I. Theory Questions

1.1 Camera Models

(a)

Since world coordinate system is the same as the camera coordinate system, $\begin{matrix} X_w \\ Y_w \\ Z_w \\ 1 \end{matrix} = \begin{matrix} X_c \\ Y_c \\ Z_c \\ 1 \end{matrix}$. So extrinsic parameter matrix is an identity matrix. Then, camera model is as follows,

$$\begin{matrix} u_i \\ v_i \\ 1 \end{matrix} = \begin{matrix} K_{11} & K_{12} & K_{13} \\ 0 & K_{22} & K_{23} \\ 0 & 0 & K_{33} \end{matrix} \ \mathrm{X} \ \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix} \ \mathrm{X} \ \begin{matrix} x_i \\ y_i \\ z_i \\ 1 \end{matrix}$$

$$\begin{matrix} u_i \\ v_i \\ 1 \end{matrix} = \begin{matrix} K_{11} & K_{12} & K_{13} & 0 \\ 0 & K_{22} & K_{23} & 0 \\ 0 & 0 & K_{33} & 0 \end{matrix} \ \mathrm{X} \ \begin{matrix} x_i \\ y_i \\ z_i \\ 1 \end{matrix}$$

$$u_i = \frac{K_{11}x_i + K_{12}y_i + K_{13}z_i}{K_{33}z_i}$$

$$v_i = \frac{K_{22}y_i + K_{23}z_i}{K_{33}z_i}$$

$$u_i K_{33} z_i = K_{11}x_i + K_{12}y_i + K_{13}z_i$$

$$v_i K_{33} z_i = K_{22}y_i + K_{23}z_i$$

$$\begin{matrix} x_i & y_i & z_i & 0 & 0 & -u_i z_i \\ 0 & 0 & 0 & y_i & z_i & -v_i z_i \end{matrix} \ \mathrm{X} \ = \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix}$$

Therefore,

A is a 200 x 6 matrix,

$$\begin{matrix} x_1 & y_1 & z_1 & 0 & 0 & -u_1 z_1 \\ 0 & 0 & 0 & y_1 & z_1 & -v_1 z_1 \\ & & \vdots & & \vdots & \end{matrix}$$

$$\begin{matrix} x_{100} & y_{100} & z_{100} & 0 & 0 & -u_{100}z_{100} \\ 0 & 0 & 0 & y_{100} & z_{100} & -v_{100}z_{100} \end{matrix}$$

x is 6 x 1 matrix,

$$\begin{matrix} K_{11} \\ K_{12} \\ K_{13} \\ K_{22} \\ K_{23} \\ K_{33} \end{matrix}$$

(b)

We could use direct linear transformation algorithm, try to minimize least squares $||Ax||^2$ subject to $x^T x = 1$, noting that $K_{33} = 1$,

$$L(t) = x^T A^T A x - \lambda(x^T x - 1)$$

$$A^T A\hat{x} - \lambda\hat{x} = 0$$

Thus λ is an eigenvalue of $A^T A$, the solution x is its eigenvector.

1.2 Epipolar Geometry

(a)

For the first camera, camera coordinate system is equivalent to the world coordinate system, therefore,

$$M = K[I \ \ 0]$$

For the second camera, we need to apply rotation and translation, therefore

$$M' = K'[R \ \ t]$$

(b)

Expression 1.

Epipoles are the image of the other camera centers on the other camera's image plane. Note that $X' = RX + t$, then $X = R^{-1}X' - R^{-1}t$. Threfore,

$$e = P[-R^T t \ \ 1]^T = KR^T t$$

$$e' = P'[0 \ \ 1]^T = K't$$

Expression 2.

First, we need to define fundamental matrix, F. We can note $p_c = K^{-1}p$ and $p_c' = K'^{-1}p'$ defined in each camera's coordinate system. From essential matrix E , $p_c'^T E p_c = p_c'^T [t]_\times R p_c = p'^T K^{-1^T} [t]_\times R K^{-1} p = 0$. Thus, we can define $F = K^{-1^T}[t]_\times RK^{-1}$.

Then, from $p'^T F p = 0$, we can say $l' = Fp$ denotes the epipolar line in the image of second camera, since l' is orthogonal to p'. Similarly, for $p^T F^T p' = 0$, $l = F^T p'$ denotes the epipolar line in the image of first camera, since l is orthogonal to p. For any point in the image of second camera p', corresponding epipolar line in the image of first camera l contains the epipole e. Therefore $e^T F p' = 0$ for all p', $e^T F^T = 0$, $Fe = 0$. Similarly, $e'^T F = 0$. Thus, epipoles e and e' can be expressed as $K^{-1^T}[t]_\times RK^{-1}e = 0$ and $e'^T K^{-1^T}[t]_\times RK^{-1} = 0$.

(c)

$e' \times x'$

$= [e']_\times x'$

$= [e']_\times K'[R \ \ t]X$

$= [e']_\times K'[R \ \ 0]X$ ($e' = K't$ and cross-product with oneself is zero)

$= [e']_\times K'R[I \ \ 0]X$

$= [e']_\times K'RK^{-1}K[I \ \ 0]X$

$= [e']_\times K'RK^{-1}x$

(d)

Since e' and x' are homogeneous coordinates, $e' \times x'$ denotes a line passing through two points e' and x', which is geometrically l'.

1.3 Optical Flow

(a)

One example can be aperture problem in barber pole illusion. This happens because in a optical flow equation $I_x u + I_y v + I_t = 0$, there are three knowns(I_x, I_y, I_t) but two unknowns($u,\ v$). Thus velocity components $u,\ v$ cannot be found uniquely so we could not recover either horizontal or vertical components of motion.

(b)

If we focus on motion of not one point but multiple points, we can always recover horizontal or vertical components of motion. For example, if we focus on three points, we have three optical flow equation $I_x u + I_y v + I_t = 0$, with three knowns(I_x, I_y, I_t) and two unknowns($u,\ v$), thus we can use least square methods to get the velocities uniquely.

Else, if we use area-based methods with various gradients with large magnitudes, we can get the velocities uniquely.

(c)

If three assumptions of optical flow still hold, which are brightness consistency, spatial coherence, temporal persistence, optical flow equation will still hold while the motion of the camera is in the forward direction. That is because optical flow catches relative motion between the camera(observer) and the scene. Thus, if we know velocity components of camera, we can simply subtract it from the observed velocity components of the points and figure out which scene point the object is moving towards.

(d)

Define velocity components $u = \frac{\Delta x}{\Delta \mathrm{t}}$, $v = \frac{\Delta y}{\Delta \mathrm{t}}$, $w = \frac{\Delta z}{\Delta \mathrm{t}}$ and projection of (x, y, z) in 3D scene on camera image plane (x', y') as $x' = f\frac{x}{z}$, $y' = f\frac{y}{z}$ Since each point has same velocity components, after t, $x' = f\frac{x+u\mathrm{t}}{z+wt}$ $y' = f\frac{y+v\mathrm{t}}{z+wt}$ When $t \to \infty$, we can see that x', y' converges to $x' = f\frac{u}{w}$ $y' = f\frac{v}{w}$ thus we get a converging fixed point, which is a focus of expansion.

II. Programming

2.1 Dominant Motion Estimation

I implemented Lucas-Kanade Method, so the basic assumptions are equal to it, which are as follows.

- Brightness consistency (color consistency)
- Spatial coherence
- Temporal persistence (small motion)

Also, other trivial assumptions added are

- Majority of the pixels correspond to stationary objects in the scene whose depth variation is small relative to their distance from the camera

Thus, we use the entire img1 as template the template to be tracked in img2, which is taken Δt after. The parameters I used therefore includes,

- ◆ img1 : image taken at time t. Used as a template.
- ◆ img2 : image taken at time $t + \Delta t$
- ◆ p : a vector containing parameters of affine transformation matrix
- ◆ Gx : image gradient in x axis
- ◆ Gy : image gradient in y axis

The specific algorithm is as follows

1. W : Warp two vertexes (0,0) (img1_w, img1_h) of img1 with affine transformation matrix $\begin{matrix} 1+p_0 & p_1 & p_2 \\ p_3 & 1+p_4 & p_5 \end{matrix}$ where parameters p_i are driven from p

   $W(x;p)$: Make a new coordinate matrix(mesh grid) of the same size as img1, whose two vertexes are warped (0,0) and warped (img1_w, img1_h)

   $I(W(x;p))$: Apply rectangular bivariate spline interpolation over each point on the mesh grid

2. Set $T(x)$ = img1 to use img1 as a template. Then compute $T(x) - I(W(x;p))$ and flatten it to size n x 1 where n is the total number of pixels in img1

3. Merge Gx and Gy to create n x 2 size gradient vector ∇I

4. Compute n x 6 size $\nabla I \frac{\partial W}{\partial p}$ where $\frac{\partial W}{\partial p}$ is a Jacobian matrix $\begin{matrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{matrix}$

5. Compute 6 x 6 size Hessian matrix with $H = (\nabla I \frac{\partial W}{\partial p})^T \nabla I \frac{\partial W}{\partial p}$

6. Compute 6 x 1 size $\Delta p = H^{-1} \sum [\nabla I \frac{\partial W}{\partial p}]^T [T(x) - I(W(x;p))]$

2.2 Moving Object Detection

I iterated Lucas-Kanade method described above until $||\Delta p|| \leq threshold$

- $threshold$ is a hand-craft parameter, which is set to 0.171 here, carefully selected from the trade-off between accurate performance and computational workload.

After each iteration I updated p with $p + \Delta p$ and after the whole iteration had ended I computed difference between warped img1 which is a warped image of img1 through affine transformation matrix using parameters of p, and img2.