Subspace optimization for streaming video stabilization

Yu-Shuen Wang, National Chiao Tung University
video stabilization

Input

Output
Goal

* A practical solution to achieve video stabilization
  * Achieves the appearance of an idealized camera motion
  * Streamable and efficient, potentially real-time
  * No limitation on camera motion and scene structure
Challenges

* Where to move video content to be stable?
* How to move (warp) video content to be stable?
Feature trajectories
**Homography video stabilization**

* Consider each video frame to be a plane in the 3D space
* Compute a homography matrix to estimate the camera motion that can transform features from \([x, y]^T\) to \([u,v]^T\)

\[
\begin{bmatrix} h1 & h2 & h3 \\ h4 & h5 & h6 \\ h7 & h8 & h9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \Rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x'/w' \\ y'/w' \end{bmatrix}
\]
Homography video stabilization

* Each homography matrix $F_t$ represent the camera motion between consecutive frames.

* By multiplying $F_t$ together, we get the original camera path $C_t$

\[ C_{t+1} = C_t F_{t+1} \Rightarrow C_t = F_1 F_2 \ldots F_t \]
* Given the original path $C_t$, we express the desired smooth path as

$$P_t = C_t B_t$$

* Standard methods smooth $C_t$ using low pass filter $B_t$ to determine $P_t$
Shaky video
Homography video stabilization
Homography video stabilization

- Cannot handle parallax
  - The motion of a background feature mainly depends on its depth
  - Homography warping is not sufficient to stabilize the features
3D video stabilization [Liu et al. 2009]

Structure from motion
3D video stabilization [Liu et al. 2009]

- Reconstruct 3D feature positions, followed by synthesizing a new smooth camera path

- Very challenging

- Cannot handle videos
  - with a large planner area
  - contain camera zooming
  - Rolling shutter
Spatially and Temporally Optimized Video Stabilization

Original trajectories

Smoothed trajectories
Main contribution

* Properly handling parallax is important
* Fewer long and reliable feature trajectories are demanded
  * Compute spatial relations first
  * Followed by smoothing the reduced parameters
Let the $i^{th}$ trajectory be $P_i = \{p_i^m, p_i^{m+1}, ..., p_i^n\}$, where $p_i = (x_i, y_i) \in R^2$ is the feature position, and $m$ and $n$ are the start and the end frames of $p_i$, respectively.
Individual feature trajectory smoothing

Input image

Output image
Individual feature trajectory smoothing
Individual feature trajectory smoothing

Filter different filter kernel support
• Individually stabilize each feature trajectory is not allowed because neighboring regions may have different treatments.

• Apply Delaunay triangulation to generate a triangular mesh and retain the triangles from distortion.
Spatial rigidity preservation

- The mesh is formed by KLT features and image corners. So we minimize

\[ \Omega_c = \sum_t \sum_f \sum_{\{i,j\} \in \epsilon(f)} \left| (p_i^t - p_j^t) - R^t (p_i^t - p_j^t) \right|^2_F \]

- \( t \) is the frame index,
- \( \epsilon(f) \) denotes the edges of triangle \( f \),
- \( R \) is the unknown rotation matrix,
- \( p' \) is the smoothed version of \( p \)
Video stabilization cannot deviate the smoothed camera motion from its original trajectory too much.

We thus minimize

\[ \Omega_p = \sum_{p_i} \sum_t |p'_i - p^t_i|^2 \]
Reliability constraints

* There is no way to determine their smoothed positions when video frames containing no features.

* We slightly enforce the corners of these extremely difficult frames to have smooth transitions between consecutive time steps.

\[ \Omega_r = \sum_{t \in U} \sum_{i=0}^{3} \| c'_i^{t-1} - 2c'_i^t + c'_i^{t+1} \|_F^2 \]
The acceleration of each feature should be small.

We thus minimize

$$\Omega_s = \sum_{p_i} \sum_{t} \left| p'_{i-1} - 2p'_i + p'_{i+1} \right|_F^2$$
Feature trajectory smoothing

original

full space stabilization  Bézier stabilization
Feature trajectory smoothing
Feature trajectory smoothing

- Represent each smoothed feature trajectory using a low dimensional representation
- Bezier curve
  - Each smoothed feature position becomes
  \[ \mathbf{p}_i^t = \sum_{\ell=0}^{d} \omega_{i}^{t,\ell} \mathbf{q}_{i}^{\ell}, \text{ where} \]
  \[ \omega^{t,\ell} = \binom{d}{\ell} (1 - r)^{d-\ell} r^{\ell}, \]
  \[ r = \frac{t-n}{n-m} \text{ is the interpolation coefficient}, \]
  \[ \mathbf{q}_i \text{ are the control points} \]
Spatial rigidity preservation

\[ \Omega_c = \sum_t \sum_f \sum_{\{i,j\}\in \mathcal{E}(f)} \left\| (p'_t - p'_t) - R^t(p_i^t - p_j^t) \right\|_F^2, \]

\[ \Omega_c = \sum_t \sum_f \sum_{\{i,j\}\in \mathcal{E}(f)} \left\| \left( \sum_{\ell=0}^d \omega_{i,\ell} q_{i,\ell} - \sum_{\ell=0}^d \omega_{j,\ell} q_{j,\ell} \right) - R^t(p_i^t - p_j^t) \right\|_F^2, \]
Original camera motion approximation

\[ \Omega_p = \sum_{p_i} \sum_{t} \sum_{i} |p_i^t - p_i^t|_F^2 \]

\[ \Omega_p = \sum_{p_i} \sum_{t} \left| \sum_{\ell=0}^{d} \omega_{i,\ell} q_{i,\ell} - p_i^t \right|_F^2 \]
By integrating the mentioned energy terms, we search for the control points of Bezier curves that can minimize the objective function

\[ \Omega = w_c \Omega_c + w_p \Omega_p \]

\[ w_c = 10, \quad w_p = 1 \]
Content aware warp

* Homography + feature preserving warp

\[
D_h^t = \sum_{\{i,j\} \in E^t} \left| (\hat{v}_i^t - \hat{v}_j^t) - H^t (v_i^t - v_j^t) \right|^2_F
\]

\[
D_p^t = \sum_{\hat{p}_i^t} \gamma_j^t \left| \sum_{j \in \mathcal{L}_i^t} \delta_j^t \hat{v}_j^t - \hat{p}_i^t \right|^2_F
\]

\[
\gamma_j^t = \min(1, \frac{t-m}{10}, \frac{n-t}{10})
\]

\[
\mathcal{L}_i^t \text{ are the indexes of vertices surrounding } \hat{p}_i^t
\]

\[
\delta_j^t \text{ is the combination weight.}
\]
Content aware warp

* Homography + feature preserving warp

\[
D_h^t = \sum_{\{i,j\} \in E^t} \left| (\hat{v}_i^t - \hat{v}_j^t) - H^t (v_i^t - v_j^t) \right|^2_F \\
D_p^t = \sum_{\hat{p}_i^t} \gamma_j^t \left| \sum_{j \in L_i^t} \delta_j^t \hat{v}_j^t - \hat{p}_i^t \right|^2_F
\]

* \(H^t\) can be considered as a 2D stabilization
  
  * It is computed according to the original and the smoothed features.
Content aware warp

* Homography + feature preserving warp

\[
D_h^t = \sum_{\{i, j\} \in E^t} \| (\hat{v}_i^t - \hat{v}_j^t) - H^t (v_i^t - v_j^t) \|_F^2
\]

\[
D_p^t = \sum_{\hat{p}_i^t} \gamma_j^t \sum_{j \in \mathcal{L}_i^t} \delta_j^t \hat{v}_j^t - \hat{p}_i^t \|_F^2
\]

\[
\gamma_j^t = \min(1, \frac{t-m}{10}, \frac{n-t}{10})
\]

\[
\mathcal{L}_i^t \text{ are the indexes of vertices surrounding } \hat{p}_i^t
\]

\[
\delta_j^t \text{ is the combination weight.}
\]
Crop non-overlapping regions

* As video frames are transformed opposite to high frequency camera motions, the regions that are not always visible should be removed.
Streaming implementation

* Solve for a subset of video frames instead of the whole video cube at a time to prevent the scalability problem.
* Set positional constraints in the overlapping frames to prevent the discontinuity artifacts caused by our streaming approach.
Results and discussions

Spatially and Temporally Optimized Video Stabilization

Yu-Shuen Wang
National Chiao-Tung University, Taiwan

Feng Liu
Portland State University

Pu-Sheng Hsu
National Cheng-Kung University, Taiwan

Tong-Yee Lee
Performance

* The computational cost depends on the number of trajectory, instead of video resolutions and lengths

<table>
<thead>
<tr>
<th>resolution</th>
<th># frame</th>
<th># trajectory</th>
<th>smoothing (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>640 × 360</td>
<td>609</td>
<td>4580</td>
<td>6.685</td>
</tr>
<tr>
<td>720 × 480</td>
<td>389</td>
<td>15415</td>
<td>7.42</td>
</tr>
<tr>
<td>720 × 480</td>
<td>331</td>
<td>7016</td>
<td>4.368</td>
</tr>
<tr>
<td>1280 × 720</td>
<td>775</td>
<td>26921</td>
<td>22.72</td>
</tr>
</tbody>
</table>

**TABLE 1**
Video information and the corresponding timing statistics. Note that the timing of KLT feature extraction is not included.
Results and discussions

* Bezier degrees
  * quadratic vs. cubic

![Original vs. Bezier curves comparison](image-url)
Results and discussions

- Bezier representation
  - High performance due to fewer unknown variables
  - Strong stabilization due to low dimensional representation
Thank you!

Question?