

# Computer Vision HW3

B04507009 電機四 何吉瑞

## Code Usage:

### P1-P3

```
python main.py
```

### Bonus

```
python bonus.py --video_path [datapath of the source video] --marker_path  
[datapath of the template image] --sticker_path [datapath of the image for pasting] --  
ar_path [datapath of the output video]
```

These files are in “video.zip”, which can be downloaded from the following link:

<https://www.dropbox.com/s/0l7zxjkm4bqe9uw/video.zip?dl=0>

The execution command with default arguments:

```
python bonus.py --video_path ./video/ar_marker.mp4 --  
marker_path ./video/marker.png --sticker_path ./video/kp.png --  
ar_path ./video/target.mp4
```

### Homography Matrix:

I implement the homography matrix using solution 1 in the slides. That is, I assume that  $h_{33}$  is not zero.

### P1: Estimating Homography

Background image:



Stickers:



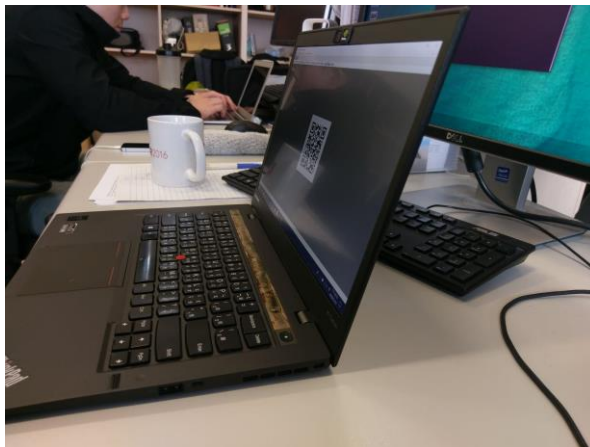


Projected result:



## **P2: Unwarp the Screen**

Source Image:



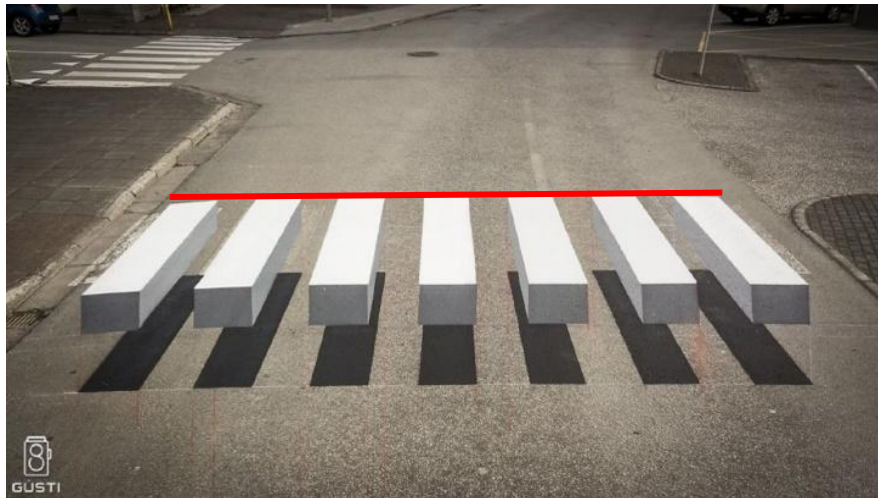
Back-warping result:



Link: <http://media.ee.ntu.edu.tw/courses/cv/18F/>

### **P3: Unwarp the 3D Illusion**

Source image:



Back-warping result:



The result is different from the ground truth. Possible reasons are as follows:

1. The top edges of textures in the source image are not totally aligned, which is shown in the red line segment. Such deviation is magnified after the transformation.
2. The 3D information may be lost in single 2D representation, and the relation between points may be distorted after the transformation.

### **Bonus: Simple AR**

Download link: <https://www.dropbox.com/s/0l7zxjkm4bqe9uw/video.zip?dl=0>

The algorithm is implemented with OpenCV, the details are illustrated as follows.

First, I use SURF to extract the feature points of the template image and each frame of source video. Then, I use the brute-force matcher with knn to find the matching between feature points in template image and each frame. From such matching, we can find a homography matrix to project the template image to each frame of the video. Therefore, after resizing the sticker to the size of template image, we can project the sticker to the video. The resulting video is the output.