Midterm Examination

CS 766: Computer Vision

November 9, 2006

LAST NAME: ________SOLUTION________________________

FIRST NAME: ______________________________________

Problem	**Score**	**Max Score**
1	________	10
2	________	14
3	________	12
4	________	13
5	________	16
6	________	15
Total	________	80

1. [10] **Camera Projection**

(a) [2] True or False: The orthographic projection of two parallel lines in the world must be parallel in the image.

```
True
```

(b) [3] Under what conditions will a line viewed with a pinhole camera have its vanishing point at infinity?

```
The line is in a plane parallel to the image plane.
```

(c) [5] A scene point at coordinates (400,600,1200) is perspectively projected into an image at coordinates (24,36), where both coordinates are given in millimeters in the camera coordinate frame and the camera's principal point is at coordinates (0,0,f) (i.e., $u_0 = 0$ and $v_0 = 0$). Assuming the aspect ratio of the pixels in the camera is 1, what is the focal length of the camera? (Note: the aspect ratio is defined as the ratio between the width and the height of a pixel; i.e., k_u/k_v.)

```
u = fx/z, so f = uz/x = 24 * 1200 / 400 = 72 mm.
```

2. [14] **Camera Calibration**

(a) [5] Show how the projection of a point in a planar scene at world coordinates (X, Y) to pixel coordinates (u, v) in an image plane can be represented using a *planar affine camera model*.

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ 0 & 0 & p_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

`or, equivalently since the overall scale of the above matrix does not matter,`

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

(b) [3] Under what conditions is the use of an affine transformation appropriate when viewing a planar scene?

`If the field of view of the scene plane is such that all points visible on the world plane are at approximately the same depth from the camera compared to the distance of the camera from the plane.`

(c) [3] How many degrees of freedom are there to solve for in (a), and what is the minimum number of calibration points needed to estimate the calibration parameters?

`There are 6 degrees of freedom since the overall scale does not matter.  Therefore, we need at least 3 points.`

(d) [3] What effects can a planar affine transformation have on parallel lines?

`Planar affine transformations preserve parallelism.`

3. [12] **Edge Detection**
Compare the Canny edge detector and the Laplacian-of-Gaussian (LoG) edge detector for each of the following questions.

(a) [3] Which of these operators is/are isotropic and which is/are non-isotropic?

LoG is isotropic and Canny is non-isotropic.

(b) [3] Describe each operator in terms of the order of the derivatives that it computes.

LoG is 2nd derivative and Canny is 1st derivative.

(c) [3] What parameters must be defined by the user for each operator?

LoG requires σ defining the scale of the Gaussian blurring. Canny requires σ and two thresholds for hysteresis.

(e) [3] Which detector is more likely to produce long, thin contours? Briefly explain.

Canny because of non-maximum suppression which thins and hyteresis thresholding which can fill in weak edge gaps.

4. [13] **Feature Detection and Description**

(a) [5] We want a method for *corner detection* for use with 3D images, i.e., there is an intensity value for each (x,y,z) voxel. Describe a generalization of either the Harris corner detector or the Tomasi-Kanade corner detector by giving the main steps of an algorithm, including a test to decide when a voxel is a corner point.

Similarly to the 2D case, we can estimate the average change in intensity for a shift (u,v,w) around a given voxel using a bilinear approximation given by

$$E(u,v,w) = [u\ v\ w]\ M \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

where M is a 3 x 3 matrix computed from partial derivatives in the 3 directions. Compute the 3 eigenvalues, λ_1, λ_2 and λ_3, of M, specifying an ellipsoid at the voxel that measures the variation in intensity in 3 orthogonal directions. Using the test in the Tomasi-Kanade detector, mark the voxel as a corner point if the smallest eigenvalue, λ_3, is greater than a threshold. Using the test in the Harris operator, mark the voxel as a corner point if $\lambda_1 \lambda_2 \lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_2)$ is greater than a threshold.

(b) [8] The *SIFT descriptor* is a popular method for describing selected feature points based on local neighborhood properties so that they can be matched reliably across images. Assuming feature points have been previously detected using the SIFT feature detector, (i) briefly describe the main steps of creating the SIFT feature *descriptor* at a given feature point, and (ii) name three (3) scene or image changes that the SIFT descriptor is invariant to (i.e., relatively insensitive to).

(i) At each point where a SIFT "keypoint" is detected, the descriptor is constructed by computing a set of 16 orientation histograms based on 4 x 4 windows within a 16 x 16 pixel neighborhood centered around the keypoint. At each pixel in the neighborhood, the gradient direction (quantized to 8 directions) is computed using a Gaussian with σ equal to 0.5 times the scale of the keypoint. The orientation histograms are computed relative to the orientation at the keypoint, with values weighted by the gradient magnitude of each pixel in the window. This results in a vector of 128 (= 16 x 8) feature values in the SIFT descriptor. (The values in the vector are also normalized to enhance invariance to illumination changes.)

(ii) Because the SIFT descriptor is based on edge orientation histograms, which are robust to contrast and brightness changes and are detected at different scales, the descriptor is translation, rotation, scale, and illumination (both intensity change by adding a constant and intensity change by contrast stretching) invariant. It is not invariant to significant viewpoint changes.

5. [16] **Hough Transform and RANSAC**

After running your favorite stereo algorithm assume you have produced a dense depth map such that for each pixel in the input image you have its associated scene point's (X, Y, Z) coordinates in the camera coordinate frame. Assume the image is of a scene that contains a single dominant plane (e.g., the front wall of a building) at unknown orientation, plus smaller numbers of other scene points (e.g., from trees, poles and a street) that are not part of this plane. As you know, the plane equation is given by $ax + by + cz + d = 0$.

(a) [8] Define a *Hough transform* based algorithm for detecting the orientation of the plane in the scene. That is, define the dimensions of your Hough space, a procedure for mapping the scene points (i.e., the (X, Y, Z) coordinates for each pixel) into this space, and how the plane's orientation is determined.

Assuming the plane is not allowed to pass through the camera coordinate frame origin, we can divide by d, resulting in three parameters, $A = a/d$, $B = b/d$, and $C = c/d$ that define a plane. Therefore the Hough parameter space is three dimensional corresponding to possible values of A, B, and C. Assuming we can bound the range of possible values of these three parameters, we then take each pixel's (X, Y, Z) coordinates and increment all points $H(p, q, r)$ in Hough space that satisfy $pX + qY + rZ + 1 = 0$. The point (or small region) in H that has the maximum number of votes determines the desired scene plane.

(b) [8] Describe how the *RANSAC algorithm* could be used to detect the orientation of the plane in the scene from the scene points.

Step 1: Randomly pick 3 pixels in the image and, using their (X, Y, Z) coordinates, compute the plane that is defined by these points.

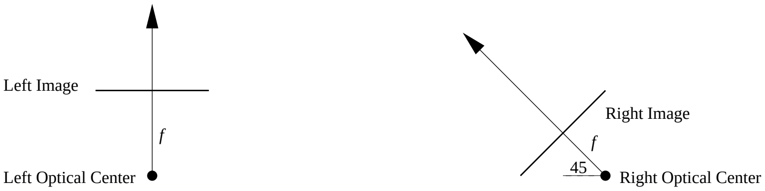
Step 2: For each of the remaining pixels in the image, compute the distance from its (X, Y, Z) position to the computed plane and, if it is within a threshold distance, increment a counter of the number of points (the "inliers") that agree with the hypothesized plane.

Step 3: Repeat Steps 1 and 2 many times, and then select the triple of points that has the largest count associated with it.

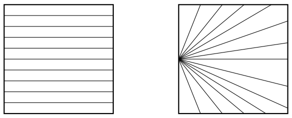
Step 4: Using the triple of points selected in Step 3 plus all of the other inlier points which contributed to the count, recompute the best planar fit to all of these points.

6. [15] **Epilolar Geometry and Stereo**

(a) [5] Consider the following top view of a stereo rig:



Draw the front views of the two 2D images and show (and clearly label) the approximate positions of the epipoles and epipolar lines for this configuration of cameras.



(b) [3] Given a conjugate pair of points, at pixel coordinates $p = (x_l, y_l)$ in the left image and at pixel coordinates $q = (x_r, y_r)$ in the right image of a stereo pair, give the equation that describes the relationship between these points when neither the intrinsic nor extrinsic parameters of the cameras are known. Also specify how many conjugate pairs of points are needed to solve for all of the unknowns in your equation.

$q^T Fp = 0$ where F is the 3 × 3 Fundamental matrix containing 8 degrees of freedom. Each correspondence generates one linear constraint on the elements of F; hence at least 8 conjugate pairs (and no noise) are needed to compute it using a linear algorithm such as the 8-point algorithm. (Note that by adding other constraints on the form of F, nonlinear techniques can be use to estimate F from 7 point correspondences.)

(c) [3] What is the *disparity gradient constraint* that is often used in solving the stereo correspondence problem.

Nearby image points should have correspondences that have similar disparities.

(d) [4] Yes or No: Can scene depth be recovered from a stereo pair of images taken under each of the following circumstances:

(i) [2] Two images produced by weak perspective projection.

No

(ii) [2] Two images taken by a single perspective projection camera that has been rotated about its optical center.

No