ENPM 673, Robotics Perception
Project 1: Augmented Reality.
Due on: 11:59:59PM on Monday – Feb 27, 2017

Prof. Yiannis Aloimonos

1 ARTag Detection and Homography Estimation - 100Pts

ARTag is a fiducial marker system that is mainly used in Augmented Reality: ViewPoint tracking and Virtual Object interaction. Needless to say ViewPoint Estimation and Tracking is an important aspect of Robotics Perception especially in Robot Navigation and the holy 'SLAM' problem. It is therefore important to understand *What ARTag is? How it is detected?* and *What's the use of it?*

1.1 ARTag Detection - 50Pts

In this section you have to detect a custom AR tag given in the image. Here is the sample marker

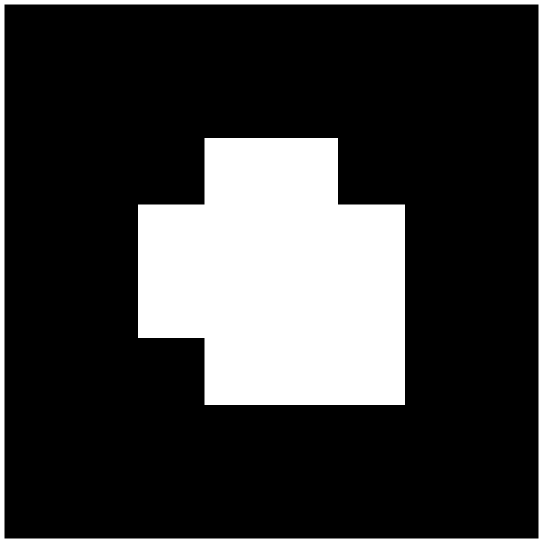
Figure 1: Sample reference tag.

The above marker is a 8×8 grid which encodes the *id* and *orientation* of the tag, see the following image

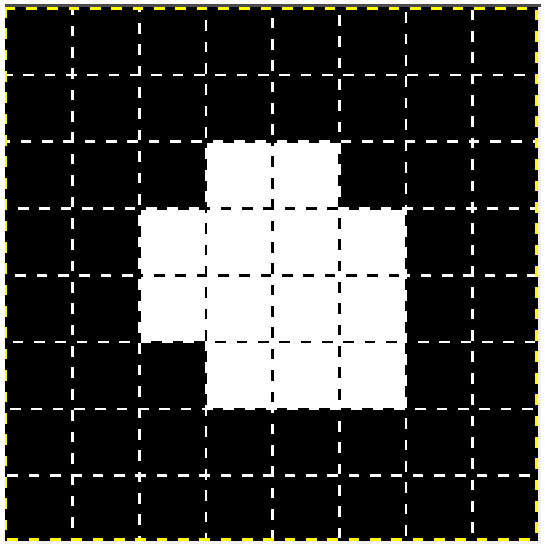
Figure 2: Grid view of reference tag.
(white represents 1 and 0 if otherwise)

Now let us understand the *Encoding Scheme of Tag*

- As you can observe in 8×8 grid there is a *padding* of width 2 all along the borders, this is to facilitate easy detection of corners when the tag is placed on a white background.
- In the inner 4×4 grid (i.e., excluding the padding in original 8×8 marker), the presence of *white* grid in *lower-right* corner indicates the upright orientation of the tag.
- Finally the innermost 2×2 grid gives the id of the marker i.e., the innermost 2×2 encodes binary representation of the id top-left grid being the least significant bit, continuing in clockwise direction to finally lower-left grid being most significant bit.

You can use any pipeline to detect the marker, BUT MAKE SURE YOU FOLLOW THE ENCODING SCHEME. Your code should return marker id and the corners in the original order (i.e., for upright orientation of the tag) which will serve as input for *Homography Estimation*. **You might have to use homography estimation for recovering corners and marker id, refer to supplementary document for more details on homography**

1.2 Homography Estimation and Virtual Cube

1.2.1 Homography Estimation – 20Pts

You need to estimate the homography between the template (`Lena.png`) image (corners of template) and tag in the input video frames. Now transform the template onto the input frame so that it replaces the tag. The orientation of the transformed template should be intact with the orientation of tag.

1.2.2 Place Virtual Cube – 30Pts

In the context of augmented reality we will find the homography as the transformation that projects a square marker board from the world coordinate system into a polygon in image plane, which we further use to project a virtual cube on to the tag.
The main steps are as follows:

- Compute the Homography between world coordinates (corners of ARTag in reference marker image) and the image plane (corners of ARTag in input frame).
- Build Projection matrix from the camera calibration matrix and Homography matrix (more details in supplementary PDF).
- Assume that the virtual cube is lying on top of the marker (in input image, which is negative Z direction), you now can get the other four corners of cube. Now all the corners of cube can be transformed to image plane using projection matrix.

Refer to the supplement document to understand how to recover projection matrix from homography matrix.

PROJECT/SUPPLEMENTARY PDF, DATA IS AVAILABLE AT ENPM 673.

2 Extra Credit

Detect multiple markers and project virtual cubes on each of them. – **40Pts**

3 Submission Guidelines

Your submission **SHOULD be a ZIP folder** (no other file extensions) with the naming convention `YourDirectoryID_proj1.zip` on to ELMS/Canvas. Additionally follow the guidelines given below:

1. You will have `AR` sub-directory under `P1_submission` directory which will be uploaded as `.zip` file.
2. Submit your codes (`.m` files) in `Scripts` sub-directory (under `AR` directory) and have a `runMe.m` which when run should generate outputs for all sections.
3. All input images i.e., selected frames for each tag should be in `Data/Tag%d` sub-directory as say `Data/Tag1` for Tag1 frames (under `AR` directory).
4. Output should be dumped to `Output` sub-directory (under `AR` directory) and for ANY `Tag%d` directory in the data directory, when the main script is run there should be a sub-directory `Tag%d` created under `AR/Output` directory (for `multipleTags` directory, the corresponding output directory should be `multipleTags` under `AR/Output` directory).

   In each of `Tag%d` directory, there should be

- `detected.jpg` showing 4 different corners of the marker with 4 different colors (color convention is R, G, B, Y for top-left, top-right, bottom-right and bottom-left corners respectively for the UPRIGHT orientation of marker) – For 1.1.
- A video `homography.mp4` (of atleast 4 seconds with 30fps) where marker in the corresponding `Tag%d.mp4` should be replaced by 'template' image (`Lena.png`) – For 1.2.1
- A video `virtual.mp4` (of atleast 4 seconds with 30fps) where a virtual cube needs to be projected onto the marker and id should be highlighted as text (say `id=7`) on top of marker – For 1.2.2
- Submit `P1_submission/Reports/AR.pdf` detailing the pipeline used for ArTag detection (for extracting the corners and finding out the id).

We have a total of 5 tags, you are given only 3 videos and grading is done on the other two. MAKE SURE THE GUIDELINES ARE FOLLOWED so that the code runs fine for *grading data* and if your code does not comply with the above guidelines, you'll be given **ZERO** credit.

4 Collaboration Policy

You are allowed to discuss the ideas with fellow students, but you need give credits in the report. But the code you turn-in should be your own and if you **DO USE** (try not to and it is not permitted) other external codes - do cite them and you might get partial credits. For other honor code refer to the University of Maryland Honor Pledge.