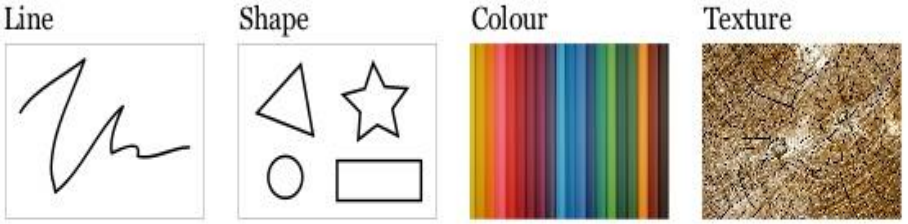
# Global Features

An Example:
**Content-Based Image Retrieval Using Global Features**

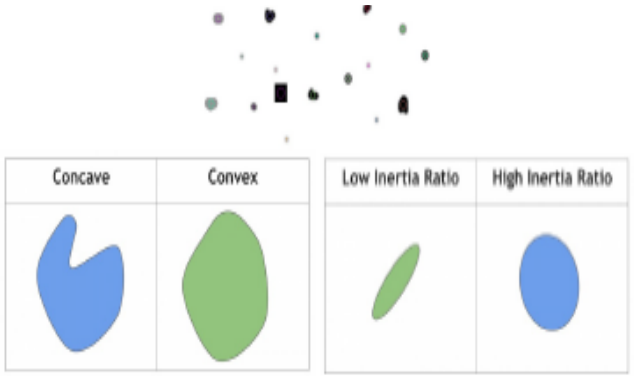
## Global Features

General features that can be extracted from the whole image. Examples: color, texture and shape.

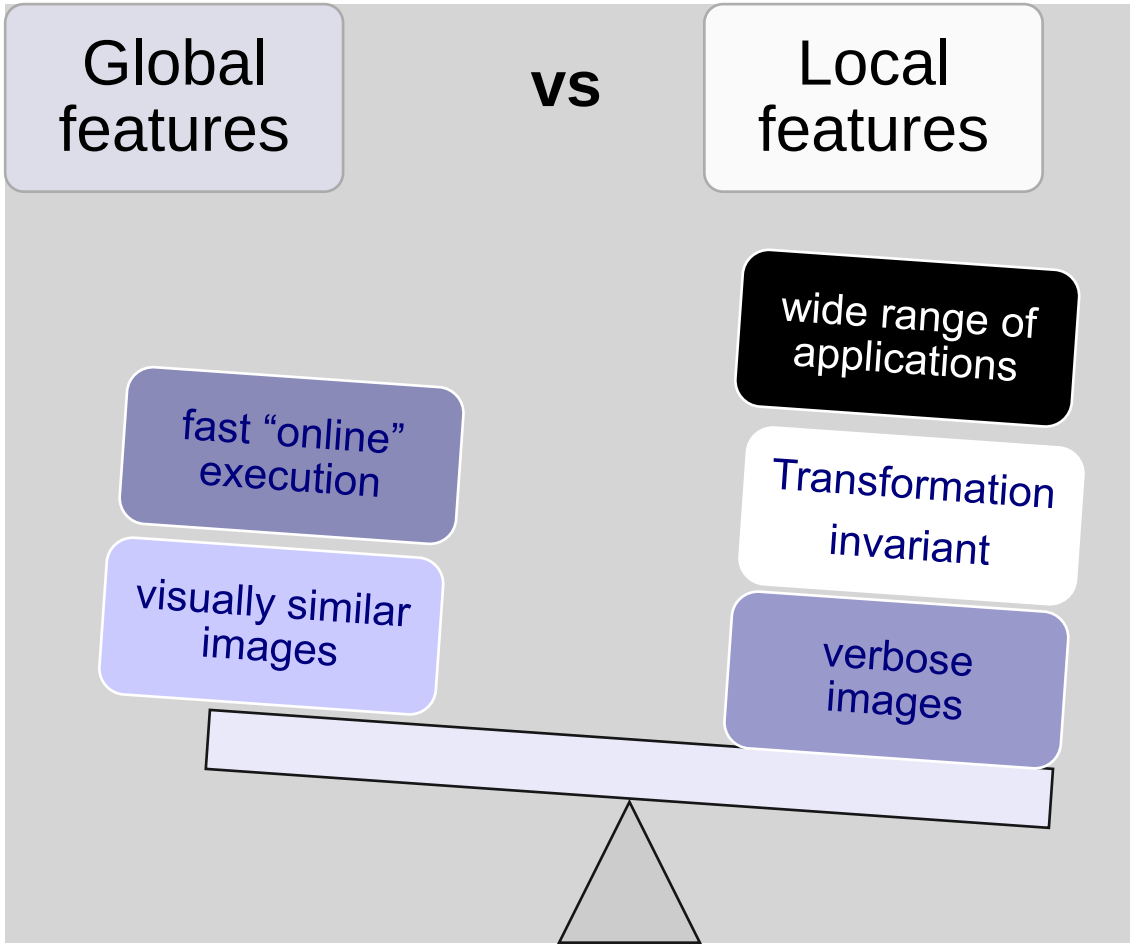



## Local Features

The features extracted from localized regions within images. For examples: Blobs, edges, and corners



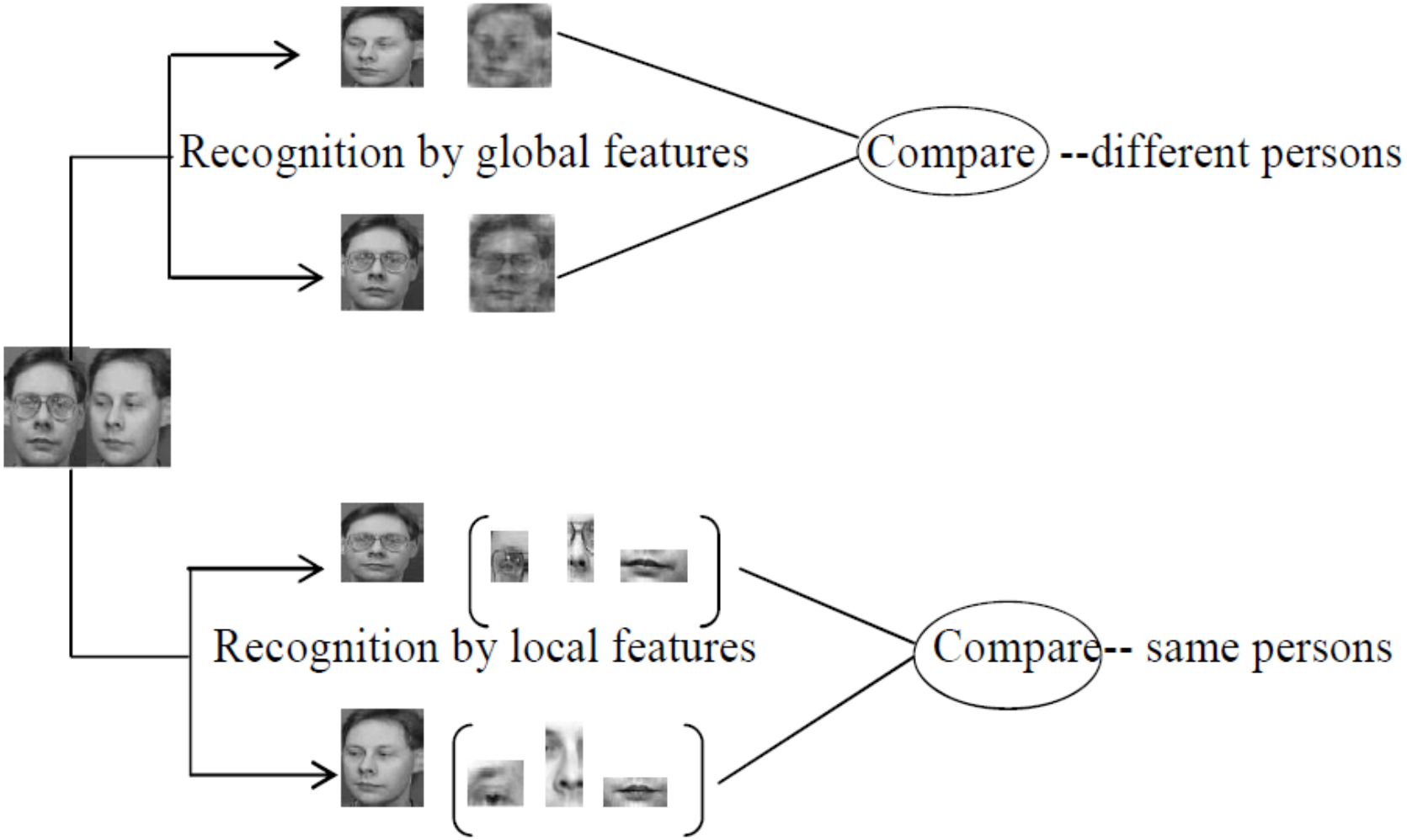
# Global vs local features for CBIR tasks




**What is considered “similar”?**

- **What does the user rate as effective image retrieval?**
- **Local or global features? Depending on the application**

# Different roles of global and local features in face perception



# Content-Based Image Retrieval (CBIR)

## What is CBIR?

Retrieving desired images from a large collection on the basis of features (such as color, texture and shape) that can be automatically extracted from the images themselves.

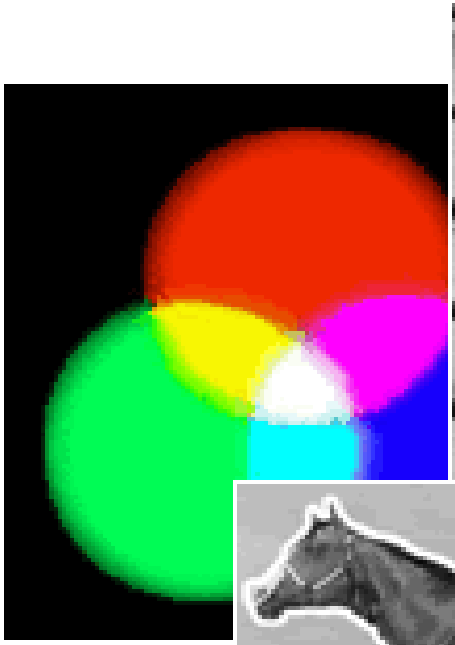
## Why we need CBIR ?

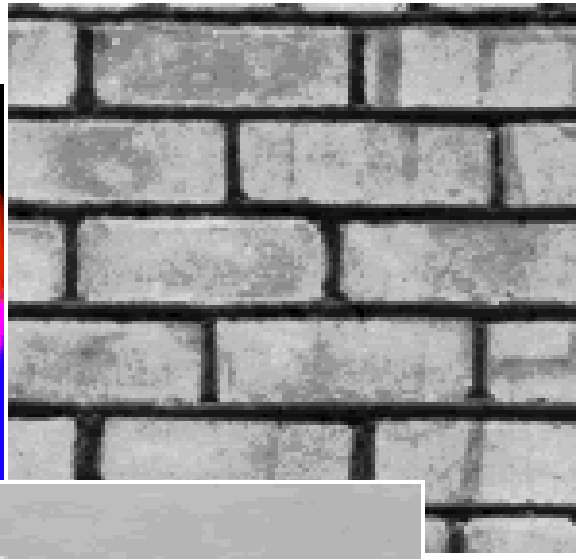
- The limits of the keywords approach.
- Large amount of pictures.
- Human perception.

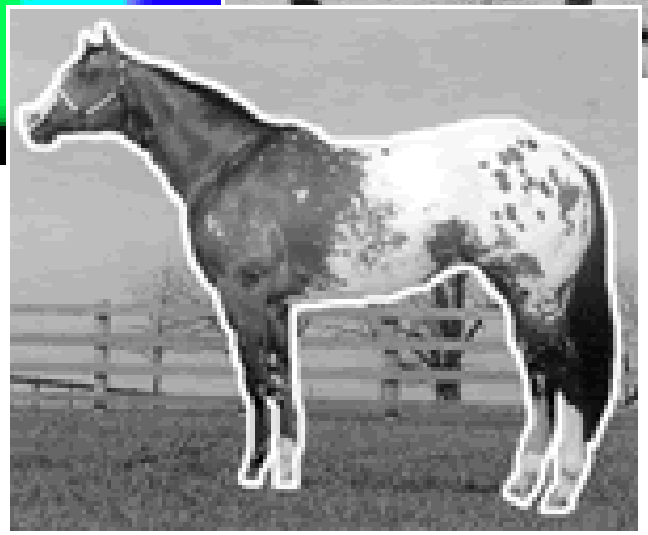
(Every person may have a different description to the same picture )

# General features of (CBIR)

- ❑ Color
- ❑ Texture
- ❑ Shape







- Other primitive features include

- ❑ Spatial location
- ❑ Pixel intensity

# Applications of CBIR

- Several application areas in which to apply content-based image retrieval have been proposed over the years like:
  - Medical applications
  - Geographic information system
  - Criminal database
  - Picture archiving and communication systems
  - Fabric and fashion design
  - Face recognition.
  - Home Entertainment

# CBIR Systems

- There are many commercial and research CBIR systems that are available for:
  - Accessing huge image databases
  - Creating image databases
  - Applying image processing operations on images

- They Support one or more of the following options:
  - Search by example
  - Search by sketch
  - Random browsing
  - Search using keywords or text.

| | EVRW | VISE | QBIC | VSEK | MARS |
|---|---|---|---|---|---|
| **Interface** | Excellent | Excellent | Excellent | Excellent | Good |
| **Initial Query** | Choose one, randomize | Choose one, randomize | Choose one, randomize | Choose one, sketch one, randomize | Choose one, randomize |
| **Feature Extractions** | Color,Shape, Texture, Structure | Histogram, Composition , Structure | Color, Texture, Sketch | Histogram, Composition , Texture | Color, Texture, Shape |
| **Relevance Feedback (RF)** | No | No | No | YES (3 levels) | YES (5-level weight) |
| **Improvement from RF** | N\A | N\A | N\A | A little | Not clear |
| **Performance Without RF** | Good | Good | Good | Good | Ok |

# Problem Description

CBIR has advantages and disadvantages with regard to the nature of image analysis and retrieval.

**Advantages of CBIR include:**

1) Easy to extract features from images
2) Able to change extracted features to other forms such as histograms
3) Easy to build an automatic process

**Main problems of CBIR approach include:**

1) Hard to determine effectiveness.

2) **Difficult to choose features for extraction and also which extraction algorithms to use.**

3) Hard to get the semantic meaning of image from low level features (Semantic Gap).

4) Difficult to process a specific region in the image.

5) Limited markets of profit for CBIR

6) No standard dataset for testing and evaluating CBIR Systems

- In content-based image retrieval, **image data representation** and the **similarity measurement** are two important tasks. In addition , the use of simple features like color, shape or texture is not sufficient

- Natural images contain both **color and texture**, so for better performance they should be considered together in the retrieval process

- Using **more than one representation** of the images in a collection can improve the results presented to a user without changing the underlying feature extraction or search technologies.

The general structur of content-based image retrieval system

Retrieved Results

Sorting Distances

Query Image

Features Extraction (GCH/ Wavelets/ Color layout)

Apply Metric

The online query phase

The database creation phase

Features Extraction (GCH/ Wavelets/ Color layout)

The database images consists of 8 categories

Buses
Horses

Roses
Buildings

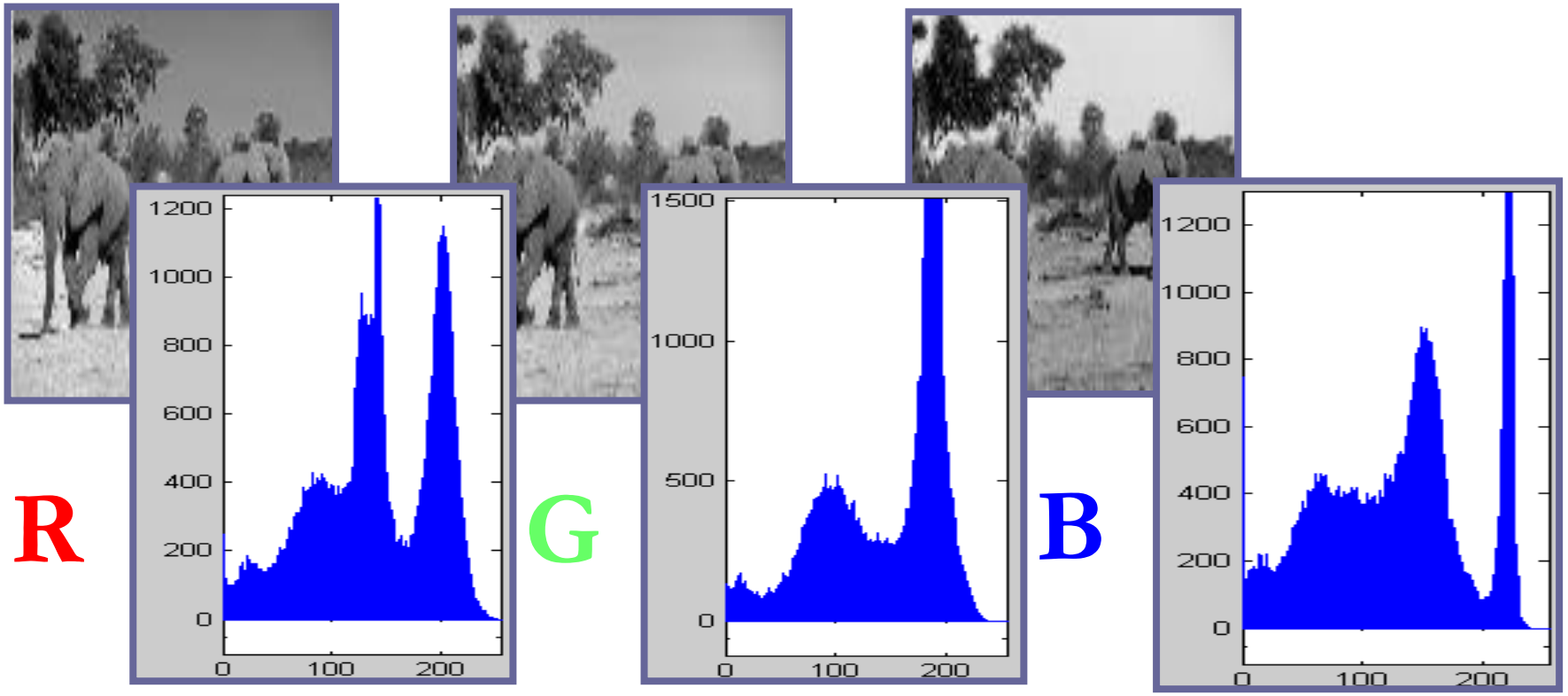
Elephants
Beach

Food
People

# Global Color Histogram (GCH)

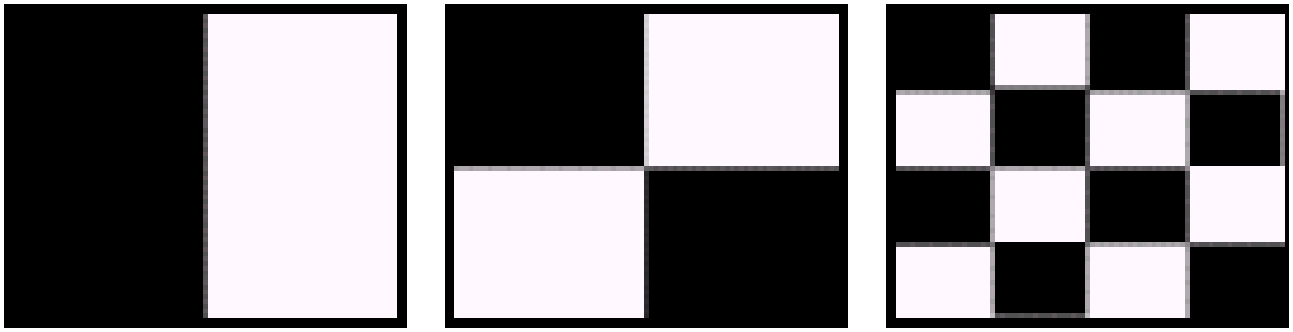
Original image

256x3



# Color Layout Feature

- Color histogram does not include any spatial information about an image and we may find images that have the same color histogram although they have different color distributions



Three images with similar color histograms but different color distributions

# Color Layout Feature vector

- **The steps for creating the color layout feature vector from an image are:**

**16x16x3**

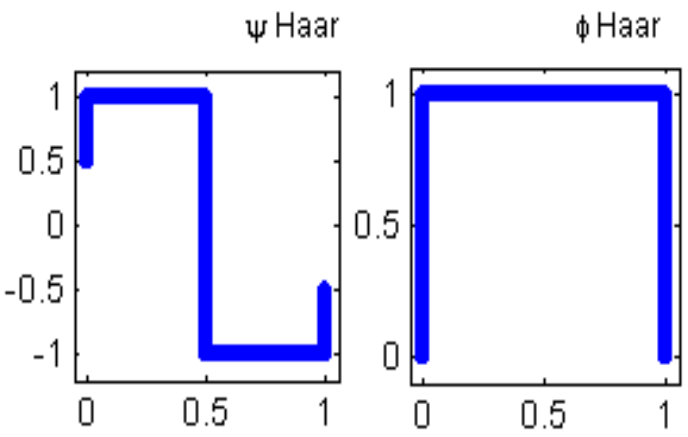
1. Divide the image into 16x16 sub-blocks.
2. Extract the color feature components for each sub-block (Identifying the three components *R*, *G*, and *B* for each block).
3. Calculate the average for each of the three components in each sub-block.
4. Then construct the color layout feature vector (16x16x3) that will represent the color layout feature.

# Haar and Daubechies Wavelets

Daubechies family includes the **Haar wavelet**, written as 'DB1, the simplest wavelet imaginable and certainly the earliest.

$$\phi := x \rightarrow \begin{cases} 1 & 0 \le x \text{ and } x < 1 \\ 0 & otherwise \end{cases}$$

$$\psi := x \rightarrow \begin{cases} 1 & 0 \le x \text{ and } x < \frac{1}{2} \\ -1 & \frac{1}{2} \le x \text{ and } x < 1 \\ 0 & otherwise \end{cases}$$

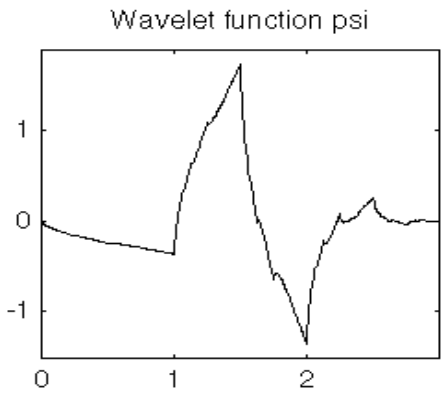



The Haar wavelet $\psi$ and its scale function $\Phi$

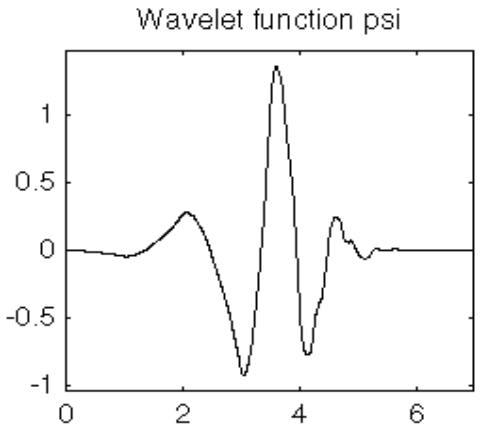
Where $\phi$ is called the scale of the Haar wavelet and $\psi$ is the actual wavelet

# Haar and Daubechies Wavelets

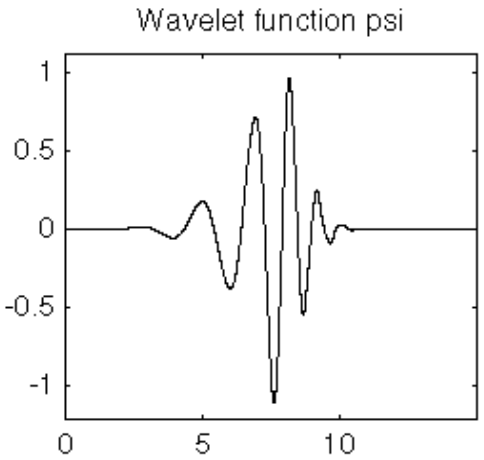
DB2




DB4




DB8

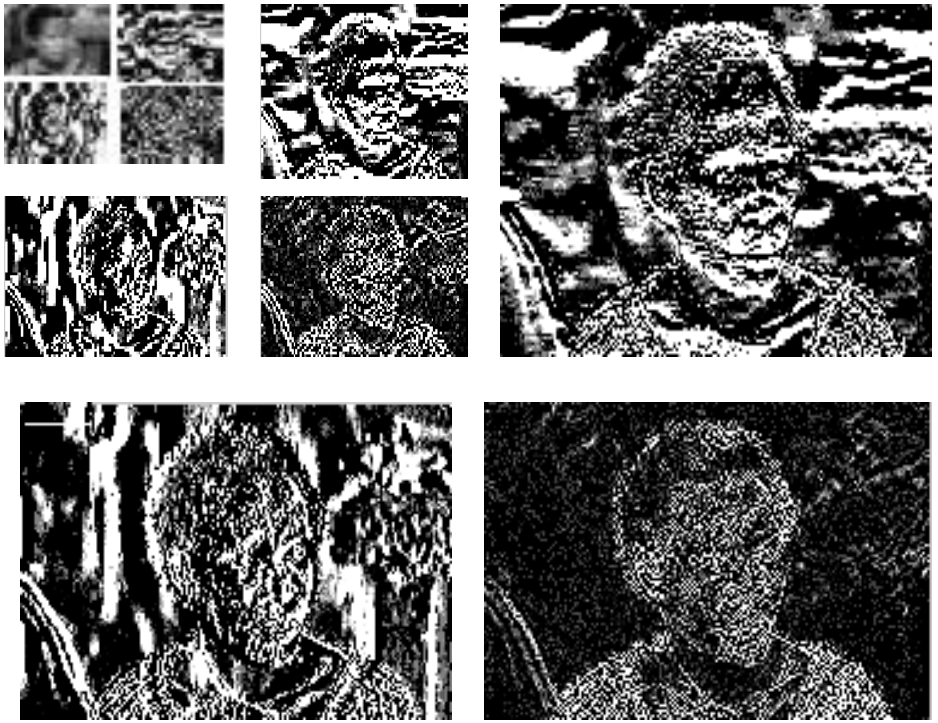



From left to right: the mother wavelets Daubechies DB2, DB4, and DB8

- The wavelet analysis of an image gives four outputs at each level of analysis *l (we used 3-levels)*, one approximation and three details: the approximation $A_l$, horizontal details $H_l$, vertical details $V_l$, and diagonal details $D_l$.

# Haar and Daubechies Wavelets

| | | | |
|---|---|---|---|
| $A_3$ | $H_3$ | $H_2$ | Horizontal Details $H_1$ |
| $V_3$ | $D_3$ | | |
| $V_2$ | | $D_2$ | |
| Vertical Details $V_1$ | | | Diagonal Details $D_1$ |



3 levels wavelet analysis

# Haar and Daubechies Wavelets

## ❑ Texture feature vector:

The energy of each sub-band image is calculated using the following relation:

**3 levels –**

$$E = \frac{1}{MN} \sum_{i=1}^{m} \sum_{j=1}^{n} \left| X(i, j) \right|$$

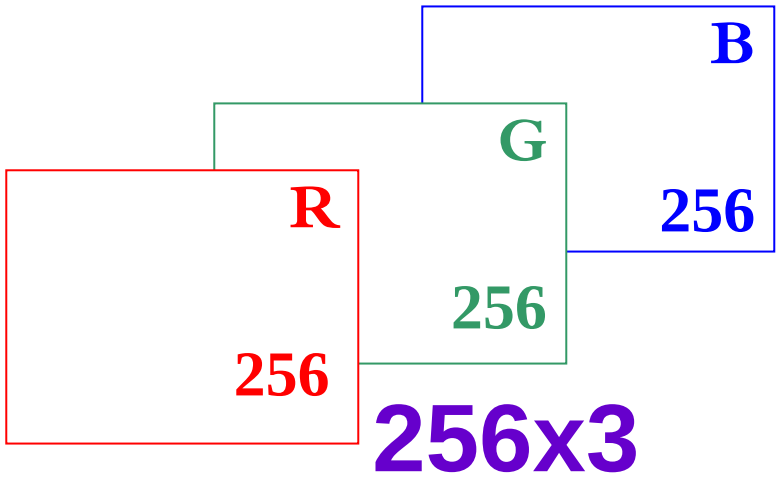
**3 sub bands**

Where:

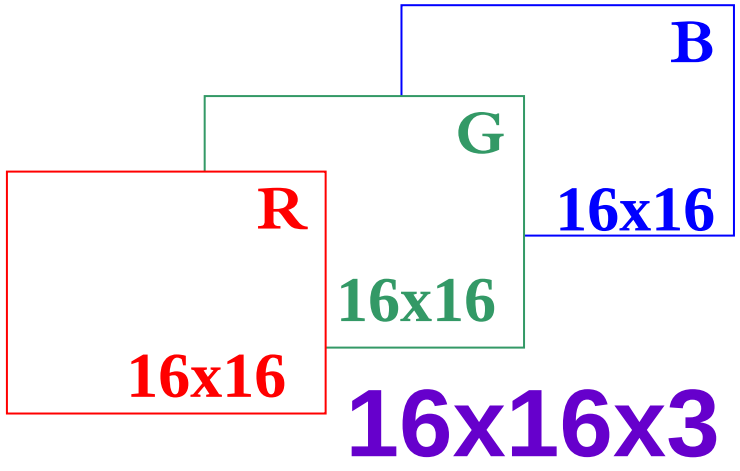
*M* and *N:* are the dimensions of the image,

*X* is the intensity of the pixel located at row *i* and column *j*.

- Color feature vector




- Color layout feature vector




- Texture feature vector (consists of energies)

| Wavelets analysis levels | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Level 1 | | | Level 2 | | | Level 3 | | |
| H1 | V1 | D1 | H2 | V2 | D2 | H3 | V3 | D3 |

# ❑ Manhattan or L1 Distance

$$D_{i,k} = \sum_{j=1}^{G} \left| \frac{H_i(j)}{M_i * N_i} - \frac{H_k(j)}{M_k * N_k} \right|$$

Distance between two histograms

Where:

$H_i(j)$ denote the histogram value for the $i^{th}$ image, $j$ is one of the G possible gray levels,

$M_i * N_i$ is the number of pixels in an image $i$,

$M_k * N_k$ is the number of pixels in image $k$, and $M$ is the number of rows and $N$ is the number of columns.

$$D^C = 0.299 * D^C(R) + 0.587 * D^C(G) + 0.114 * D^C(B)$$

# ❑ Euclidean Distance

## ❑ Euclidean Distance measure:

$$D_i^{CL} = \sqrt{\sum_{s=1}^{S} \left(M_s - N_{s,i}\right)^2}$$

**Texture & Color layout**

Where:

$S$ is the length of the color layout feature vector.

$D_i^{CL}$ is the Euclidean distance between the query image feature vector $M$ and $N_i$ is the feature vector of the $i^{th}$ database image.

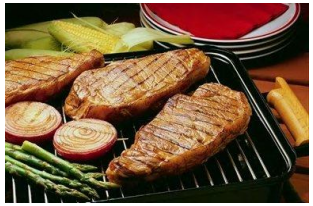
$$D^{CL} = 0.299 * D^{CL}(R) + 0.587 * D^{CL}(G) + 0.114 * D^{CL}(B)$$


**Animals**


**Buses**


**Beach**


**Horses**

**Food**


**Roses**


**People**


**Buildings**

The test database downloaded from: http://wang.ist.psu.edu/~jwang/test1.tar

# Performance Evaluation

The Methods used to measure the performance of content-based image retrieval systems include:

- **Precision and Recall**

$$\Pr ecision = \frac{r}{N} \qquad \text{Re}\, call = \frac{r}{R}$$

Where r is the no. of relevant retrieved images, N total number of retrieved images, and R is the no. of relevant images in the database

- **Before after comparison**
- **Rank of the best match**
- **Retrieval Accuracy and Noise Graphs**

# Performance Evaluation

## ❑ Retrieval Accuracy:

The ratio between the number of relevant (belongs to the same category) retrieved images and the total number of retrieved image

$$\text{Retrieval Accuracy} = \frac{\text{No. of relevant retrieved images}}{\text{Total No. of retrieved images}}$$

Sometimes known as a single precision

## ❑ Retrieval Accuracy and Noise Graphs

# GCH alone

3


Query Image
0.3882
0.402772
0.41372
0.435226
0.449579
0.455721
0.470281
0.486527
0.489863
0.494662
0.501086
0.503067
0.505442
0.510568
0.515943
0.519795
0.51987
0.524827
0.527702

# The combination of GCH and Haar wavelet

# The combination of GCH and Haar wavelet when adding the color layout

17