Efficient Real-time Video Stabilization with A Novel Least Squares Formulation

Jianwei Ke   Alex J Watras   Jae-Jun Kim   Hewei Liu
Hongrui Jiang   Yu Hen Hu

Department of Electrical and Computer Engineering
University of Wisconsin-Madison

ICASSP, June 2021

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Content

- Most existing video stabilization methods are offline. However, there is an increasing demand (robotics, drone, etc.) of real-time online methods.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Content

- Most existing video stabilization methods are offline. However, there is an increasing demand (robotics, drone, etc.) of real-time online methods.
- Some real-time stabilization methods have high latency as they require future frames for path planning.

Content

- Most existing video stabilization methods are offline. However, there is an increasing demand (robotics, drone, etc.) of real-time online methods.
- Some real-time stabilization methods have high latency as they require future frames for path planning.
- Other real-time methods rely heavily on Kalman filter where the covariance matrix and the state transfer matrix must be assumed.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Content

- Most existing video stabilization methods are offline. However, there is an increasing demand (robotics, drone, etc.) of real-time online methods.
- Some real-time stabilization methods have high latency as they require future frames for path planning.
- Other real-time methods rely heavily on Kalman filter where the covariance matrix and the state transfer matrix must be assumed.
- We present a simple and efficient real-time online video stabilization algorithm(LSstab) with minimum latency.

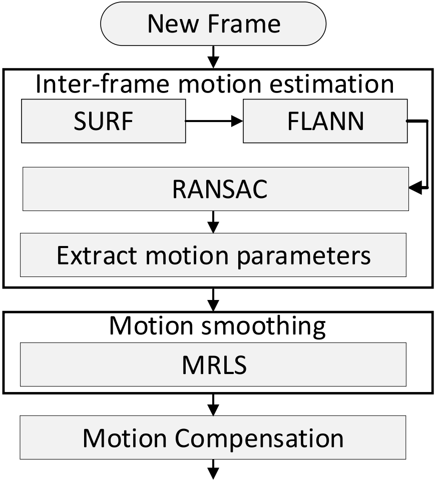
Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Content

- Most existing video stabilization methods are offline. However, there is an increasing demand (robotics, drone, etc.) of real-time online methods.
- Some real-time stabilization methods have high latency as they require future frames for path planning.
- Other real-time methods rely heavily on Kalman filter where the covariance matrix and the state transfer matrix must be assumed.
- We present a simple and efficient real-time online video stabilization algorithm(LSstab) with minimum latency.
- LSstab is based on a novel Least squared formulation of the smoothing cost function. A recursive solver is derived for optimizing the cost function.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Algorithm Flowchart



Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Least Squared based Motion Smoothing

- Assume the inter-frame motion model is the simplified affine transformation $A = \begin{bmatrix} s \cdot \cos(\theta) & -s \cdot \sin(\theta) & t_x \\ s \cdot \sin(\theta) & s \cdot \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}$

Least Squared based Motion Smoothing

- Assume the inter-frame motion model is the simplified affine transformation $A = \begin{bmatrix} s \cdot \cos(\theta) & -s \cdot \sin(\theta) & t_x \\ s \cdot \sin(\theta) & s \cdot \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}$
- s, θ, t_x, t_y are the scale, rotation, and translations in x, y axis.

Least Squared based Motion Smoothing

- Assume the inter-frame motion model is the simplified affine transformation $A = \begin{bmatrix} s \cdot \cos(\theta) & -s \cdot \sin(\theta) & t_x \\ s \cdot \sin(\theta) & s \cdot \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}$
- s, θ, t_x, t_y are the scale, rotation, and translations in x, y axis.
- Once A is estimated by RANSAC, the four parameters are extracted and will be smoothed.

Least Squared based Motion Smoothing

- Start by considering motion smoothing as

$$\underset{\overline{m}_1,\ldots,\overline{m}_n}{\arg\min} \sum_{i=1}^{n}(m_i - \overline{m}_i)^2 + \lambda \sum_{i=2}^{n}(\overline{m}_i - \overline{m}_{i-1})^2 \tag{1}$$

where m_i and $\overline{m}_i$ are the original and smoothed motion for the i^{th} frame.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Least Squared based Motion Smoothing

- Start by considering motion smoothing as

$$\underset{\overline{m}_1,\ldots,\overline{m}_n}{\arg\min} \sum_{i=1}^{n} (m_i - \overline{m}_i)^2 + \lambda \sum_{i=2}^{n} (\overline{m}_i - \overline{m}_{i-1})^2 \tag{1}$$

  where m_i and $\overline{m}_i$ are the original and smoothed motion for the i^{th} frame.
- Find all stable motions $\overline{m}_1, \ldots, \overline{m}_n$ at once. Real-time solver needed.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Least Squared based Motion Smoothing

- Start by considering motion smoothing as

$$\arg\min_{\overline{m}_1,\ldots,\overline{m}_n} \sum_{i=1}^{n} (m_i - \overline{m}_i)^2 + \lambda \sum_{i=2}^{n} (\overline{m}_i - \overline{m}_{i-1})^2 \tag{1}$$

  where m_i and $\overline{m}_i$ are the original and smoothed motion for the i^{th} frame.
- Find all stable motions $\overline{m}_1, \ldots, \overline{m}_n$ at once. Real-time solver needed.
- Even it can be solved in real-time, fail to capture the relationship between successive solutions $(\overline{m}_1^n, \ldots, \overline{m}_n^n)$ and $(\overline{m}_1^{n+1}, \ldots, \overline{m}_{n+1}^{n+1})$

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

A direct approach

- A direct approach

$$\overline{m}_n = \arg\min_{\overline{m}_n} \sum_{i=1}^{n-1} (m_i - \widehat{m}_i)^2 + (m_n - \overline{m}_n)^2 + \\ \lambda \sum_{i=2}^{n} (\widehat{m}_i - \widehat{m}_{i-1})^2 + \lambda(\overline{m}_n - \widehat{m}_{n-1})^2 \tag{2}$$

where $\widehat{m}_1, \ldots, \widehat{m}_{n-1}$ are the previous smoothed motions. $\overline{m}_n$ is the current smoothed estimate.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

A direct approach

- A direct approach

$$\overline{m}_n = \arg\min_{\overline{m}_n} \sum_{i=1}^{n-1} (m_i - \widehat{m}_i)^2 + (m_n - \overline{m}_n)^2 + \lambda \sum_{i=2}^{n} (\widehat{m}_i - \widehat{m}_{i-1})^2 + \lambda(\overline{m}_n - \widehat{m}_{n-1})^2 \quad (2)$$

where $\widehat{m}_1, \ldots, \widehat{m}_{n-1}$ are the previous smoothed motions. $\overline{m}_n$ is the current smoothed estimate.

- We can further simplify (2) to:

$$\overline{m}_n = \arg\min_{\overline{m}_n} (m_n - \overline{m}_n)^2 + \lambda(\overline{m}_n - \widehat{m}_{n-1})^2 \quad (3)$$

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

A direct approach

- We can further simplify (2) to:

$$\overline{m}_n = \arg\min_{\overline{m}_n} (m_n - \overline{m}_n)^2 + \lambda(\overline{m}_n - \widehat{m}_{n-1})^2 \tag{3}$$

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

A direct approach

- We can further simplify (2) to:

$$\overline{m}_n = \arg\min_{\overline{m}_n}(m_n - \overline{m}_n)^2 + \lambda(\overline{m}_n - \widehat{m}_{n-1})^2 \tag{3}$$

- Since (3) is quadratic, the minimum occurs when the derivative is zero. The solution is

$$\overline{m}_n = \frac{m_n + \lambda\widehat{m}_{n-1}}{1+\lambda} \tag{4}$$

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

A direct approach

- We can further simplify (2) to:

$$\overline{m}_n = \arg\min_{\overline{m}_n} (m_n - \overline{m}_n)^2 + \lambda(\overline{m}_n - \widehat{m}_{n-1})^2 \tag{3}$$

- Since (3) is quadratic, the minimum occurs when the derivative is zero. The solution is

$$\overline{m}_n = \frac{m_n + \lambda \widehat{m}_{n-1}}{1 + \lambda} \tag{4}$$

- However, (4) only involves the last stabilized motion $\hat{m}_{n-1}$ and the current motion m_n.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

A direct approach

- We can further simplify (2) to:

$$\overline{m}_n = \arg\min_{\overline{m}_n} (m_n - \overline{m}_n)^2 + \lambda(\overline{m}_n - \widehat{m}_{n-1})^2 \quad (3)$$

- Since (3) is quadratic, the minimum occurs when the derivative is zero. The solution is

$$\overline{m}_n = \frac{m_n + \lambda \widehat{m}_{n-1}}{1+\lambda} \quad (4)$$

- However, (4) only involves the last stabilized motion $\hat{m}_{n-1}$ and the current motion m_n.
- Thus has a limited ability to stabilizing the current frame.

Modified Recursive Least Squares Stabilization

- Modified Recursive Least Squares formulation (MRLS)

$$\underset{\overline{m}_1,\ldots,\overline{m}_n}{\arg\min} \sum_{i=1}^{n} (m_i - \overline{m}_i)^2 + \lambda_1 \sum_{i=2}^{n} (\overline{m}_i - \overline{m}_{i-1})^2 + \lambda_2 \sum_{i=1}^{n-1} (\overline{m}_i - \widehat{m}_i)^2 \quad (5)$$

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Modified Recursive Least Squares Stabilization

- Modified Recursive Least Squares formulation (MRLS)

$$\underset{\overline{m}_1,\ldots,\overline{m}_n}{\arg\min} \sum_{i=1}^{n} (m_i - \overline{m}_i)^2 + \lambda_1 \sum_{i=2}^{n} (\overline{m}_i - \overline{m}_{i-1})^2 + \lambda_2 \sum_{i=1}^{n-1} (\overline{m}_i - \widehat{m}_i)^2 \quad (5)$$

- Similar to (1), (5) also tries to estimate an augmented path $(\overline{m}_1, \ldots, \overline{m}_n)$.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Modified Recursive Least Squares Stabilization

- Modified Recursive Least Squares formulation (MRLS)

$$\arg\min_{\overline{m}_1,\ldots,\overline{m}_n} \sum_{i=1}^{n}(m_i-\overline{m}_i)^2+\lambda_1\sum_{i=2}^{n}(\overline{m}_i-\overline{m}_{i-1})^2+\lambda_2\sum_{i=1}^{n-1}(\overline{m}_i-\widehat{m}_i)^2 \quad (5)$$

- Similar to (1), (5) also tries to estimate an augmented path $(\overline{m}_1,\ldots,\overline{m}_n)$.
- $(\overline{m}_1,\ldots,\overline{m}_n)$ is guaranteed to be smooth by the minimization of $\lambda_1\sum_{i=2}^{n}(\overline{m}_i-\overline{m}_{i-1})^2$.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Modified Recursive Least Squares Stabilization

- Modified Recursive Least Squares formulation (MRLS)

$$\arg\min_{\overline{m}_1,\dots,\overline{m}_n} \sum_{i=1}^{n}(m_i - \overline{m}_i)^2 + \lambda_1 \sum_{i=2}^{n}(\overline{m}_i - \overline{m}_{i-1})^2 + \lambda_2 \sum_{i=1}^{n-1}(\overline{m}_i - \widehat{m}_i)^2 \quad (5)$$

- Similar to (1), (5) also tries to estimate an augmented path $(\overline{m}_1, \dots, \overline{m}_n)$.
- $(\overline{m}_1, \dots, \overline{m}_n)$ is guaranteed to be smooth by the minimization of $\lambda_1 \sum_{i=2}^{n}(\overline{m}_i - \overline{m}_{i-1})^2$.
- In (5), $\lambda_2 \sum_{i=1}^{n-1}(\overline{m}_i - \widehat{m}_i)^2$ secures that $(\overline{m}_1, \dots, \overline{m}_n)$ is similar to $(\widehat{m}_1, \dots, \widehat{m}_{n-1})$, making sure that $(\widehat{m}_1, \dots, \widehat{m}_{n-1}, \widehat{m}_n = \overline{m}_n)$ will also form a smooth curve.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

- First attempt

$$\underset{\overline{m}_1,\ldots,\overline{m}_n}{\arg\min} \sum_{i=1}^{n} (m_i - \overline{m}_i)^2 + \lambda \sum_{i=2}^{n} (\overline{m}_i - \overline{m}_{i-1})^2 \tag{1}$$

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

- First attempt

$$\underset{\overline{m}_1,\ldots,\overline{m}_n}{\arg\min} \sum_{i=1}^{n} (m_i - \overline{m}_i)^2 + \lambda \sum_{i=2}^{n} (\overline{m}_i - \overline{m}_{i-1})^2 \quad (1)$$

- Second attempt

$$\overline{m}_n = \underset{\overline{m}_n}{\arg\min} (m_n - \overline{m}_n)^2 + \lambda (\overline{m}_n - \widehat{m}_{n-1})^2 \quad (3)$$

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

- First attempt

$$\underset{\overline{m}_1,\ldots,\overline{m}_n}{\arg\min} \sum_{i=1}^{n} (m_i - \overline{m}_i)^2 + \lambda \sum_{i=2}^{n} (\overline{m}_i - \overline{m}_{i-1})^2 \tag{1}$$

- Second attempt

$$\overline{m}_n = \underset{\overline{m}_n}{\arg\min} (m_n - \overline{m}_n)^2 + \lambda (\overline{m}_n - \widehat{m}_{n-1})^2 \tag{3}$$

- MRLS

$$\underset{\overline{m}_1,\ldots\overline{m}_n}{\arg\min} \sum_{i=1}^{n} (m_i - \overline{m}_i)^2 + \lambda_1 \sum_{i=2}^{n} (\overline{m}_i - \overline{m}_{i-1})^2 + \lambda_2 \sum_{i=1}^{n-1} (\overline{m}_i - \widehat{m}_i)^2 \tag{5}$$

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

An Recursive Solver

- (5) is convex. Setting all partial derivatives to zero results in

$$\underbrace{\begin{bmatrix} (1+\lambda_1+\lambda_2) & -\lambda_1 & 0 & \cdots & \cdots & 0 \\ -\lambda_1 & (1+2\lambda_1+\lambda_2) & -\lambda_1 & \ddots & \ddots & \vdots \\ 0 & -\lambda_1 & (1+2\lambda_1+\lambda_2) & -\lambda_1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -\lambda_1 & (1+2\lambda_1+\lambda_2) & -\lambda_1 \\ 0 & \cdots & \cdots & 0 & -\lambda_1 & (1+\lambda_1) \end{bmatrix}}_{\mathbf{S}_n} \underbrace{\begin{bmatrix} \overline{m}_1 \\ \overline{m}_2 \\ \overline{m}_3 \\ \vdots \\ \overline{m}_{n-1} \\ \overline{m}_n \end{bmatrix}}_{\overline{\mathbf{m}_n}} = \underbrace{\begin{bmatrix} m_1+\lambda_2\widehat{m}_1 \\ m_2+\lambda_2\widehat{m}_2 \\ m_3+\lambda_2\widehat{m}_3 \\ \vdots \\ m_{n-1}+\lambda_2\widehat{m}_{n-1} \\ m_n \end{bmatrix}}_{\mathbf{Y}_n}$$

- The key idea is the last row of S_n^{-1} can be derived from S_{n-1}^{-1} in O(n).

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Theorem 1

Let $\mathcal{S}_{n-1} = [\boldsymbol{S}_{n-1}|I]$ be an augmented matrix, where I is the identity matrix. $\mathcal{S}^e_{n-1} = [S^e_{n-1}|A_{n-1}]$ is the echelon form of $\mathcal{S}_{n-1}$ with $b = -\lambda_1$ as leading term for each row. Then $\{S^e_{n-1}\}_{n-2} \in \mathbb{R}^{2(n-1)}$ and $\{S^e_n\}_n \in \mathbb{R}^{2n}$ have the form:

$$\{\mathcal{S}^e_{n-1}\}_{n-2} = \begin{bmatrix} 0 & \cdots & b & x_1^{(n-1)} & y_1^{(n-1)} & y_2^{(n-1)} & \cdots & y_{n-3}^{(n-1)} & 0 \end{bmatrix} \tag{6}$$

$$\{\boldsymbol{S}_n^{-1}\}_n = \begin{bmatrix} \frac{y_1^{(n-1)}b}{q_n} & \cdots & \frac{y_{n-3}^{(n-1)}b}{q_n} & \frac{-b}{q_n} & \frac{c-x_1^{(n-1)}}{q_n} \end{bmatrix} \tag{7}$$

where $\{\cdot\}_i$ denotes the i^{th} row of a matrix and $a = (1 + \lambda_1 + \lambda_2), b = -\lambda_1, c = (1 + 2\lambda_1 + \lambda_2), d = (\lambda_1 + 1)$.

Timing performance

Table: Overall timing performance of the proposed algorithm on public data set in [6, 8, 18]

Resolution	SURF and FLANN	RANSAC	Motion Smoothing	Total(ms)
320×240	6.83	2.29	0.14	9.26
640×360	8.48	3.30	0.56	12.34
1280×720	17.30	5.28	1.04	23.62

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Stabilization performance

- ITF(Inter-frame Transformation Fidelity) is a popular evaluation metric of stabilization quality.
- ITF defined as

$$ITF = \frac{1}{N-1} \sum_{k=1}^{N-1} PSNR(F_{k+1}, F_k) \tag{8}$$

where N is the total number of frames.

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Stabilization performance

Table 2. Comparison of ITF for different video stabilization methods

	Offline			Real-time		
Videos	[6]	[8]	[18]	[19]	[12]	Ours
18AF	21.33	27.09	27.55	21.54	23.15	**24.98**
Fountain	27.16	32.27	31.87	27.62	28.14	**28.68**
Talking	23.30	25.94	25.97	23.40	23.52	**23.64**
Dancing	22.62	22.40	22.49	21.32	22.96	**21.87**
Traffic	23.74	26.87	26.77	21.59	24.09	**24.18**
Street	22.48	24.32	27.98	21.85	23.78	**24.33**
Park	21.66	27.47	27.49	22.04	23.19	**23.56**
Parking lot	28.81	25.95	27.02	18.52	21.76	**22.38**
Dynamic	21.12	22.04	22.03	20.24	21.42	**21.44**
Degenerate	22.37	23.18	22.63	20.78	22.02	**22.72**

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

More results: https://ieee-dataport.org/open-access/lsstab-results
(DOI: 10.21227/e1dj-g876)

Thank you!