Multimedia
(Lab 07)

Spring, 2020

Department of Software

Yong Ju Jung (정용주)

미래창조과학부 SW중심대학
가천대학교 소프트웨어학과
Gachon University Department of Software

Summary

- In this lab, you will learn about
  - Video Capture from a camera or file
  - Block matching algorithm for motion estimation in video
  - Chroma subsampling of image

[Lab07-1] Video Capture

- VideoCapture
  - This class provides C++ API for capturing video from cameras or for reading video files

```cpp
#include "opencv2/opencv.hpp"

using namespace cv;

int main(int, char**)
{
    VideoCapture cap(0); // open the default camera
    if(!cap.isOpened())  // check if we succeeded
        return -1;

    Mat edges;
    namedWindow("edges",1);
    for(;;)
    {
        Mat frame;
        cap >> frame; // get a new frame from camera
        cvtColor(frame, edges, COLOR_BGR2GRAY);
        GaussianBlur(edges, edges, Size(7,7), 1.5, 1.5);
        Canny(edges, edges, 0, 30, 3);
        imshow("edges", edges);
        if(waitKey(30) >= 0) break;
    }
    // the camera will be deinitialized automatically in VideoCapture destructor
    return 0;
}
```

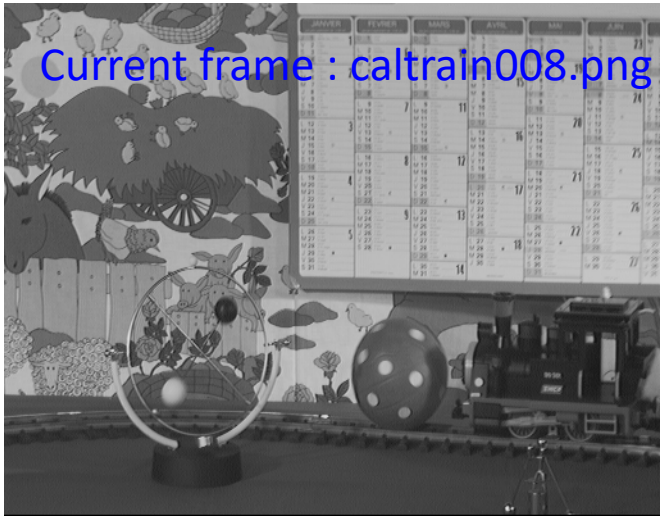
* source: docs.opencv.org

Exercise

- Open a video file using “VideoCapture” class
- Apply an Affine transform for each frame, as you have learned in the previous Lab.
- Playback the geometrically transformed frames in real-time

[Lab07-2]

- Implement the block matching algorithm for motion estimation
  - Load two consecutive image frames
  - Estimate a motion vector for each block
    - Write a motion magnitude map calculated from each motion vector
    - Block size = 16x16, search range 31x31 pixels
  - Display the motion magnitude map for a frame



[Lab07-3]

- Chroma subsampling
  - Load a color Lena image in RGB (using **cv::imread**)
  - Do color transform from RGB to YCbCr 4:4:4
  - Perform Chroma subsampling from 4:4:4 to 4:2:0
  - Reconvert the subsampled YCbCr 4:2:0 to RGB image
  - Display and compare the original & result images (using **cv::imwrite** and **cv::imshow**)

  - You can also try to perform a 4:4:4 to 4:2:2 conversion.

  - You can use the following OpenCV library:
    - cvtColor(src, dst, CV_BGR2YCrCb);
    - cvtColor(src, dst, CV_YCrCb2BGR);

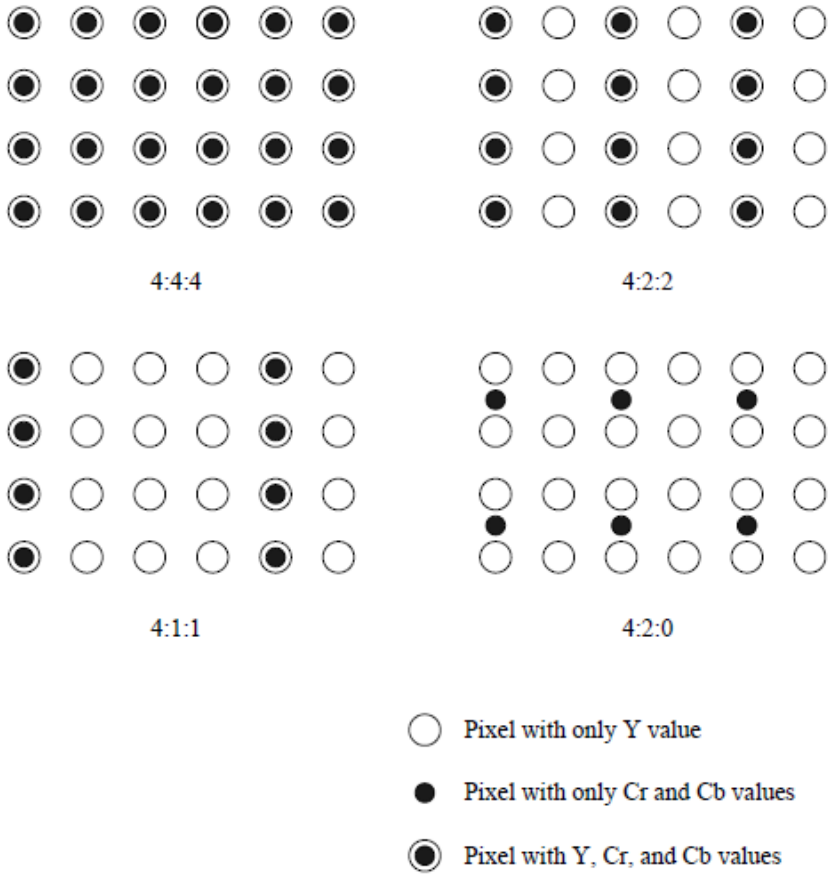


Fig. 5.6: Chroma subsampling.