Robust Face Detection with FDM using Adaboost

Daijin Kim
Intelligent Multimedia Lab.
POSTECH
April 30, 2005

Outline

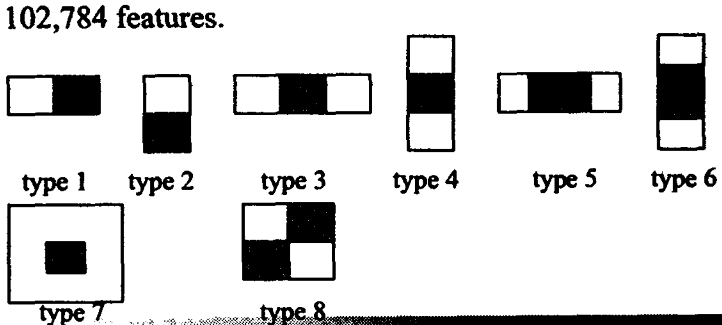
- Introduction
- Rectangular features and Integral image
- Adaboost algorithm
- Cascade algorithm
- FDM (Face Detection Map)
- Experiment results
- Conclusion

Introduction

- Face detection
  - A quick feature extraction using "integral image"
  - A robust learning algorithm using Adaboost
  - A rapid classification using a cascade of classifiers.
  - A robust face detection using FDM (Face Detection Map)

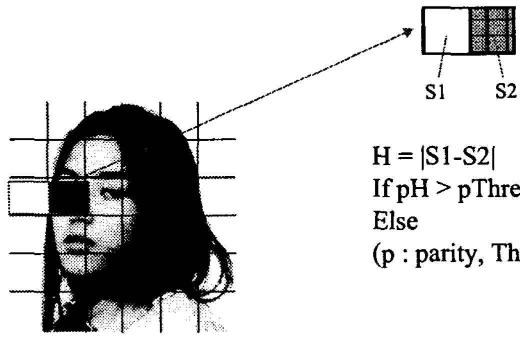
Features (I)

- Definition of simple features for face detection
  - 8 rectangular features types
  - Using a 30x35 pixel base detection window, with all the possible combination of horizontal and vertical location and scale of these feature types the full set of features has 102,784 features.



Features (II)

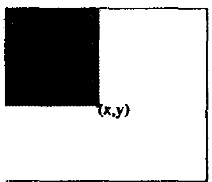
- Example for features
  - S1, S2 is average gray value of rectangular feature



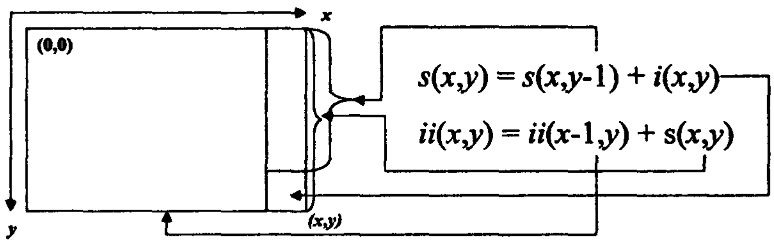
H = |S1-S2|
If pH > pThresh then Face
Else then NonFace
(p : parity, Thresh : threshold)

Integral image (I)

- Integral image



Def: The *integral image* at location (x,y), is the sum of the pixel values above and to the left of (x,y), inclusive. Using the following two recurrences, where $i(x,y)$ is the pixel value of original image at the given location and $s(x,y)$ is the cumulative column sum, we can calculate the integral image representation of the image in a single pass.

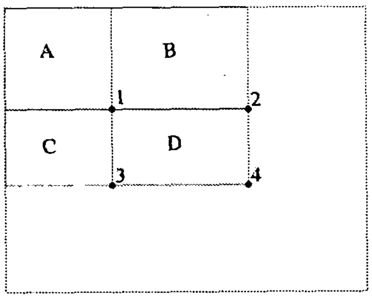


$$s(x,y) = s(x,y-1) + i(x,y)$$

$$ii(x,y) = ii(x-1,y) + s(x,y)$$

Integral image (II)

- Rapid evaluation of rectangular features



Using the integral image representation one can compute the value of any rectangular sum in constant time.

For example the integral sum inside rectangle D we can compute as:

$$ii(4) + ii(1) - ii(2) - ii(3)$$

Challenge

- Challenges for learning a classification function
  - Given a feature set and labeled training set of images one can apply number of machine learning techniques.
  - Recall however, that there is 102,784 features associated with each image sub-window, hence the computation of all features is computationally prohibitive.
  - Hypothesis: A combination of only a small number of these features can yield an effective classifier.
  - Challenge: Find these discriminant features.

Adaboost Training (I)

- Strong classifier formed from weak classifiers:

$$H_M(x) = \sum_{m=1}^{M} h_m(x)$$

- At each stage, new weak classifier chosen to minimize bound on classification error

$$\varepsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$$

Adaboost Training (II)

- A variant of Adaboost for aggressive feature selection
  - Given example images $(x_1,y_1), \ldots, (x_n,y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
  - Initialize weights $w_{1,i} = 1/(2m), 1/(2l)$ for training example i, where m and l are the number of negatives and positives respectively.

  For $t = 1 \ldots T$

  1) Normalize weights so that w_t is a distribution
  2) For each feature j train a classifier h_j and evaluate its error ε_j with respect to w_t.
  3) Chose the classifier h_j with lowest error.
  4) Update weights according to:

  $$w_{t+1,i} = w_{t,i}\beta_t^{1-\varepsilon_i}$$

  where $e_i = 0$ is x_i is classified correctly, 1 otherwise, and

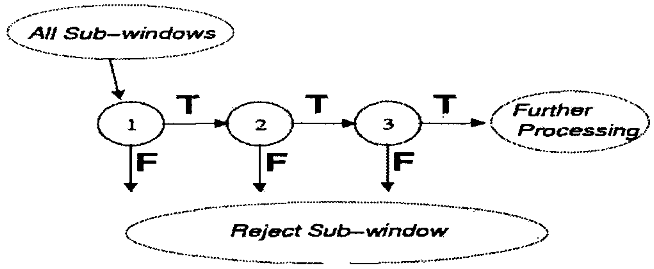
  $$\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

  - The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2}\sum_{t=1}^{T} \alpha_t, \\ 0 & otherwise \end{cases} \quad \text{where} \quad \alpha_t = \log(\frac{1}{\beta_t})$$

Cascade (I)

- Speed-up through the Attentional Cascade
  - Simple, boosted classifiers can reject many of negative sub-windows while detecting all positive instances.
  - Series of such simple classifiers can achieve good detection performance while eliminating the need for further processing of negative sub-windows.



Cascade (II)

- Algorithm for training a cascade of classifiers

• User selects values for f, the maximum acceptable false positive rate per layer and d, the minimum acceptable detection rate per layer.
• User selects target overall false positive rate F_{target}.
• P = set of positive examples
• N = set of negative examples
• $F_0 = 1.0$; $D_0 = 1.0$; $i = 0$

While $F_i > F_{target}$
  i++
  $n_i = 0$; $F_i = F_{i-1}$
  while $F_i > f \times F_{i-1}$
    o n_i++
    o Use P and N to train a classifier with n_i features using AdaBoost
    o Evaluate current cascaded classifier on validation set to determine F_i and D_i
    o Decrease threshold for the ith classifier until the current cascaded classifier has a detection rate of at least $d \times D_{i-1}$ (this also affects F_i)
  $N = \emptyset$
  If $F_i > F_{target}$ then evaluate the current cascaded detector on the set of non-face images and put any false detections into the set N.

FDM (I)

- An early approach
  - □ Only Concentrating on High Performance of Detecting Algorithm
    - Possible generation of False Acceptance
    - For detecting single face from multiple candidates, select the nearest candidate rather than use relationship of each candidate
- FDM (Face Detection Map) approach
  - □ Reducing FAR substantially
  - □ No need to compute the nearest candidate

FDM (II)

- Face detection using adaboost

$$\sum_{t=1}^{M_t} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{M_t} \alpha_t$$

: Face(M_t rectangular filter num, h_t weak classifier, α_t weight)

- Face detection using adaboost with cascade

$$\sum_{n=1}^{C_n} \sum_{t=1}^{M_t} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{n=1}^{C_n} \sum_{t=1}^{M_t} \alpha_t$$

: Face(Cn cascade num)

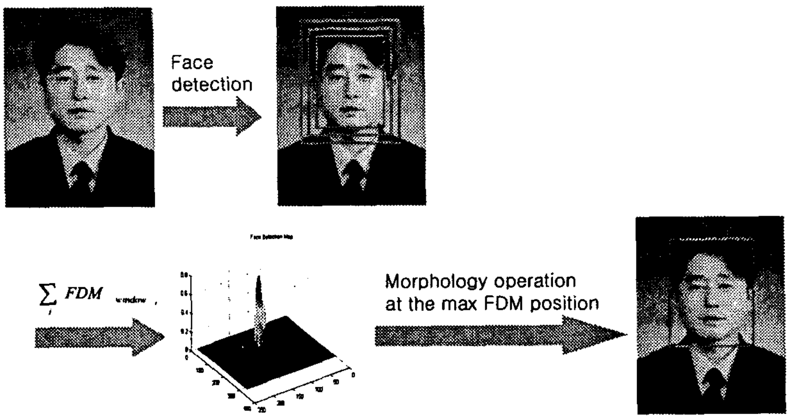
- Face detection using FDM

$$FDM_{window_i} = \frac{\sum_{n=1}^{C_n} \sum_{t=1}^{M_t} \alpha_t h_t(x)}{\sum_{n=1}^{C_n} \sum_{t=1}^{M_t} \alpha_t} \geq \frac{1}{2}$$

FDM (III)

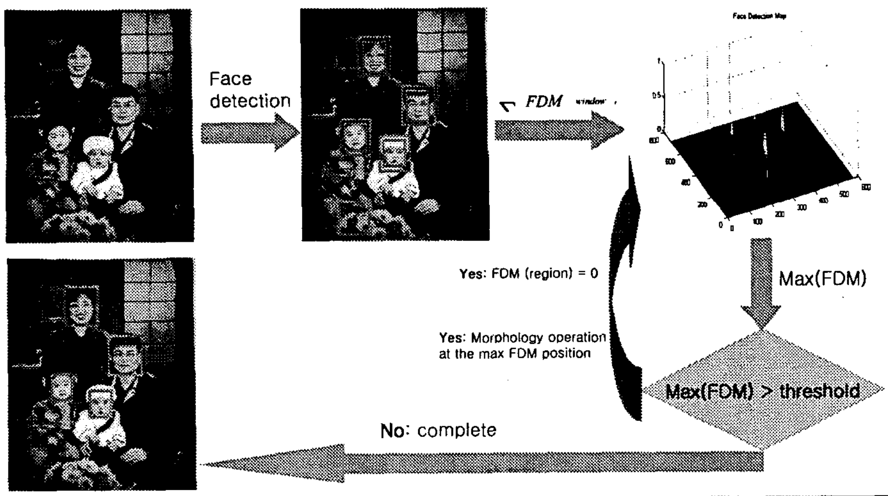
- Face detection using FDM

$$FDM = \sum_i FDM_{window_i}$$



FDM (IV)

- Face detection using FDM

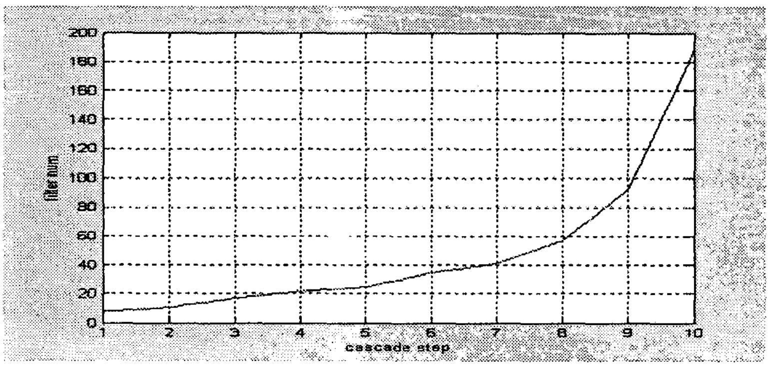


Experiment Results (I)

- Dataset for training
  - 5,012 positive training examples were hand picked aligned, normalized, and scaled to a base resolution of 30x35
  - 20,000 negative examples were selected by randomly picking sub-windows from 2000 images which did not contain faces.

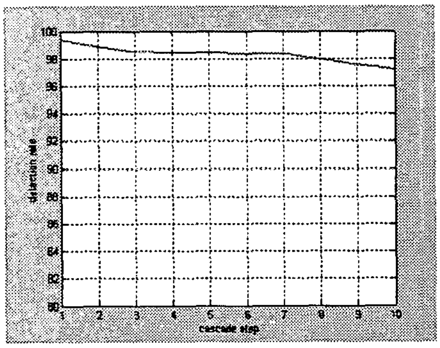
Experiment Results (II)

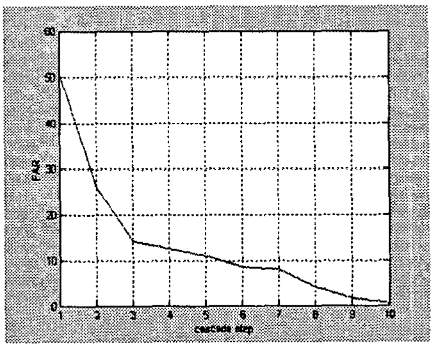
- Structure of the detector cascade
  - The final detector had 10 layers



Experiment Results (III)

- Detection rate of 10^{th} layer: 97.24%
- FAR of 10^{th} layer: 0.6511%
- The processing time of a 320 by 240 pixel image on a conventional personal computer about 48msec.





Experiment Results (IV)

Performance	FAR
Adaboost	0.6511%
FDM	0.0529%

- Experiments on a Real-World Test Set

Adaboost

FDM

Conclusion

- Using the integral image representation and simple rectangular features, we eliminate the need of expensive calculation of multi-scale image pyramid.
- Simple modification to Adaboost gives a general technique for efficient feature selection.
- Cascade can reject most of the negative examples at early stages of processing thereby significantly reducing computation time.
- FDM can reduce FAR substantially.
- FDM does not need to compute the nearest candidate.

Face Recognition using Adaboost and LFA

Daijin Kim
Intelligent Multimedia Lab.
POSTECH
April 30, 2005

Outline

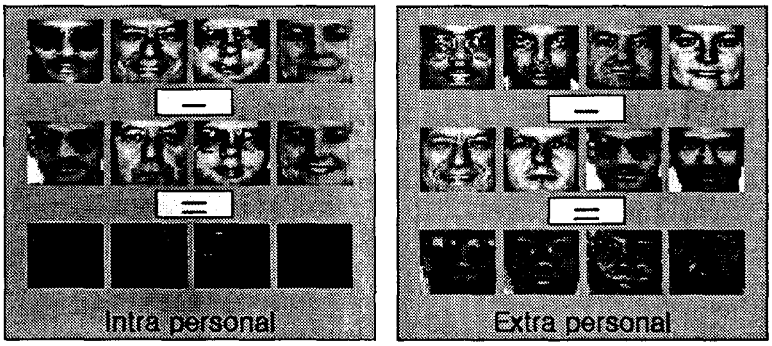
- Introduction
- 2-class problem
- LFA
- Adaboost
- Experiment results
- Conclusion

Introduction

- Face recognition
  - A robust feature extraction using LFA
    - LFA (Local Feature Analysis) is known as a local method for face recognition, because it constructs kernels which detect local structures of a face.
    - However LFA addressed only image representation, and has problems for recognition.
  - Adaboost learning with Intra/Extra
    - After extracting local structures using LFA, we select a subset of them, which are efficient for recognition.

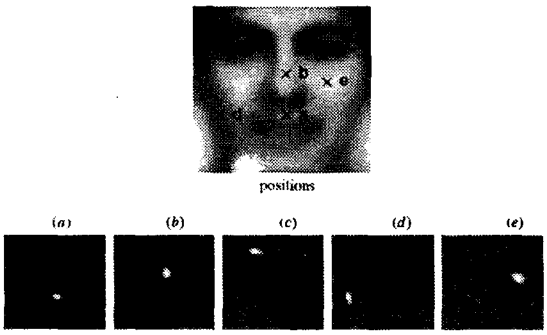
2-class problem

- Face recognition is a multi-class problem, but, we propose to train adaboost based on the intra-personal and extra-personal variation in the LFA feature space.
- The thinking of the face recognition method of Moghaddam and Pentland is to convert the multi-class problem to two-class problem.



LFA (I)

- LFA locally uses the eigenface method for a few single parts of the face(e.g. eyes, nose, mouth) and additionally determines their geometric proportions to each other.
- LFA-based face recognition is relatively insensitive with respect to changes in expression and illumination.



LFA (II)

- The kernels of LFA are derived by enforcing topology into eigenvectors of PCA.
- In a matrix form, a set of kernels, the output matrix and correlation of outputs are written as

$$K = W \Lambda W^T$$
$$O = K^T I$$
$$P = WW^T$$

- λ and W denotes eigenvalue and eigenvector of covariance matrix of face images (I), Λ is

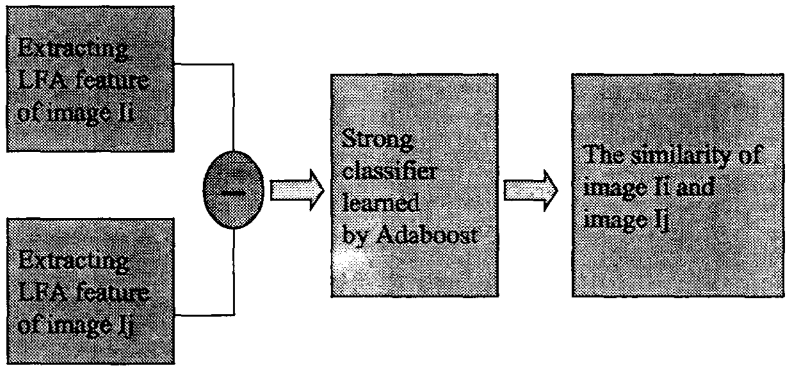
$$\Lambda = diag\left(\frac{1}{\sqrt{\lambda_i}}\right)$$

LFA (III)

- An early LFA approach
  - LFA chooses a set of kernels whose outputs produced the biggest reconstruction error in the sense of minimum reconstruction error.
  - Although mean reconstruction error is a useful criterion for representing data, there is no reason to assume that it must be useful for discriminating between data in different classes.
- Proposed approach
  - After extracting local structures using LFA, we select kernels suitable for recognition by Adaboost.

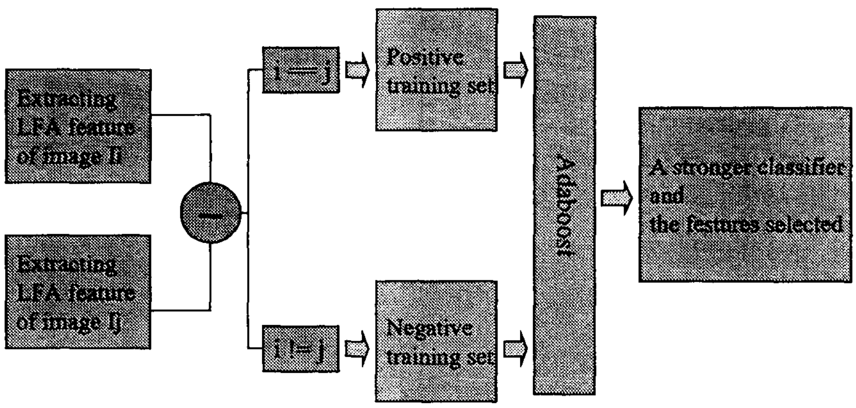
Adaboost (I)

- The flowchart of the proposed face recognition method.



Adaboost (II)

- Framework of the proposed training process



Experiment Results

- Dataset for training
  - 2500 positive training examples (intra personal set) and 2500 negative examples (extra personal set) were hand picked aligned, normalized, and scaled to a base resolution of 40x40
- Recognition rate of 12th layer: 89.2%
- FAR of 12th layer: 0.0315%

Conclusion

- LFA chooses a set of kernels whose outputs produced the biggest reconstruction error in the sense of minimum reconstruction error.
- LFA addressed only image representation, and has problems for recognition.
- After extracting local structures using LFA, we select kernels suitable for recognition by Adaboost.
- Adaboost is successfully applied to face recognition by introducing the intra-personal and extra-personal difference space in the Local feature space.