COMP 558 Assignment 4

Prepared by Prof. Michael Langer and T.A.s Chatavut and Florian
Posted: Wednesday, Nov. 25, 2010
Due*: Friday, Dec. 3 ,11:59 PM

*You will receive 1 bonus point if you submit by the due date. You may submit anytime up to Sunday, Dec. 12, 11:59 PM without penalty. Assignments will not be accepted after that date.

Introduction:

The purpose of this assignment is to give you experience with camera calibration (Q1) and homographies such as used in image stitching applications (Q2).

The images **camera1.jpg** and **camera2.jpg** show two views of a simple scene. The images were shot with the camera on a tripod. The camera was rotated between the two shots. The camera internals were the same for the two shots, but the externals are obviously different. (There is also a small lighting change between shots, but this is irrelevant to the assignment.)

Several 3D points **XYZ** are given to you. These were obtained with a tape measure. Units are **mm**. The corresponding image pixel projections **xy** are also given. They were obtained by hand labeling using Matlab's Data Cursor. The **XYZ** and **xy** data are found in **readPositions.m**.

Question 1 (7 points)

a) See starter code **Q1.m** which displays each image, and calls other scripts including **readPositions.m**. Write the script **calibrate.m** that estimates the 3x4 projection matrix **P**, given coordinates **XYZ** and corresponding image pixels **xy.**

b) You are given a script **decomposeP.m** which factors **P** into **K R [I c].** The values of **P** and hence of **K, R, c** are somewhat sensitive to the accuracy of **xy** values (and also to the accuracy of X,Y,Z, but we ignore these here). To examine the sensitivity to **xy,** modify **Q1.m** so that it run many trials (at least 50). Each trial will add mean zero Gaussian noise of standard deviation **sigma_n** to all the original **x,y** values. Examine and briefly discuss how the standard deviation of the elements of the **K** matrix depend on the standard deviation of the Gaussian noise. You should consider two different values of **sigma_n**.

   Note: **R** and **c** also vary but you may ignore them here. Also, to keep it simple, just discuss the results for one of the images e.g. **camera1.jpg**.

   The purpose here is for you to appreciate how the estimate of **P** is limited by the accuracy of the **XYZ** and **xy** positions. To get around this limitation, it is recommended that one use many points. For example, in Hartley and Zisserman's book "Multiview Geometry in Computer Vision", a rule of thumb is given that the number of data values (16=8x2, in our case) should be 5 times greater than the number of unknowns (11, in the case of **P)**. Note that our example does not follow their rule of thumb since 16 is much less than 11x5.

Question 2 (8 points)

The camera rotation between images induces a small translation, since the center of rotation of the tripod is different from the center of projection of the camera. If the camera had only been rotated about the center of projection, then the two images would have been related by homography.

a) *Assume* (incorrectly) that the camera was *only rotated, and* also assume (incorrectly) that the calibration matrices you obtained for the two images in Question 1a were correct (up to noise). Write a script **homography1.m** that computes a homography based on those matrices.

b) Write a script **homography2.m** that fits a homography directly to the given **xy** points.

c) Modify the script **Q2.m** so that it constructs and displays a new image called **newCamera1** whose **xy** pixel *positions* are aligned with those of **camera1**, but whose RGB intensities are taken from the **camera2** image, using the inverse mapping method described in the lecture 20 slides. Compare the **newCamera1** images that are obtained using the two homographies in (a) and (b). Briefly discuss which performs better and why.

In part (c), to facilitate comparison between the **camera1** and **newCamera1** images, overlay **camera1** and **newCamera1**, similar to what was done in the example at the end of lecture 20 slides, namely use the red channel for one image and the green channel for the other. Also, explicitly plot the inverse mapped **xy** points along with the original camera1 **xy** points in the **newCamera1** image, using similar point plots as in Question 1.

The image **newCamera1** should be larger than the original **camera1**, since there are pixels in **camera2** that inverse map outside the domain of **camera1**. When writing/debugging your code, we suggest you begin by making **newCamera1** the same size as **camera1.** Once you have achieved a solution for this case, generalize the solution by expanding the domain of **newCamera1**.

[ASIDE: (added Thurs. Nov. 25 2 pm) As mentioned in the WebCT Discussion Board under topic Assignment 4, I have added smaller versions of camera1 and camera2 images. These smaller images are just subsampled (every 4^{th} pixel in the x direction and y direction). We suggest that you use the smaller images for Q2c, in particular. You may of course use the smaller images for Q1 also. *However, if you do use the smaller images, then you will need to divide the (x,y) coordinates by 4*.]