Student ID: ____________________

SAMPLE EXAMINATION PAPER 2013

COMPUTER SCIENCE

Paper COSC470

Special Topic: 3D Reconstruction

Full Year

(TIME ALLOWED: THREE HOURS)

This examination comprises 5 pages.

Candidates should answer questions as follows:

Candidates must answer **all** questions.
Questions are worth 20 marks each and submarks are shown thus: (5)
The total number of marks available for this examination is 80.

The following material is provided:

Nil.

Use of calculators:

No calculators are permitted.

Candidates are permitted copies of:

Nil.

Other instructions:

Please write your Student ID number at the top of this page.

At the end of the exam, hand in the entire exam attached to your answer book.

1. **Filters and Features**

(a) The kernel for a mean filter is shown below

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

(i) What is the result of applying the mean filter at the centre of the following image fragment, where the numbers indicate the intensity value of the pixels? Show your working.

8	7	3	2	1
9	8	3	1	2
7	7	2	3	1
9	8	2	2	2
8	4	2	1	3

(2)

(ii) The mean filter is a *separable* filter. What does it mean for a filter to be separable? (2)

(iii) Explain how a separable filter can be efficiently implemented, using the mean filter as an example. (2)

(iv) Explain what issues arise when applying filters near the boundaries of an image, and give a possible solution to this problem. (2)

(b) The SIFT feature descriptor is robust to orientation changes. Explain how a feature's orientation can be estimated, and how this can be used to compute a descriptor that is invariant to changes in orientation. (5)

(c) Feature descriptors typically have many dimensions (SIFT descriptors, for example, are 128-dimensional). Explain how kd-Trees can be used to find matching feature descriptors in these high-dimensional spaces. (4)

(d) Feature matches using kd-Trees are often only *approximate nearest neighbour* matches. Explain why a kd-Tree based search might not find the true nearest neighbour match. (3)

2. **Cameras and Transforms**

(a) The relationship between two views of a planar surface can be described as a *homography*:

$$x' = Hx,$$

where x is a point in one view, x' is the corresponding point in the second view, and H is a 3×3 matrix.

(i) The vectors x and x' are represented in homogeneous co-ordinates. Explain what this means, and why homogeneous co-ordinates are used. (3)

(ii) The matrix H has 9 elements, but is only defined up to a scale. How many *degrees of freedom* does H have? (1)

(iii) The matrix H can be found using the Direct Linear Transform (DLT) algorithm. This requires the solution of an equation of the form

$$Ah = 0,$$

where A is a matrix made from the co-ordinates of the matches $x \leftrightarrow x'$, and h is a 9-vector containing the elements of H. The vector $h = 0$ is one solution, but explain how the *singular value decomposition* (SVD) can be used to find non-zero solutions to such equations. (5)

(iv) The DLT algorithm is sensitive to the choice of co-ordinate frame used to represent the image feature locations. Explain how *normalising transforms* can be used to overcome this problem. (5)

(b) The pinhole camera model used in this course can be expressed with the following equation:

$$x = K[R|t]X,$$

where X is a 3D point in the world, and x is the corresponding 2D image point.

(i) Explain what K, R and t represent. (3)

(ii) The matrix R has 9 elements but only 3 degrees of freedom, and so is often represented using an *axis-angle* representation. Explain how this allows a 3D rotation to be represented using just 3 parameters. (3)

3. **Stereo and Motion**

(a) Explain, with a diagram, the concepts of *epipolar points* and *epipolar lines* in stereo geometry. (4)

(b) The stereo geometry between two images is captured by the *fundamental matrix*. Given corresponding points, x and x' in two views, the fundamental matrix relates them by the equation

$$x'Fx = 0.$$

(i) Given F and x, how can the corresponding epipolar line, ℓ', be computed? (1)

(ii) What is the relationship between ℓ' and x'? (1)

(c) A related quantity, the *essential matrix*, is defined by

$$E = K'^T F K,$$

where K and K' are the camera calibration matrices for the two images. The matrix E can be decomposed to give the rotation and translation (up to a scale) between the two cameras.

(i) This decomposition gives four potential solutions. Explain the geometric relationship between them. (4)

(ii) Explain how you can determine which of the four solutions is correct. (3)

(d) The 3D location of a point can be estimated by intersecting two rays.

(i) Explain, with the help of a diagram, how these two rays are formed. (3)

(ii) These to rays do not usually intersect exactly. One solution is to minimise the *reprojection error*. Explain what the term ‘reprojection error’ means. (2)

4. **Optimisation and Bundle Adjustment**

(a) Many of the estimation methods used in computer vision are very sensitive to *outliers*. One common solution to remove outliers is the RANSAC (Random Sample and Consensus) algorithm. The number of iterations required for RANSAC can be estimated by the formula

$$n = \frac{\log(p)}{\log(1 - \phi^s)},$$

where n is the number of iterations, p is the probability that we do not find a correct solution after n iterations, ϕ is the proportion of inliers in our measurements, and s is the sample size.

(i) Explain what an outlier is, and how they arise from feature matching. (3)

(ii) How do the values of s and ϕ affect the number of iterations required? (2)

(iii) In practice we do not know ϕ in advance. Explain how the value of ϕ can be estimated during the RANSAC process. (3)

(b) An important step in 3D reconstruction from images is *bundle adjustment*, a large non-linear least squares optimisation problem. This minimises the squared difference between some measured values and a prediction of the measurements based on a set parameters.

(i) What are the measurements and parameters in the case of bundle adjustment? (3)

(ii) Explain why this problem is non-linear. (3)

(c) The Levenberg-Marquardt algorithm for non-linear least squares optimisation uses a matrix of derivatives called the Jacobian. In the case of bundle adjustment, the Jacobian is a *sparse* matrix.

(i) What does it mean for a matrix to be sparse? (1)

(ii) Why is the Jacobian sparse in the case of bundle adjustment? (3)

(iii) What advantages can we gain by knowing that the Jacobian is sparse? (2)