Robust Digital Image Stabilization Using the Kalman Filter

ISL Lab Seminar
Han-Sol Kang

INDEX

01 Introduction

02 Camera Trajectory Model

03 FP-Based DIS Algorithm

04 Experimental Results and Analysis

05 Conclusion
Introduction

🌟 Image Stabilization

[Hand held]

[Mounted on vehicle]
Introduction

Image Stabilization
Introduction

Type of Image Stabilization

- Image Stabilization (IS)
  - Optical IS (OIS)
    - Electric IS (EIS)
      - Digital IS (DIS)
        - Block Matching Algorithm (BMA)
        - Feature Based Matching (FBM)
Camera Trajectory Model

Camera Trajectory : Represents the intentional motion. Camera shake : Noise version of intentional motion.

[Dynamic equation]

\[ X_t = AX_{t-1} + W_t \]
\[ Z_t = HX_t + V_t \]

\( X_t (t = 0, 1, \cdots) \) : the state at time \( t \) that represents the camera position, velocity, and acceleration
\( Z_t (t = 0, 1, \cdots) \) : the observed camera position at time \( t \)

\( A, H \) : the state transition matrix and the measurement one, respectively

Assume that \( W_t \sim N(0, Q_t) \) and \( V_t \sim N(0, R_t) \) are white Gaussian noise that are independent of each other.

\( X \) : Intentional motion
\( Z \) : Global motion
02 Camera Trajectory Model

Camera Trajectory Model

[Initial expected value]
\[ \hat{X}_0^+ = E(X_0) \]
\[ P_0^+ = E[(X_0 - \hat{X}_0^+)(X_0 - \hat{X}_0^+)^T] \]

[Prediction step]
\[ \hat{X}_t^- = A\hat{X}_{t-1} \]
\[ P_t^- = AP_{t-1}^+A^T + Q_{t-1} \]

[Correction step]
\[ K_t = P_t^-H^T[H P_t^- H^T + R_t]^{-1} \]
\[ P_t^+ = [I - K_t H] P_t^- \]
\[ \hat{X}_t^+ = \hat{X}_t^- + K_t[Z_t - H\hat{X}_t^-] \]

Enter prior estimate \( \hat{X}_0^- \) and its error covariance \( P_0^- \)

Project ahead:
\[ \hat{x}_{k+1}^- = \phi_k \hat{x}_k \]
\[ P_{k+1}^- = \phi_k P_k \phi_k^T + Q_k \]

Compute Kalman gain:
\[ K_k = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \]

Update estimate with measurement \( z_k \):
\[ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \]

Compute error covariance for updated estimate:
\[ P_k = (I - K_k H_k) P_k \]

Figure 4.1 © John Wiley & Sons, Inc. All rights reserved.
Camera Trajectory Model

Motion Model

[Affine transform]

\[
\begin{bmatrix}
x'
y'
z'
1
\end{bmatrix} = 
\begin{bmatrix}
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x
y
z
1
\end{bmatrix}
\]

- Rotation
- Translation
- Scaling
- Shearing

Rigid Body Transformation: Translation + Rotation

Similarity Transformation: Translation + Rotation + Uniform scaling

Affine Transformation: Translation + Rotation + Scaling + Shearing

[Rigid Body Transformation (2-D)]

\[
\begin{bmatrix}
x_t \\
y_t \\
\end{bmatrix} = 
\begin{bmatrix}
\cos \theta_t & -\sin \theta_t \\
\sin \theta_t & \cos \theta_t
\end{bmatrix}
\begin{bmatrix}
x_{t-1} \\
y_{t-1}
\end{bmatrix} + 
\begin{bmatrix}
T_x^t \\
T_y^t
\end{bmatrix}
\]

\[U_t = F_tU_{t-1} + T_t\]

Accumulative motion model

\[
U_t = \left( \prod_{k=1}^{t} F_k \right) U_0 + \sum_{k=1}^{t} \left( \prod_{j=k+1}^{t} F_j \right) T_k = F_t^A U_0 + T_t^A
\]

where

\[
\prod_{j=k+1}^{t} F_j = \begin{pmatrix}
\cos \theta' & -\sin \theta' \\
\sin \theta' & \cos \theta'
\end{pmatrix}
\]

\[\theta' = \sum_{j=k+1}^{t} \theta_j\]
Camera Trajectory Model

**Design of the KF**

[Rotation angle at time $t$]

$$\hat{\theta}_t = \hat{\theta}_{t-1} + n_t$$

$n_t \sim N(0, \sigma_r)$: white noise with variance $\sigma_r$

[Translation (assume constant velocity)]

$$\hat{T}^v_t = \hat{T}^v_{t-1} + n^v_t$$

$$\hat{T}^y_t = \hat{T}^y_{t-1} + \hat{T}^y_t$$

$n^v_t \sim N(0, \sigma^v_r)$: white noise with variance $\sigma^v_r$

$$\begin{bmatrix} \hat{\theta}_t \\ \hat{T}^x_t \\ \hat{T}^{xy}_t \\ \hat{T}^y_t \\ \hat{T}^{yy}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{\theta}_{t-1} \\ \hat{T}^x_{t-1} \\ \hat{T}^{xy}_{t-1} \\ \hat{T}^y_{t-1} \\ \hat{T}^{yy}_{t-1} \end{bmatrix} + \begin{bmatrix} n^\theta_t \\ n^x_t \\ n^{xy}_t \\ n^y_t \end{bmatrix}$$

$$Z_t = HX_t + V_t \rightarrow \begin{bmatrix} \theta^o_t \\ T^o_{tx} \\ T^o_{ty} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta^o_t \\ n^\theta_t \\ n^{oT}_t \end{bmatrix}$$

The auxiliary variable $\hat{T}^v_t$ cannot be observed in $Z_t$

Superscript "o" denotes the variables for observed

$n^o_{\theta} \sim N(0, \sigma^o_r)$ and $n^o_T \sim N(0, \sigma^o_v)$ are white Gaussian noise
Camera Trajectory Model

Design of the KF

\[ Q_t = \text{diag}(\sigma_r,0,\sigma_T^v,0,\sigma_T^v) \]

\[ R_t = \text{diag}(\sigma_r,\sigma_T^o,\sigma_T^o) \]

\[ \hat{X}_0^+ = (0,0,0,0,0)^T \]

\[ P_0^+ = \text{diag}(\sigma_r,\sigma_T^o,\sigma_T^v,\sigma_T^v) \]

\[ \text{diag}(\cdot) : \text{diagonal matrix} \]

\[ [Z_t, \text{accumulative motion}] \]

\[ \theta_t^o = \sum_{j=1}^{t} \theta_j = \theta_{t-1}^o + \tilde{\theta}_t \]

where \( \prod_{j=k+1}^{t} F_j = \begin{pmatrix} \cos \theta^o & -\sin \theta^o \\ \sin \theta^o & \cos \theta^o \end{pmatrix} \)

\[ \theta = \sum_{j=1}^{t} \theta_j \]

\[ T_t^o = \sum_{k=1}^{t} \left( \prod_{j=k+1}^{t} F_j \right) T_k = F_t T_{t-1}^o + \tilde{T}_t \]

\( \tilde{\theta}_t \) and \( \tilde{T}_t \) are computed using

\[ x_{t-1} = \begin{pmatrix} \cos \theta_{t-1} & -\sin \theta_{t-1} \\ \sin \theta_{t-1} & \cos \theta_{t-1} \end{pmatrix} \]

\[ y_{t-1} = \begin{pmatrix} \cos \theta_{t-1} & -\sin \theta_{t-1} \\ \sin \theta_{t-1} & \cos \theta_{t-1} \end{pmatrix} \]

\[ \sum_{j=1}^{t} \theta_j + \tilde{T}_t \]
FB-Based DIS Algorithm

The process of the proposed DIS algorithm

1. Initialize
2. Select good FPs
   - Track FPs using KF-based prediction
   - Scene Changed?
     - Yes
     - Estimate motions
     - No
3. Adaptively update Kalman filter
   - Compensate unintentional motions
   - Last Frame?
     - Yes
     - Finish
FB-Based DIS Algorithm

Selection of Good FPs

\[ G = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} I_x^2(x, y) & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y^2(x, y) \end{bmatrix} \]

\((p_x, p_y)\) : the point \((x, y)\)

\(w_x, w_y\) : the searching window size

\(I_x(x, y), I_y(x, y)\) : spatial derivative of the image frame in the horizontal (vertical) direction

\[ \min(\lambda_1, \lambda_2) > \lambda_{thresh} \]

\(\lambda_1, \lambda_2\) : eigen-values of \(G\)

\(\lambda_{thresh}\) : generally set as \(\lambda_{thresh} = r\lambda_{\max}(0 < r < 1)\)

\(\lambda_{\max}\) : maximum eigen-value obtained from eigen-values of all points
FB-Based DIS Algorithm

**FP Tracking Using KF-Based Prediction**

[Current camera motion \( \tilde{M} \)]

\[
\tilde{M}_t = [\tilde{\theta}_t, \tilde{T}_t]^T
\]

is obtained by solving the equation

\[
M^o_t = M^o_{t-1} + \tilde{M}_t
\]

where the related parameters \( M^o_t = [\theta^o_t, T^o_t]^T \) and \( M^o_{t-1} = [\theta^o_{t-1}, T^o_{t-1}]^T \) are extracted from \( \hat{X}_t^- \) and \( \hat{X}_{t-1}^+ \), respectively

\[
\theta^o_t = \theta^o_{t-1} + \tilde{\theta}_t
\]

\[
T^o_t = F T^o_{t-1} + \tilde{T}_t
\]

where \( \tilde{\theta}_t, \tilde{T}_t \) are obtained by solving the equation

\[
M^o_t = M^o_{t-1} + \tilde{M}_t
\]

\[
[x_t, y_t] = \begin{bmatrix}
\cos \theta_t & -\sin \theta_t \\
\sin \theta_t & \cos \theta_t
\end{bmatrix} \begin{bmatrix}
x_{t-1} \\
y_{t-1}
\end{bmatrix} + \begin{bmatrix}
T^x_t \\
T^y_t
\end{bmatrix}
\]

\[
(x^i_t, y^i_t)
\]

is predicted by performing

\[
(x^i_t, y^i_t) \quad (1 \leq i \leq N_0)
\]

Finally, the **predicted FP** instead of the previous one \((x^i_{t-1}, y^i_{t-1})\) is set as the initial position for the correspondence tracking, which reduces the searching range and thus speeds up the tracking process
FB-Based DIS Algorithm

Detection of Scene Change

Assume that $N_t$ pairs of FPs are obtained after finding the correspondence.

$N_t \geq N_{thresh} : \text{no scene change}$

If scene change is detected,

- Current image frame
- new reference frame
- FPs : re-selection
- Parameter of KF : re-initialization

[Irregular condition and the lack of features]

$N_t < N_{thresh} : \text{scene change??} \ X$

$\Delta t \geq \Delta T_{thresh} : \text{reference frame change} \ \ \Delta t : \text{time difference between the current detection of scene change and the previous one}$
FB-Based DIS Algorithm

🌟 Adaptive Measurement-Update of KF

\[ R_t = \text{diag}(\sigma_r^o, \sigma_T^o, \sigma_T^o) \times \]

the practical camera shake at different time may be greatly different

We develop a new scheme, called and adaptive KF

the camera shake results in the sign change of the estimated motion \( \tilde{M}_t \) offset by their mean values

camera shakes : period 1 > period 2

zero-crossing cases : period 1 > period 2

\[
R_t = \begin{cases} 
R_{t4} & \text{if } C_t > CT_4 \\
R_{t3} & \text{if } CT_3 < C_t \leq CT_4 \\
R_{t2} & \text{if } CT_2 < C_t \leq CT_3 \\
R_{t1} & \text{if } CT_1 < C_t \leq CT_2 \\
R_{t0} & \text{otherwise}
\end{cases}
\]

\( C_t \): the number of zero-crossing

\( R_n(i=0\cdots4) \): measurement noise with different variance

\( CT_i(i=1\cdots4) \): predefined threshold
FB-Based DIS Algorithm

**Compensation of Unintentional Motion**

The global motion $M_t^o$ contains the intentional motion $\hat{M}_t$ and the unintentional one $M_t^c$.

$M_t^c$ should be computed first.

Accumulative motion model

$$U_t = \left( \prod_{k=1}^{t} F_k \right) U_0 + \sum_{k=1}^{t} \left( \prod_{j=k+1}^{t} F_j \right) T_k = F_t^A U_0 + T_t^A$$

$$U_t^o = F_t^{Ao} U_o + T_t^{Ao}$$

$$\hat{U}_t = \hat{F}_t^A U_o + \hat{T}_t^A$$

$$\hat{U}_t = \hat{F}_t^{Ac} U_t^o + (\hat{T}_t^A - F_t^{Ac} T_t^{Ao})$$

**Unintentional motion**

$$\theta_t^c = \theta_t^o - \hat{\theta}_t$$

$$T_t^c = \hat{T}_t^A - F_t^{Ac} T_t^{Ao} = \hat{T}_t - \begin{bmatrix} \cos \theta_t^c & -\sin \theta_t^c \\ \sin \theta_t^c & \cos \theta_t^c \end{bmatrix} T_t^{Ao}$$

**Compensated image frame**

$$I_t^c \begin{bmatrix} x \\ y \end{bmatrix} = I_t \begin{bmatrix} \cos \theta_t^c & -\sin \theta_t^c \\ \sin \theta_t^c & \cos \theta_t^c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} T_t^{cx} \\ T_t^{cy} \end{bmatrix}$$
Experimental Results and Analysis

Sample image

(a) Island  (b) Road  (c) University

(a) : small motions less than 2 pixels and high frequent jiggles.
(b) : small motion less than or equal to 1 pixel and irregular conditions.
(c) : small motions less than 2 pixels and high frequent rotational and translational jiggles.

Parameter

\[ r = 0.1, \quad N_0 = 100, \quad N_{\text{thresh}} = 5, \quad \Delta T_{\text{thresh}} = 10, \quad \Delta T_p = 10, \quad p = 20 \]

Adaptive KF

\[ \Delta T_p = 10, \quad CT_1 = 1, \quad CT_2 = 3, \quad CT_3 = 5, \quad CT_4 = 7, \quad R_1 = 4R_{\tau_0}, \quad R_2 = 20R_{\tau_0}, \quad R_3 = 40R_{\tau_0}, \quad R_4 = 80R_{\tau_0} \]

where \( R_{\tau_0} \) depends on specific applications and is usually set as \( R_{\tau_0} = 1.0 \)
Experimental Results and Analysis

Example

- Effectiveness of Motion Prediction With KF

<table>
<thead>
<tr>
<th>Video Seq.</th>
<th>With MP</th>
<th>Without MP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Island</td>
<td>33.33</td>
<td>31.66</td>
</tr>
<tr>
<td>Road</td>
<td>31.66</td>
<td>31.66</td>
</tr>
<tr>
<td>University</td>
<td>31.94</td>
<td>30.55</td>
</tr>
</tbody>
</table>

10 times
avg : over 100 frames

(a) : Original image frames
(b) : Stabilized image frames
**Experimental Results and Analysis**

**Performance of Adaptive KF**

(a) Performance with adaptive KF

(b) Performance without adaptive KF

(1) high-frequency jiggles exist in the horizontal direction

(2) the amount of jiggles between the 30 and 50th frames changes frequently

(a) : Performance with adaptive KF

(b) : Performance without adaptive KF
Experimental Results and Analysis

Computational Complexity

<table>
<thead>
<tr>
<th>Video Seq.</th>
<th>100 frames (ms)</th>
<th>150 frames (ms)</th>
<th>210 frames (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Island</td>
<td>30.0</td>
<td>33.3</td>
<td>38.1</td>
</tr>
<tr>
<td>Road</td>
<td>31.6</td>
<td>33.3</td>
<td>38.0</td>
</tr>
<tr>
<td>University</td>
<td>31.3</td>
<td>40.0</td>
<td>39.9</td>
</tr>
</tbody>
</table>

Computational time in ms/frame unit

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Pixel ratio</th>
<th>C-GIM (ms)</th>
<th>C-UM (ms)</th>
<th>Speed (fps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>720x480</td>
<td>4.5</td>
<td>8.6</td>
<td>28.8</td>
<td>26.23</td>
</tr>
<tr>
<td>640x480</td>
<td>4</td>
<td>10.0</td>
<td>24.3</td>
<td>29.08</td>
</tr>
<tr>
<td>320x240</td>
<td>1</td>
<td>2.8</td>
<td>6.5</td>
<td>106.78</td>
</tr>
</tbody>
</table>
05 Conclusion

☆ We proposed a **new DIS algorithm** which can obtain a good stabilized performance in **real time**.
   (KLT tracker + Kalman filter)

☆ The intentional motion is obtained with the proposed **adaptive Kalman filter**

☆ The developed algorithm has **better robustness against irregular conditions** than the conventional ones.