

Comp. 4102: Assignment #3  
Due: Friday March 28, 2014

- 1) The goal of the first questions is to implement some code that performs calibration using the method described in the book; by first computing a projection matrix, and then decomposing that matrix to find the extrinsic and intrinsic camera parameters. On the web site I have given you a program, written in C that uses OpenCV, called `assign3_projection_shell_14.c`. This program takes ten given 3d points and projects them into a 2d image using the given camera calibration matrix, rotation matrix and translation vector. Your goal is to write the two routines that are missing, which are `computeprojectionmatrix` and `decomposeprojectionmatrix`. The first routine computes the projection matrix using the method described in Section 6.3.1 of the book, and the second uses the method in Section 6.3.2 to decompose the projection matrix into a camera calibration matrix, rotation matrix and translation vector. It should be the case that the computed camera matrix, rotation matrix and translation vector are the same (or very similar) to the original versions that were used to create the projected points. This shows that your two routines are working properly. You hand in your program source and the resulting output file `assign2-out` created by running this modified program.

**5 marks**

- 2) The goal of this question is to create a program that take as input three images that are related by a homography; a left (`keble_a_half`), middle (`keble_b_long`) and right image (`keble_c_right`), and creates a single panoramic image (same size as `keble_b_long`) as output. This is done by warping the left and right image “into” the middle image. I have made the middle image big enough to hold both the warped left and right image. Your program needs to find some features between the left and middle images, match these features to compute a homography, and use this homography to warp the left image into the middle image. Then this process is repeated for the right image relative to the middle image. To make life easier I include the final output image that I created with my version of this program as `three_images.jpg`. You also need to combine the two warped images and the original center image in a way that makes sense so that the final image is visually correct, which means that it looks similar to my output image. In the zip file for this assignment I have included a new version of the Visual Studio OpenCVExample which has the proper library included for compiling and running the program `find_obj.cpp` (which has been copied into `OpenCVExample.cpp`). The reason for this is that the easiest way to answer the question is to modify this example program but still use the same basic idea; find surf features in all images, find matches between images and then compute the appropriate homographies. If you do not use Visual Studio you must be sure to link with the library `opencv_nonfree248.lib` (or `opencv_nonfree248d.lib`) to be able to compile and run this example program because it uses Surf features (Surf is not free, thus the library name).

**5 marks**