Full-Frame Video Stabilization with Motion Inpainting

Yasuyuki Matsushita,
Eyal Ofek
Weina Ge
Xiaoou Tang
Heung-Yeung Shum

IEEE Trans on PAMI, July 2006

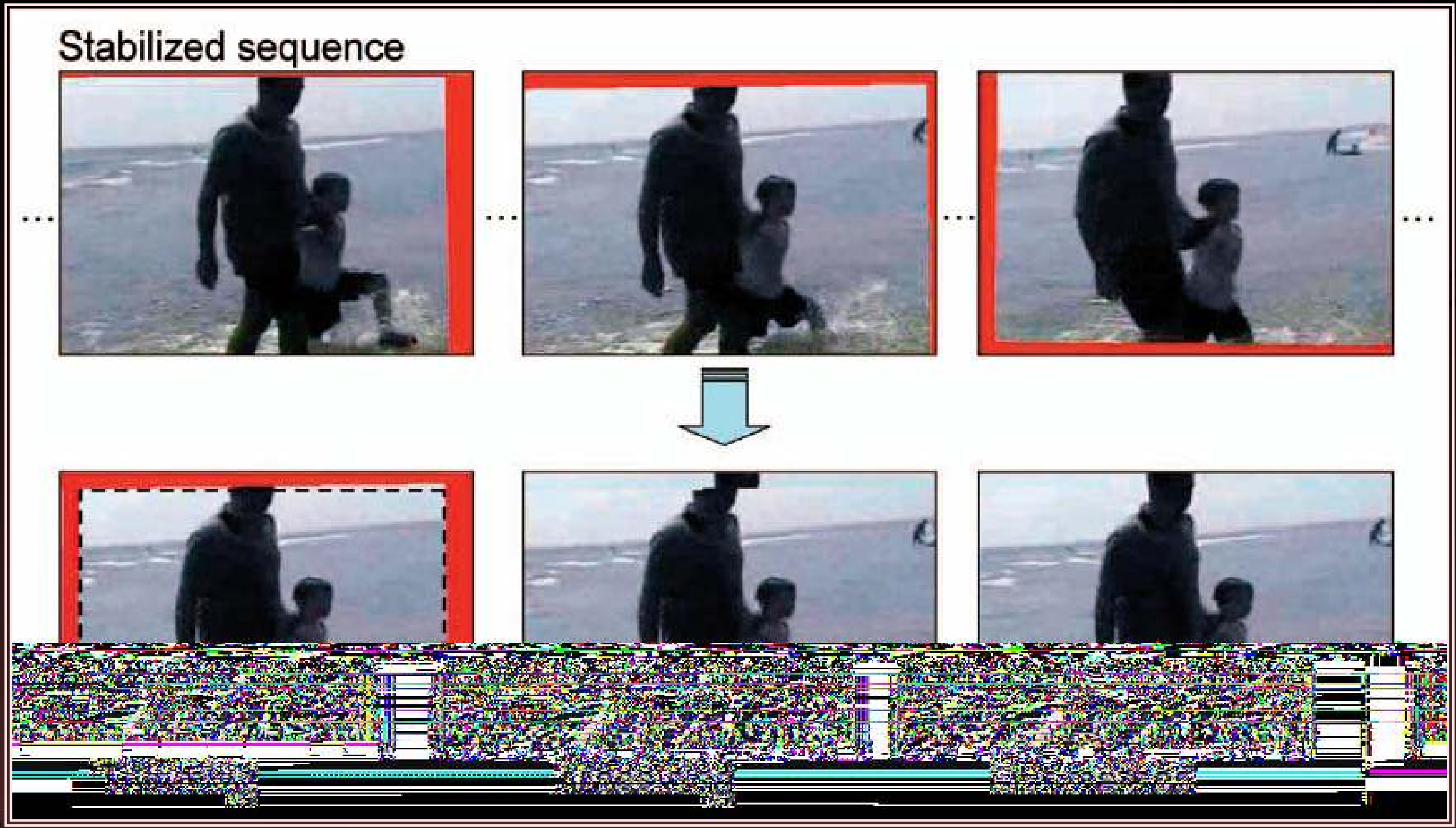
Outline

- Introduction
- Proposed Method
- Experimental results
- Quantitative Evaluation
- Computation Cost
- Conclusion

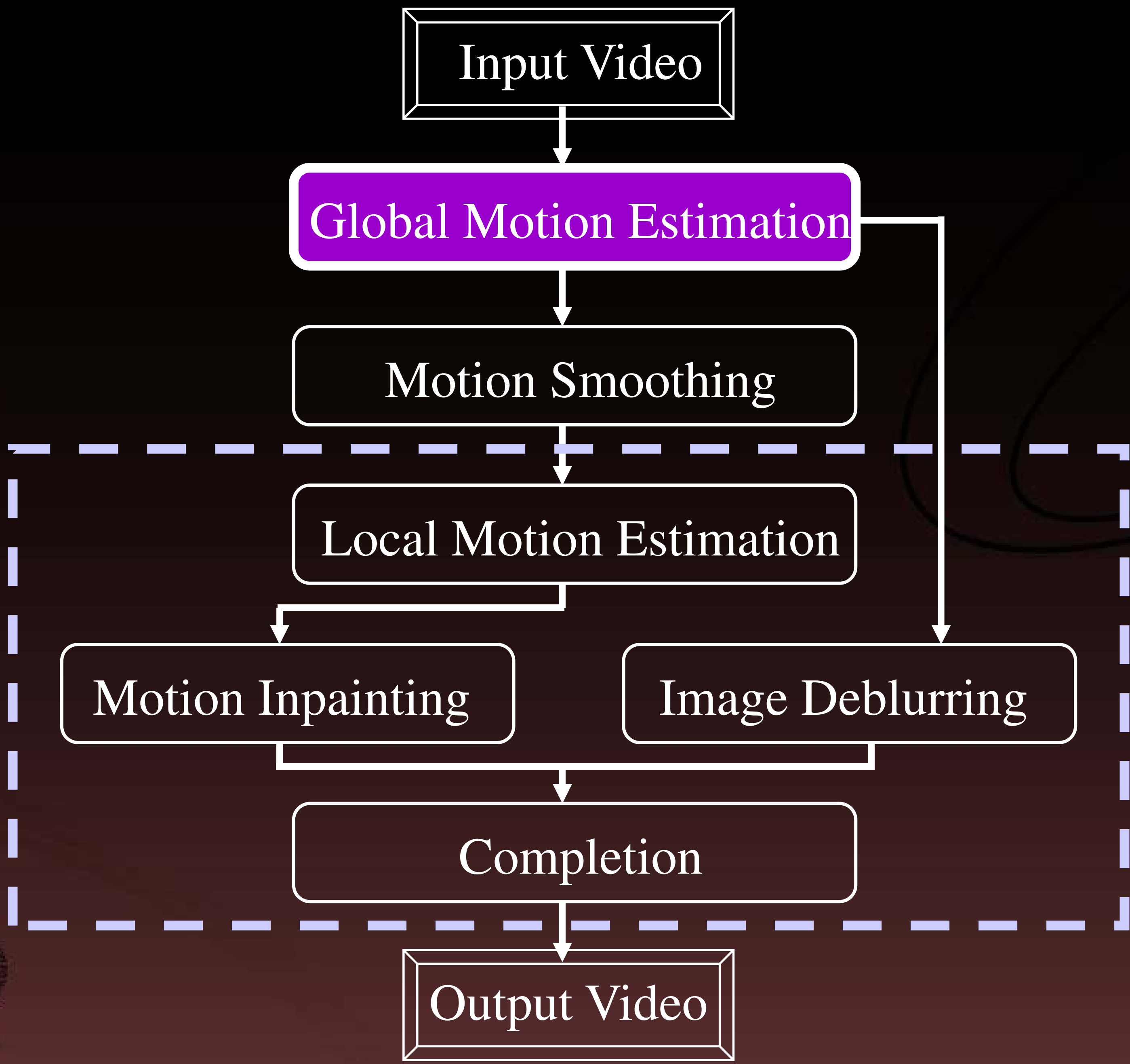
Introduction

- Stabilization：
  - Remove undesirable motion caused by unintentional shake of a human hand.
    - remove high frequency camera motion vs. completely remove camera motion.
    - full frame vs. trimming
    - motion inpainting vs. mosaicing

Prior Work vs. Now



CMLAB NATIONAL TAIWAN UNIVERSITY



Global Motion Estimation

- GM is estimated by aligning pair-wise adjacent frames.
  - $\min_T (I_{t'}(Tp_t) - I_t(p_t))$
- Hierarchical motion estimation
  - construct an image pyramid
  - start from the coarsest level
- By applying the parameter estimation for every pair of adjacent frames, a global transformation chain T_i^j is obtained.

H.-Y. Shum and R. Szeliski, "Construction of Panoramic Mosaics with Global and Local Alignment," Int'l J. Computer Vision, vol. 36, no. 2, pp. 101-130, 2000.

CMLAB NATIONAL TAIWAN UNIVERSITY

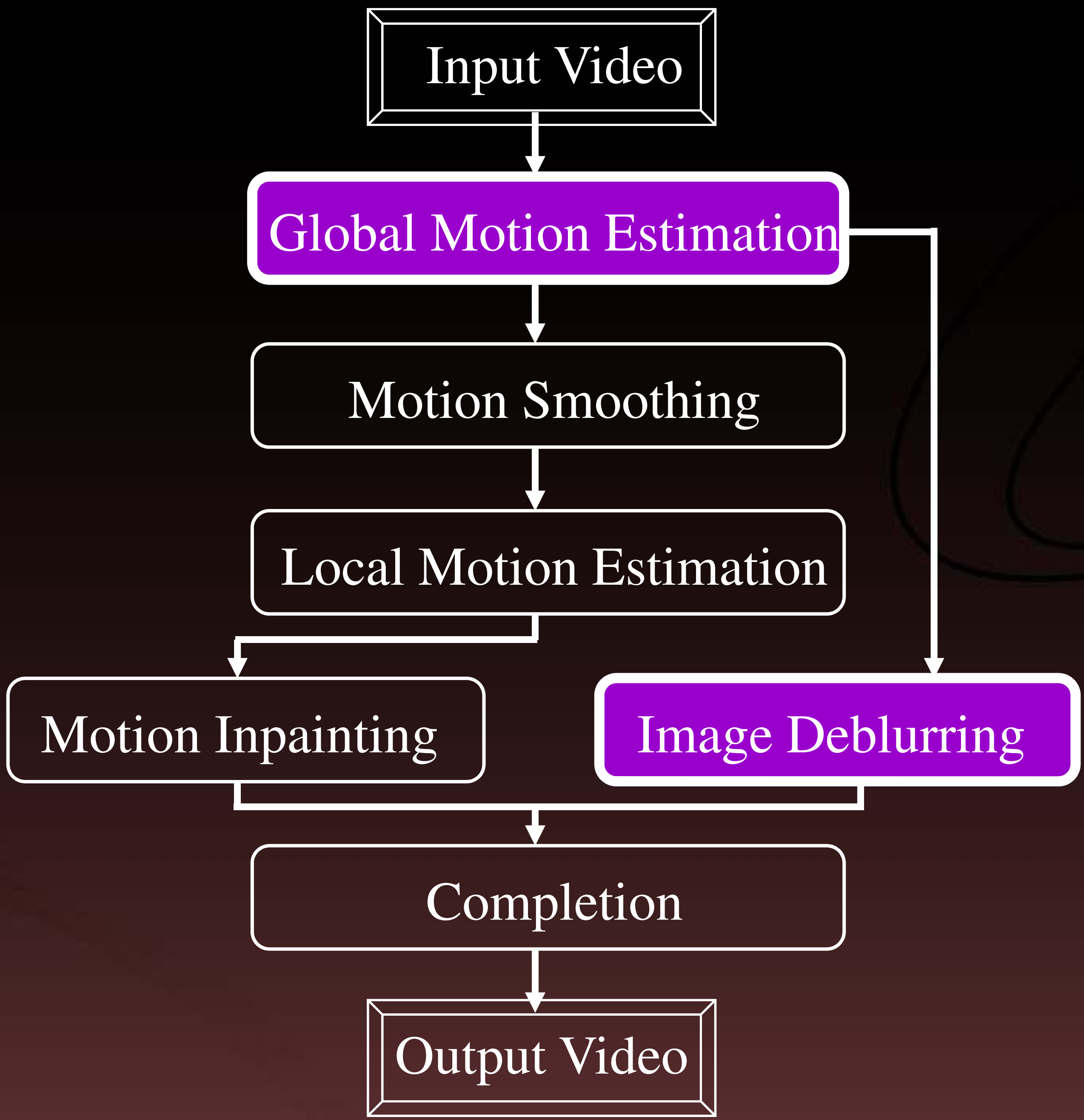


Image Deblurring

- Transferring sharper image pixels from neighboring frames.
  - evaluates the "relative blurriness"

$$b_t = \frac{1}{\sum_{p_t} \{((f_x \otimes I_t)(p_t))^2 + ((f_y \otimes I_t)(p_t))^2\}}$$

  - evaluates the "alignment error"

$$E_{t'}^{t}(p^{t}) = | I_{t'}(T_t^{t'} p_t) - I_t(p_t) |$$

Image Deblurring

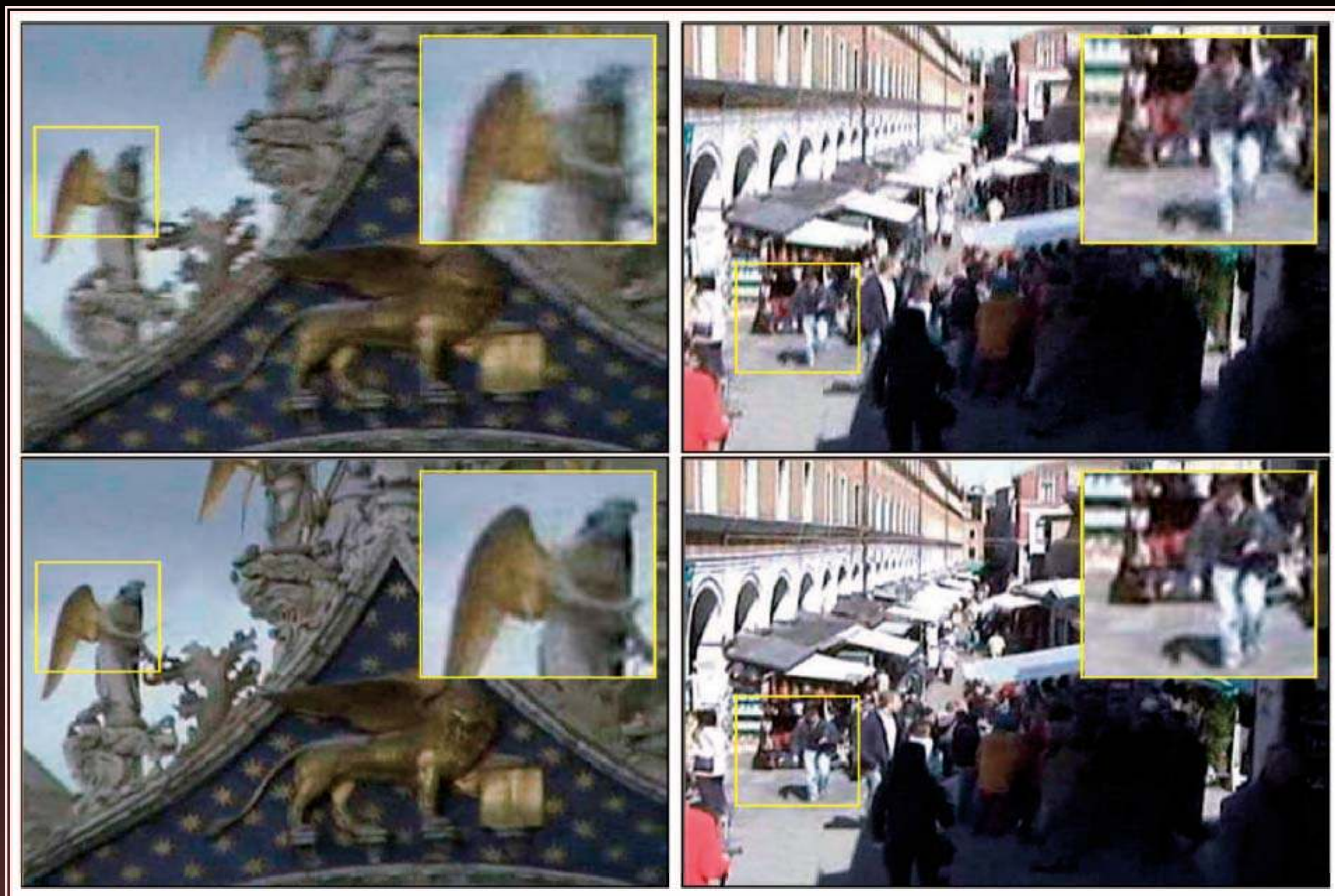
- Blurry pixel are replaced by interpolating shaper pixels.

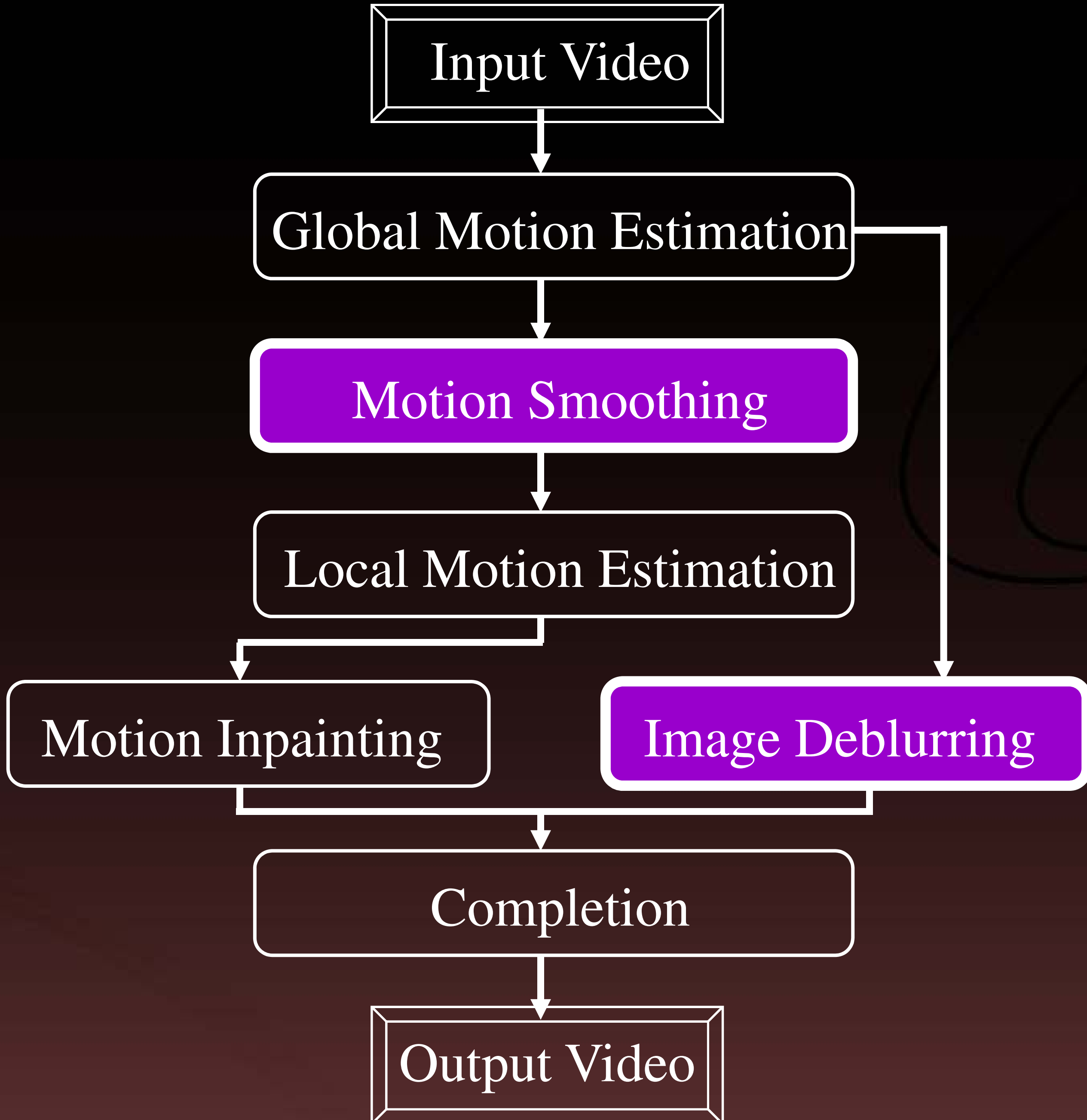
$$\tilde{I}_t(p_t) = \frac{I_t(p_t) + \sum_{t' \in N} w_{t'}^{t}(p_t) I_{t'}(T_t^{t'} p_t)}{1 + \sum_{t' \in N} w_{t'}^{t}(p_t)}$$

- w is the weight factor which consists of the pixel-wise alignment error and relative blurriness

$$w_{t'}^{t}(p_t) = \begin{cases} 0 & \text{if } \frac{b_t}{b_{t'}} < 1 \\ \frac{b_t}{b_{t'}} \frac{\alpha}{E_{t'}^{t}(p_t) + \alpha} & \text{otherwise} \end{cases}$$

Image Deblurring

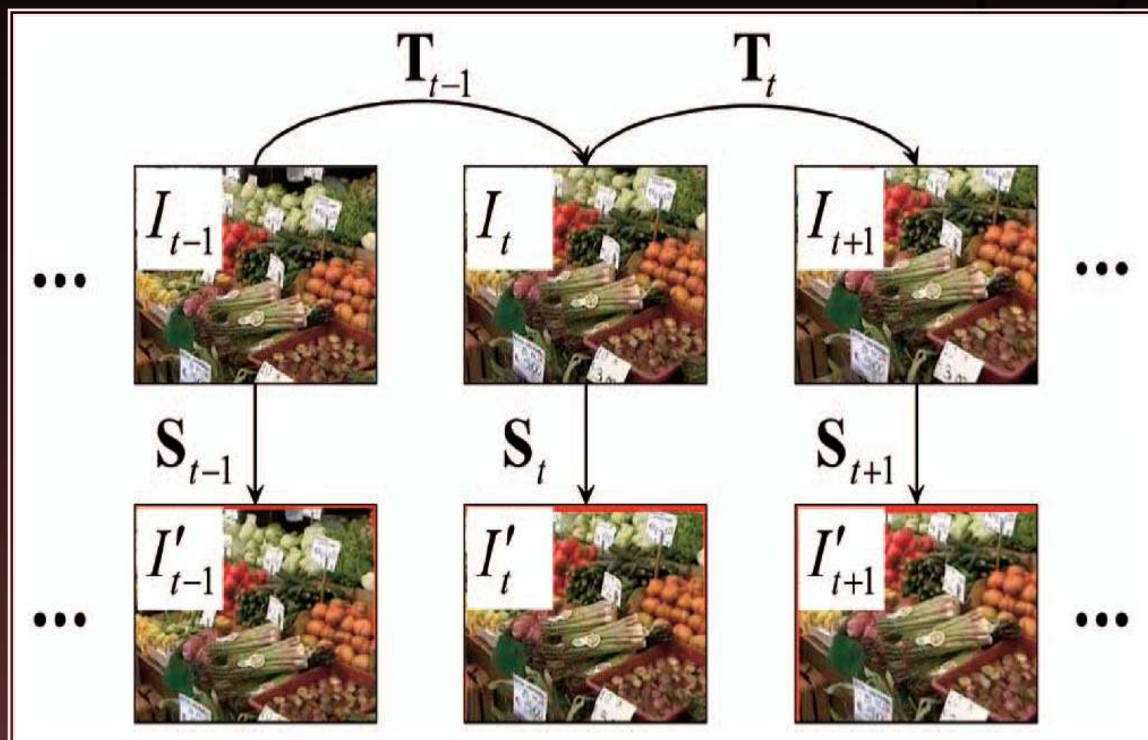



Input Video
Global Motion Estimation
Motion Smoothing
Local Motion Estimation
Motion Inpainting
Image Deblurring
Completion
Output Video

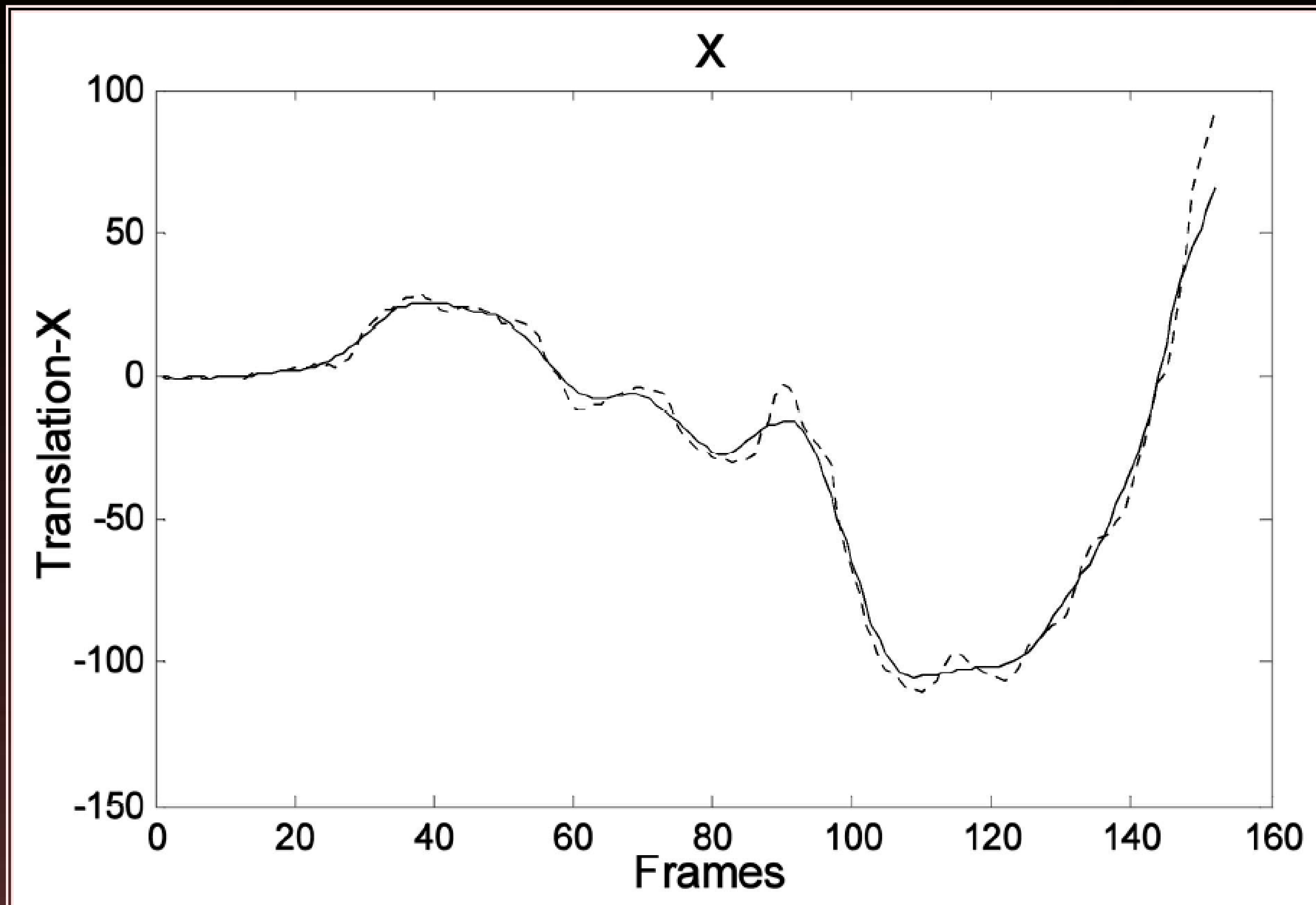
Motion Smoothing

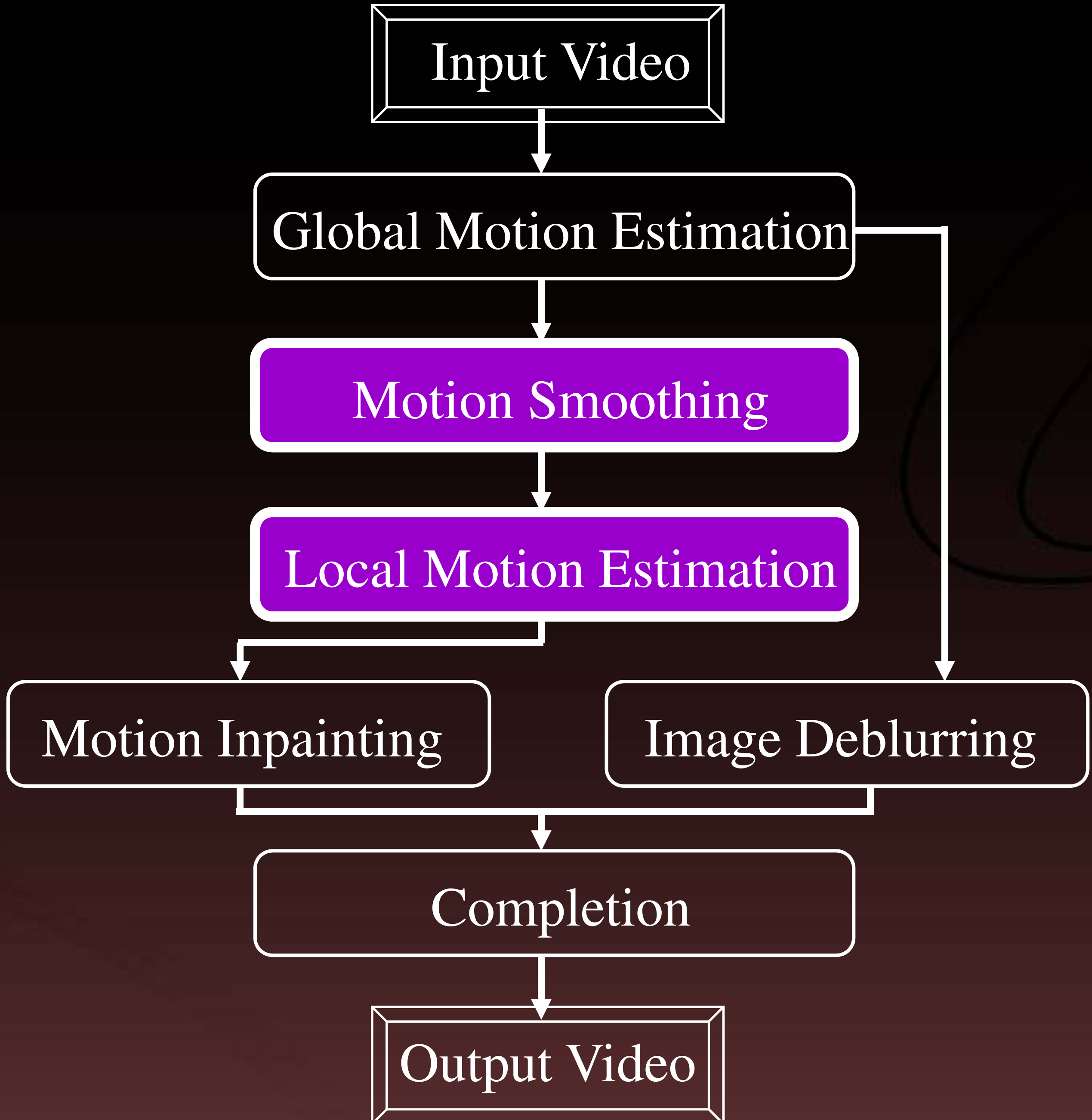
$$S_t = \sum_{i \in N_t} T_t^i \otimes G(k)$$

$$N_t = \{ j : t-k \le j \le t+k \}, \quad G(k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-k^2/2\sigma^2}, \quad \sigma = \sqrt{k}$$



Motion Smoothing



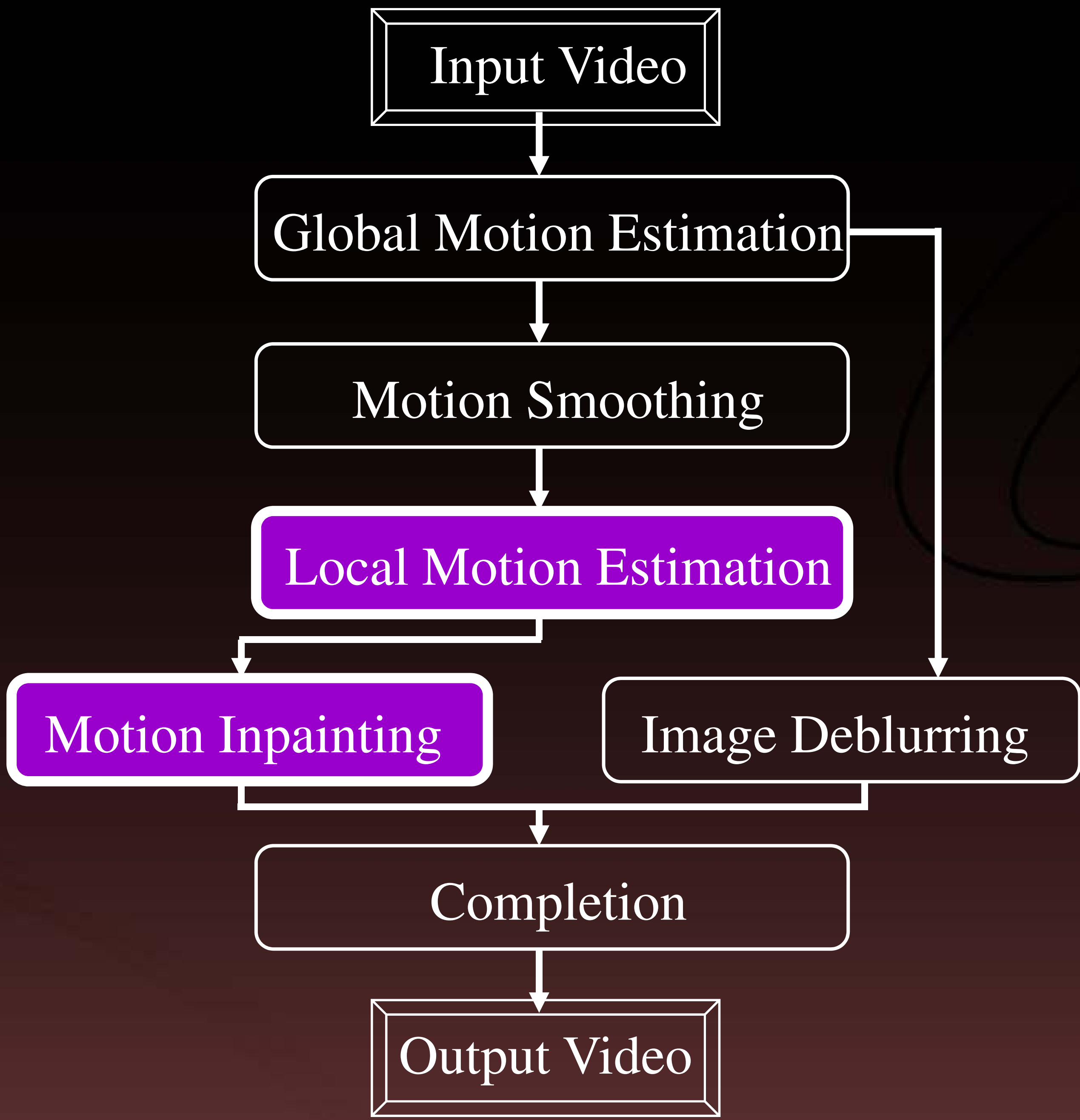


CMLAB NATIONAL TAIWAN UNIVERSITY

Local Motion Estimation

- A pyramidal version of Lucas-Kanade optical flow computation is applied to obtain the local motion field.

J. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm," OpenCV Document, Intel, Microprocessor Research Labs, 2000.


Input Video
Global Motion Estimation
Motion Smoothing
Local Motion Estimation
Motion Inpainting
Image Deblurring
Completion
Output Video

Motion Inpainting

- Mosaicing with consistency constraint.

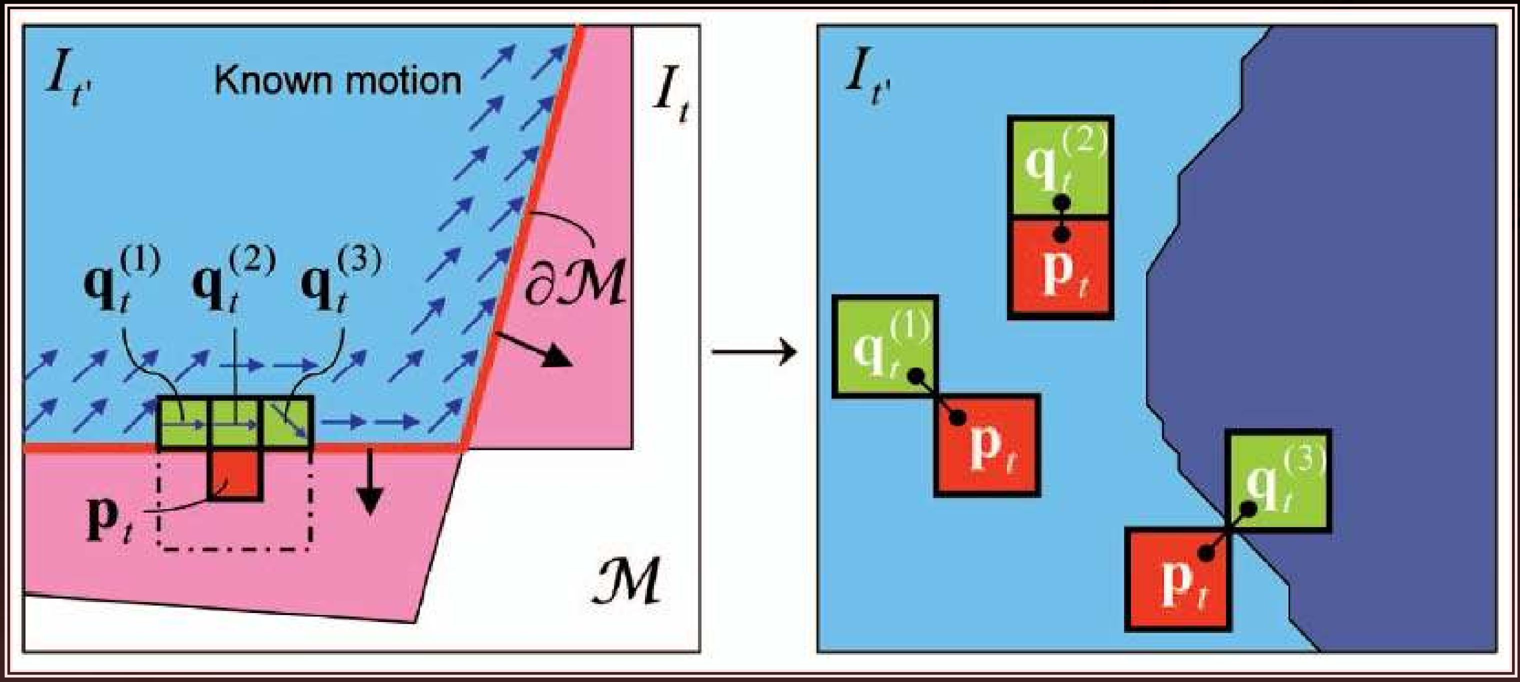
$$I_t(p_t) = \begin{cases} midian_{t'}(I_{t'}(T_t^{t'} p_t)) & if\ v_t(p_t) < T \\ keep\ it\ as\ missing & otherwise \end{cases}$$

where

$$v_t(p_t) = \frac{1}{n-1} \sum_{t' \in M_t} (I_{t'}(T_t^{t'} p_t) - \overline{I_{t'}}(T_t^{t'} p_t))^2$$

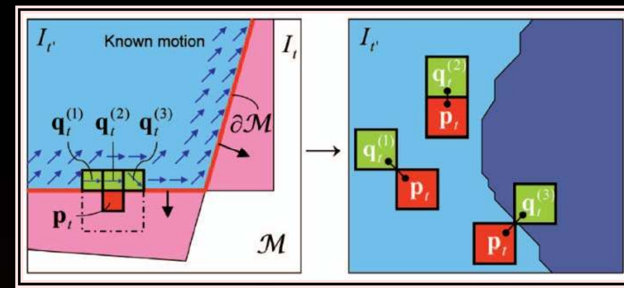
$$\overline{I_{t'}}(T_t^{t'} p_t) = \frac{1}{n} \sum_{t' \in M_t} I_{t'}(T_t^{t'} p_t)$$

Motion Inpainting



A. Telea, "An Image Inpainting Technique Based on the Fast Marching Method," J. Graphics Tools, vol. 9, no. 1, pp. 23-34, 2004.

Motion Inpainting



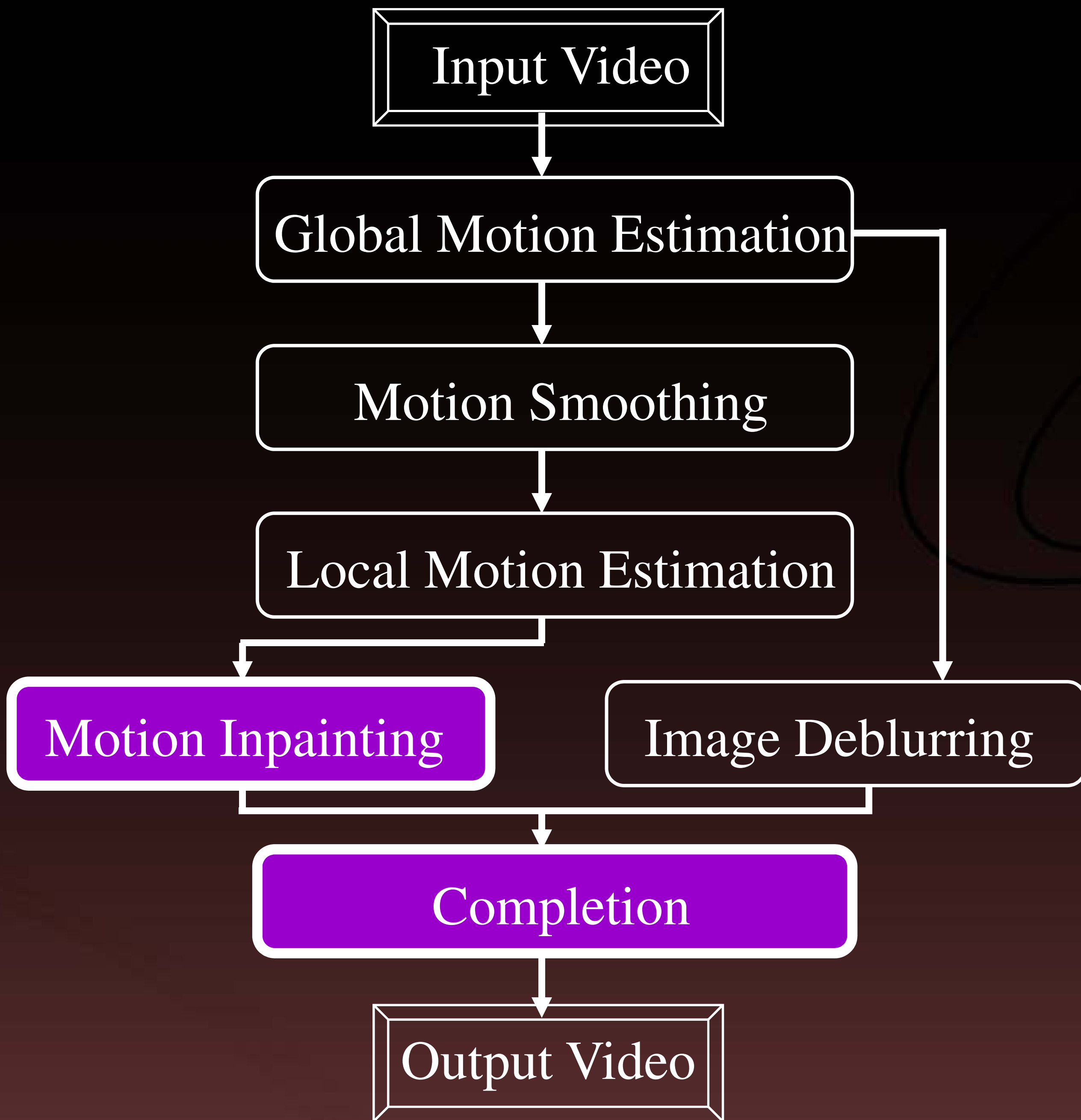
- The motion value for pixel p_t is generated by a weighted average of the motion vectors of the pixels $H(p_t)$

$$F(p_t) = \frac{\sum_{q_t \in H(p_t)} w(p_t, q_t) F(p_t \mid q_t)}{\sum_{q_t \in H(p_t)} w(p_t, q_t)}$$

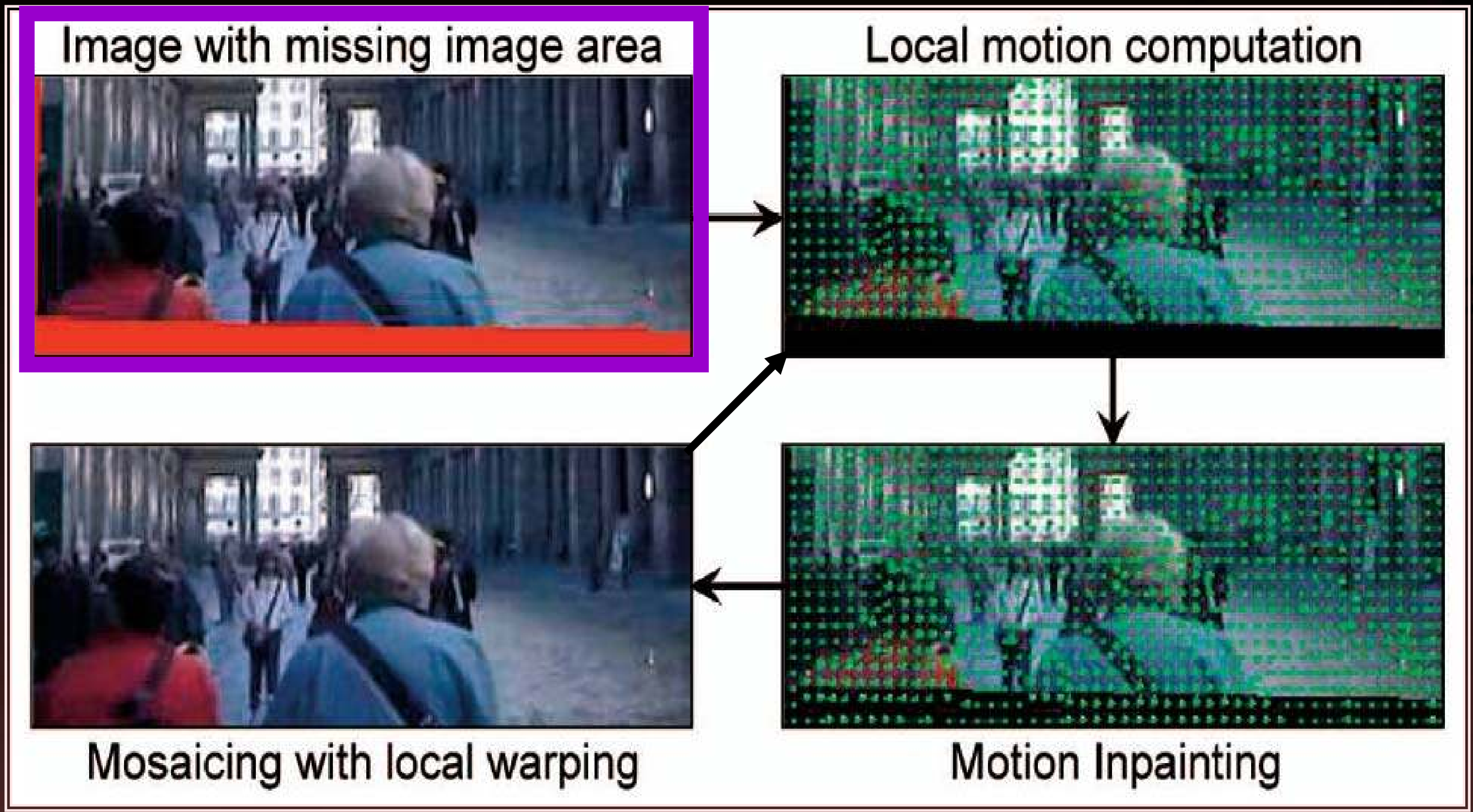
where

$$F(p_t \mid q_t) = F(q_t) + \nabla F(q_t)(p_t - q_t) = F(q_t) + \begin{bmatrix} \frac{\partial F_x(q_t)}{\partial x} & \frac{\partial F_x(q_t)}{\partial y} \\ \frac{\partial F_y(q_t)}{\partial x} & \frac{\partial F_y(q_t)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$w(p_t, q_t) = \frac{1}{\| p_t - q_t \|} \frac{1}{\| I_{t'}(q_{t'} + p_t - q_t) - I_{t'}(q_{t'}) \| + \varepsilon}$$


Input Video
Global Motion Estimation
Motion Smoothing
Local Motion Estimation
Motion Inpainting
Image Deblurring
Completion
Output Video

Summary of the Algorithm



Experimental Results

- 30 video clips (about 80 minutes) with different types of scenes
- k = 6 for motion smoothing
- 5x5 filter for motion inpainting

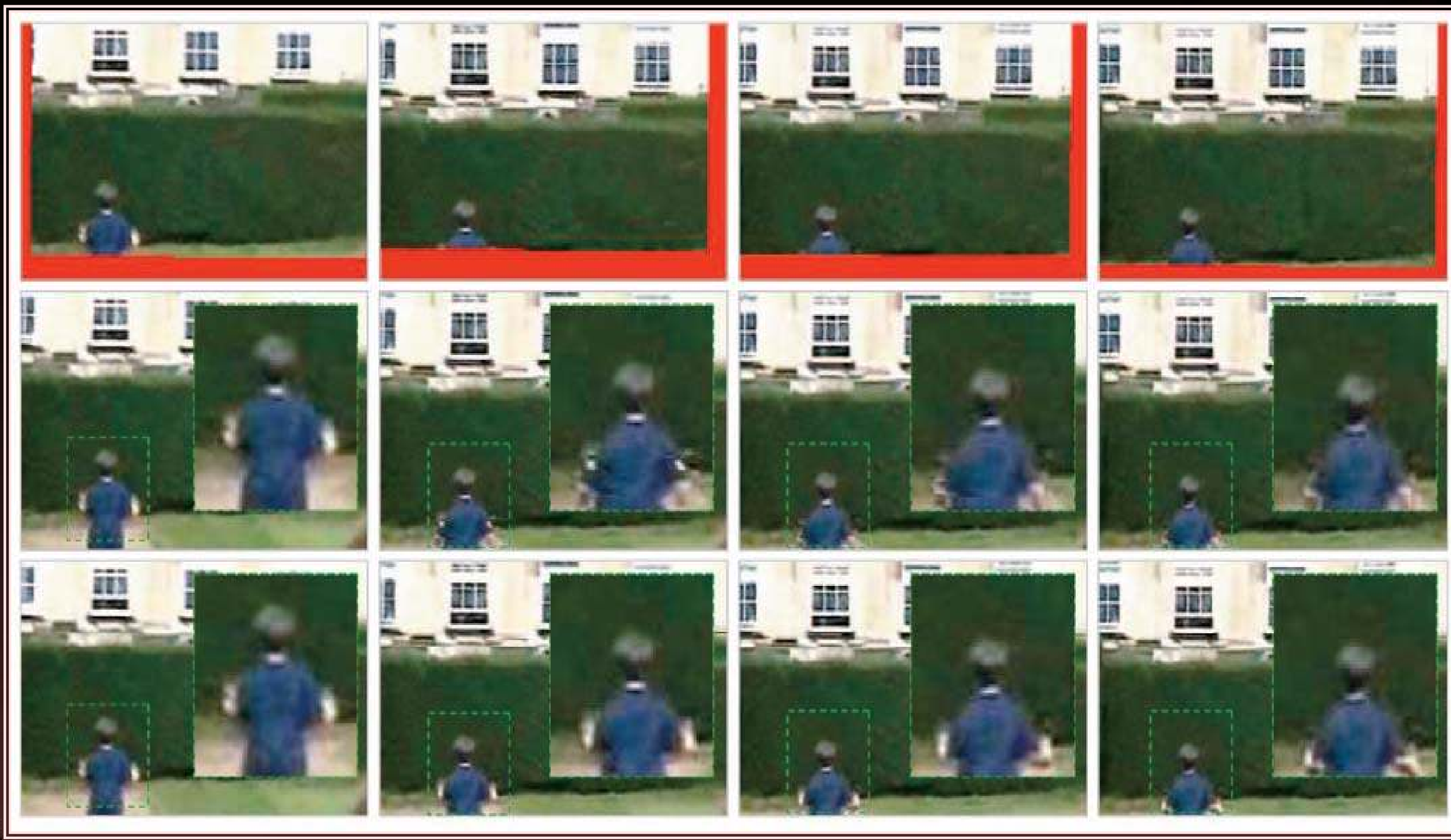
Experimental Results (1)

Experimental Results (2)

Experimental Results (3)

Failure Cases – incorrect estimation of motion

Failure Cases – abrupt changes of motion



Quantitative Evaluation

- Deviation from the Ground Truth.
- MAD of intensity

Our method	Mosaicing
9.87	12.2
4.18	7.83
7.64	8.27
6.65	9.14
10.5	23.6

Quantitative Evaluation

- Evaluation of Spatio-Temporal Smoothness.
  - The normalized discontinuity measure D is defined as

$$D = \frac{1}{n}\sum_{i}^{n} \| \nabla I_i \| = \frac{1}{n}\sum_{i}^{n} \sqrt{\nabla I_i \cdot \nabla I_i}$$

$$\nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial t} \end{bmatrix} \approx \begin{bmatrix} I(x+1, y, t) - I(x-1, y, t) \\ I(x, y+1, t) - I(x, y-1, t) \\ I(x, y, t+1) - I(x, y, t-1) \end{bmatrix}$$

  - The relative smoothness is evaluated by $(D_M - D_O)/(D_M - D_A)$
  - 5.9%~23.5% smoother than mosaicing

Computation Cost

- 2.2 frames/s for 720x486 video with P4 2.8GHz CPU

	Computational Cost (%)	Number of times
Global motion estimation	5.26%	N
Motion smoothing	0.05%	N (using $2k$ motions)
Local motion estimation	84.25 %	$2kN$
Motion inpainting	7.20%	$2kN$
Image warping	1.47 %	(2k+1)N for global warping, $2kN$ for local warping
Image deblurring	1.77%	N (using $2k+1$ images)

Conclusion

- Motion inpainting instead of cropping.
- Deblurring without estimating PSFs.
- Spatial smoothness is indirectly guaranteed by the smoothness of the extrapolated motion.
- Temporal consistency on both static and dynamic areas is given by optical flow from the neighboring frames.

Thank You

presented by

蕭志傑

Hsiao, Chih-Chieh