

# Video Texture

---



© A.A. Efros

15-463: Computational Photography  
Alexei Efros, CMU, Fall 2006

# Weather Forecasting for Dummies™

---

Let's predict weather:

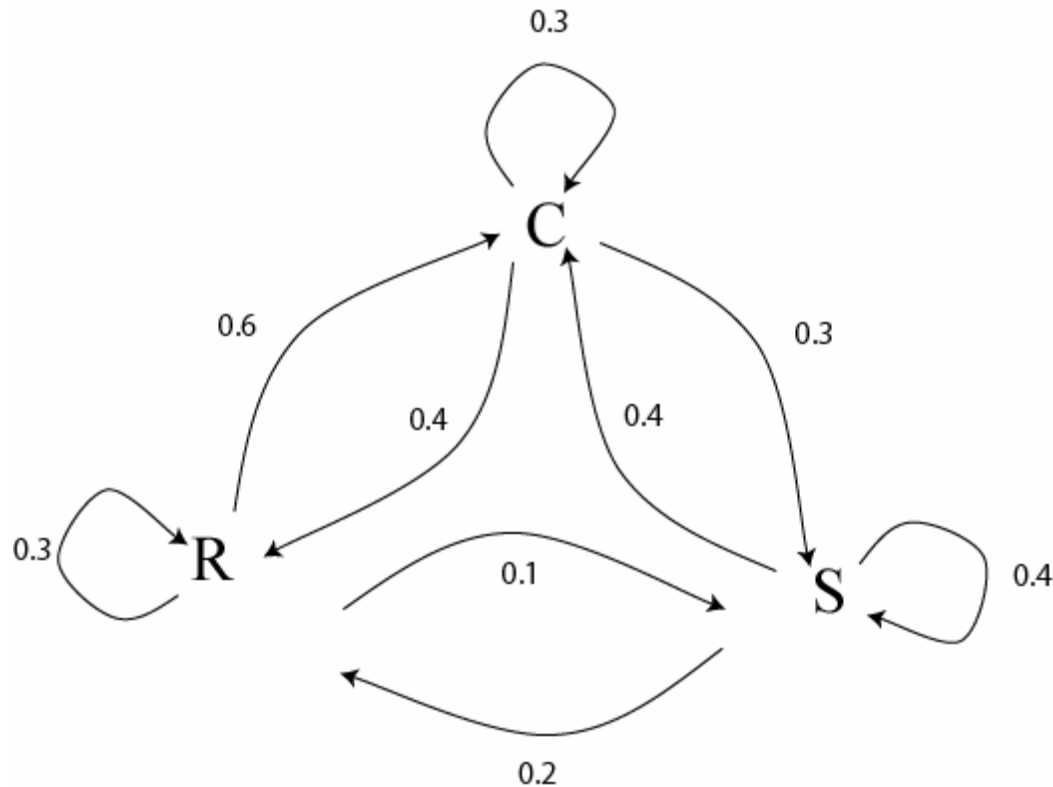
- Given today's weather only, we want to know tomorrow's
- Suppose weather can only be {Sunny, Cloudy, Raining}

The “Weather Channel” algorithm:

- Over a long period of time, record:
  - How often S followed by R
  - How often S followed by S
  - Etc.
- Compute percentages for each state:
  - $P(R|S)$ ,  $P(S|S)$ , etc.
- Predict the state with highest probability!
- It's a Markov Chain

# Markov Chain

---



$$\begin{pmatrix} 0.3 & 0.6 & 0.1 \\ 0.4 & 0.3 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{pmatrix}$$

What if we know today and yestarday's weather?

# Text Synthesis

---

[Shannon, '48] proposed a way to generate English-looking text using N-grams:

- Assume a generalized Markov model
- Use a large text to compute prob. distributions of each letter given N-1 previous letters
- Starting from a seed repeatedly sample this Markov chain to generate new letters
- Also works for whole words

**WE NEED TO EAT CAKE**

# Mark V. Shaney (Bell Labs)

---

## Results (using alt.singles corpus):

- *“As I've commented before, really relating to someone involves standing next to impossible.”*
- *“One morning I shot an elephant in my arms and kissed him.”*
- *“I spent an interesting evening recently with a grain of salt”*

# Video Textures

Arno Schödl

Richard Szeliski

David Salesin

Irfan Essa

Microsoft Research, Georgia Tech

# Still photos



# Video clips





# Video textures



# Problem statement



video clip



video texture

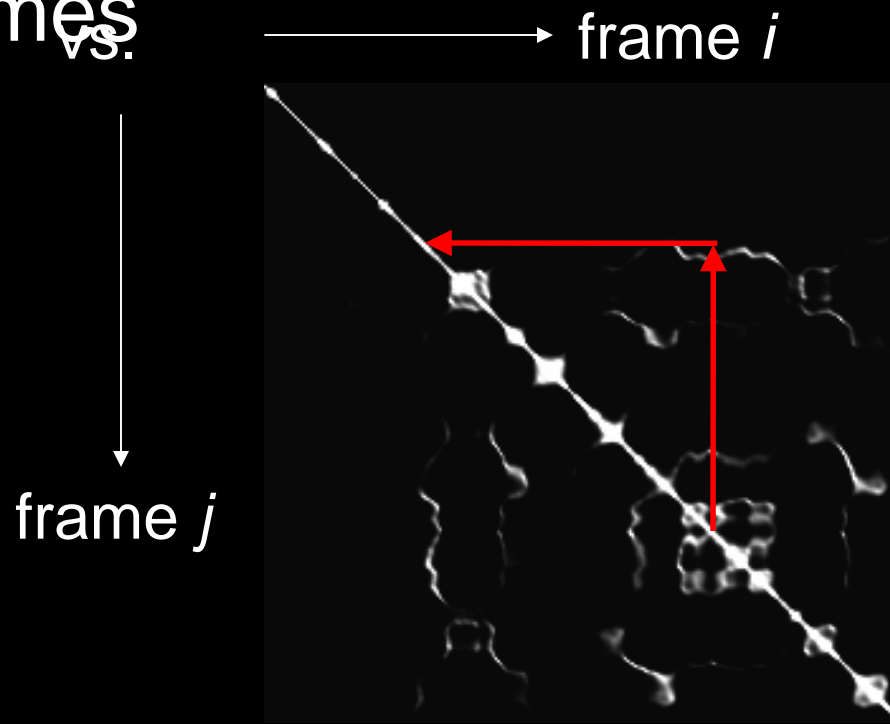
# Our approach



- How do we find good transitions?

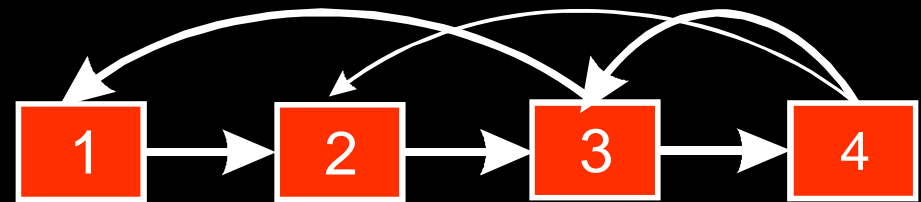
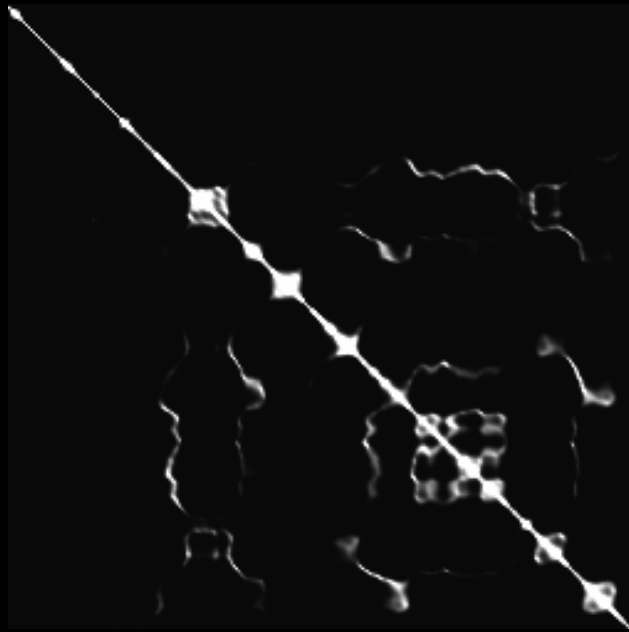
# Finding good transitions

- Compute  $L_2$  distance  $D_{i,j}$  between all frames



Similar frames make good transitions

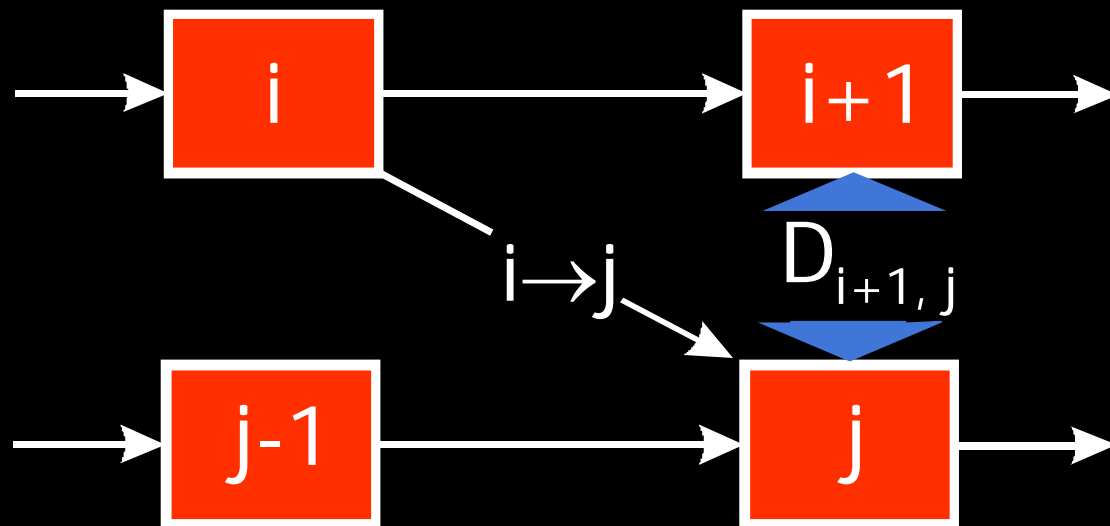
# Markov chain representation



Similar frames make good transitions

# Transition costs

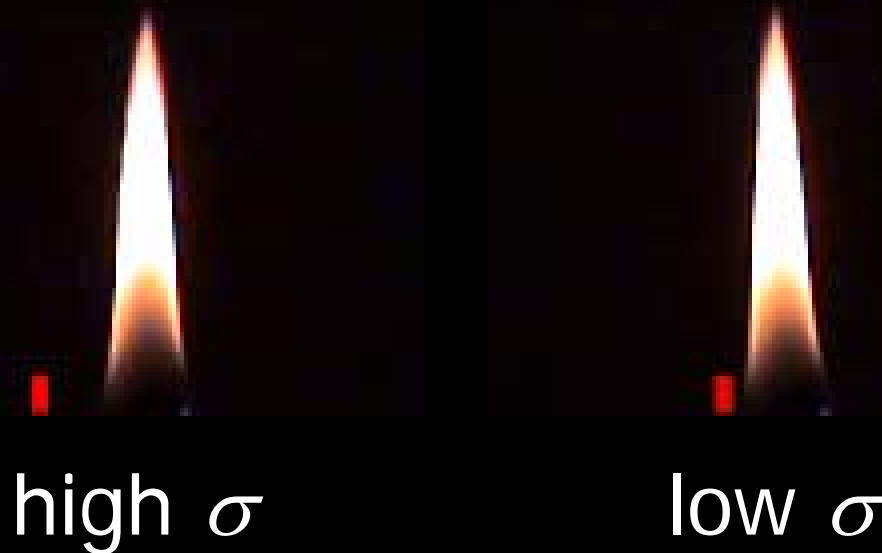
- Transition from  $i$  to  $j$  if successor of  $i$  is similar to  $j$ 
  - Cost function:  $C_{i \rightarrow j} = D_{i+1, j}$



# Transition probabilities

- Probability for transition  $P_{i \rightarrow j}$  inversely related to cost:

- $P_{i \rightarrow j} \sim \exp ( - C_{i \rightarrow j} / \sigma^2 )$

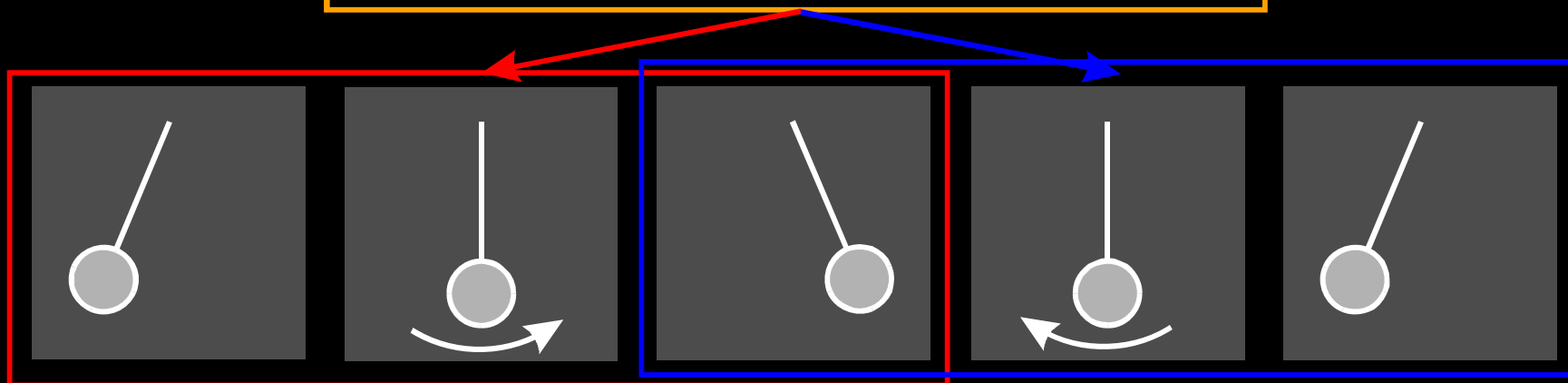
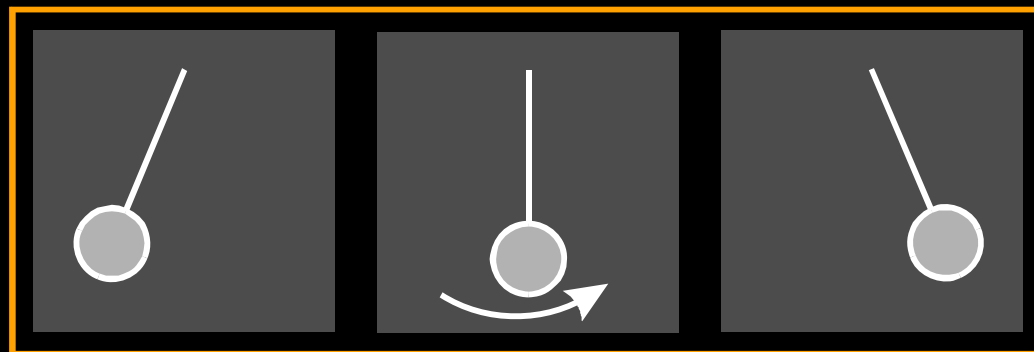


# Preserving dynamics





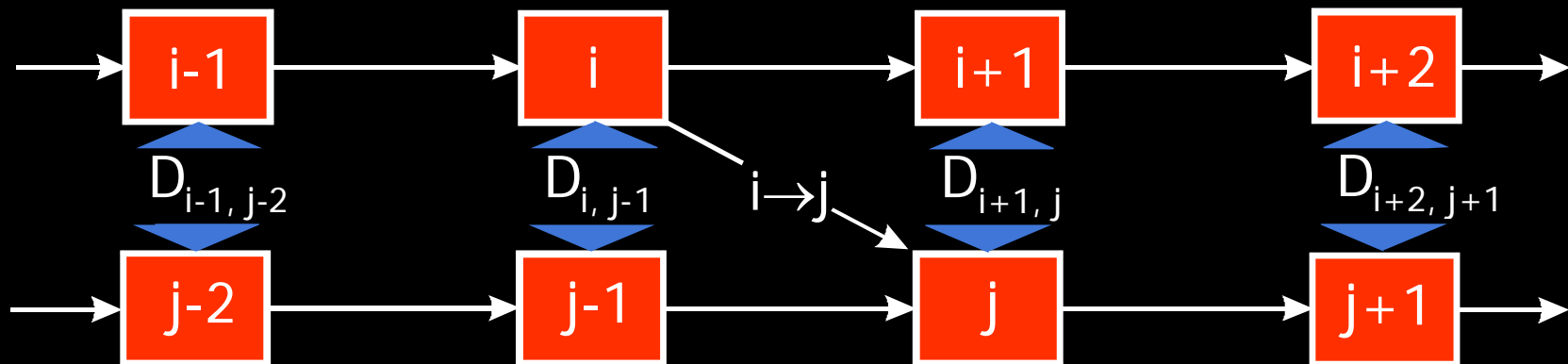
# Preserving dynamics



# Preserving dynamics

- Cost for transition  $i \rightarrow j$

- $$C_{i \rightarrow j} = \sum_{k=-N}^{N-1} w_k D_{i+k+1, j+k}$$



# Preserving dynamics – effect

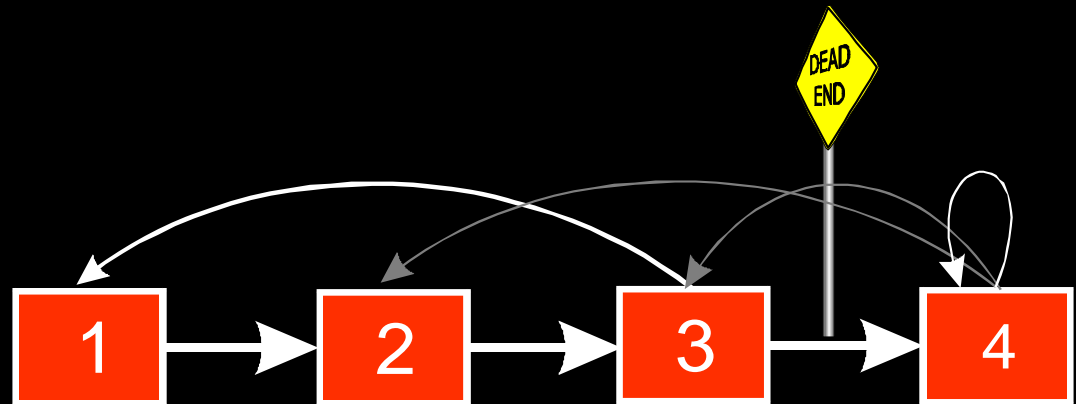
- Cost for transition  $i \rightarrow j$

- $$C_{i \rightarrow j} = \sum_{k=-N}^{N-1} w_k D_{i+k+1, j+k}$$



# Dead ends

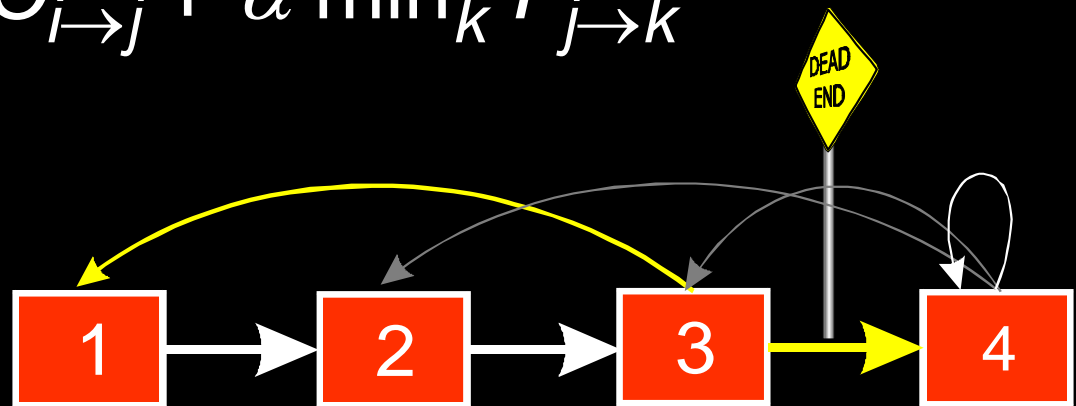
- No good transition at the end of sequence



# Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

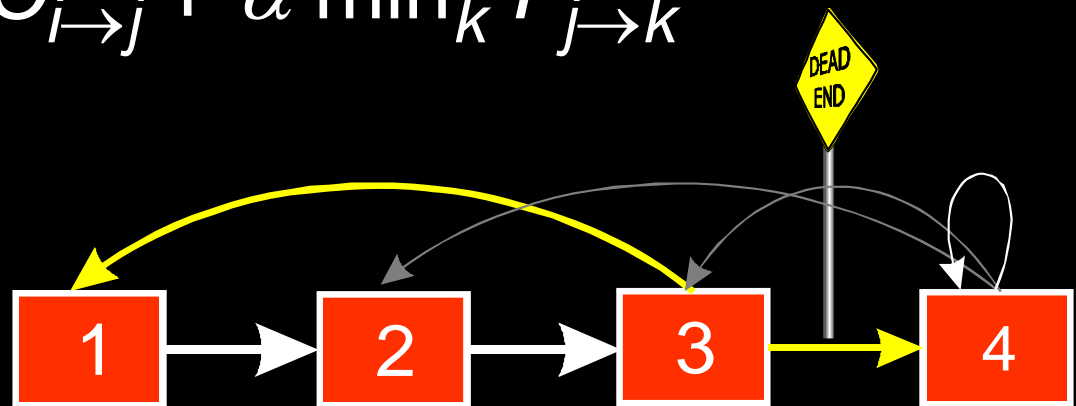
- $$F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k}$$



# Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

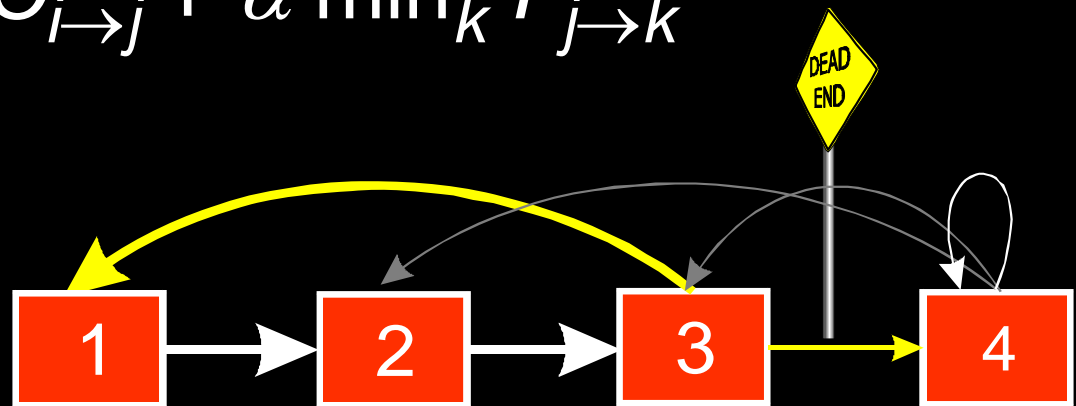
- $$F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k}$$



# Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

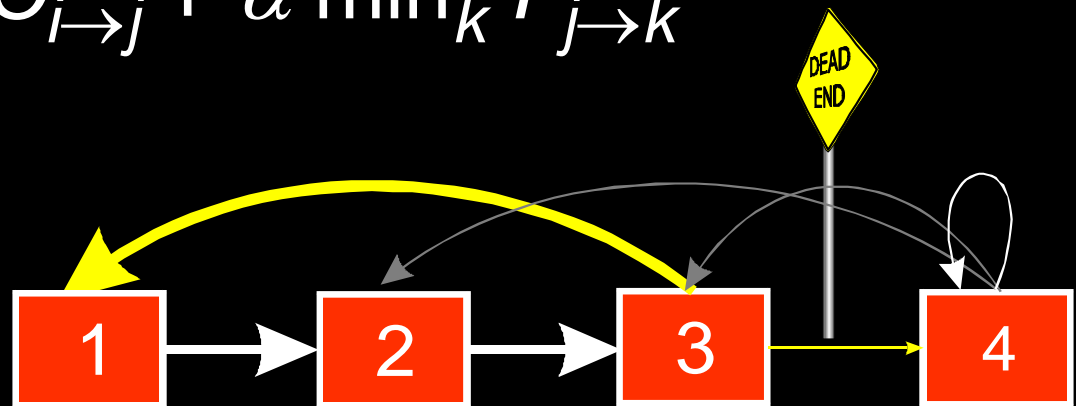
- $$F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k}$$



# Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

- $$F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k}$$



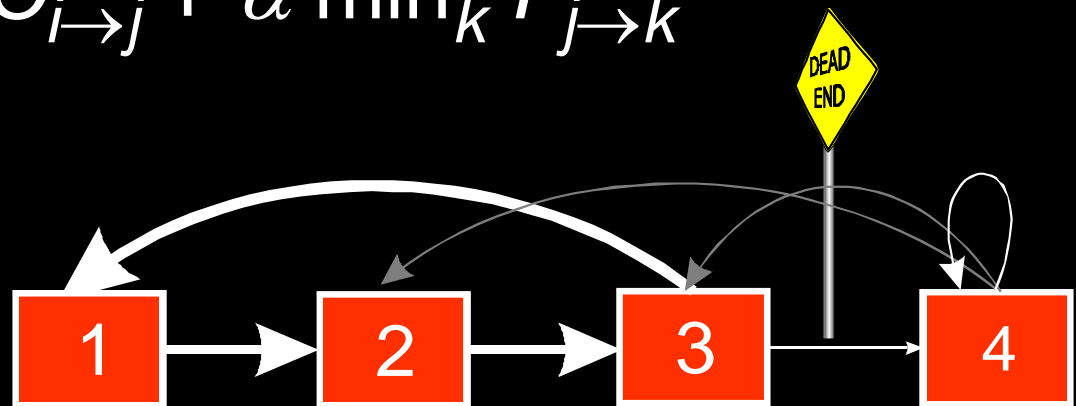


# Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

- $$F_{i \rightarrow j} = C_{i \rightarrow j} + \alpha \min_k F_{j \rightarrow k}$$

- Q-learning



# Future cost – effect



# Finding good loops

- Alternative to random transitions
- Precompute set of loops up front



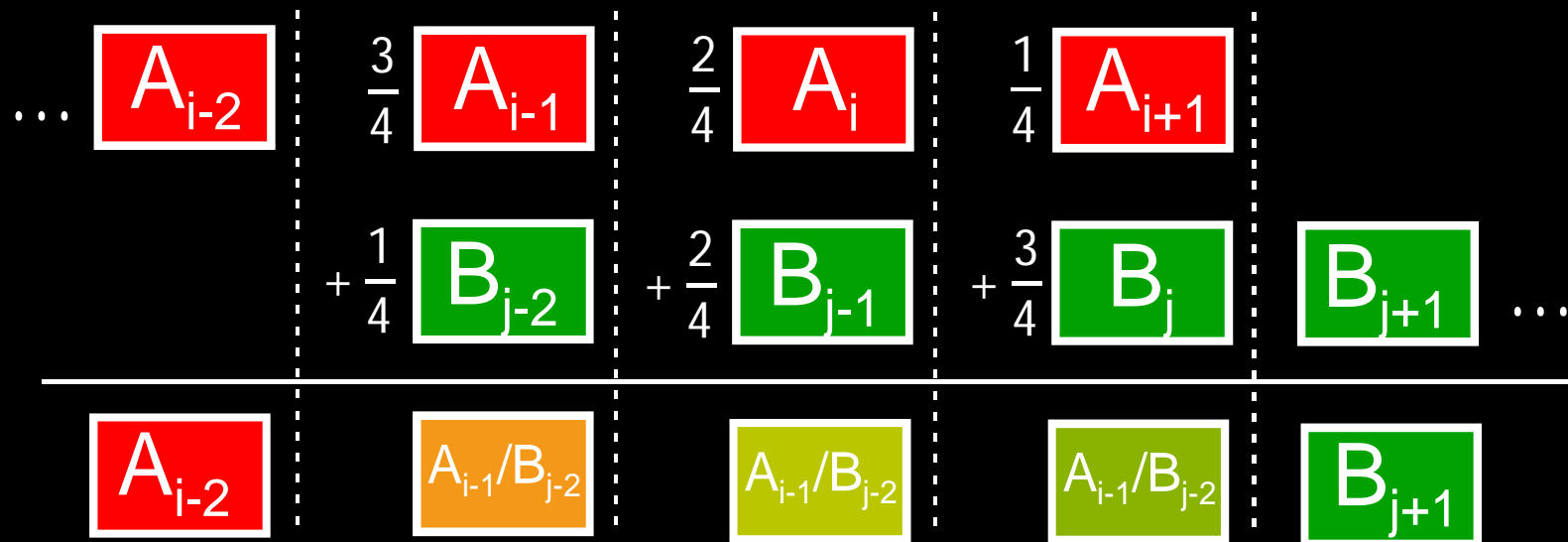
# Visual discontinuities

- Problem: Visible “Jumps”



# Crossfading

- Solution: Crossfade from one sequence to the other.



# Morphing

- Interpolation task:

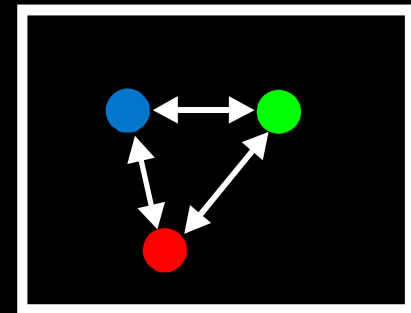
$$\frac{2}{5} \text{ [A] } + \frac{2}{5} \text{ [B] } + \frac{1}{5} \text{ [C]}$$

# Morphing

- Interpolation task:

$$\frac{2}{5} \boxed{\text{A}} + \frac{2}{5} \boxed{\text{B}} + \frac{1}{5} \boxed{\text{C}}$$

- Compute correspondence between pixels of all frames

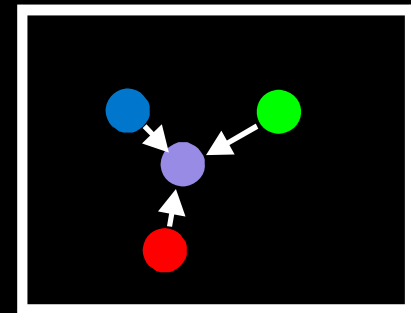


# Morphing

- Interpolation task:

$$\frac{2}{5} \boxed{\text{A}} + \frac{2}{5} \boxed{\text{B}} + \frac{1}{5} \boxed{\text{C}}$$

- Compute correspondence between pixels of all frames
- Interpolate pixel position and color in morphed frame
- based on [Shum 2000]





# Results – crossfading/morphing



# Results – crossfading/morphing



Jump Cut

Crossfade

Morph

# Crossfading



# Frequent jump & crossfading



# Video portrait



- Useful for web pages

# Region-based analysis

- Divide video up into regions



- Generate a video texture for each region

# Automatic region analysis



# Video-based animation

- Like sprites  
computer games
- Extract sprites  
from real video
- Interactively control  
desired motion



©1985 Nintendo of America Inc.



# Video sprite extraction

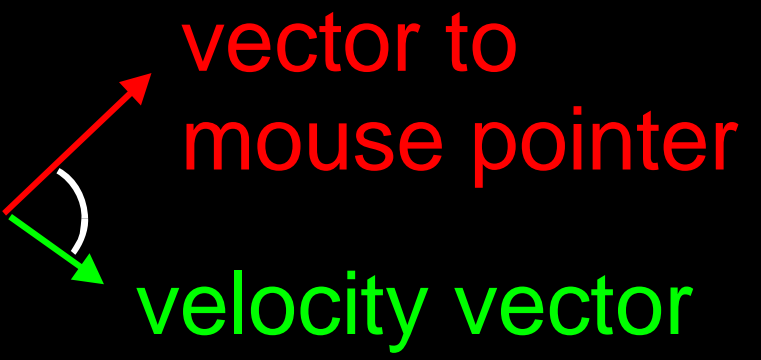


blue screen matting  
and velocity estimation



# Video sprite control

- Augmented transition cost:

$$C_{i \rightarrow j}^{\text{Animation}} = \alpha \underbrace{C_{i \rightarrow j}}_{\text{Similarity term}} + \beta \underbrace{\text{angle}}_{\text{Control term}}$$


vector to mouse pointer

velocity vector

# Interactive fish



# Lord of the Flies



# Summary

- Video clips → video textures
  - define Markov process
  - preserve dynamics
  - avoid dead-ends
  - disguise visual discontinuities

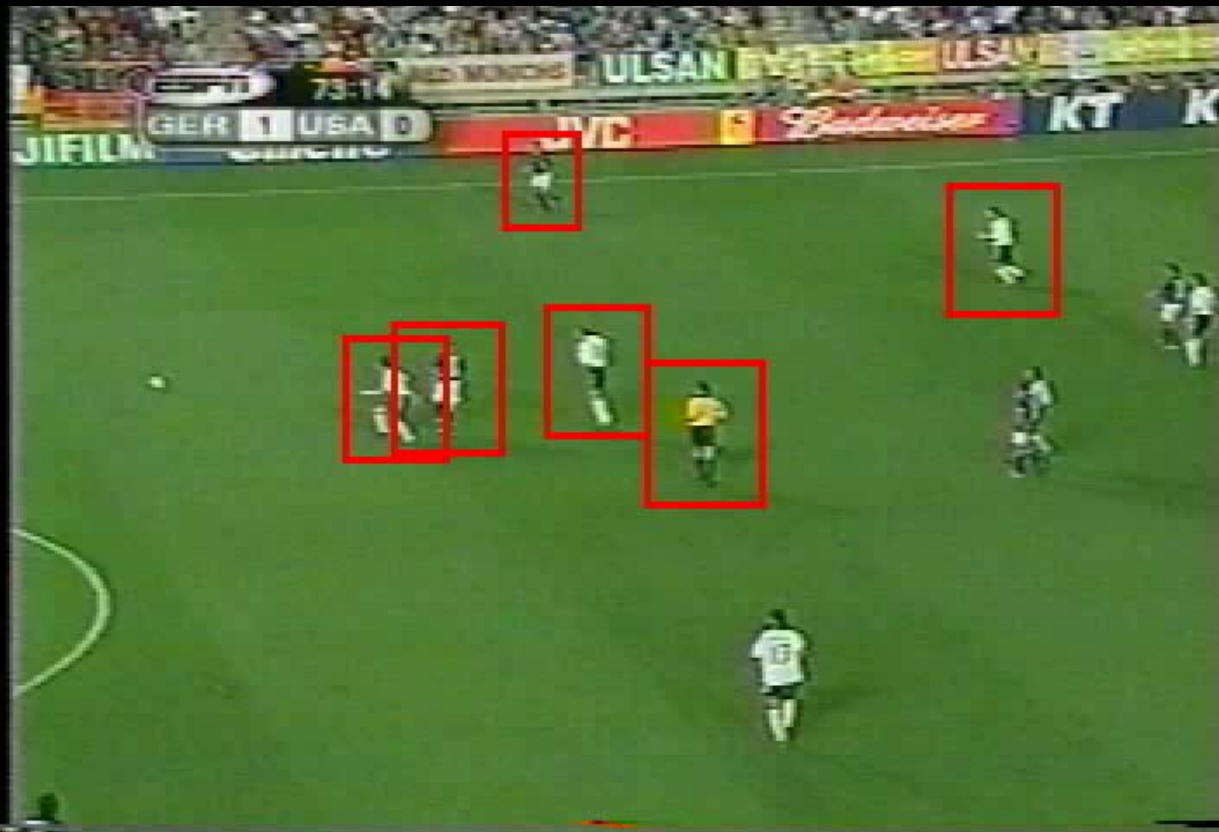


# Motion Analysis & Synthesis [Efros '03]



- What are they doing?
  - Activity recognition, surveillance, anti-terrorism
- Can we do the same?
  - Motion retargeting, movies, video games, etc.

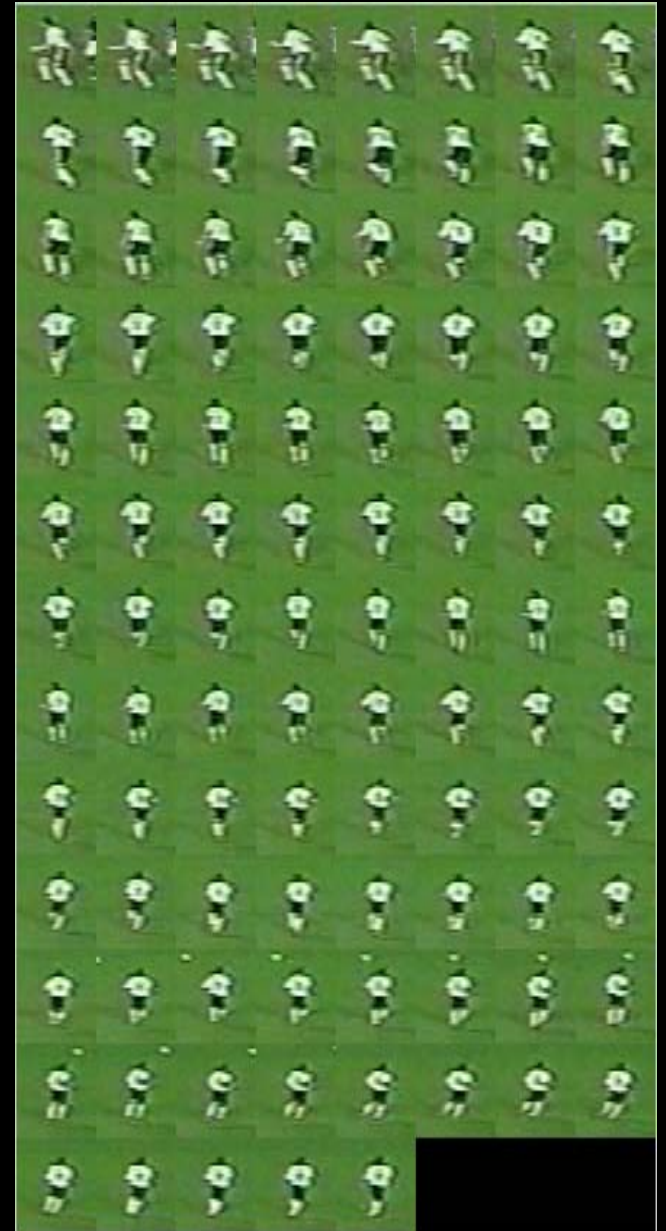
# Gathering action data



- Low resolution, noisy data
- Moving camera
- Occlusions

# Figure-centric Representation

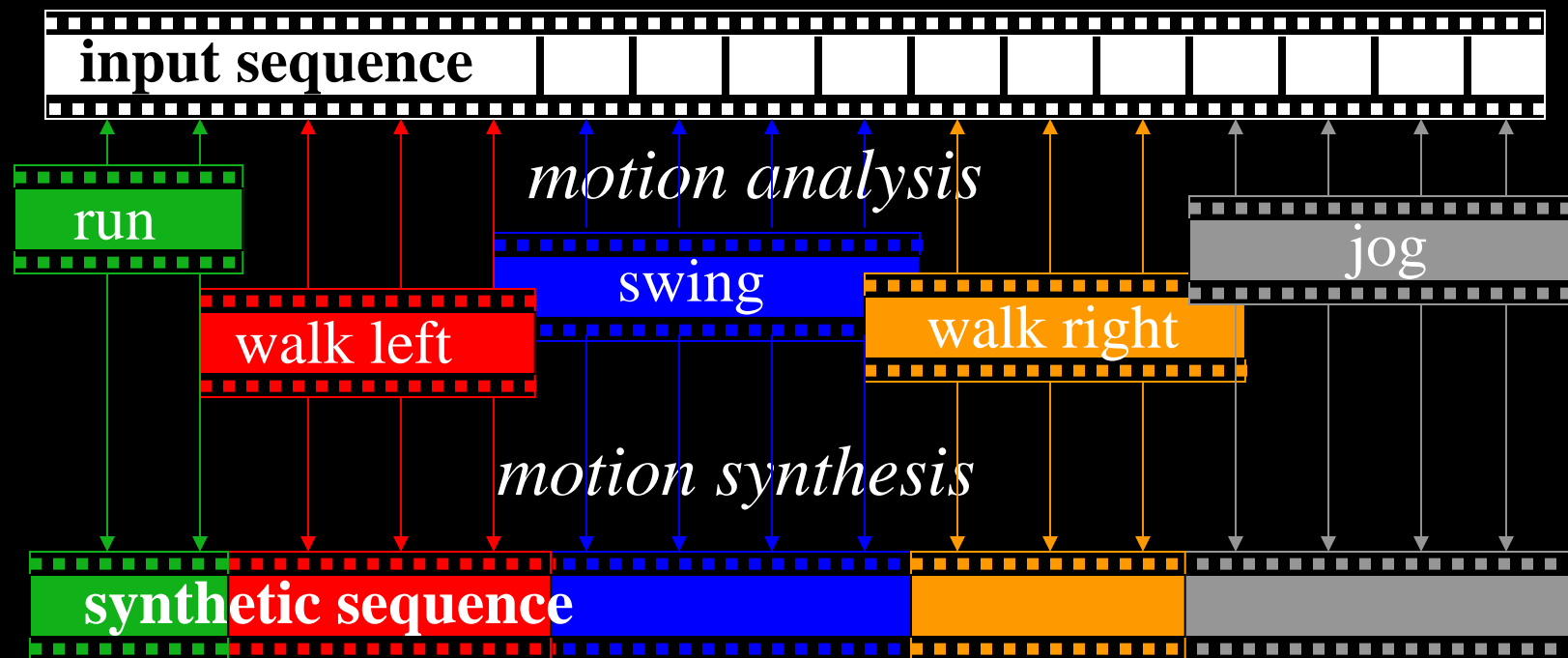
- Stabilized spatio-temporal volume
  - No translation information
  - All motion caused by person's limbs
    - Good news: indifferent to camera motion
    - Bad news: hard!
- Good test to see if actions, not just translation, are being captured





# Remembrance of Things Past

- “Explain” novel motion sequence with bits and pieces of previously seen video clips



Challenge: how to compare motions?

# How to describe motion?

- Appearance
  - Not preserved across different clothing
- Gradients (spatial, temporal)
  - same (e.g. contrast reversal)
- Edges
  - Too unreliable
- Optical flow
  - Explicitly encodes motion
  - Least affected by appearance
  - ...but too noisy

# Motion Descriptor

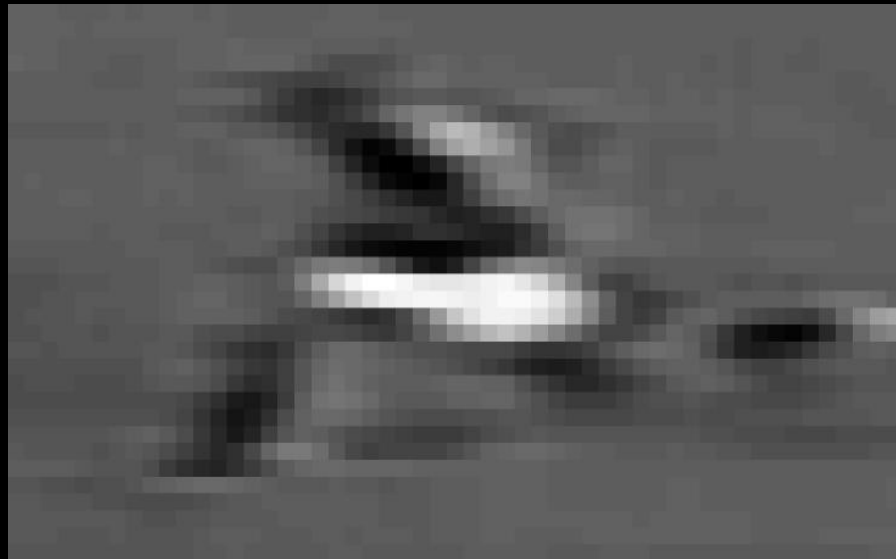
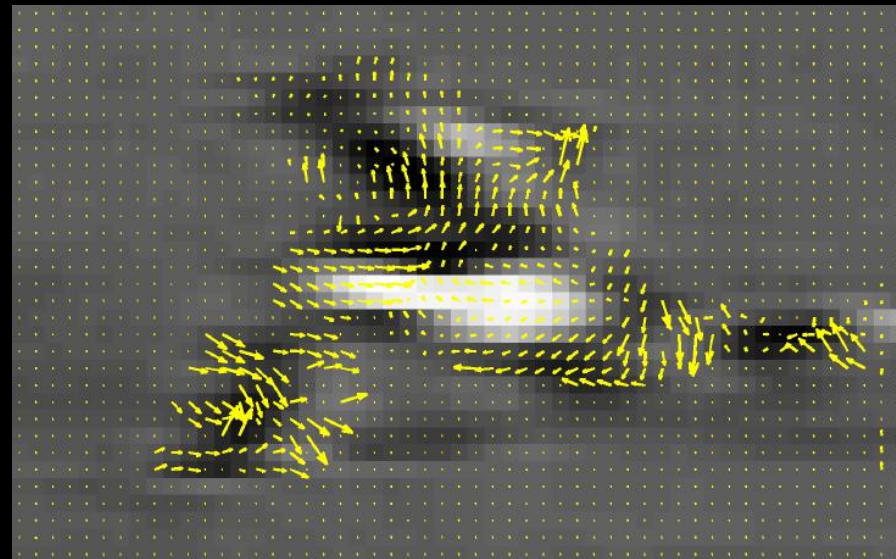
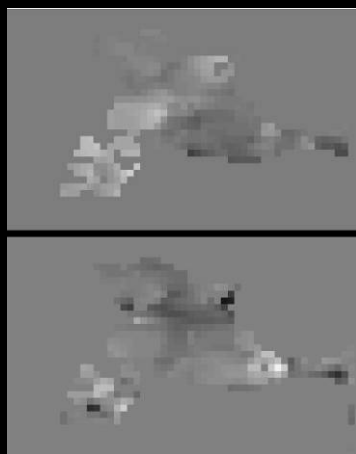


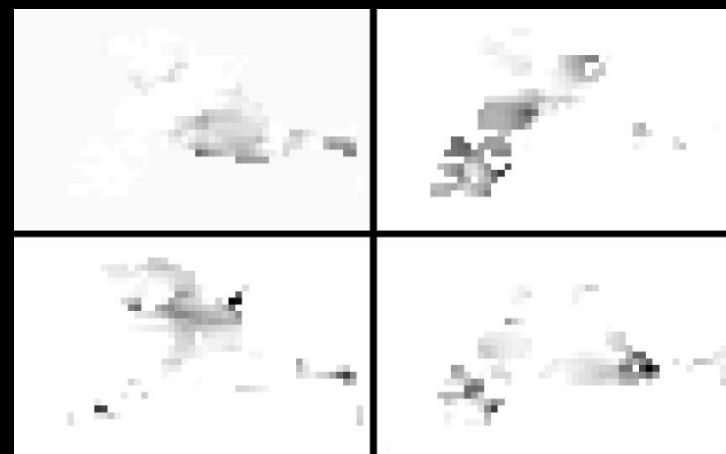
Image frame



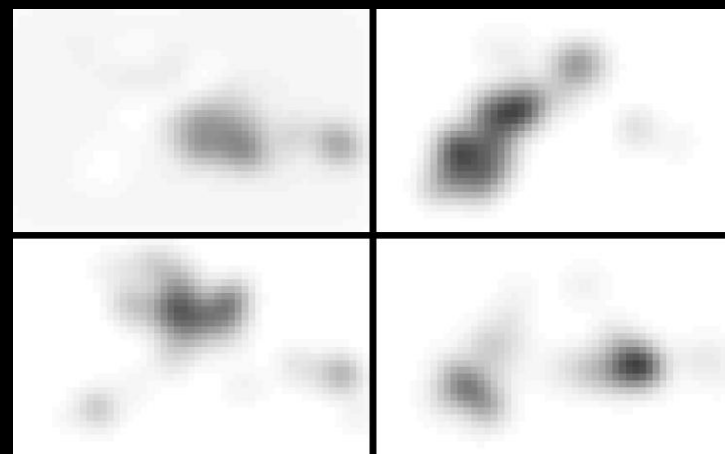
Optical flow  $F_{x,y}$



$F_x, F_y$

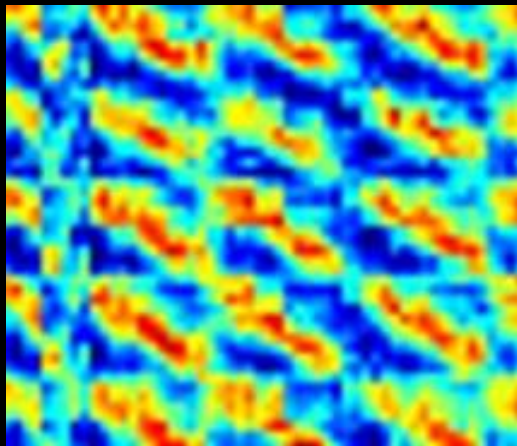
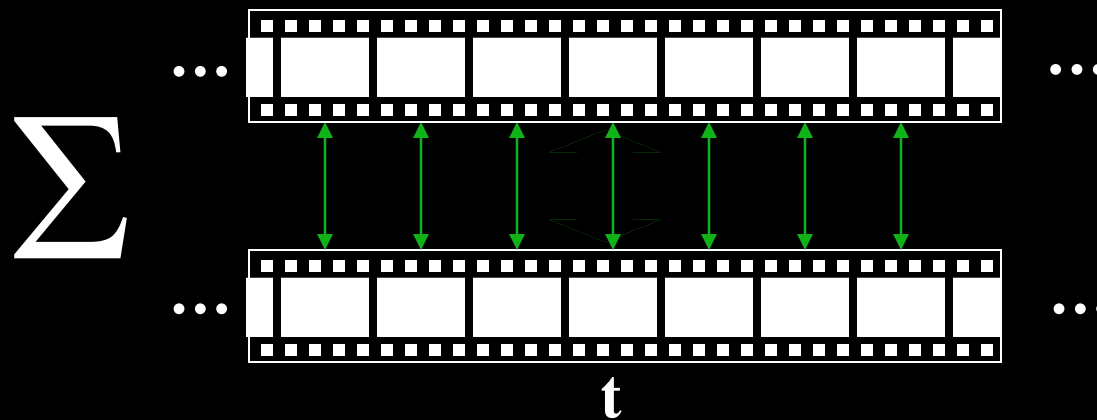


$F_x^-, F_x^+, F_y^-, F_y^+$

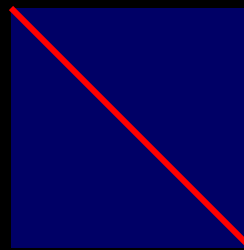


blurred  $F_x^-, F_x^+, F_y^-, F_y^+$

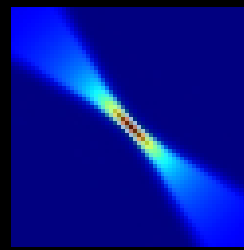
# Comparing motion descriptors



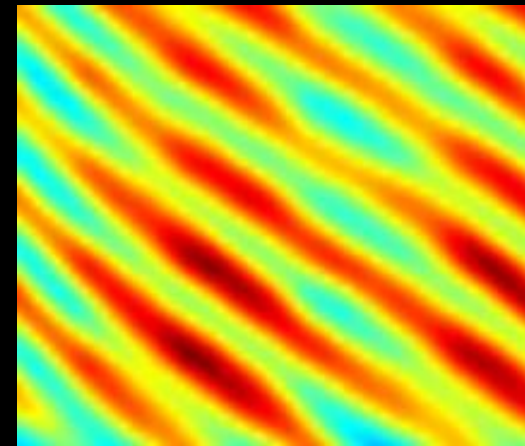
frame-to-frame  
similarity matrix



I matrix

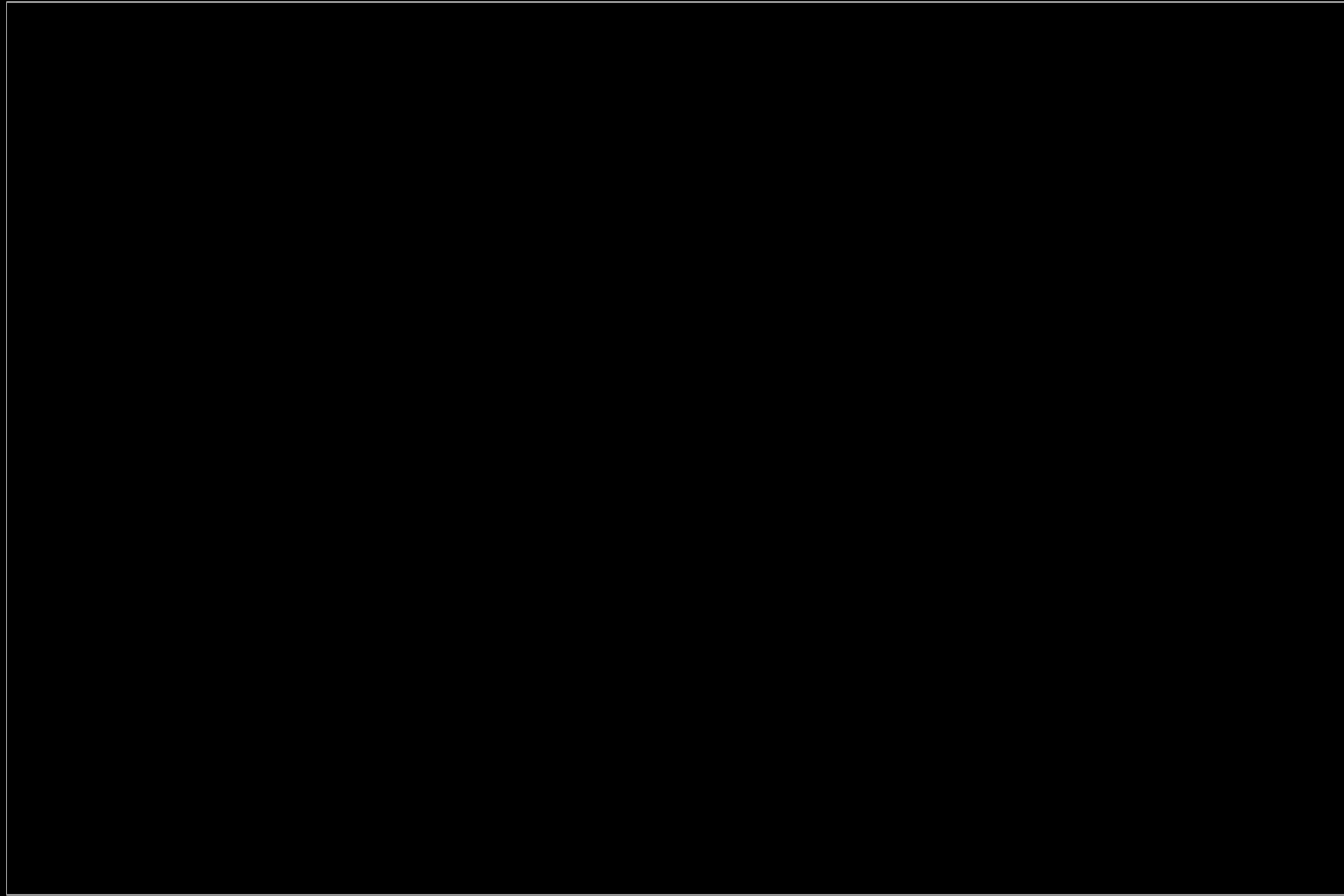


blurry I



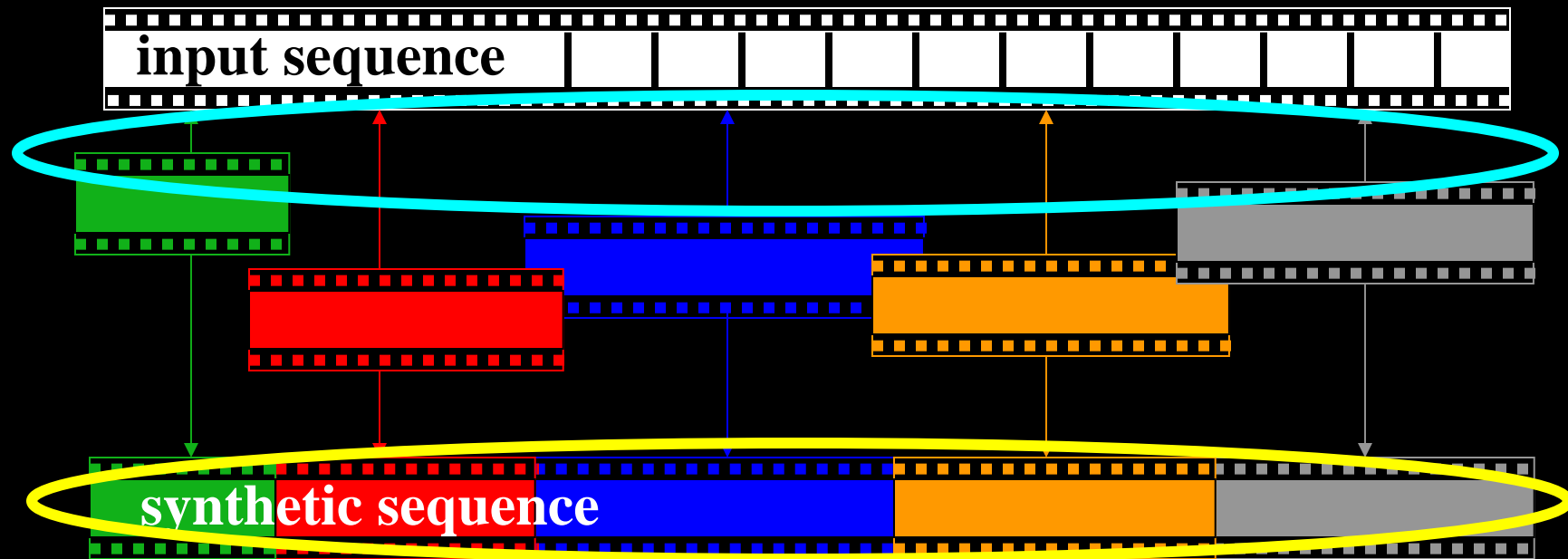
motion-to-motion  
similarity matrix

# Recognizing Tennis



- Red bars show classification results

# “Do as I Do” Motion Synthesis



- Matching two things:
  - Motion similarity across sequences
  - Appearance similarity within sequence
- Dynamic Programming

# Smoothness for Synthesis

- $W_{act}$  is similarity between source and target frames
- $W_{app}$  is appearance similarity within target frames
- For every source frame  $i$ , find best target frame  $\pi_i$
- by maximizing following cost function:

$$\sum_{i=1}^n \alpha_{act} W_{act}(i, \pi_i) + \sum_{i=2}^n \alpha_{app} W_{app}(\pi_i, \pi_{i-1} + 1)$$

- Optimize using dynamic programming

# “Do as I Do”

Source Motion

Source Appearance

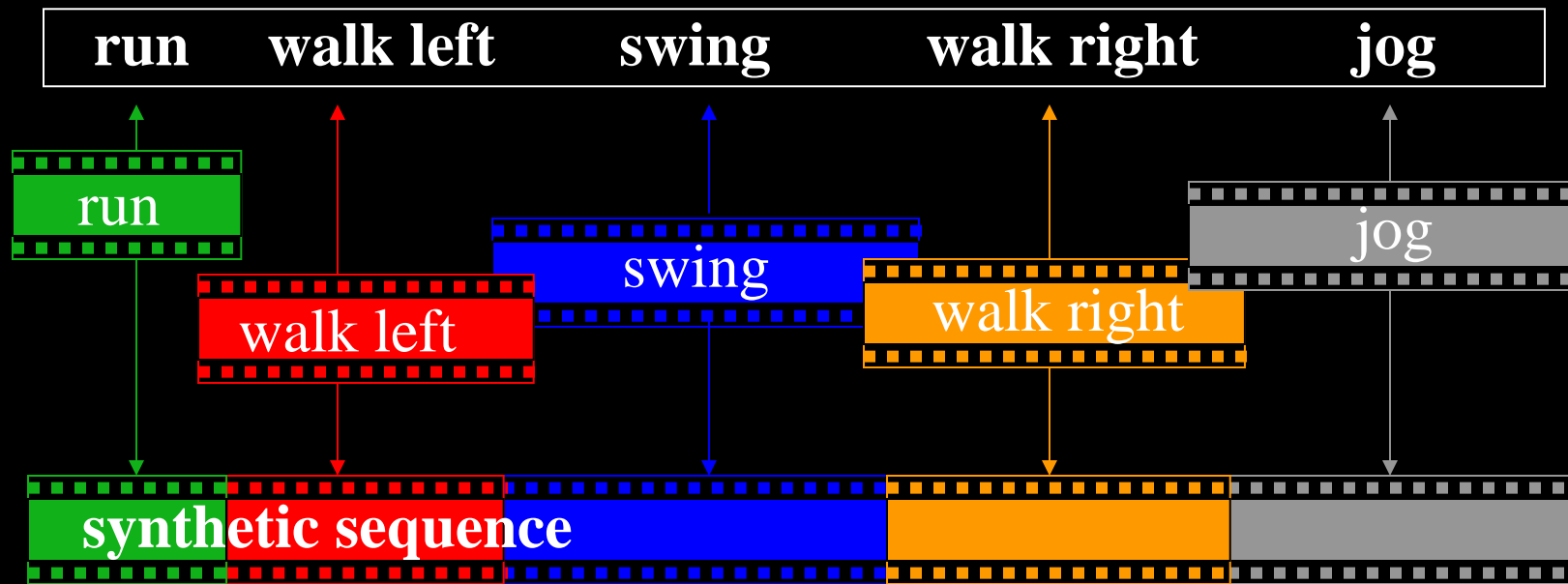


3400 Frames

Result



# “Do as I Say” Synthesis



- Synthesize given action labels
  - e.g. video game control

# “Do as I Say”



- Red box shows when constraint is applied

# Application: Motion Retargeting

- Rendering new character into existing footage
- Algorithm
  - Track original character
  - Find matches from new character
  - Erase original character
  - Render in new character
    - Need to worry about occlusions

# Actor Replacement

SHOW VIDEO