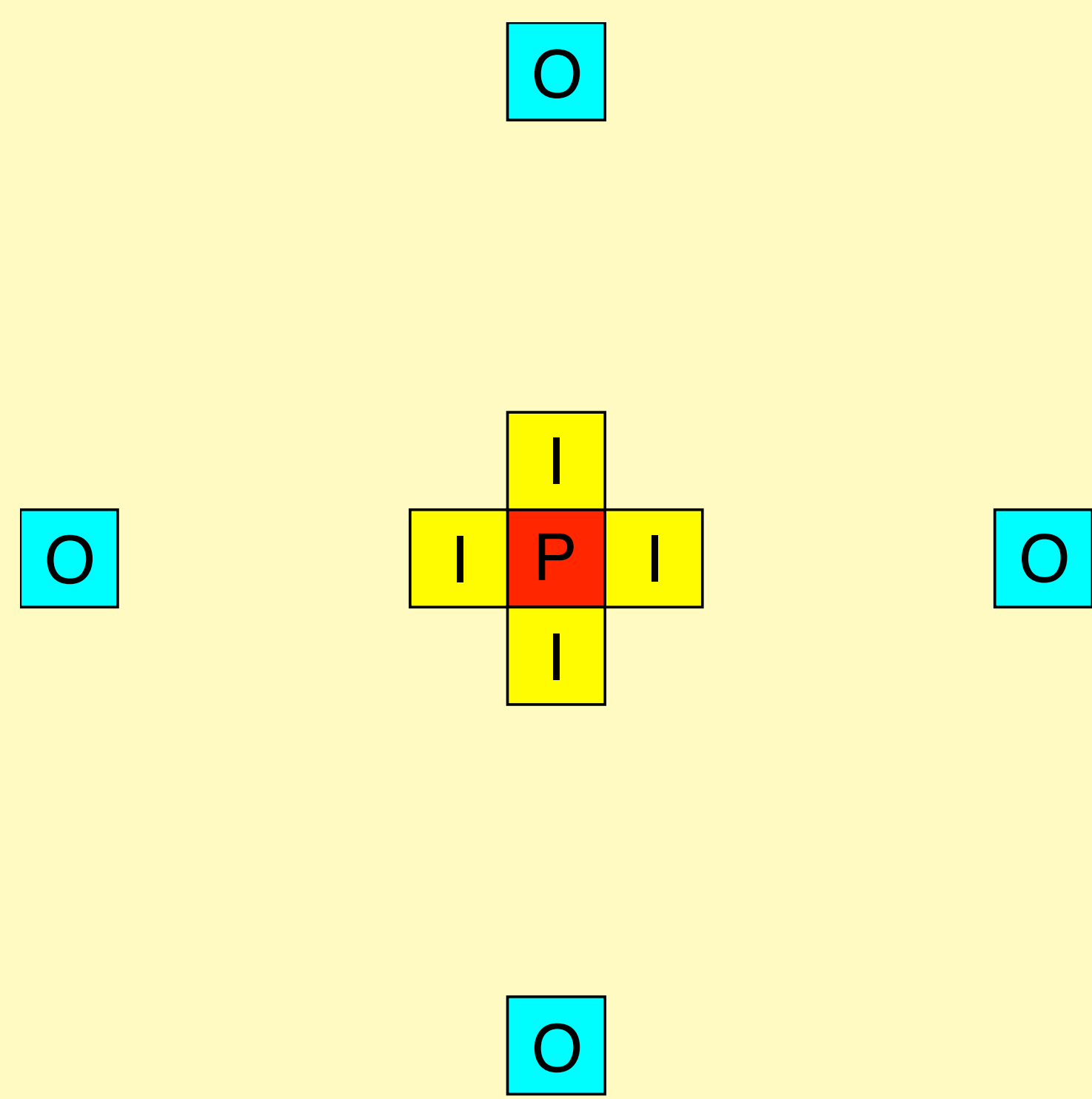


Abstract

Project Laser pointer drawing is about computer processing images captured through the webcam. In image from camera is found lighted point, coordinates of this point are transformed to dataprojector coordinates and they are used to draw on screen or do something else. Application is written in Objective C and uses Mac OS X libraries.

Processing image

Images are captured by framework Cocoa-SequenceGrabber. We need to find the lighted point, if it is on image. To find it, it is used cross method. In this method there are three types of points:



- Central point  $P$ , which is the current tested point
- Inner points  $I$ , whose are on the top, bottom left and right of the central point.
- Outer points  $O$ , whose are five pixels on the top, bottom, left and right of the central point and represents the background

The central point goes point by point through all image and computes average of inner and outer points, whose are aviable and compare them with configured values. If it's lower than in configuration, they are discarted. From not discarted points is chosen point with maximal values.

Transformation

We need to transform coordinates from image to projector, because the axis of the camera isn't parallel with axis of the screen, so the image is distorted – the screen is not rectangle on the image from camera, but some quadrilateral. To transform it to rectangle is used perspective transformation, where each point is transformed with these equations:

$$x' = \frac{a_0x + a_1y + a_2}{a_6x + a_7y + 1} \tag{1}$$

$$y' = \frac{a_3x + a_4y + a_5}{a_6x + a_7y + 1} \tag{2}$$

The transformation could be written to matrix:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cdot \begin{bmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ a_6 & a_7 & 1 \end{bmatrix}$$

To get transformed values of  $x$  and  $y$  we need to normalise them:

$$\begin{aligned} x' &= \frac{u}{w} \\ y' &= \frac{v}{w} \end{aligned}$$

Viewing

There are many possible ways how to visualise results, some of them are mentioned below.

Single color drawing

On the current position of laser pointer is showed cross and the position is added to the `NSBezierPath` object which is drawn with green color. If there isn't a point from laser pointer, cross isn't drawn and the path is diconnected. If the light is back again, virtual "pen" is moved to new position and starts drawing again. The width of "pen" could be changed from GUI. From GUI could be also reseted a drawing.

Multi color drawing

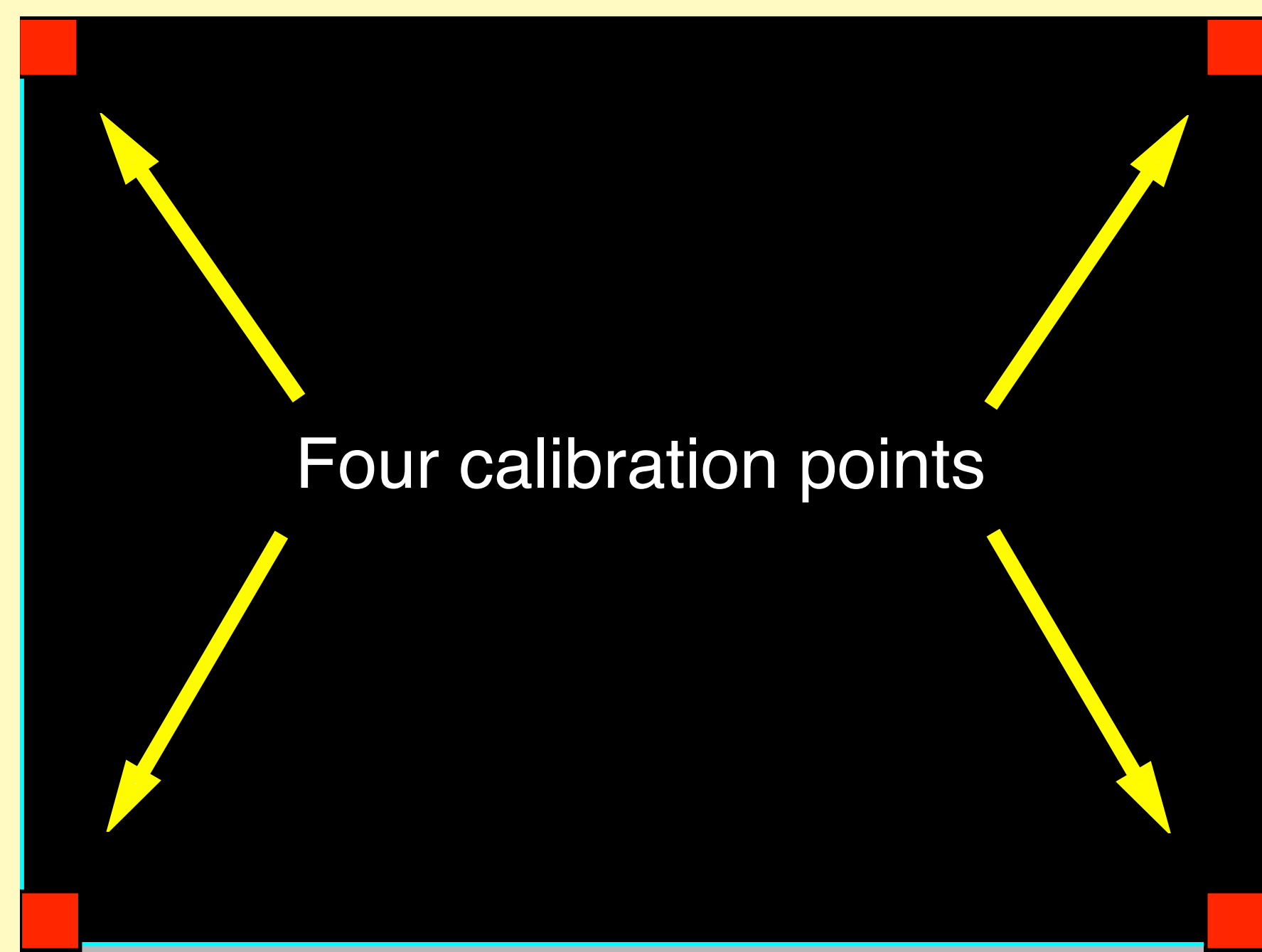
Principle of this drawing is the same as in single color drawing. There are some additional things – three squares with three colors. If the point from pointer is in this square, color of the virtual "pen" changes to the color of square.

Shooting baloons

This is future application, which will be probably developed in few weeks or months. There will be moving baloons represented by circles on the screen. there will be moving with variable speeds and player will try to shoot them with laser pointer. In more advanced version of this game there could be more lasers with different colors, one for each player.

Calibration

To get the right values of  $a_0, \dots, a_7$  we need some calibration points. For this reason, there is an calibration. After start of calibration, four calibration points are showed in sequence by dataprojector in corners of drawable area.



For each point program finds its position in image using cross method from Processing image. After all points are showed, coordinates of their positions are sent to transformation classes, which computes its constants for transformation.

References

[1] D.-K. Kim, B.-T. Jang, C.-J. Hwang A Planar Perspective Image Matching using Point Correspondences and Rectangle-to-Quadrilateral Mapping in *Image Analysis and Interpretation, IEEE Southwest Symposium 2002*

Thanks

Thanks to Marek Blaha, Tomáš Gavenčák, Lukáš Langer, Jan Olšina and other people, who helped me with the project.