ECSE 6650 - Computer Vision

Project #4: Features Tracking and Shape & Structure from Motion

Peng-Jui Ku, Lu-Yun Chen, Jie Zou

December 3, 2001

1. Introduction

Feature tracking is the problem of matching features from frame to frame in a long sequence of images. A popular technique for feature tracking is called *Kalman filtering*. It is a recursive procedure that estimates the position of a point as well as its uncertainty from the current frame to the next one.

Given the motion field estimated from an image sequence, we can recover the 3D shape of the 3D objects and their 3D motion relative to the viewing camera. Among many methods, *factorization method* is simple to implement and gives very good and numerically stable results for objects viewed from rather large distances.

In this project, a tracking algorithm based on Kalman filter is implemented to track a moving polyhedron. Its 3D shape and motion is then recovered using the factorization method. This report is organized as following. Section 2 and section 3 describe the Kalman filtering and factorization method respectively. Section 4 discusses the experiment procedure and results. The conclusion is summarized in Section 5.

2. Mathematic discussion of Kalman filtering

A popular technique for feature tracking is based on Kalman filter. It is a recursive algorithm that estimates the position and uncertainty of a moving feature point in the next frame. In other terms, we are interested in determining a predicted position and a search region in which the feature point can be found within a certain confidence. Kalman filtering assumes 1) linear state model, and 2) uncertainty is zero-mean Gaussian.

We consider only one feature point $p_t = (x_t, y_t)$, where t represents time instant. Assume the velocity is $v_t = (v_{x,t}, v_{y,t})$. The state at t is represented by the point's location and velocity, so $s_t = [x_t, y_t, v_{x,t}, v_{y,t}]^t$. The goal of Kalman filter is to compute the state vector from frame to frame, i.e., estimate s_{t+1} given s_t.

Kalman filtering consists of two parts: state predication and state updating. State prediction is performed using the state model, while state updating is performed using measurement model.

State model describes the temporal part of the system. According to the theory of Kalman filtering, the state model is a linear model:

$$s_{t+1} = \Phi s_t + w_t \qquad (1)$$

where Φ is the state transition matrix and w_t represents system perturbation. If we assume the feature movement between two consecutive frames is small enough to consider the motion of feature positions from frame to frame is uniform, the state transition matrix can be parameterized as $\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. The system perturbation is a zero-mean, normal Gaussian distribution: $w_t \sim N(0, Q)$.

For the tracking, the measurement is obtained via a feature detection process. Measurement model describes the spatial features of the system. It is also a linear model:

$$z_t = Hs_t + v_t \qquad (2)$$

where H relates current state to current measurement and v_t represents measurement uncertainty. Since z_t only involves position, H can be represented as $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$. Measurement uncertainty is normally distributed as zero-mean Gaussian $v_t \sim N(0, R)$.

The Kalman filter is a recursive process with the following four steps:

1. State and covariance prediction
   Given current state s_t and its covariance matrix Σ_t, state prediction involves two steps:

   1. State predication: $s^-_{t+1} = \Phi s_t$ (3)
   2. Covariance prediction; $\Sigma^-_{t+1} = \Phi \Sigma_t \Phi^t + Q$ (4)

2. Actual measurement to obtain z_{t+1}
   The feature detector (e.g., thresholding or correlation) searches for the region determined by the covariance matrix Σ^-_{t+1} to find the feature point at time t+1, z_{t+1}. The search region which contains the actual state with a given probability c^2 is an ellipse, and satisfied the following equation:

$$\left(p - p^-_{t+1}\right)\left(\Sigma^{p-}_{t+1}\right)^{-1}\left(p - p^-_{t+1}\right)^T \le c^2 \qquad (5)$$

   where p represents the point coordinates in the image frame. The points within the search region should satisfy the above equation. p^-_{t+1} is the center of the search region, which is also the predicted position. Σ_{t+1}^{p-} is the upper-left 2-by-2 sub-matrix of Σ_{t+1}^{-}. In this experiment, we set $c = 0.95$.

To find the corresponding feature point in the next image frame, we apply the SSD (sum of squared differences) correlation method by sliding a square correlation window within the search region. The size of window has to be large enough to correctly characterize the surrounding area of feature point. Experimentally, we found that the size of correlation window has to be at least 9*9 to reliably identify a feature point (which is about the maximum size of black tapes marking the corners of polyhedron). Larger window size (e.g. 15*15) provides slightly better results, but the computational time may increase significantly.

3. Computing gain matrix
   The gain matrix K is a weighting factor that determines the contribution of the actual measurement z_{t+1} and the predication Hs_{t+1}^{-} to the posterior state estimate s_{t+1}. Gain matrix can be computed by the following formula:

$$K_{t+1} = \Sigma_{t+1}^{-} H^T \left(H \Sigma_{t+1}^{-} H^T + R \right)^{-1} \quad (6)$$

4. Posterior state and covariance estimation
   The posterior state estimation is the combination of the state predication s_{t+1}^{-} and the measurement z_{t+1}:

$$s_{t+1} = s_{t+1}^{-} + K_{t+1} \left(z_{t+1} - Hs_{t+1}^{-} \right) \quad (7)$$

   The posterior covariance estimation can be obtained by:

$$\Sigma_{t+1} = \left(I - K_{t+1} H \right) \Sigma_{t+1}^{-}$$

   In order to avoid the numerical problem, it is better to use the following equation:

$$\Sigma_{t+1} = \left(I - K_{t+1} H \right) \Sigma_{t+1}^{-} \left(I - K_{t+1} H \right)^T + K_{t+1} R K_{t+1}^{\ T} \quad (8)$$

After each prediction and measurement update pair, Kalman filter recursively conditions current estimate on all of the past measurements. This process is repeated using the previous posterior estimates as the new prior estimate. The trace of the state covariance matrix is often used as an indicator on the uncertainty of the estimated position.

In order for the Kalman filter to work, it needs to be initialized. The Kalman filter is activated after the feature is detected in the first two frames i and $i+1$. The initial vector s_0 can be specified as:

$$x_0 = x_{i+1},\ y_0 = y_{i+1},\ v_{x,0} = x_{i+1} - x_i,\ v_{y,0} = y_{i+1} - y_i \quad (9)$$

The initial covariance matrix Σ_0 can be defined as: $\Sigma_0 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$. Σ_0 is usually initialized to very large values. It should decrease and reach a stable state after a few iterations. We also need to initialize the system and measurement error covariance matrices Q and R. The standard deviation from positional system error is 4 in both x and y directions. We further assume that the standard deviation from velocity error is 2 pixels/frame. Therefore, the state covariance matrix can be quantified as:
$Q = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$. Similarly, we can also assume the error for measurement model as 2 pixels in both x and y directions. Thus, $R = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$. Both Q and R are assumed to be constant.

Kalman Filtering has two limitations. One is that it assumes that the state dynamics can be modeled linearly. The other is that it assumes the state vector has uni-modal and is Gaussian distribution, therefore it can not track multiple feature points and require multiple Kalman filters to tracking multiple feature points.

3. Mathematic discussion of Factorization method

Factorization method can be used to recover the 3D shape of 3D objects and their 3D motion relative to the viewing camera based on the motion field estimated from the image sequence. There are two assumptions for factorization method: 1) the camera model is orthographic; 2) there are n non-coplanar 3D points and N images ($N \geq 3$).

Let $p_{ij} = (c_{ij}, r_{ij})$ denote the j-th image point on the i-th image frame. Let $\bar{c}_i$ and $\bar{r}_i$ be the centroid of the image points on the i-th image frame. Let $P_{ij} = (x_j, y_j, z_j)$ be the j-th 3D points relative to the object frame and let $\overline{P}$ be the centroid of the 3D points. Let $c'_{ij} = c_{ij} - \bar{c}_i$, $r'_{ij} = r_{ij} - \bar{r}_i$ and $P'_j = P_j - \overline{P}$.

Due to orthographic projection assumption, we have $\begin{pmatrix} c'_{ij} \\ r'_{ij} \end{pmatrix} = \begin{pmatrix} r_{i,1} \\ r_{i,2} \end{pmatrix} P'_j$, where $r_{i,1}$ and $r_{i,2}$ are the first and second rows of the rotation matrix between camera frame i and object frame. By stacking rows and columns, the equation can be written compactly in the form:

$$W = RS \tag{10}$$

where R is $2N\times 3$ matrix and gives the relative orientation of each frame to the object frame while S is $3\times n$ matrix and contains the 3D coordinates of the feature points.

$$R=\begin{pmatrix} r_{1,1} \\ r_{1,2} \\ \vdots \\ r_{N,1} \\ r_{N,2} \end{pmatrix},\ S=\begin{pmatrix} P_1' & P_2' & \cdots & P_n' \end{pmatrix} \text{ and } W=\begin{pmatrix} c_{11}' & c_{12}' & \cdots & c_{1n}' \\ r_{11}' & r_{12}' & \cdots & r_{1n}' \\ \vdots & \vdots & \vdots & \vdots \\ c_{N1}' & c_{N2}' & \cdots & c_{Nn}' \\ r_{N1}' & r_{N2}' & \cdots & r_{Nn}' \end{pmatrix} \quad (11)$$

According to the rank theorem, the matrix W (often called *registered measurement matrix*) in ideal case has a maximum rank of 3. This is evident since the columns of W are linearly dependent on each other due to orthographic projection assumption and the rotation matrix is identical for all image points in the same frame.

In reality, due to image noise, the rank of W may be greater than 3. To impose the rank theorem, we first perform a SVD on W: $W=UDV^T$. Then, we extract the 3 largest diagonal elements of D (i.e. singular values of W) to form a 3-by-3 diagonal sub-matrix D'. Similarly, we extract the first 3 columns of U and V that are associated with the 3 largest singular values to form U' and V' matrices. These transformations lead to $W'=U'D'V'^T$, where W' is the closest matrix to W satisfying the rank theorem.

Given W', we can perform a decomposition on W' to obtain estimates for S and R. Based on the SVD of $W'=U'D'V'^T$, we have $\hat{R}=U'D'^{1/2}$, and $\hat{S}=D'^{1/2}V'^T$. The solutions of $\hat{R}$ and $\hat{S}$ are however, determined only up to an affine transformation, since for any invertible 3×3 matrix Q, $R=\hat{R}Q$ and $S=Q^{-1}\hat{S}$ also satisfy the equation. We can find matrix Q using the constraints that from the first row, every successive two rows of R are orthonormal, i.e., $\hat{r}_{i,1}QQ^t\hat{r}_{i,1}^t=1$, $\hat{r}_{i,2}QQ^t\hat{r}_{i,2}^t=1$, $\hat{r}_{i,1}QQ^t\hat{r}_{i,2}^t=0$.

Given $i=1,2,\ldots,N$ frames and using the above constraints, we can linearly solve for $A=QQ^t$. Since A is a 3-by-3 symmetric matrix by definition, there are 6 unknowns in A, and for one frame, we have following 3 equations:

$$\begin{pmatrix} \hat{r}_{i,11} & \hat{r}_{i,12} & \hat{r}_{i,13} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \begin{pmatrix} \hat{r}_{i,11} \\ \hat{r}_{i,12} \\ \hat{r}_{i,13} \end{pmatrix} = 1,$$

$$\begin{pmatrix} \hat{r}_{i,21} & \hat{r}_{i,22} & \hat{r}_{i,23} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \begin{pmatrix} \hat{r}_{i,21} \\ \hat{r}_{i,22} \\ \hat{r}_{i,23} \end{pmatrix} = 1, \text{ and,}$$

$$\begin{pmatrix} \hat{r}_{i,11} & \hat{r}_{i,12} & \hat{r}_{i,13} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \begin{pmatrix} \hat{r}_{i,21} \\ \hat{r}_{i,22} \\ \hat{r}_{i,23} \end{pmatrix} = 0 .$$

After rearrangement, we can set up a linear equation:

$$\begin{pmatrix} \hat{r}_{i,11}{}^2 & 2\hat{r}_{i,11}\hat{r}_{i,12} & 2\hat{r}_{i,11}\hat{r}_{i,13} & \hat{r}_{i,12}{}^2 & 2\hat{r}_{i,12}\hat{r}_{i,13} & \hat{r}_{i,13}{}^2 \\ \hat{r}_{i,21}{}^2 & 2\hat{r}_{i,21}\hat{r}_{i,22} & 2\hat{r}_{i,22}\hat{r}_{i,23} & \hat{r}_{i,22}{}^2 & 2\hat{r}_{i,22}\hat{r}_{i,23} & \hat{r}_{i,23}{}^2 \\ \hat{r}_{i,11}\hat{r}_{i,21} & \hat{r}_{i,12}\hat{r}_{i,21} + \hat{r}_{i,11}\hat{r}_{i,22} & \hat{r}_{i,13}\hat{r}_{i,21} + \hat{r}_{i,11}\hat{r}_{i,23} & \hat{r}_{i,12}\hat{r}_{i,22} & \hat{r}_{i,13}\hat{r}_{i,22} + \hat{r}_{i,12}\hat{r}_{i,23} & \hat{r}_{i,13}\hat{r}_{i,23} \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{22} \\ a_{23} \\ a_{33} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

With N frames, A can be solved by least-square method. Then, Q can be obtained via Choleski factorization.

Given Q, the final motion estimate is $R = \hat{R}Q$ and the final structure estimate is $S = Q^{-1}\hat{S}$.

Based on the above analysis, we can summarize the algorithm of factorization method as below:

1. From the tracking results, construct the W matrix.
2. Obtain $\hat{R}$ and $\hat{S}$ matrices through SVD.
3. Apply orthonormal constraints to solve A matrix. Since A is symmetric, there are 6 unknowns for A matrix. The elements of A can be solved by the least-square method.
4. Obtain Q matrix through Choleski factorization.
5. Compute R and S by $R = \hat{R}Q$ and $S = Q^{-1}\hat{S}$.

4. Experiment procedure and results

We have implemented Kalman filter and factorization method in MatLab. We have done the experiments on a sequence of moving polyhedron. The following describes the experiment procedure and results.

1. Kalman filter initialization

   We set Φ, H, Q and R constant matrices and initialize them as:

$$\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \; H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}; \; Q = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}; \; R = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}.$$

We set initial covariance matrix $\Sigma_0 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$.

We manually identify the nine vertices of the polyhedron at the first two frames. They are (from left to right, from top to bottom):

{(118, 78), (117, 89), (122, 158), (232, 92), (254,80), (273,79), (273, 163), (297, 69), (296, 138)} and
{(115, 78), (114, 89), (120, 158), (229, 91), (251,80), (270,78), (270, 163), (295, 69), (295, 138)}

So we set the initial state vectors as:

s_0^1	[115, 78, -3, 0]
s_0^2	[114, 89, -3, 0]
s_0^3	[120, 158, -2, 0]
s_0^4	[229, 91, -3, -1]
s_0^5	[251, 80, -3, 0]
s_0^6	[270, 78, -3, -1]
s_0^7	[270, 163, -3, 0]
s_0^8	[295, 69, -2, 0]
s_0^9	[295, 138, -1, 0]

2. Tracking for each of nine points

After the Kalman filter for each feature point is initialized, we can track the point from frame to frame till the last frame according to the recursive steps summarized in section 2 and by using the formula (3) (4) (5) (6) (7) (8) (9). The outputs (the final state vectors) of Kalman filter at each frame are shown in the table below. After compare the state vectors to the images, we observe that the Kalman filter can track the feature point quite accurately.

S1				S2			
x	y	v_x	v_y	x	y	v_x	v_y
113.0000	79.0000	-3.0000	0.0000	113.0000	90.0000	-3.0000	-1.0000
110.0000	79.0000	-3.0000	0.0000	110.0000	89.9724	-3.0000	-0.8276
107.0000	79.0000	-3.0000	0.0000	107.0000	90.0648	-3.0000	-0.3196
104.0000	79.0000	-3.0000	0.0000	104.0000	89.7452	-3.0000	-0.3196
100.1015	79.0000	-3.3587	0.0000	100.1015	90.3241	-3.3587	0.0391
95.8484	79.0000	-3.6981	0.0000	95.8484	90.3632	-3.6981	0.0391
91.2573	79.0000	-4.0306	0.0000	91.2573	90.4023	-4.0306	0.0391

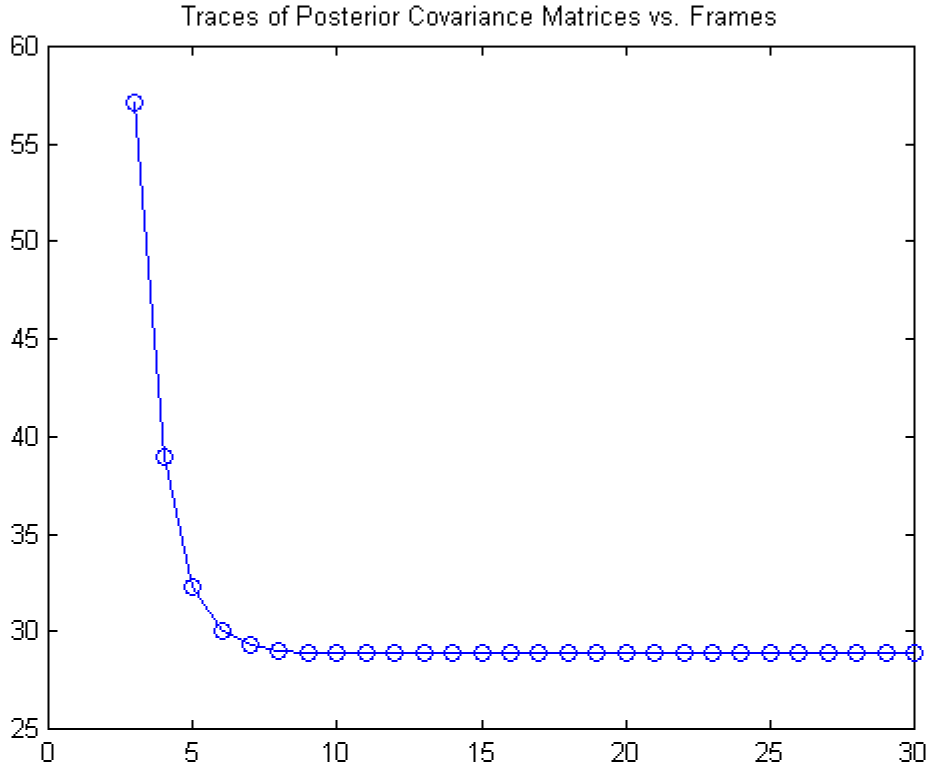
86.3344	79.0000	-4.3605	0.0000	87.2268	90.4414	-4.0306	0.0391
81.9738	79.0000	-4.3605	0.0000	83.1962	90.4806	-4.0306	0.0391
76.7212	79.0000	-4.6893	0.0000	78.2735	89.6276	-4.3593	-0.2896
72.0320	79.0000	-4.6893	0.0000	73.0221	90.2300	-4.6879	0.0390
67.3427	79.0000	-4.6893	0.0000	68.3342	90.2690	-4.6879	0.0390
61.7614	79.0000	-5.0178	0.0000	62.7542	90.3080	-5.0165	0.0390
56.7436	79.0000	-5.0178	0.0000	57.7377	90.3470	-5.0165	0.0390
51.7257	79.0000	-5.0178	0.0000	52.7212	90.3860	-5.0165	0.0390
46.7079	79.0000	-5.0178	0.0000	47.7048	90.4250	-5.0165	0.0390
42.5822	79.0000	-4.6893	0.0000	43.5804	90.4640	-4.6879	0.0390
38.7850	79.0000	-4.3607	0.0000	39.7845	89.6110	-4.3594	-0.2895
35.3163	79.0000	-4.0322	0.0000	36.3172	90.2135	-4.0308	0.0390
32.1762	79.0000	-3.7036	0.0000	33.1784	90.2525	-3.7023	0.0390
29.3646	79.0000	-3.3751	0.0000	30.3681	90.2915	-3.3738	0.0390
25.9895	79.0000	-3.3751	0.0000	26.9944	90.3305	-3.3738	0.0390
22.6144	79.0000	-3.3751	0.0000	23.6206	90.3695	-3.3738	0.0390
20.1314	79.0000	-3.0466	0.0000	21.1389	90.4085	-3.0452	0.0390
17.0848	79.0000	-3.0466	0.0000	18.0937	90.4475	-3.0452	0.0390
14.0383	79.0000	-3.0466	0.0000	15.0485	90.4865	-3.0452	0.0390
10.9917	79.0000	-3.0466	0.0000	12.0033	90.5255	-3.0452	0.0390
7.9452	79.0000	-3.0466	0.0000	8.9581	90.5645	-3.0452	0.0390
5.7907	79.0000	-2.7180	0.0000	6.8049	90.6035	-2.7167	0.0390
S3				S4			
x	y	v_x	v_y	x	y	v_x	v_y
119.0000	158.0000	-3.0000	0.0000	231.0000	91.0000	-3.0000	0.0000
116.0000	158.0000	-3.0000	0.0000	228.0000	90.0276	-3.0000	-0.1724
113.0000	158.0000	-3.0000	0.0000	225.0000	89.8552	-3.0000	-0.1724
110.0000	158.0000	-3.0000	0.0000	221.0919	89.6828	-3.4095	-0.1724
106.1015	158.0000	-3.3587	0.0000	216.7839	89.5103	-3.7682	-0.1724
101.8484	158.0000	-3.6981	0.0000	212.1213	90.2324	-4.1076	0.1670
98.1503	158.0000	-3.6981	0.0000	207.1208	89.5065	-4.4401	-0.1654
94.4521	158.0000	-3.6981	0.0000	201.7884	89.3410	-4.7700	-0.1654
89.8618	158.0000	-4.0272	0.0000	197.0183	89.1756	-4.7700	-0.1654
85.8346	158.0000	-4.0272	0.0000	192.2483	89.0102	-4.7700	-0.1654
81.8075	158.0000	-4.0272	0.0000	186.5862	88.8447	-5.0986	-0.1654
77.7803	158.8921	-4.0272	0.3286	181.4876	88.6793	-5.0986	-0.1654
73.7531	159.2206	-4.0272	0.3286	175.4969	88.5139	-5.4272	-0.1654
69.7259	158.6571	-4.0272	0.0000	169.1777	89.2405	-5.7557	0.1631
65.6988	158.6572	-4.0272	0.0000	163.4220	89.4036	-5.7557	0.1631
61.6716	159.5492	-4.0272	0.3286	157.6663	88.6747	-5.7557	-0.1654
57.6444	159.8778	-4.0272	0.3286	152.8026	88.5092	-5.4272	-0.1654
53.6172	160.2064	-4.0272	0.3286	148.2675	89.2358	-5.0986	0.1631
49.5900	159.6429	-4.0272	0.0000	144.0609	89.3990	-4.7701	0.1631
46.4549	159.6429	-3.6986	0.0000	140.1829	88.6700	-4.4415	-0.1654
42.7563	159.6429	-3.6986	0.0000	135.7413	88.5046	-4.4415	-0.1654
39.9497	160.5350	-3.3701	0.3286	131.2998	90.1233	-4.4415	0.4917
36.5796	161.7556	-3.3701	0.6571	127.7503	89.7229	-4.1130	0.1631
34.1016	163.3048	-3.0415	0.9857	123.6373	89.8860	-4.1130	0.1631
31.0600	163.3984	-3.0415	0.6571	119.5243	90.0491	-4.1130	0.1631
28.0185	163.1634	-3.0415	0.3286	117.1954	90.2122	-3.4559	0.1631
24.9769	163.4920	-3.0415	0.3286	113.7395	90.3753	-3.4559	0.1631
21.9354	162.9285	-3.0415	0.0000	111.1757	89.6464	-3.1274	-0.1654
19.7859	163.8206	-2.7130	0.3286	108.0483	90.3730	-3.1274	0.1631

S5				S6			
x	y	v_x	v_y	x	y	v_x	v_y
251.0000	78.0000	-4.0000	0.0000	271.0000	78.0000	-3.0000	-1.0000
247.9724	77.0276	-3.8276	-0.1724	268.0000	77.0000	-3.0000	-1.0000
244.1448	76.8552	-3.8276	-0.1724	265.0000	76.9199	-3.0000	-0.4920
240.3172	76.6828	-3.8276	-0.1724	261.0919	77.3361	-3.4095	-0.0825
235.5912	76.5103	-4.1863	-0.1724	256.7839	77.2536	-3.7682	-0.0825
230.5105	77.2324	-4.5257	0.1670	252.1213	76.2766	-4.1076	-0.4220
225.9847	76.5065	-4.5257	-0.1654	247.1208	74.9617	-4.4401	-0.7544
221.4590	76.3410	-4.5257	-0.1654	241.7884	74.2073	-4.7700	-0.7544
216.0411	75.2834	-4.8548	-0.4945	236.1262	73.4529	-5.0991	-0.7544
211.1864	73.8968	-4.8548	-0.8232	231.0271	71.8064	-5.0991	-1.0831
205.4395	73.9657	-5.1834	-0.4946	225.9281	71.6153	-5.0991	-0.7545
199.3641	74.3632	-5.5119	-0.1660	220.8290	71.7529	-5.0991	-0.4260
192.9601	74.1971	-5.8405	-0.1660	214.8379	72.2190	-5.4276	-0.0974
187.1196	74.0311	-5.8405	-0.1660	209.4103	72.1216	-5.4276	-0.0974
181.2791	73.8651	-5.8405	-0.1660	203.0906	72.0242	-5.7562	-0.0974
175.4386	73.6991	-5.8405	-0.1660	197.3344	71.9268	-5.7562	-0.0974
168.7061	73.5330	-6.1690	-0.1660	190.6862	71.8293	-6.0847	-0.0974
163.4291	74.2591	-5.8405	0.1625	185.4936	71.7319	-5.7562	-0.0974
157.5886	74.4216	-5.8405	0.1625	179.7374	71.6345	-5.7562	-0.0974
152.6402	73.6920	-5.5119	-0.1660	175.7654	71.5371	-5.0991	-0.0974
148.9123	73.5260	-4.8549	-0.1660	171.5584	72.3318	-4.7705	0.2311
144.0575	75.1441	-4.8549	0.4911	166.7878	71.6709	-4.7705	-0.0974
139.2026	75.6351	-4.8549	0.4911	162.0173	72.4655	-4.7705	0.2311
135.2398	76.1262	-4.5263	0.4911	157.2468	71.8046	-4.7705	-0.0974
130.7135	75.7252	-4.5263	0.1625	153.3683	71.7072	-4.4420	-0.0974
127.0792	75.8877	-4.1978	0.1625	148.9263	71.6098	-4.4420	-0.0974
123.7735	76.0502	-3.8692	0.1625	145.3764	71.5124	-4.1134	-0.0974
120.7963	76.2127	-3.5407	0.1625	141.2630	72.3070	-4.1134	0.2311
118.1477	76.3752	-3.2121	0.1625	138.0416	71.6461	-3.7849	-0.0974
S7				S8			
x	y	v_x	v_y	x	y	v_x	v_y
271.0000	164.0000	-3.0000	0.0000	295.0000	67.0000	-2.0000	-2.0000
268.0000	164.0000	-3.0000	0.0000	293.0000	65.9724	-2.0000	-1.8276
265.0000	164.0000	-3.0000	0.0000	291.0000	65.9847	-2.0000	-0.8116
262.0000	164.0000	-3.0000	0.0000	289.0000	65.1731	-2.0000	-0.8116
258.1015	164.0000	-3.3587	0.0000	286.1015	64.3615	-2.3587	-0.8116
253.8484	164.0000	-3.6981	0.0000	282.8484	62.6555	-2.6981	-1.1510
249.2573	164.0000	-4.0306	0.0000	280.1503	61.5045	-2.6981	-1.1510
244.3344	164.0000	-4.3605	0.0000	277.4521	61.2459	-2.6981	-0.8211
239.0817	164.0000	-4.6896	0.0000	273.8618	60.4248	-3.0272	-0.8211
234.3921	164.0000	-4.6896	0.0000	270.8346	58.7117	-3.0272	-1.1498
228.8104	164.0000	-5.0182	0.0000	267.8075	57.5619	-3.0272	-1.1498
223.7922	164.0000	-5.0182	0.0000	264.7803	57.3042	-3.0272	-0.8212
218.7741	164.0000	-5.0182	0.0000	261.7531	57.3750	-3.0272	-0.4927
213.7559	164.0000	-5.0182	0.0000	258.7259	56.8823	-3.0272	-0.4927
208.7377	164.0000	-5.0182	0.0000	255.6988	57.2817	-3.0272	-0.1641
202.8274	164.8921	-5.3467	0.3285	252.6716	57.1176	-3.0272	-0.1641
198.3727	165.2206	-5.0182	0.3285	249.6444	56.9535	-3.0272	-0.1641
193.3545	164.6571	-5.0182	0.0000	246.6172	56.7894	-3.0272	-0.1641
189.2284	164.6571	-4.6896	0.0000	243.5900	56.6252	-3.0272	-0.1641
184.5388	164.6571	-4.6896	0.0000	241.4549	57.3532	-2.6986	0.1644

180.7412	164.6571	-4.3611	0.0000	238.7563	56.6255	-2.6986	-0.1641
177.2721	165.5492	-4.0326	0.3285	236.0577	57.3535	-2.6986	0.1644
172.3475	166.7698	-4.3611	0.6571	233.3590	56.6258	-2.6986	-0.1641
167.0943	168.3189	-4.6896	0.9856	229.7683	57.3538	-3.0272	0.1644
163.2968	169.3045	-4.3611	0.9856	226.7411	56.6261	-3.0272	-0.1641
158.9356	170.2902	-4.3611	0.9856	224.6060	57.3541	-2.6986	0.1644
154.5745	171.2758	-4.3611	0.9856	222.7995	56.6264	-2.3701	-0.1641
151.1055	172.2614	-4.0326	0.9856	221.3214	57.3544	-2.0415	0.1644
147.0729	173.2471	-4.0326	0.9856	219.2799	56.6267	-2.0415	-0.1641

S9			
x	y	v_x	v_y
295.0000	137.0000	-1.0000	-1.0000
293.0276	136.9724	-1.1724	-0.8276
290.9352	137.0648	-1.6804	-0.3196
289.2548	136.7452	-1.6804	-0.3196
286.6759	137.3241	-2.0391	0.0391
284.6368	137.3632	-2.0391	0.0391
282.5977	137.4023	-2.0391	0.0391
279.6662	137.4414	-2.3691	0.0391
277.2971	137.4806	-2.3691	0.0391
274.9281	136.6276	-2.3691	-0.2896
271.6669	137.2300	-2.6977	0.0390
268.9693	137.2690	-2.6977	0.0390
266.2716	137.3080	-2.6977	0.0390
262.6819	137.3470	-3.0262	0.0390
259.6556	137.3860	-3.0262	0.0390
256.6294	137.4250	-3.0262	0.0390
253.6032	137.4640	-3.0262	0.0390
250.5770	136.6110	-3.0262	-0.2895
248.4428	137.2135	-2.6977	0.0390
245.7452	137.2525	-2.6977	0.0390
243.9395	137.2915	-2.3691	0.0390
241.5704	137.3305	-2.3691	0.0390
240.0933	137.3695	-2.0406	0.0390
238.0528	137.4085	-2.0406	0.0390
236.0122	137.4475	-2.0406	0.0390
233.9716	137.4865	-2.0406	0.0390
231.9310	136.6334	-2.0406	-0.2895
229.8904	138.1280	-2.0406	0.3675
227.8498	138.4955	-2.0406	0.3675

The figure below shows the plot of the mean traces of posterior covariance matrices Σ versus the number of frames. From the curve, it is obvious that the trace reduces and eventually becomes constant as the number of frames increases. This means that the area of the search region around the predicted position becomes smaller and smaller as the Kalman filter gains confidence on its prediction. Hence, the search region can be limited to smaller area. Notice that the values of the posterior covariance matrix, and thus its trace, become constant after a couple of iterations because the Kalman filter has been sufficiently trained to find the right value for its gain. The final covariance matrix is equal to



3. Recovering shape and structure from motion

From the outputs of the nine Kalman filters of all the frames, we can estimate the polyhedron's shape and motion (rotation) by the factorization method introduced in section 3. Following the steps summarized in section 3, we can obtain R and S. The following tables show R and S in the format as the matrices definition (equation 11), where every two rows of R represent the rotation of each frame, and every column of S represents the 3D coordinates of one feature point.

R matrix:

-0.9823	-0.0009	0.1779
-0.0010	0.9374	0.0032
-0.9852	-0.0018	0.1730
0.0024	0.9454	0.0001
-0.9874	-0.0018	0.1666
0.0029	0.9466	-0.0001
-0.9879	0.0044	0.1548
0.0039	0.9496	0.0037
-0.9923	0.0013	0.1441
0.0050	0.9523	0.0045
-0.9946	0.0084	0.1316
0.0083	0.9599	0.0058
-0.9953	0.0091	0.1189
0.0120	0.9686	0.0018
-0.9955	0.0092	0.1056
0.0155	0.9770	-0.0022

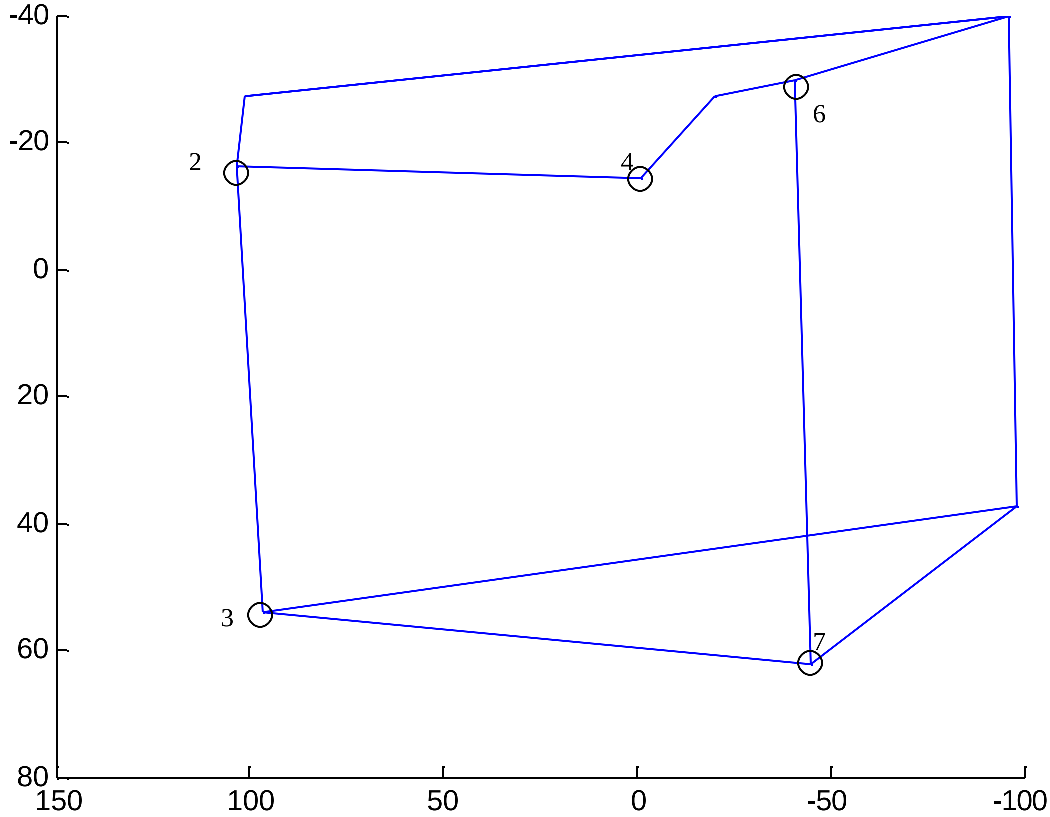
-0.9964	0.0101	0.0917
0.0182	0.9832	-0.0029
-0.9975	0.0114	0.0790
0.0220	0.9916	-0.0062
-0.9954	0.0081	0.0634
0.0233	0.9948	-0.0034
-0.9991	0.0107	0.0484
0.0248	0.9988	-0.0020
-0.9994	0.0190	0.0310
0.0241	0.9974	-0.0004
-1.0000	0.0247	0.0156
0.0252	1.0001	0.0004
-0.9989	0.0269	-0.0009
0.0250	0.9996	-0.0010
-1.0015	0.0298	-0.0180
0.0263	1.0096	0.0000
-0.9991	0.0355	-0.0347
0.0253	1.0094	-0.0001
-0.9989	0.0343	-0.0469
0.0246	1.0031	0.0005
-0.9975	0.0331	-0.0586
0.0253	1.0050	-0.0024
-0.9955	0.0253	-0.0701
0.0255	1.0053	-0.0032
-0.9960	0.0259	-0.0834
0.0240	1.0023	-0.0005
-0.9947	0.0331	-0.0954
0.0254	1.0117	0.0060
-0.9921	0.0373	-0.1130
0.0259	1.0168	0.0087
-0.9886	0.0381	-0.1268
0.0279	1.0307	0.0106
-0.9880	0.0368	-0.1375
0.0262	1.0381	0.0112
-0.9873	0.0363	-0.1489
0.0245	1.0418	0.0135
-0.9879	0.0325	-0.1563
0.0236	1.0481	0.0169
-0.9855	0.0231	-0.1690
0.0206	1.0493	0.0189
-0.9810	0.0215	-0.1778
0.0198	1.0546	0.0208

S matrix:

101.174	103.543	96.562	-1.143	-19.987	-40.822	-44.899	-96.458	-97.971
-27.063	-15.943	54.309	-14.384	-27.295	-29.503	62.441	-39.773	37.210
-19.599	-7.703	-17.671	58.151	72.967	69.991	48.270	-98.017	-106.388

From R matrix, we can observe that the rotation matrices for each frame are almost the same (every two rows are almost the same). This means the movement of polyhedron is almost just translation; there is little rotation movement. This makes the orthographic projection almost hold, and we expect we can reconstruct 3D shape of polyhedron well.

From S matrix, we can reconstruct the 3D shape of the polyhedron as shown below. From the figure, we observe that the reconstruction is quite successful. However, we can only reconstruct the 3D shape up to a scale factor. The ratio between line 2-3, line 2-4 and line 6-7 is 1: 1.74: 1.33 (the theoretical ratio is 1: 1.65: 1.10).



5. Summary

In this project, we have implemented a tracking algorithm based on Kalman filter and a factorization method to recover the shape and motion of a polyhedron. The results showed that the tracking algorithm was capable to correctly track the feature points (i.e. corners of the polyhedron) from one frame to another. In addition, we have verified that the orthographic projection assumption was valid in our case to apply the factorization method, as the structure of the polyhedron was relatively flat and the motion of the polyhedron was almost purely translational (notice that the values of rotation matrices were nearly constant for all the frames). Therefore, the 3D shape of the polyhedron could be successfully reconstructed.