Midterm Examination

CS 294-6: Advanced Topics in Computer Vision

April 11, 2003

HONOR SYSTEM: This examination is strictly individual. You are now allowed to talk, discuss, exchange solutions, etc., with other fellow students. Furthermore, you are only allowed to use the book and your class notes. You may only ask questions to the class instructors. Any violation of the honor system, or any of the ethic regulations, will be immediately reported according to UC Berkeley regulations.

1. **(30 points) Motion Estimation from Paracatadioptric Cameras.** A paracatadioptric camera combines a paraboloidal mirror of focal length 1/2 and focus at the origin

$$Z = \frac{1}{2}(X^2 + Y^2 - 1) \tag{1}$$

with an orthographic lens. Therefore, the projection (image) $\boldsymbol{x} = (x, y, 0)^T$ of a 3-D point $q = (X, Y, Z)^T$ is obtained by intersecting a parameterized ray with the equation of the paraboloid to yield $\boldsymbol{b}$ (see Figure 1), and then orthographically projecting $\boldsymbol{b}$ onto the image plane $Z = 0$.

(a) **(3 points)** Show that the image of $q = (X, Y, Z)^T$ is given by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{-Z + \sqrt{X^2 + Y^2 + Z^2}} \begin{bmatrix} X \\ Y \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} X \\ Y \end{bmatrix}. \tag{2}$$

(b) **(3 points)** The back-projection ray $\boldsymbol{b} \in \mathbb{R}^3$ is a ray from the optical center in the direction of the 3D point $q \in \mathbb{R}^3$ being imaged (see Figure 1). Show that $\lambda \boldsymbol{b} = q$, where $\boldsymbol{b} = (x, y, z)^T$ and $z = (x^2 + y^2 - 1)/2$.

(c) **(3 points)** Given two views $\{(\boldsymbol{x}_1^i, \boldsymbol{x}_2^i)\}_{i=1}^N$ related by a discrete motion $(R, T) \in SE(3)$, can one apply the 8-point algorithm to the corresponding back-projection rays $\{(\boldsymbol{b}_1^i, \boldsymbol{b}_2^i)\}_{i=1}^N$ to compute R and T?

(d) **(8 points)** Assume that the camera undergoes a linear velocity $v \in \mathbb{R}^3$ and an angular velocity $\omega \in \mathbb{R}^3$, so that the coordinates of a static 3D point $q \in \mathbb{R}^3$ evolve in the camera frame as $\dot{q} = \widehat{\omega} q + v$. Show that the time derivative of the back-projection ray is given by

$$\dot{\boldsymbol{b}} = -(I + \boldsymbol{b} e_3^T)\widehat{\boldsymbol{b}}\omega + \frac{1}{\lambda}\left(I + \boldsymbol{b} e_3^T - \frac{\boldsymbol{b}\boldsymbol{b}^T}{1 + e_3^T \boldsymbol{b}}\right) v, \tag{3}$$

and therefore the paracatadioptric optical flow is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} xy & z - x^2 & -y \\ -(z - y^2) & -xy & x \end{bmatrix} \omega + \frac{1}{\lambda} \begin{bmatrix} 1 - \rho x^2 & -\rho xy & (1 - \rho z)x \\ -\rho xy & 1 - \rho y^2 & (1 - \rho z)y \end{bmatrix} v,$$

where $\rho = 1/(1 + z)$.

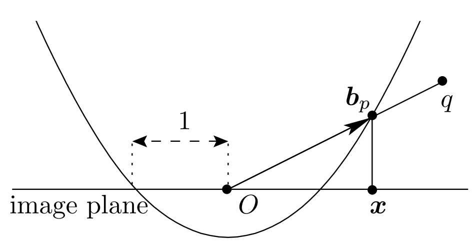

Figure 1: Showing the projection model for paracatadioptric cameras, and the back-projection ray $\boldsymbol{b}$ associated with image point $\boldsymbol{x}$.

(e) **(8 points)** Derive a factorization method for estimating the motion of the camera and the structure of the scene from multiple paracatadioptric views as follows. Let $(x_i, y_i)^T$, $i = 1, \dots, N$, be a pixel in the zeroth frame and let $(\dot{x}_{if}, \dot{y}_{if})^T$ be its optical flow in frame $f = 1, ..., F$, relative to the zeroth frame. Define the *optical flow* matrix $W \in \mathbb{R}^{2N \times F}$ as:

$$W = \begin{bmatrix} \dot{x}_{11} & \cdots & \dot{x}_{N1} & \dot{y}_{11} & \cdots & \dot{y}_{N1} \\ \vdots & & \vdots & \vdots & & \vdots \\ \dot{x}_{1F} & \cdots & \dot{x}_{NF} & \dot{y}_{1F} & \cdots & \dot{y}_{NF} \end{bmatrix}^T \tag{4}$$

and the matrix of *rotational flows* Ψ and the matrix of *translational flows* Φ as:

$$\Psi = \begin{bmatrix} \{xy\} & \{z - x^2\} & -\{y\} \\ -\{z - y^2\} & -\{xy\} & \{x\} \end{bmatrix} \in \mathbb{R}^{2N \times 3}, \quad \Phi = \begin{bmatrix} \{\frac{1-\rho x^2}{\lambda}\} & \{\frac{-\rho xy}{\lambda}\} & \{\frac{(1-\rho z)x}{\lambda}\} \\ \{\frac{-\rho xy}{\lambda}\} & \{\frac{1-\rho y^2}{\lambda}\} & \{\frac{(1-\rho z)y}{\lambda}\} \end{bmatrix} \in \mathbb{R}^{2N \times 3},$$

where, *e.g.*, $\{xy\} = (x_1 y_1, \cdots, x_N y_N)^T \in \mathbb{R}^N$. Show that the *optical flow* matrix $W \in \mathbb{R}^{2N \times F}$ satisfies

$$W = [\Psi \ \ \Phi]_{2N \times 6} \begin{bmatrix} \omega_1 & \cdots & \omega_F \\ v_1 & \cdots & v_F \end{bmatrix}_{6 \times F} = SM^T, \tag{5}$$

where ω_f and v_f are the rotational and linear velocities, respectively, of the object relative to the camera between the zeroth and the f-th frames. Show that $\text{rank}(W) \leq 6$. Under the additional assumption that $\text{rank}(W) = 6$, derive a factorization algorithm for estimating S, M and $\{\lambda\} \in \mathbb{R}^N$ from W.

2. **(20 points) Motion Estimation from Multiple Views of Multiple Line Features.** Following the development of Section 8.3.3, derive a multiple view factorization algorithm for the line case using the rank condition on M_l. The algorithm is in spirit similar to algorithm 8.1. for point features, with the main difference being initialization. In particular, answer the following questions

   (a) **(3 points)** Given M_l describe how to compute the distance and the direction of each line $\boldsymbol{l}^j$ assuming known motions R_i, T_i and hence known M_l.

   (b) **(3 points)** Given the known 3-D line parameters show how to estimate $[R_i, T_i]$ for $i = 1, \dots n$.

   (c) **(3 points)** Integrate the above steps into an overall factorization based algorithm.

   (d) **(3 points)** How many lines in general position are needed? Why a minimum of three line features is needed?

   (e) **(8 points)** Show how to initialize the algorithm from three views as follows.

      i. Consider the trilinear constraint in (8.52)

      $$(\boldsymbol{l}_2^T R_2 T_3^T \boldsymbol{l}_3 - \boldsymbol{l}_3^T R_3 T_2^T \boldsymbol{l}_2)\widehat{\boldsymbol{l}}_1 = 0$$

      and show that it can be re-written as

      $$\left(\boldsymbol{l}_2^T G_1 \boldsymbol{l}_3 \quad \boldsymbol{l}_2^T G_2 \boldsymbol{l}_3 \quad \boldsymbol{l}_2^T G_3 \boldsymbol{l}_3\right) \widehat{\boldsymbol{l}}_1 = 0 \tag{6}$$

      where $G_1 = \boldsymbol{r}_2^1 T_3^T - T_2 \boldsymbol{r}_3^{1T} \in \mathbb{R}^{3\times3}$, $G_2 = \boldsymbol{r}_2^2 T_3^T - T_2 \boldsymbol{r}_3^{2T} \in \mathbb{R}^{3\times3}$ and $G_3 = \boldsymbol{r}_2^3 T_3^T - T_2 \boldsymbol{r}_3^{3T} \in \mathbb{R}^{3\times3}$, with $R_2 = [\boldsymbol{r}_2^1 \ \boldsymbol{r}_2^2 \ \boldsymbol{r}_2^3]$ and $R_3 = [\boldsymbol{r}_3^1 \ \boldsymbol{r}_3^2 \ \boldsymbol{r}_3^3]$.

      ii. Show that one can solve linearly for the 27 unknowns in G_1, G_2 and G_3 up to a scale factor from 13 line features using (6).

      iii. Show that $G_1^T \widehat{T_2} \boldsymbol{r}_2^1 = G_2^T \widehat{T_2} \boldsymbol{r}_2^2 = G_3^T \widehat{T_2} \boldsymbol{r}_2^3 = 0$. Thus, assuming that G_1, G_2 and G_3 are rank 2 matrices, the following matrix is known (with each column up to a scale factor) $H_2 = [\widehat{T_2} \boldsymbol{r}_2^1 \ \widehat{T_2} \boldsymbol{r}_2^2 \ \widehat{T_2} \boldsymbol{r}_2^3]$.

      iv. Show that the range of H_2 is the same as the range of the essential matrix $E_2 = \widehat{T_2} R_2$ and that one can obtain E_2 from the SVD of H_2. Outline a similar procedure to obtain $E_3 = \widehat{T_3} R_3$ from the right null space of G_1, G_2 and G_3.

      v. Show that one can now recover (R_2, T_2) and (R_3, T_3) using the last step in the 8-point algorithm.

3. **(20 points) Segmentation of Planar Motions.** Let $\{(\boldsymbol{x}_j, \boldsymbol{y}_j) \in \mathbb{R}^2 \times \mathbb{R}^2\}_{j=1}^N$ be a given set of corresponding points in the XY plane. Each point $\boldsymbol{y}_j$ is obtained by transforming $\boldsymbol{x}_j$ according to one of the following n planar motions $\{\mathcal{M}_i : \mathbb{R}^2 \to \mathbb{R}^2\}_{i=1}^n$

$$\boldsymbol{x} \mapsto f_i \left[\begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} u_i \\ v_i \end{pmatrix} \right]. \tag{7}$$

Each motion represents a rotation along Z by an amount $\theta_i \in \mathbb{S}^1$, a translation in the XY plane by an amount $(u_i, v_i)^T \in \mathbb{R}^2$ and scaling by a factor $f_i \in \mathbb{R}$. Let $\boldsymbol{z}_1^j \in \mathbb{C}$ and $\boldsymbol{z}_2^j \in \mathbb{C}$ be the complex representation of $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$, respectively, for $j = 1, \ldots, N$.

   (a) **(6 points)** Write (7) as a single equation relating $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ in which the original motion parameters $\{(f_i, \theta_i, u_i, v_i)\}_{i=1}^n$ become $R_i = f_i \exp(\sqrt{-1}\theta_i) \in \mathbb{C}$ and $T_i = f_i(u_i + \sqrt{-1}v_i) \in \mathbb{C}$.

   (b) **(3 points)** Show that the estimation of $\{(R_i, T_i)\}_{i=1}^n$ is a GPCA problem with complex data. What is the value of K? What are the data vectors? What are the normal vectors?

   (c) **(5 points)** Is it possible to solve this GPCA problem with complex data the same way (with just the obvious modifications) as GPCA for real data? If yes, why? If no, how can standard GPCA be modified to deal with complex data?

   (d) **(3 points)** Following standard GPCA (or the modified version in (c)), derive a formula for the number of motions n and briefly say how to estimate the motion parameters $\{(R_i, T_i)\}_{i=1}^n$.

   (e) **(3 points)** Show how to recover the original motion parameters $\{(f_i, \theta_i, u_i, v_i)\}_{i=1}^n$.

4. **(30 points) Experimental Evaluation of the 8-point Algorithm.** In this problem, you are asked to test the performance of the 8-point algorithm for different levels of noise, baseline, depth variation, and field of view.

   (a) **(6 points)** Write a function `X = points(N,fov,Zmin,Zmax)` that generates a cloud of N points uniformly distributed in front of the camera in a truncated pyramid specified by the field of view and the depth variation, as illustrated in Figure 2. Also write a function `x = project(X)` that projects a cloud of points $X \in \mathbb{R}^{3\times N}$ onto their perspective images $\boldsymbol{x} \in \mathbb{R}^{3\times N}$, without using a for loop.

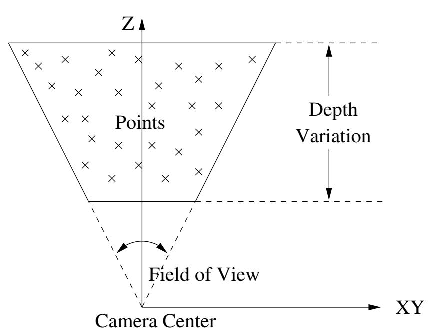

Figure 2: Truncated pyramid used to generate the structure.

   (b) **(6 points)** Generate $N = 20$ with a FOV of $60°$ and a depth variation from $Z_{\min} = 1$ to $Z_{\max} = 20$ units of focal length. Project those 3D points onto two perspective images related by motion (R, T), where R is a randomly chosen rotation of $20°$ and T is a randomly chosen translation with norm $\|T\| = \tau Z_{\min}$, where $\tau = 1$. Add zero-mean Gaussian noise with standard deviation $\sigma = 0, 0.5, 1, 1.5, 2$ pixels to the image data. When adding noise, use a calibration matrix of `K = [500 0 250; 0 500 250; 0 0 1]`. For each noise level, run 100 trials of the 8-point algorithm (*i.e.* 100 randomly chosen 3D points and motions) and compute the estimated rotation $\tilde{R}$ and translation $\tilde{T}$. For each trial compute the error in rotation and translation as

$$\text{Rot. error} = \text{acos}\Big(\frac{\text{trace}(R\tilde{R}^T) - 1}{2}\Big) \qquad \text{(degrees)}. \tag{8}$$

$$\text{Trans. error} = \text{acos}\Big(\frac{T^T\tilde{T}}{\|T\|\|\tilde{T}\|}\Big) \qquad \text{(degrees)}. \tag{9}$$

Plot of the mean error (over the trials) for both rotation and translation as a function of noise. How does the error behave as a function of the noise level?

(c) **(6 points)** For $\sigma = 1$, plot the mean errors as a function of the baseline $\tau = 0.1, 0.4, 0.7, 1.0, 1.3$. What is the effect of τ in the performance of the algorithm? Explain.

(d) **(6 points)** For $\sigma = 1$, and $\tau = 1$ plot the mean errors as a function of the depth variation $Z_{\max} = 1.1, 5, 10, 15, 20, 25$. What is the effect of $Z_{\max}$ in the performance of the algorithm? Explain.

(e) **(6 points)** For $\sigma = 1$, $\tau = 1$ and $Z_{\max} = 20$ plot the mean errors as a function of field of view $FOV = 20, 40, 60, 80, 100°$. What is the effect of FOV in the performance of the algorithm? Explain.

5. **(10 points) Calibration with Partial Knowledge of the Structure and K.** Consider a camera with the calibration matrix

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where the only unknown parameter is the focal length f. Assume that the image of the center of projection is known, the skew is zero and the aspect ratio is 1. Suppose you have a single view of the rectangular planar structure, whose four end points in the world coordinate frame have following coordinates $\boldsymbol{X}_1 = [0, 0, 0, 1]^T$, $\boldsymbol{X}_2 = [\alpha b, 0, 0, 1]^T$, $\boldsymbol{X}_2 = [0, b, 0, 1]^T$, $\boldsymbol{X}_4 = [\alpha b, b, 0, 1]^T$, where one of the dimension of the plane b as well as the ratio α between the two sides of the rectangle are unknown.

(a) Write down the projection equation for this special case relating the 3-D coordinates of the planar points to their image projections.

(b) Show that the image coordinates $\boldsymbol{x}$ and the 3-D coordinates of points on the world plane are in fact related by a 3×3 homography matrix of the following form

$$\lambda \boldsymbol{x} = H[X, Y, 1]^T.$$

Write down the explicit form of H in terms of camera pose R, T and the intrinsic parameters of the camera H.

(c) Assuming the known structure (up to scales α, b) describe an algorithm for recovering the unknown homography H.

(d) Given H describe steps which would enable you to factor it and recover the unknown focal length f and rotation R. Also recover translation T and ratio α up to a universal scale factor.

6. **(5 points) 6-point Algorithm for the Recovery of Fundamental (Essential) Matrix.** The relationship between homography and Fundamental (Essential) matrix suggest a simple alternative algorithm for recovery of the fundamental matrix (section 5.3.4). Outlines the steps of the algorithm by assuming that you have available correspondences between at least 4 planar points and at least two points which do not lie in the plane.