Video stabilization

Problem description

Change positions of image frames through time to remove rapid motion (e.g. hand-held camera, external shaking)

Original

Stabilized

M. Grundmann and V. Kwatra and I. Essa , „Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths“, CVPR2011

Stabilization approaches

- Mechanic
  - Move sensor or lenses
  - Stabilize image before it is digitalized
  - Lenses (Nikon - 1994, Canon - 1995): detect vibrations and move lens with magnetic field
  - Sensor: move sensor with motors (supports lens changes)
  - External: Steadicam, tripod, dolly
- Digital
  - Post processing
  - Move images, apply geometrical transformations
  - Digital filters in case of blurring

Digital stabilization types

- Global
  - Making camera motion smooth
  - Can be fully automatic or initialized manually

- Object-centric
  - Object's position does not change significantly in the camera frame
  - Manual object selection

Stabilization by alignment

Two consecutive images, aligned by shifting one of them

Frame A

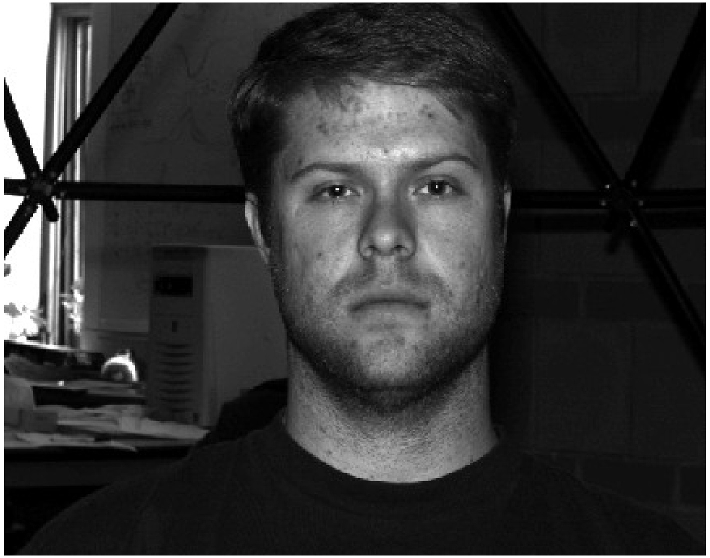
Frame B

Color composite (frame A = red, frame B = cyan)

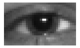
Stabilization by feature tracking

- Only look for regions in image that can be reliably positioned in frames
  - Corners
  - Blobs
- Features have to be visible all the time
- Use difference in position to determine transformation

Normalized cross correlation



Search image, F



Model, H

$\psi(\mathbf{A})$ ... reshape pixels in A in a vector.

$\mathbf{F}_{ij}$ ... sub-image from ***F*** *centered at* (i,j).

$$\mathbf{h} = \psi(\mathbf{H})$$
$$\mathbf{f}_{ij} = \psi(\mathbf{F_{ij}})$$

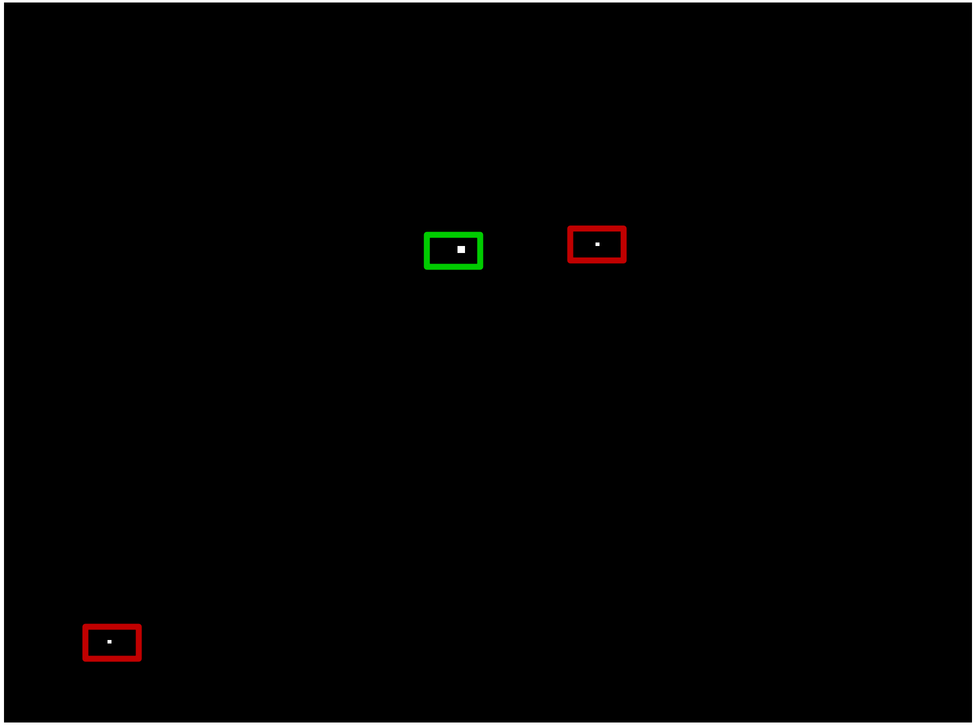
$\hat{h}$ ... average brightness of ***H***

$\hat{f}_{ij}$ ... average brightness of $\boldsymbol{F_{ij}}$

Normalized cross correlation:

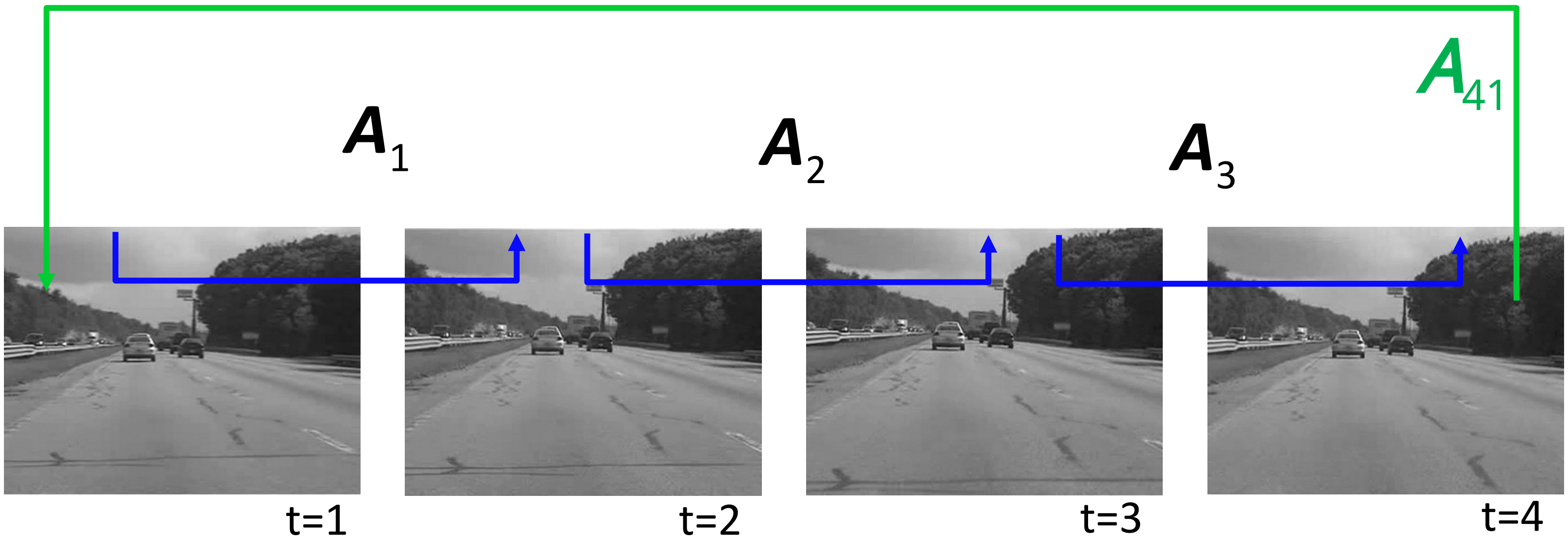
$$G(i,j) = \frac{(\mathbf{h}^T - \hat{h})(\mathbf{f}_{ij} - \hat{f})}{\sqrt{\mathbf{h}^T\mathbf{h}}\sqrt{\mathbf{f}_{ij}^T\mathbf{f}_{ij}}}$$

Example



More positions are equally suitable according to NCC

Transformation chain



$$\mathbf{x}_{t=1} = \mathbf{A}_1 \mathbf{x}_{t=2}$$
$$\mathbf{x}_{t=2} = \mathbf{A}_2 \mathbf{x}_{t=3}$$
$$\mathbf{x}_{t=3} = \mathbf{A}_3 \mathbf{x}_{t=4}$$

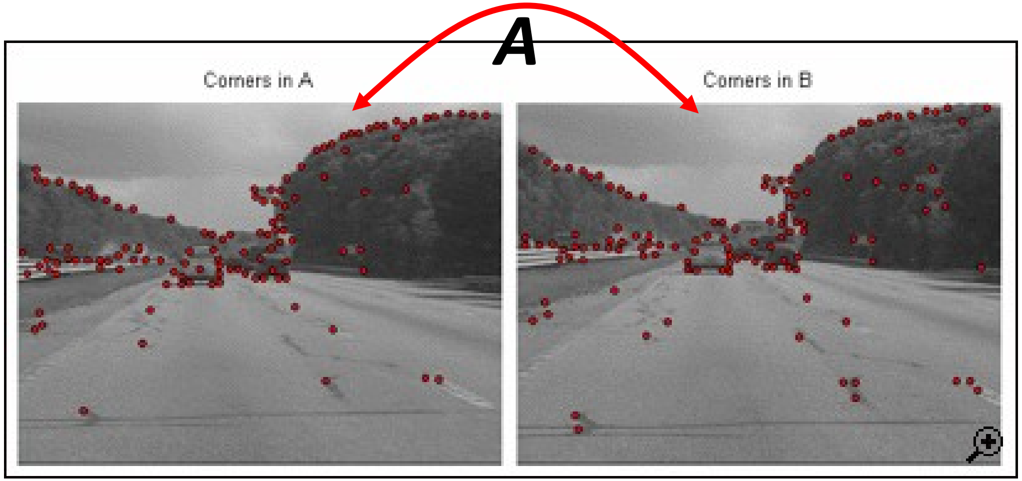
$$\mathbf{x}_{t=1} = \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 \mathbf{x}_{t=4}$$
$$\mathbf{A}_{41} = \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3$$

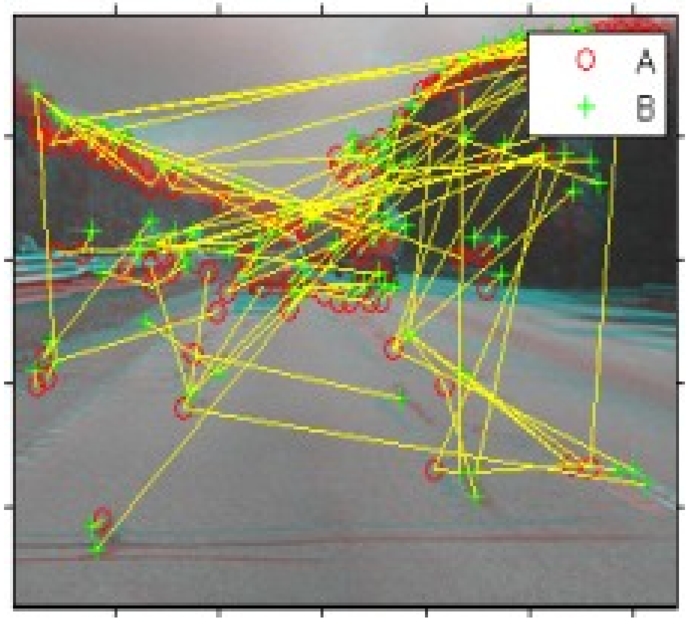
Number of features

- One feature – translation
- Two features – translation + rotation or scale
- Three features – affine transformation
  - Only planar motion
- Four features – perspective transform
  - Assumes planar scene
  - Can lead to destroyed depth illusion

Stabilization using keypoints

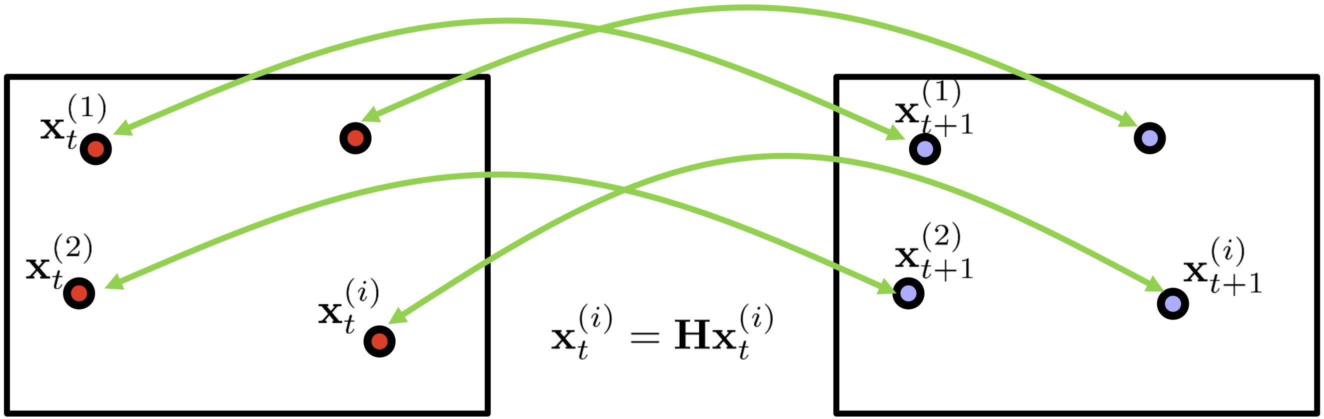
- Detect keypoints in both images, compute correspondences, estimate transformation
  - How to find best matches
  - How to estimate transformation





Deformation model

- Planar transformation - homography (3x3 matrix - 8 parameters, one constant)
- Affine transformation - more simple, small movements



Algorithm

- Detect key-points in each image
- Search for correspondences between key-points in image pairs
  - If we are not sure which matches between first and second image are correct we have to use robust estimation methods (RANSAC)
- Compute transformations
- Align images to each other

Global stabilization example

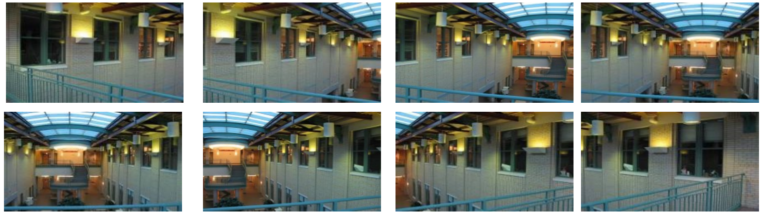
What to do with black border?

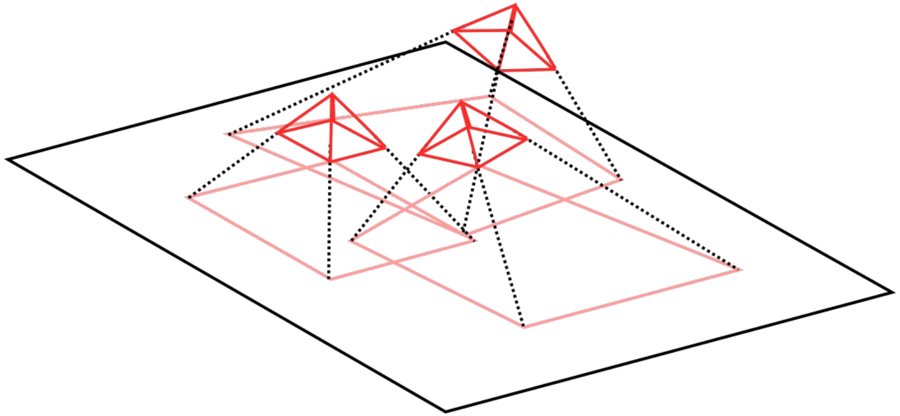
Filling in missing information

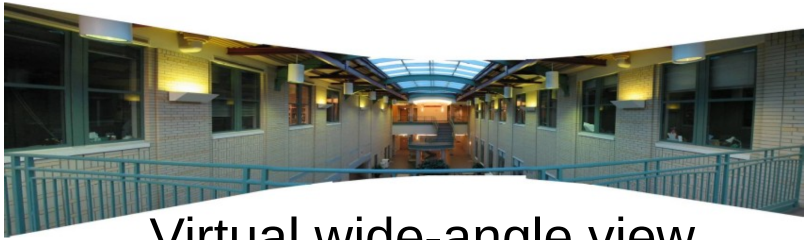
- Cropping viewport
  - Only focusing on always visible part of video
  - Can be problematic with large shifts
- Smoothing trajectory
  - Transfromation filtered with low-pass filter
  - Only jerky motion removed, camera still moves
- Mosaicking

Video mosaicking

- Find transformation between frames
- Assume planarity (expect distortion if not planar)
- Re-project images to a common image plane



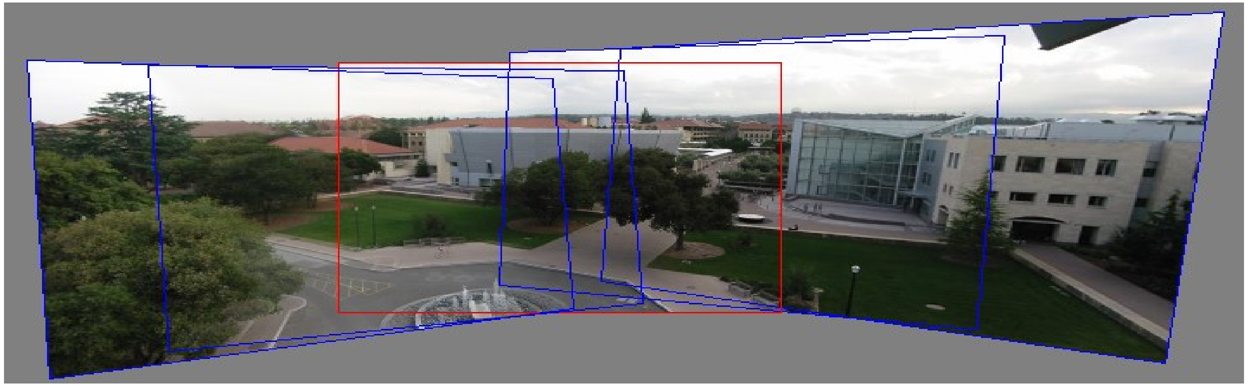




Virtual wide-angle view

Video mosaicking algorithm

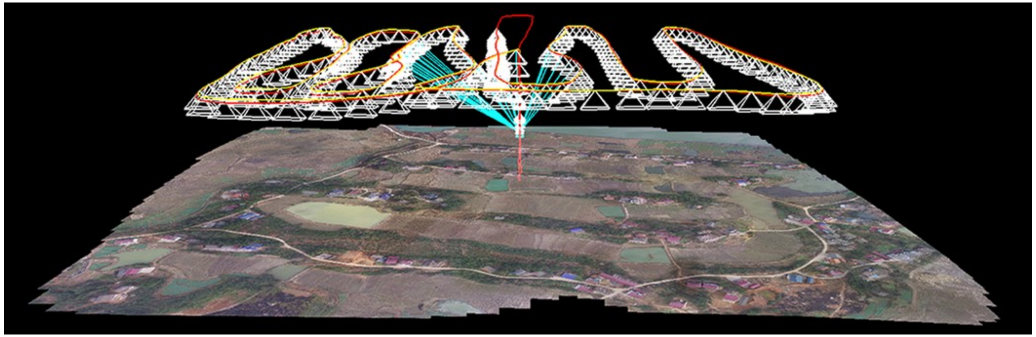
- For each N-th image in video (N fixed or dynamic)
  - Search for keypoints in image and determine correspondences to previous image
  - Estimate homography based on correspondences (RANSAC)
- Determine reference image and recalculate transformations
- Merge images (with blending)



Use cases

- Panoramas
- Areal images
- Video stabilization

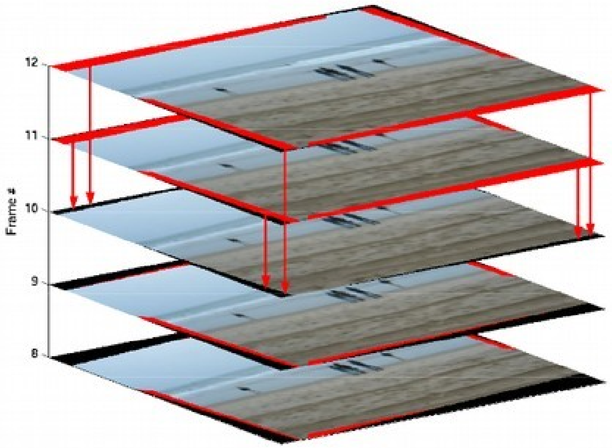
Image Processing Group, Madrid



Pilot Intelligent Group

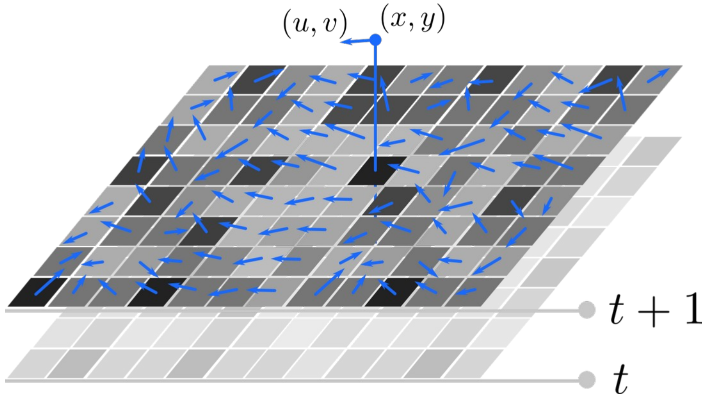
Mosaicking in video stabilization

- Warp each frame to match smoothed motion
- Fill in missing regions from nearby frames
  - Single frame
  - Averaging



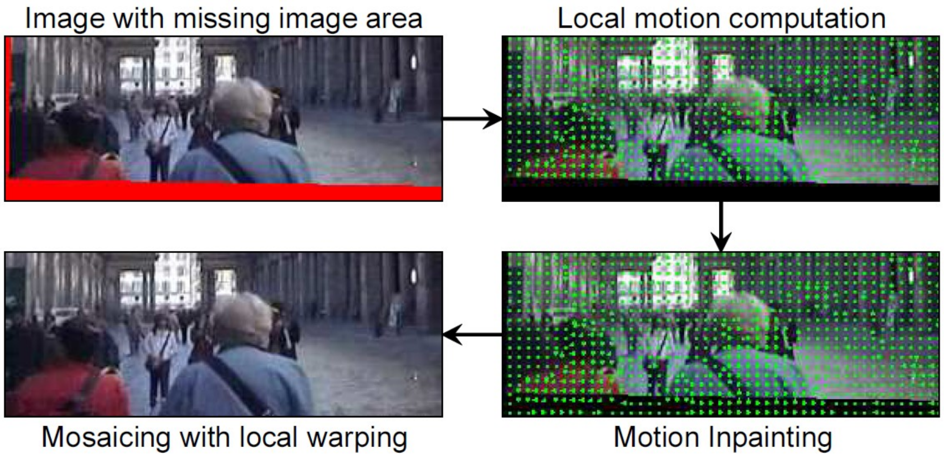
Optical flow stabilization

- Use optical flow instead of keypoints, more dense
  - Lucas & Kanade – fast, local
  - Horn & Schunk – slow, global
- For each pixel compute its most likely translation in the next image
- Fit global transformation to multiple optical flow vectors



Motion inpainting

- Use optical flow to predict which pixels will move where
- Improve mosaicking using these predictions
  - Warp images
  - Inpaint missing information



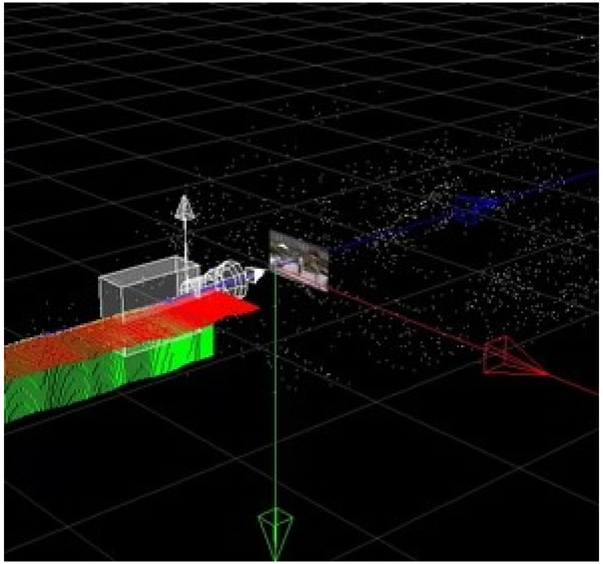
2D stabilization result

Raw video

2d stabilization

Stabilization in space

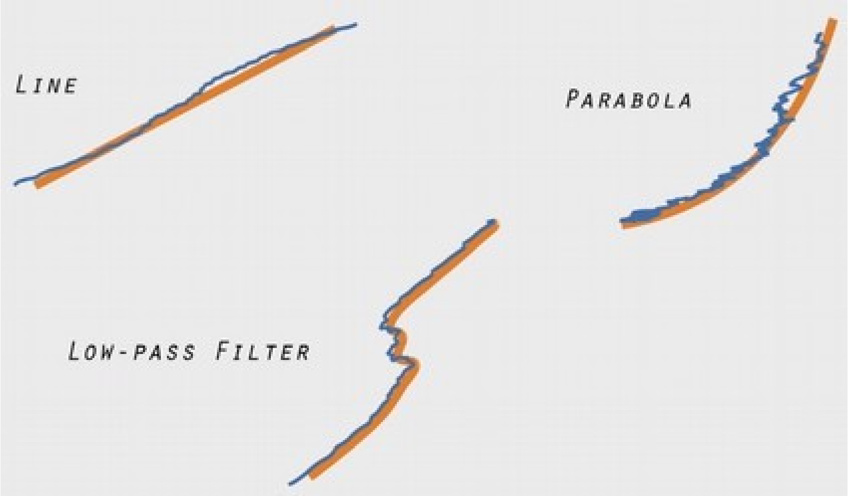
- Reconstruct 3D geometry using Structure from Motion
  - Reconstruction also gives us camera location and translation
- Filter camera path to get smooth path
- Compute warps for modified camera positions and apply them to frames



F. Liu, M. Gleicher, H. Jin and A. Agarwala. Content-Preserving Warps for 3D Video Stabilization. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2009), 2009.

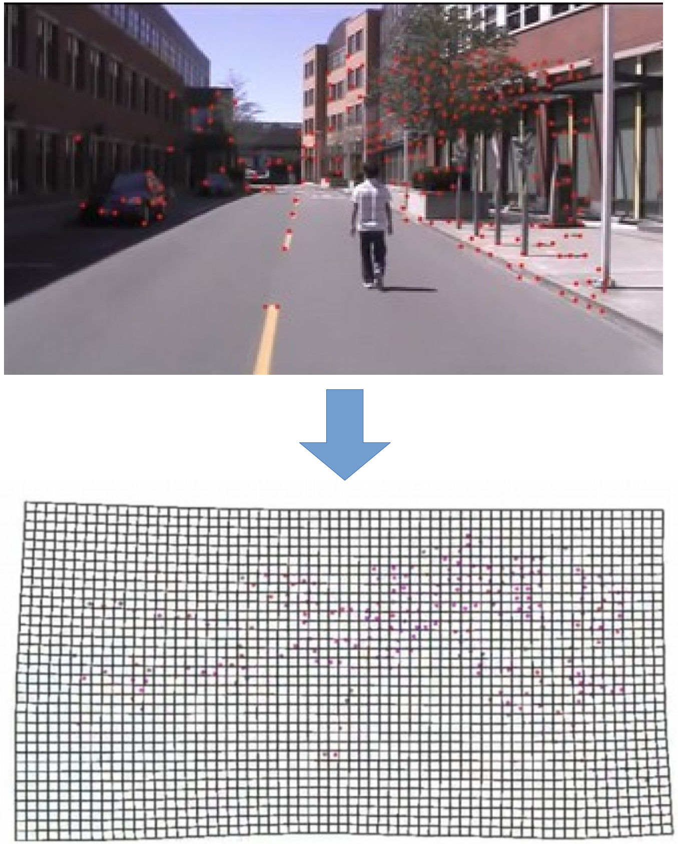
Camera motion types

- 2D stabilization is only removing image motion
- 3D camera path can be used to fit a parametric behaviour



Content-preserving warps

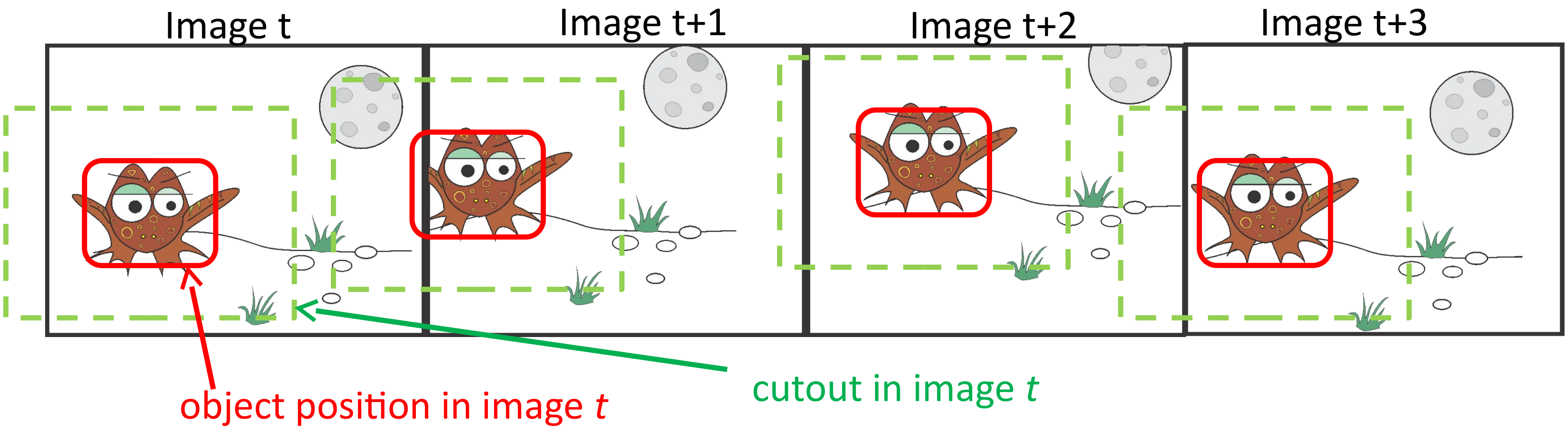
- Non-linear transform
  - 3D points from SFM algorithm
  - Transformation quad-mesh
- Fake small content shifts
  - Small displacements
  - Preserves illusion of depth



3D stabilization result

Local stabilization / re-centering

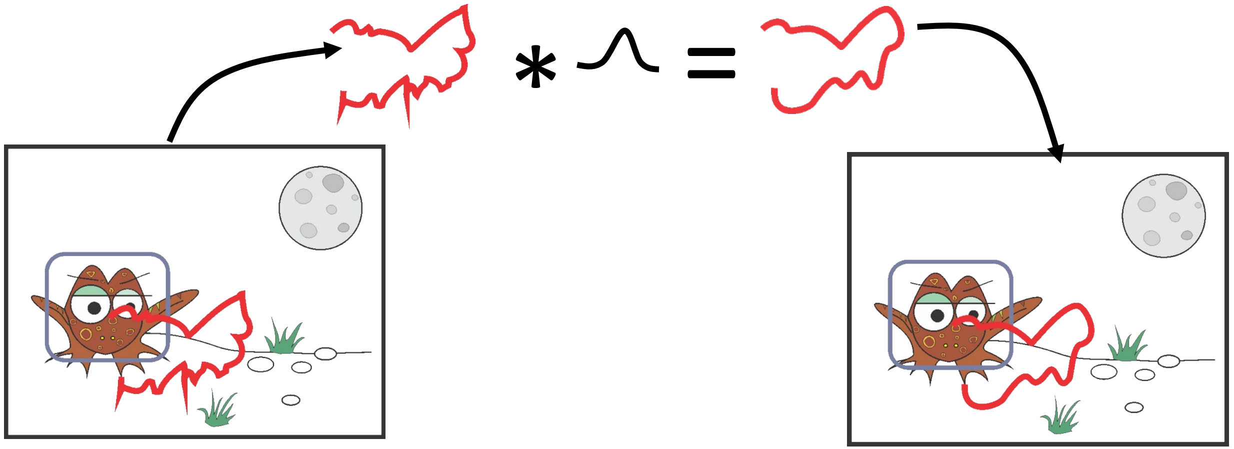
- Manually set object in first image
- Track object through the sequence
- Cut images so that the object is centered



Simple recentering example

Smoothing trajectory

- Shaking = high frequencies
- We can filter trajectory with 1D Gaussian filter (by x and y separately)
- Object will not be centered, but we get rid of the shaking



Questions

- How to find object in the following images?
  - Find position that is a best match to object's appearance
  - Object tracking
- What to do if a part of the cut-out image is out of bounds?
  - Image extrapolation
  - Clamping center to source image