

Computer Vision (600.461/600.661)

Homework 5: Two View Geometry and Optical Flow

Greg Kiar

November 14, 2014

1. (20 points) Properties of $so(3)$ and $SO(3)$.

(a) Given the structure of \hat{w} , we can determine its rank and nullspace as follows

$$\hat{w} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}$$

When we apply the following linear operation on w_3 , we see that the rows are linearly dependent on one another. $\therefore \text{rank}(\hat{w}) = 2$.

$$w_3 = w_1 \frac{-w_1}{w_3} + \frac{-w_2}{w_3}$$

By definition, $\hat{w} = w \times A$, where A is an arbitrary vector. $\therefore \hat{w}w = 0$. We can conclude from this that w is the nullspace of \hat{w} , since w is parallel to \hat{w} .

(b) We can solve for \hat{w}^2 is shown below

$$\begin{aligned} \hat{w} &= \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \\ \hat{w}^2 &= \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} -w_2^2 - w_3^2 & w_1w_2 & w_1w_3 \\ w_1w_2 & -w_1^2 - w_3^2 & w_2w_3 \\ w_1w_3 & w_2w_2 & -w_1^2 - w_2^2 \end{bmatrix} \\ &= \begin{bmatrix} w_1^2 & w_1w_2 & w_1w_3 \\ w_1w_2 & w_2^2 & w_2w_3 \\ w_1w_3 & w_2w_2 & w_3^2 \end{bmatrix} - \begin{bmatrix} w_1^2 + w_2^2 + w_3^2 & 0 & 0 \\ 0 & w_1^2 + w_2^2 + w_3^2 & 0 \\ 0 & 0 & w_1^2 + w_2^2 + w_3^2 \end{bmatrix} \\ \hat{w}^2 &= ww^T - \|w\|^2 I \end{aligned}$$

Now, similarly for \hat{w}^3

$$\begin{aligned} \hat{w}^3 &= \hat{w}\hat{w}^2 \\ &= \hat{w}(ww^T - \|w\|^2 I) \\ &= \hat{w}ww^T - \hat{w}\|w\|^2 \\ \hat{w}^3 &= \|w\|^2 \hat{w} \end{aligned}$$

(c) R is found by solving the following differential equation as a discrete time system

$$\begin{aligned}
R(t)R(t)^T &= I \\
(R(t)\dot{R}(t)^T) &= \dot{I} \\
\dot{R}(t)R(t)^T + R(t)\dot{R}(t)^T &= 0 \\
\dot{R}(t)R(t)^T &= -R(t)\dot{R}(t)^T \\
\dot{R}(t)R(t)^T &= -(\dot{R}(t)R(t)^T)^T \\
\hat{w} &= \dot{R}(t)R(t)^T \\
\text{Now,} \\
\dot{R}(t)R(t)^T R(t) &= \hat{w}R(t) \\
\dot{R}(t) &= \hat{w}R(t)
\end{aligned}$$

Where \hat{w} is now set to be skew symmetric matrix $R(t)'R^T$ is, solving this using the laplace transform yeilds the following

$$\begin{aligned}
\dot{R}(t) &= \hat{w}R(t) \\
\mathcal{L}\{\dot{R}(t)\} &= \mathcal{L}\{\hat{w}R(t)\} \\
sR(s) - R_0 &= \hat{w}R(s) \\
R(s) &= \frac{R_0}{\hat{w} + s} \\
\mathcal{L}^{-1}\{R(s)\} &= \mathcal{L}^{-1}\left\{\frac{R_0}{\hat{w} + s}\right\} \\
R(t) &= R_0 \exp(\hat{w}t)
\end{aligned}$$

(d) A simplification of R from the result obtained above is as follows

$$\begin{aligned}
R &= \exp(\hat{w}\theta) \\
&= \sum_{n=0}^{\infty} \frac{(\hat{w}\theta)^n}{n!} \\
&= \sum_{n=0}^{\infty} \frac{\hat{w}^n}{n!} \sum_{n=0}^{\infty} \frac{\theta^n}{n!}
\end{aligned}$$

We know from expanding the properties of \hat{w} discovered above, that $\hat{w} = -\hat{w}^3$, $\hat{w} = \hat{w}^4$. We also know that $\hat{w}^2 = -\hat{w}^4$, $\hat{w} = \hat{w}^6$. From this we can state the following

$$\begin{aligned}
R &= \hat{w}^0 + \hat{w} \sum_{n=0}^{\infty} \frac{(-1)^n (\theta)^n}{n!} + \hat{w}^2 \sum_{n=0}^{\infty} \frac{(-1)^{n+1} \theta^n}{n!} \\
&= I + \hat{w} \sum_{n=0}^{\infty} \frac{(-1)^n (\theta)^n}{n!} + \hat{w}^2 \sum_{n=0}^{\infty} \frac{(-1)^{n+1} \theta^n}{n!}
\end{aligned}$$

Applying infinite series expansion we find that

$$R = I + \sin(\theta)\hat{w} + (1 - \cos(\theta))\hat{w}^2$$

(e) The trace of R is computed as follows

$$\begin{aligned}
tr(R) &= tr(I + \sin(\theta)\hat{w} + (1 - \cos(\theta))\hat{w}^2) \\
&= tr(I) + tr(\sin(\theta)\hat{w}) + tr((1 - \cos(\theta))\hat{w}^2) \\
&= 3 + tr\left(\sin(\theta)\begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}\right) + tr((1 - \cos(\theta))(ww^T - \|w\|^2 I)) \\
&= 3 + (1 - \cos(\theta))tr\left(\begin{bmatrix} w_1^2 & w_1w_2 & w_1w_3 \\ w_1w_2 & w_2^2 & w_2w_3 \\ w_1w_3 & w_2w_2 & w_3^2 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \\
&= 3 + (1 - \cos(\theta))(1 - 3) \\
&= 3 - 2(1 - \cos(\theta)) \\
tr(R) &= 1 + 2\cos(\theta)
\end{aligned}$$

(f) We show that w is an eigen vector of R below.

$$\begin{aligned}
Rw &= (I - \sin(\theta)\hat{w} + (1 - \cos(\theta))\hat{w}^2)w \\
Rw &= Iw \\
Rw &= w
\end{aligned}$$

We can see that w is an eigen vector for R for the eigen value of $\lambda = 1$.

2. (15 points) Optical flow of a perspective camera.

(a) The optical flow of a perspective projection at $t = 0$ can be given using the following calculation

$$\begin{aligned}
x(t) &= \pi(R(t)X + T(t)) \\
&= R(t)\frac{X}{Z} + T(t)\frac{1}{Z} \\
\dot{x} &= \dot{R}(t)x + \dot{T}(t)\frac{1}{Z}
\end{aligned}$$

Where $\dot{R}(t)R(t)^T = \hat{w}(t)$, and $\because R(0) = I, \dot{R}(t) = \hat{w}$. For clarity let $w(t) = w$.

$$\begin{aligned}
\dot{x} &= \hat{w}x + \frac{1}{Z}v \\
&= -\hat{x}w + \frac{1}{Z}v \\
(I - xe_3^T)\dot{x} &= -(I - xe_3^T)\hat{x}w + \frac{1}{Z}(I - xe_3^T)v \\
\begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} &= (-\hat{x} + xe_3^T\hat{x})w + \frac{1}{Z}(I - xe_3^T)v \\
\begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} &= \left(-\begin{bmatrix} 0 & -1 & y \\ 1 & 0 & -x \\ -y & x & 0 \end{bmatrix} + \begin{bmatrix} -xy & -x^2 & 0 \\ -y^2 & xy & 0 \\ y & x & 0 \end{bmatrix}\right)w + \frac{1}{Z}\begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 0 \end{bmatrix}v \\
\begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} &= \begin{bmatrix} -xy & 1+x^2 & -y \\ -1-y^2 & xy & -x \\ 0 & 0 & 0 \end{bmatrix}w + \frac{1}{Z}\begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 0 \end{bmatrix}v \\
\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} &= \begin{bmatrix} -xy & 1+x^2 & -y \\ -1-y^2 & xy & -x \end{bmatrix}w + \frac{1}{Z}\begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \end{bmatrix}v
\end{aligned}$$

(b)

3. (15 points) 8-point algorithm.

Shown in the figure below, are the errors of the implemented two view SFM algorithm. From top to bottom, this plots the error in rotation, translation, and resulting image. The errors are all represented in degrees and computed based on matrices normalized to 1. Every $n = 100$ samples, the σ value increases over a range of $(0, 1 \times 10^{-4})$. We can see that the algorithm implemented is quite accurate for no noise, but the addition of even small noise decreases performance significantly. We can see from the graph of \mathbf{X} error, that the error is relatively constant for higher σ values, with spikes of being either less or more accurate. However, we can see that our transforms both significantly fluctuate with the addition of a small amount of noise. The true 3D coordinates of the dataset, \mathbf{X} , are only reliably computed for the no noise situation using this method.

