Video processing

Overview

- Processing video frames
  - One-by-one (image processing)
  - Multiple frames at once
- Video stabilization
- Virtual background
- Shot separation

Video stabilization

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 digitized
  - 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

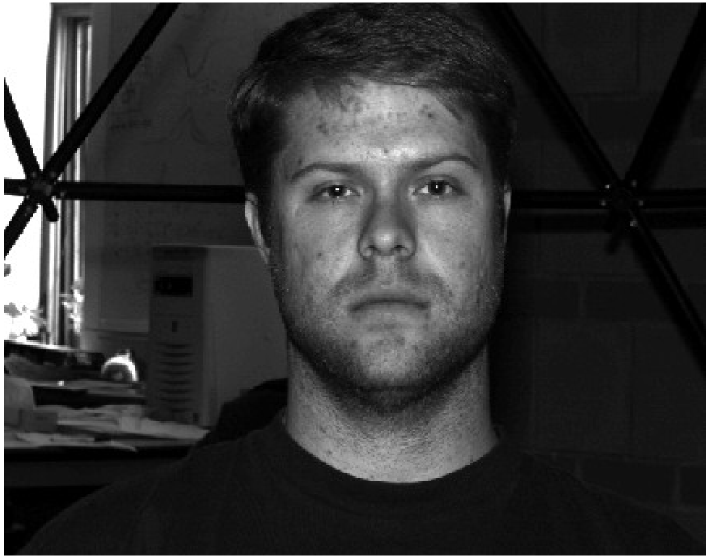
Stabilization by alignment

Two consecutive images, aligned by shifting one of them

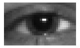
Stabilization by feature tracking

- 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).

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

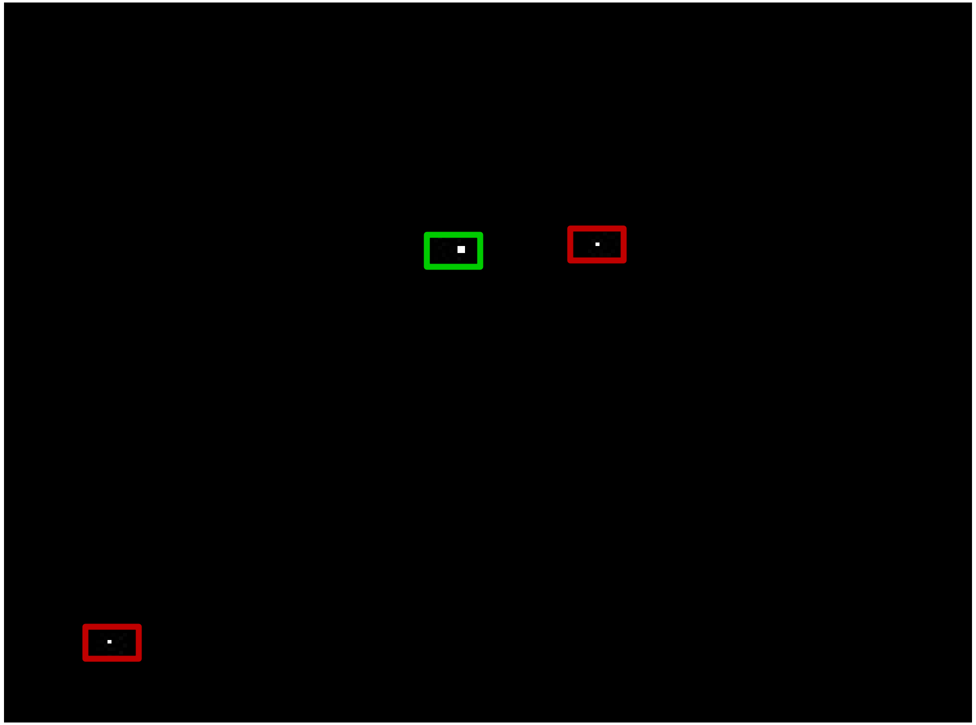
$\hat{h}$ ... average brightness of ***H***

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

Normalized cross correlation:

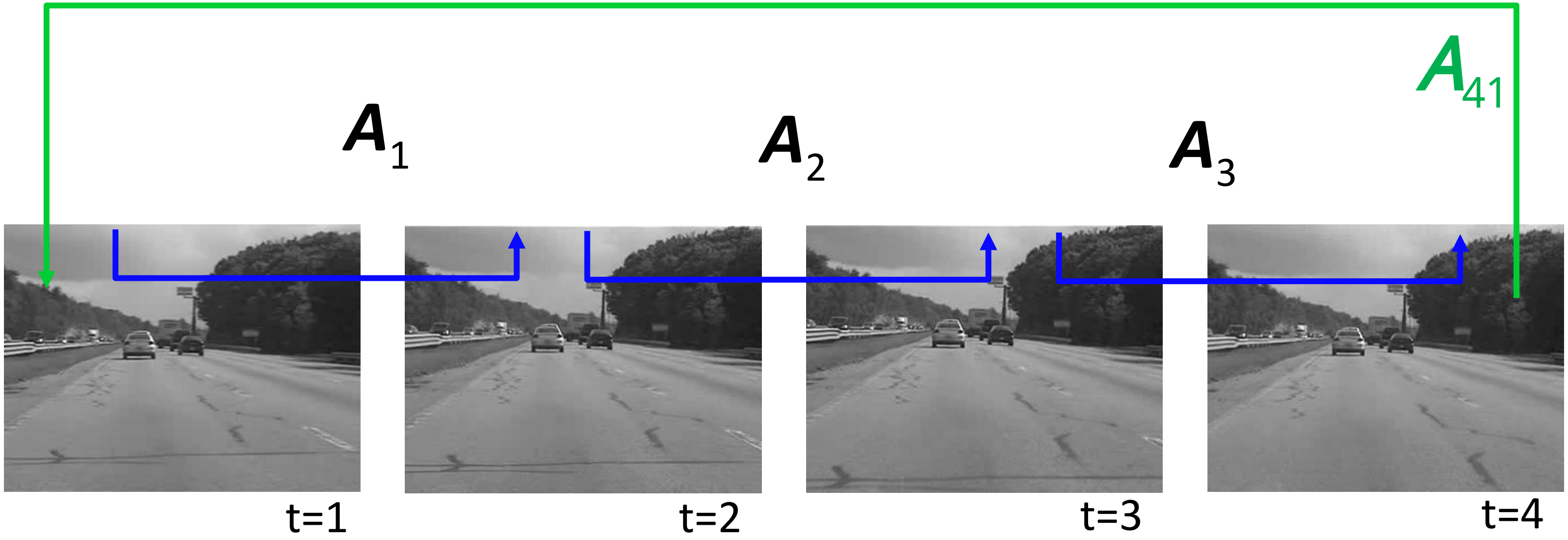
$$G(i,j) = \frac{(\mathbf{h}^T - \hat{h})(\mathbf{f}_{ij} - \hat{f})}{\sqrt{\mathbf{h}^T\mathbf{h}}\sqrt{\mathbf{f}_{ij}^T\mathbf{f}_{ij}}}$$

Example



More positions are equally suitable according to NCC

Transformation chain



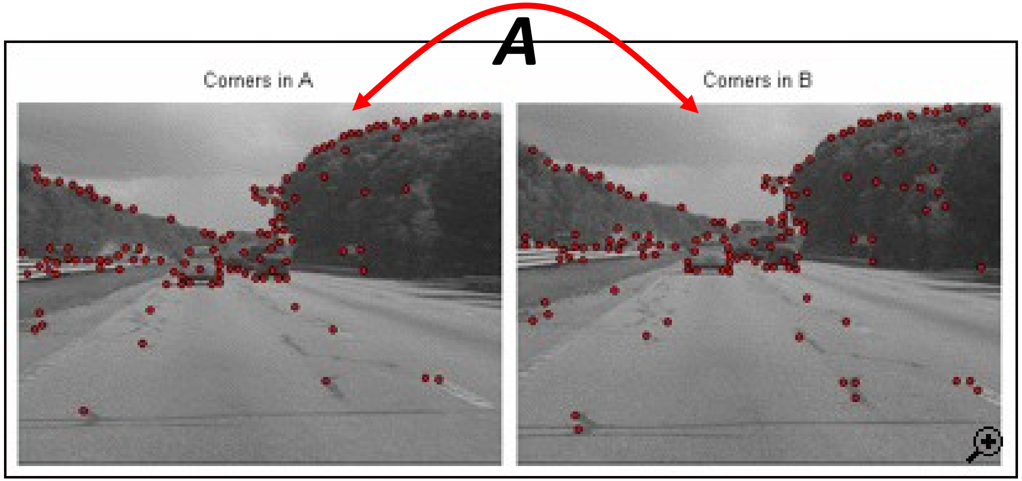
$$\left.\begin{aligned}\mathbf{x}_{t=1} &= \mathbf{A}_1\mathbf{x}_{t=2}\\ \mathbf{x}_{t=2} &= \mathbf{A}_2\mathbf{x}_{t=3}\\ \mathbf{x}_{t=3} &= \mathbf{A}_3\mathbf{x}_{t=4}\end{aligned}\right\} \begin{aligned}\mathbf{x}_{t=1} &= \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{x}_{t=4}\\ \mathbf{A}_{41} &= \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\end{aligned}$$

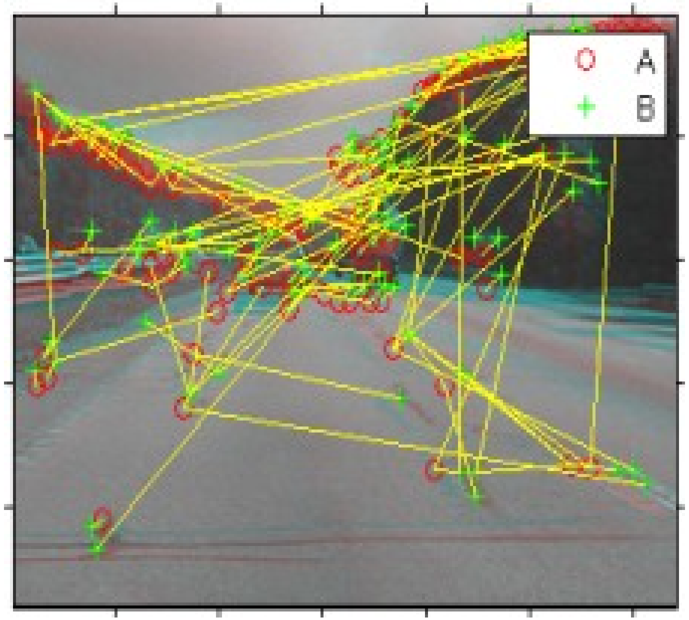
Number of features

- 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 illusion of depth

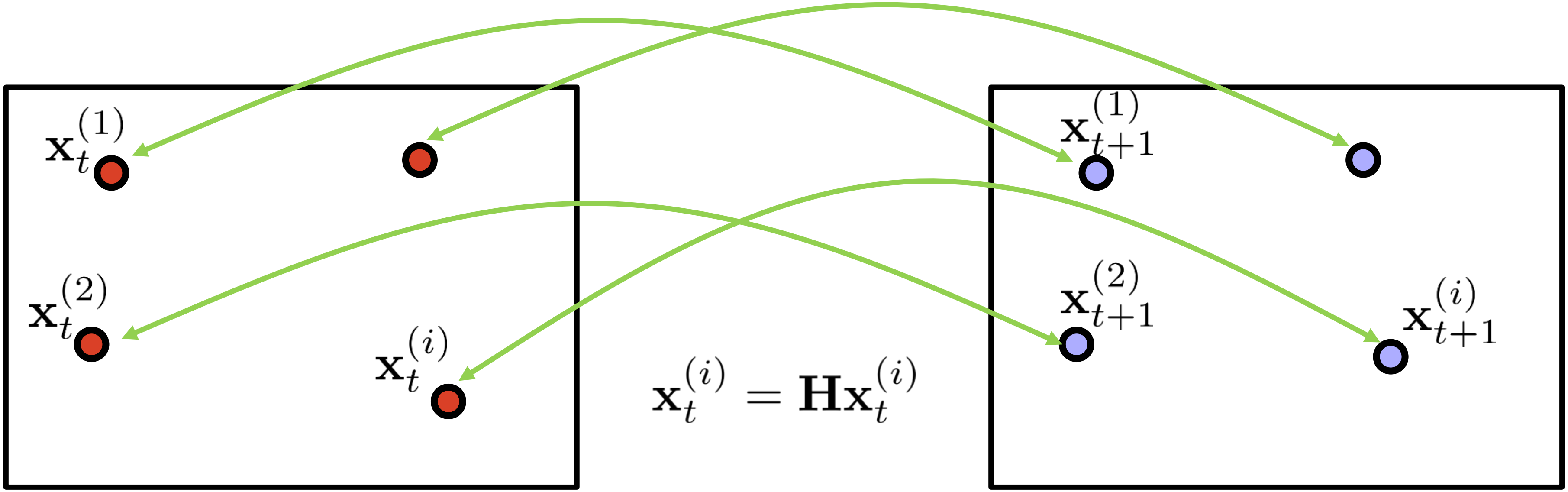
Stabilization using keypoints

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





Deformation model



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

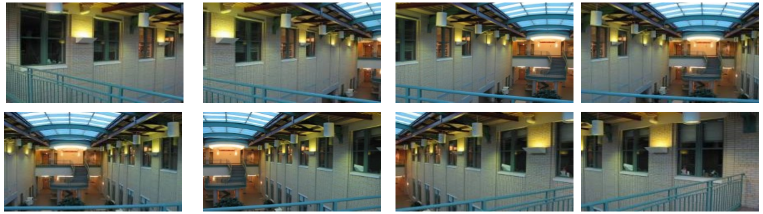
What to do with black border?

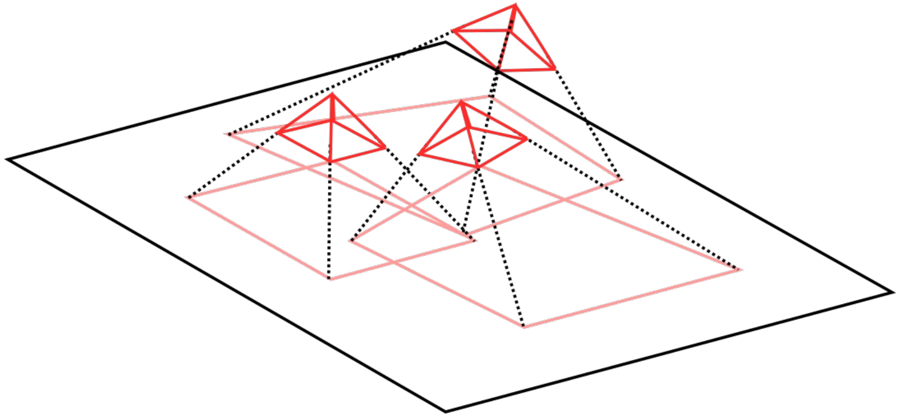
Filling in missing information

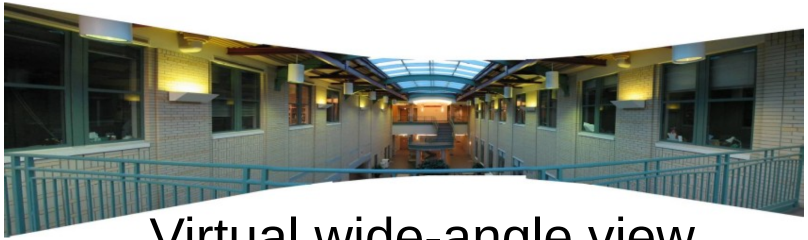
- Cropping viewport
  - Only focusing on always visible part of video
  - Can be problematic with large shifts
- Smoothing trajectory
  - Transfromation filtered with low-pass filter
  - Only jerky motion removed, camera still moves
- Mosaicking

Video mosaicking

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



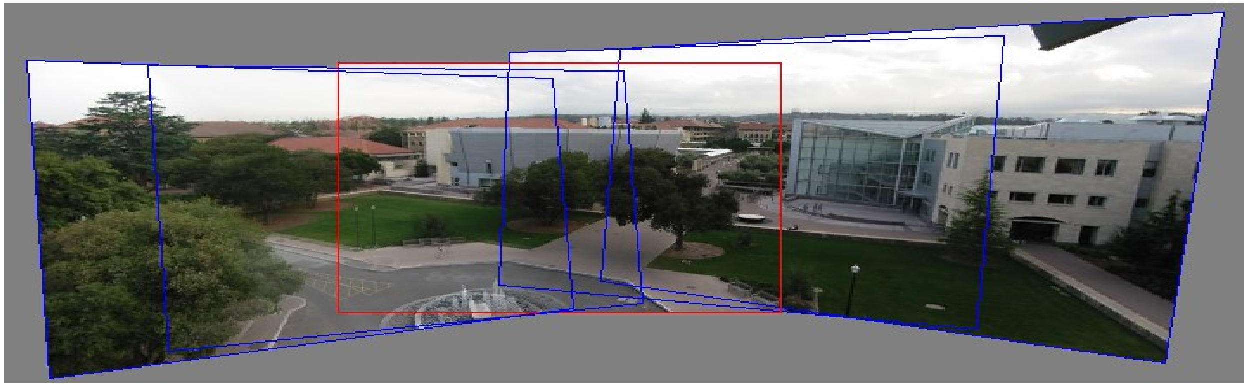




Virtual wide-angle view

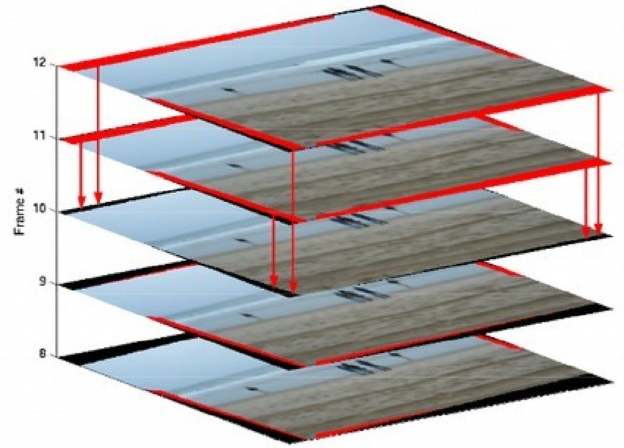
Video mosaicking algorithm

- 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)



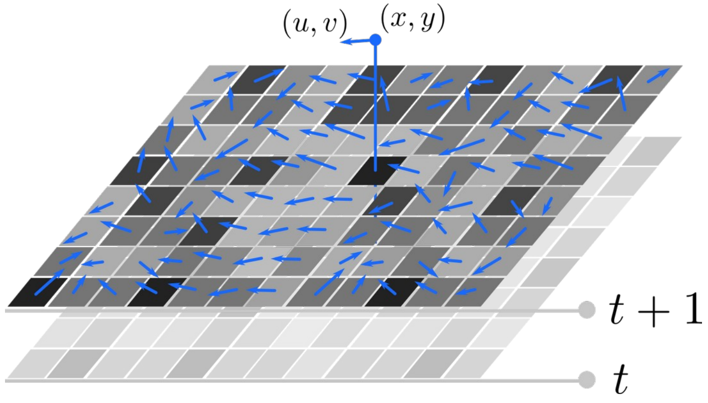
Mosaicking in video stabilization

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



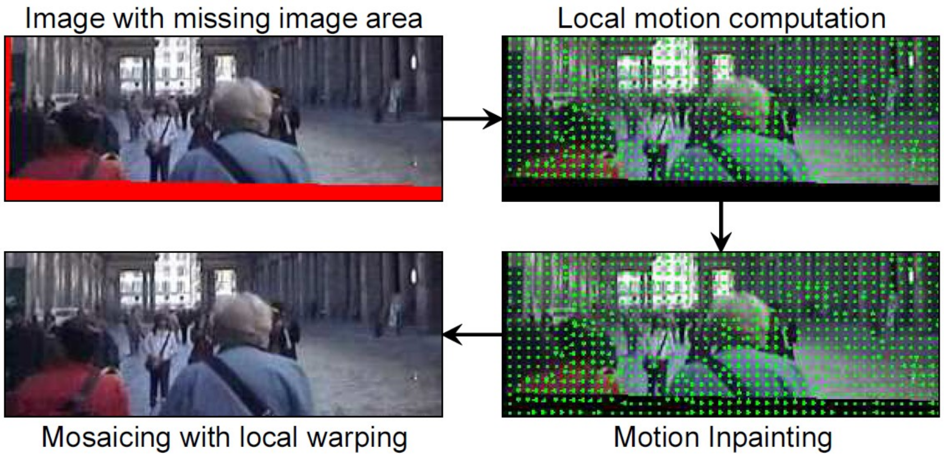
Optical flow stabilization

- Use optical flow instead of keypoints, more dense
  - Lucas & Kanade – fast, local
  - Horn & Schunk – slow, global
  - RAFT – deep learning
- 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



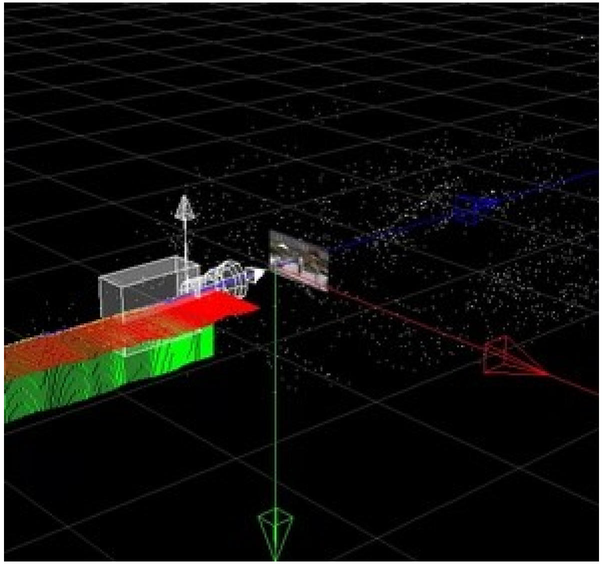
2D stabilization result

Raw video

2D stabilization

Stabilization in space

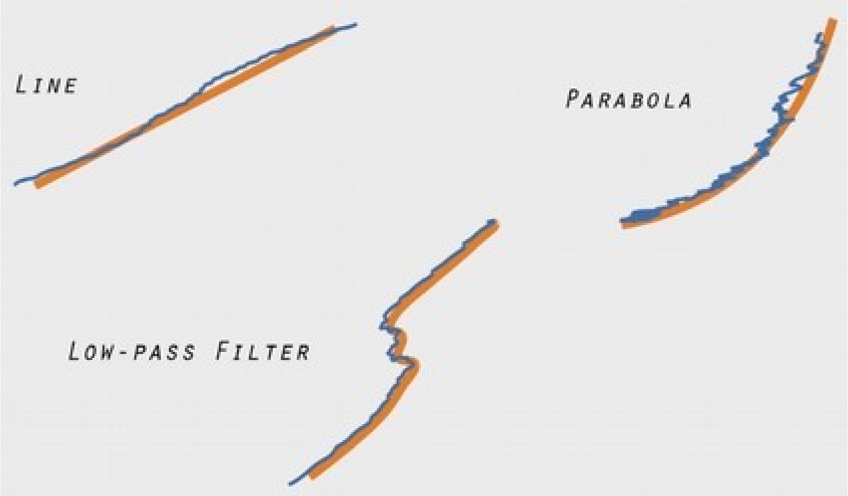
- Reconstruct 3D geometry using Structure from Motion
  - Reconstruction also gives us camera location and translation
- Filter camera path to get smooth path
- Compute warps for modified camera positions and apply them to frames



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.

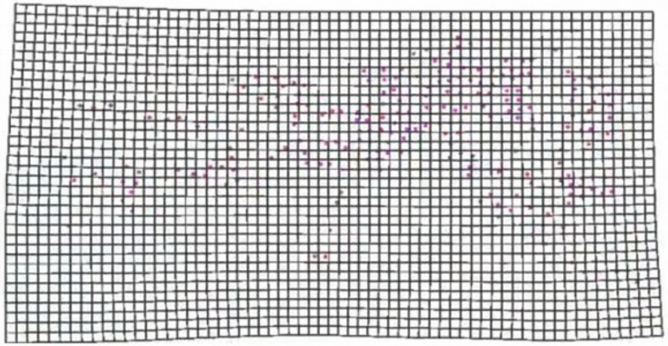
Camera motion types

- 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



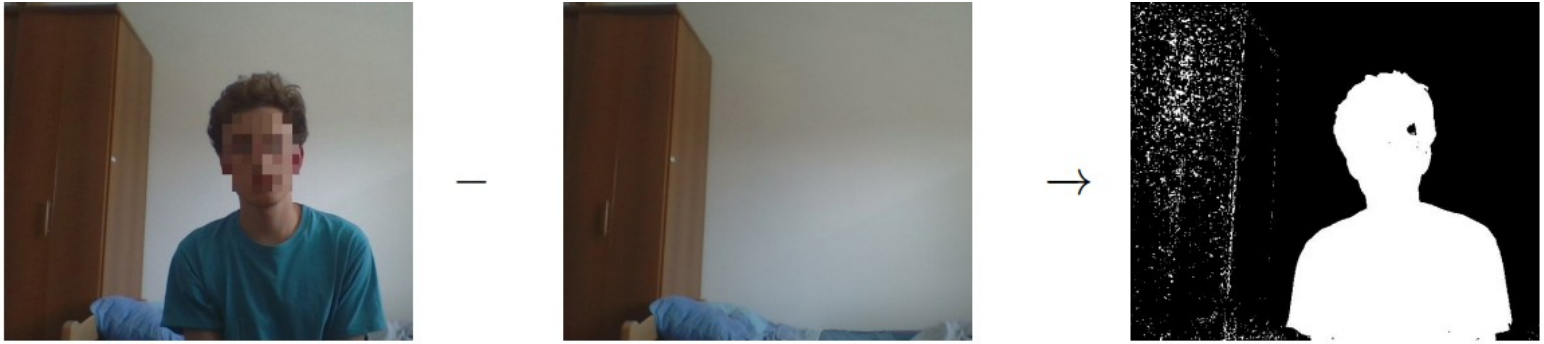
3D stabilization result

Compositing in video

- Replace background
  - Movies
  - Live shows
- Techniques
  - Rotoscoping
  - Background subtraction
  - Chroma key
  - Semantic matting

Background subtraction

- Known background
  - Model per-pixel statistics
  - One or more warm-up frames
  - Compute distance
- Simple implementation
  - Noisy output
  - Static scene
  - Video surveillance



Chroma key

- Monotonous background color
  - Green screen
  - Blue screen
- Reference color distance
  - Threshold
  - Postprocessing

Chroma key issues

- Limits foreground
  - Wardrobe issues
  - Reflective surfaces
- Color bleed/spill

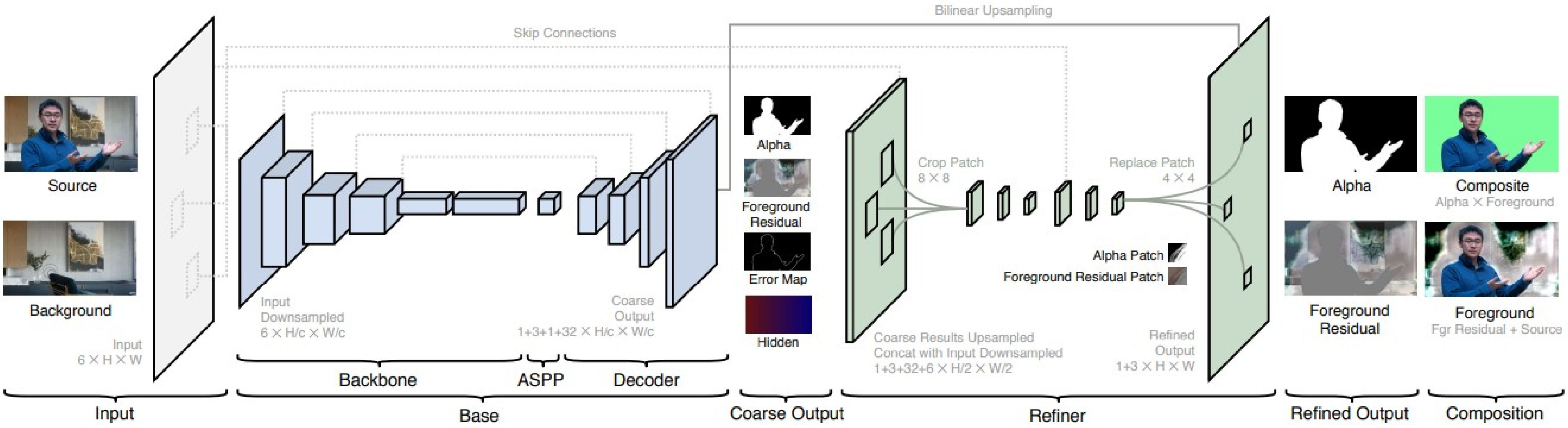
Virtual sets

- Projected backgrounds
  - Pre-recorded video
- LED screens
  - Camera tracking
  - Real-time rendering

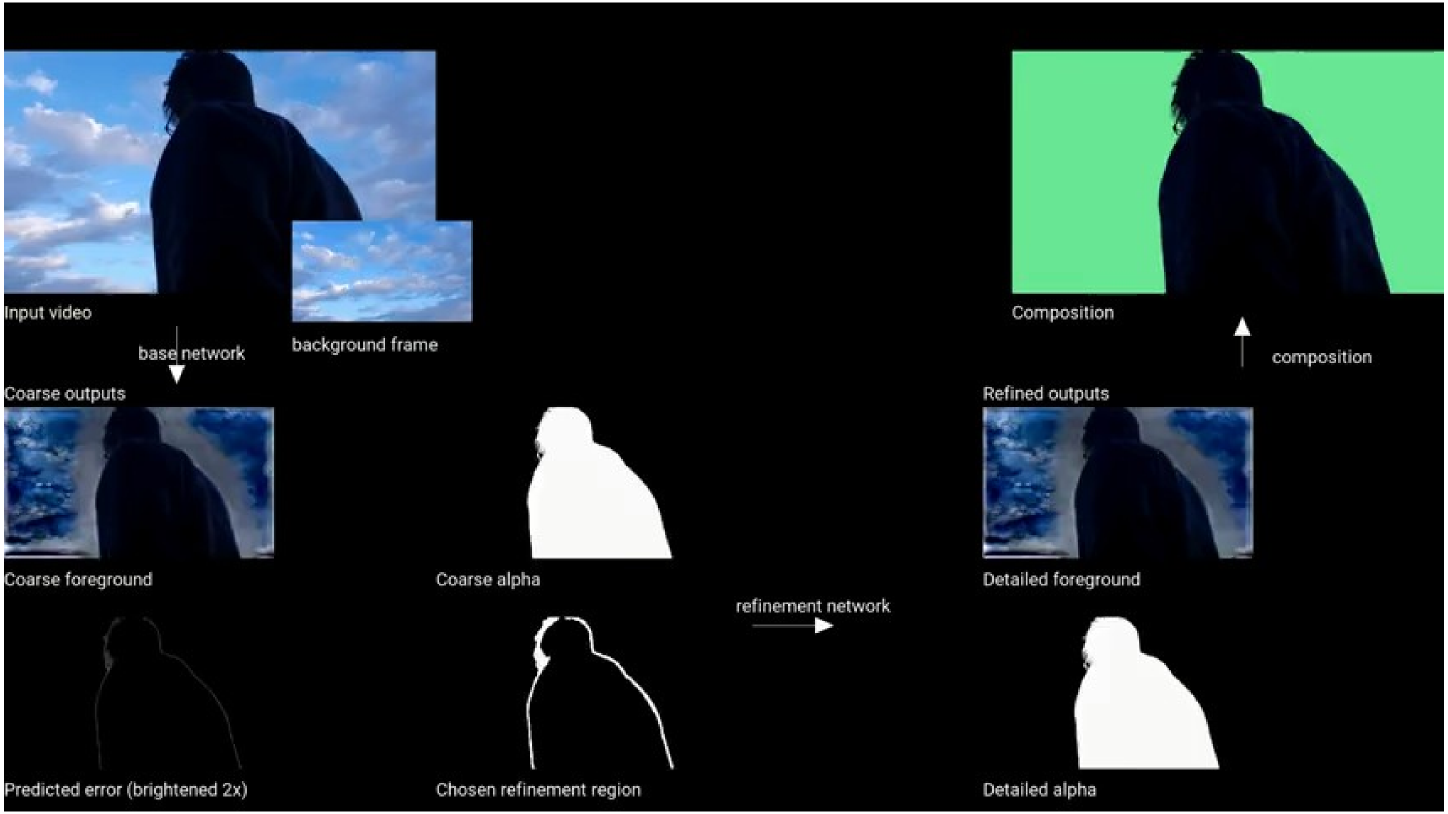
Semantic segmentation

- Use deep learning to predict mask
  - Foreground separation
  - Matting
- Training
  - Green-screen videos
  - Focus on borders

Model pipeline



Examples

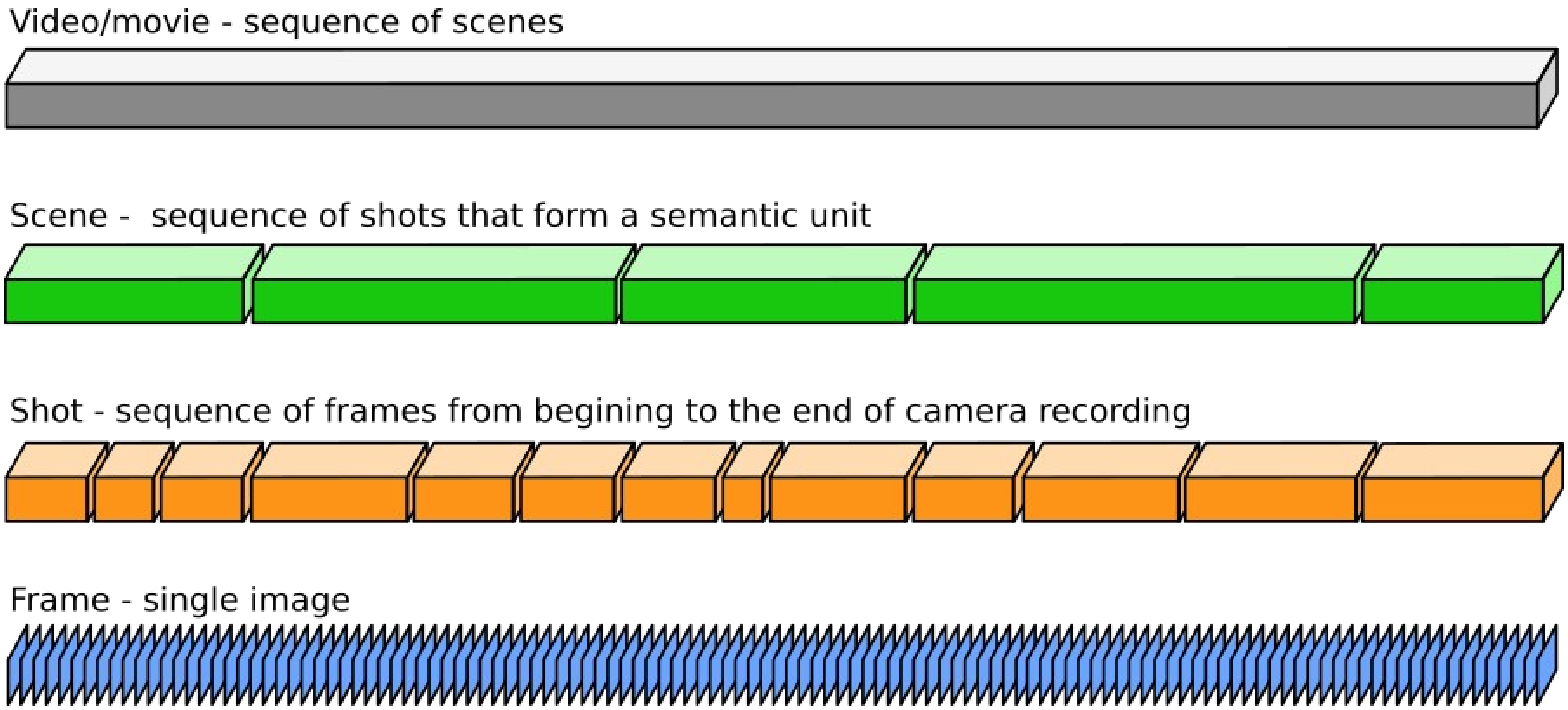


Input video
background frame
base network
Coarse outputs
Coarse foreground
Coarse alpha
Predicted error (brightened 2x)
Chosen refinement region
refinement network
Composition
composition
Refined outputs
Detailed foreground
Detailed alpha

Video as sequence of shots

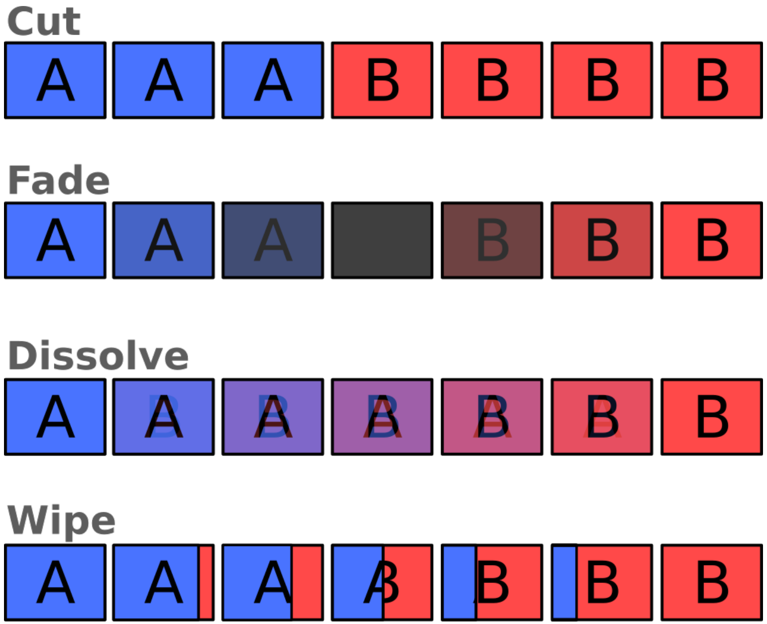
- Shots are useful start to detect scenes
  - Grouping shots into semantic units
  - Enable semantic retrieval in video
  - Easier navigation, understanding
- Manual segmentation of video into shots is slow
  - About 10 hours per 1 hour of video (for a movie)
  - Easier if edit decision list is available (unreliable)
- Automatic detection of shots
  - Detecting boundaries - transitions

Video structure



Transition types

- Cut
  - Sharp transitions between shots
  - Sudden change of all pixels in the frame
- Fade
  - Fade-out – gradual transition to color
  - Fade-in – gradual transition from color
  - Dissolve – gradual transition between shots
  - Wipe – gradual erase

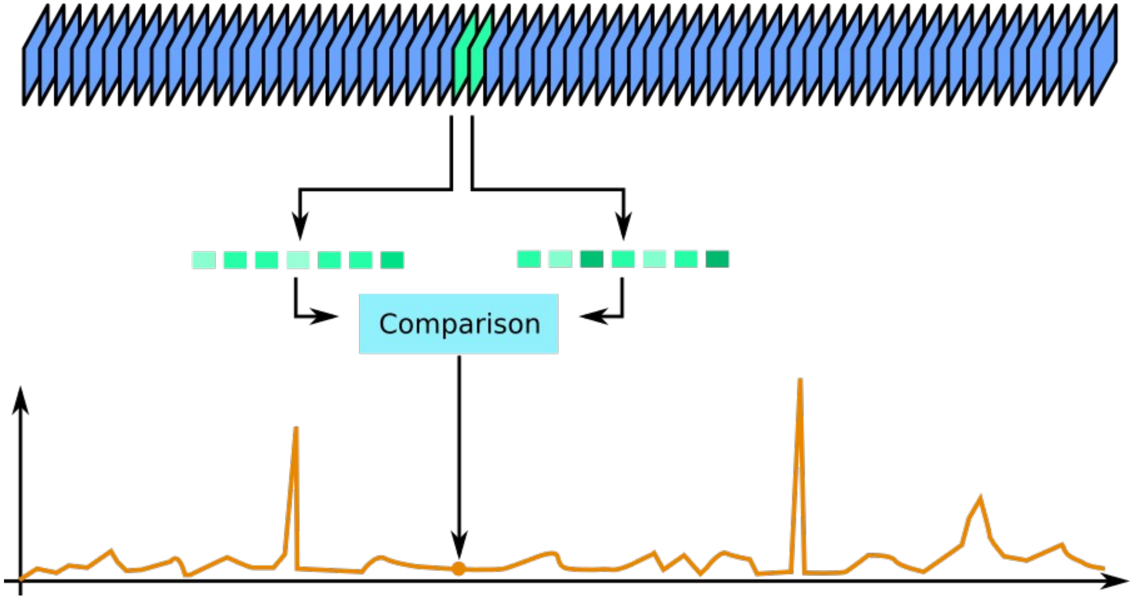


Detecting transitions

- Describe frame content
  - Features: color, texture, edges, etc.
- Measure difference
  - Two frames
  - Multiple frames
- Difference large enough
  - Threshold
  - Adaptive measures

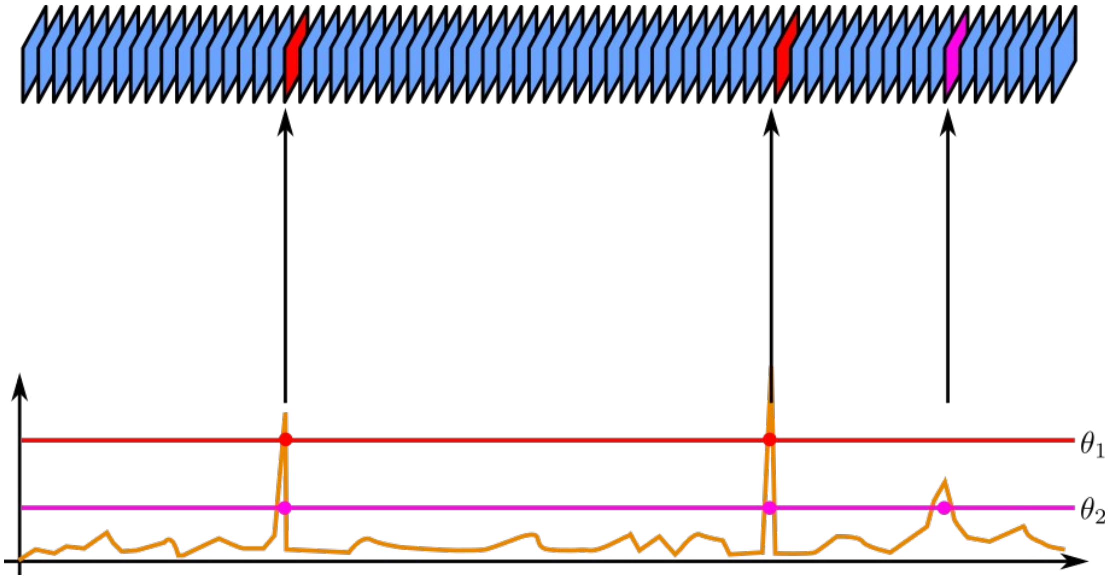
Detecting cuts

- Assumptions
  - Almost stationary
  - Almost constant scene
  - Constant illumination
- Cut if significant change
  - Color
  - Intensity
- Descriptors
  - Gaussian model
  - Histograms



Setting a threshold

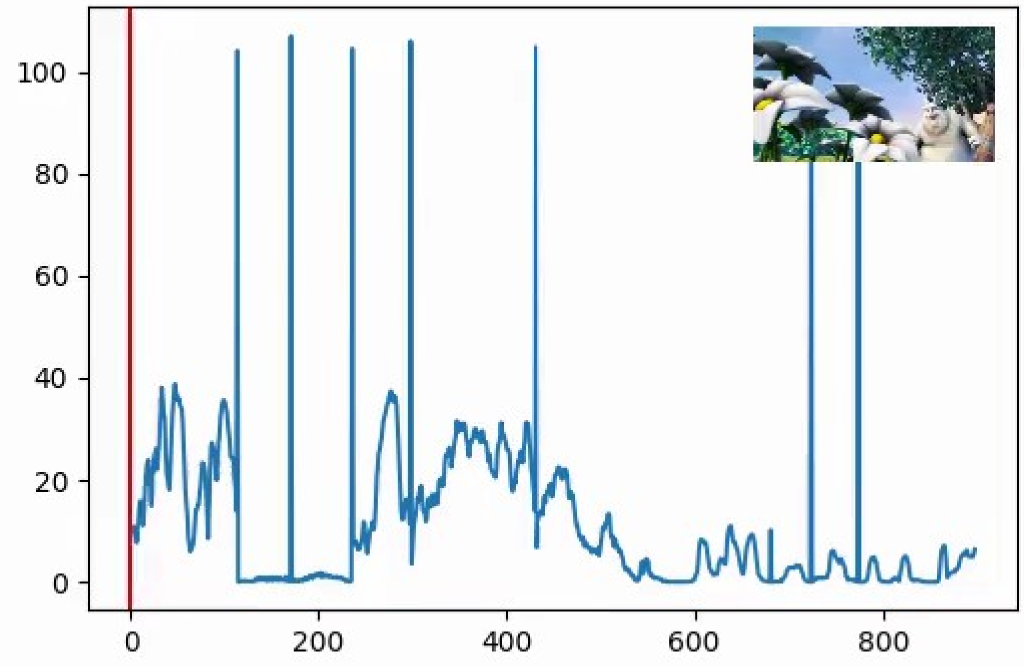
- Distance between consecutive frames
- How to set cut detection threshold?
  - Global methods
  - Adaptive methods



Y. Yusoff, W. Christmas, J. Kittler, „Video Shot Cut Detection Using Adaptive Thresholding“, BMVC 2000

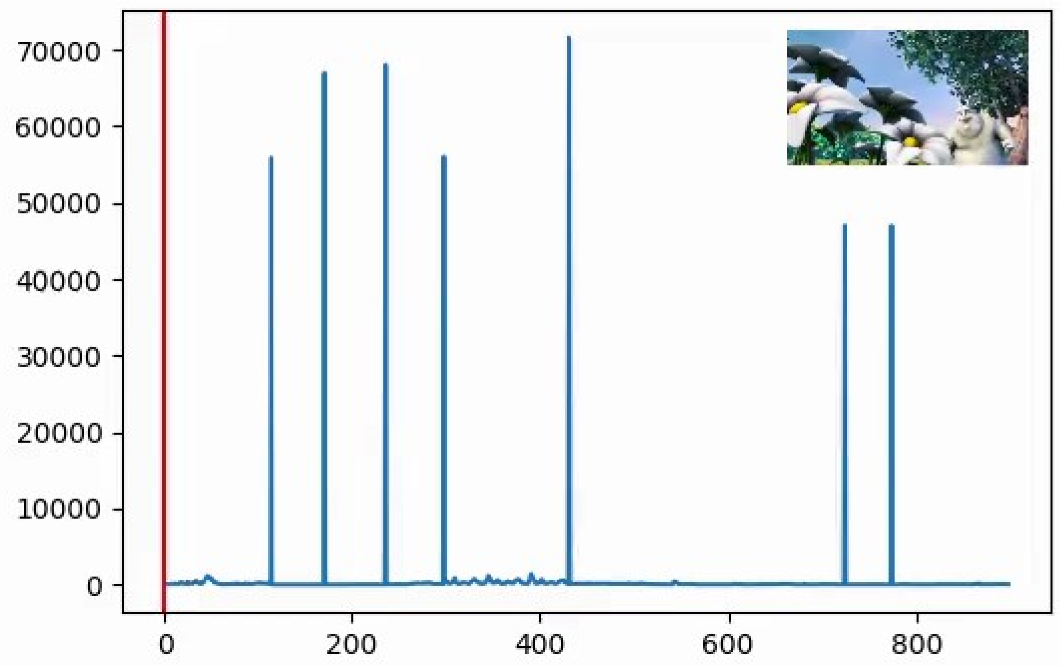
Detecting cuts with MSE

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Xi - Yi)^2$$



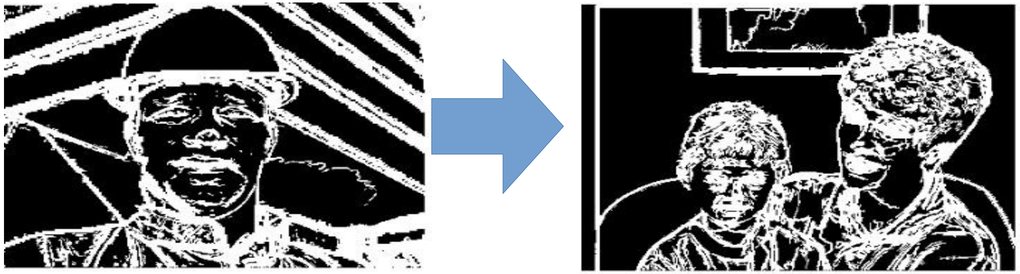
Detecting cuts with histograms

$$X^2 = \frac{1}{2}\sum_{i=1}^{B}\frac{(x_i - y_i)^2}{(x_i + y_i)}$$



Detecting cuts with edges

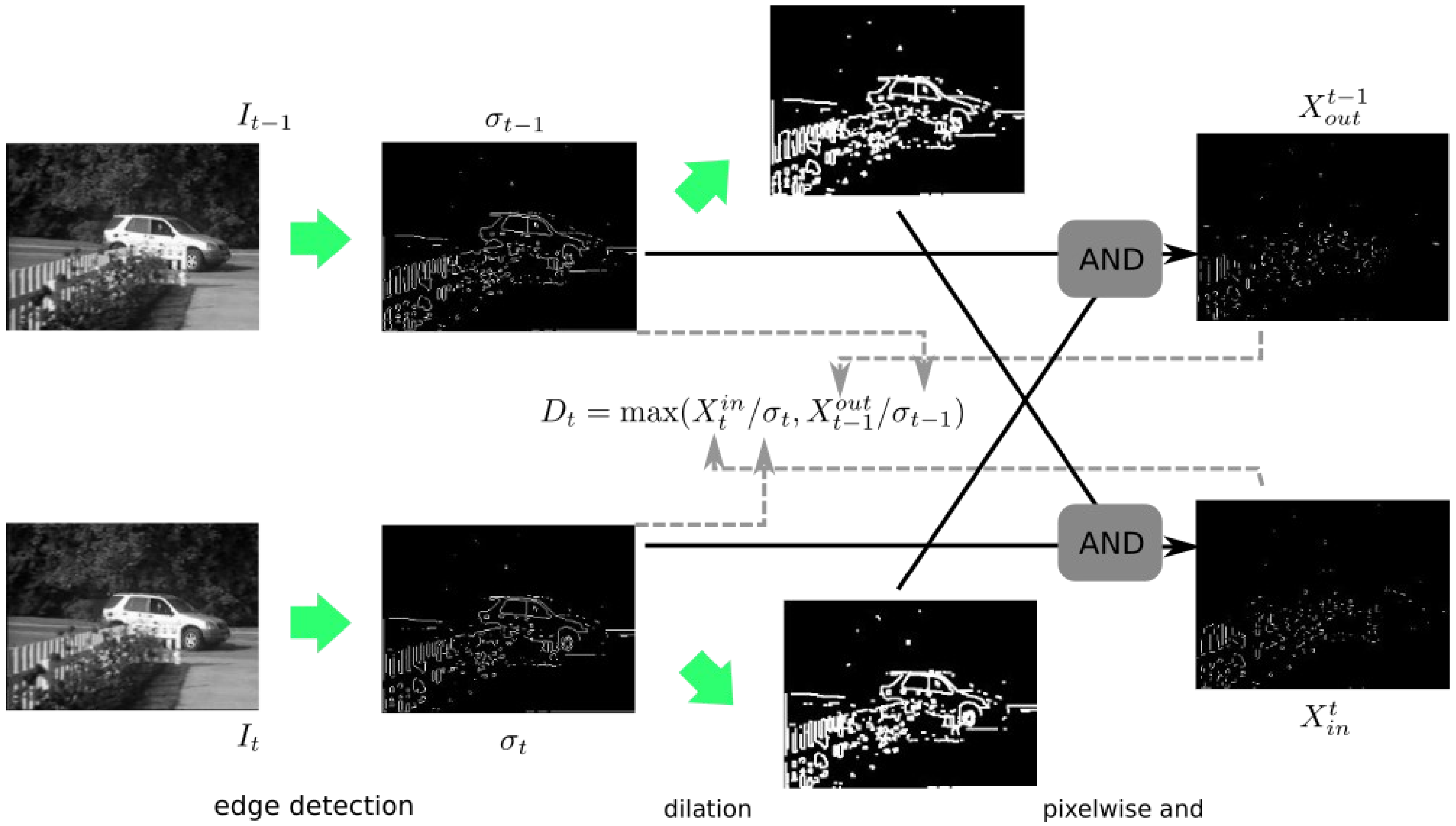
- Color methods are not robust to illumination changes
- Compare edge pixels
  - How many appeared
  - How many vanished



$$D_t = \max(X_t^{in}/\sigma_t, X_{t-1}^{out}/\sigma_{t-1})$$

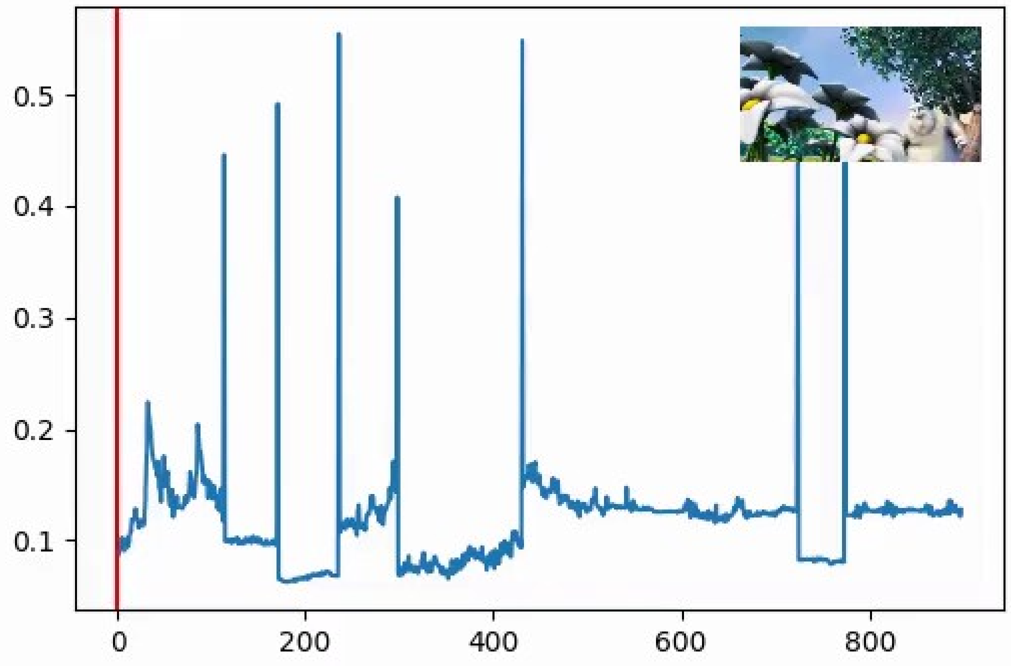
$X_t^{in} \cdots$ number of new edges at time t
$X_{t-1}^{out} \cdots$ number of vanished edges at time $t-1$
$\sigma_t \cdots$ number of all edges at time t
$\sigma_{t-1} \cdots$ number of all edges at time $t-1$

Algorithm



I_{t-1}
σ_{t-1}
X_{out}^{t-1}
AND
$D_t = \max(X_t^{in}/\sigma_t, X_{t-1}^{out}/\sigma_{t-1})$
AND
I_t
σ_t
X_{in}^t
edge detection
dilation
pixelwise and

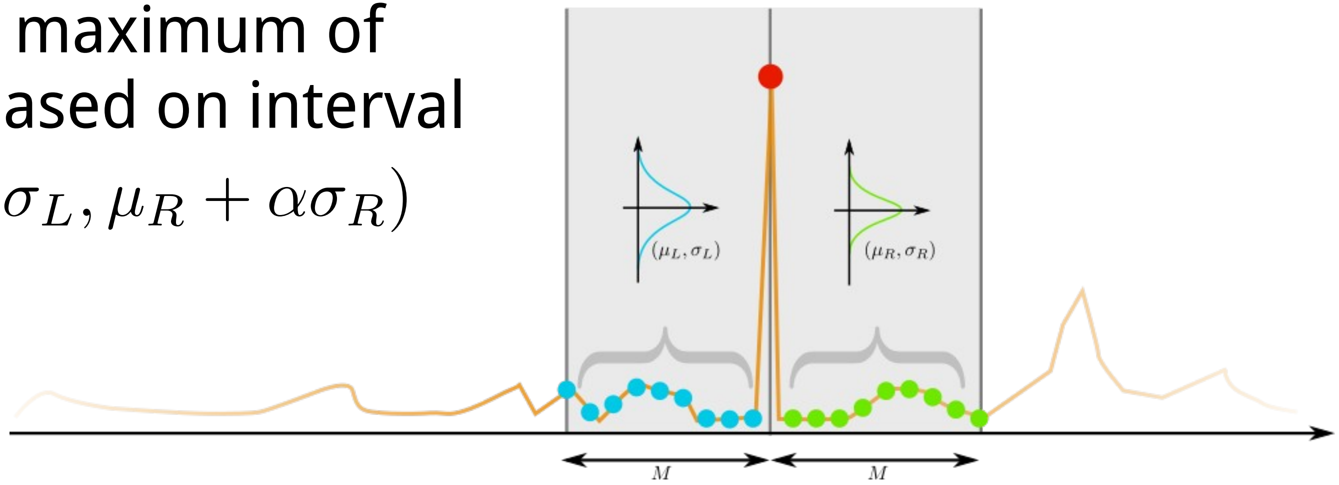
Detecting cuts with edges



Adaptive threshold

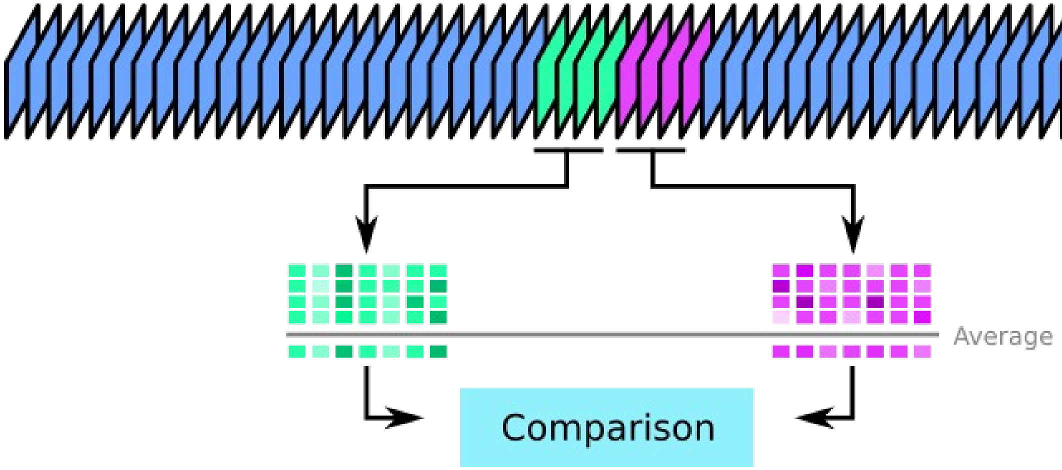
- Cut changes result in sharp peaks
- Frame t is a cut frame if D_t
  - is the largest in interval $[t - M, t + m]$
  - is larger than the maximum of scaled variance based on interval

$$D_t > \max(\mu_L + \alpha\sigma_L, \mu_R + \alpha\sigma_R)$$



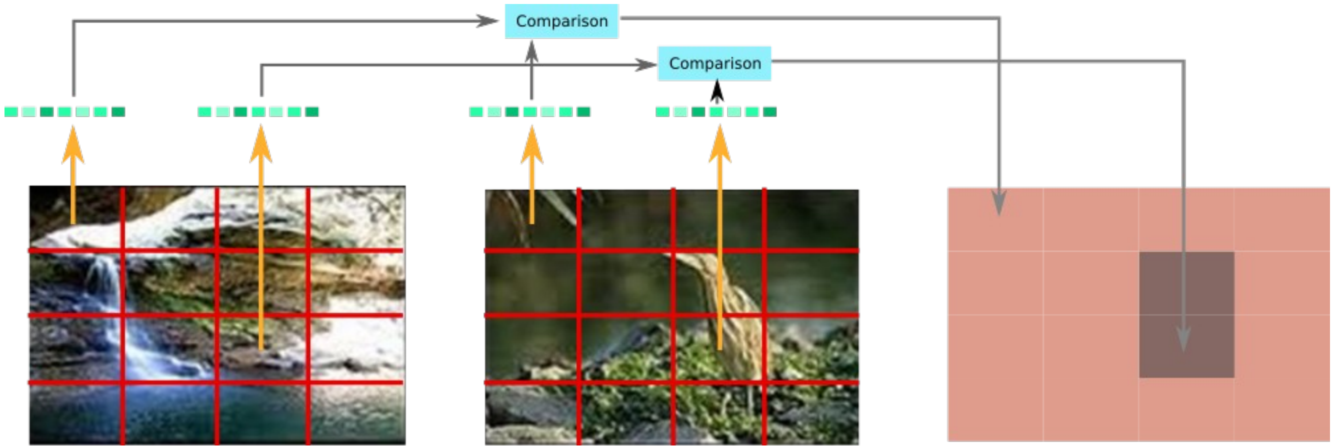
Temporal averaging

- Not enough or too much change between two frames
- Average several consecutive descriptors



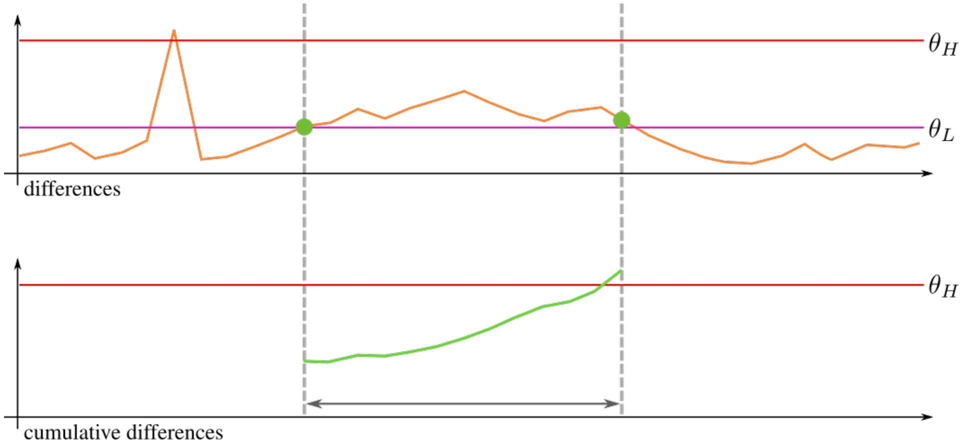
Partial changes

- Global descriptors do not consider locality of changes
- Compute distances between frames for blocks
  - Ignore change if less than N blocks change
  - Compute overall distance



Detecting fades

- Not a lot of change between two frames
- Two stage threshold
  - Low threshold – potential fade start
  - Comparing to the start frame
  - Measure difference until it is increasing
  - Compare to the high threshold



B. Shahraray, "Scene change detection and content-based sampling of video sequences" Digital Video Compression: Algorithms and Technologies, 1995