Augmented Reality

SIGB Assignment 3

Christoffer Stougaard Pedersen, Morten Roed Frederiksen, Sigurt Bladt Dinesen

{mrof,sidi,cstp}@itu.dk

IT-University of Copenhagen
SIGB, F2013
Dan Witzner Hansen & Diako Mardanbeigi
May 13, 2013

Contents

1 Introduction

This is the first half of assignment three in the course SIGB on the IT-University. It will cover the touch upon the subjects outlined, but these will all be elaborated in depth in the second half

2 Camera Calibration

A homography can be defined as a tranformation from a projective space onto itself. As such, it is insufficient when trying to transform objects from a three-dimensional space to a two-dimensional space, e.g. a cube onto an image plane. To properly display our cube in the image plane, a projectivty from our world coordinate system to the image plane is needed. The process of finding this projectivty is commonly referred to as *camera calibration*

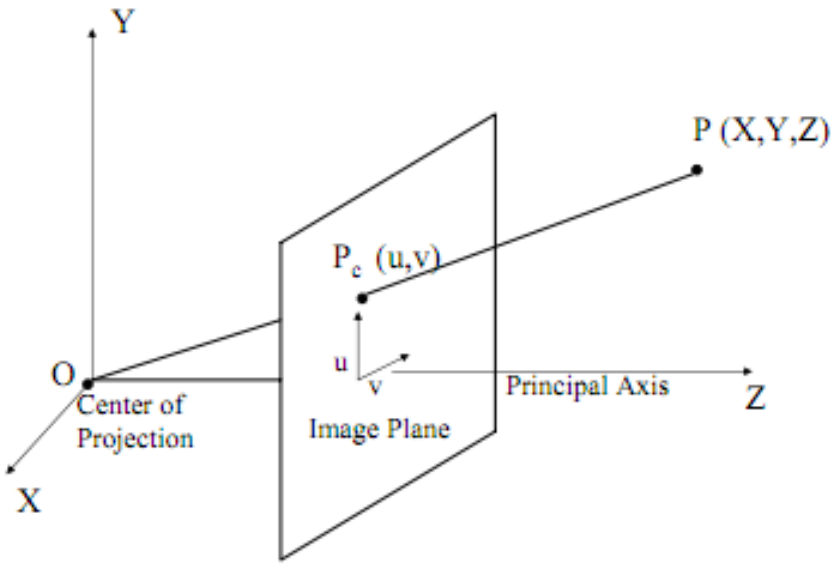

Figure 1: Visualization of world-coordinate *(X, Y, Z)*, projection on to the image plane. *O* corresponds to the focal point of the camera.

For this assignment, two different methods for calibrating the camera was utilised:

The first method is that of OpenCV's cameraCalibrate function. By introducing an object of known dimensions, the camera's parameters can be inferred. The

focal length('s) can be found by the formula

$$fx = \frac{dx}{dX} * dZ, \quad fy = \frac{dy}{dY} * dZ$$

Where fx and fy are the focal length expressed in pixels on the x and y axes of the image respectively.

3 Augmented Cube

In this assignment we compare two methods of calculating camera views.Both utilize that the camera is already calibrated, (See earlier section for info about calibration) but uses different methods for calculating the extrinsic parameters.

3.1 Method one, using homography

This method is the same as in the last assignment, so we'll explain it very briefly. As K is known in our projection matrix K*[R—T] we need to calculate R and T. As we have an existing view and four points within that view, that corresponds to four ponts in our current frame. we can calculate a homography between the two planes and use that to calculate our existing projection matrix. We do that simply by multiplying our existing matrix with the homography. cam2 = H * cam1 in the last assignment we also used a homography to transfer the extrinsic parameters to a new space.There we estimated the "Z" axis by finding the cross product between the "X" and "Y" vectors, finding the orthogonal vector. This time, the z axis is already calculated in the first camera, and are "transfered" as well with the homography. Our results so far shows that the projection of the z axis is not nearly as precise as when we use the second method.

3.2 Method two, direct calculation

The second method also utilizes that the kamera matrix is already known. Where method one tracked the chessboard patterne in two different views, this method only tracks points in the current frame. The other chesboard points are calculated from knowledge about the pattern. When we know the starting coordinate, the size of each square and number of squares, the points can be calculated precisely from the decided origin. cv2.solvePnP is used to estimate a camera position relative to the surface.This relation effectively gives us the extrinsic vectors.We convert those to 3x3 matrices (Rodrigues), and stack them together. Lastly we take the dot product between the resulting matrix and our kamera matrix, ending up with our new projection matrix.

4 Back-Face Culling

The more polygons we have to draw, the more slower our execution of the program gets. In order to cut down on the expensive operations went want a method that sorts out the surfaces of our cube, that are not visible. For that we use Back-Face culling. BFC is a method to determine whether a given surface is visible. More specifically it's a way to determine if a polygon of an object is occluded.

To achieve this two vectors are needed. The normal on the surface of the object, and the vector from the camera center to the object surface. The angle between these two vectors determine weather the surface is viewable from the camera's view point. As the angle increases, the surface reaches a point where its only visible as a line, and after that we only see the back side of the surface, and it should no longer be drawn.

The orthonormal of the object surface can be found by the cross product of the surface's standard basis. For cube's faces, this is easily defined as two vectors between the corners (avoiding diagonals), if we relax the claim of orthonormality to that of orthogonality.

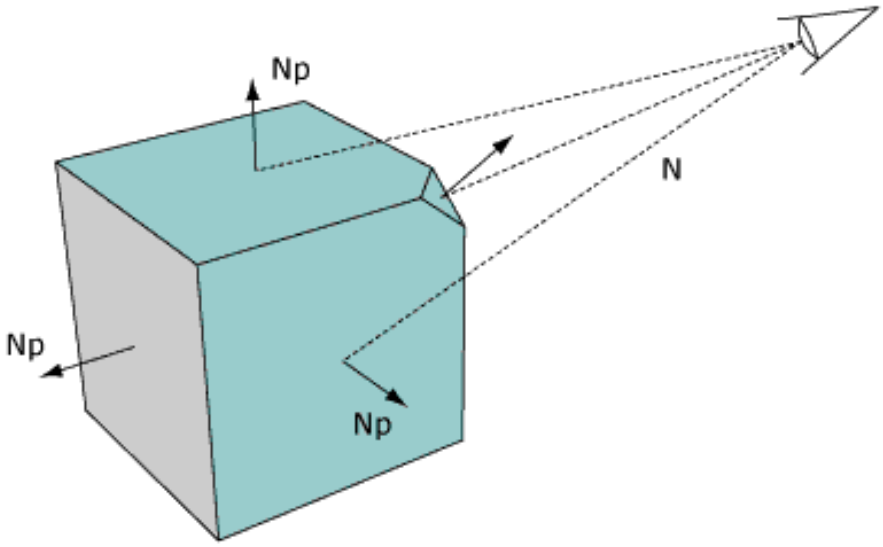

Figure 2: Normals (denoted Np) of a cube's faces, and the vectors from the camera center to the centers of those faces)