影像處理、電腦視覺及深度學習概論 (Introduction to Image Processing, Computer Vision and Deep Learning)

Homework 2

TA:

Kevin: i1007673219@gmail.com

Office Hour: 19:00~21:00, Mon.
09:00~11:00, Wed.
At CSIE 9F Robotics Lab.

Notice (1/2)

- Copy homework is strictly prohibited!! Penalty: Grade will be zero for both persons!!
- If the code can’t run, you can come to our Lab within one week and show that your programming can work. Otherwise you will get zero!!
- Due date => 2020/01/02 (Thu.) 23:59:59
  - No delay. If you submit homework after deadline, you will get 0.
- Upload to => 140.116.154.1 -> Upload/Homework/OpenCvdl_Hw2
  - User ID: opencvdl2019   Password: opencvdl2019
- Format
  - Filename: Hw2_StudentID_Name_Version.rar
    - Ex: Hw2_F71234567_林小明_v1.rar
    - If you want to update your file, you should update your version to be v2, ex: Hw2_F71234567_林小明_v2.rar
  - Content: project folder*(including the pictures)
    *note: remove your “Debug” folder to reduce file size

Notice (2/2)

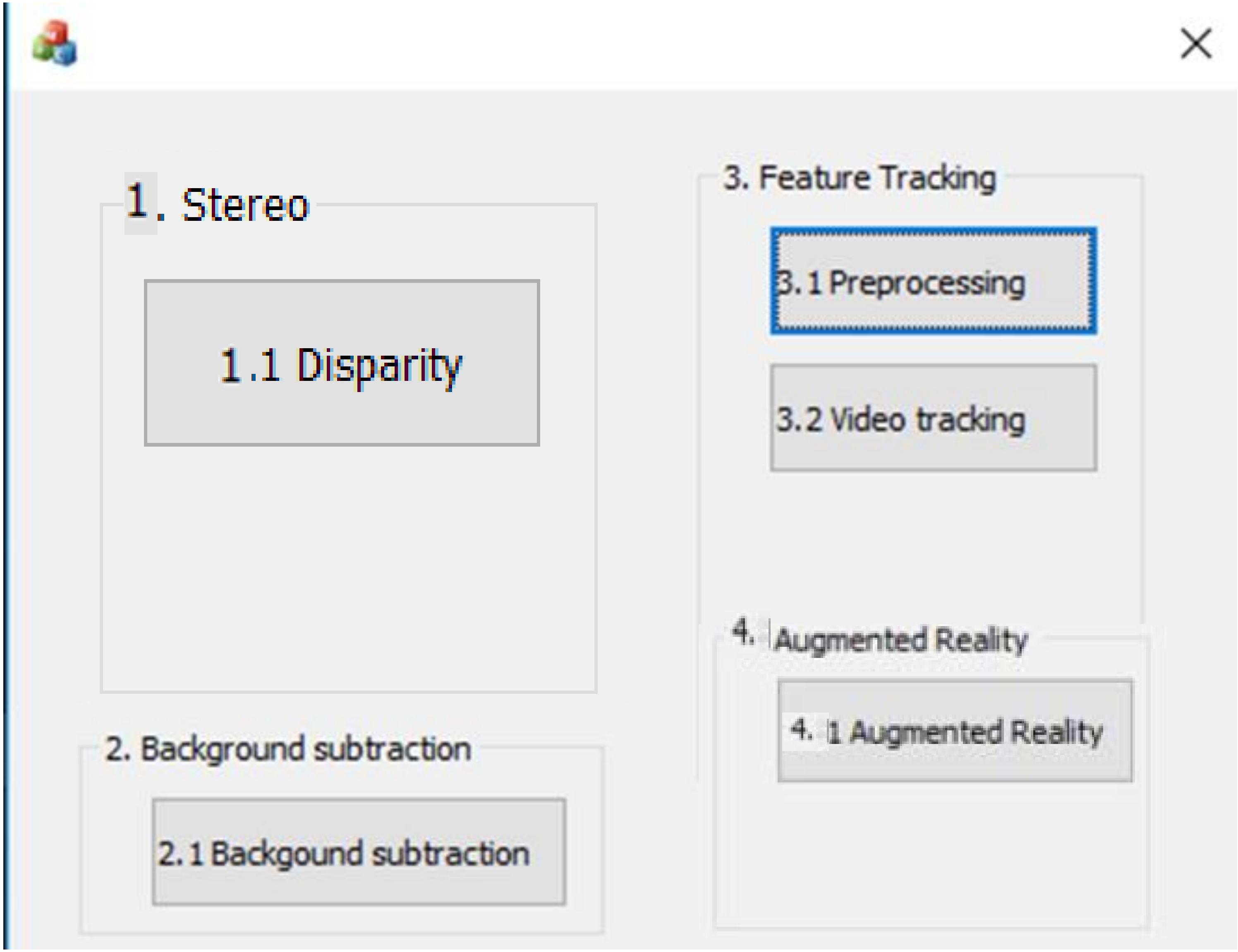
- C++ (check MFC guide in ftp)
  - OpenCV 3.3.1 (https://opencv.org/release.html)
  - Visual Studio 2015 (download from http://www.cc.ncku.edu.tw/download/)
  - UI framework: MFC
- Python
  - Python 3.7 (https://www.python.org/downloads/)
  - Tensorflow 2.0 / PyTorch 1.3.0
  - opencv-contrib-python (3.4.2.17)
  - Matplotlib 3.1.1
  - UI framework: pyqt5 (5.11.3)

Assignment scoring (Total: 100%)

0. GUI	
1. Stereo (30%)	(出題：Kris)
2. Background Subtraction (20%)	(出題：Yi Yuan)
3. Optical Flow (30%)	(出題：Shaku)
3.1 Preprocessing (10%)	
3.2 Video tracking (20%)	
4. AR (20%)	(出題: Rex)

0. GUI

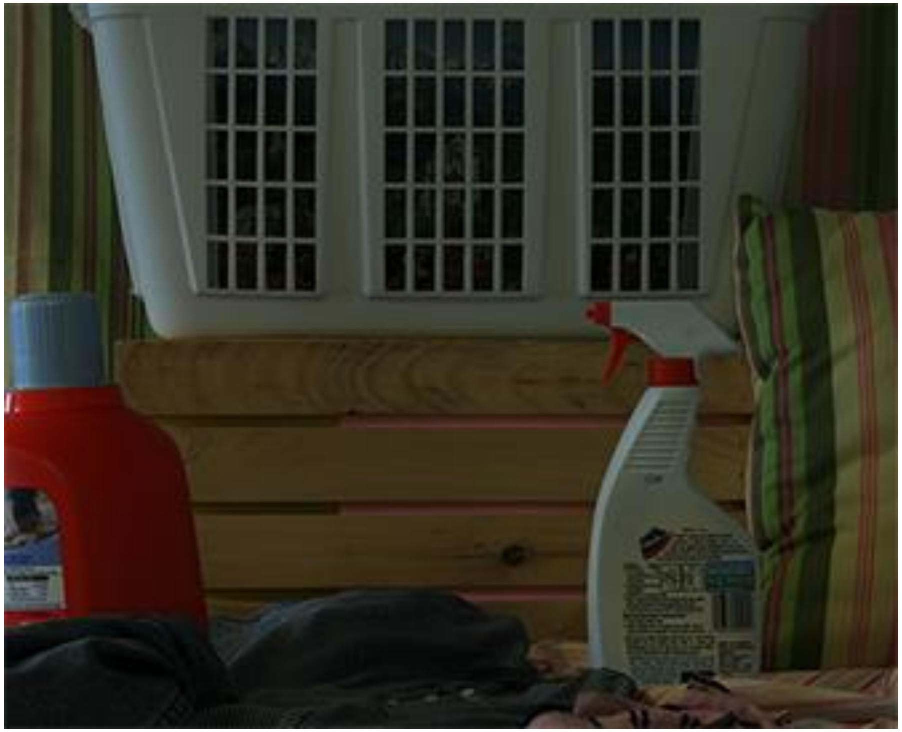
- Use MFC to create GUI like following figure.



1. (30%) Stereo Disparity Map

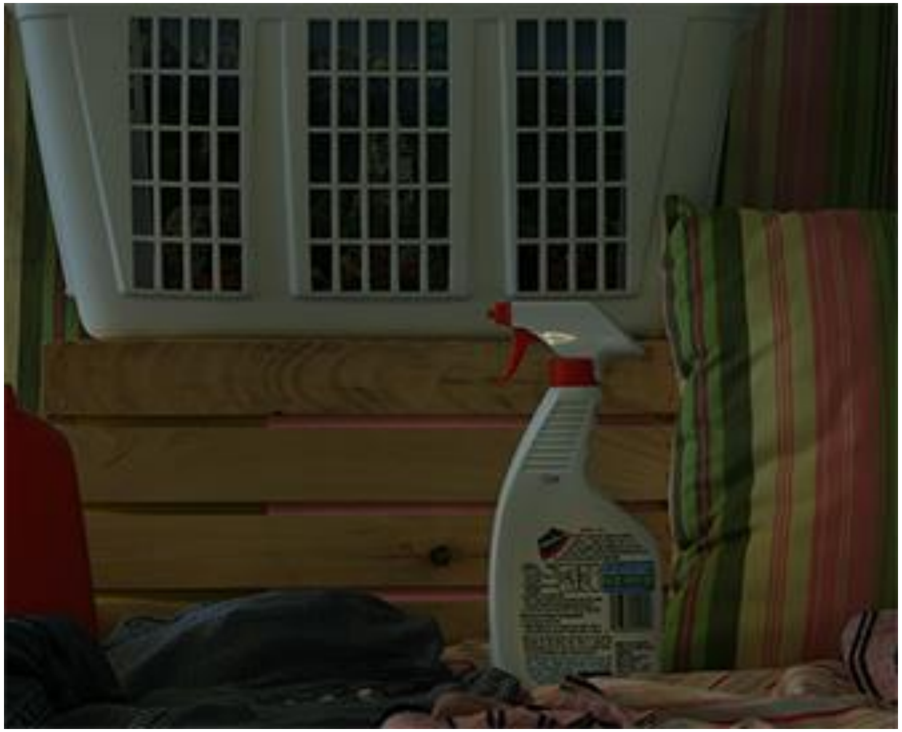
(出題：Kris)

- ❑ Given: a pair of images, imL.png and imR.png (have been rectified)
- ❑ Q: 1) Find the disparity map/image based on Left and Right stereo images.
  2) Compare the result of the disparity map with and without the left-right disparity check, and mark the difference between the above result in red.



imL.png

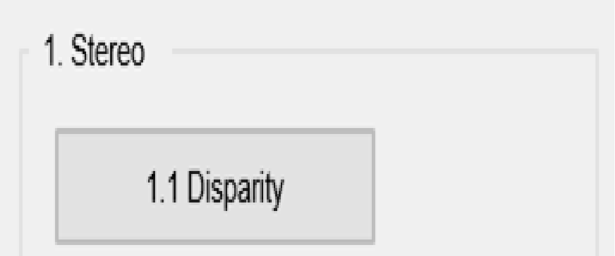
Left Image (Reference Image)



imR.png

Right Image

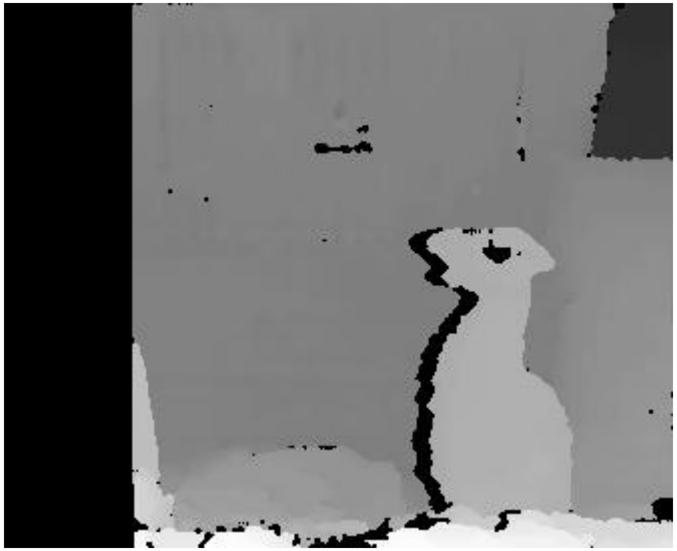
1.1 (30%) Disparity Map



- ❑ Q: 1) Click button “1.1” to show the disparity map
- ❑ Guides:
  - (1) Window Size: 9 = 3*3 pixel
  - (2) Search range and direction:
    - ▪ Disparity range: 0~64 pixels.
      - • Map disparity range 0~64 pixels to gray value range 0~255 for the purpose of visualization.
    - ▪ If the left image is the reference image (the one used to cal. depth info for each pixel of that img), then the search direction at right image will go from the right to left direction.

- ❑ Hint: OpenCV Textbook Chapter 12 (P.451)

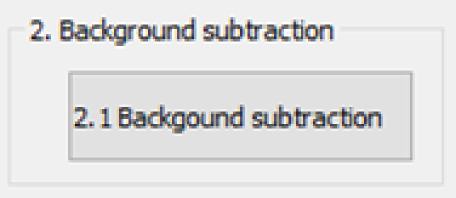
  StereoBM::create(64, 9);



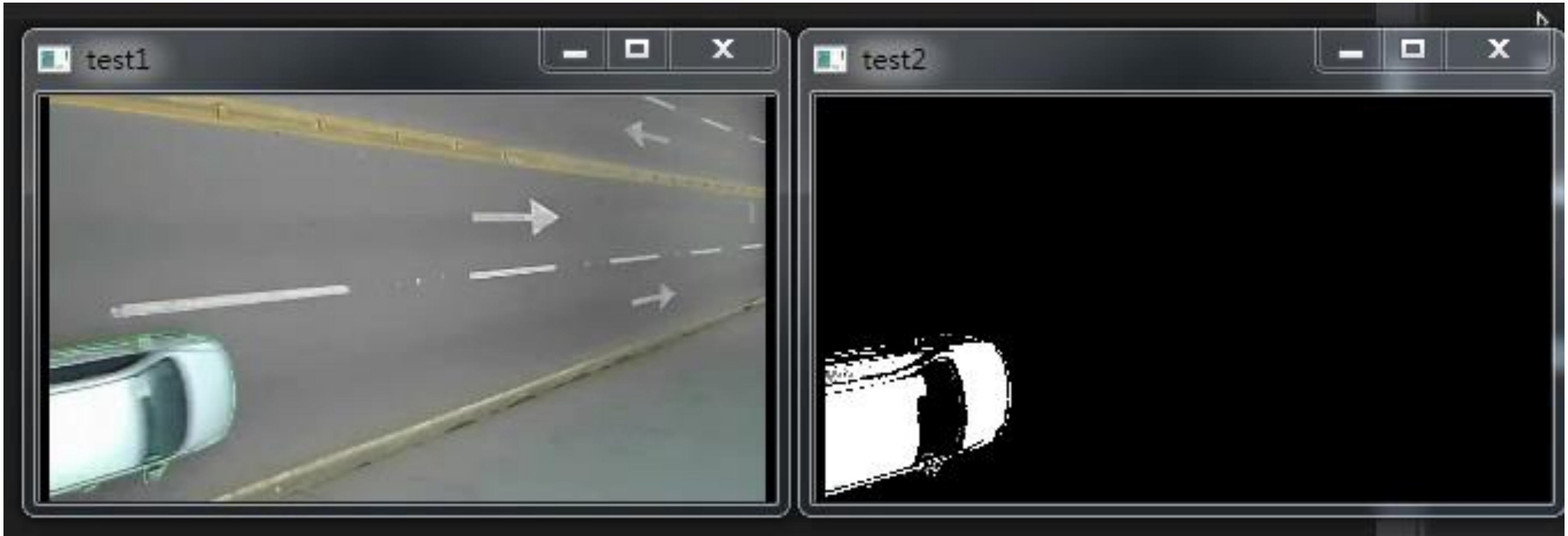
Result

2. (20%) Background Subtraction (出題：Yi Yuan)

- Given a video: bgSub.mp4
- Q: 1) Click the button “2.1 Background Subtraction” to open two windows:
    - One shows the original video, bgSub.mp4
    - The other is the foreground video.
    - Use first 50 frames to train the background model
    - You need to subtract background model to find the foreground object (white part)



- Hint : textbook p372 ~ p376
- Demo video:



3. (30%) Optical Flow (出題：Shaku)

3.1 Preprocessing (10%)

3.2 Video tracking (20%)

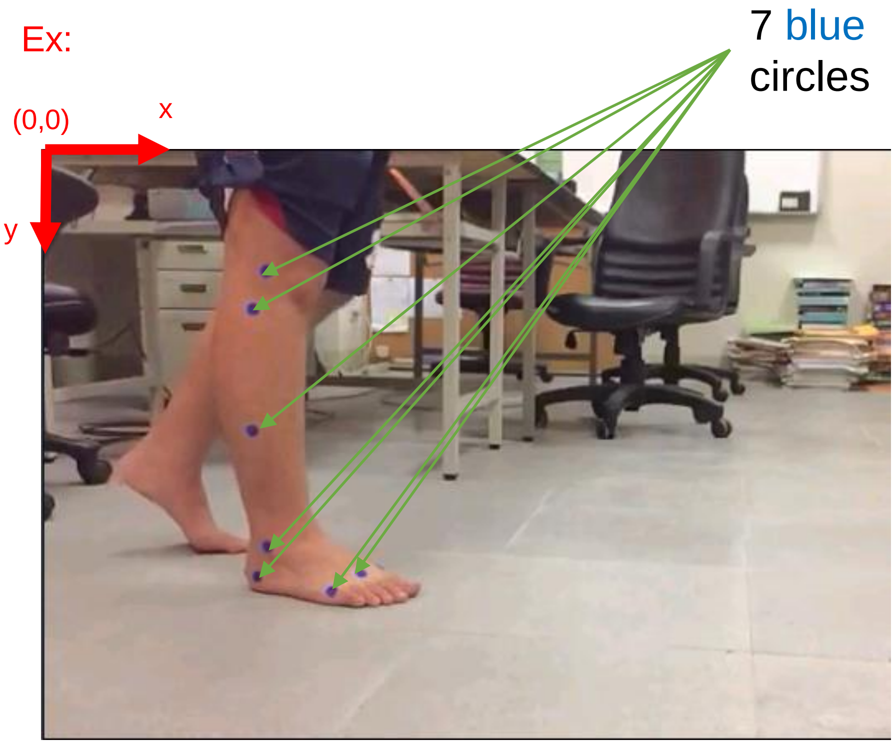
3.1 Preprocessing (10%)

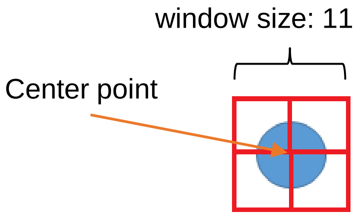
(出題：Shaku)

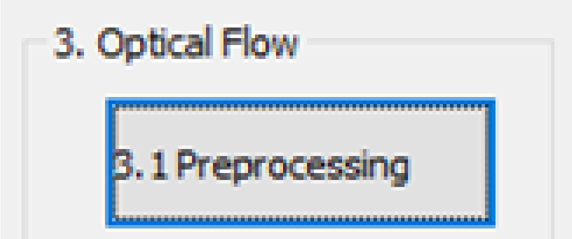
❑ Given a video: opticalFlow.mp4

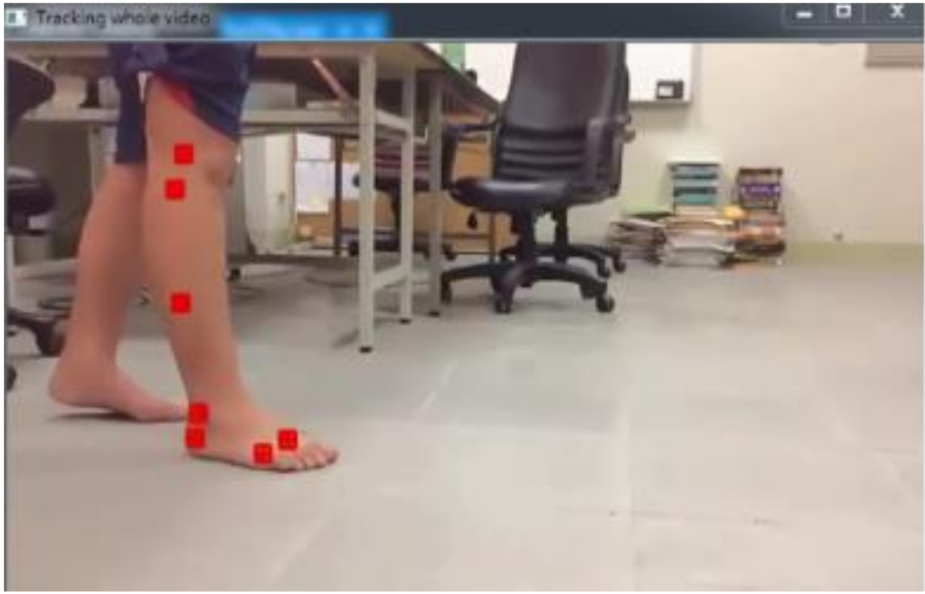
❑ Q: 1) Click the button “3.1 Preprocessing” to:

- Manually (setMouseCallback) or automatically (SimpleBlobDetector) detect 7 center points of 7 blue circles of the 1st frame.
- Show the square boundary (not red filled like the ex. image).









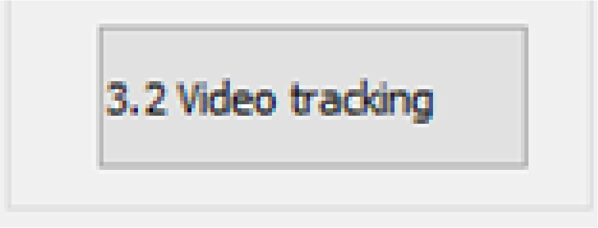
3.2 Video tracking (20%)

(出題：Shaku)

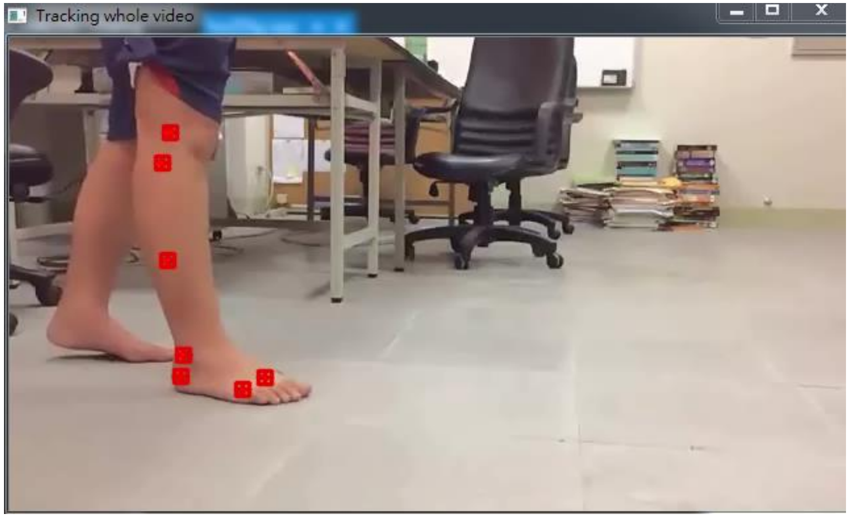
❑ Q: 2) Click button “3.2 Video tracking” to

- (10%) Track the 7 center points on the whole video using OpenCv function calcOpticalFlowPyrLK(args, winsize=21x21). Ex: https://github.com/opencv/opencv/blob/master/samples/cpp/lkdemo.cpp
- (10%) Connect 7 corresponding tracking points of all frames (the whole video) to get 7 trajectories (flow). Ex: the demo video.

❑ Hint: textbook Chapter 10 (p332 ~ p334)



Demo video:

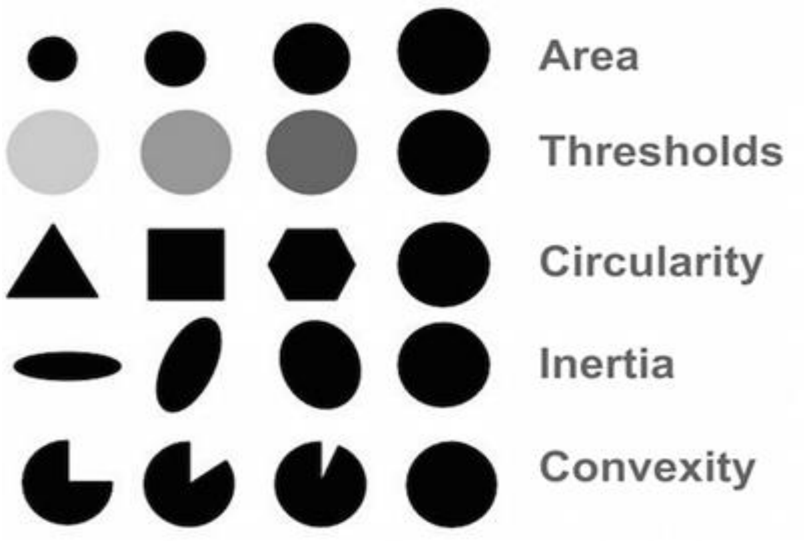


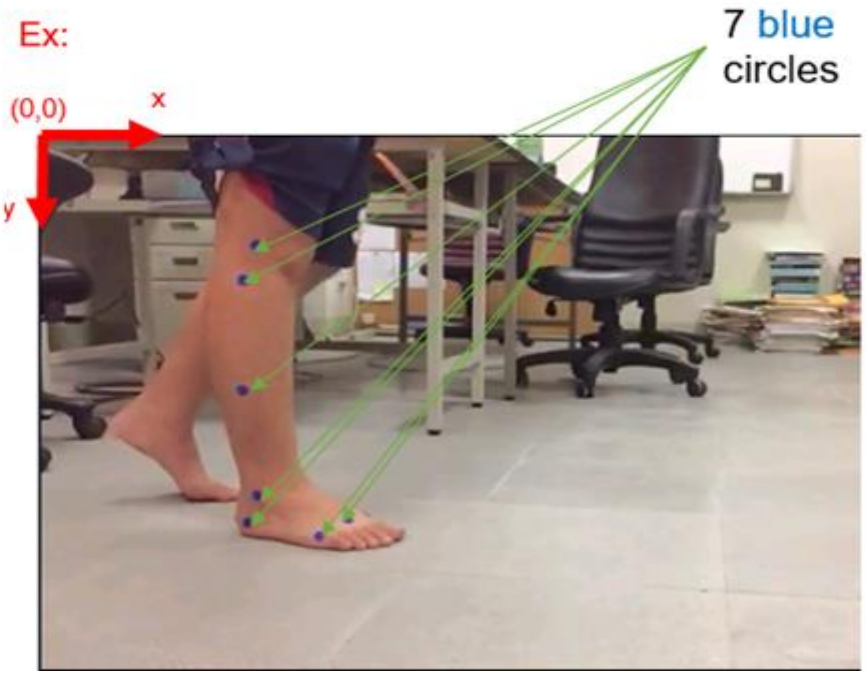
※ Tracking process may be fail, you should make the result as good as possible (at least 4~5 points success)

How to detect 7 center points of 7 blue circles

❑ You can:

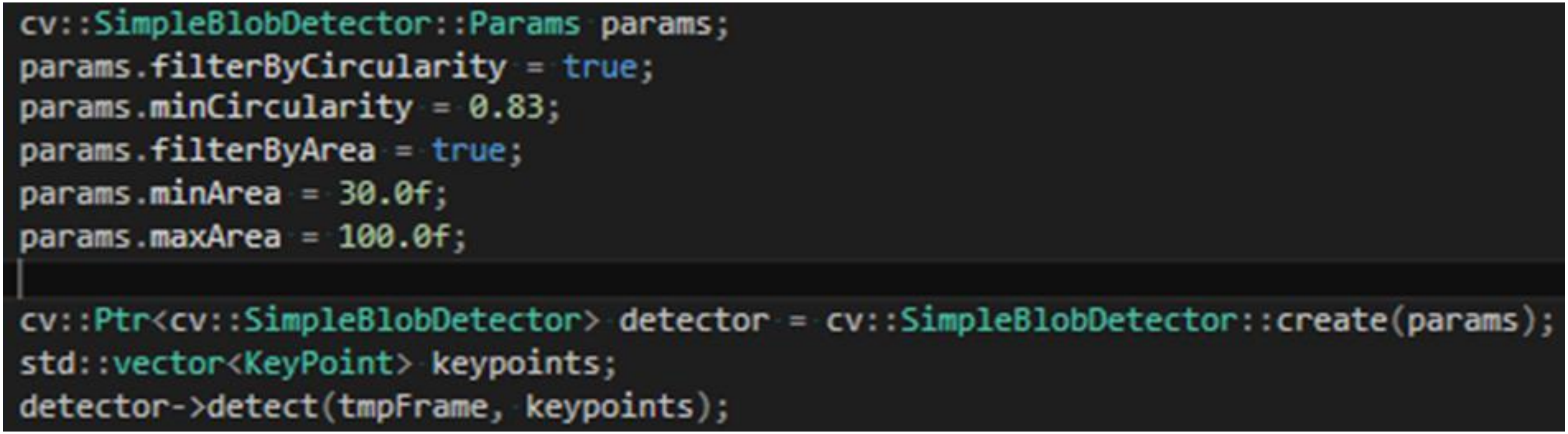
1. Using mouse callback function (Please explain how to manipulate your GUI)
2. Using SimpleBlobDetector to detect blobs (binary large object), it can filter out blobs by examining their:
   1) Area (size)
   2) Thresholds (gray level)
   3) Circularity (if perfect circle, then it is 1.0)
   4) Inertia (**ratio of the minor and major axes)**
   5) Convexity (if no gap, then it is 1.0)





SimpleBlobDetector Usage (C++ Example)

1. Setup parameters (only circularity and area examination is used)
2. Create SimpleBlobDetector with parameters
3. Detect by use detector->detect(input_images,output_keypoints)

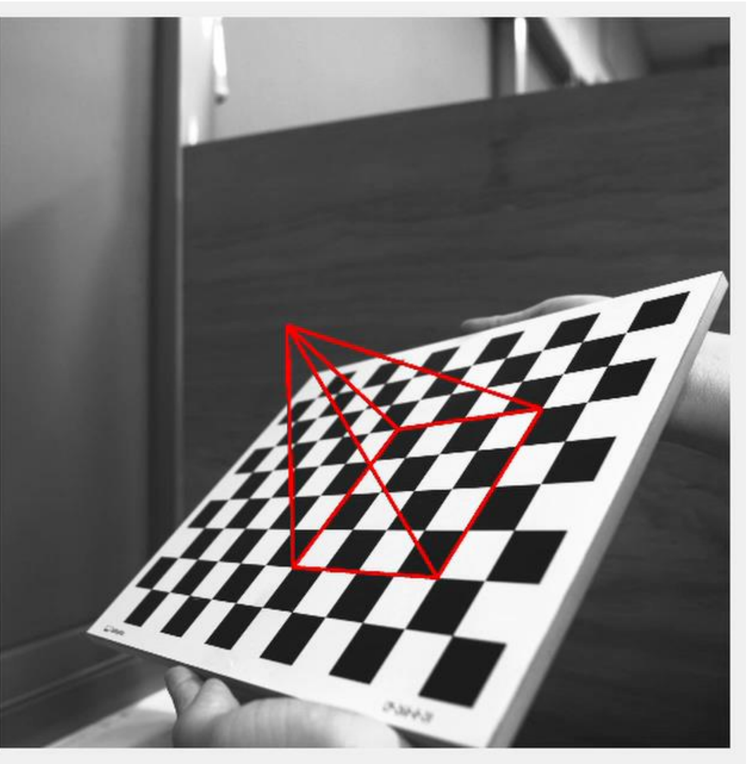


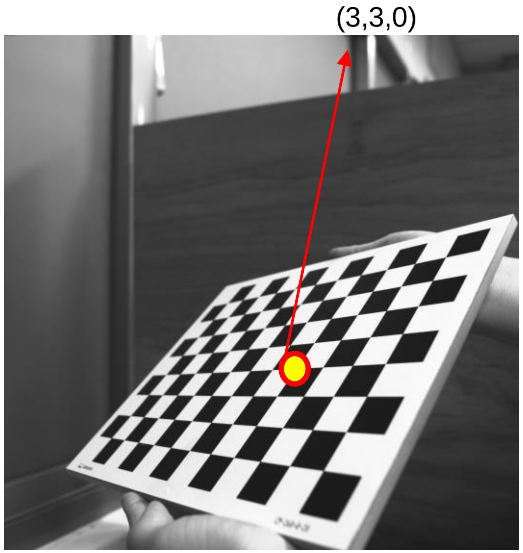
```cpp
cv::SimpleBlobDetector::Params params;
params.filterByCircularity = true;
params.minCircularity = 0.83;
params.filterByArea = true;
params.minArea = 30.0f;
params.maxArea = 100.0f;

cv::Ptr<cv::SimpleBlobDetector> detector = cv::SimpleBlobDetector::create(params);
std::vector<KeyPoint> keypoints;
detector->detect(tmpFrame, keypoints);
```

4. (20%) Augmented Reality (出題：Rex)

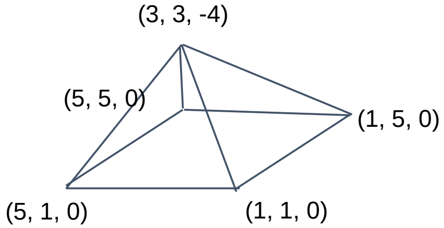
- ❑ Given: intrinsic and extrinsic parameters, distortion coefficients, 5 images: 1~5.bmp
- ❑ Q: 1) Draw a “pyramid” on the chessboards images(1.bmp to 5.bmp) then
  2) Click the button ”2” to show the pyramid on the picture. Show each picture 0.5 seconds (total 5 images)
- ❑ Hint : Textbook Chapter 11, p.387~395 Calibration
  p.405~412 Projection
  cv::projectPoints()

Demo video:

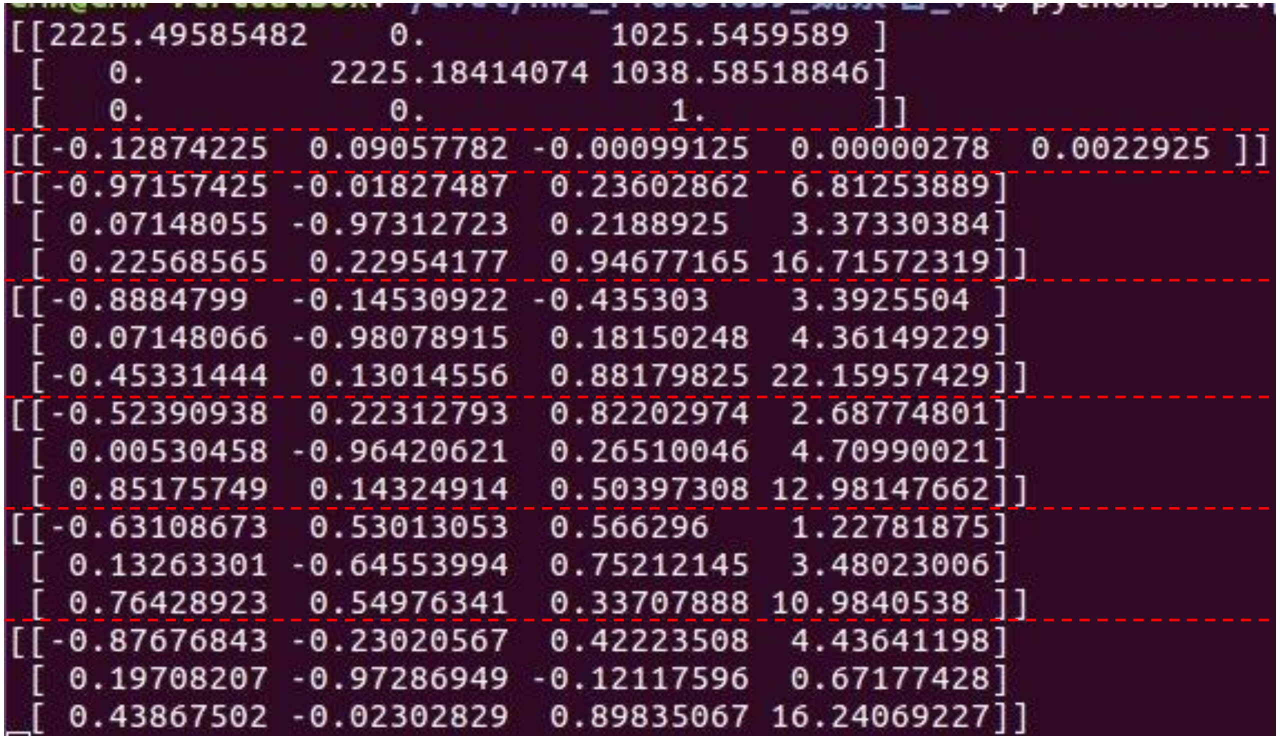




- 3D Object coordinates:
  Vertex (3, 3, -4)
  Corners(1, 1, 0)(1, 5, 0)(5, 5, 0)(5, 1,



4. Augmented Reality (出題：Rex)



```
[[2225.49585482    0.           1025.5459589 ]
 [   0.         2225.18414074 1038.58518846]
 [   0.            0.             1.        ]]
[[-0.12874225  0.09057782 -0.00099125  0.00000278  0.0022925 ]]
[[-0.97157425 -0.01827487  0.23602862  6.81253889]
 [ 0.07148055 -0.97312723  0.2188925   3.37330384]
 [ 0.22568565  0.22954177  0.94677165 16.71572319]]
[[-0.8884799  -0.14530922 -0.435303    3.3925504 ]
 [ 0.07148066 -0.98078915  0.18150248  4.36149229]
 [-0.45331444  0.13014556  0.88179825 22.15957429]]
[[-0.52390938  0.22312793  0.82202974  2.68774801]
 [ 0.00530458 -0.96420621  0.26510046  4.70990021]
 [ 0.85175749  0.14324914  0.50397308 12.98147662]]
[[-0.63108673  0.53013053  0.566296    1.22781875]
 [ 0.13263301 -0.64553994  0.75212145  3.48023006]
 [ 0.76428923  0.54976341  0.33707888 10.9840538 ]]
[[-0.87676843 -0.23020567  0.42223508  4.43641198]
 [ 0.19708207 -0.97286949 -0.12117596  0.67177428]
 [ 0.43867502 -0.02302829  0.89835067 16.24069227]]
```

Intrinsic

distortion

1.bmp extrinsic

2.bmp extrinsic

3.bmp extrinsic

4.bmp extrinsic

5.bmp extrinsic