Solution 1

(a)

$$P(All inliers for K) = \frac{\binom{N-J}{K}}{\binom{N}{K}} \tag{1}$$

(b)

$$P(at least once outlier free) = 1 - P(not outlier free) = 1 - [1 - \frac{\binom{N-J}{K}}{\binom{N}{K}}]^T >= P \tag{2}$$

so

$$T >= \frac{log(1-P)}{log(1 - \frac{\binom{N-J}{K}}{\binom{N}{K}})} \tag{3}$$

(c)

$$P(All inliers for K) = \frac{\binom{I_1}{K} + \binom{I_2}{K}}{\binom{N}{K}} \tag{4}$$

Solution 2

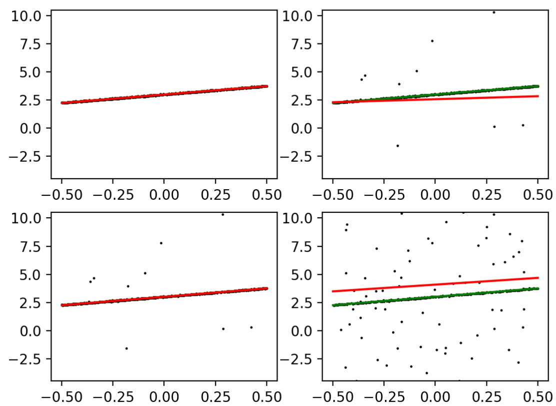

Figure 1: 2a

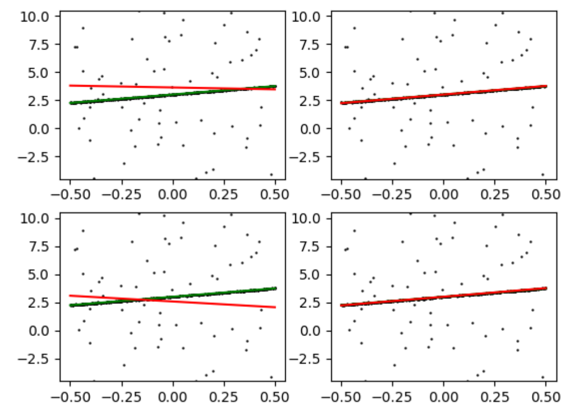

Figure 2: 2b

(a) When there are no outliers, simple linear fitting performs as good as robust line fitting. When there are a small number of outliers, the robust line fitting fits better than the simple linear fitting, since it filters out outliers during iterations. When there are a great number of outliers, the robust fitting still performs well but with some acceptable error 1.19.

(b) The bottom right choice always performs the best with least errors while the top left always has more error. The second one sometimes have significant error because it has a rather huge variance. The third one fits well but always has some minor errors, thus it has a better performance than the second one. K=5, N=1000 is the best fit.

Solution 3

(a) The point projected on camera1 is $P_1 = [x_1, y_1, 1]^T$, the point projected on camera2 is $P_2 = [x_2, y_2, 1]^T$, the original point is P'.

$$P' = \begin{bmatrix} f_1 & 0 & \frac{W}{2} \\ 0 & f_1 & \frac{H}{2} \\ 0 & 0 & 1 \end{bmatrix}^{-1} P_1 = \begin{bmatrix} f_2 & 0 & \frac{W}{2} \\ 0 & f_2 & \frac{H}{2} \\ 0 & 0 & 1 \end{bmatrix}^{-1} P_2 \tag{5}$$

Then,

$$\begin{bmatrix} \frac{1}{f_1} & 0 & \frac{-W}{2f_1} \\ 0 & \frac{1}{f_1} & \frac{-H}{2f_1} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{f_2} & 0 & \frac{-W}{2f_2} \\ 0 & \frac{1}{f_2} & \frac{-H}{2f_2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{6}$$

Thus

$$x_1 = \frac{2f_1x_2 + Wf_2 - Wf_1}{2f_2} \tag{7}$$

$$y_1 = \frac{2f_1y_2 + Hf_2 - Hf_1}{2f_2} \tag{8}$$

(b) In this case, this is a 3-D world case. Thus $p = [x \;\; y \;\; z \;\; 1]^T$

$$\tilde{p_1} = \lambda_1 K_1 [R_1 | t_1] p = \lambda_1 K_1 [R_1 [x \;\; y \;\; z]^T + t_1] \tag{9}$$

$$[x \;\; y \;\; z]^T = \frac{1}{\lambda_1} R_1^{-1} K_1^{-1} \tilde{p_1} - R_1^{-1} t_1 \tag{10}$$

$$[x \;\; y \;\; z]^T = \frac{1}{\lambda_2} R_1^{-1} K_2^{-1} \tilde{p_2} - R_2^{-1} t_2 \tag{11}$$

We can then set the world plane to be represented as $A \cdot [x \;\; y \;\; z \;\; 1]^T = 0$, where $A = [a \;\; b \;\; c \;\; d]$

$$A \cdot [\frac{1}{\lambda_1} R_1^{-1} K_1^{-1} \tilde{p_1} - R_1^{-1} t_1 \quad 1]^T = 0 \tag{12}$$

$$\frac{1}{\lambda_1} [a \;\; b \;\; c][R_1^{-1} K_1^{-1} \tilde{p_1} - R_1^{-1} t_1] + d = 0 \tag{13}$$

$$\frac{1}{\lambda_1} = \frac{-d}{[a \;\; b \;\; c][R_1^{-1} K_1^{-1} \tilde{p_1} - R_1^{-1} t_1]} \tag{14}$$

$$\frac{1}{\lambda_2} = \frac{-d}{[a \;\; b \;\; c][R_2^{-1} K_2^{-1} \tilde{p_2} - R_2^{-1} t_2]} \tag{15}$$

Then we can derive a equation with info from tow cameras:

$$\frac{\lambda_1}{R_1^{-1} K_1^{-1} \tilde{p_1} - R_1^{-1} t_1} = \frac{\lambda_2}{R_2^{-1} K_2^{-1} \tilde{p_2} - R_2^{-1} t_2} \tag{16}$$

So we can relate the mapping of two camera in this way.

Solution 4

Figure 3: prob4

Solution 5

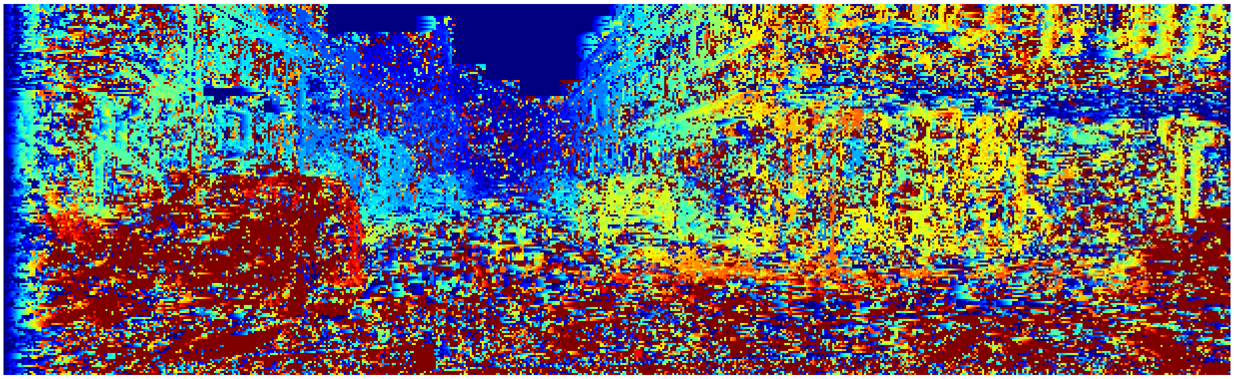

Figure 4: prob5

Information

This problem set took approximately 20 hours of effort.

I discussed this problem set with:

- Mingyu Cao

I also got hints from the following sources:

- https://math.stackexchange.com/questions/131590/derivation-of-the-formula-for-ordinary-least-squares-linear-regression
- https://math.stackexchange.com/questions/494238/how-to-compute-homography-matrix-h-from-corresponding-points-2d-2d-planar-homog
- http://pages.cs.wisc.edu/ dyer/cs534/slides/08-mosaics.pdf
- https://stackoverflow.com/questions/12729228/simple-efficient-bilinear-interpolation-of-images-in-numpy-and-python