EECS 442 F14
midterm

WU Tongshuang 40782356

October 30, 2014

Problem 1 Panoramic Imaging Problem

a

Suppose P_1 be the original point for point p_1' in image 1 and p_1 in image 2.

For image 1:

$$\begin{aligned} p_1' &= M_1 P_1 \\ &= K_1 \begin{bmatrix} R_1 & T_1 \end{bmatrix} P_1 \\ &= K \begin{bmatrix} I & 0 \end{bmatrix} P_1 \end{aligned}$$

Similarly, for image 2:

$$\begin{aligned} p_1 &= M_2 P_1 \\ &= K_2 \begin{bmatrix} R_2 & T_2 \end{bmatrix} P_1 \\ &= K \begin{bmatrix} R & 0 \end{bmatrix} P_1 \end{aligned}$$

So,

$$\begin{aligned} p_1 &= K \begin{bmatrix} R & 0 \end{bmatrix} P_1 \\ &= KRK^{-1} K \begin{bmatrix} I & 0 \end{bmatrix} P_1 \\ &= KRK^{-1} p_1' \end{aligned}$$

This procedure can be repreated for P_i, p_i', p_i, for i = 1, 2, 3, 4.

b

Yes. to verify the above equation, computation is needed. since K have 5 DOF $(\alpha, \beta, c_x, c_y, s)$ and R has 3 DOF, there are 8 DOF in total and requires at least $N = 4$ points for verification.

c

Since the only differences between two cameras are the extrinsic rotation matrix, and images taken under the above camera configurations give rise to a set of "panoramic" images, meaning that we can have $m \geq$ views acquired by camera rotating about its center with fixed internal parameters.

The procedure is as the following:

1. **Inter-image homographies:** Compute the homography H^i between each view i and a reference view such that $x^i = H^i x$ using, for example, algorithm 4.6(p123). Normalize the matrices such that $detH^i = 1$.
2. **Compute** ω:
   - In the case of constant calibration: $\omega = (H_i)^{-T}\omega(H_i)^{-1}, i = 1, ..., m$ as $Ac = 0$ where A is a $6m \times 6$ matrix, and c the elements of the conic ω arranged as a 6-vector, or
   - For variable calibration parameters, use the equation $\omega^i = (H^i)^{-T}\omega(H^i)^{-1}$ to express linear constraints on entries of ω^i as linear equations in the entries of ω.
3. **compute** K: Determine the Cholesky decomposition of ω as $\omega = UU^T$, and thence $K = U^{-T}$
4. **compute** R^i: $R^i = K^{-1}H^iK$
5. **Iterative improvement**: Refine the linear estimate of K by minimizing

$$\sum_{i=2,m;j=1,n} d(x^i_j, KR^iK^{-1}x_j)^2$$

   Over K and R^i, where x_j, x^i_j are the position of the j-th point measured in the first and i-th images respectively.

d

Given projection matrices for the two views, $P = \begin{bmatrix} I & 0 \end{bmatrix}$, and $P' = \begin{bmatrix} A & a \end{bmatrix}$ and a plane defined by $\pi^T X = 0$ with $\pi = (v^T, 1)^T$, the homography induced by the plane is $x' = Hx$ with

$$H = A - av^T$$

Since for first view $x = \begin{bmatrix} I & 0 \end{bmatrix} X$, any points on the ray $X = (x^T, \rho)^T$ projects to x, where ρ parametrizes the point on the ray.

$\pi^T x = 0$ determines ρ, $X = (x^T, -v^T x)^T$. So 3D point X projects into second view:

$$\begin{aligned} x' &= P'X \\ &= \begin{bmatrix} A & a \end{bmatrix} \\ &= Ax - av^T x \end{aligned}$$

With: $\begin{bmatrix} v^T \\ 1 \end{bmatrix}$, for points on the plane:

$$v^T X + 1 = 0$$

For cameras $M_1' = \begin{bmatrix} I & 0 \end{bmatrix}$, and $M_2' = \begin{bmatrix} R & t \end{bmatrix}$, the homography is:

$$H = R - tv^T$$

Applying the transformation K to the images, we get $M_1 = K \begin{bmatrix} I & 0 \end{bmatrix}$, and $M_2 = K \begin{bmatrix} R & t \end{bmatrix}$, and that

$$H = K(R - tv^T)k^{(-1)}$$

Problem 2 Computing H

Brief introduction of DLT:

- For each correspondence $x_i \leftrightarrow x_i'$ compute the matrix A_i from:

$$\begin{bmatrix} 0^T & -w_i' x_i^T & y_i' x_i^T \\ w_i' x_i^T & 0^T & -x_i' x_i^T \\ -y_i' x_i^T & x_i' x_i^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0$$

- Assumble the $n2 \times 9$ matrices A_i into a single $2n \times 9$ matrix A.
- Obtain the SVD of A. The unit singular vector corresponding to the smallest singular value is the solution h. Specifically, if $A = UDV^T$ with D diagonal with positive diagonal entries, arranged in descending order down the diagonal, then h is the last column of V.
- The matrix H is determined from h as:

$$\begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} = 0$$

The obtained H is:

$$H = \begin{bmatrix} -1.8619 & -0.5207 & 591.3803 \\ -0.8225 & -2.2354 & 441.5196 \\ -0.0044 & -0.0046 & 1 \end{bmatrix}$$

For the code, please refer to *Homography.m*

Problem 3 Removing Outliers

a

The obtained H is:

$$H = \begin{bmatrix} -0.4894 & -0.6123 & 312.9422 \\ -0.3892 & -0.5873 & 262.4202 \\ -0.0015 & -0.0021 & 1 \end{bmatrix}$$

To fix this while sticking with a least square fitting method, we could normalize the points.

b

For this question, a normalized and an unormalized version are conducted. For normalized version, please refer to *RANSAC_norm.m, Homography_norm.n, normMatrix.m, q3Output_norm.txt* for code and output. As for the normalized one, please refer to *RANSAC.m, Homography.m, q3Output.txt*.

Images are as the following.

Figure 1: figure1

Discussion of the results:

While the *un-normalized* version could sometimes give an acceptable result (e.g. $support = 108$), most of the results are very poor (e.g. $support < 10$). This should be due to

- Very small σ making the inlier-test very unlikely to be satisfied

Figure 2: figure2

- When randomly picking correspondences among spared and sometimes incorrect mapped points, it's very likely that we selected less representative (very cornered in the image), or even totally wrong points. In etreme cases where all selected points are outliers, poor performances can be expected.

In the mean time, since normalization of the total 200 points fix the problem of H not being reliable, and better conditioned points, its result is much more desirble. Concretely, when running it repeatedly, the resulted *support* number tends to stablize around 170 to 179, which is very close to the accurate outlier rate, i.e. 176 / 200.

Due to the significant advantage of normalized construction, the final step, i.e. re-estimate H from all correspondences classified as inliers, is processed only for the normalized version. Its fitting error is 66.1095.

The re-estimated H is:

$$H = 10^3 \times \begin{bmatrix} 0.0049 & 0.0006 & -1.4712 \\ 0.0013 & 0.0043 & -0.6647 \\ 0.0000 & 0.0000 & 0.0010 \end{bmatrix}$$

Which is the fitting line.

[Bonus 5 Pt] Solving the Image Panorama Problem

Please refer to $1(c)$ for algorithm outline.