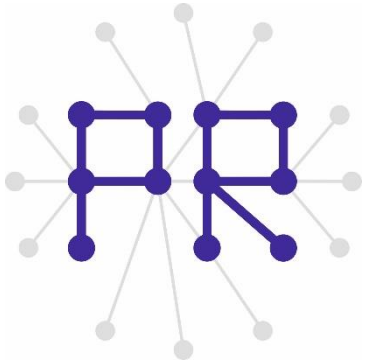
Multimedia Retrieval Exercise Course

3 Different Image Processing Functions

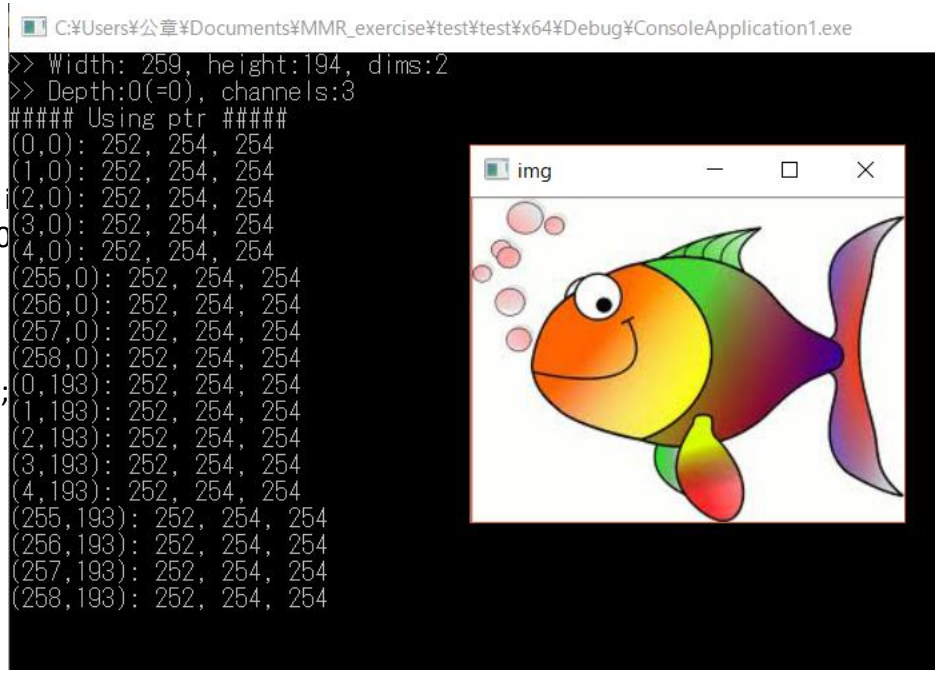
Kimiaki Shirahama, D.E.

Research Group for Pattern Recognition
Institute for Vision and Graphics
University of Siegen, Germany



Last lesson's Program

```
#include "stdafx.h"
#include <iostream>
#include <opencv2/core.hpp>
#include <opencv2/highgui.hpp>

using namespace std;

int main(int argc, const char* argv[]){

    cv::Mat img = cv::imread("C:\\opencv\\sources\\samples\\data\\HappyFish.jpg", 1);
    cout << ">> Width: " << img.cols << ", height:" << img.rows << ", dims:" << img.dims << endl;
    cout << ">> Depth:" << img.depth() << "(=" << CV_8U << "), channels:" << img.channels() << endl;

    cout << "##### Using ptr #####" << endl;
    for (int y = 0; y < img.rows; y++) {
        cv::Vec3b *row = img.ptr<cv::Vec3b>(y);
        for (int x = 0; x < img.cols; x++) {
            if ((y == 0 || y == img.rows - 1) && (x < 5 || x > i
                printf("(%d,%d): %d, %d, %d\n", x, y, row[x][0
        }
    }

    cv::namedWindow("img", cv::WINDOW_AUTOSIZE);
    cv::imshow("img", img);
    cv::waitKey(0);
    cv::destroyAllWindows();

    return 0;

}
```



Overview of Today's Lesson

1. Image resize
2. Color space conversion
3. Edge detection
4. Line detection
5. Image Data Preparation

Image Resize (1/2)

void cv::resize (InputArray src, OutputArray dst, Size dsize,
double fx = 0, double fy = 0, int interpolation = INTER_LINEAR)

- The original image “src” (cv::Mat) is resized and stored into the destination image “dst” (cv::Mat)

- Two typical approaches to specify the “resized size”
  1. Create an empty cv::Mat for “dst”,
     and then use “fx” and “fy” to specify how much you want to scale “src” up/down

  2. Create cv::Mat for “dst” by specifying the desired size, and used “dst.size()” for the “dsize”
     Check the constructor of cv::Mat *http://docs.opencv.org/3.1.0/d3/d63/classcv_1_1Mat.html*

- **One tip** to read function definitions in OpenCV is that, if an input variable involves “=“,
  this variable has the default value specified on the right side of “=“. So, you don’t have to
  specify the value for such a variable, and if this is case, the default value is automatically used.

For more detail, please refer to
http://docs.opencv.org/3.1.0/da/d54/group__imgproc__transform.html#ga47a974309e9102f5f08231edc7e7529d

Image Resize (2/2)

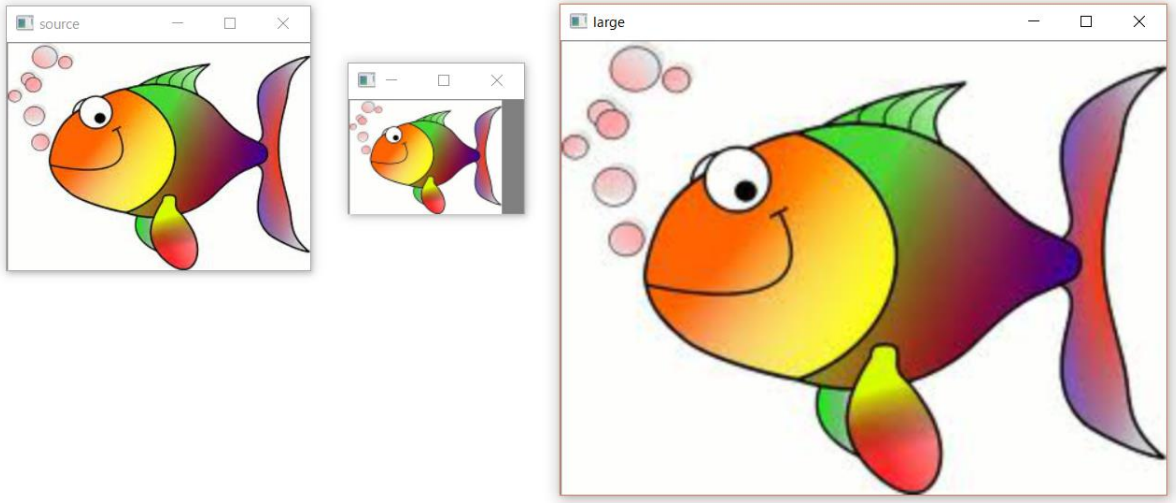
Assuming that the original image has been already loaded into “img”, then

Approach 1

```
cv::Mat img2;
cv::resize(img, img2, cv::Size(), 0.5, 0.5);
```

Approach 2

```
cv::Mat img3(img.rows*2, img.cols*2, img.type());
cv::resize(img, img3, img3.size());
```



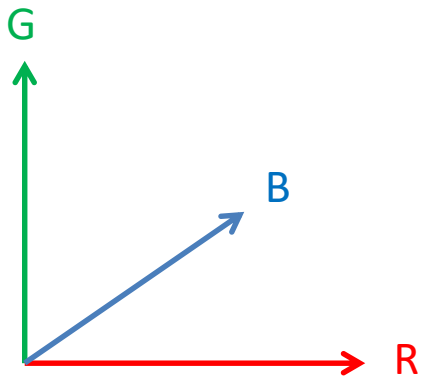
If you are not satisfied with the quality of the resized image, change the interpolation value.

Colour Space Conversion (1/2)

RGB color space

- Red
- Green
- Blue

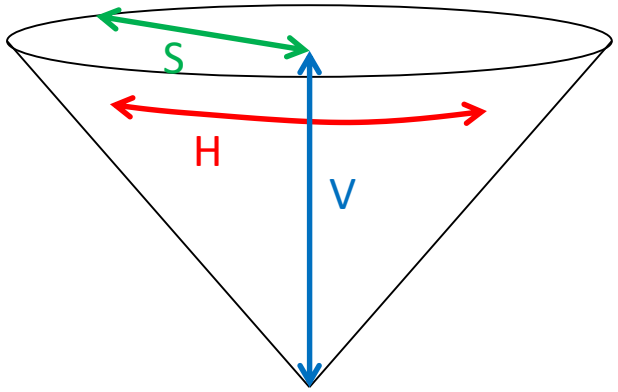
Not intuitive for humans



HSV color space

- Hue: Primary color
- Saturation: Effect of white
- Value: Darkness (effect of black)

More intuitive for humans



void cv::cvtColor(InputArray src, OutputArray dst, int code, int dstCn = 0)

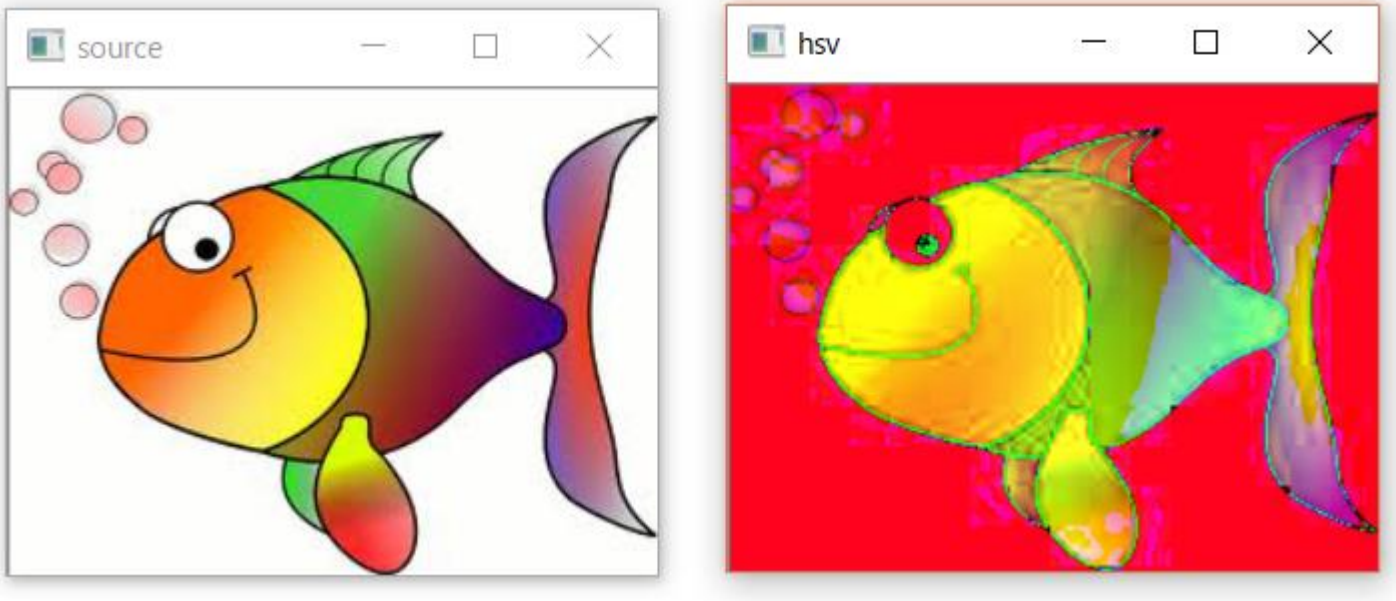
- The original image “src” (cv::Mat) is converted into the destination image “dst” (cv::Mat), by changing the former’s colour space based on “code”

For more detail, please refer to
http://docs.opencv.org/3.1.0/d7/d1b/group__imgproc__misc.html#ga397ae87e1288a81d2363b61574eb8cab

Colour Space Conversion (2/2)

Assuming that the original image has been already loaded into “img”, then

```
cv::Mat hsv;
cv::cvtColor(img, hsv, CV_BGR2HSV);
```



Since the visualisation of PC monitors is based on RGB colour space, HSV images cannot be visualised appropriately.

Edge Detection (1/2)

void cv::Canny(InputArray image, OutputArray edges, double threshold1,
double threshold2, int apertureSize = 3, bool L2gradient = false)

- The input image “image” (cv::Mat) should be grayscale
  (Apply colour space conversion to the original RGB image)
- Detected edges are stored in the grayscale image “edges” (cv::Mat) where
  pixels regarded as edges and non-edges are depicted by white and black, respectively
- “threshold1” is used to examine the connection of edges
- “threshold2” is used to detect strong edges in the initial process.
- “apertureSize” specifies the size of Sobel filter.

Please test various values

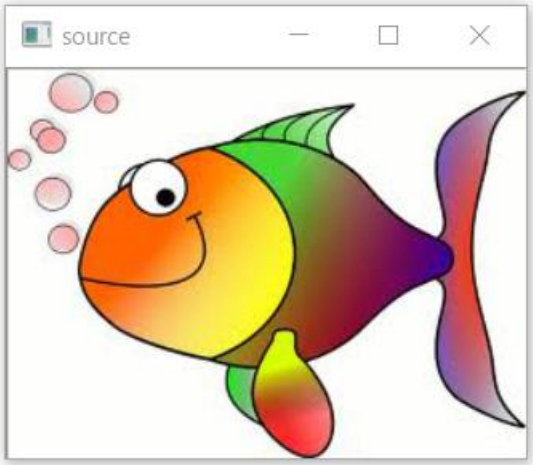
For more detail, please refer to
http://docs.opencv.org/3.1.0/dd/d1a/group__imgproc__feature.html#ga04723e007ed888ddf11d9ba04e2232de

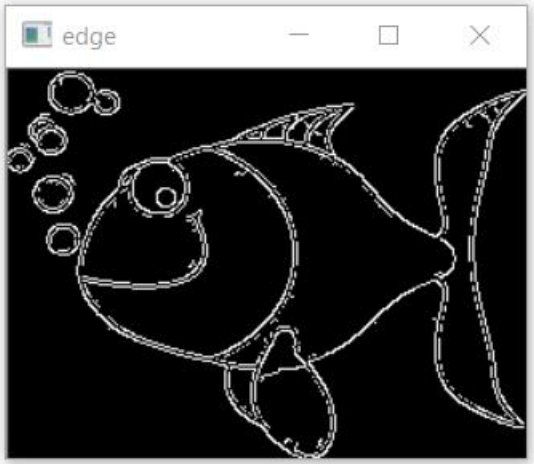
And also this is a good tutorial:
http://docs.opencv.org/3.1.0/da/d5c/tutorial_canny_detector.html

Edge Detection (2/2)

Assuming that the original image has been already loaded into “img”, then

```
cv::Mat gray;
cv::cvtColor(img, gray, CV_BGR2GRAY);
cv::Mat edge;
cv::Canny(gray, edge, 50, 200);
```





Line Detection

void cv::HoughLinesP(InputArray image, OutputArray lines, double rho, double theta, int threshold, double minLineLength = 0, double maxLineGap = 0)

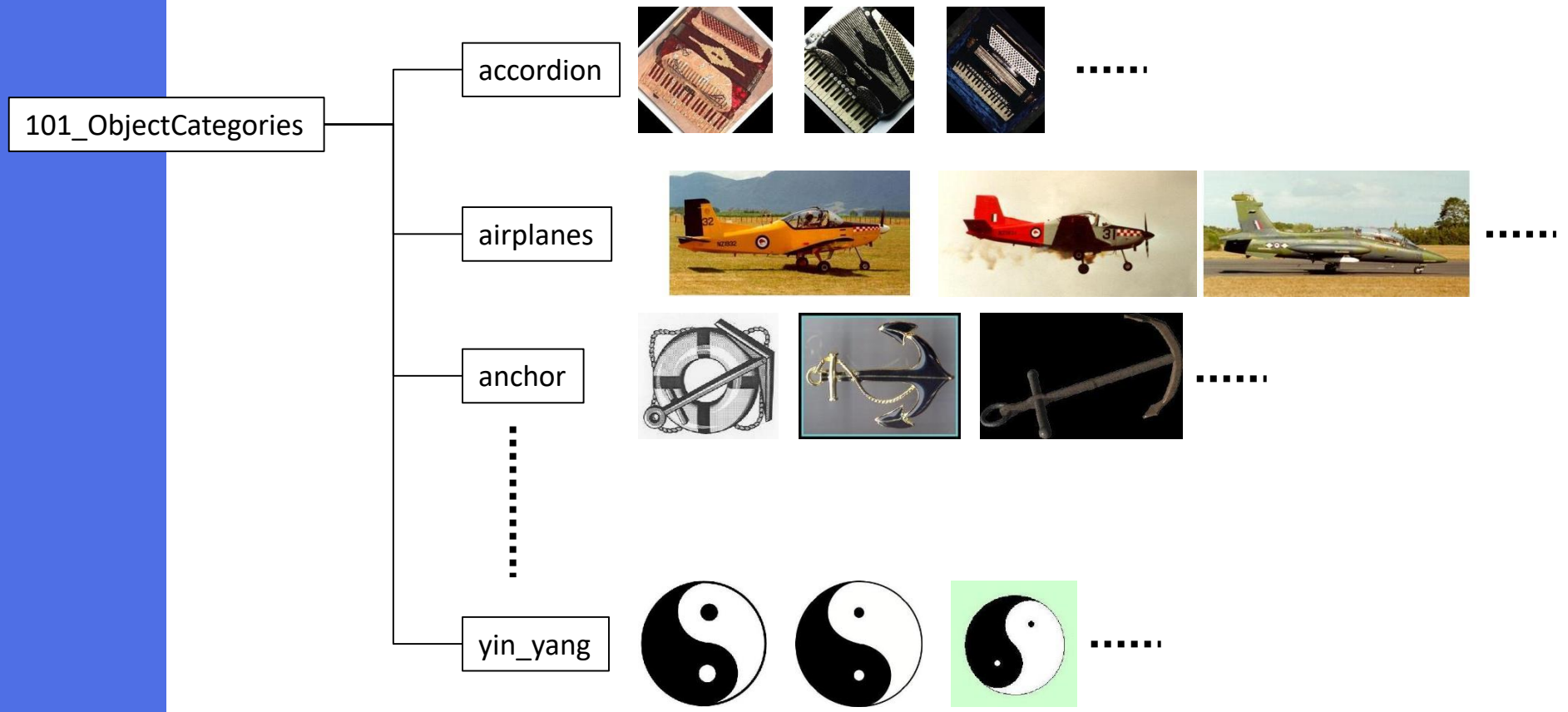
- "image": Input grayscale (8bit, binary) image (usually an image representing detected edges)
- "lines": Array where each element represents a detected line by cv::Vec4i
  (cv::Vec4i is a 4-element vector (x_1, y_1, x_2, y_2) where (x_1, y_1) and (x_2, y_2) are the start and end pixels of a line, respectively)
- "rho" and "theta" are the distance resolution and the angle resolution, respectively
  (Usuallys "rho" is 1 pixel, and "theta" is 1 degree)
- "threshold": Threshold used to detect each line
- "minLineLength": Minimum acceptable line length
- "maxLIneGap": Maximum acceptable gap between points on the same line

http://docs.opencv.org/3.1.0/dd/d1a/group__imgproc__feature.html#ga8618180a5948286384e3b7ca02f6feeb
On this web page, you can find a detailed explanation and a sample code for line detection.

Image Data Preparation

Caltech 101 (http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html)

- Traditional benchmark image data set for evaluating image retrieval/classification methods
- This includes 101 categories, each category contains about 40 to 800 images
- Total data size is about 150MB



We will develop an image retrieval system for this dataset.