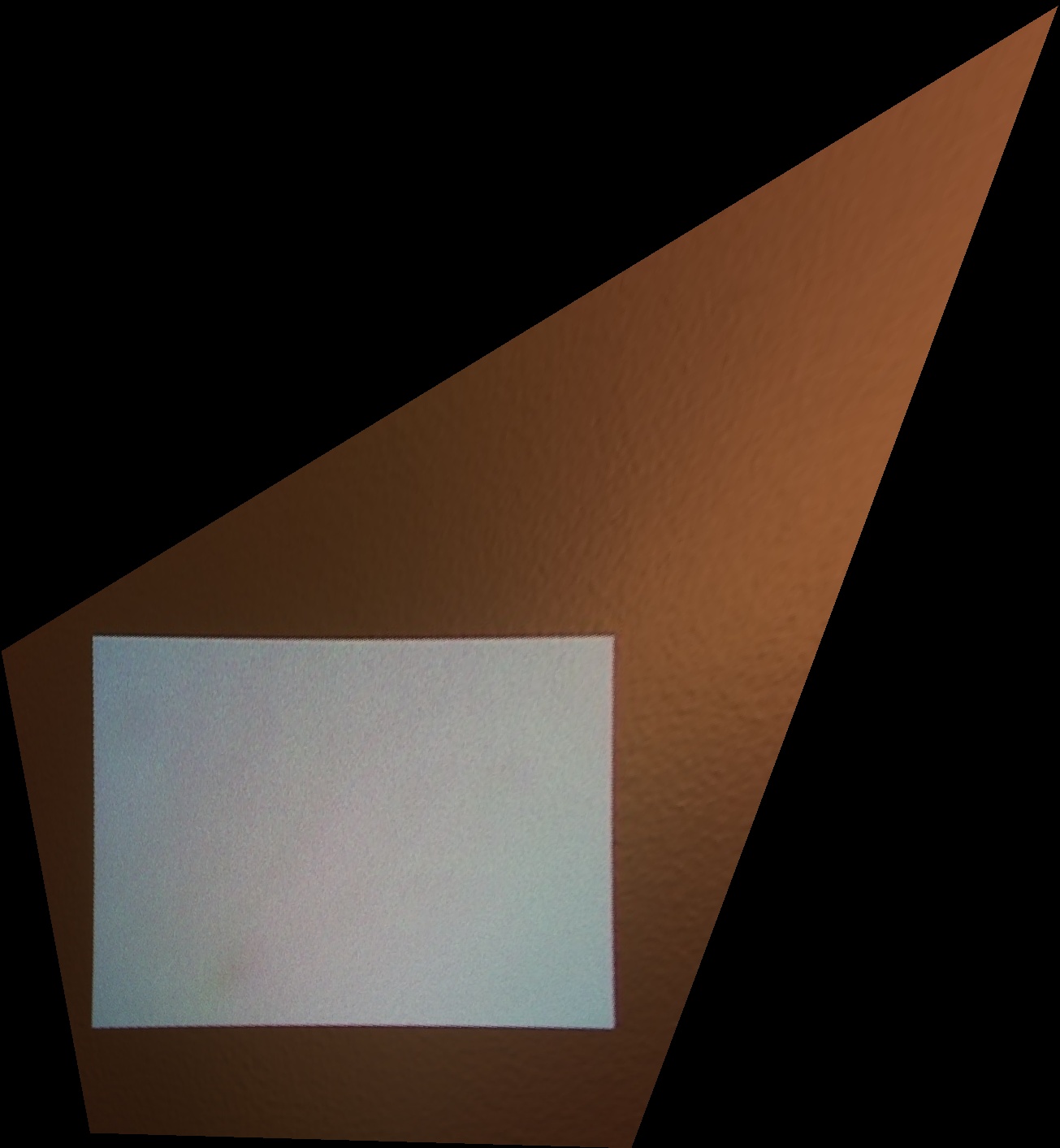
This week I’ve been working on the Plane Anamorphosis program.

My last week’s implementation of H didn’t work (possibly because numpy array dimensions are height x width, and not width x height), so I replaced it with opencv’s cv2.findHomography. I verified that the and are correct by checking if they map corners correctly between the camera and projector image.

I implemented backward and forward warping that generate an anamorphic image. These were not working at the beginning because of the numpy height x width issue. After correcting the warping algorithm, I tested it on the initial camera image, and got the expected correctly warped image ------>

I then tested the program on the actual image that I want to warp (a white rectangle), but when projected on the wall, it was still not correct. This is most probably because I scale the generated anamorphic image up/down when I do “Full screen” view to show the entire image in the projector screen.

-> *This still needs to be fixed perhaps by using some full screen feature in opencv.*

Next part of this project: Multiplanar Anamorphs

Reference: Ashdown M., *A Flexible Projector-Camera System for Multi-Planar Displays*

I don’t have questions about the paper so far, at least not until I actually start to code.

I have other questions though about the next goals of this project:

1. Given viewers: how do I find an optimal anamorphic transformation?
   1. Find for each viewer and somehow average them?
   2. I suppose I need to minimize something to get the optimal transformation?
2. What are the next surfaces I should generate anamorphosis for? Up to any random surface?
   1. Planar
   2. Multiplanar
   3. Half-cylindrical (i.e. curved in one direction) concave/convex
   4. Spherical (curved in 2 directions??) concave/convex
   5. Irregular

<https://mathspig.wordpress.com/category/topics/co-ordinates/>

<http://docs.opencv.org/trunk/doc/py_tutorials/py_core/py_basic_ops/py_basic_ops.html>

Sometimes, you will have to play with certain region of images. For eye detection in images, first face detection is done all over the image and when face is obtained, we select the face region alone and search for eyes inside it instead of searching whole image. It improves accuracy (because eyes are always on faces :D ) and performance (because we search for a small area)