

Comp. 4102A: Assignment #3  
Due: Friday March 22, 2013, 9:00 AM

- 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 parameters. On the web site I have given you a program, written in C that uses OpenCV, called assign3-projection-shell.c. This program takes ten 3d points, and projects them using the given camera 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 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. The program assign2-projection-shell.c is on CuLean and the web site. You hand in your program source and the resulting output file assign2-out created by running the program. **5 marks**
- 2) The goal of this question is to create a program that take as input two images that are related by a homography, and which “warps” the second image (piscine2.bmp) to align with the first image (piscine1.bmp). Both of these images are on the course web site. The output image that is created should have the original first image, along with the warped version of the second image. I have made the first image big enough to hold both the original first image along with the warped version of the second image. Your program needs to find some features in the two images, match these features to compute a homography, and use this homography to help create the final image which is a combination of the two input images. The simplest way to accomplish this is to use the code from the program findmatch.cpp in the C Samples directory (which you can find in the Windows OpenCV program menu) and modify it as necessary. Send me your source code and the final output image that you create which combines the two input images. In the area of overlap between the two images we have a choice of using pixels from the first image, or the second image.  
**5 marks**