파일(F) 보기(V) 컨트롤(C) 도움말(H)

Evaluation of Face Detection and Recognition Algorithms on Avatar Face Datasets

Cybersecurity Lab
University of Louisville

Presented
by

Gyuchoon Cho

[◄] [►]

[Face Detection]

- Face detection is a computer technology which determines the location and size of human faces in arbitary images. *(Wikipedia)*
- Nowadays, face detection technology is ubiquitous in the world.
  - Surveillance cameras - with suspect detection.
  - Fuji film's face-focusing, Sony's smile shutter, Apple iPhoto.
  - System log on with face recognition

[Examples of Face Detection]

■ Recording

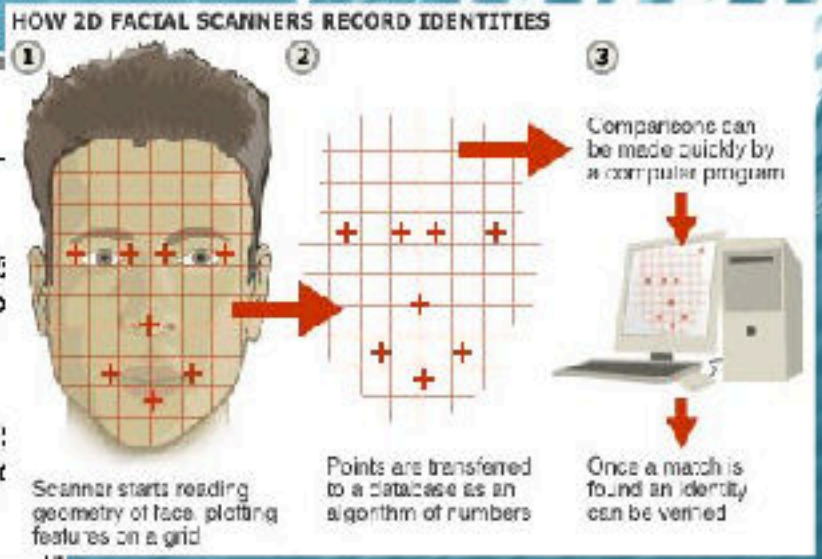
Detecting....

Matching with Database

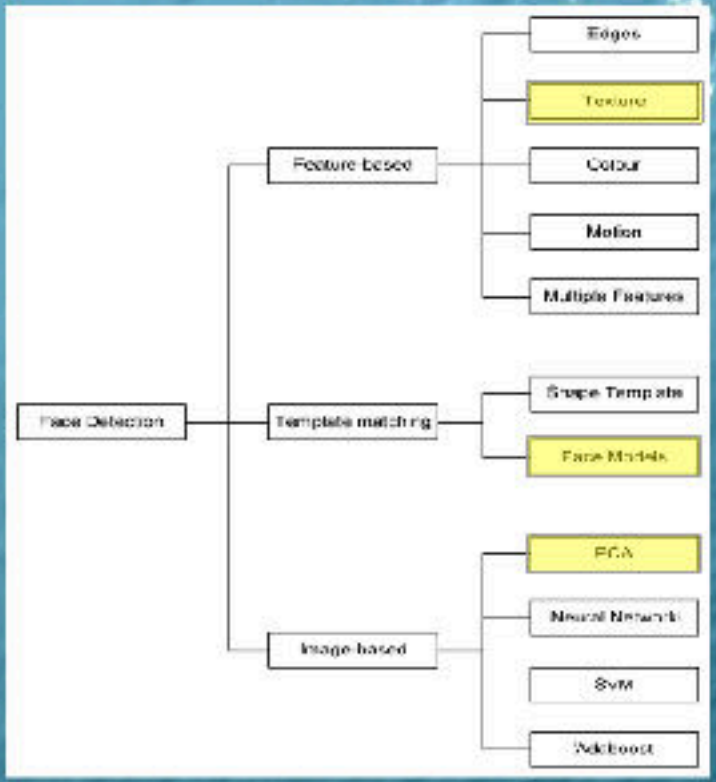
Name: Alireza,
Date: 25 May 2007 19
Place: Main corrido

Name: Unknown
Date: 25 May 2007 1
Place: Main corrido

Report



Face Detection Techniques



Face Detection tasks

face detection can be regarded as a specific case of

- Object - Class detection
  The task is to find the locations and size of all objects in an image that belong to a given class. Examples include upper torsos, pedestrians, and cars.
- Face Localization
  The task is to find the locations and size of a known number of faces.

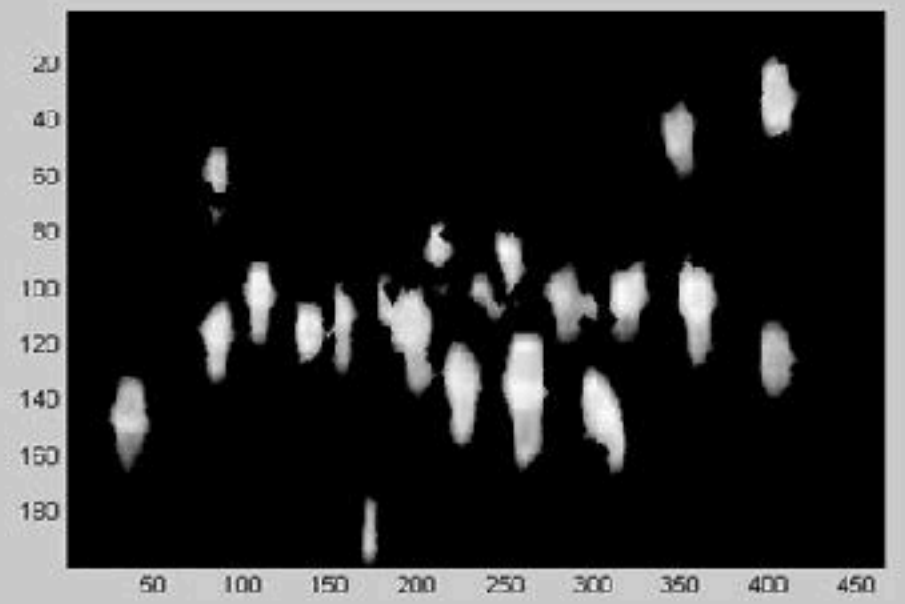
Face Localization

Probable Face Pixels

Lighter pixels mean higher probability of being a face pixel.

Next

[Face Localization]

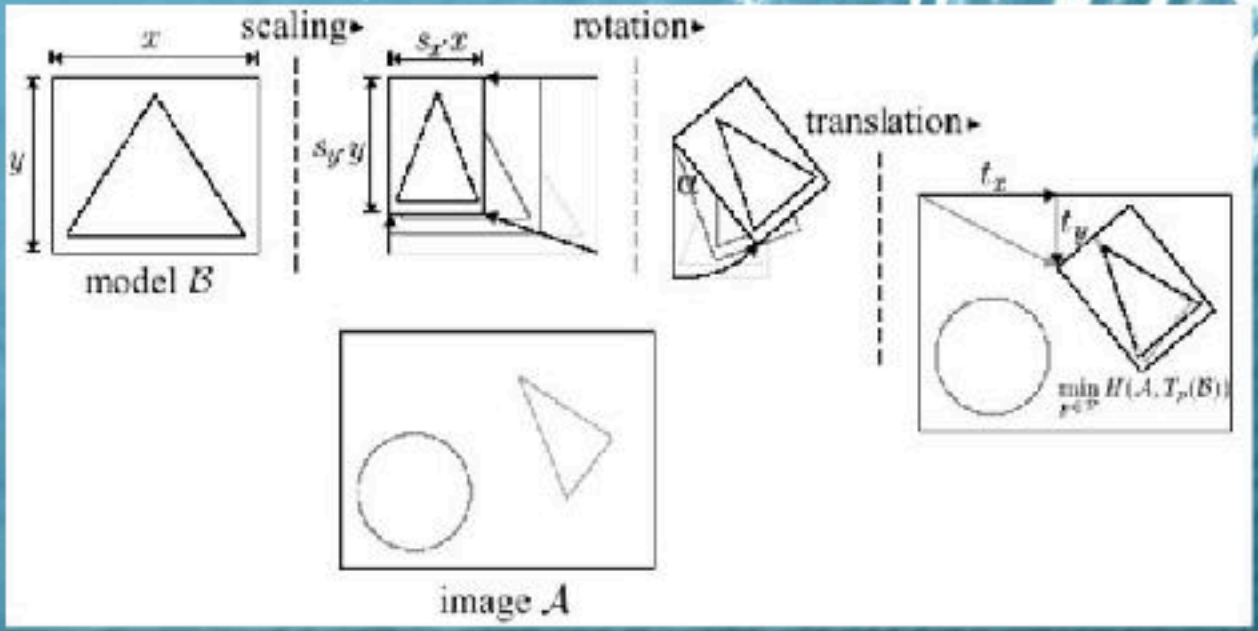


Next

Color Segmented Mask

Mask produced from thresholding the filtered probability image.

[Objecct Class Detection]



- Model fitting by scaling, translation and rotation.
  ex) Hausdorff Distance is a specific case of Object Class Detection

[Software Developement Kits]

OpenCV

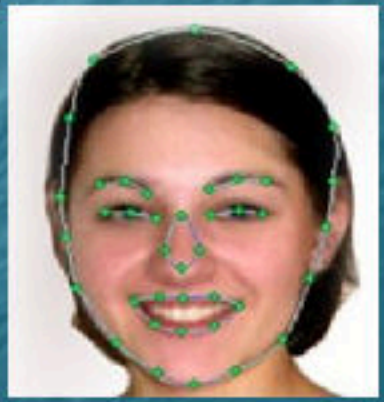
OpenCV is a computer vision library originally developed by Intel.
The library is cross-platform, and runs on Windows, Mac OS X, Linux, VCRT (Real-Time OS on Smart camera) and other embedded devices. It focuses mainly on real-time image processing

VeriLook

VeriLook SDK is based on the VeriLook PC-based face recognition technology and is intended for biometric systems developers and integrators.
It allows rapid development of biometric applications using functions from the VeriLook library, which ensure high reliability of the face identification.
VeriLook facial recognition software is now available for Mac OS X

Luxand Face SDK

-Detection of 40 facial feature points such as eyes, mouth, nose, and face.
-Detection time: 0.9 seconds
-up to 5,000 faces per second

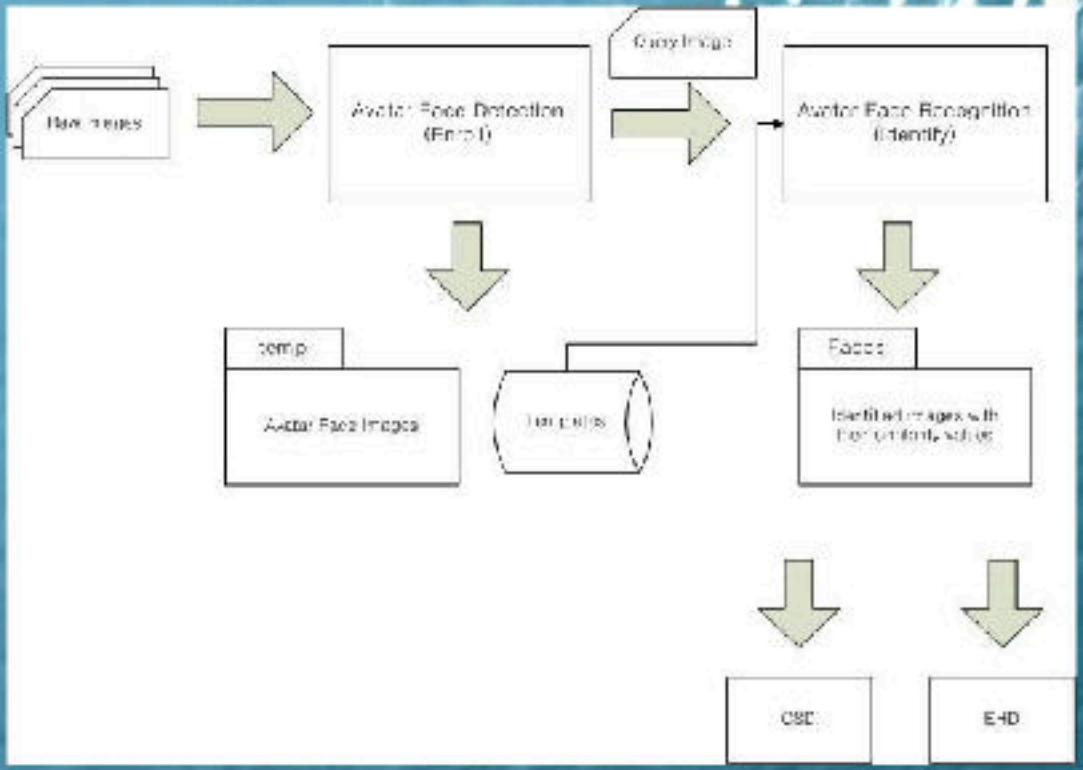


System Environment

- MATLAB R2009a
- Codegear RAD studio. (Delphi 2009)
  - Rapid Application Development Programming Tool
- Verilook 4.0
- SQL lite

SQLite

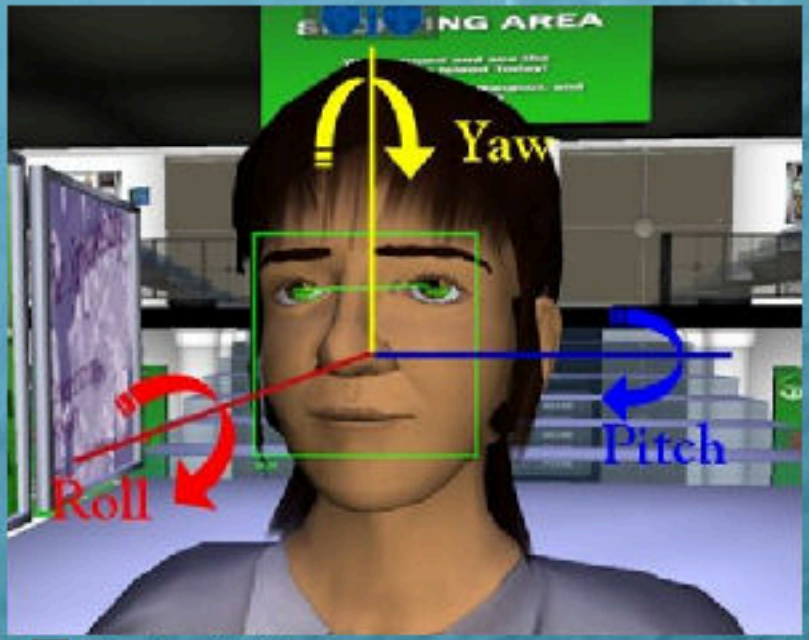
[System Work Flow]



[Algorithm]

- Advanced face localization
  - The algorithm implements advanced face localization, enrollment and matching using robust digital image processing algorithms.
- Fast identification speed - 100,000 faces per second
  - Tested on a PC with Intel Core 2 processor running at 2.66 GHz and 640 × 480 pixels of image

[Roll, Pitch and Yaw #1]



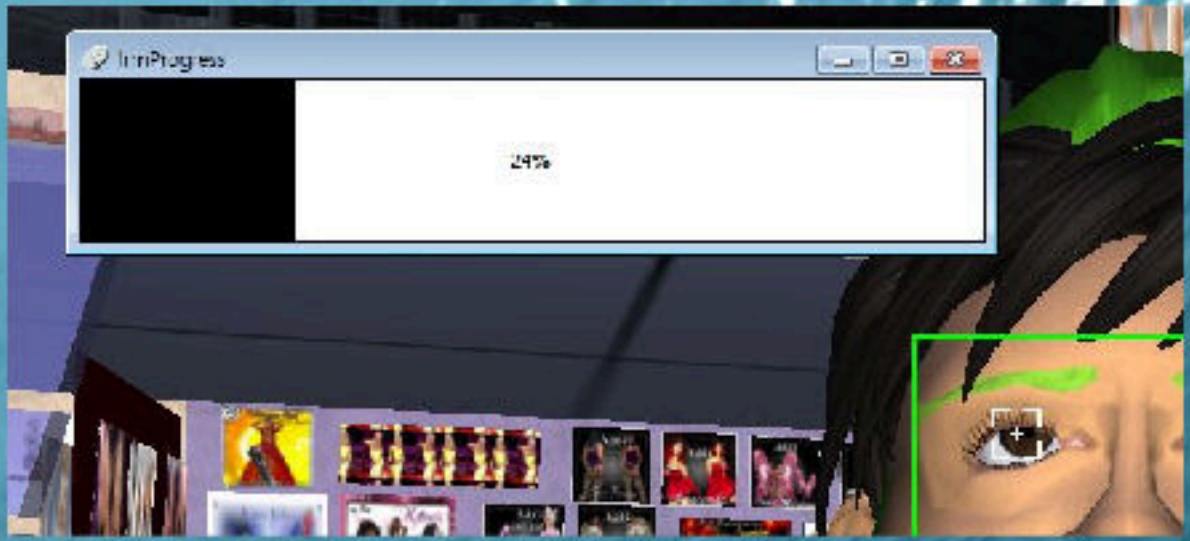
- Head roll (tilt) - ±180 degrees (configurable); ±15 degrees recommended
- Head pitch (nod) - ±15 degress from frontal postion;
- Head yaw (bobble) - ±15 degress from frontal positioin;

[Roll, Pitch and Yaw #2]

	Maximized Template Size	Medium Template Size	Minimum Template Size
Detection time for all faces in a frame (±15° head roll tolerance)			10 milliseconds
Detection time for all faces in a frame (±180° head roll tolerance)			135 milliseconds
Single face template extraction time (1) (milliseconds)	111	62	31
Matching speed (2) (face records per second)	24,000	44,000	200,000
Template size in database (3) (bytes)	20,440	11,368	2,256

파일(F) 보기(V) 컨트롤(C) 도움말(H)

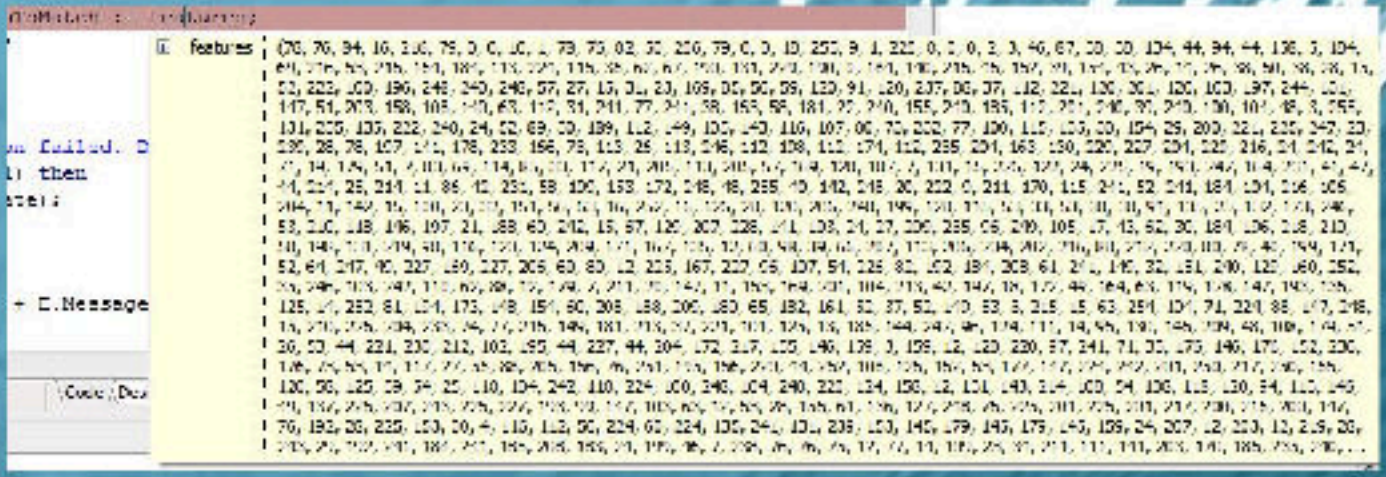
[Enrollment]



701	SL-100g.tif	YES	2.284	60.58301	50	50
702	Total Detected	559	Percentag	79.85714		
703	Average of Confidence	56.18021				
704	Total Process Time	1259.339				
705						

파일(F) 보기(V) 컨트롤(C) 도움말(H)

[Database #1]

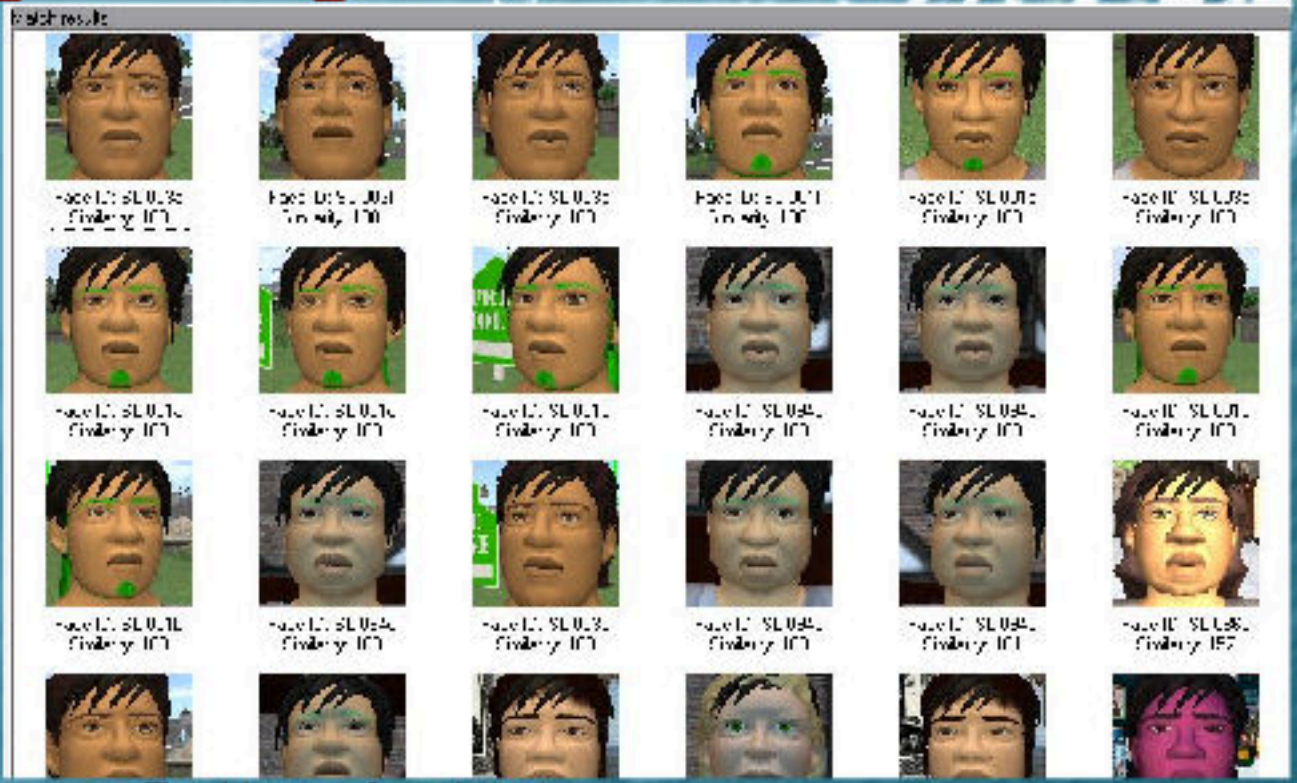


[database #2]

```
if (not database.TableExists('Templates')) then
begin
  createTable := 'CREATE TABLE Templates (Id INTEGER PRIMARY KEY, '+
                                         'TemplateId TEXT NOT NULL, '+
                                         'Template BLOB NOT NULL, '+
                                         'Thumbnail BLOB)';
  database.ExecSQL(createTable);
end;
if (not database.TableExists('Images')) then
begin
  createTable := 'CREATE TABLE Images (Id INTEGER PRIMARY KEY, '+
                                      'TemplateId INTEGER NOT NULL, '+
                                      'BiometricType INTEGER NOT NULL,'+
                                      'RecordIndex INTEGER NOT NULL, '+
                                      'GenIndex INTEGER NOT NULL, '+
                                      'FrameIndex INTEGER NOT NULL, '+
                                      'Image BLOB NOT NULL)';
  database.ExecSQL(createTable);
end;
```

파일(F) 보기(V) 컨트롤(C) 도움말(H)

Result #1



- Total Image found: 331

파일(F) 보기(V) 컨트롤(C) 도움말(H)

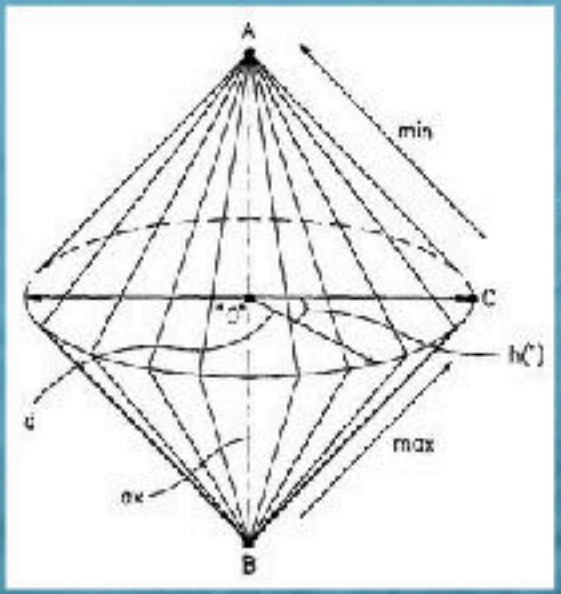
[Result #2]

- Total Image found: 317

Color Structure Descriptor

- Main functionality of this descripter is **image to image** matching
  - it expresses local color structure in an image by use of a structuring element that is comprised of several images sample.
- The CSD is a generalization of the color histogram that captures some spatial characteristics of the color distribution in an image.
  - It is defined in the HMMD color space using non-uniform color quantization to between 32-256 colors.

[HMMD]



max value=maximum value from input r, g, b values;
min value=minimum value from input r, g, b values;
diff=max value−min value;
sum=(max+min)/2; and

hue=(g−b)/(max−min)*60,if (r=max∩(g−b)≧0)
hue=(g−b)/(max−min)*60+360, if (r=max∩(g−b)<0)
hue=(2.0+(b−r))/(max−min)*60, if (g=max)
hue=(4.0+(r−g))/(max−min)*60, if (g=max)
hue= 0, if max=min.

파일(F) 보기(V) 컨트롤(C) 도움말(H)

[CSD result]

CSD result using 64 bins

Edge Histogram Descriptor

- The image array is divided into 4x4 subimages.
- The edges in each image-block is categorized into one of the following six types:
  vertical, horizontal, 45± diagonal, 135± diagonal, nondirectional edge and no-edge.
- A 5-bin edge histogram of each subimage can be obtained.
- Each bin value is normalized by the total number of image-blocks in the subimage.
- The normalized bin values are nonlinearly quantized.

[EHD #1]

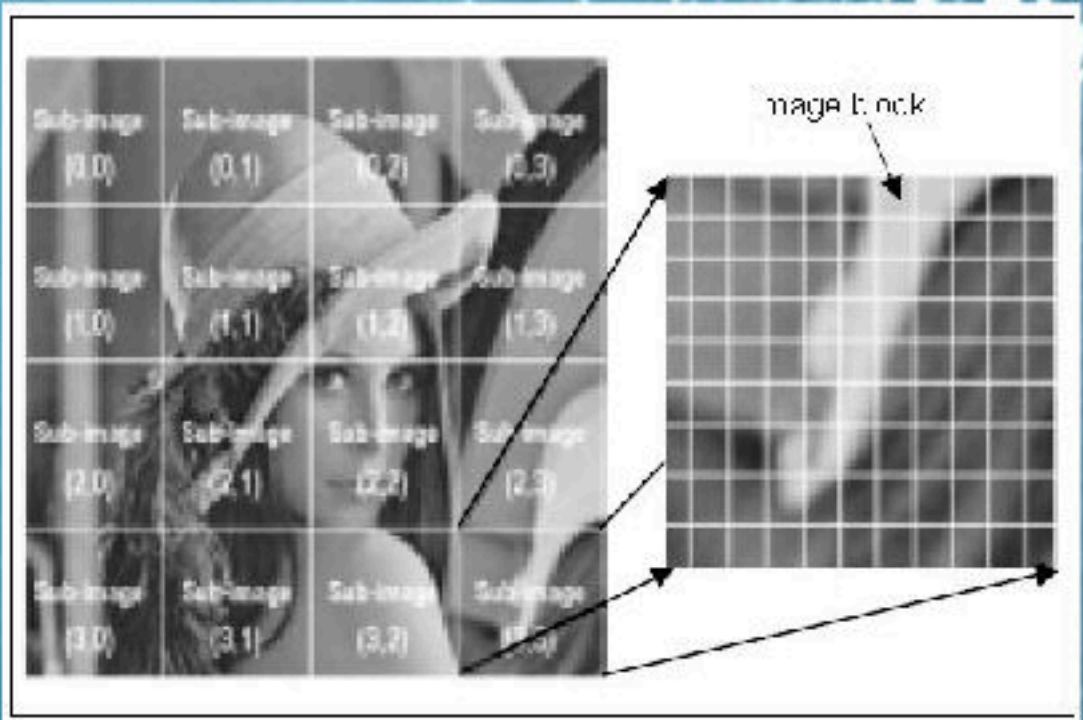


Fig. 1. Definition of sub-image and image-block.

[EHD #2]

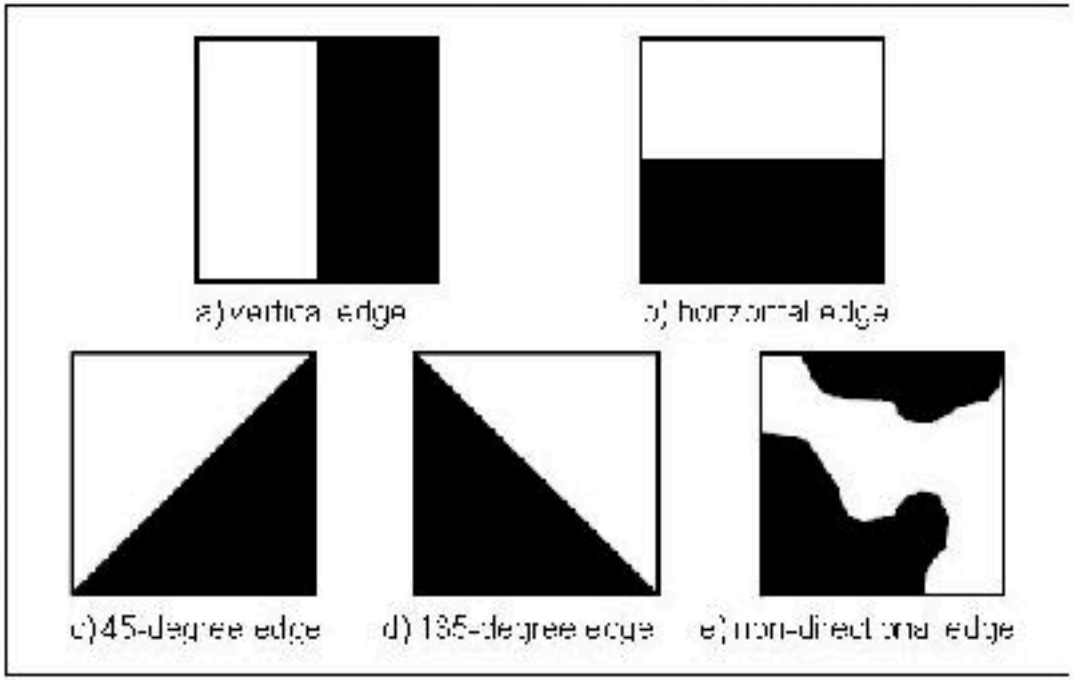


Fig. 2. Five types of edges.

[EHD #3]

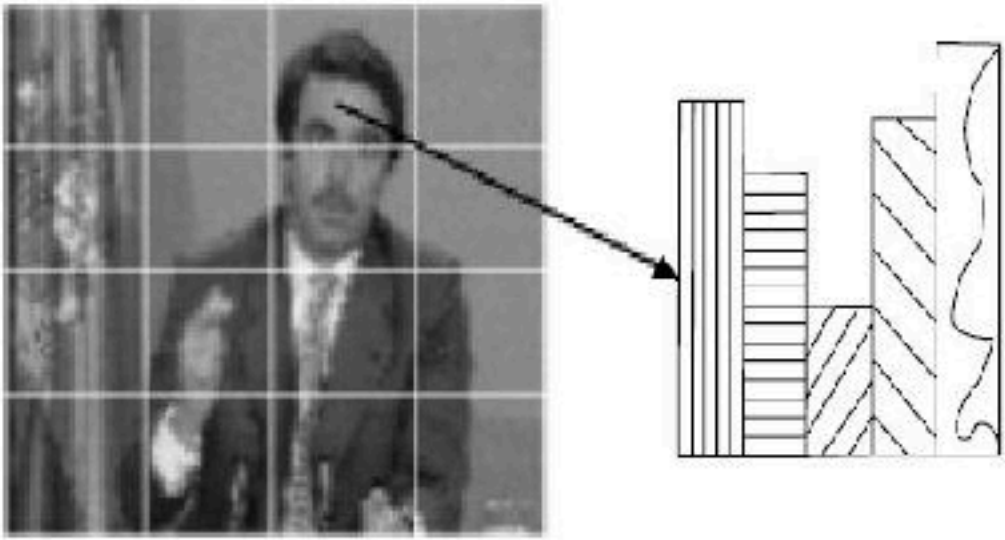


Fig. 3. Five types of edge bins for each sub-image.

[EHD #4]

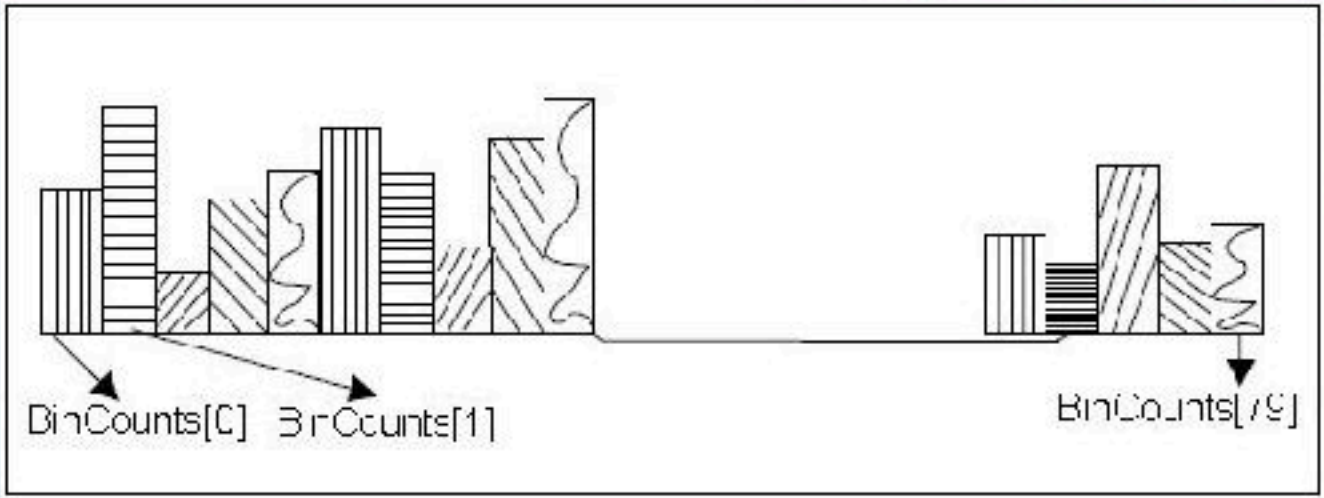


Fig. 4. 1-D array of 80 bins of EHD.

[EHD #5]

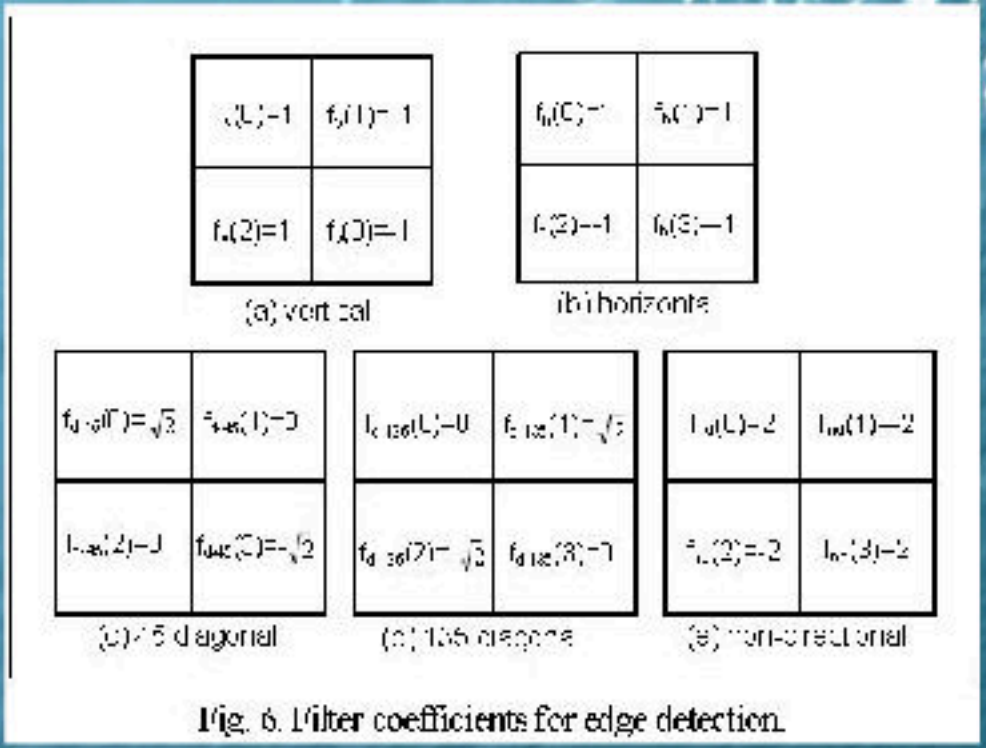


Fig. 6. Filter coefficients for edge detection.

파일(F) 보기(V) 컨트롤(C) 도움말(H)

[EHD result]

- Result: EHD using 80 bins

[Thank YOU SO MUCH]