Re-Cinematography: Improving the Camera Dynamics of Casual Video

Michael Gleicher
Feng Liu
Department of Computer Sciences
University of Wisconsin-Madison
Motivation: More video doesn’t mean better video

More Video!
- Cameras everywhere
- Players everywhere
- Sharing everywhere
Motivation:
More video doesn’t mean better video

Good video takes effort!
Problem: Bad Camera Motion

No planning
No tripod
Problem: Bad Camera Motion
Prior Work: Image Stabilization

One part of the problem: jitter
Helped by Image Stabilization
Problem: Bad Camera Motion
Solution: Re-Cinematography

Re-Cinematography:
Post-process video clips so that the camera motions better follow the rules of good video.
Rubber duck races
Vail, CO, USA, 19 August, 2007

Source Footage
Re-Cinematography Result
What the art of cinematography tells us about camera motion

Camera motions should be intentional
- Avoid movement if not necessary
- Move in directed ways

Re-Cinematography:
Post-process video clips so that the camera motions appear to better follow the rules.
Re-Cinematography Pipeline

Source Video → Motion Estimation → Motion Synthesis → Image Transform → Result Video
Re-Cinematography Pipeline (1)

Source Video → Motion Estimation → Motion Synthesis → Image Transform → Result Video

How did the camera move?
Re-Cinematography Pipeline (2)

Source Video → Motion Estimation → Motion Synthesis → Image Transform → Result Video

Figure out what motion we want in the result
Re-Cinematography Pipeline

Source Video → Motion Estimation → Motion Analysis → Motion Synthesis → Image Transform → Result Video

Scene Analysis
Motion Synthesis Steps

Source Video → Motion Estimation → Motion Synthesis → Image Transform → Result Video

- Segment Video
- Create Motions
- Optimize Motions
3 Key Ideas

• Analyze motion estimates to break video into segments
• Use local mosaics to keyframe new camera motions
• Consider both motion and image quality to automatically keyframe cameras
Background: Camera Motion Estimation and Projective Transformations

\[ x', y' = \frac{ax + by + c}{gx + hy + 1}, \frac{dx + ey + f}{gx + hy + 1} \]

\[
\begin{bmatrix}
a & b & c \\
d & e & f \\
g & h & 1 \\
\end{bmatrix}
\]
Mosaicing

Source Images

Base Image

All images transformed to common base image
3 Key Ideas

- Analyze motion estimates to break video into segments
- Use local mosaics to keyframe new camera motions
- Consider both motion and image quality to automatically keyframe cameras
Local Mosaics

Limit error and motion in each segment
Break videos into segments with like motions

- Move in a direction
- Small movement
- Zoom in or out
- Bad estimation
Break videos into segments with like motions

Static  Moving  Bad
Break videos into segments with like motions
3 Key Ideas

- Analyze motion estimates to break video into segments
- **Use local mosaics to keyframe new camera motions**
- Consider both motion and image quality to automatically keyframe cameras
Photograph the Mosaic with a virtual camera
Virtual camera does not have to be where the real camera was

Result frames shown in magenta
Source frames shown in yellow
What paths do we want?

1. Preserve the intent of the source
2. Obey the rule of cinematography:

   Camera motion should be intentional
The key insight: Translate cinematography to implementation

Motion should be intentional

- Static shots should be static
- Moving shots are goal directed
  - Constant velocity with ease in/out
Directed Paths

Interpolate with direct constant* velocity paths

* Possibly with ease-in and out.
Moving the Camera

Interpolate transformations in projective space

$$\text{mlerp}(A,B,\alpha) = \exp( \alpha \log(A) + (1-\alpha) \log(B) )$$

A,B are matrices
Matrix logarithm interpolation of transforms
Smooth Paths Depart from Original

Source motion

Result motion
Changing motion means transforming frames

Source motion

Result motion
Transforming frames might cause problems

Source frame

Result frame
3 Key Ideas

• Analyze motion estimates to break video into segments
• Use local mosaics to keyframe new camera motions
• Consider both motion and image quality to automatically keyframe cameras
Penalties for each frame

Offscreen
Uncovered
Distortion
Offscreen
Uncovered
Finding good motions

An optimization problem:

Find motion $M$ that minimizes: $\text{nonsmooth}(M) + \text{sum image penalties}$

Or a constrained optimization problem:

Find motion $M$ that minimizes: $\text{nonsmooth}(M)$

Subject to: $\text{sum image penalties} < \text{thresh}$
Static Segments

• If initial video was nearly static

• Make it a static segment
  – No camera motion
Keyframing Dynamic Segments

Start with direct path

Is the worst frame penalty below threshold?

Yes

No

Insert a key at worst frame
A contrived synthetic example to explain key insertion
Try the smooth motion first
Insert a key at the worst point
Inserting keys creates velocity discontinuities
Implementation

• Analyze video (slow-preprocess)
  – Motion estimation, salience detection
• Re-Cinematography (a few seconds for up to 2 minutes of video)
  – Break video into segments
  – Keyframe segments
• Create result (30fps playback using graphics hardware)
  – Transform each frame
  – In-Paint (draw frames +/- 2 seconds)
Examples

- Sanyo XACTI camera
- Source footage with image stabilization
Mini-Golf
Pico Mountain, VT, 2006

Source Footage

Re-Cinematography Result
2X Speed

Source Footage

Re-Cinematography Result
Learning to run
Vail, CO, 19 August 2006

Re-Cinematography Result
2X speed comparison

Source Footage

Re-Cinematography Result
2X speed comparison

Source Footage

Re-Cinematography Result
Sam’s First Steps, July 6\textsuperscript{th}, 2006

Re-Cinematography
Result
Re-Cinematography “Works”
Velocity profiles meet goals

![Graph showing velocity profiles meeting goals](image-url)
Static segments are static
Moving segments have piecewise constant velocity
Ease in and out
But there are problems

Show source images when motion estimation fails

Jitter from bad motion estimation

Visual Artifacts from bad inpainting
Problems

- Bad camera motion estimation
- Bad motion estimation assessment
- Bad important object detection
- Bad inpainting

- These are standard questions being explored in Computer Vision!
Motion Blur

- Hard for Estimation
- Wrong for Changed Motion
A more interesting question: To swing or not to swing
The Connecticut Valley Train

Try out the NEW (beta) version of this page!

Added: July 29, 2006
From: avivofir
This is the Connecticut Valley Train ...
(more)

Category: Travel & Places
Tags: steam, train
URL: http://www.youtube.com/watch?v=7CEX-PRLA
Embed:

Related
More from this user
Playlists

Showing 1–20 of 27
Steam Trains 2006 - Keeping the Past Alive
04:18
From: railfunny
Views: 14369
CPR 2816 steam train (Canada) 2006 vol.2
04:36
From: altromercato
Summary

Re-cinematography changes the camera motions in video to better follow the rules of good video

Key ideas to do this:
• Break video into local mosaics
• Animate a camera viewing the local mosaics
• Automatically keyframe the camera to optimize tradeoffs

Research supported in part by NSF grant IIS-0416284 and the UW Graduate School Research Committee.
Because I thought you’d ask….

Answers to Common Questions

• I don’t know.
• No, we don’t introduce cuts.
• The details are in the paper, send me email if it’s not clear.
• Friends in industry say they can do the camera motion estimation robustly, in real time.
• Yes, I would like to go to Oktoberfest Friday.
• Our in-painter builds a 4 second mosaic for each frame.
• 2
• Logarithms and exponents of 3x3 matrices can be computed robustly and efficiently with iterative methods.
• Yes, this slide is an old joke – but I haven’t used it in years.