-(2d+1)-c \) network
Improving Neural Network: Stopped Training

• Excessive training can lead to poor generalization since the net is tuned to the specific training data
  – In training 2-layer networks, the complexity of decision boundary is not changed during training (always a hyperplane) -- without fear of excessive training
  – Nonlinearity of neural units --> warping of decision boundary
• Stop training when the error on a separate validation set reaches a minimum

Cf. Fig. 6.6