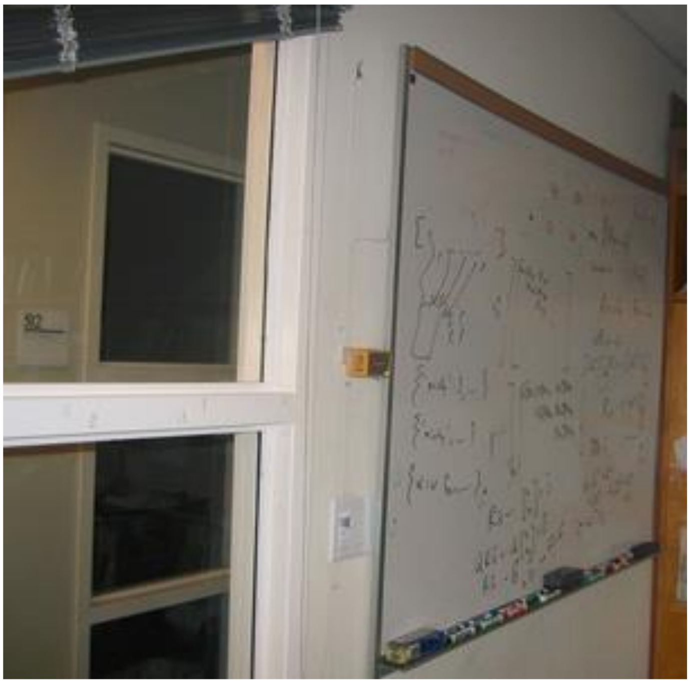
Perspective Rectification

The goal for this project is to remove perspective from an image of a flat surface.

Consider the following image of the whiteboard in the office:



We will synthesize a new image. This new image has never been captured by a camera; in fact, it would be impossible using a traditional camera since it completely lacks perspective:

The technique for carrying out this transformation makes use of the idea of a coordinate system. Actually, it requires us to consider two coordinate systems and to transform between a representation within one system and a representation within the other. Think of the original image as a grid of rectangles, each assigned a color. (The rectangles correspond to the pixels.) Each such rectangle in the image corresponds to a parallelogram in the plane of the whiteboard. The perspective-free image is created by painting each such parallelogram the color of the corresponding rectangle in the original image. Forming the perspective-free image is easy once we have a function that maps pixel coordinates to the coordinates of the corresponding point in the plane of the whiteboard. The basic approach to derive this mapping is by example. We find several input-output pairs—points in the image plane and corresponding points in the whiteboard plane—and we derive the function that agrees with this behavior.

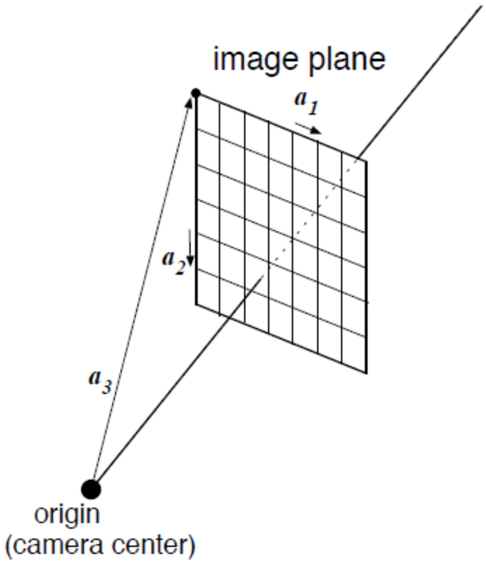
So, we need camera basis, whiteboard basis. Then we are mapping from pixels to points on the whiteboard. After mapping a point not on the whiteboard to the corresponding point on the whiteboard we are computing the change-of-basis matrix. And finally, We are making image representation.

The camera basis

We use the camera basis a_1, a_2, a_3 where:

- The origin is the camera center.
- The first vector a_1 goes horizontally from the top-left corner of the top-left sensor element to the top-right corner.
- The second vector a_2 goes vertically from the top-left corner of the top-left sensor element to the bottom-left corner.
- The third vector a_3 goes from the origin (the camera center) to the top-left corner of sensor element (0,0).

This basis has the advantage that the top-left corner of sensor element (x_1, x_2) has coordinate representation $(x_1, x_2, 1)$.

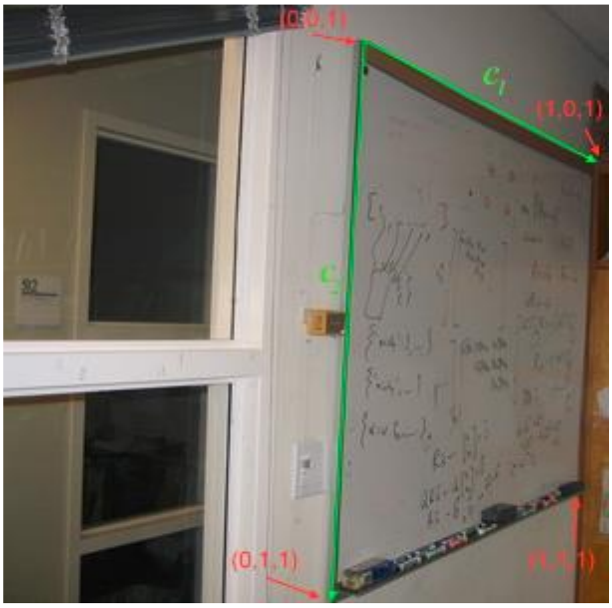


The whiteboard basis

In addition, we define a whiteboard basis c_1, c_2, c_3 where:

- The origin is the camera center.
- The first vector c_1 goes horizontally from the top-left corner of whiteboard to top-right corner.
- The second vector c_2 goes vertically from the top-left corner of whiteboard to the bottom left corner.
- The third vector c_3 goes from the origin (the camera center) to the top-left corner of the whiteboard.

As we will see, this basis has the advantage that, given the coordinate representation (y_1, y_2, y_3) of a point q, the intersection of the line through the origin and q with the whiteboard plane has coordinates $(y_1/y_3, y_2/y_3, y_3/y_3)$.



Mapping from pixels to points on the whiteboard

Our goal is to derive the function that maps the representation in camera coordinates of a point in the image plane to the representation in whiteboard coordinates of the corresponding point in the whiteboard plane.

At the heart of the function is a change of basis. We have two coordinate systems to think about, the camera coordinate system, defined by the basis a_1, a_2, a_3, and the whiteboard coordinate system, defined by the basis c_1, c_2, c_3. This gives us two representations for a point.

Each of these representations is useful:

1. It is easy to go from pixel coordinates to camera coordinates: the point with pixel coordinates (x_1, x_2) has camera coordinates $(x_1, x_2, 1)$.
2. It is easy to go from the whiteboard coordinates of a point q in space to the whiteboard coordinates of the corresponding point p on the whiteboard: if q has whiteboard coordinates (y_1, y_2, y_3) then p has whiteboard coordinates $(y_1/y_3, y_2/y_3, y_3/y_3)$.

In order to construct the function that maps from pixel coordinates to whiteboard coordinates, We need to add a step in the middle: mapping from camera coordinates of a point q to whiteboard coordinates of the same point. To help us keep track of whether a vector is the coordinate representation in terms of camera coordinates or is the coordinate representation in terms of whiteboard coordinates, we will use different domains for these two kinds of vectors.

A coordinate representation in terms of camera coordinates will have domain $C=\{'x_1','x_2','x_3'\}$.

A coordinate representation in terms of whiteboard coordinates will have domain $R=\{'y_1','y_2','y_3'\}$.

Our aim is to derive the function $f : \mathbb{R}^C \rightarrow \mathbb{R}^R$ with the following spec:

- input: the coordinate representation x in terms of camera coordinates of a point q

- output: the coordinate representation y in terms of whiteboard coordinates of the point p such that the line through the origin and q intersects the whiteboard plane at p.

There is a little problem here; if q lies in the plane through the origin that is parallel to the whiteboard plane then the line through the origin and q does not intersect the whiteboard plane. We'll disregard this issue for now.

We will write f as the composition of two functions $f = g \circ h$, where

- $h : \mathbb{R}^C \rightarrow \mathbb{R}^R$ is defined thus:

  – input: a point's coordinate representation with respect to the camera basis

  – output: the same point's coordinate representation with respect to the whiteboard basis

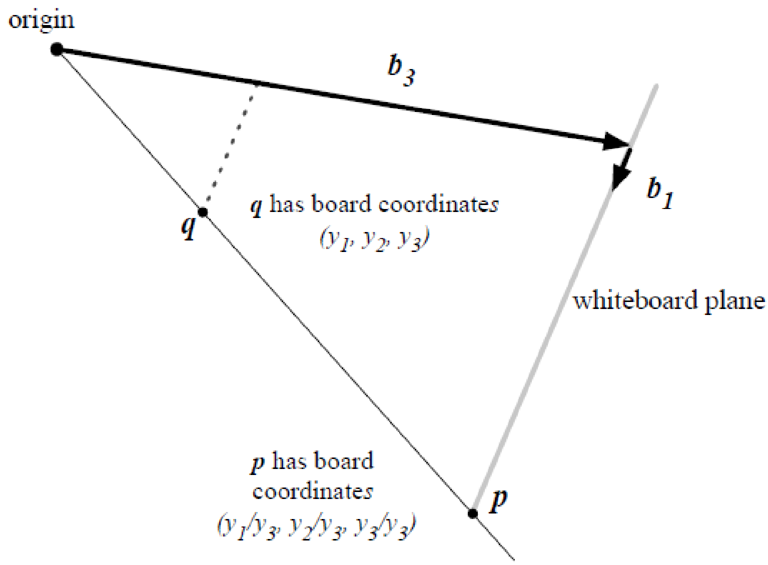
- $g : \mathbb{R}^R \rightarrow \mathbb{R}^R$ is defined thus:

  – input: the coordinate representation in terms of whiteboard coordinates of a point q

  – output: the coordinate representation in terms of whiteboard coordinates of the point p such that the line through the origin and q intersects the whiteboard plane at p.

Mapping a point not on the whiteboard to the corresponding point on the Whiteboard

In this section, we develop a procedure for the function g. We designed the whiteboard coordinate system in such a way that a point on the whiteboard has coordinate y_3 equal to 1. For a point that is closer to the camera, the y_3 coordinate is less than 1.
Suppose q is a point that is not on the whiteboard, e.g. a point closer to the camera. Consider the line through the origin and q. It intersects the whiteboard plane at some point p.



This figure shows a top-view of the situation.

In this view, we see the top edge of the whiteboard, a point q not on the whiteboard, and the point p on the whiteboard that corresponds to q.

Let the whiteboard-coordinate representation of q be (y_1, y_2, y_3). In this figure, y_3 is less than 1. Elementary geometric reasoning (similar triangles) shows that the whiteboard-coordinate representation of the point p is $(y_1/y_3, y_2/y_3, y_3/y_3)$. Note that the third coordinate is 1, as required of a point in the whiteboard plane.

The change-of-basis matrix

You have developed a procedure for g. Now we begin to address the procedure for h. Writing a point q in terms of both the camera coordinate system a_1, a_2, a_3 and the whiteboard coordinate system c_1, c_2, c_3, and using the linear-combinations definition of matrix-vector multiplication, we have:

$$\begin{bmatrix} \\ q \\ \\ \end{bmatrix} = \left[\begin{array}{c|c|c} & & \\ a_1 & a_2 & a_3 \\ & & \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \left[\begin{array}{c|c|c} & & \\ c_1 & c_2 & c_3 \\ & & \end{array}\right] \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Let $A = \left[\begin{array}{c|c|c} & & \\ a_1 & a_2 & a_3 \\ & & \end{array}\right]$ and let $C = \left[\begin{array}{c|c|c} & & \\ c_1 & c_2 & c_3 \\ & & \end{array}\right]$. Since the function from $\mathbb{R}^3$ to $\mathbb{R}^3$ defined by $y \mapsto Cy$ is an invertible function, the matrix C has an inverse C^{-1}. Let $H = C^{-1}A$. Then a little algebra shows

$$\begin{bmatrix} & & \\ & H & \\ & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

This is just a recapitulation of the argument that change of basis is matrix-multiplication.

Write $H = \begin{bmatrix} h_{y_1,x_1} & h_{y_1,x_2} & h_{y_1,x_3} \\ h_{y_2,x_1} & h_{y_2,x_2} & h_{y_2,x_3} \\ h_{y_3,x_1} & h_{y_3,x_2} & h_{y_3,x_3} \end{bmatrix}$.

Let $\boldsymbol{q}$ be a point on the image plane. If $\boldsymbol{q}$ is the top-left corner of pixel x_1, x_2 then its camera coordinates are $(x_1, x_2, 1)$, and

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} h_{y_1,x_1} & h_{y_1,x_2} & h_{y_1,x_3} \\ h_{y_2,x_1} & h_{y_2,x_2} & h_{y_2,x_3} \\ h_{y_3,x_1} & h_{y_3,x_2} & h_{y_3,x_3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

where (y_1, y_2, y_3) are the whiteboard coordinates of $\boldsymbol{q}$.

Multiplying out, we obtain

$$y_1 = h_{y_1,x_1} x_1 + h_{y_1,x_2} x_2 + h_{y_1,x_3} \quad (5.5)$$
$$y_2 = h_{y_2,x_1} x_1 + h_{y_2,x_2} x_2 + h_{y_2,x_3} \quad (5.6)$$
$$y_3 = h_{y_3,x_1} x_1 + h_{y_3,x_2} x_2 + h_{y_3,x_3} \quad (5.7)$$

Computing the change-of-basis matrix

Now that we know a change-of-basis matrix H exists, we don't use the camera basis or the whiteboard basis to compute it because we don't know those bases! Instead, we will compute H by observing how it behaves on known points, setting up a linear system based on these observations, and solving the linear system to find the entries of H.

If we had a point with known camera coordinates and known whiteboard coordinates, we could plug in these coordinates to get three linear equations in the unknowns, the entries of H. By using three such points, we would get nine linear equations and could solve for the entries of H.

Image representation

A generalized image consists of a grid of generalized pixels, where each generalized pixel is a quadrilateral (not necessarily a rectangle).

The points at the corners of the generalized pixels are identified by pairs (x, y) of integers, the pixel coordinates.

Each corner is assigned a location in the plane, and each generalized pixel is assigned a color. The mapping of corners to points in the plane is given by a matrix, the location matrix. Each corner corresponds to a column of the location matrix, and the label of that column is the pair (x, y) of pixel coordinates of the corner. The column is a {'x','y','u'}-vector giving the location of the corner. Thus the row labels of the location matrix are 'x', 'y', and 'u'. The mapping of generalized pixels to colors is given by another matrix, the color matrix.

Each generalized pixel corresponds to a column of the color matrix, and the label of that column is the pair of pixel coordinates of the top-left corner of that generalized pixel. The column is a {'r','g','b'}-vector giving the color of that generalized pixel. We will apply a transformation to the locations to obtain new locations, and view the resulting image.