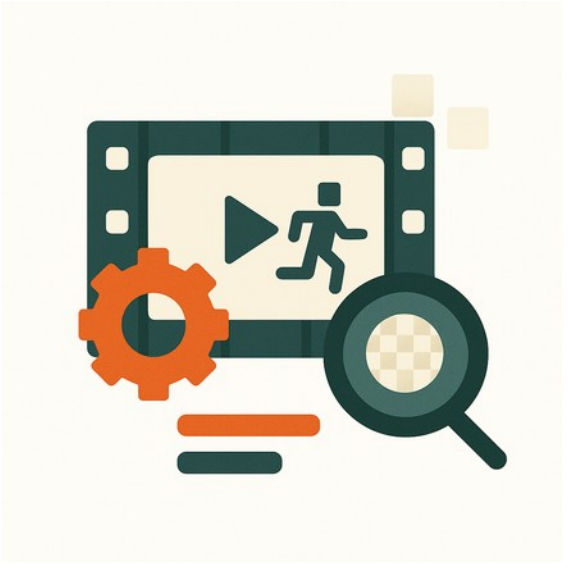


Video Processing

Video Perception

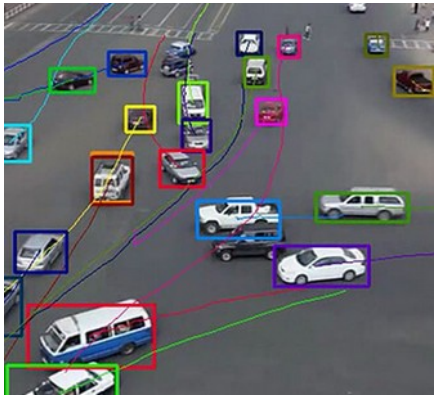
- Human vision integrates over time (~10–20 ms windows)

Processing Video

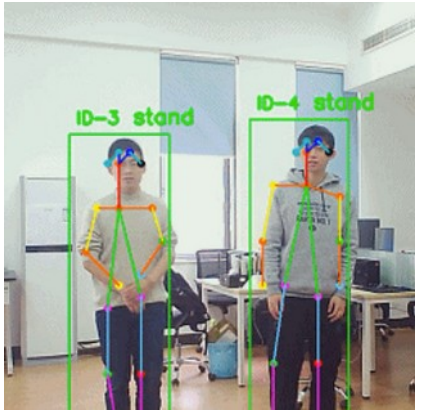
- Video = sequence of images
  - Frames per second
  - Locality of motion
- Animate image processing
  - Process every frame individually
  - Change parameters over time
- Analyze motion (work with multiple frames)
- Detect transitions

Motion Analysis

Optical Flow
Pixel-level, Local



Object Tracking
Entire video, Entire objects



Action Recognition
Actions, Understanding

Motion Estimation in Multimedia

- Estimate movement in video
  - Scene movement
  - Object movement
  - Camera movement
- Building block of many applications
  - Video stabilization
  - Video compression
  - 3D reconstruction
  - Augmented reality

Optical Flow

- Perceived motion in pixel (x,y) at time t
- Basic building block
  - Video compression
  - Stabilization
  - Object tracking
  - Scene understanding

https://www.bi.mpg.de/opticflow

Problem Definition

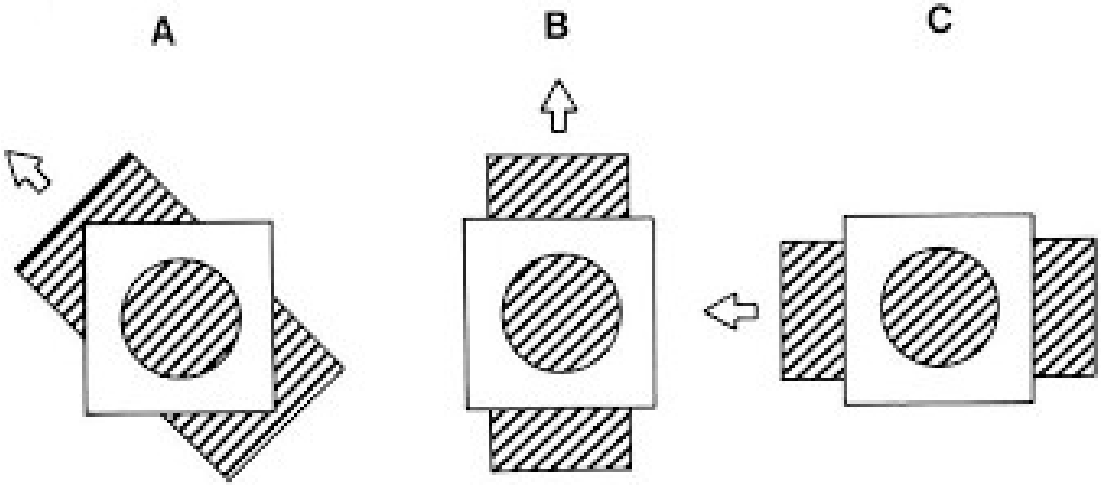
- Assumptions
  - Constant brightness
  - Small displacements
- Calculate displacements over pair of frames

$$I(x, y, t) - I(x + u, y + v, t + 1) = 0$$

$$I_x u + I_y v + I_t = 0$$

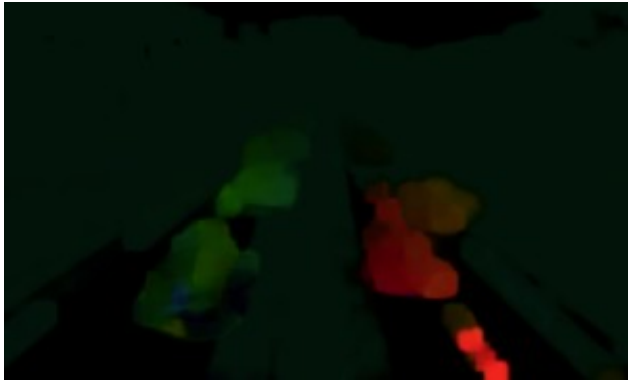
Challenges

- Projection of 3D motion to image
- Aperture problem



Optical Flow Approaches

- Lucas & Kanade – fast
  - Assumes constant velocity in a local patch
  - Considers nearby pixels (least squares)
- Horn & Schunk – slow
  - Optimizes entire flow field
  - Slower
- RAFT
  - Deep learning
  - Synthetic data



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:

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

Simple Feature Tracking

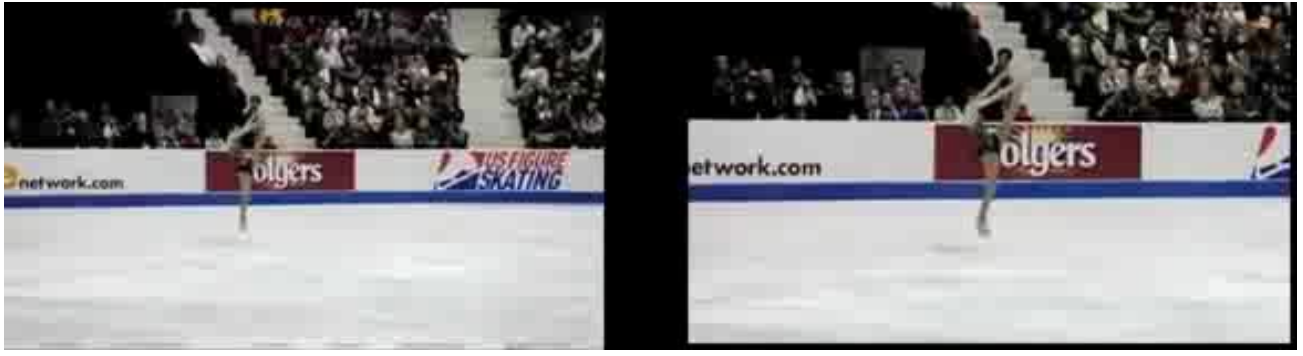
- Input: initial position P_1, frame #1
  - Cut patch of size NxN around P_1 from frame #1
- For fames T = 2 … N
  - Cut search region of size MxM around P_{T-1} from frame #T
  - NCC between patch and search region
  - P_T = Best match location (adjusted for search region offset)

Camera 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

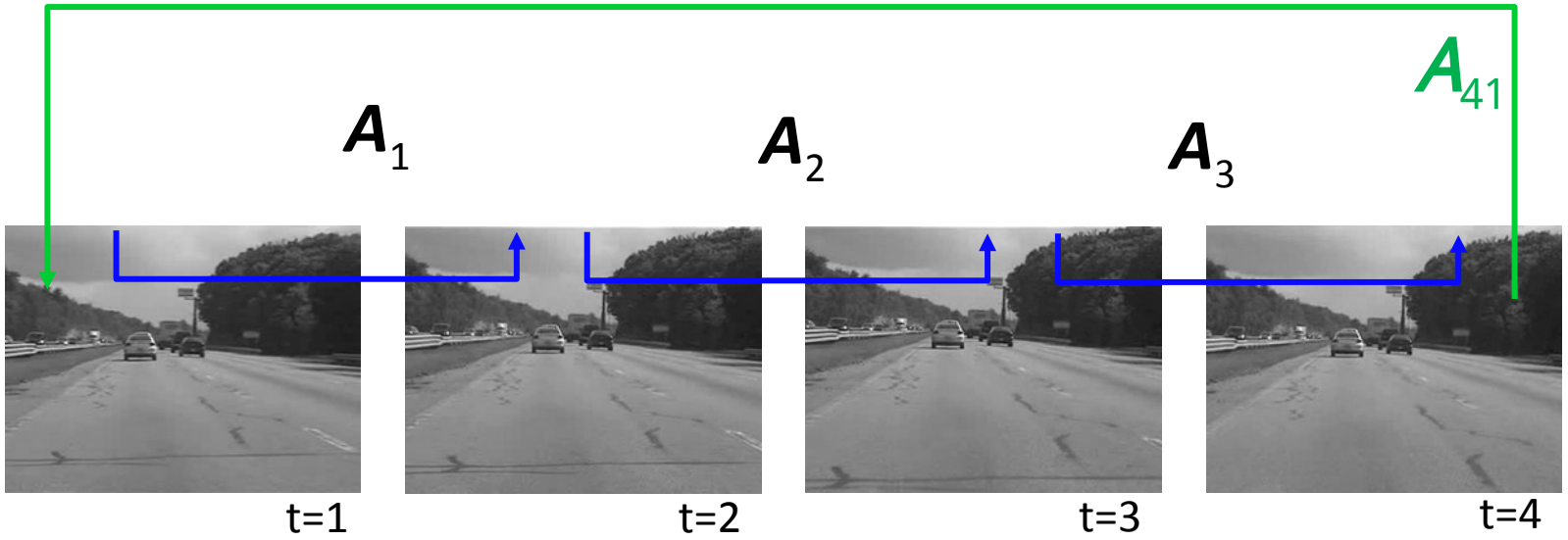
Camera Stabilization Approaches

- 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

Simple Digital Stabilization

- Input: $\{I_1 \ldots I_N\}$
- Output $\{O_1 \ldots O_N\}$
- A := Identity matrix
- For image pair I_i, I_{i+1}
  - A_i := Estimate movement from I_i to I_{i+1}
  - A := A * Ai
  - O_{i+1} := transform using A^{-1}

Transformation Chain



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

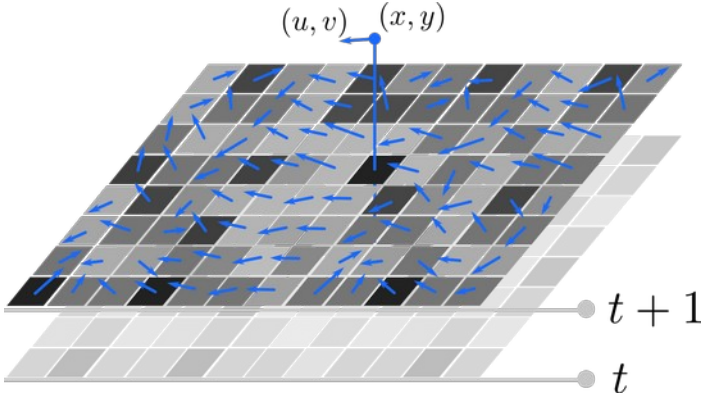
$$\mathbf{A}_{41} = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$$

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 illusion of depth

Optical Flow Stabilization

- Use optical flow instead of keypoints, more dense
- For each pixel compute its most likely translation in the next image
- Fit global transformation to multiple optical flow vectors



Soft Stabilization

- Feature trajectory as a signal
  - High frequencies = shaking, noise
  - Low frequencies = intentional
- Use low-pass filter
  - Filter out noise, keep intentional movement
  - Mind the boundaries (start, end)

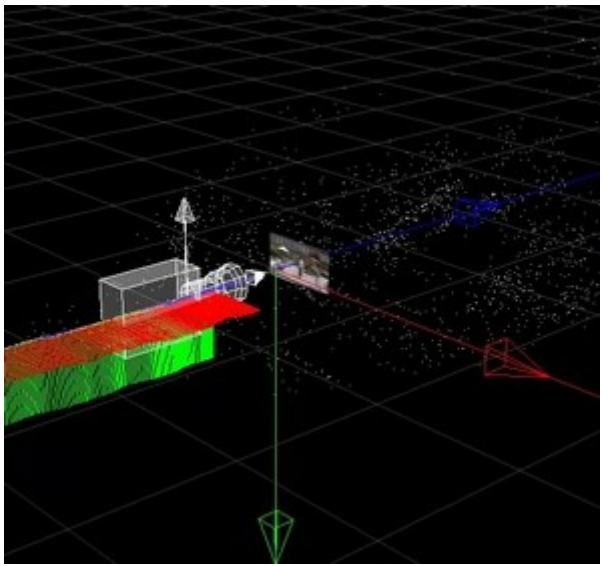
2D stabilization result

Raw video

2D stabilization

Stabilization in space

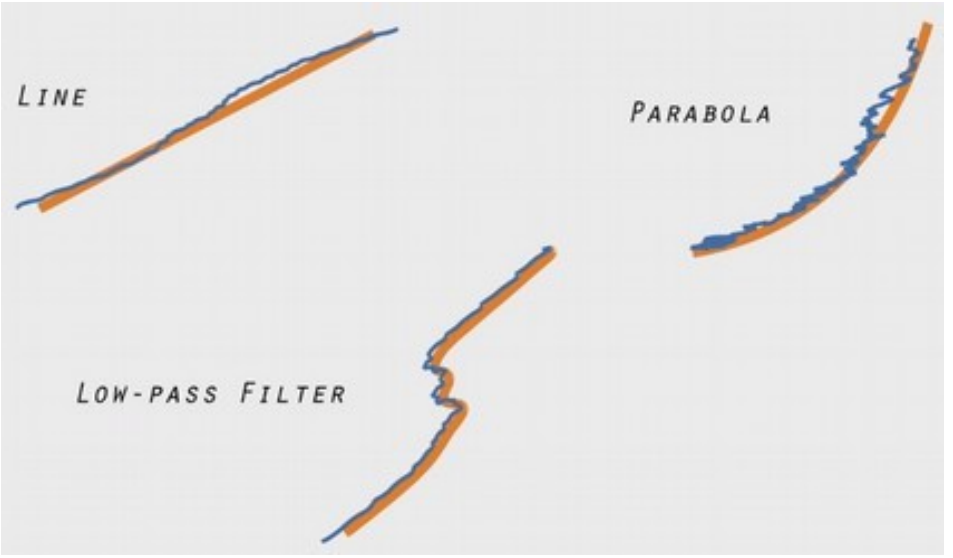
- Reconstruct 3D geometry using Structure from Motion
  - Reconstruction also gives us camera location and translation
- Filter camera path to get smooth path
- Compute warps for modified camera positions and apply them to frames



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.

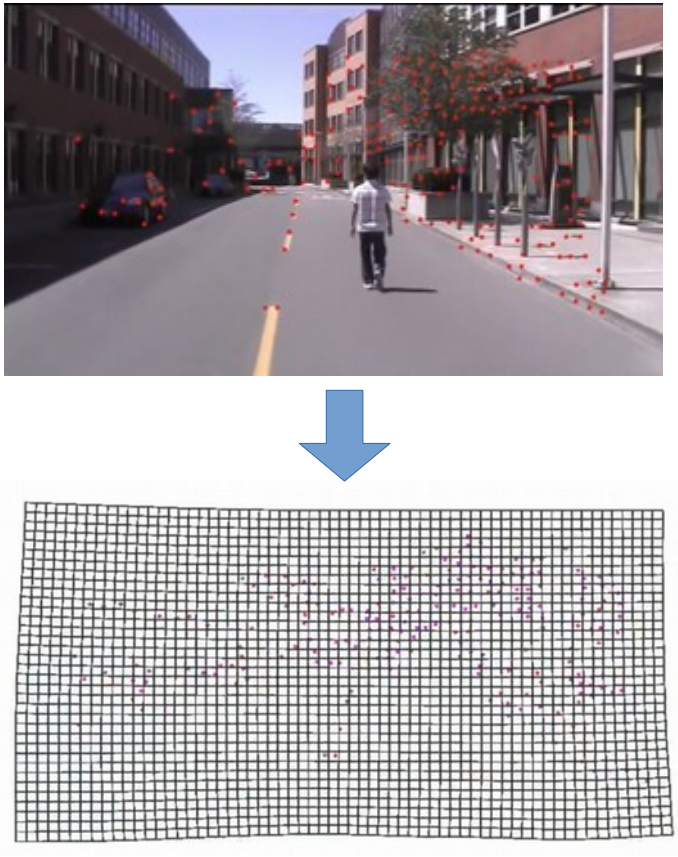
Camera motion types

- 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



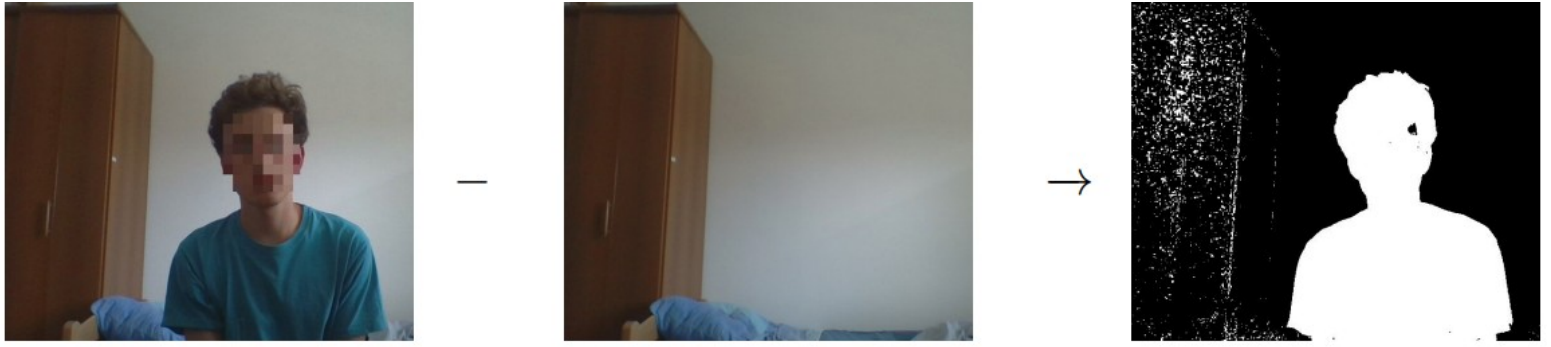
3D Stabilization Result

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 algorithm

- Input: {I_1 ... I_N}, reference color, threshold, background image (B)
- For i := 1 ... N
  - D := distance of all pixels in I_i to reference color
  - M := D < threshold
  - I_i[M] = B[M]

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

Data-driven segmentation

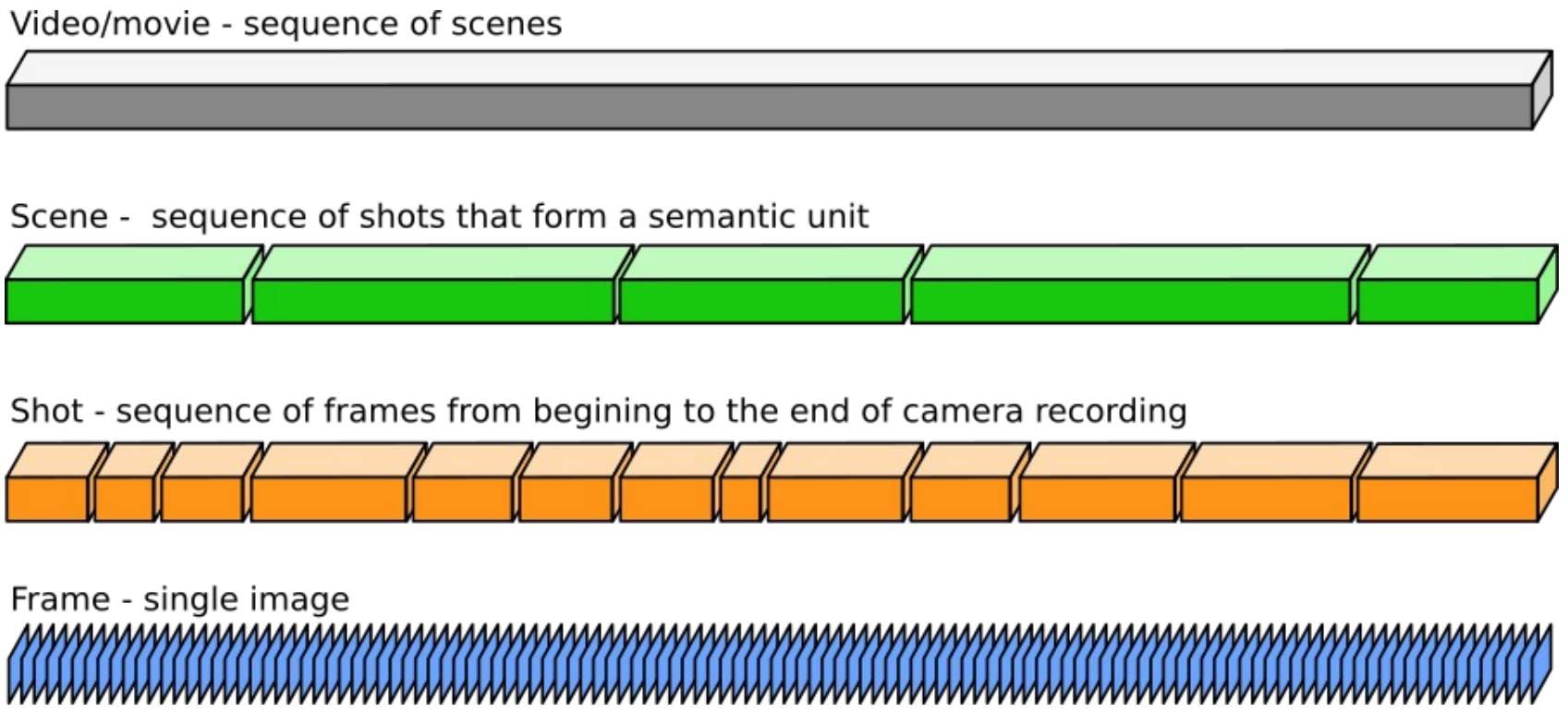
- Use deep learning to predict mask
  - Foreground separation
  - Matting
- Training
  - Green-screen videos
  - Focus on borders

Shanchuan Lin et al., Real-Time High-Resolution Background Matting, CVPR 2021

Video as sequence of shots

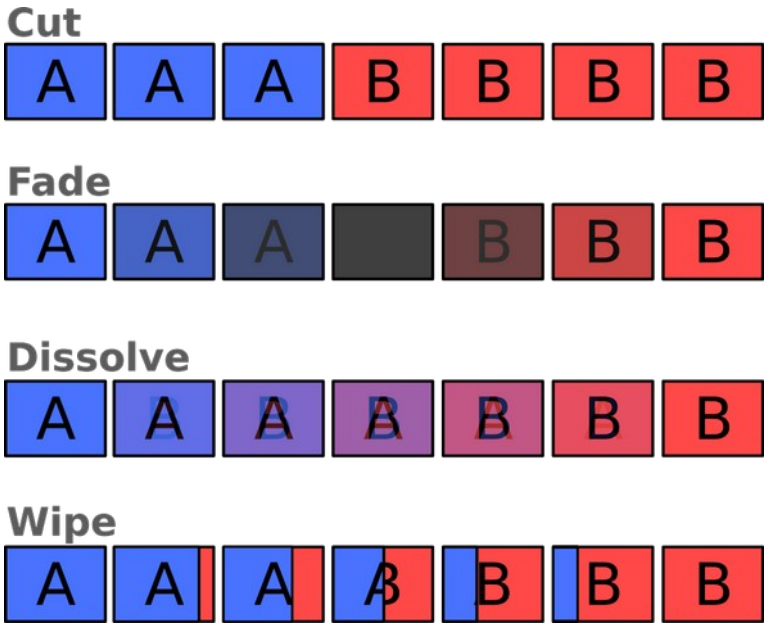
- Shots are useful start to detect scenes
  - Grouping shots into semantic units
  - Enable semantic retrieval in video
  - Easier navigation, understanding
- Manual segmentation of video into shots is slow
  - About 10 hours per 1 hour of video (for a movie)
  - Easier if edit decision list is available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

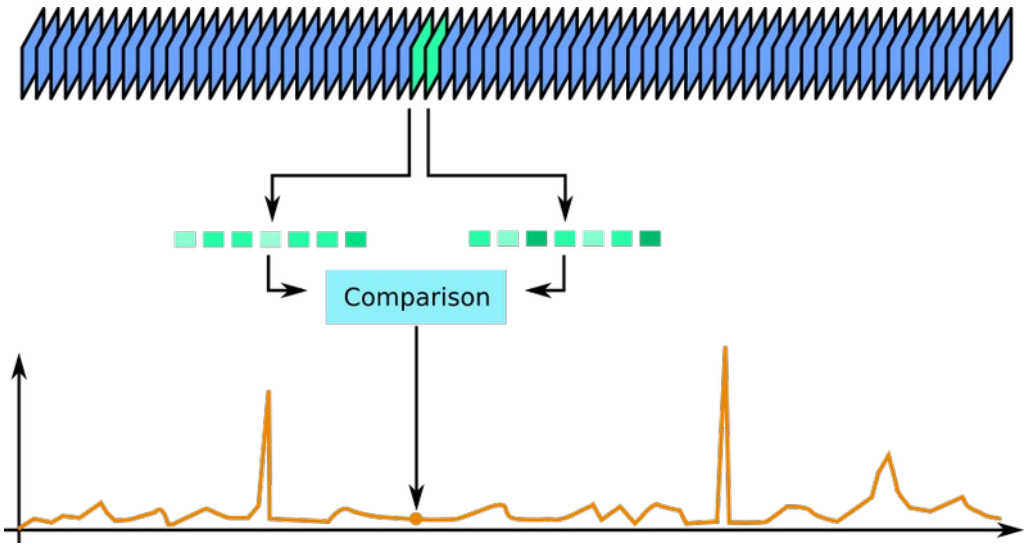


Detecting transitions

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

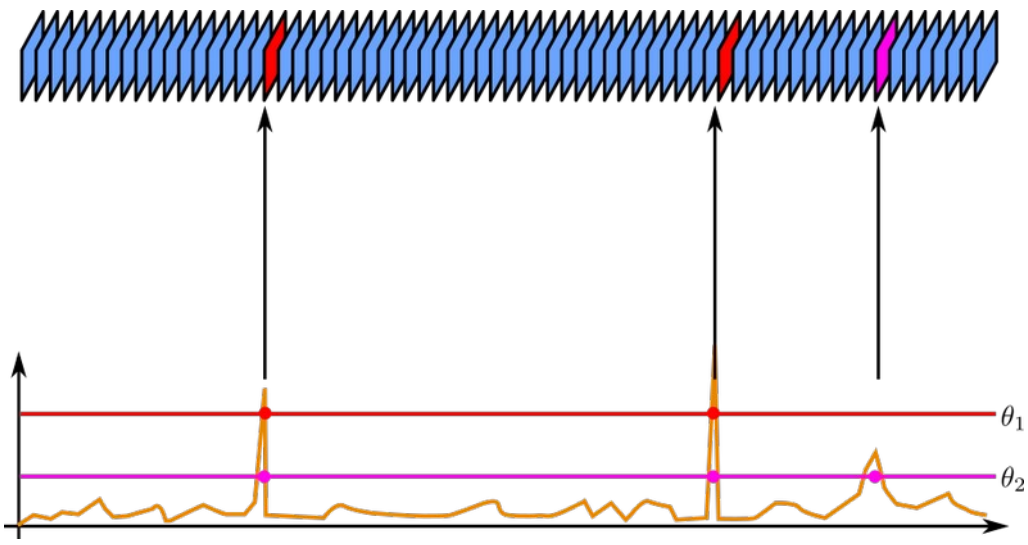
Detecting cuts

- Assumptions
  - Almost stationary
  - Almost constant scene
  - Constant illumination
- Representations
  - Gaussian model
  - Histograms
  - Deep-learning



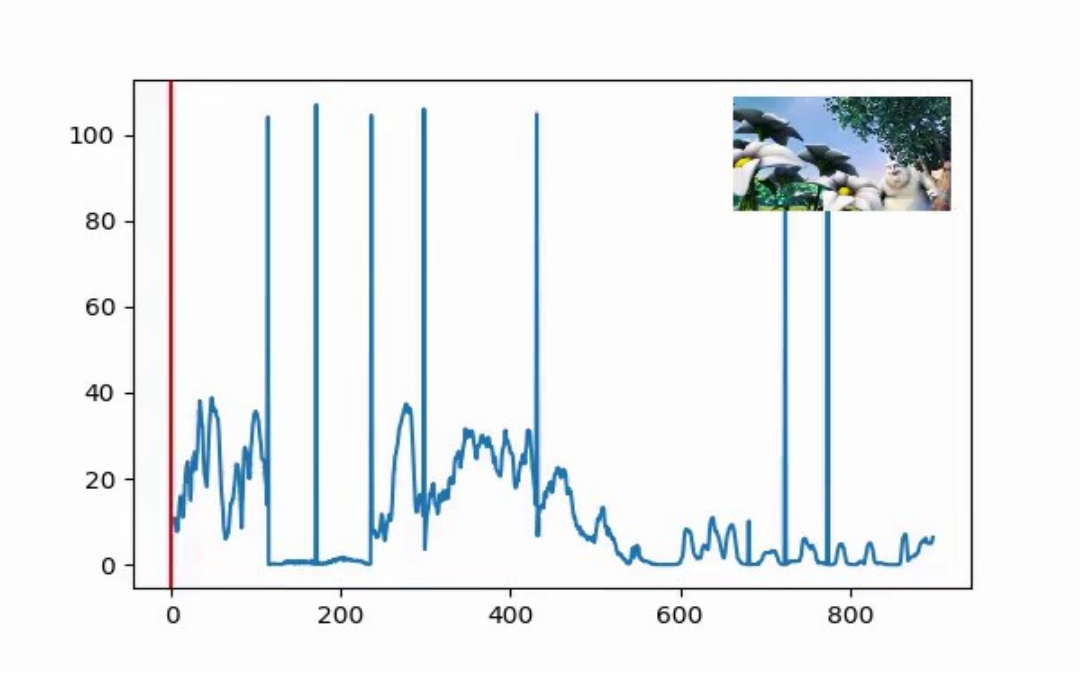
Setting a threshold

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



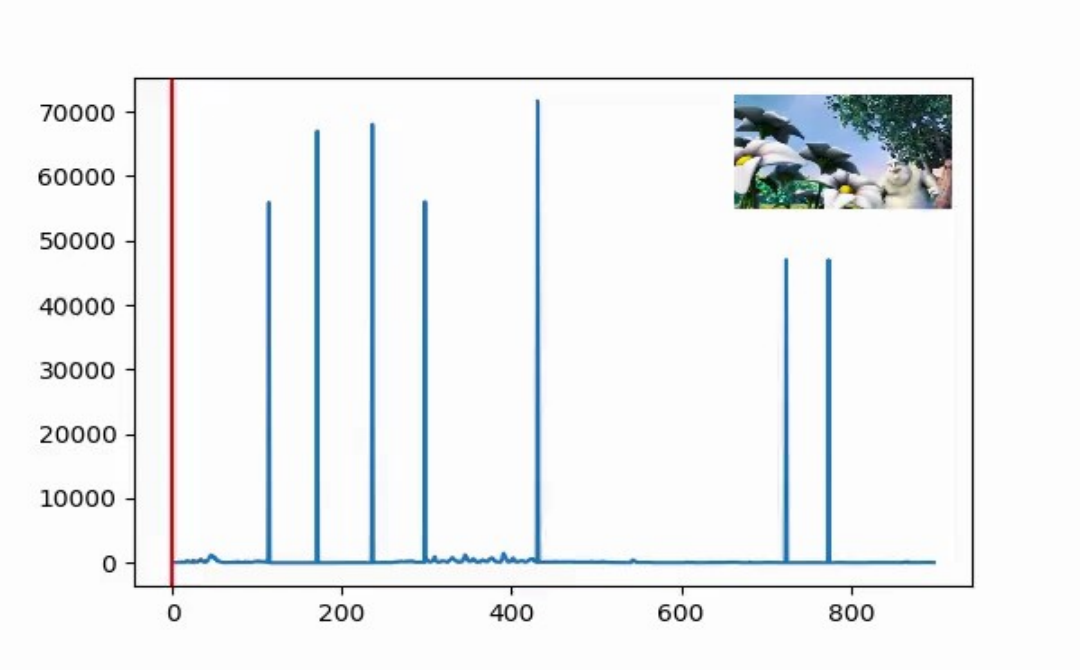
Detecting cuts with MSE

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



Detecting cuts with histograms

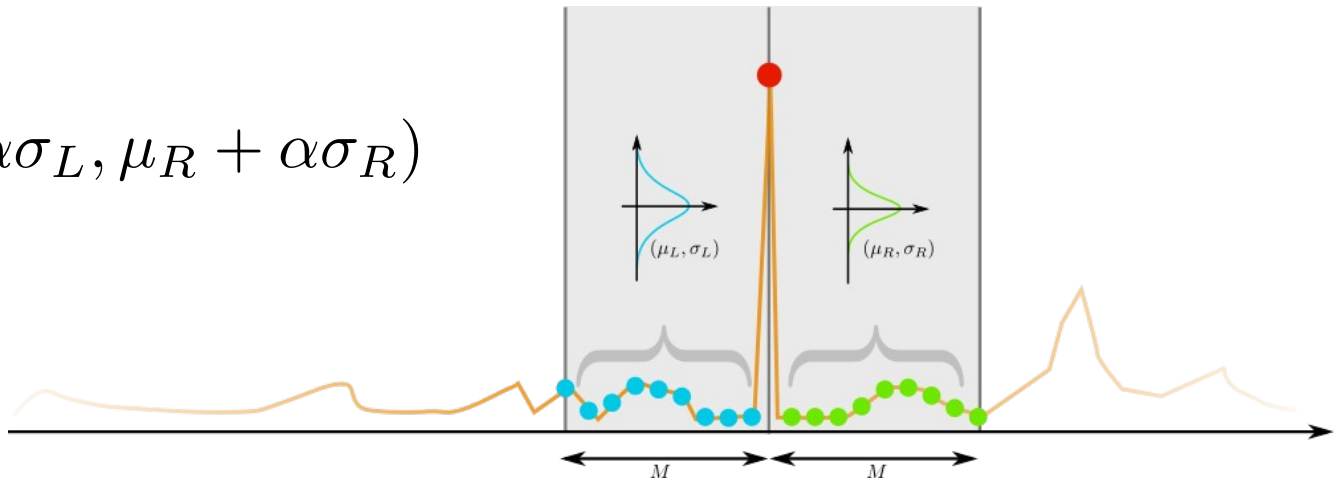
$$X^2 = \frac{1}{2}\sum_{i=1}^{B}\frac{(x_i - y_i)^2}{(x_i + y_i)}$$



Adaptive threshold

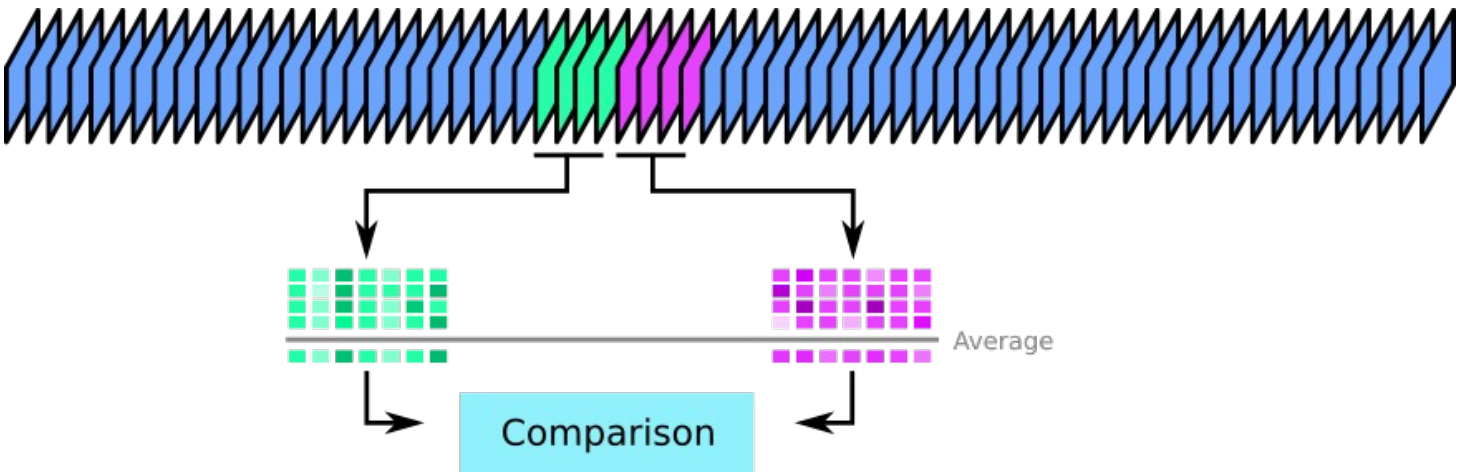
- Cut changes result in sharp peaks
- Frame t is a cut frame if D_t
  - is the largest in interval $[t - M, t + m]$
  - is larger than the maximum of scaled variance based on interval

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



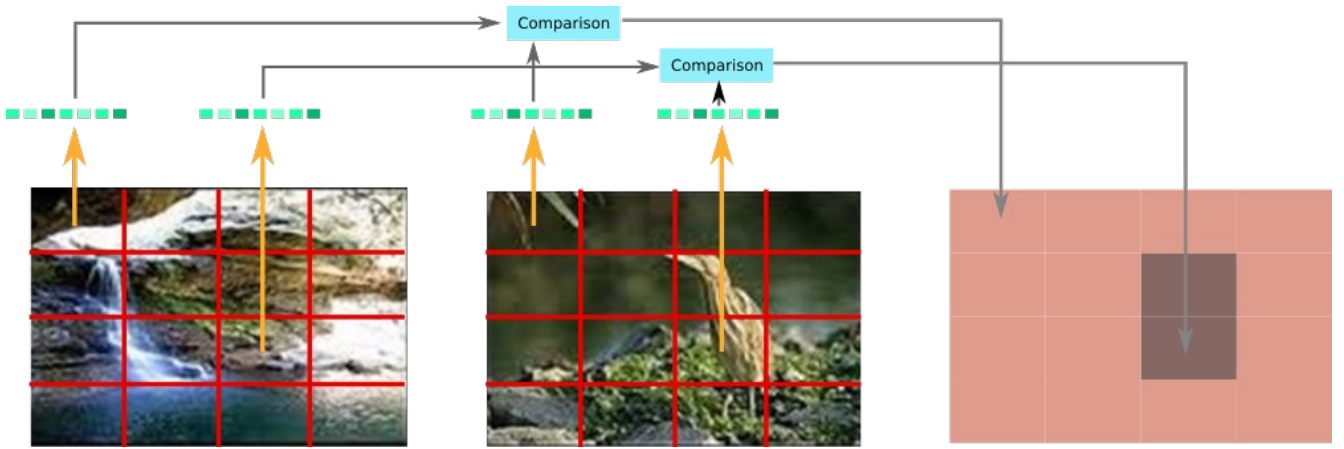
Temporal averaging

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



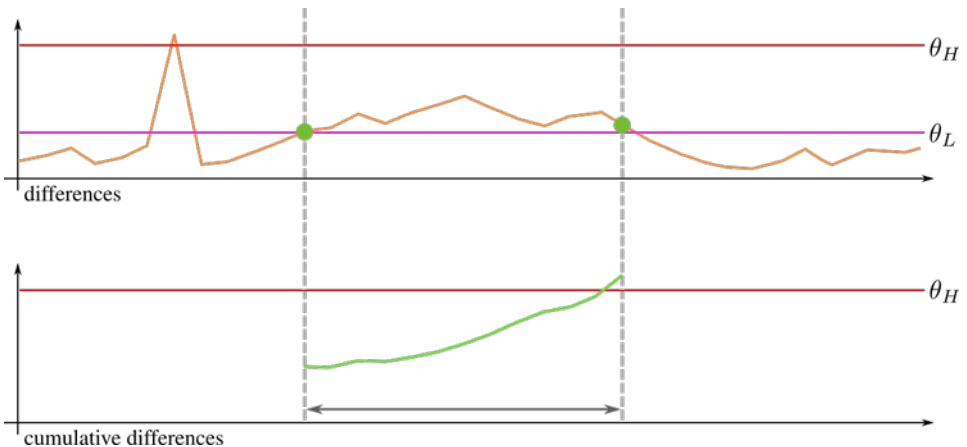
Partial changes

- 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



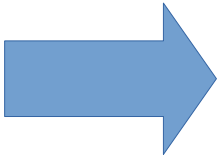
Detecting fades

- 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

Image morphing



How to deform image A to image B?

Control points

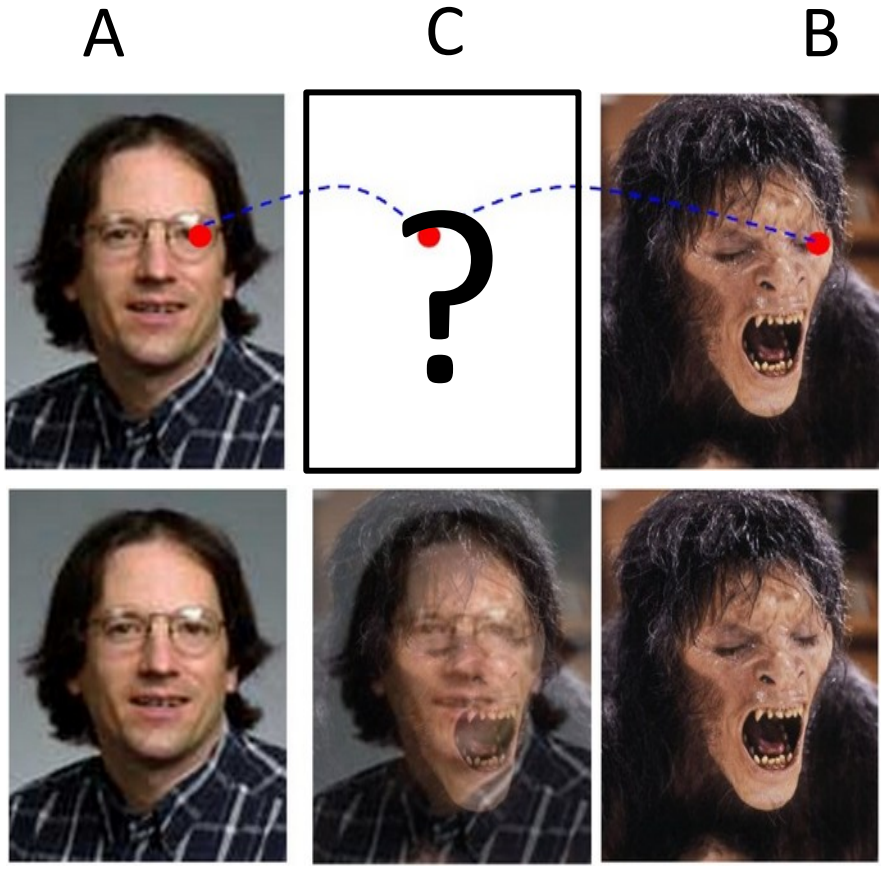
Control points mark matching pixels in both images

Image morphing

- How to compute intermediate image C
- Naive approach - weighted sum of pixels

$$C_t = \alpha_t A + (1 - \alpha_t) B$$
$$0 \leq \alpha_t \leq 1$$

- Not realistic - not combining semantic parts



Deformation field approximation

- Determining entire field is time-consuming
- Locally linear transformation
- Correspondences – control points
- Delaunay triangulation, interpolation

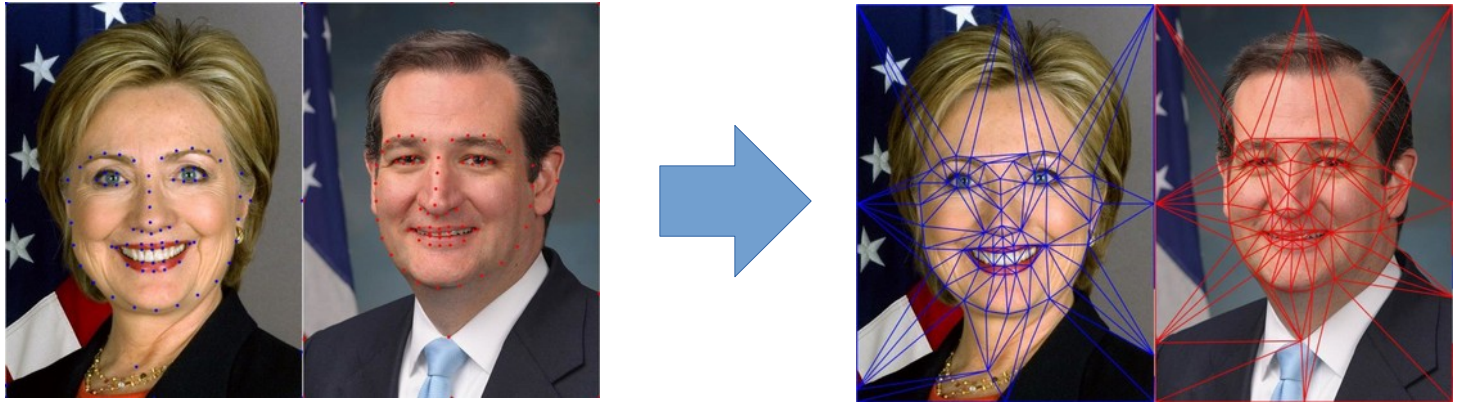
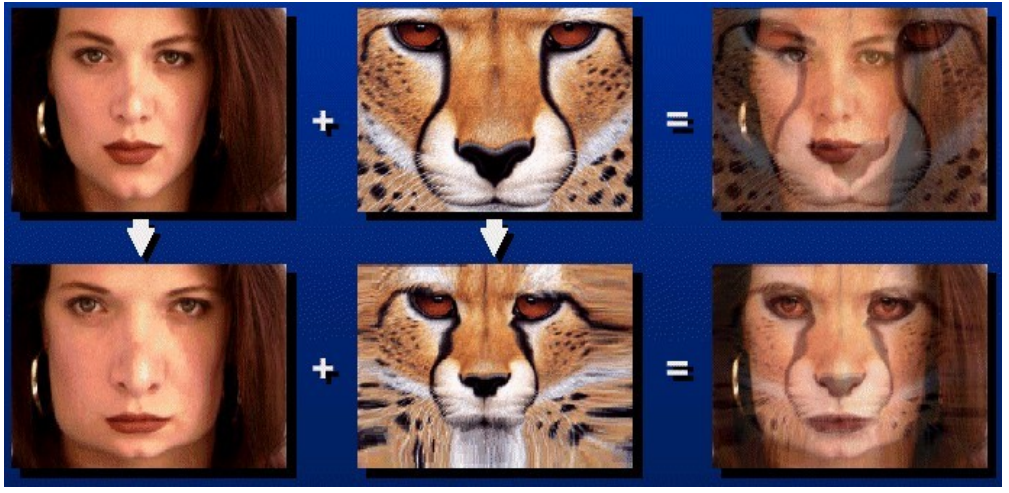


Image morphing overview

- For each image Ct compute …
  - Interpolated position of control points
  - Two transformations: dA = A - Ct and dB = B - Ct
  - Blend colors of interpolated images

$$\mathbf{x}_i^C = \alpha_t \mathbf{x}_i^A + (1 - \alpha_t)\mathbf{x}_i^B$$

$$C_t = \alpha_t dA + (1 - \alpha_t)dB$$

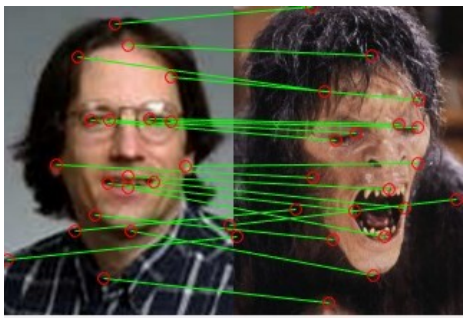


Naive

Correct

Morphing example

Input images



Control points

Warped images

Blended image

One more morphing example