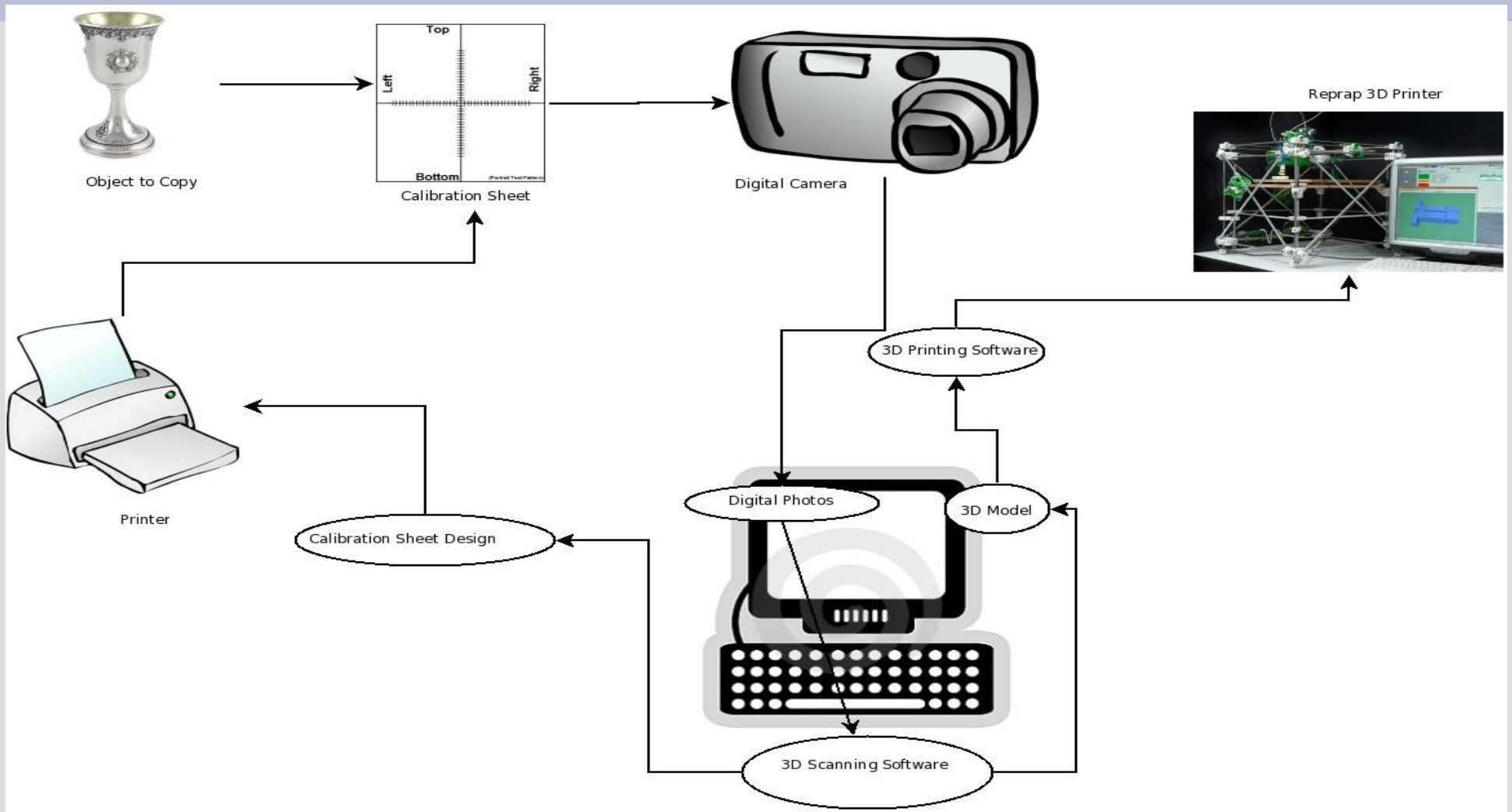
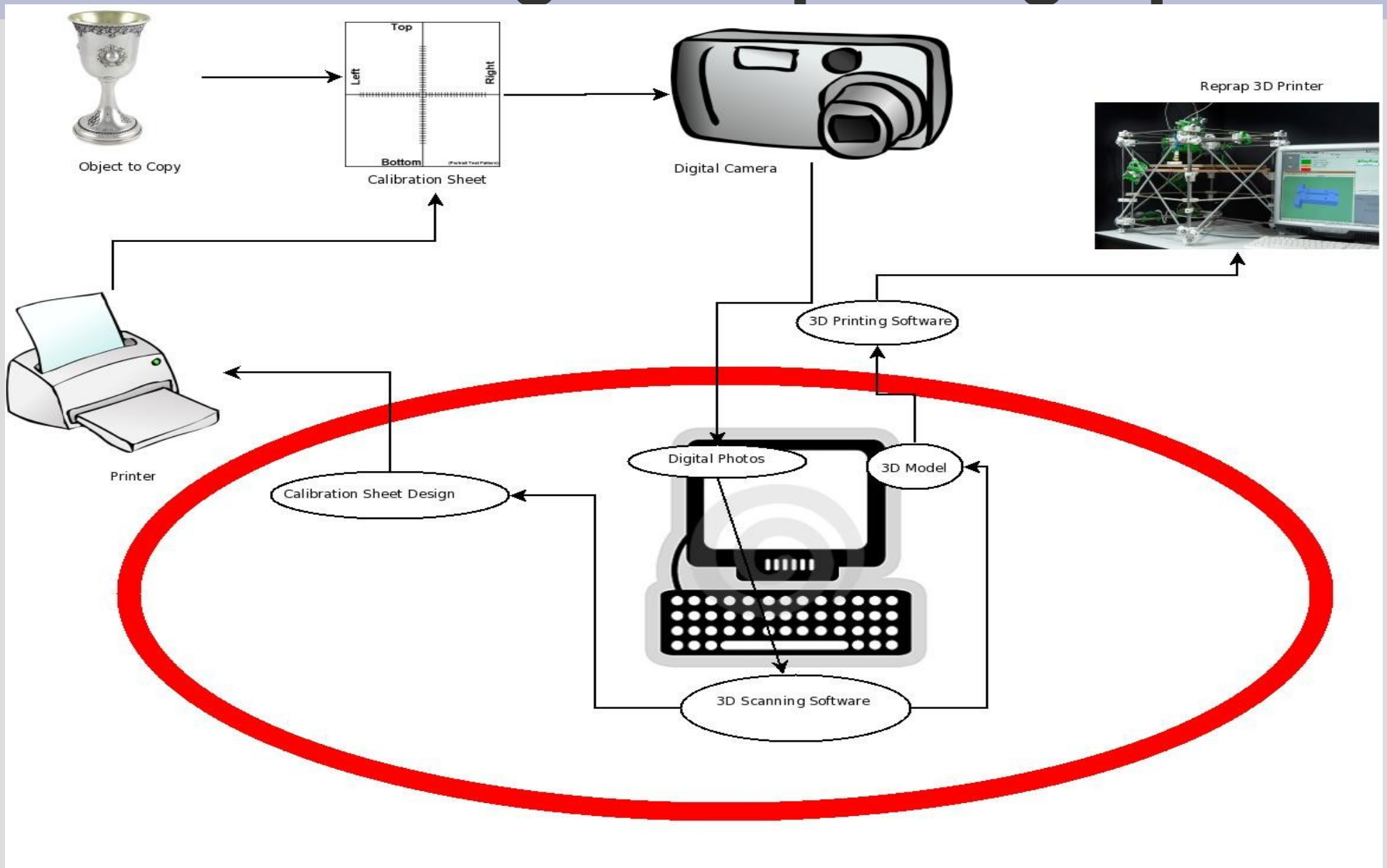
Proof of concept 3D photocopier



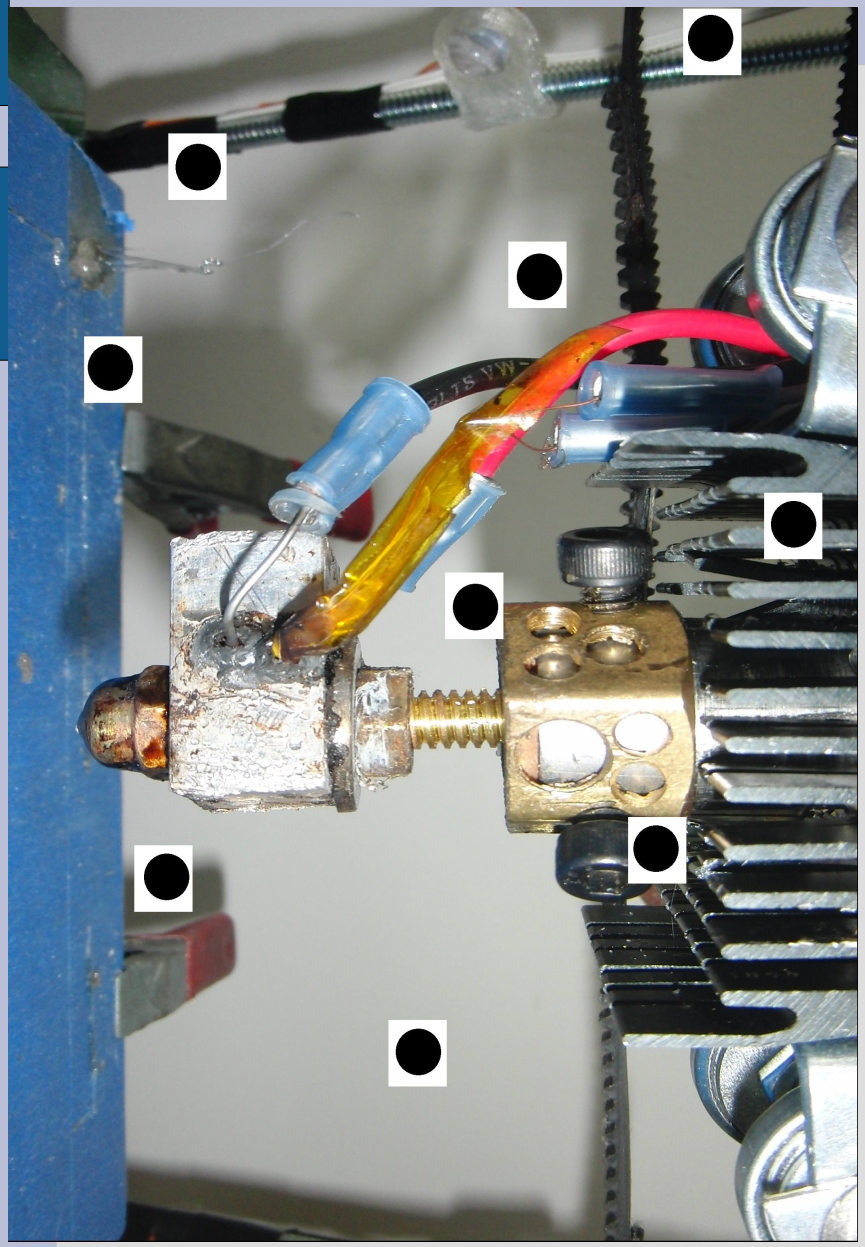
My part of the problem
3D scanning from photographs

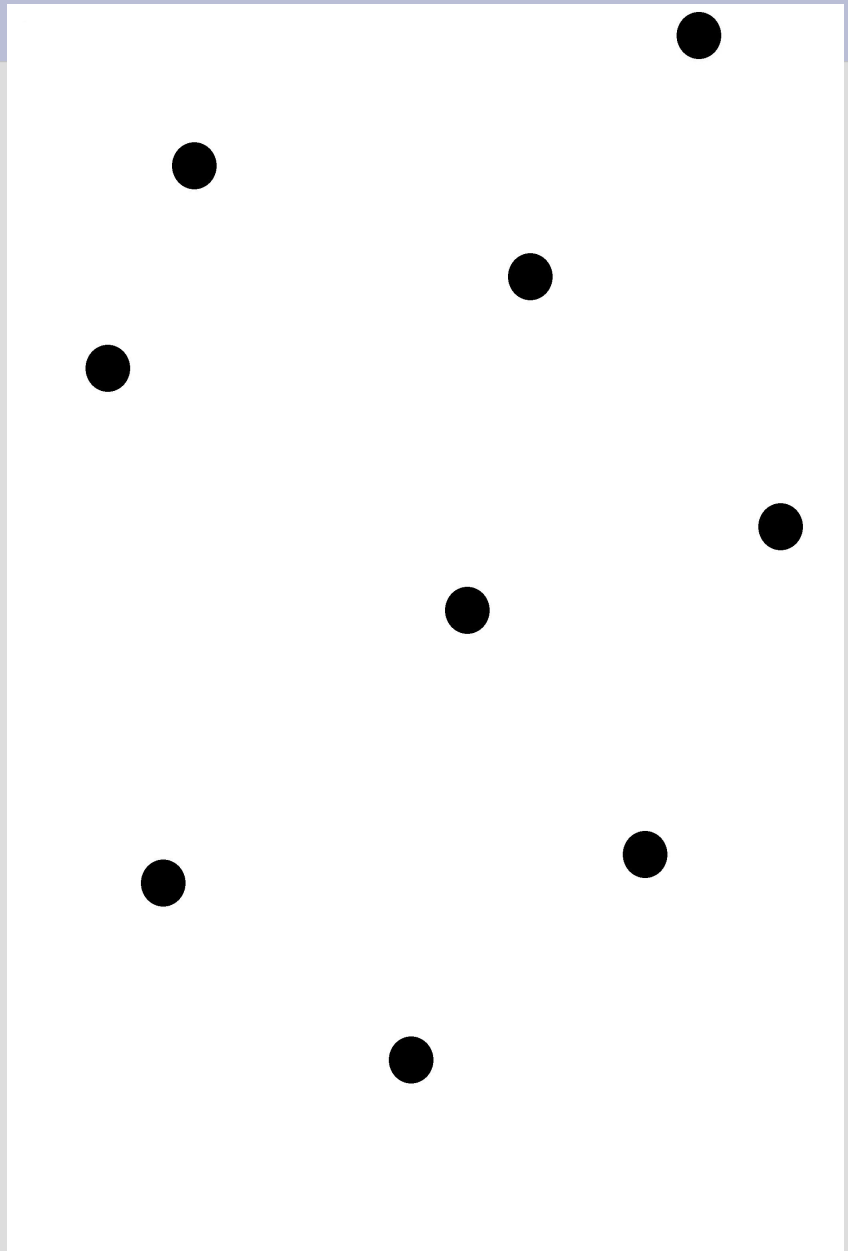


Outline of 3d scanning from photographs (in O(n) time)

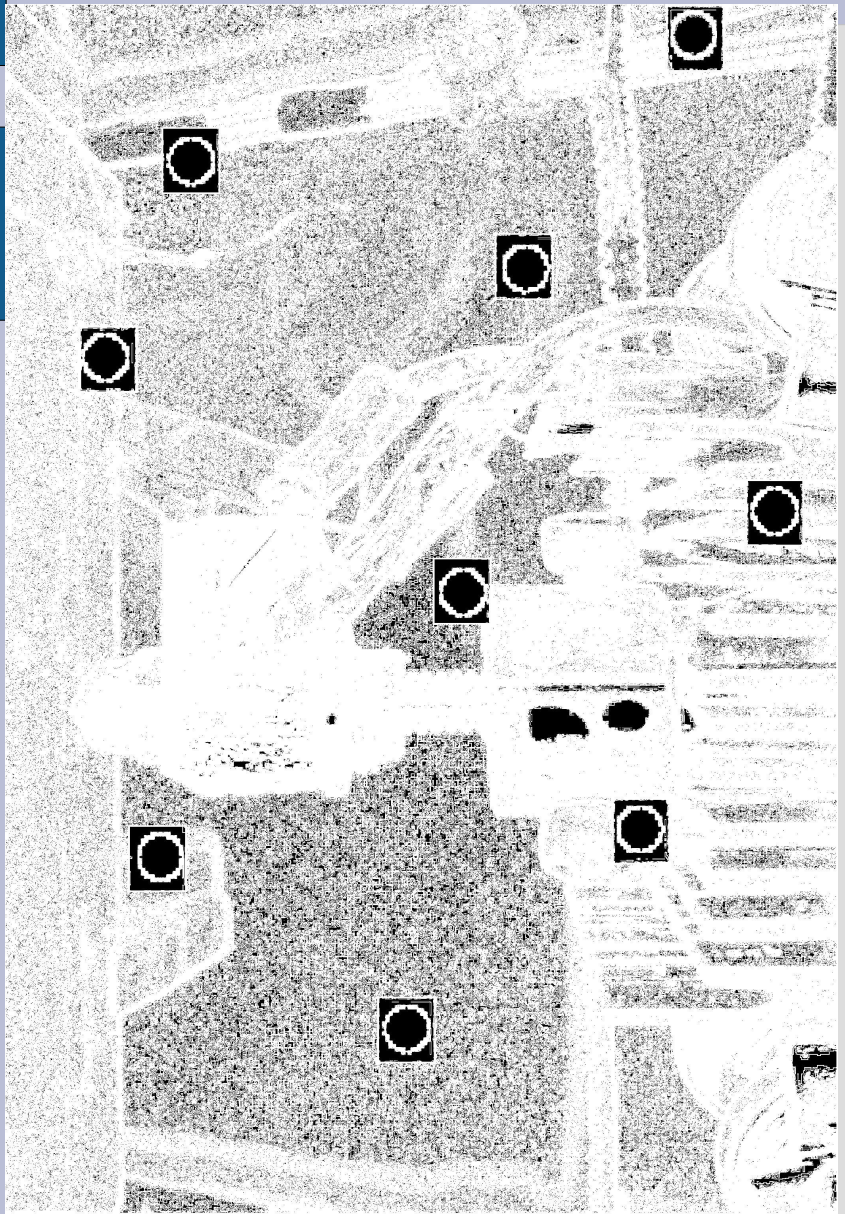
1) Calibration Sheet Interrogation
2) Find Ellipses in image
3) Point Pair Matching
4) Estimation of Camera Parameters
5) Undo Lens Distortion
6) Image Segmentation
7) Find Voxelised Visual Hull
8) Optionally output Voxel STL file
9) Restrict Image Space
10) Surface Texture Matching
11) Output STL file

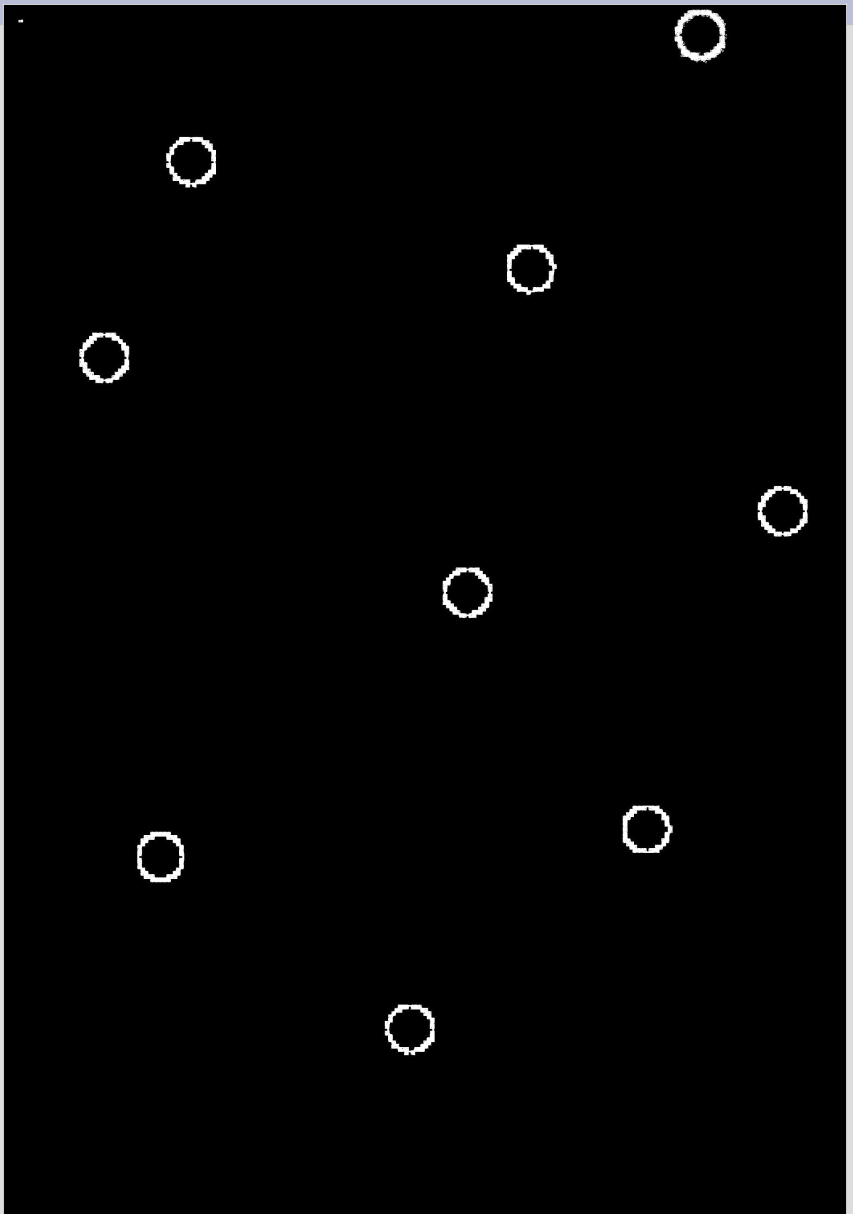
Calibration Sheet Interrogation



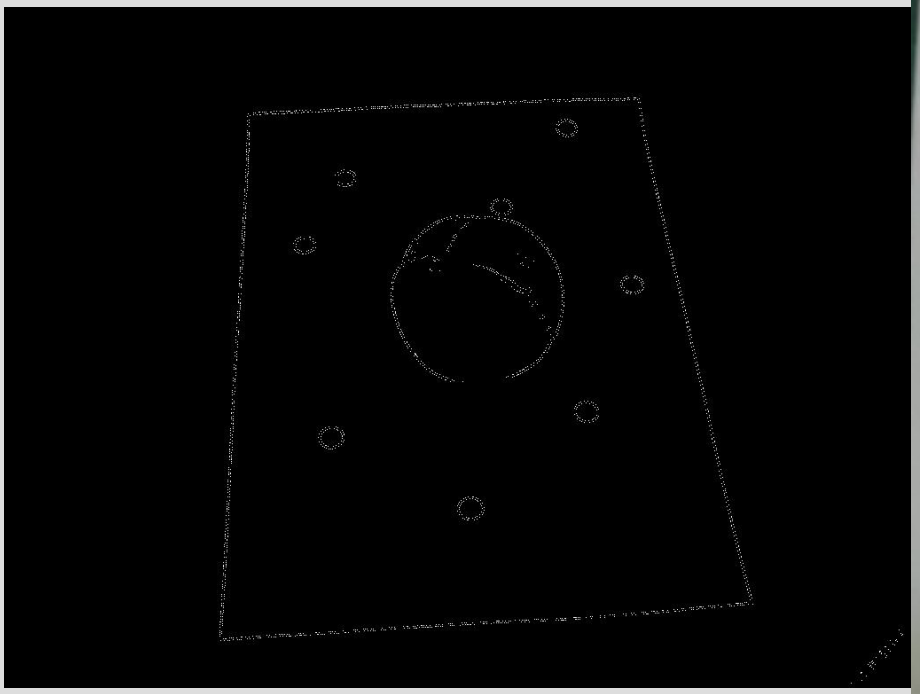


Calibration Sheet Interrogation

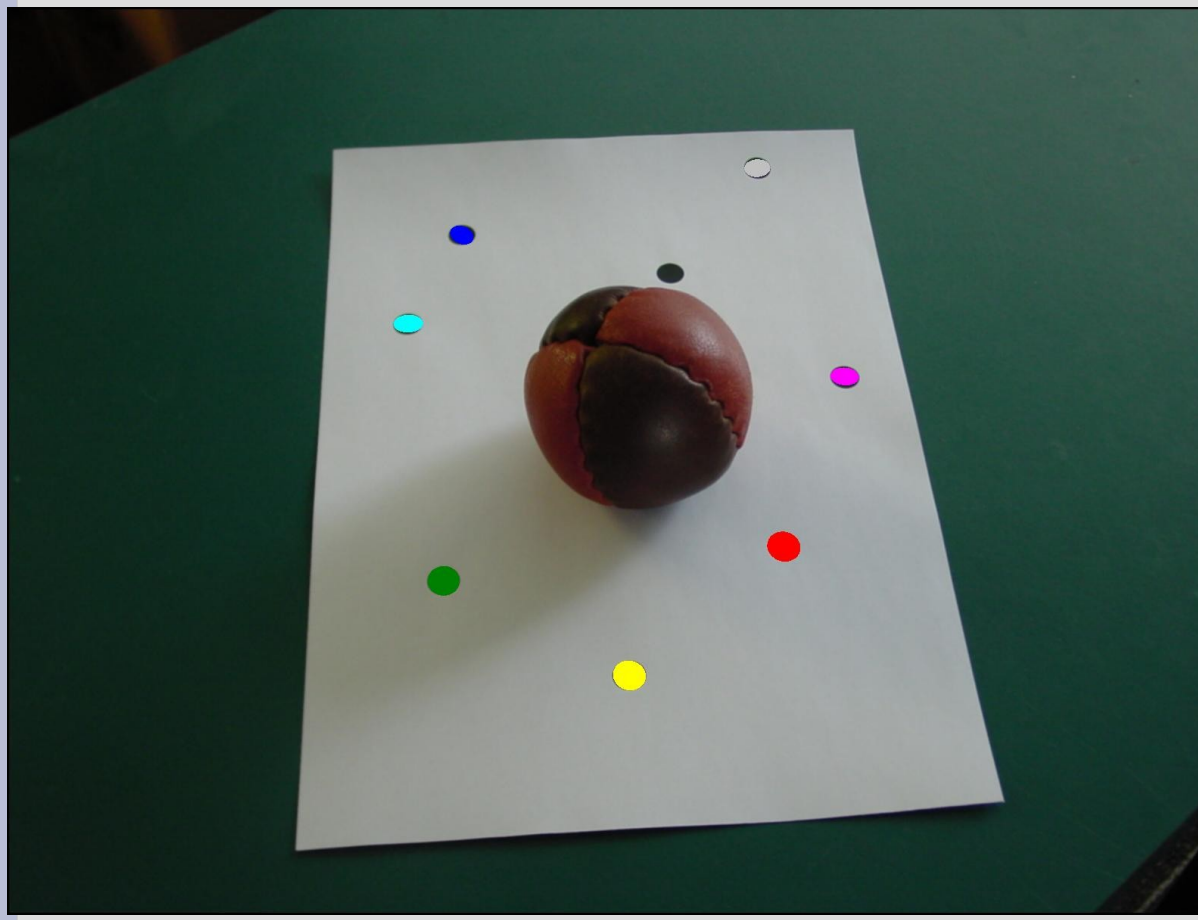


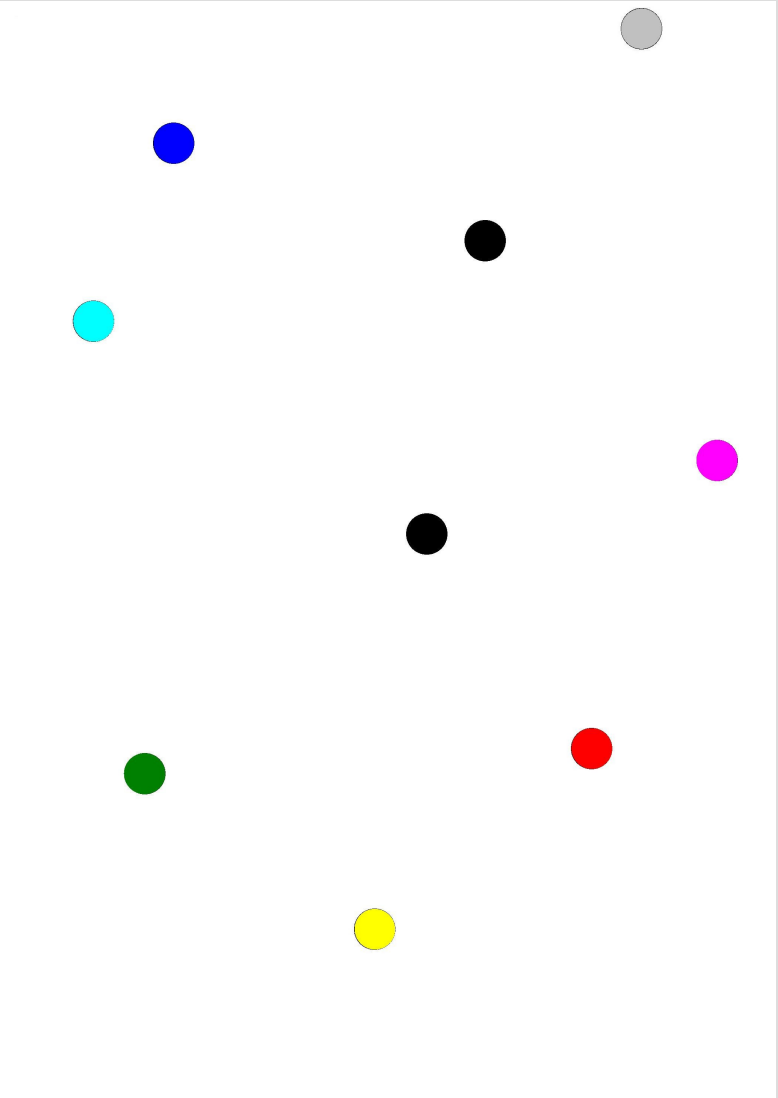


Find Ellipses in Image



Point Pair Matching





Estimation of Camera Parameters

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = d_u\,(u', v')$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}, P = K[R|t]Z, K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

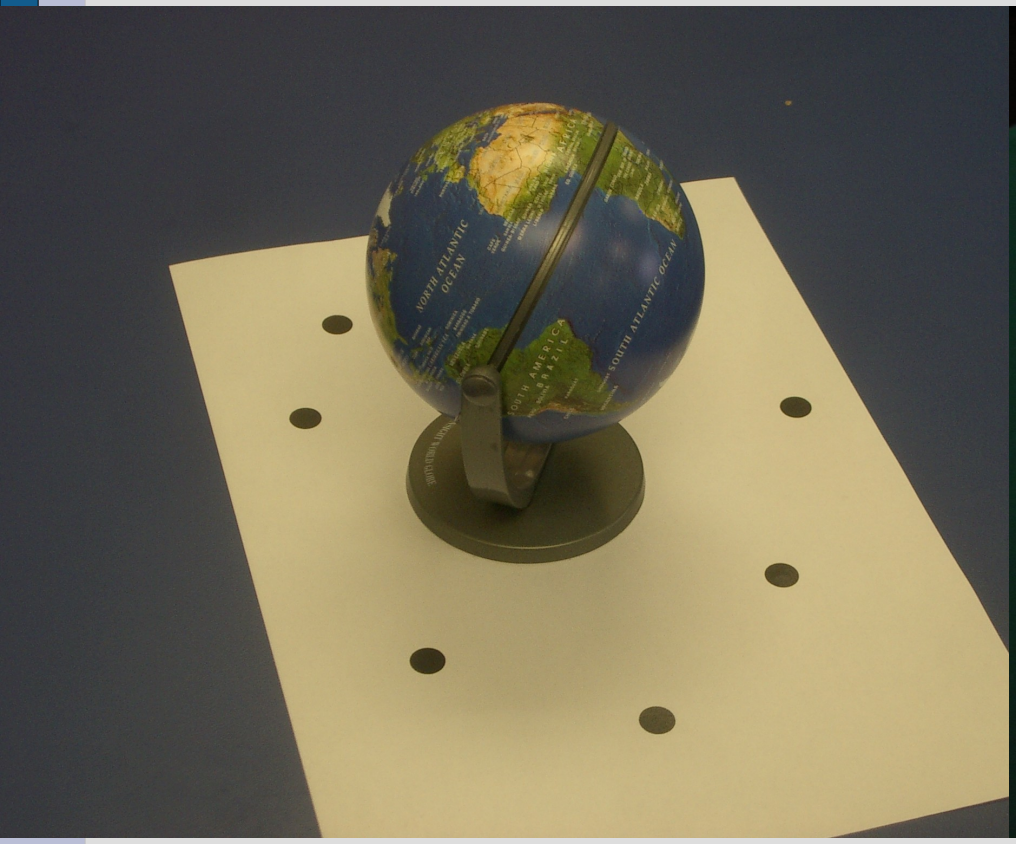
$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad P = \begin{bmatrix} h_{11} & h_{12} & ? & h_{13} \\ h_{21} & h_{22} & ? & h_{23} \\ h_{31} & h_{32} & ? & h_{33} \end{bmatrix}$$

Undo Lens Distortion

One way function: can go from distorted to undistorted pixels but not the other way

Need to undistort all pixels and then interpolate to get pixelised undistorted image

Image Segmentation



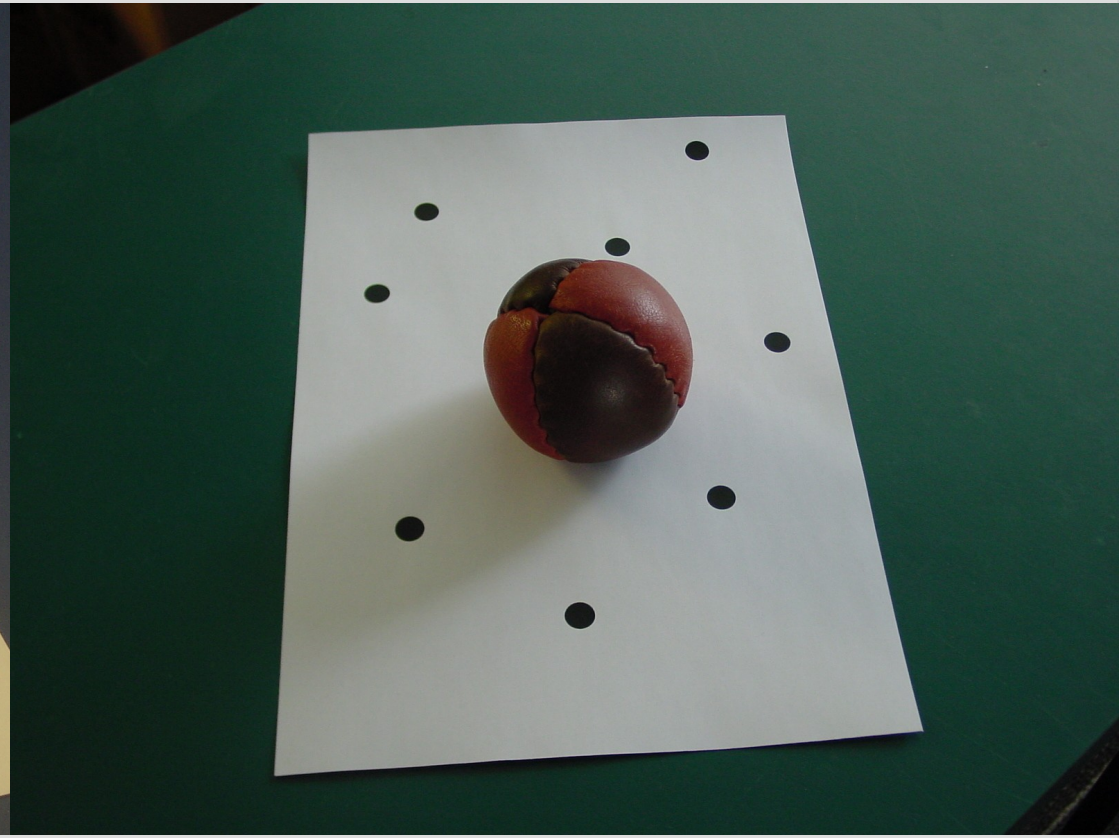
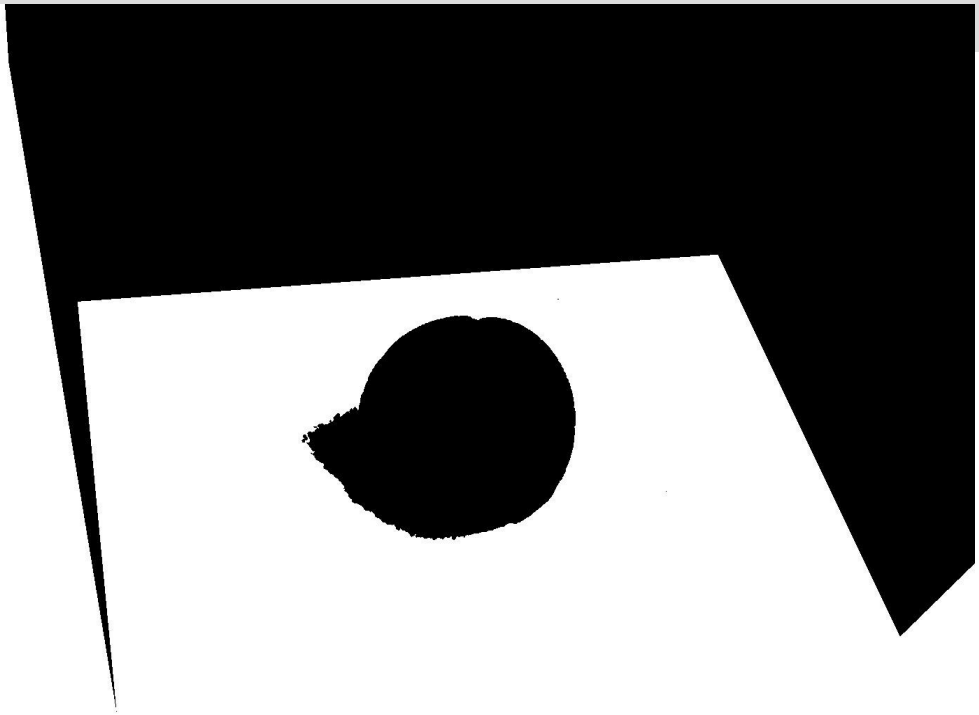
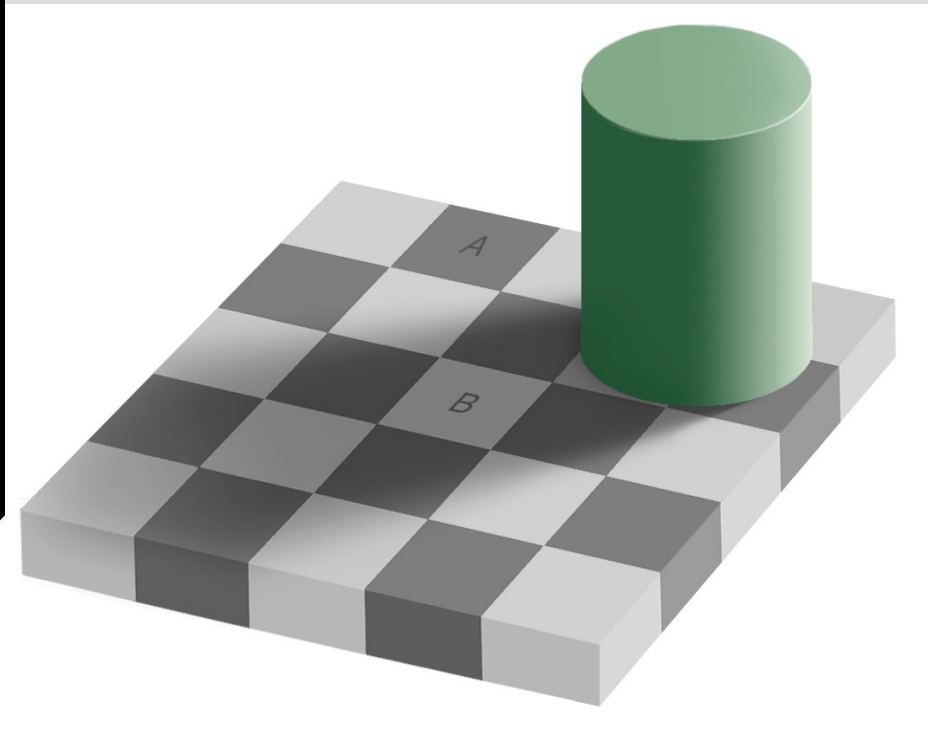


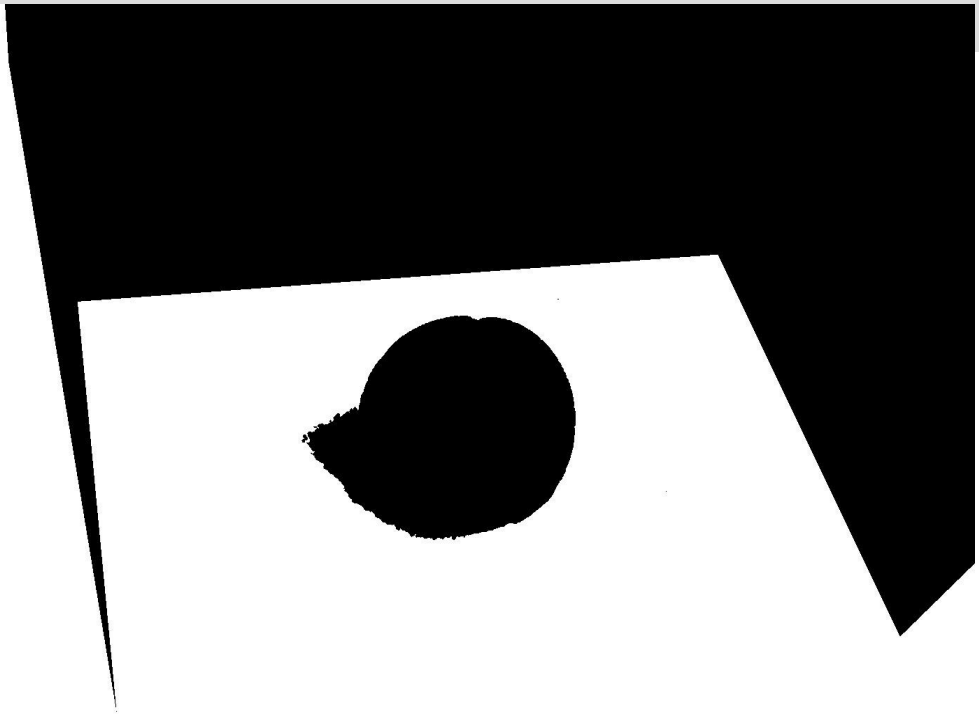
Image Segmentation

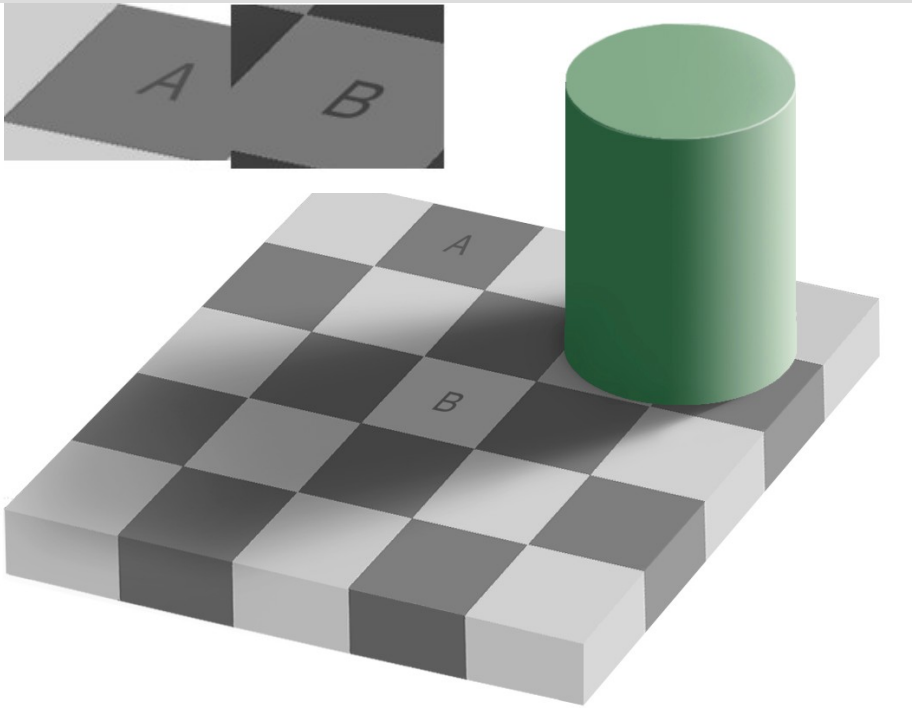




Adelson checkerboard shadow illusion Image from wikipedia

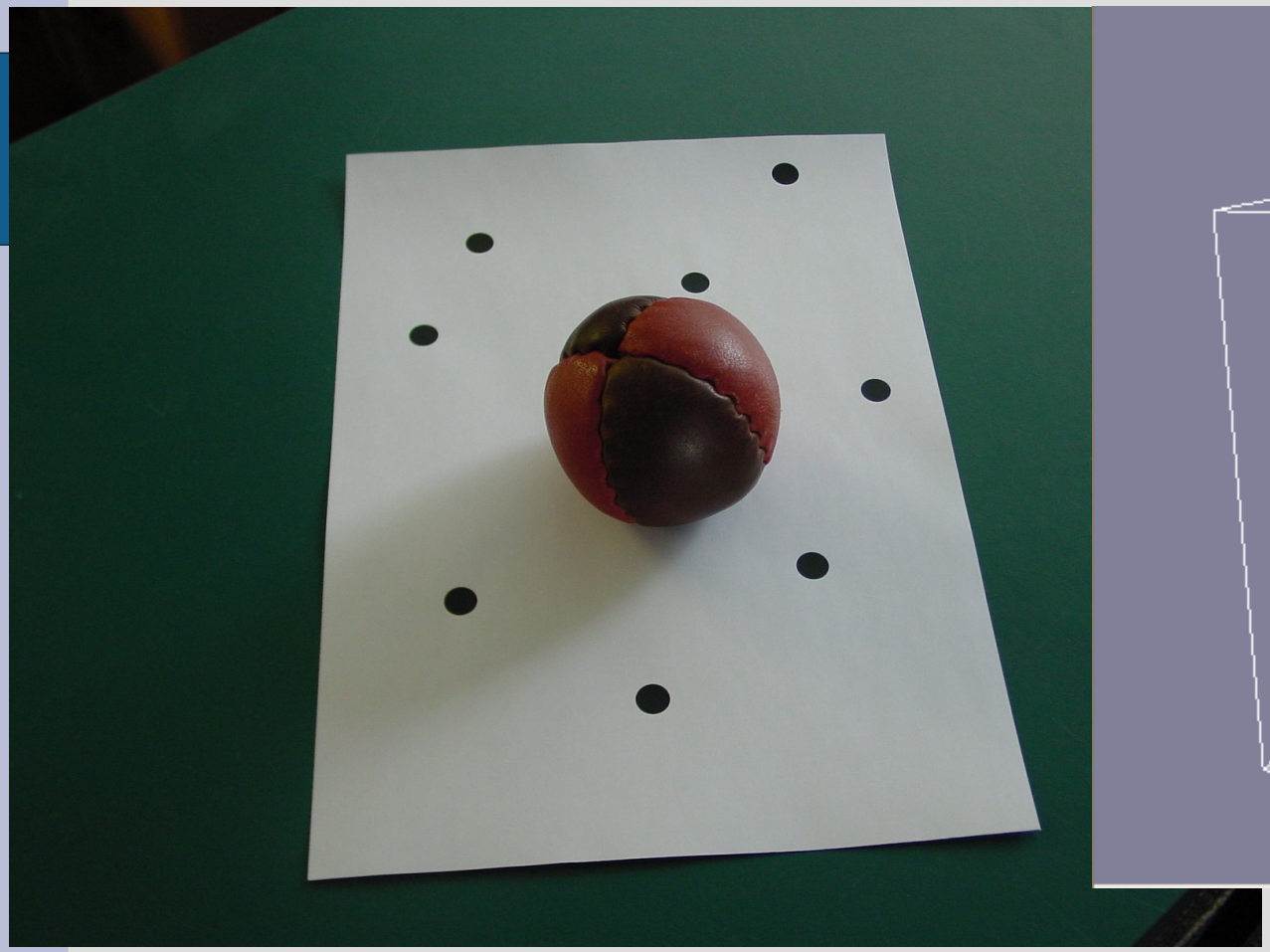
Image Segmentation

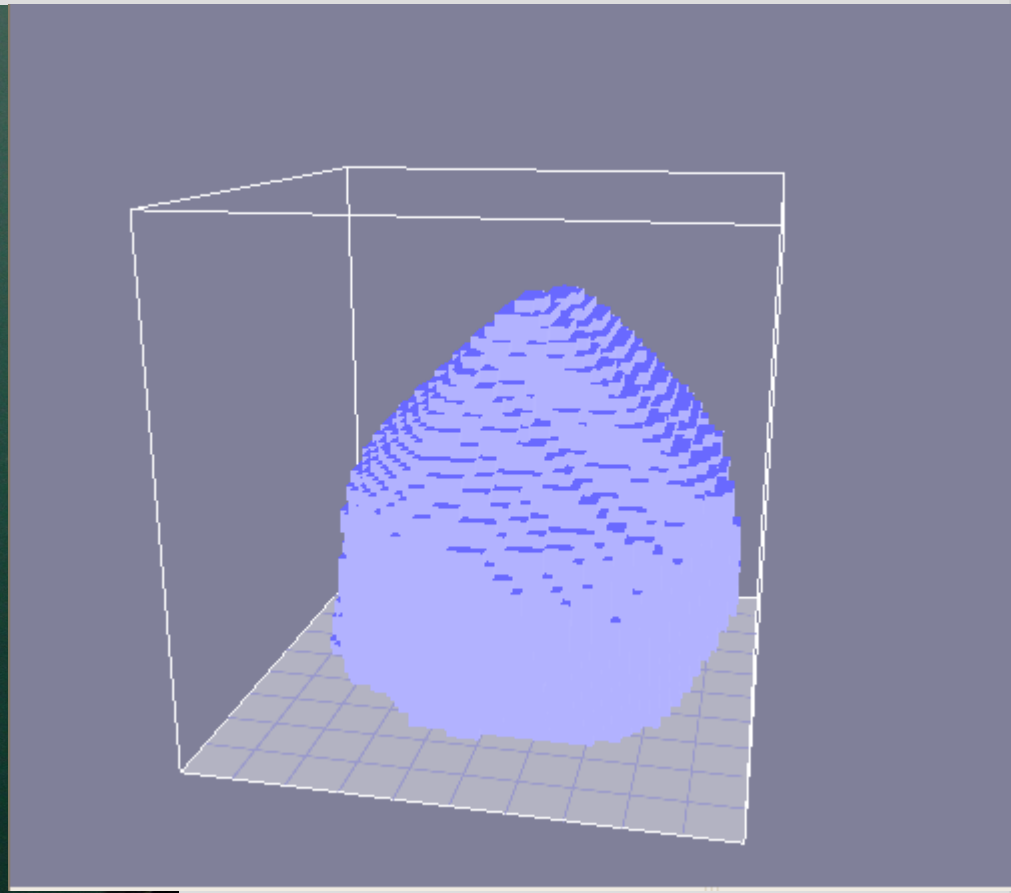




Adelson checkerboard shadow illusion Image from wikipedia

Find Voxelised Visual Hull





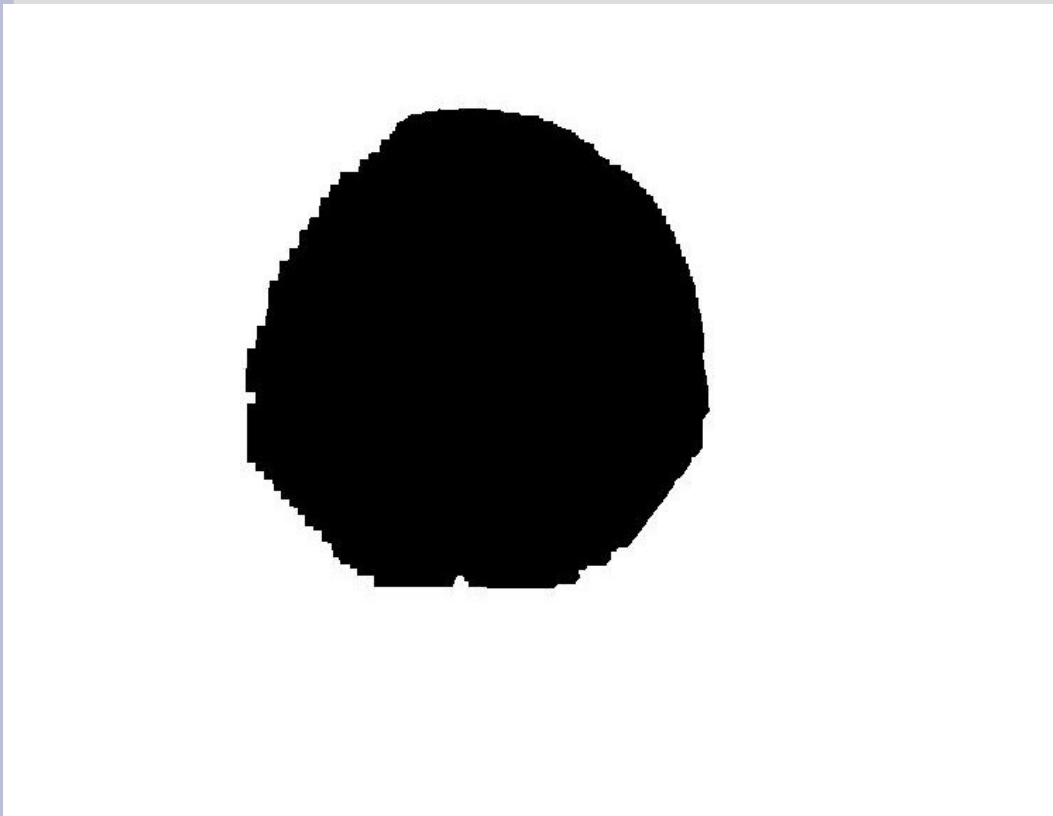
(Optionally output Voxel STL file)

Restrict Image Search Space

Used so don't need the voxel data structure for further processing

Backproject the voxels into each image and set boolean 2d array appropriately

Only those pixels corresponding to rays that intersect voxelised visual hull need to be further processed.



Surface Texture Matching

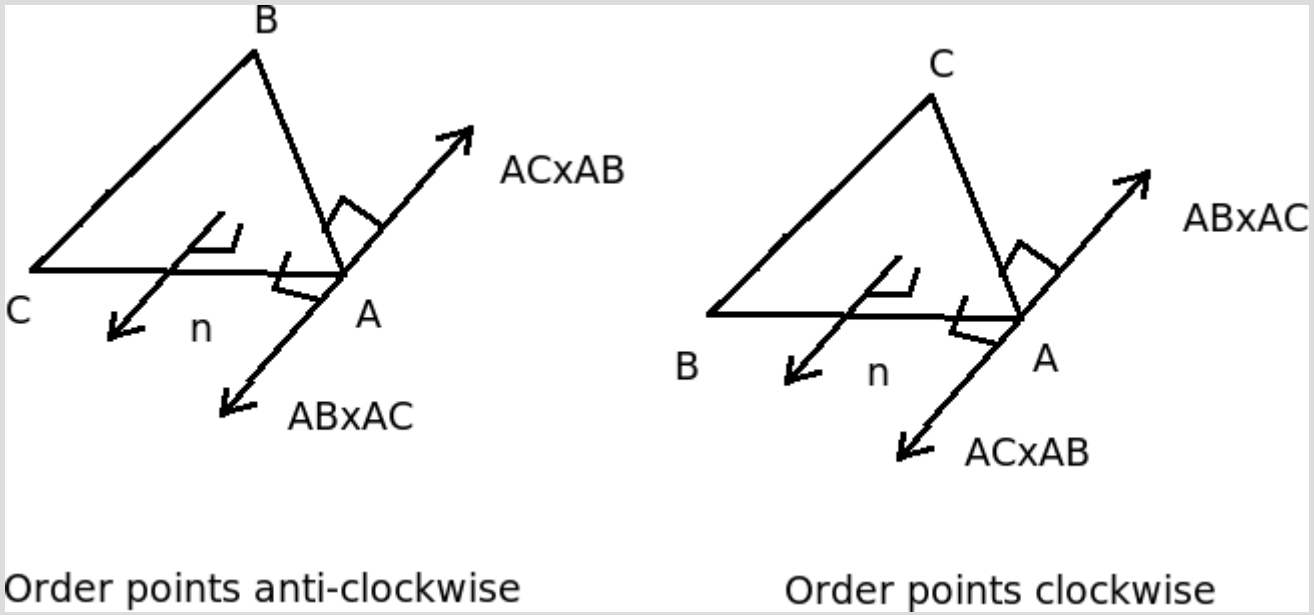
From a known good line segment find the third point that produces a triangle that has the best similarity match score.

Use the two new line segments as seed points for new triangles and continue.

Note that relative best not absolute correct match means we can handle camera estimates being out by a little.

Output STL file

File format contains redundant information



Supplementary Section – The Kinect

How to get around these problems if you are Microsoft

Control the hardware involved and use the right magic numbers to give the user a 2.5D image i.e. An image where each pixel (x,y) has a depth range number (0-255) associated with it.

Disclaimer: I have not spent a lot of time looking into this and most of that time was reading fluff pieces.

What a Kinect looks like

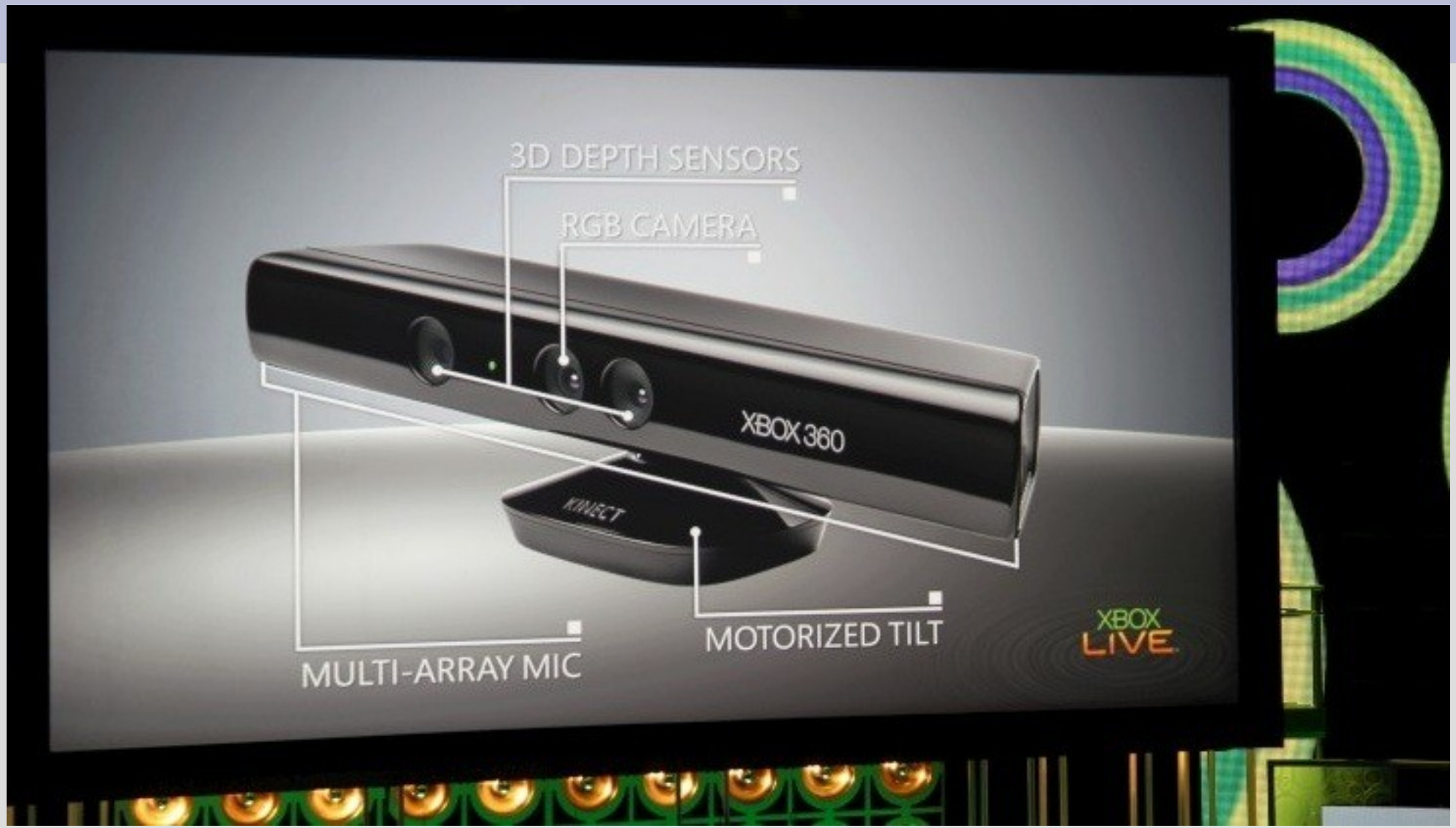


Image from wikipedia (http://en.wikipedia.org/wiki/File:KinectTechnologiesE3.jpg)
of a slide shown at Microsoft's E3 2010 slideshow

How does the Kinect get depth data?

IR laser and IR camera are a known distance apart and have known internal parameters that don't change.

IR laser projects a known pattern of dots that the camera can pick up.

IR laser projection cone doesn't intersect camera view cone until approx 0.6-0.7m and light is too faint after approx 10m to give the effective range.

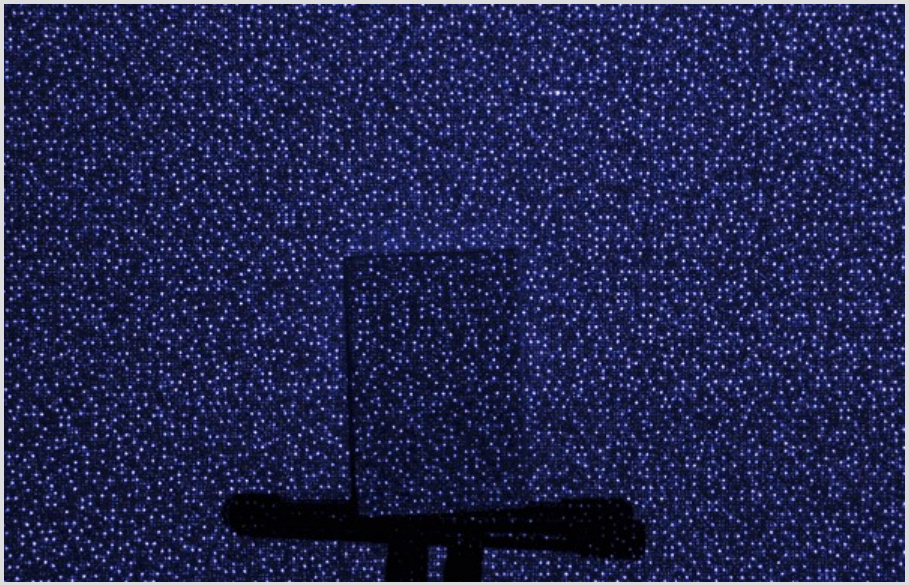
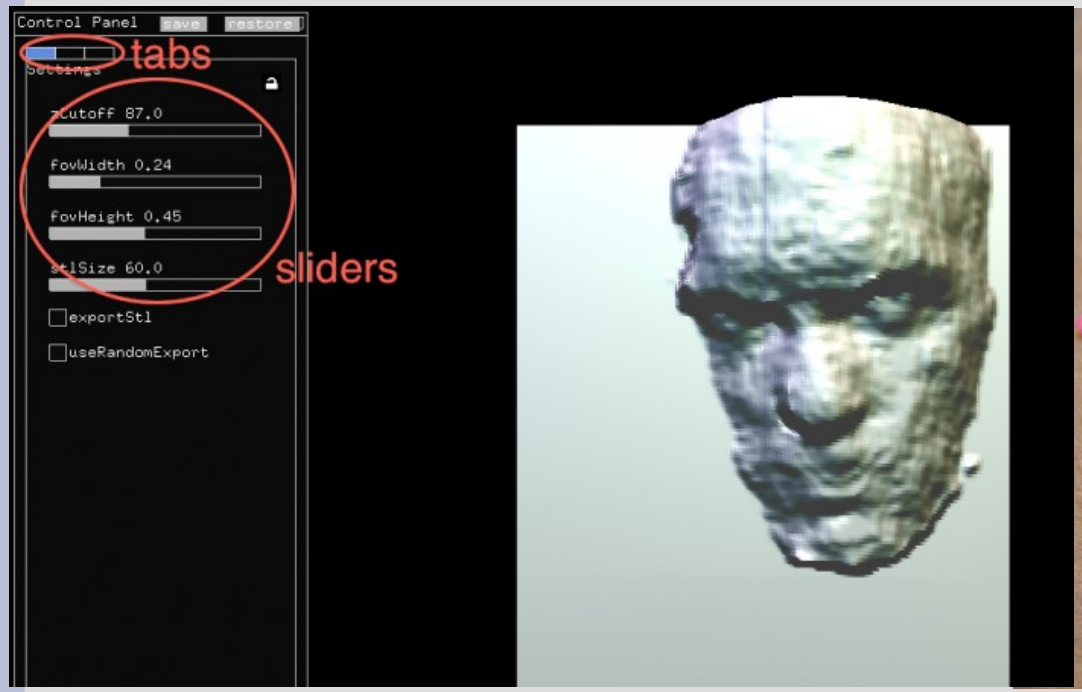


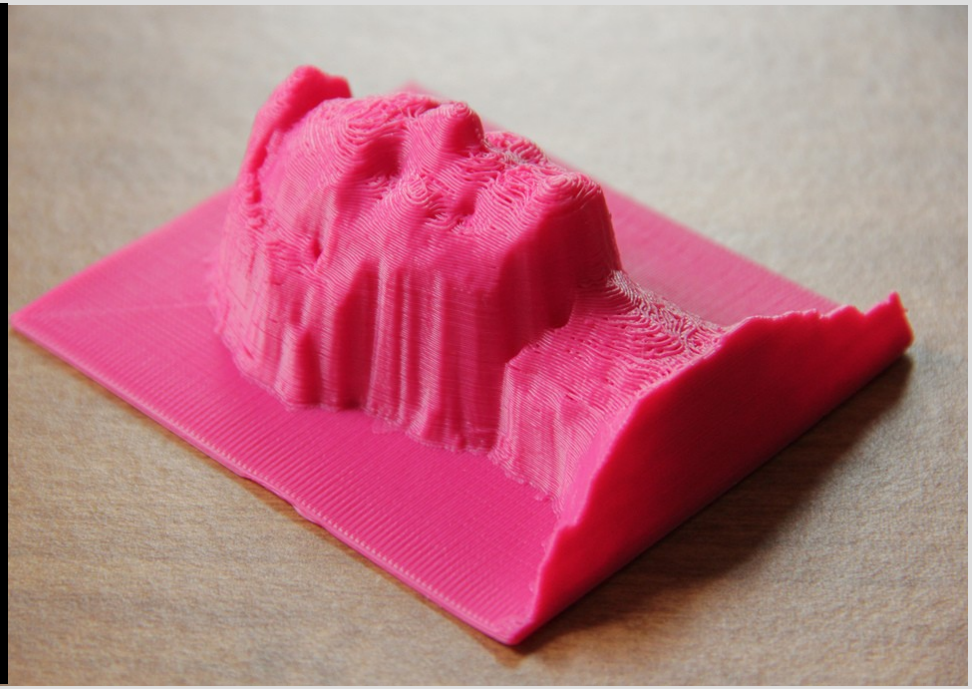
Image from http://www.futurepicture.org/?p=116

If something is in between the background and the Kinect the dots are perturbed and we can work out the depth value from the intersection of rays from laser and camera.

Uses of the Kinect

Simple “face protruding from carbonite” type 3d model with KinectToSTL





Images from http://www.makerbot.com/blog/2011/05/26/3d-printing-with-kinect/

Uses of the Kinect

Merging of multiple images into one full 3D scan

For example put object on turntable take 36 snapshots from Kinect, each one with turntable rotated 10 degrees and then merge the 2.5D information and the known rotations to get 3D point cloud in common coordinates then turn into triangular mesh.

http://kaikostack.no-ip.com/ftp/temp/kinect/
http://blenderartists.org/forum/showthread.php?202543-Kinect-3D-Scanner-development/

For more Kinect hacks

... see http://kinecthacks.net/ (of course)

For example:

- Create your own Autostereogram (go crosseyed and see the image jump out at you)
- Capture source for holographic Princess Leia i.e. Low budget motion capture
- Make any surface multi-touch

Cool near-future “light field” camera

Take the largest consumer camera sensor size and multiply by the densest number of sensors -> approx 100MP can be done with current tech.

What would we use the additional information for?

Add micro-lenses in front of them to get a “light field” camera.

3 visualisations of the extra information:
Ng PhD thesis pages 28,30,32

http://www.lytro.com/

Post-process refocusing

3d information using
- depth from de-focus
- depth from parallax

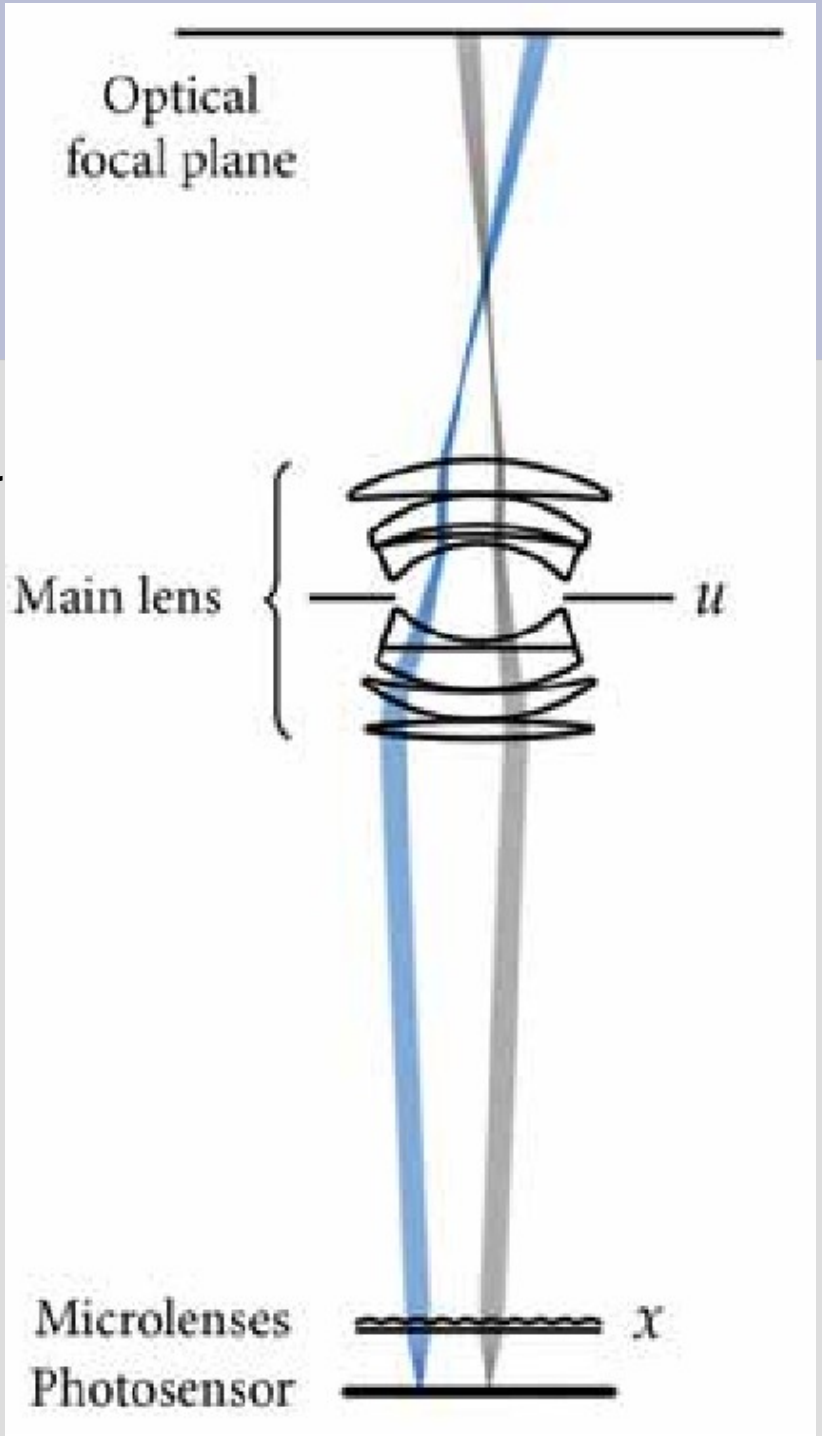


Image also from thesis