2019 Solution to CA1

1a. **Ans: (20 marks)**
First, recall that line is difficult to represent using homogeneous coordinates in the 3D space, so you might want to stick with inhomogeneous coordinates throughout and perform the derivation. (It is still possible to use homogeneous coordinates though; see the answer to Q3c (i))

A line in 3D space can be represented by the following parametric representation:

$$\mathbf{l_i} = \mathbf{p_i} + \lambda_i \mathbf{v_i}$$

where the subscript **i** is for the i^{th} line in the *family* of parallel lines. These parallel lines meet at the point of infinity along **l** (by letting λ go to infinity). The projection of this point is the point of intersection on the image.

Formally, the point of infinity along $\mathbf{l}_i$ is given by $\lim_{\lambda i \to \infty}$ $\mathbf{l_i} = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}_C + \lambda_i \begin{bmatrix} a \\ b \\ c \end{bmatrix}$

Thus the projection is given by

$x = \lim_{\lambda_i \to \infty} f \dfrac{X_i + \lambda_i a}{Z_i + \lambda_i c} = f \dfrac{a}{c}$, and similarly, y = $f \dfrac{b}{c}$.

(b) Expand (P'=RP-T), we have, for a particular corresponding pair (X,Y,Z) and (X',Y',Z'):

r11 X + r21 Y + r31 Z + t1 = X'
r12 X + r22 Y + r32 Z + t2 = Y'
r13 X + r23 Y + r33 Z + t3 = Z'

This can be written in the form

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & Y & Z & 1 \\ : \\ : \end{bmatrix} \begin{bmatrix} r11 \\ r12 \\ r13 \\ t1 \\ r21 \\ r22 \\ r23 \\ t2 \\ r31 \\ r32 \\ r33 \\ t3 \end{bmatrix} = \begin{bmatrix} X' \\ Y' \\ Z' \\ : \end{bmatrix}$$

(c) The correct transformation matrix from pts_prime.txt to pts.txt is

1	0	0	0
0	0	1	-4
0	-1	0	4
0	0	0	1.

However, since some Gaussian noise is added, the estimated matrix will be

2019 Solution to CA1

$$\begin{matrix} 0.9782 & -0.0084 & 0.0094 & -0.0365 \\ 0.0202 & -0.0046 & 0.9979 & -4.0009 \\ -0.0016 & -0.9940 & -0.0007 & 4.0021 \\ 0 & 0 & 0 & 1.0000. \end{matrix}$$

The problem of this estimate is the rotation part (the 1^{st} 3x3 submatrix) is not an orthogonal matrix (we did not impose any orthogonality constraint in our estimation process), thus the determinant of the matrix will be expected to be deviating from 1.

(d) i. ox=oy=0; no sign change between camera and image coordinate, so the intrinsic matrix is given by

$$M_{\text{int}} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

Using $\boldsymbol{P_c} = \boldsymbol{RP_w} + \boldsymbol{t}$, the rigid transformation is given by first a rotation about the world X axis, then followed by a translation of h/sinα along the rotated Z axis (***R*** is the orientation of the world with respect to the camera, i.e. <u>the column of ***R*** is the axis of the world expressed in the camera frame</u>; ***t*** is the world origin expressed in the camera frame):

$$M_{ext} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(90+\alpha) & -\sin(90+\alpha) & 0 \\ 0 & \sin(90+\alpha) & \cos(90+\alpha) & h/\sin\alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin\alpha & -\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & h/\sin\alpha \end{bmatrix}$$

Overall, M is given by:

$$M = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & -f\sin\alpha & -f\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & h/\sin\alpha \end{bmatrix}$$

You could obtain the same solution for M using the $\boldsymbol{P_c} = \boldsymbol{RP_w} - \boldsymbol{Rt}$ in the textbook, in which case ***t*** should be $[0, -h/\tan\alpha, h]^T$. (NB: in this case, ***t*** will be the camera origin expressed in the world frame)

(ii) Let's denote sinα by S and cosα by C. Points on this plane is given by (X,Y,d,1). Multiply with M gives us the image point (fX, -fSY-fCd, CY-Sd+h/S). After some manipulation, this can be written as a 3x3 linear transformation D:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & -fS & -fCd \\ 0 & C & -Sd+h/S \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ W_W \end{bmatrix} \text{ -- (1)}$$

where it is clear that $D = \begin{bmatrix} f & 0 & 0 \\ 0 & -fS & -fCd \\ 0 & C & -Sd+h/S \end{bmatrix}$

(Comment: The whole idea of using a matrix to represent the transformation is that the matrix part should not contain variables like X_W, Y_W. However, since Z_W =d is a constant now, your task is to absorb it into the matrix part.)

2019 Solution to CA1

Q2 (20 marks)

b. Singular values decrease dramatically.

d. As k increases, the quality of the corresponding image improves. Since the singular values decrease exponentially, the first few singular values and their corresponding eigenvectors approximate the original image well.

e. For this image, it is not worth transmitting the K=100 approximation, as doing so would require transmitting the following: 261 * K + 161 * K + K =42300 values, whereas the original image has only 261*161=42021 values. There is no compression achieved while some details are possibly lost.

f. We have A=U*S*V'. Each pixel of A can be expressed as $a_{ij} = \sum_{r=1}^{n} \sigma_r u_{ir} v_{jr}$; this is the full rank version of A. The rank-k version of A is $a_{ij}{}^{k} = \sum_{r=1}^{k} \sigma_r u_{ir} v_{jr}$. Thus the per-pixel error is:

$e_{ij} = a_{ij} - a_{ij}{}^{k} = \sum_{r=k+1}^{n} \sigma_r u_{ir} v_{jr} \le \sum_{r=k+1}^{n} \sigma_r u_{\max} v_{\max} \le u_{\max} v_{\max} \sum_{r=k+1}^{n} \sigma_r$. We see that the per-pixel error is bounded from the above by a multiple of the sum of the 'remaining' singular values.

g. Intuitively, if there is very little or slow changes to the images throughout the sequence, you will expect that the rank of the matrix you have constructed to be small (in the perfectly stationary case, rank=1). Conversely, fast changes/motions will contribute to a large increase in rank. Thus, when you take only the first few principal components, you are essentially filtering out the faster changes/motions. For e.g., you will find that the motion of clouds (low frequency) is preserved, however, motion of airplanes (high frequency) are severely blurred.

Comment 1: The main impact of the "compression" is its effect on the dynamics of the video, not so much in details of individual still image.

Comment 2: For those who experienced out-of-memory problem, you can try the following:

a. use im2single instead of im2double, which will reduce the size of the image by half;
b. try [U S V] = svd(A,'econ'), which will reduce the size of U, S, and V.
c. In case your problem is still not solved, you can resize the original image to a smaller size to reduce the burden of svd.

NB: The mean-centering operation should be done in such a way that each variable (here a pixel) has mean = 0 across time. In other words, you compute a mean image (from averaging over all images), then subtract the mean image from each image.

Q3 (24 marks + 3 bonus)

a.

2019 Solution to CA1

$$P_A = \begin{bmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ where m_{ij} is the elements of A and t_i is that of b.

To show that the point at infinity in space $(X,Y,Z,0)^T$ is mapped to point of infinity in the image plane: show that its image point is $(x,y,0)^T$, which is the point at infinity in the image plane.

The projections of parallel lines in space onto the image plane are also parallel.

b.

Any point on $l0$ can be expressed as $(\cos\alpha, \cos\beta, \cos\gamma, r)$ in homogeneous coordinates. When $r \to 0$, $l0$ reaches the infinite point. Applying the camera projection to this infinite point, we obtain the vanishing point:

$$
\begin{aligned}
v &= \lim_{r\to 0} \mathbf{K}[\mathbf{I}_3 \mid \mathbf{0}_3]\begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \\ r \end{bmatrix} \\
&= \mathbf{K}[\mathbf{I}_3 \mid \mathbf{0}_3]\begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & \mathbf{1} \end{bmatrix}\begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \\ 0 \end{bmatrix} \\
&= \mathbf{K}[\mathbf{I}_3 \mid \mathbf{0}_3]\begin{bmatrix} \mathbf{Rd} \\ 0 \end{bmatrix} \\
&= \mathbf{K}\mathbf{I}_3\mathbf{Rd} \\
&= \mathbf{KRd}
\end{aligned}
$$

K & **R** are non-singular. Hence $\mathbf{d} = \mathbf{R}^{-1}\mathbf{K}^{-1}\mathbf{v}$

To obtain the orthogonality constraint, we have $\mathbf{Rd} = \mathbf{K}^{-1}\mathbf{v}$

$$
\begin{aligned}
\mathbf{d}_i \cdot \mathbf{d}_j &= 0 \Rightarrow \\
\mathbf{d}_i^{\mathrm{T}}\mathbf{d}_j &= 0 \Rightarrow \\
\mathbf{d}_i^{\mathrm{T}}\left(\mathbf{R}^{\mathrm{T}}\mathbf{R}\right)\mathbf{d}_j &= 0 \Rightarrow \\
\left(\mathbf{Rd}_i\right)^{\mathrm{T}}\left(\mathbf{Rd}_j\right) &= 0 \Rightarrow
\end{aligned}
$$

Substituting:

$$\mathbf{v_i}^{\mathrm{T}}\,\mathbf{K}^{-\mathrm{T}}\,\mathbf{K}^{-1}\mathbf{v_j} = \mathbf{0}$$

Bonus part:

A “simple” way is to linearize the above. Specifically, let $\mathbf{G} = \mathbf{K}^{-\mathrm{T}}\,\mathbf{K}^{-1}$, and solve for **G** via linear least squares (ignoring the specific form of $\mathbf{K}^{-\mathrm{T}}\,\mathbf{K}^{-1}$ which **G** must satisfy). Then invert the estimated **G** to obtain $\mathbf{G}^{-1}$. Since $\mathbf{G}^{-1} = \mathbf{K}\,\mathbf{K}^{\mathrm{T}}$, we can perform a Cholesky factorization on $\mathbf{G}^{-1}$ to obtain **K** (if you have not heard of Cholesky factorization, you can do look up its wiki page).

A more complex scheme, taking into account the nonlinear nature of the equation $\mathbf{v_i}^{\mathrm{T}}\,\mathbf{K}^{-\mathrm{T}}\,\mathbf{K}^{-1}\mathbf{v_j} = \mathbf{0}$, would result in a convex optimization problem (specifically, a positive semi-definite programming problem), which is beyond the scope of this course.

c.

2019 Solution to CA1

i. A line in 3D space can be represented by the following parametric representation:

$$\mathbf{l_i} = \mathbf{p_i} + \lambda_i \mathbf{v_i}$$

where the subscript **i** is for the i^{th} line in the *family* of parallel lines. $\mathbf{p_i}$ and $\mathbf{v_i}$ are the position vector of a point that lies on this line and the direction of the particular line respectively. Since we are considering only parallel lines, the directions for this family should be the same and hence all the lines have the *same* $\mathbf{v_i}$ for different lines. λ_i is a scalar that determine the *length* of this line. Rewriting the above representation with this condition and denoting the direction vector explicitly with its elements we have:

$$\mathbf{l_i} = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}_C + \lambda_i \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Parallel lines in this family that shares the same direction vector $\mathbf{v_i}$ will meet at infinity. The concept of a *point at infinity* cannot be clearly defined in $\Re^3$ *Euclidean* space and hence we will represent such points in P^4 *projective* space instead. Using the 4^{th} homogenous coordinate, we represent such points at infinity with a 0 with the 3 other coordinates to represent in which direction is the point found in $\Re^3$. In our example, it is obviously in the common direction of the family of parallel lines. We can thus represent the point at infinity for this family with the homogenous coordinate representation:

$$\tilde{\mathbf{p}}_\infty = \begin{bmatrix} a & b & c & 0 \end{bmatrix}^T$$

For the given set of parallel lines, $\tilde{\mathbf{p}}_\infty = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$

Projecting this point at infinity to the image plane (i.e. multiply by the projection matrix) and converting back to actual 2D coordinates of this image point, we have $(p_{11}/p_{31}, p_{21}/p_{31})$.

No, an infinite line in 3D space does not always yield an infinite line in the 2D image plane, since $(p_{11}/p_{31}, p_{21}/p_{31})$ can be finite.

ii.
It is the image of the origin of the world reference frame (0,0,0,1). Or you may say that it is the image point where the translation vector T (relating the world and camera reference frame) cuts the image plane.

iii.
It is the camera center expressed in the world frame represented as a homogeneous 4-vector. This is the unique point for which its image is not well defined (since LOS is not defined here).

(d): The homogeneous form of the conic is $ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$. Expressing in matrix form $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$, we obtain $\mathbf{C}$ as

2019 Solution to CA1

$$\begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix}$$

The vector $\mathbf{x} = \mathbf{l} \times \mathbf{m}$ is the null vector of $\mathbf{C}$ because
$\mathbf{Cx} = (\mathbf{l}\,\mathbf{m}^T + \mathbf{m}\,\mathbf{l}^T)(\mathbf{l} \times \mathbf{m}) = \mathbf{l}\,\mathbf{m}^T(\mathbf{l} \times \mathbf{m}) + \mathbf{m}\,\mathbf{l}^T(\mathbf{l} \times \mathbf{m})$
$= \mathbf{l}\,[\mathbf{m}^T(\mathbf{l} \times \mathbf{m})] + \mathbf{m}\,[\mathbf{l}^T(\mathbf{l} \times \mathbf{m})]$
but $\mathbf{l} \times \mathbf{m}$ is perpendicular to both $\mathbf{l}$ and $\mathbf{m}$, and thus the above evaluates to $\mathbf{Cx} = 0$. QED.
Points on $\mathbf{l}$ satisfy $\mathbf{l}^T\mathbf{x} = 0$; and are on the conic since $\mathbf{x}^T\mathbf{C}\,\mathbf{x} = (\mathbf{x}^T\mathbf{l})\,\mathbf{m}^T\mathbf{x} + \mathbf{x}^T\mathbf{m}\,(\mathbf{l}^T\mathbf{x}) = 0$.
Similarly, points satisfying $\mathbf{m}^T\mathbf{x} = 0$ also satisfy $\mathbf{x}^T\mathbf{C}\,\mathbf{x} = 0$. (note degenerate conic doesn't mean there is no 2nd order terms because we multiply two 1st order factors (representing line equations) together.)

e. It is a conic with parameters (up to an arbitrary scale factor)

0.00954056094243955 % a
0.00333960309258388 % b
0.00705787126602786 % c
−0.118534068061405 % d
−0.105429838562329 % e
−0.987259963257305 % f

Q4. (24 marks + 4 bonus)

4.3 What to submit

(1) Points on corners, prominent features are chosen as corresponding points because they are easy to locate.

(2) One possible T to normalize the data is $T = \begin{bmatrix} 1/s_x & 0 & -m_x/s_x \\ 0 & 1/s_y & -m_y/s_y \\ 0 & 0 & 1 \end{bmatrix}$

(4) When repeating the computation with different set of points, or different n, we may find the results of F differ from each other. This inaccuracy may be due to the following reasons.

i) error in locating the corresponding points.

ii) error measure is not properly normalized. Thus the residual that we minimized does not have proper geometrical meaning.

iii) scene structure in view is near a planar scene, resulting in ambiguity. This will be aggravated by improper choice of corresponding points that clustered in a small region (closer to a plane).

iv) error introduced due to the linear least square method, ignoring nonlinear constraint like the rank 2 constraint. Imposing rank 2 constraint afterward relieves but does not remove the problem.

For more robust estimation, large numbers of corresponding points may be used and methods like RANSAC may be implemented .

Q5 Motion Perception (12 marks)

2019 Solution to CA1

(a) The edge motions (e.g., 1) are ambiguous, due to the aperture problem, whereas the corner motions (e.g., 2) are unambiguous. The T-junction motions (e.g., 3) are also unambiguous, but their motion is spurious because the corner is not a genuine physical corner.

(b) The arrows indicate the perceived direction of motion of the barber-pole. Apparently, the visual system seems to know that the line terminators aligned with the occluders are not created by the end of the line itself but rather a result of occlusion by another surface. Therefore the motion signals of these line terminations tend to become suppressed and have less influence on the perceived motion. Thus the perceived motion tends to be biased in the direction orthogonal to the occlusion boundary.

(c) The change in perceived motion that occurs from Figure c to d is easy to explain in terms of junctions. In Figure c, T-junctions are formed where the occluders overlap the crossbars, and offer a plausible cue that the motions of the bar endpoints are spurious and should be discounted. One could suppose that the motions of the bar endpoints are simply ignored by the visual system when the occluders generate T-junctions at those locations. (Most of you didn't explain why these "unique points" are ignored by the visual system). When the endpoints are suppressed, all of the remaining local motions (of the bar edges and intersection) are consistent with a single circular motion, which is what is seen. Without the occluders and the T-junctions they produce, the endpoint motions are not ignored, and two motions, one for each bar, are necessary to explain the image data.

(d) **Ax**=**b** should be used. In this case, the degree 0 term of the conic (i,e., f in Question 3e) is not an unknown. It is the flow measurement (u and v) and thus should be shifted to the right hand side as **b**. If we want to write in the **Ax**=**0** form, it would require us to have the constant 1 in the unknown vectors **x**, which is inappropriate.