16-822: Geometry Based Methods
Assignment 1

Andrew ID: sajaved

Question 1

1.1

a

Let there be two points, P & and P' on the plane which are mirror images around the line L. We fix the origin in a way that makes the projection of the first point be as follows:

$$p = K[I|0]P$$

Assuming the reflection matrix R defines the mirror image relationship between the two 3D points, the other point's projection is given by:

$$p' = K[R|0]P$$

Therefore the homography H between the two projected points is given by:

$$p' = KRK^{-1}p$$
$$H = KRK^{-1}$$

b

$$H = KRK^{-1}$$
$$HH = KRK^{-1}KRK^{-1}$$

Since the matrix R is a reflection, reflecting any point twice using it, gives back the original point and makes the effective transformation identity. Therefore:

$$H^2 = I$$

c

Let a be a unit vector along the direction of the line L. The reflection matrix applied to this vector will give the same vector as it lies on the line along which the reflection is symmetric. More specifically, the vector a is a unit eigenvector of the rotation matrix R

$$Ra = a$$

Pre-mutiplying by K, we get:

$$KRa = Ka$$

We also know that:

$$\begin{aligned} H &= KRK^{-1} \\ HK &= KR \\ HKa &= KRa \end{aligned}$$

Combining the above two results:

$$HKa = Ka$$

Therefore, Ka is an eigenvector of H. Ka is also the vanishing line of L since a is a unit vector along the line L. Hence, the eigenvector of H is the vanishing line of L.

1.2

a

The slope of the line represented by the 3-vector [u v w]^T is given by -u/v.

b

By duality, a pencil of lines in a plane is the same as a set of collinear points which lie in a 1D projective space. Hence, any two points can be represented as a linear combination of two others. More formally, let l_1 and l_2 be two lines and let their point of intersection P be represented as the cross product between them. For any arbitrary line l which is a linear combination of l_1 and l_2, we can show that l also intersects at p:

$$\begin{aligned} lp &= 0 \\ (\lambda_1 l_1 + \lambda_2 l_2)(l_1 \times l_2) &= 0 \\ \lambda_1 l_1 (l_1 \times l_2) + \lambda_2 l_2 (l_1 \times l_2) &= 0 \\ \lambda_1 \times 0 + \lambda_2 \times 0 &= 0 \end{aligned}$$

c

Since the pencil of lines lie in 1D projective space, the dimensionality of homography H between two pencil of lines is a 2×2 matrix.

d

The relationship between the two set of lines l and l' is given by the homography H:

$$\begin{bmatrix} l'_1 \\ l'_2 \\ l'_3 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & 0 \\ h_3 & h_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$$

Simplifying them:

$$\begin{aligned} l'_1 &= h_1 l_1 + h_2 l_2 \\ l'_2 &= h_3 l_1 + h_4 l_2 \\ l'_l &= l_3 \end{aligned}$$

The slope of the line l' is defined as $s' = l'_1/l'_2$ simplifying which gives the following expression:

$$s' = -\frac{l'_1}{l'_2}$$
$$s' = -\frac{h_1 l_1 + h_2 l_2}{h_3 l_1 + h_4 l_2}$$

Dividing all the terms by l_2 and replacing $s = -l_1/l_2$ we get:

$$s' = \frac{h_1 s - h_2}{-h_3 s + h_4}$$
$$s' = \frac{as + b}{cs + d}$$

e

The homography H in terms of a, b, c, d is given by:

$$H = \begin{bmatrix} h_1 & h_2 & 0 \\ h_3 & h_4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$H = \begin{bmatrix} a & -b & 0 \\ -c & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For it to be a valid line homography, the matrix should be non-singular and its determinant should not be zero. Therefore, the condition is $ad - bc \neq 0$.

1.3

a

Rodrigues' formula in matrix form can be written by converting the cross product to its skew-symmetric form:

$$Rp = p + sin(\theta) w \times p + (1 - cos(\theta)) w \times (w \times p)$$
$$Rp = p + sin(\theta)[w]_\times p + (1 - cos(\theta))[w]_\times^2 p$$
$$\text{where, } [w]_\times = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}$$

b

Rodrigues' formula for small angle approximations is given by replacing $\sin(\theta)$ by θ and $\cos(\theta)$ by $1 - \frac{\theta^2}{2}$. The Rodrigues' formula in terms of $\Omega = \theta[w]_\times$ will be:

$$Rp = p + \theta[w]_\times p + \frac{\theta^2 [w]_\times^2 p}{2}$$
$$Rp = p + \Omega p + \frac{\Omega^2 p}{2}$$

c

For calibrated cameras, the essential matrix will be:

$$\begin{aligned} E &= [t]_\times R \\ &= [t]_\times (I + \Omega + \frac{\Omega^2}{2}) \end{aligned}$$

d

Let the epipoles be related by the matrix R. Also, the epipole $e = t$,

$$\begin{aligned} e' &= Re \\ &= (I + \Omega + \frac{\Omega^2}{2})t \\ &= (I + \theta[w]_\times + \frac{(\theta[w]_\times)^2}{2})t \end{aligned}$$

Since the vector w is parallel to t, t will be perpendicular to $[w]_\times$.

$$\begin{aligned} e' &= (t + 0 + 0) \\ e' &= t \end{aligned}$$

Hence the second epipole is at t (up to scale).

e

Let (X, Y, Z) be the point in 3D being projected. The point coordinates in one image plane given in normalized coordinates is:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}$$

For calibrated cameras, we express the 3D points X and Y as:

$$\begin{aligned} X &= Zu \\ Y &= Zv \end{aligned}$$

The equation for projecting the point X, Y, Z on the other image is given by:

$$\begin{bmatrix} u + du \\ v + dv \\ 1 \end{bmatrix} = [R|t] \begin{bmatrix} Zu \\ Zv \\ Z \\ 1 \end{bmatrix}$$

Rearraaging the terms, we get du, dv in terms of u, v, Ω, t and Z:

$$\begin{bmatrix} du \\ dv \\ 0 \end{bmatrix} + \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = ZR \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + t$$
$$\begin{bmatrix} du \\ dv \\ 0 \end{bmatrix} = (ZR - I) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + t$$
$$\begin{bmatrix} du \\ dv \\ 0 \end{bmatrix} = (Z(I + \Omega + \frac{\Omega^2}{2}) - I) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + t$$

f

If the point P lies on the plane given by (n,d), then it satisfies the equation of the plane:

$$n^T P + d = 0$$
$$\begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + d = 0$$

We substitute the values of X, Y form the previous section:

$$\begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} Zu \\ Zv \\ Z \end{bmatrix} + d = 0$$
$$Z(n_1 u + n_2 v + n_3) + d = 0$$
$$Z = \frac{-d}{(n_1 u + n_2 v + n_3)}$$

Substituting this in the expression of du, dv, we get:

$$\begin{bmatrix} du \\ dv \\ 0 \end{bmatrix} = (\frac{-d}{(n_1 u + n_2 v + n_3)}(I + \Omega + \frac{\Omega^2}{2}) - I) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + t$$

The above equation is independent of Z.

g

The above expression of du, dv describes a hyperbola-like curve which is a quadratic expression in u, v. To see this more clearly, we can choose a specific case where the rotation matrix to be identity and translation to be 0. Hence, for any general case, the curve is quadratic.

1.4

a

The pixel coordinates of the three points P, Q and S are given by:

$$s = MS$$

Since the light travels from S to P and makes a shadow of P at Q, the points P,Q and S are collinear in physical space. Thus the determinant of the matrix formed by these three points is 0. Using this we can show that the three projected points in the image are collinear too and thus their triple product is zero.

$$
\begin{aligned}
s.(p \times q) &= 0 \\
MS(p \times q) &= 0
\end{aligned}
$$

We can flatten and rearrange the above matrices such that it forms a system of equations of the form $AM = 0$ and find its solutions using SVD. Then we can decompose it into K and R using QR decomposition.

b

The above equation is linear in elements of M. M itself is composed of the intrinsic (5 dof) and the extrinsic matrix (6 dof). Since we are given that we can assume the camera to be at origin, we can remove the translation. Now we have 3 dof left for the rotations and 5 dof in the intrinsic matrix of the camera. Hence we need 8 set of points to solve for M. For decomposing it into K and R, we can use QR decomposition. Also, we need the images from multiple days as we need distinct set of collinear points to make the system of equation. For certain alignments of the sun and the 3D point, the points might be a linear combination of each other.

1.5

a

It's given that:

$$MP = p = p' = M'P$$

The epipolar constraint can be rewritten using the above:

$$p'^T Fp = 0 = p^T Fp$$

b

Any matrix can be decomposed into the sum of a symmetric matrix S and skew-symmetric matrix X

$$
\begin{aligned}
F &= \frac{F + F^T}{2} + \frac{F - F^T}{2} \\
F &= S + X
\end{aligned}
$$

c

We have proved that:

$$
\begin{aligned}
p^T Fp &= 0 \\
p^T (S + X)p &= 0 \\
p^T Sp + p^T Xp &= 0
\end{aligned}
$$

The skew-symmetric matrix can be written in a cross product notation in the form of a vector

$$
\begin{aligned}
p^T Sp + p^T Xp &= 0 \\
p^T Sp + p^T [x]_\times p &= 0 \\
p^T Sp + p^T (x \times p) &= 0
\end{aligned}
$$

Since the triple product in the second term is zero, only the first term remains.

$$p^T S p = 0$$

Hence, the points belonging to E are a quadratic conic in S.

d

We know from epipolar geometry that $Fe = 0$. Using this and results from previous sections:

$$\begin{aligned} Fe &= 0 \\ e^T F e &= 0 \\ e^T S e + e^T X e &= 0 \\ e^T S e + e^T (x \times e) &= 0 \\ e^T S e &= 0 \end{aligned}$$

This can also be shown for the other epipole e'. Therefore, both the epipoles belong to the given conic.

Question 2

The code first does a affine rectification of the image which preserves parallel lines followed by a metric rectification which also preserves orthogonal angles.

For the affine rectification, we take two pairs of parallel lines which meet at the vanishing point at infinity.

$$\begin{aligned} p\_inf_1 &= l_1 \times l_2 \\ p\_inf_2 &= l_3 \times l_4 \end{aligned}$$

The vanishing line $l = (l_1, l_2, l_3)$ passing through these two points is given by:

$$l = p\_inf_1 \times p\_inf_2$$

Since we want the homography to preserve parallelism, we know that the affine matrix H is given by:

$$H_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix}$$

Next, for preserving orthogonality, we rewrite the cosine of the angles between two lines in the form as follows:

$$\begin{aligned} cos(\theta) &= \frac{l_1^T C_\infty^* l_2}{\sqrt{l_1 C_\infty^* l_1 l_2 C_\infty^* l_2}} \\ C_\infty^* &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Let the affine matrix to preserve orthogonality be given by:

$$H = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix}$$

We choose pairs of orthogonal lines $l = (l_1, l_2)$ and $m = (m_1, m_2)$ in the affinely transoformed image and enforce the cosine angle between them to be zero. This results in the equation:

$$l^T H C_\infty^* H^T m = 0$$

$$l^T \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} A^T & 0 \\ t^T & 1 \end{bmatrix} m = 0$$

$$(l'_1, l'_2)\, AA^T\, (m'_1, m'_2)^T = 0$$

We let $S = AA^T$, where S will be a symmetric matrix $S = \begin{bmatrix} s_{11} \\ s_{12} \end{bmatrix}$. We solve the following equation to get the matrix elements s:

$$(l'_1 m'_1, l'_1 m'_2 + l'_2 m'_1) \begin{pmatrix} s_{11} \\ s_{12} \end{pmatrix} = -l'_2 m'_2$$

We can use SVD to solve for the matrix A and in turn, H.

Running the Code

The function **[metric_rectified, H_final] = rectifyImage(filename, debug)** can be run with debug=0 to obtain the results stated below. The first two angles printed are cosines of the selected perpendicular lines. For the third pair, another function called **checkCosine** is run at the end which picks another set of perpendicular lines and checks its cosine. The data mat files for the selected points shown below are stored in **data/** folder and the resulting images are stored in **results/**.

Difficulties in Rectification

There are multiple things which can influence the rectification:

- The precision in annotating the parallel and perpendicular points results in a lot of variation in the quality of rectification.
- If we choose the same set of lines (or parallel set of lines) for affine or metric rectification, we will not recover a solution.
- The choice of lines also influences the quality of rectification for cases with high non-uniform distortion. For example, if an image has significantly high distortion at the corners and we select parallel lines which are already 'almost parallel' in the original image, it will not give a lot of information about the warping required for the high-distortion regions.

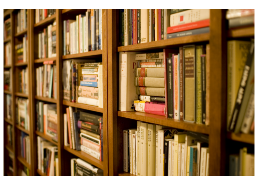
Results

Following are the 7 images from the given folder and 3 images from the internet:

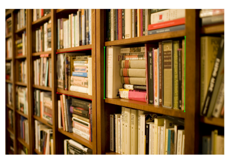
Before	After
0.86825	-3.2263e-17
-0.87413	3.379e-17
-0.84052	-0.037891

Table 1: cosine θ between lines

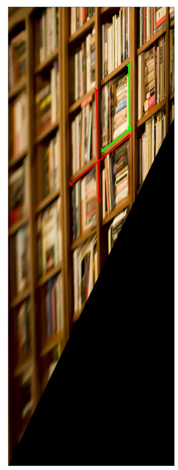
Image 1



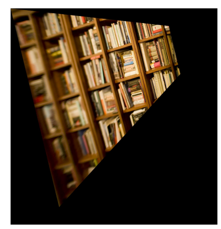
(a) Original



(b) Parallel lines



(c) Perpendicular lines



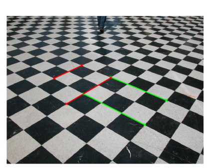
(d) Metric rectified

Image 2

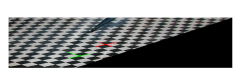
Before	After
0.97186	-4.8974e-16
-0.9312	4.1671e-16
-0.9669	-0.0146

Table 2: cosine θ between lines

(a) Original



(b) Parallel lines



(c) Perpendicular lines

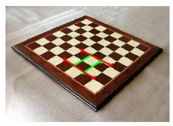
(d) Metric rectified

Image 3

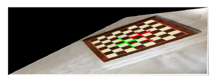
Before	After
0.53192	3.1984e-16
-0.55866	-3.2469e-16
0.43361	0.064005

Table 3: cosine θ between lines

(a) Original



(b) Parallel lines



(c) Perpendicular lines

(d) Metric rectified

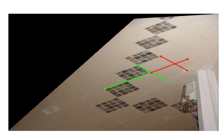
Image 4

Before	After
0.61659	1.8174e-16
0.43596	1.257e-16
-0.75292	0.019373

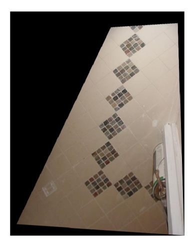
Table 4: cosine θ between lines

(a) Original

(b) Parallel lines



(c) Perpendicular lines



(d) Metric rectified

Before	After
-0.32176	-5.9979e-17
-0.2099	-1.9339e-16
0.21264	-0.011899

Table 5: cosine θ between lines

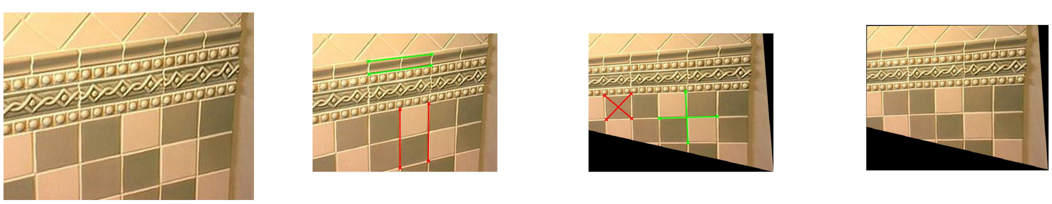
Image 5

(a) Original (b) Parallel lines (c) Perpendicular lines (d) Metric rectified

Image 6

Before	After
-0.19441	1.9189e-17
-0.079126	6.1191e-17
0.2307	0.0013956

Table 6: cosine θ between lines



(a) Original (b) Parallel lines (c) Perpendicular lines (d) Metric rectified

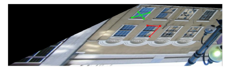
Image 7

Before	After
-0.95718	7.9506e-17
0.98107	1.618e-16
-0.98841	-0.052243

Table 7: cosine θ between lines

(a) Original

(b) Parallel lines



(c) Perpendicular lines

(d) Metric rectified

Image 8

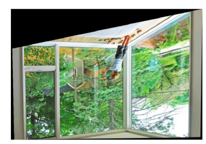
Before	After
-0.07197	-8.4036e-18
0.0099082	-3.7956e-18
-0.019318	0.0080722

Table 8: cosine θ between lines

(a) Original

(b) Parallel lines

(c) Perpendicular lines



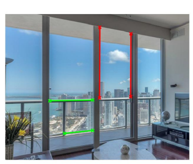
(d) Metric rectified

Image 9

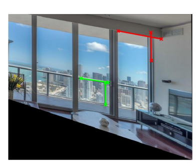
Before	After
-0.28233	-4.4894e-17
-0.19896	-7.098e-17
0.28423	-0.044721

Table 9: cosine θ between lines

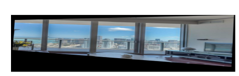
(a) Original



(b) Parallel lines



(c) Perpendicular lines

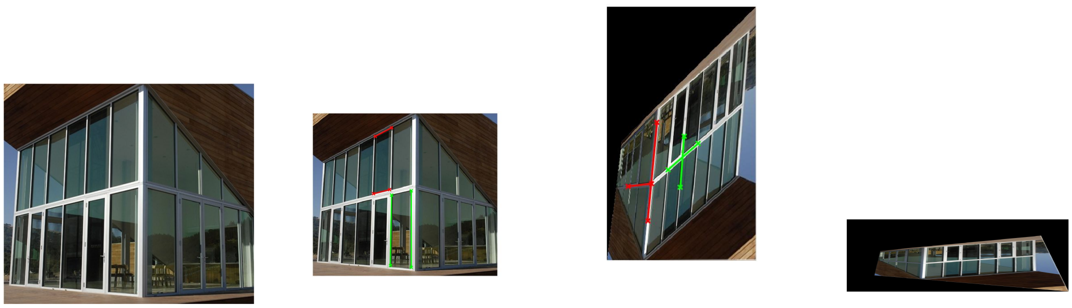


(d) Metric rectified

Before	After
-0.90238	-9.3808e-16
-0.74161	-9.1037e-16
-0.90396	-0.078519

Table 10: cosine θ between lines

Image 10



(a) Original (b) Parallel lines (c) Perpendicular lines (d) Metric rectified

Collaboration and online resources used

Some approaches were discussed with Talha Ahmad Siddique and Rishi Madhok. A few online resources were consulted during the derivation:

- Derivation of Image Rectification
- Tutorial on Projective Geometry
- Epipolar Geometry and the Fundamental Matrix from Multi-View Geometry textbook