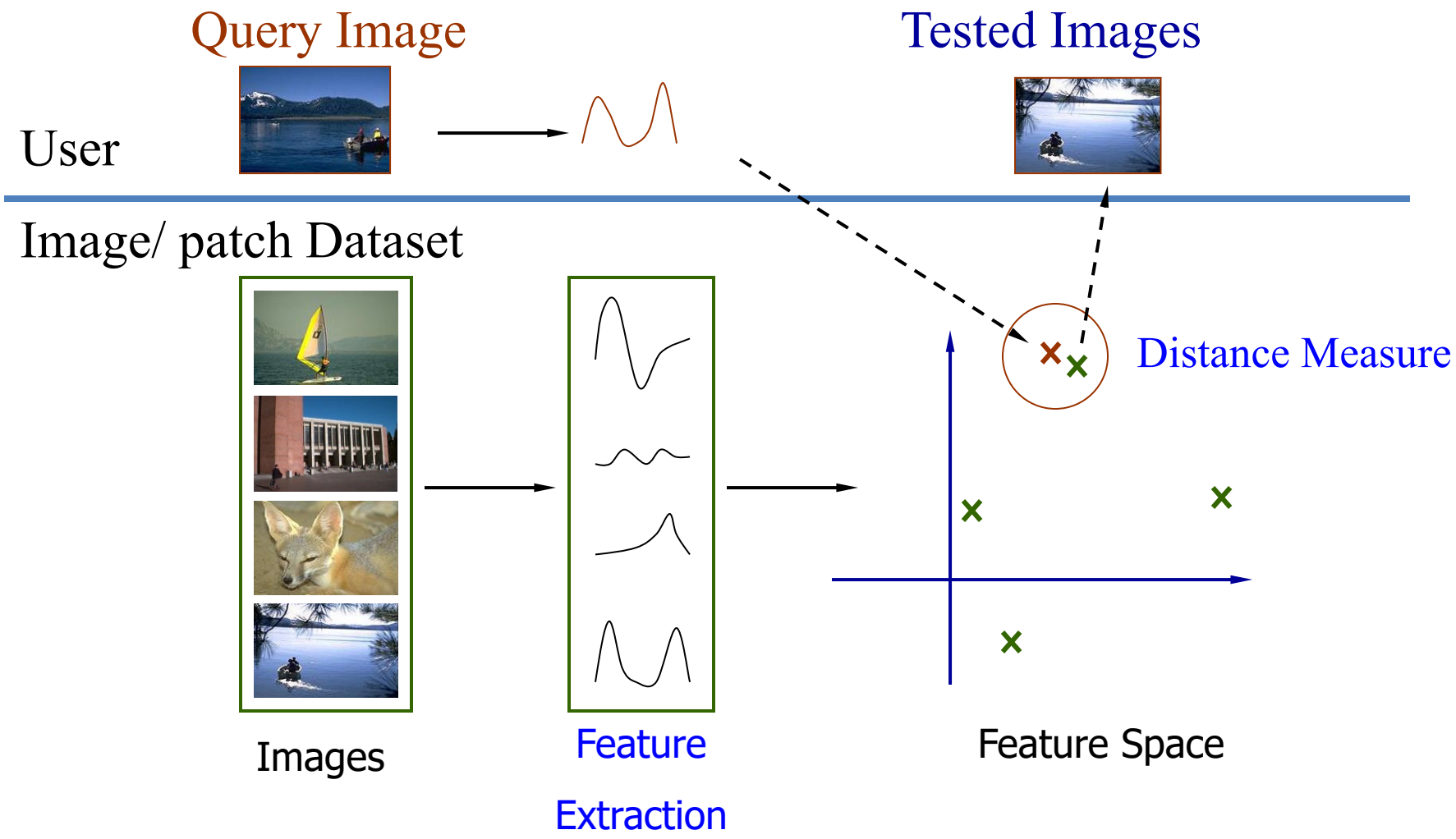
Project Tutorial 2

CS4185 Multimedia Technologies and Applications

Image Features / Distance Measures



How to improve the performance?

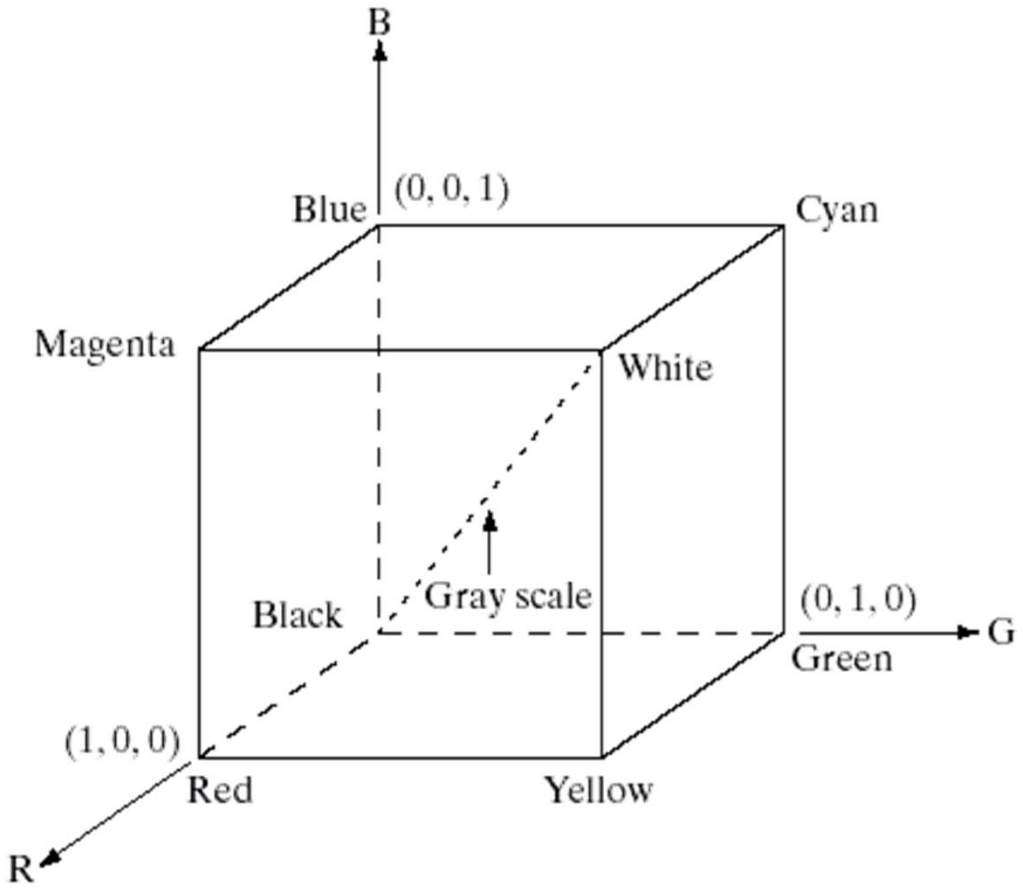
- Possible solutions:
  - Utilize color information.
  - Utilize edge and shape information.
  - Using different layout.
  - Features fusion.

Color Models

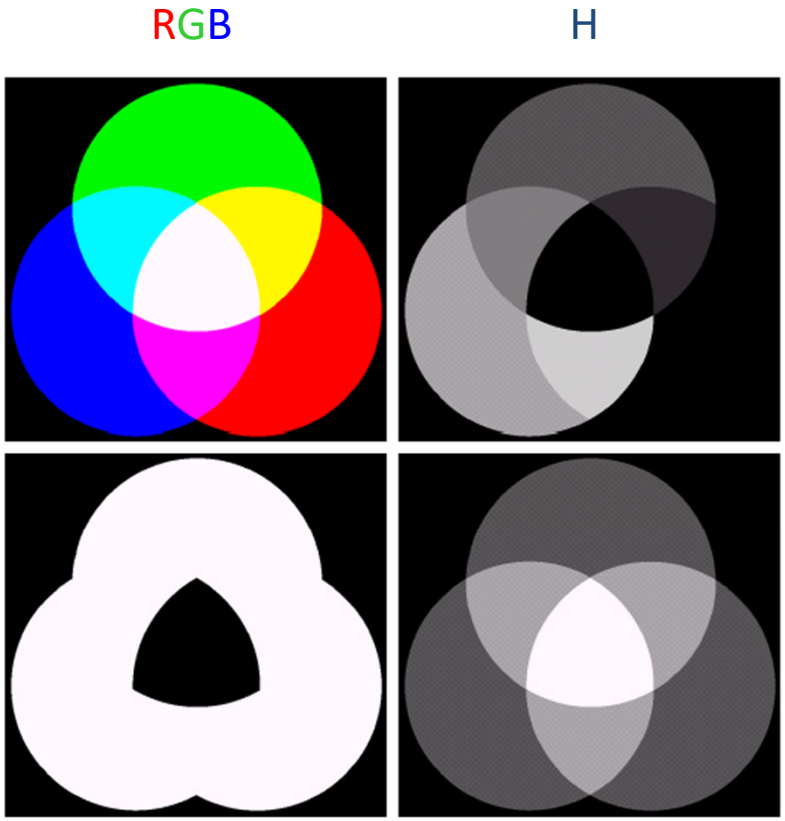
- The purpose of a color model (also called color space or color system) is to facilitate the specification of colors in some standard way.
- **RGB** (red, green, blue) for monitor, video camera.
- **CMY** (cyan, magenta, yellow), **CMYK** (CMY, black) model for color printing.
- **HSI** model, which corresponds closely to the way humans describe and interpret colors.

The RGB Color Models

FIGURE 6.7
Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).



The HSI Color Models



a b
c d

FIGURE 6.16 (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.

Try different models in OpenCV

```
cvtColor(img_rgb,img_hsv,CV_RGB2HSV);
```

Fig. 13a. Color photograph

b. CIELAB L^*

c. Rec. 601 luma Y'

d. Component average: "intensity" I

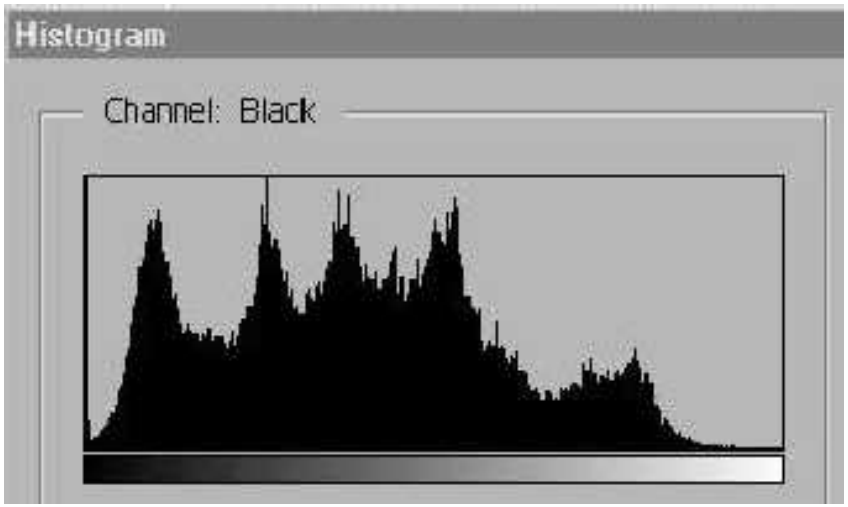
e. HSV value V

f. HSL lightness L

Histogram

- Frequency count of each individual color
- Most commonly used color feature representation

Image



Corresponding histogram

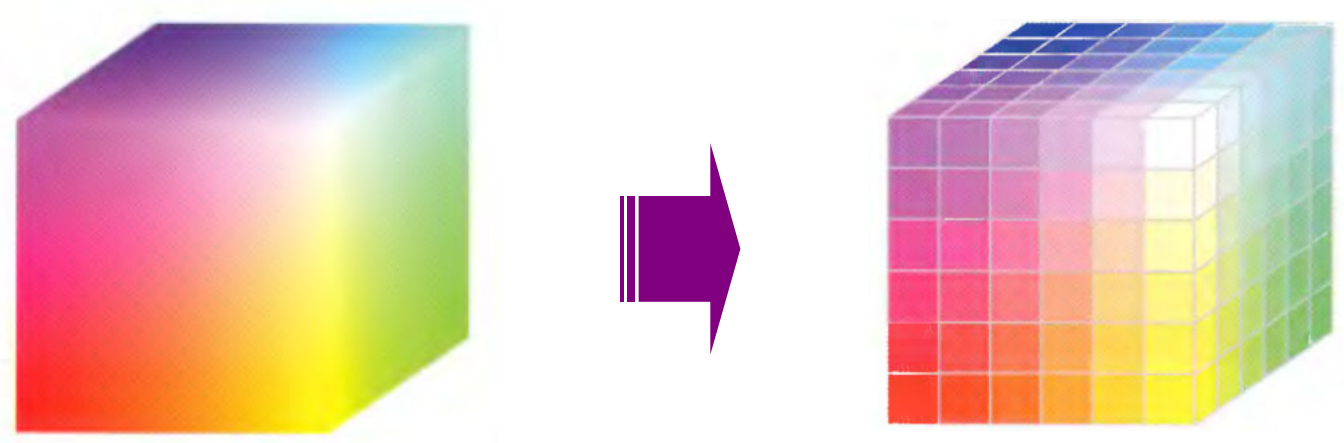
Histogram

$$\mathbf{H}(b) = \sum_{q=1}^{W} Q(\mathbf{I}_q, b), \quad b = 1, \ldots, B,$$

$$Q(I_q, b) = \begin{cases} 1, & I_q = b \\ 0, & otherwise \end{cases}$$

Color Histograms

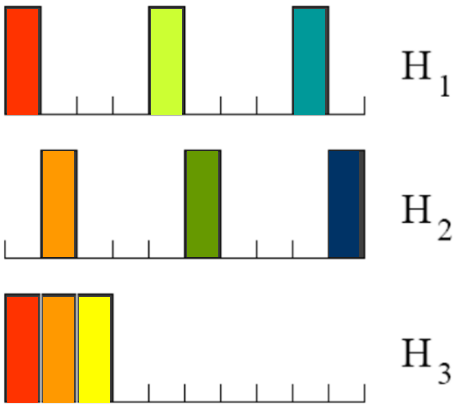
- Quantization of color space



  - Quantization is important: size of the feature vector.
  - When no color similarity function used:
    - Too many bins – similar colors are treated as dissimilar.
    - Too little bins – dissimilar colors are treated as similar.

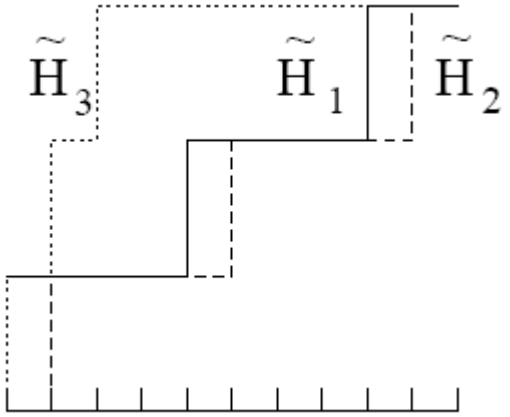
Color histograms: main disadvantages

1. Color similarity across histogram bins is not considered:



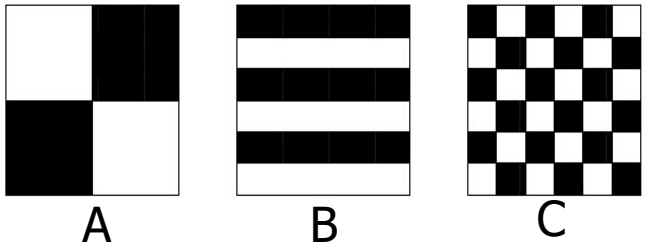
$d(H_1, H_2) > d(H_1, H_3)$

- Cumulative histograms

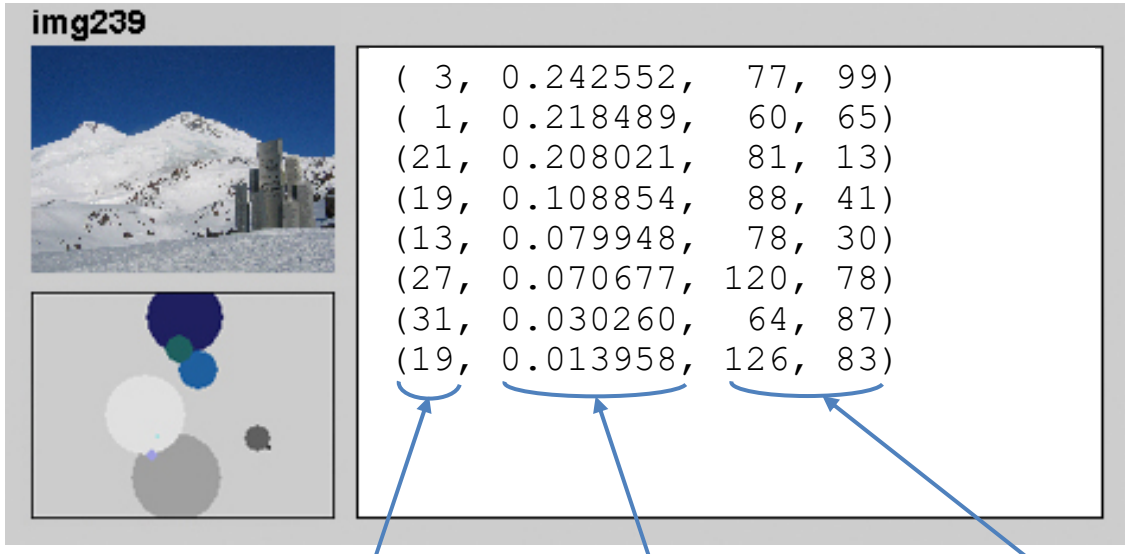


Color histograms: main disadvantages

2. Spatial color layout is not considered:



$H_A = H_B = H_C$



Practice Time

- Use OpenCV to read&show an image
- Use OpenCV to convert the color space
- Calculate histogram of an image
- Apply histogram to Image Retrieval

Use OpenCV to read&show an image

- C++

```cpp
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/nonfree/nonfree.hpp"
#include "opencv2/nonfree/features2d.hpp"

#include <iostream>
#include <stdio.h>
#include <algorithm>

using namespace std;
using namespace cv;


int main() {
    // read&show an image from file
    Mat img = cv::imread("D:/2.jpg");
    if (!img.data) {
        cout << "Can't find the image!" << endl;
        exit(-1);
    }
    cv::imshow("Show An Image", img);

    //wait util that user press Esc Key
    while (cvWaitKey() != 27);
    return 0;
}
```

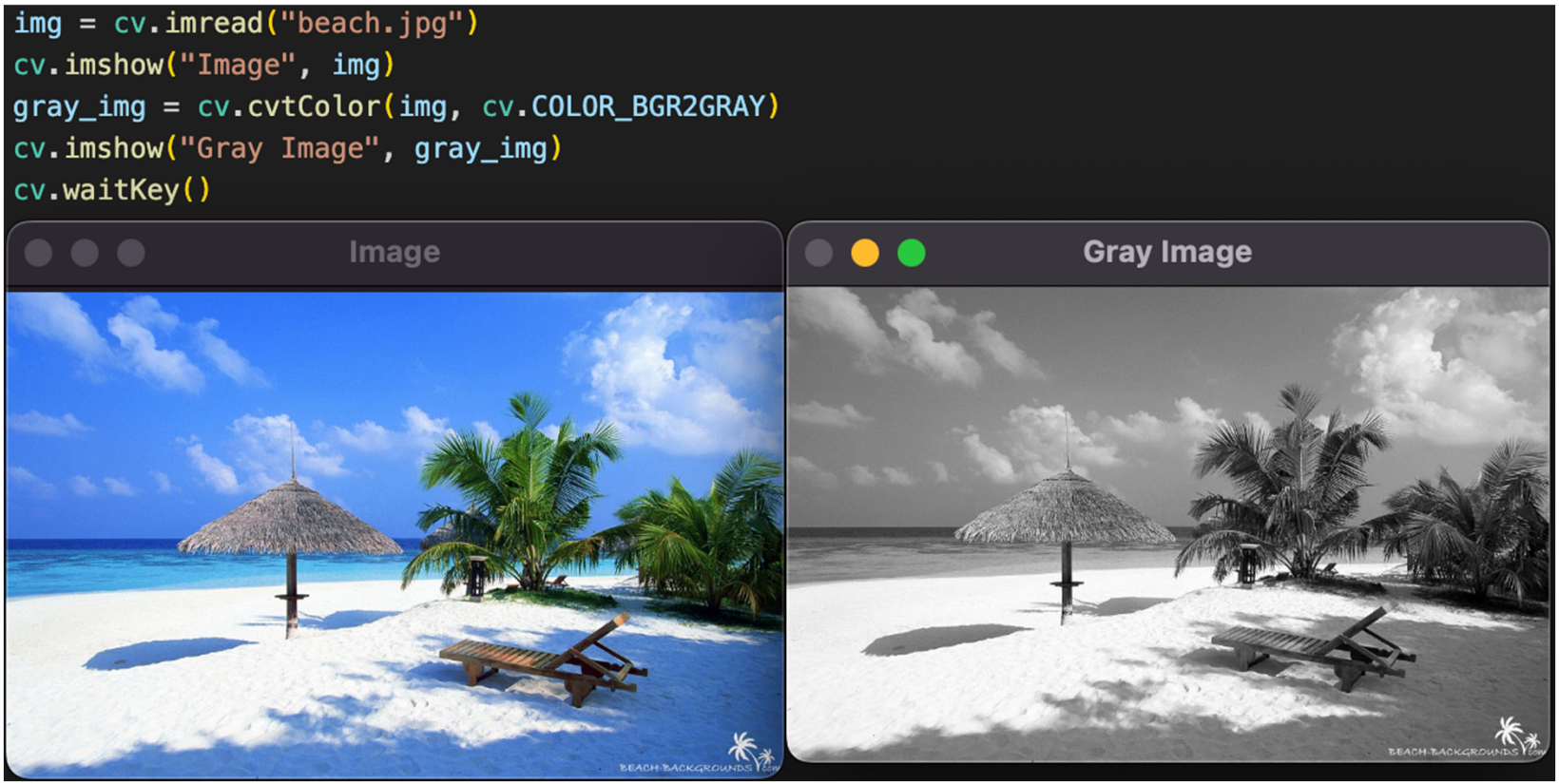
Use OpenCV to read&show an image

- Python

```python
import cv2 as cv
def main():
    img = cv.imread("beach.jpg")
    cv.imshow("Image", img)
    cv.waitKey()
```

Use OpenCV to convert the color space

```
img = cv.imread("beach.jpg")
cv.imshow("Image", img)
gray_img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
cv.imshow("Gray Image", gray_img)
cv.waitKey()
```



Calculate historgram of an image

cv.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])

1. images : it is the source image of type uint8 or float32. it should be given in square brackets, i.e., "[img]".
2. channels : it is also given in square brackets. It is the index of channel for which we calculate the histogram. For example, if input is a grayscale image, its value is [0]. For a color image, you can pass [0], [1] or [2] to calculate the histogram of blue, green or red channel, respectively.
3. mask : mask image. To find histogram of a full image, it is given as "None". But if you want to find histogram of a particular region of an image, you have to create a mask image for that and give it as mask. (I will show an example later.)
4. histSize : this represents our BIN count. Need to be given in square brackets. For full scale, we pass [256].
5. ranges : this is our RANGE. Normally, it is [0,256].

Apply histogram to Image Retrieval

- Original

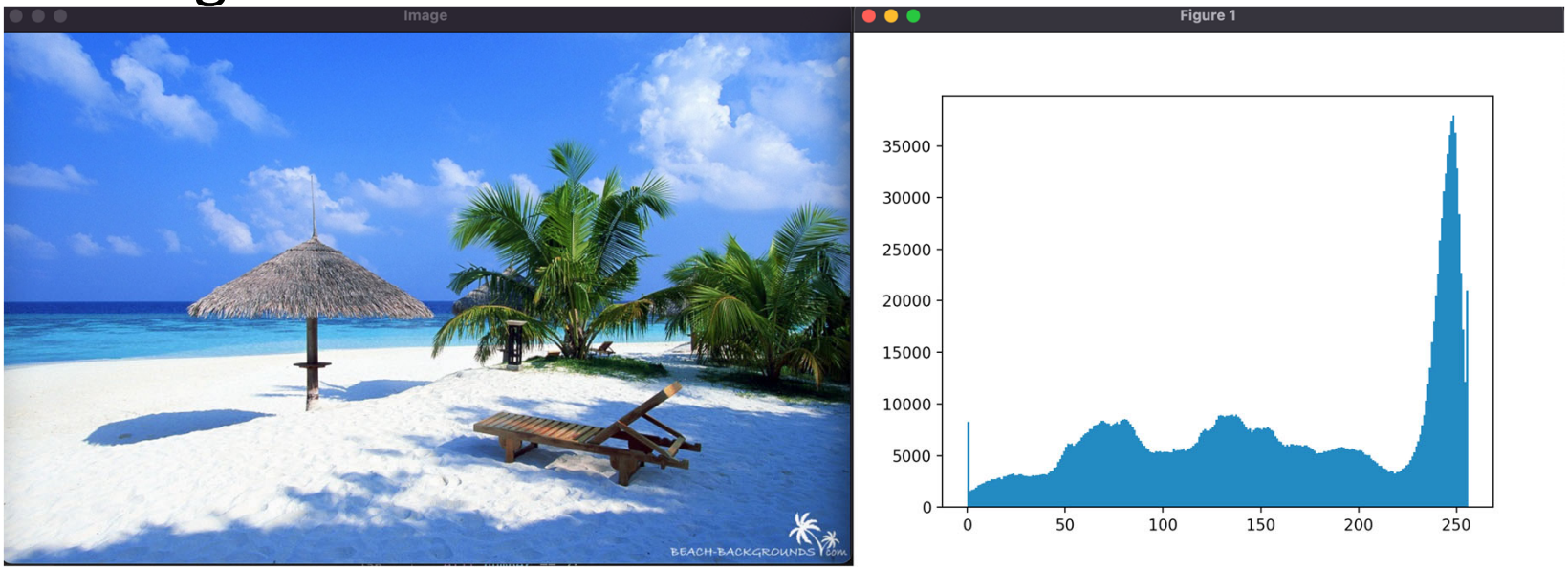
```python
# Compute pixel-by-pixel difference and return the sum
def compareImgs(img1, img2):
    # resize img2 to img1
    img2 = cv.resize(img2, (img1.shape[1], img1.shape[0]))
    diff = cv.absdiff(img1, img2)
    return diff.sum()
```

- Using Hist

```python
def compareImgs_hist(img1, img2):
    width, height = img1.shape[1], img1.shape[0]
    img2 = cv.resize(img2, (width, height))
    num_bins = 10
    hist1 = [0] * num_bins
    hist2 = [0] * num_bins
    bin_width = 255.0 / num_bins + 1e-4
    # compute histogram from scratch

    # for w in range(width):
    #   for h in range(height):
    #       hist1[int(img1[h, w] / bin_width)] += 1
    #       hist2[int(img2[h, w] / bin_width)] += 1

    # compute histogram by using opencv function
    # https://docs.opencv.org/4.x/d6/dc7/group__imgproc__hist.html#ga4b2b5fd75503ff9e6844cc4dcdaed35d

    hist1 = cv.calcHist([img1], [0], None, [num_bins], [0, 255])
    hist2 = cv.calcHist([img2], [0], None, [num_bins], [0, 255])
    sum = 0
    for i in range(num_bins):
        sum += abs(hist1[i] - hist2[i])
    return sum / float(width * height)
```

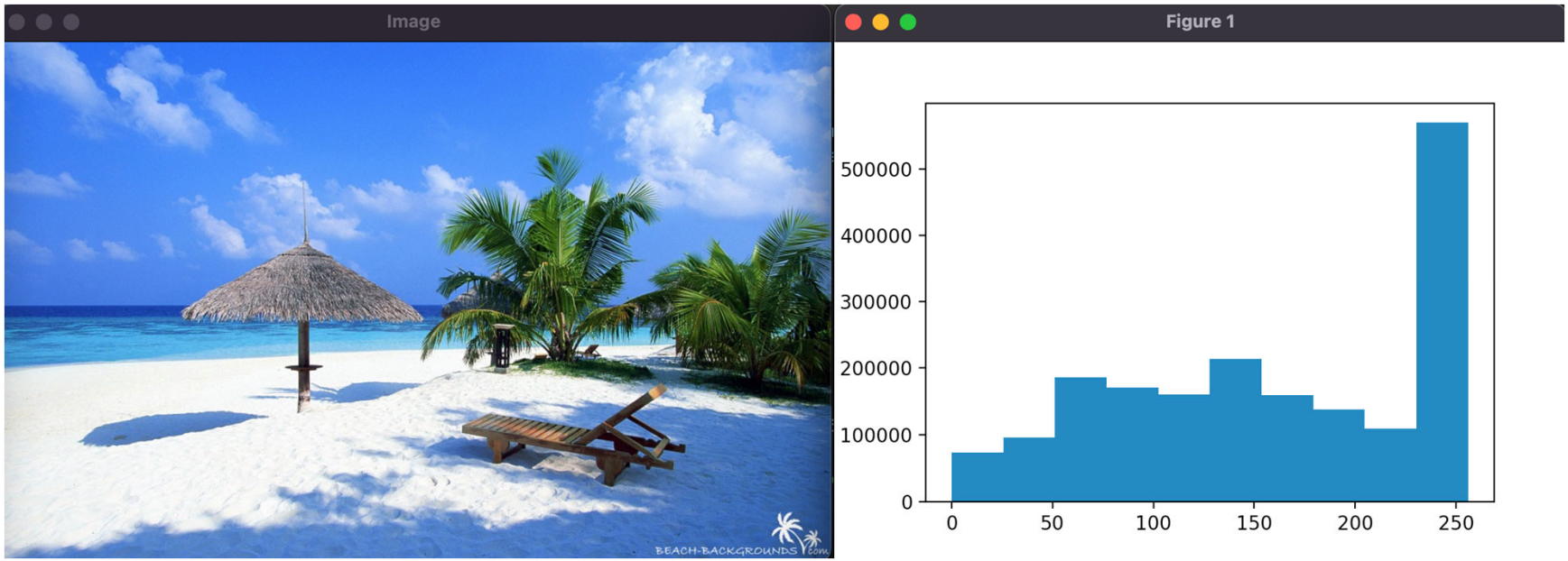
Example 1

- Histogram: the number of bins = 256



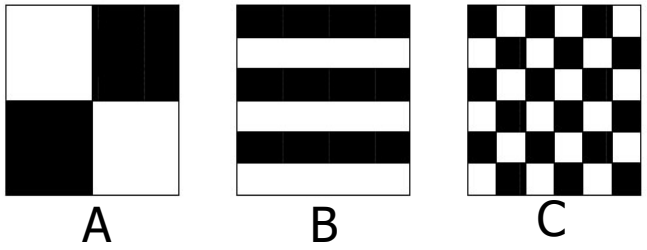
Example 2

- Histogram: the number of bins = 10

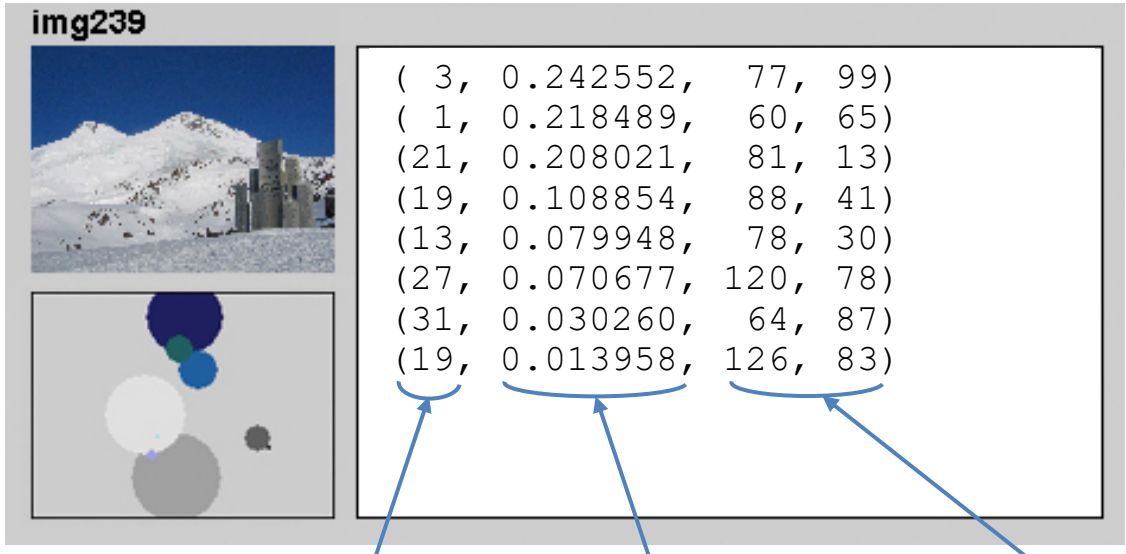


Color histograms: main disadvantages

2. Spatial color layout is not considered:



$H_A = H_B = H_C$



How to improve the performance further? (Next time ...)

- Possible solutions:
  - Utilize color information. (statistics on color information, such as histogram)
  - Using different layout.
  - Utilize edge and shape information.
  - Features fusion.