ADVANCED TOPICS IN COMPUTER VISION

Pose from a Planar Target

Francesc Moreno-Noguer
Institut de Robòtica i Informàtica Industrial, CSIC-UPC

May 16, 2016

The goal of this exercise is to compute the pose of the camera for an image using a planar object, and to use this pose to augment the image with a virtual object.

1 HOMOGRAPHY COMPUTATION

We will first estimate the homography $\mathbf{H}$ that maps the plane defined by the book cover and the image plane. For simplicity, lets choose the world coordinate system so that the book cover lies on the plane $Z = 0$. $\mathbf{H}$ is then the 3×3 matrix that:

$$\begin{bmatrix} ku \\ kv \\ k \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1.1}$$

where $\mathbf{p} = [x, y, 0]^\top$ is a 3D point on the book cover and $\mathbf{u} = [u, v]^\top$ its projection in the image.
Knowing that the dimensions of the book are $17.3cm \times 24.6cm$, you can retrieve the 3D locations of the corners of the book, and also their projections using an image visualization software.
The coefficients of $\mathbf{H}$ can be estimated from these 3D-2D correspondences using the DLT algorithm, explained for a projection matrix.
For numerical stability reasons, it is better to take the origin of the world coordinate system at the center of the book. The specific reference systems I used for both the 2D and 3D coordinates are shown in Fig. 1.1.
I found the following matrix:

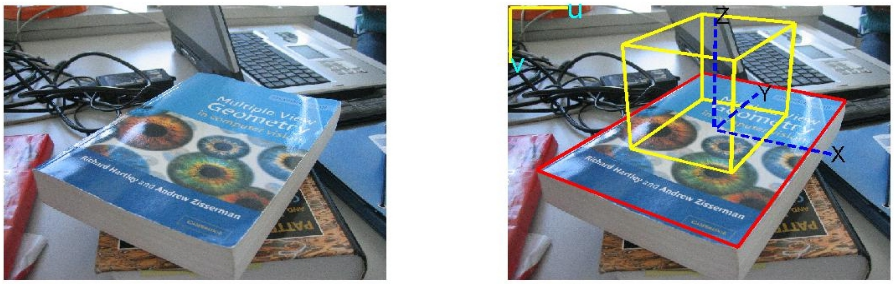

Figure 1.1: In this exercise, we will use the book in the image to register the camera, and use the computed pose to augment the image with a simple virtual 3D object.

$$\mathbf{H} = \begin{bmatrix} 0.02954320 & 0.03944075 & 0.84447694 \\ 0.00448948 & -0.00841732 & 0.53322916 \\ -0.00002768 & 0.00004912 & 0.00244085 \end{bmatrix} \tag{1.2}$$

But remember that $\mathbf{H}$ is defined up to a scale factor (which can be even negative!!).

2 Pose Estimation

Let's assume that the matrix $\mathbf{A}$ of internal parameters is equal to

$$\mathbf{A} = \begin{bmatrix} 800 & 0 & 320 \\ 0 & 800 & 240 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

Use the method described during the class to compute the rotation and translation of the camera from $\mathbf{H}$ and $\mathbf{A}$.

Note: Depending on how you estimated $\mathbf{H}$, it can yield to two different poses (For instance, what happens when $\mathbf{H}$ is changed to $-\mathbf{H}$?). To understand which one of the two possible poses is correct, we can use the fact that the Z coordinate of the camera center should be positive if we want the Z-axis to point towards the top of the image (see Fig. 1.1). The camera center $\mathbf{C}$ can be computed from the rotation matrix and the translation vector as $\mathbf{C} = -\mathbf{R}^\top \mathbf{t}$. Where does this expression come from?

I found the following pose matrix:

$$[\mathbf{R}|\mathbf{t}] = \begin{bmatrix} 0.88571704 & 0.46420481 & -0.00438161 & 1.32421615 \\ 0.20784735 & -0.40498346 & -0.89038635 & -1.09807761 \\ -0.41509611 & 0.78771967 & -0.45518450 & 40.78287458 \end{bmatrix} \tag{2.2}$$

and the corresponding position of the camera center: $\mathbf{C} = [15.98 \quad -33.18 \quad 17.59]^\top$.

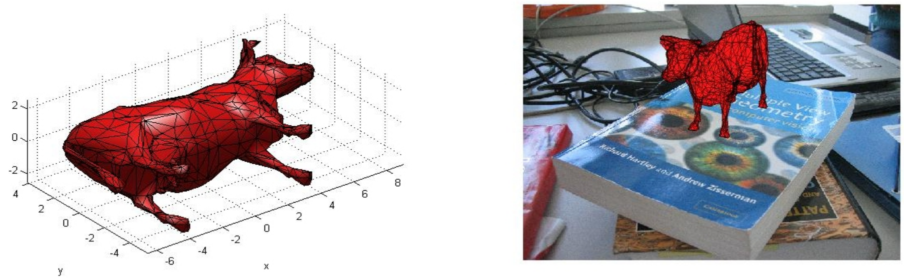

Figure 3.1: Augmenting the image with a virtual object.

3 Projection of a Virtual Object

Finally we simply have to multiply the retrieved $[\mathbf{R}|\mathbf{t}]$ matrix by the matrix $\mathbf{A}$ on the left to compute the projection matrix for the image.
Use this projection matrix to compute the projections of the vertices of some simple 3D object you can define, and use these projections to draw a wireframe of your object over the input image.
Compute also the projection of the world coordinate reference frame, placed on the center of the book, and with axes lengths of $10cm$, as shown in Fig. 1.1-right.
Use also this matrix to project a more complex 3D model, like the *cow* shown in Fig. 3.1-left. Note that before projecting the 3D model onto the image, you will need to align it to the world coordinate reference frame. Below you will find part of the code you will need to load the 3D model and display it both in 3D and 2D.

```
load('model3d-cow.mat','fv');
V=fv.vertices;
F=fv.faces;

%plot original model in 3D
figure; subplot(1,2,1);
trisurf(F,V(:,1),V(:,2),V(:,3),'FaceColor',[1,0.1,0.1],...
     'EdgeColor',[0.0 0.0 0.0]);
light('Position',[-1.0,-1.0,100.0],'Style','infinite');
lighting phong;
xlabel('x');
ylabel('y');
zlabel('z');
axis equal;

%rotate and translate model 3d to align it with the book
  % TO DO

%project model onto the image
  % TO DO

%plot projected model in 2D
subplot(1,2,2); imshow(I);
```

```
Color=repmat([1,0.1,0.1],nV,1);
fv2d.vertices=[uCow',vCow'];
fv2d.faces=F;
fv2d.facevertexcdata=Color;
p = patch(fv2d,'FaceAlpha',1,'EdgeAlpha',0.25);
shading faceted;
```

Send your code and a **very short report to fmoreno@iri.upc.edu before May 31st, 11.59 pm.**