Computational Photography and Capture:

Video Texture Synthesis

Gabriel Brostow & Tim Weyrich

TA: Frederic Besse

Markov Chains

- Probability of going from state *i* to state *j* in *n* time steps:

$$p_{ij}^{(n)} = \Pr(X_n = j \mid X_0 = i)$$

and the single-step transition as:

$$p_{ij} = \Pr(X_1 = j \mid X_0 = i)$$

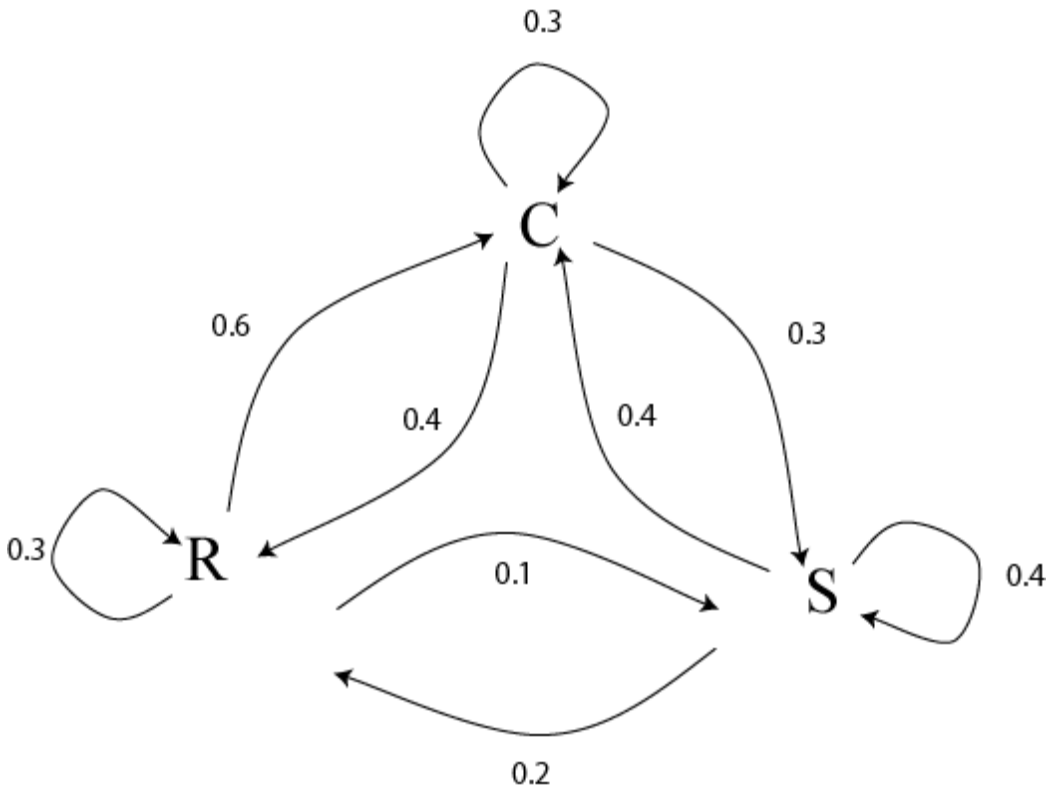
The *n*-step transition satisfies the Chapman-Kolmogorov equation, that for any $0<k<n$:

$$p_{ij}^{(n)} = \sum_{r \in S} p_{ir}^{(k)} p_{rj}^{(n-k)}$$

Markov Chains

- Regular Markov chain: class of Markov chains where the starting state of the chain has little or no impact on the p(X) after many steps.

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 yesterday'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

(1D predecessor to Graphcut Textures)

Arno Schödl
Richard Szeliski
David Salesin
Irfan Essa

Microsoft Research, Georgia Tech

Link to local version | Gondry Example

Still photos

Video clips

Video textures

Problem statement

video clip

video texture

Video Textures Approach

- How do we find good transitions?

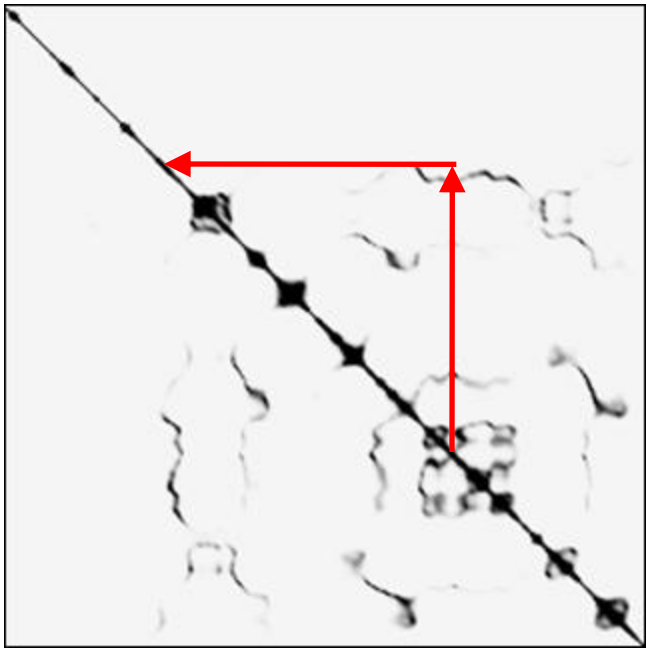
Finding good transitions

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

vs.

frame *i*

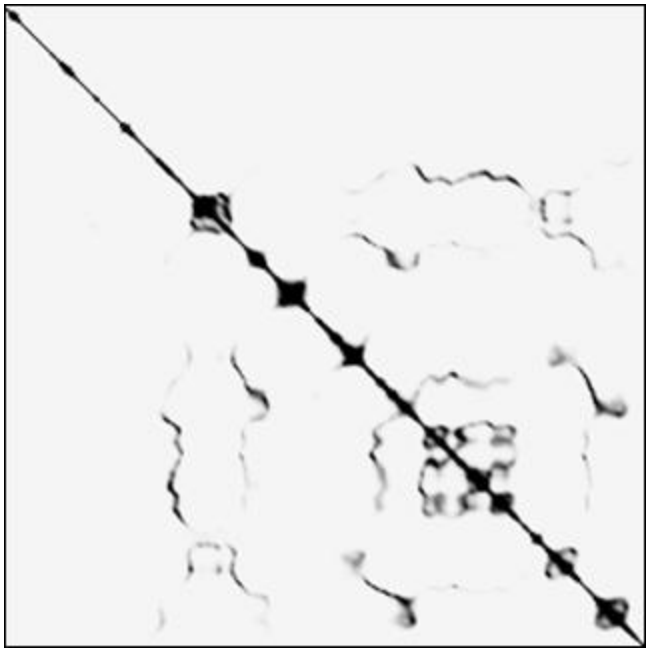
frame *j*

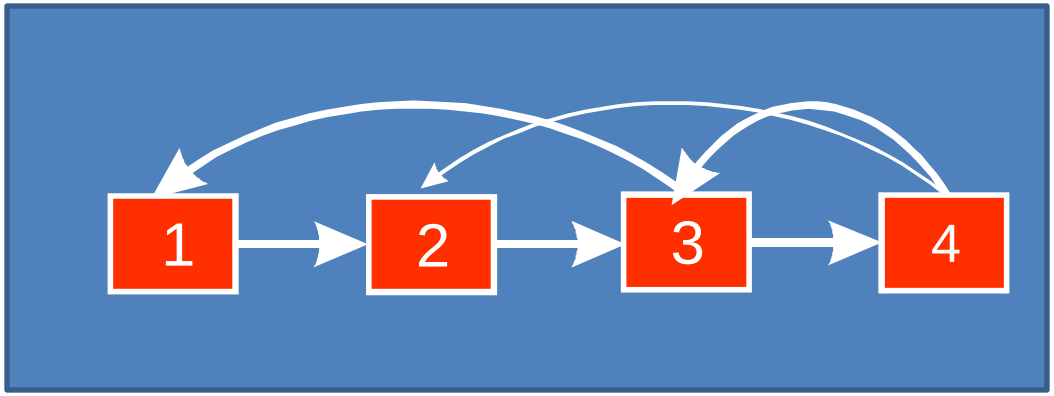


$$D_{ij} = \|\mathcal{I}_i - \mathcal{I}_j\|_2$$

Similar frames make good transitions

Markov chain representation

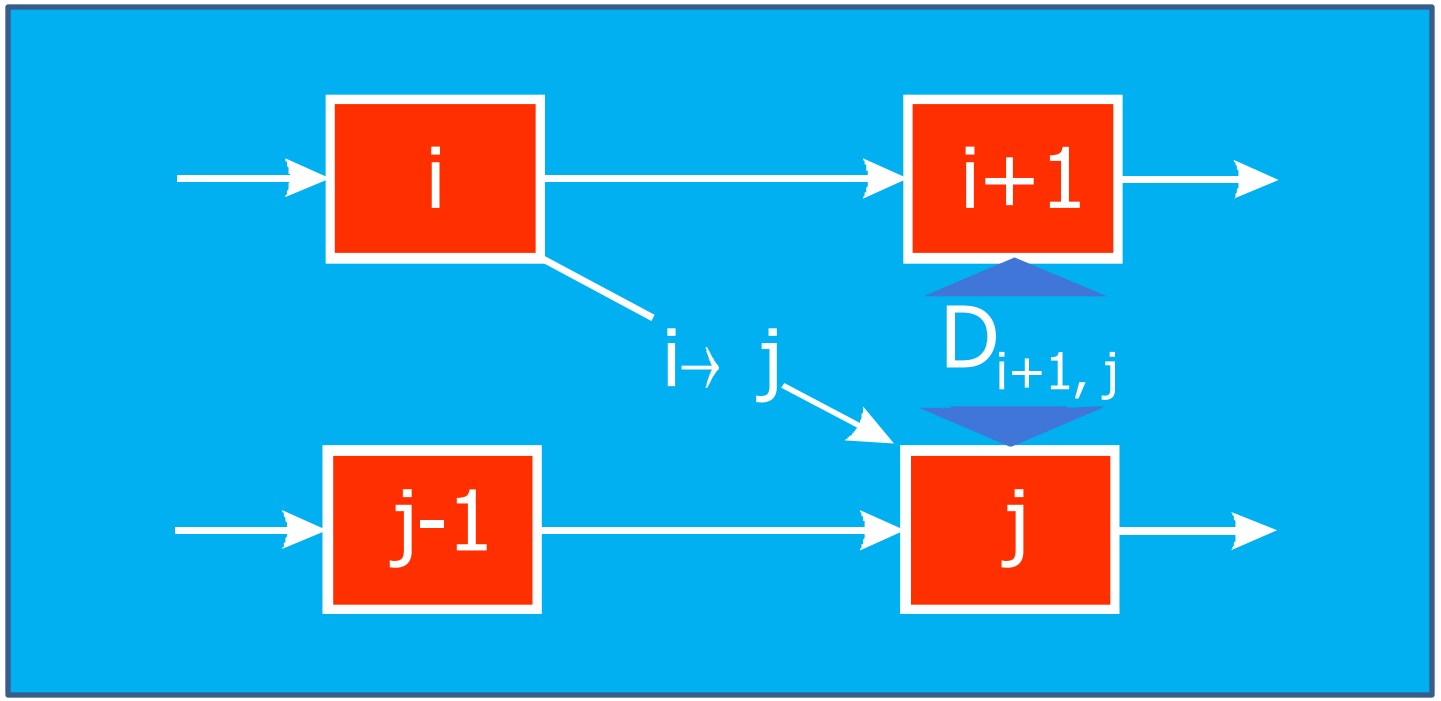




Similar frames make good transitions

Transition costs

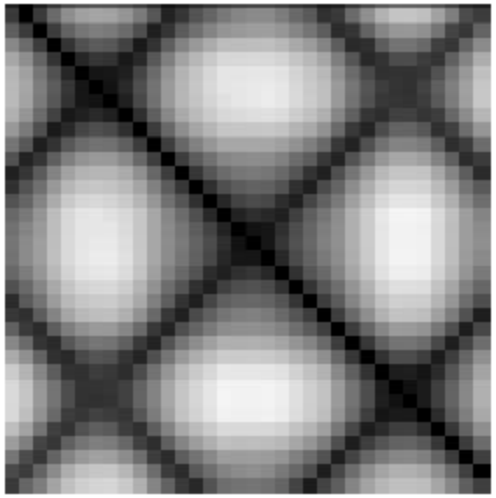
- Transition from i to j if successor of i is similar to j
  - Cost function: $C_{i \rightarrow j} = D_{i+1, j}$
-



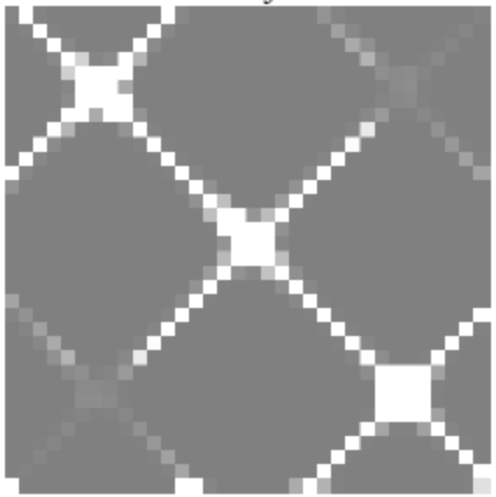
Transition probabilities

- Probability for transition $P_{i \to j}$ inversely related to cost:
  - $P_{i \to j} \sim \exp\left(- C_{i \to j} / \sigma^2 \right)$

high σ low σ



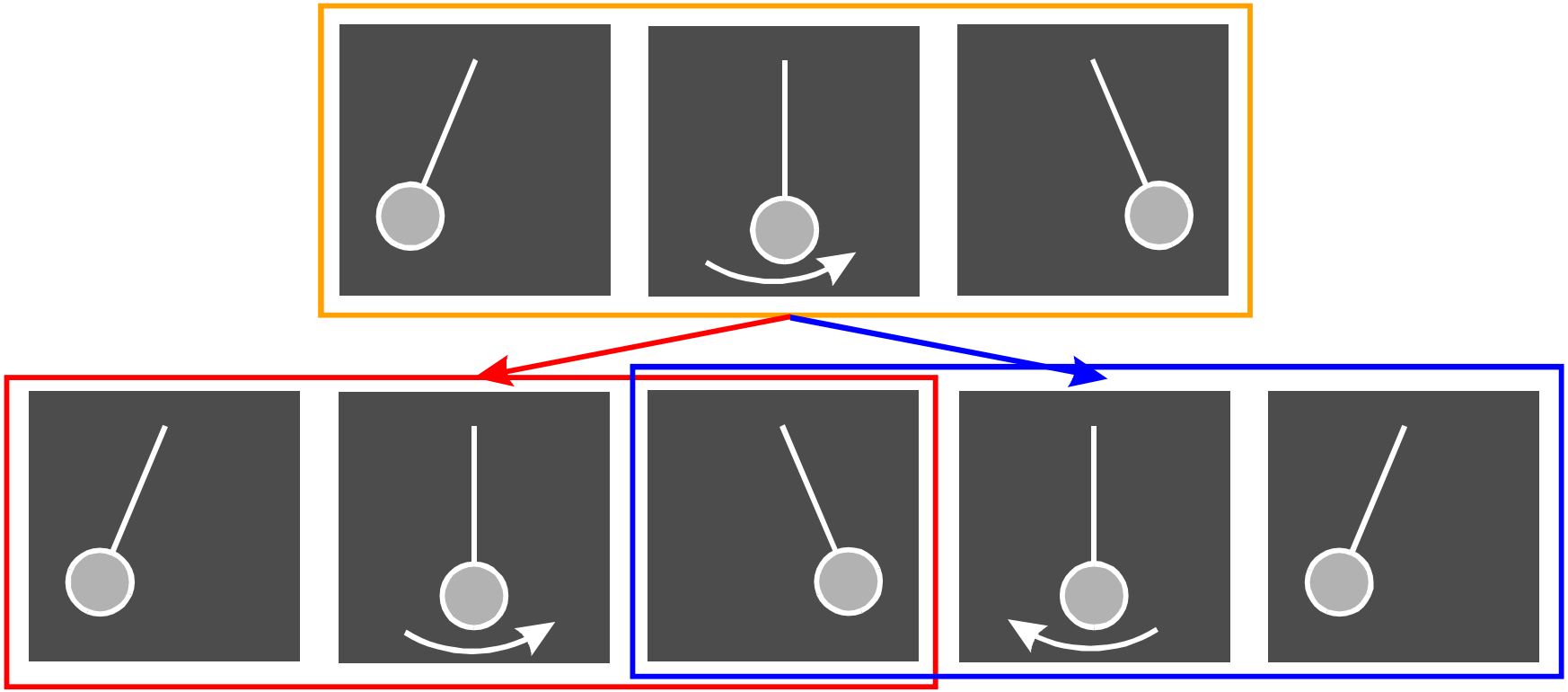
D_{ij}



P_{ij}

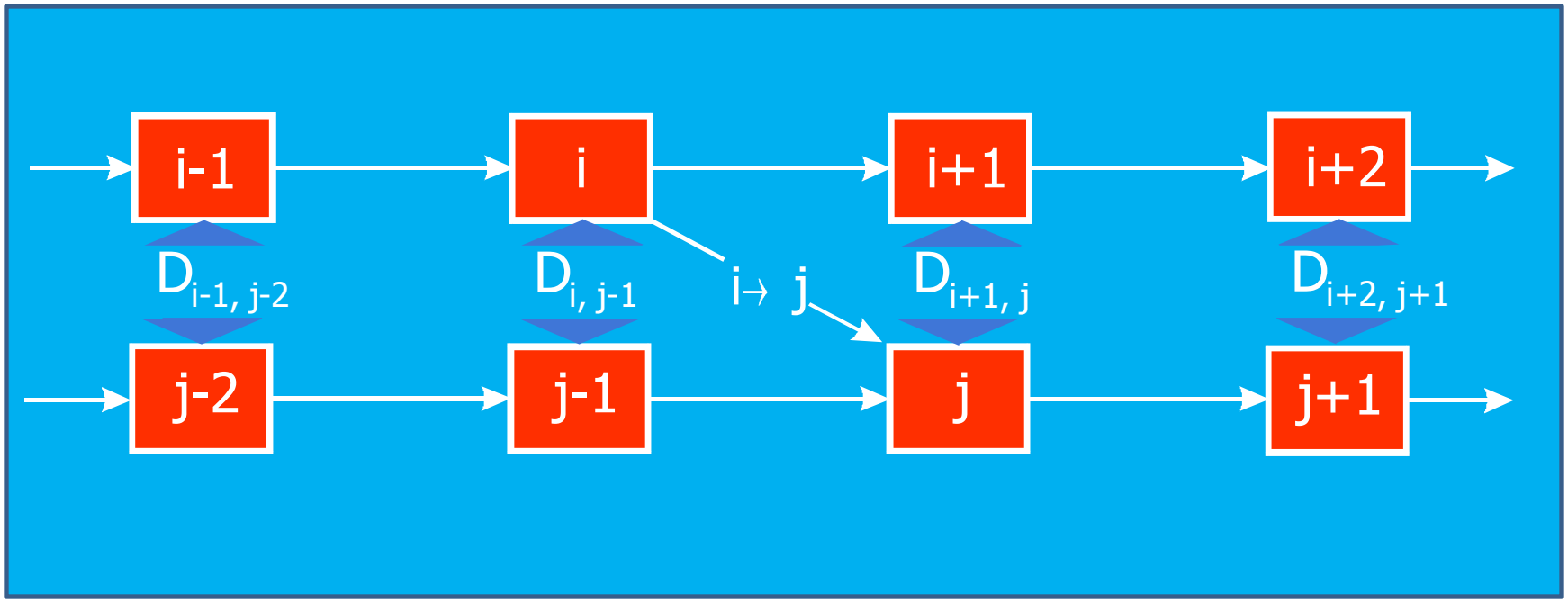
Preserving dynamics

Preserving dynamics



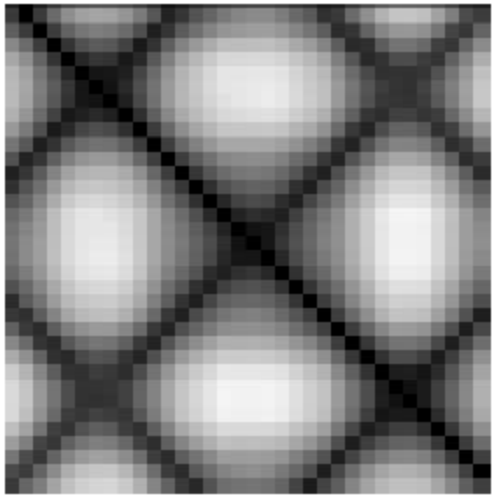
Preserving dynamics

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

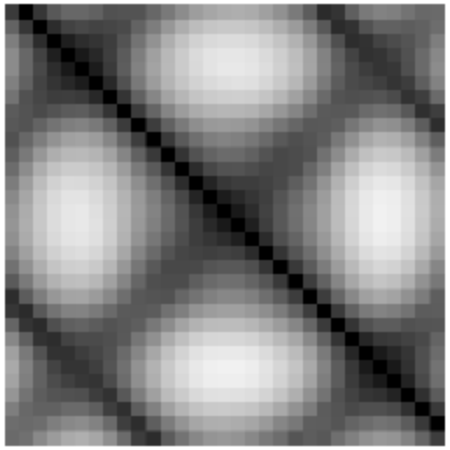


Preserving dynamics

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

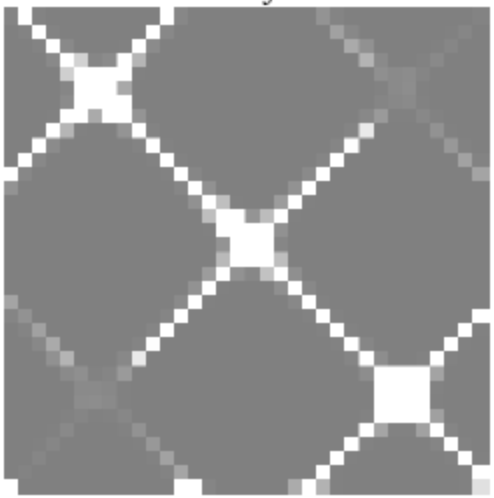


D_{ij}

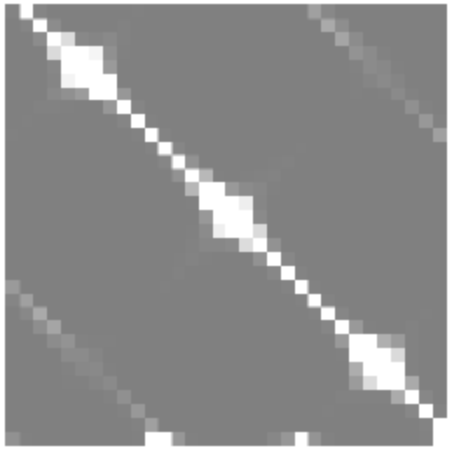


D'_{ij}

$$D'_{ij} = \sum_{k=-m}^{m-1} w_k \, D_{i+k,j+k}$$



P_{ij}

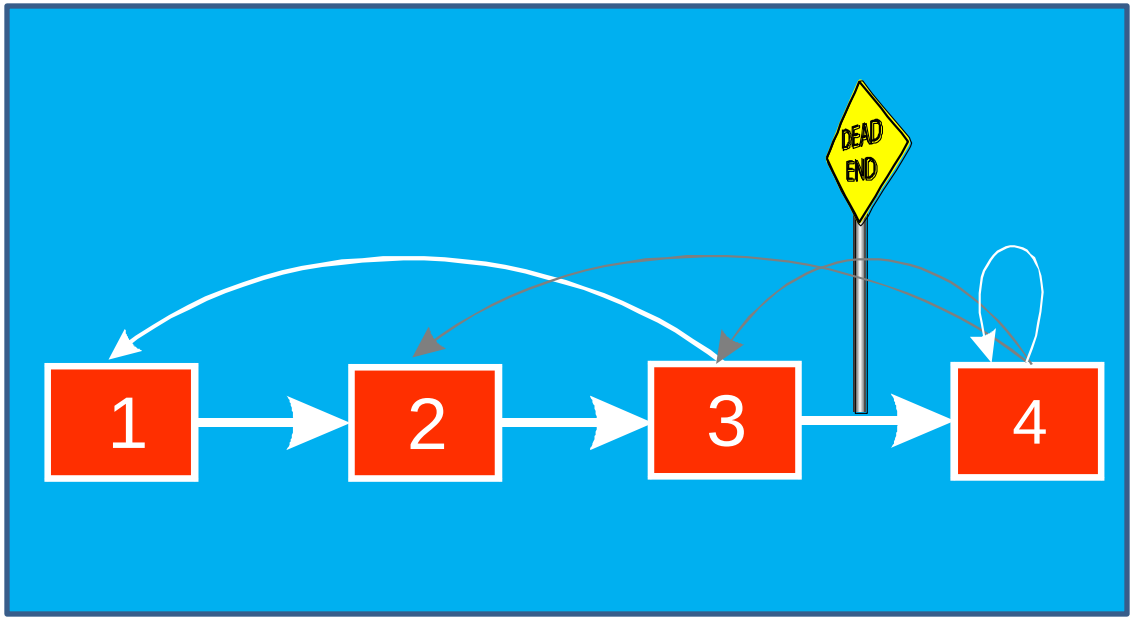


P'_{ij}

- Filter with diagonal kernel, weights *w.*

Dead ends

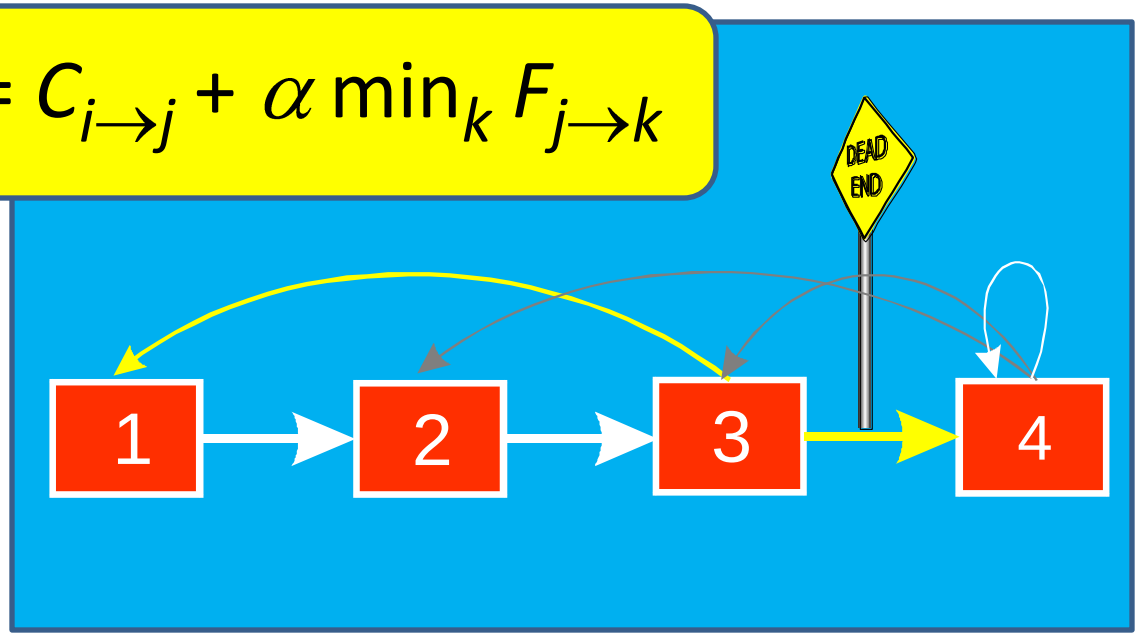
- No good transition at the end of sequence



Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

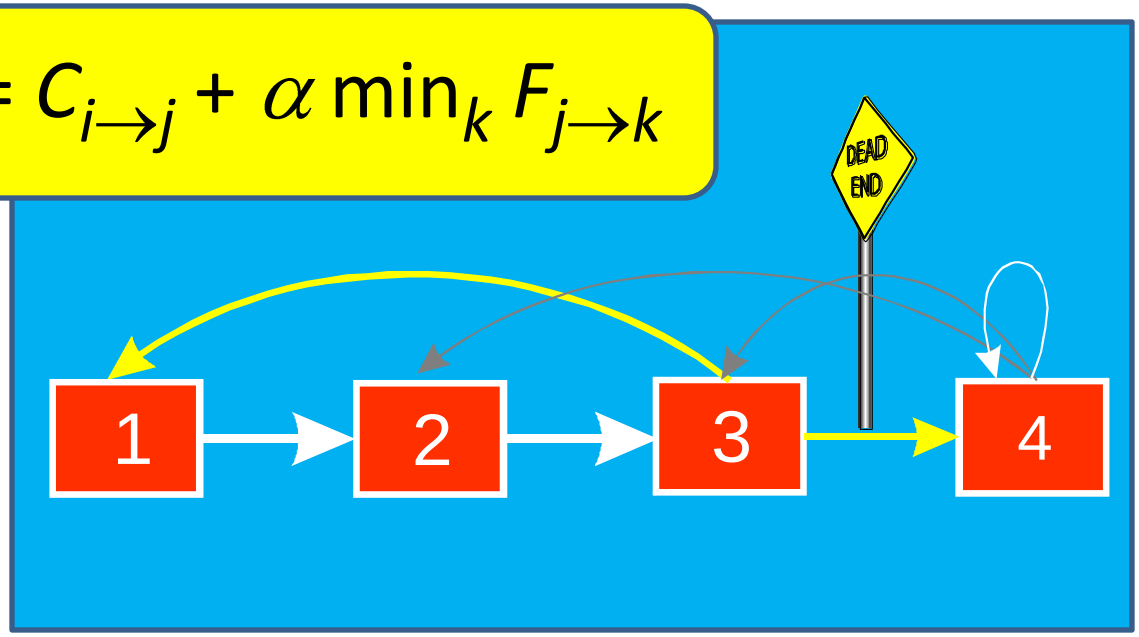
- $F_{i \to j} = C_{i \to j} + \alpha \min_k F_{j \to k}$



Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

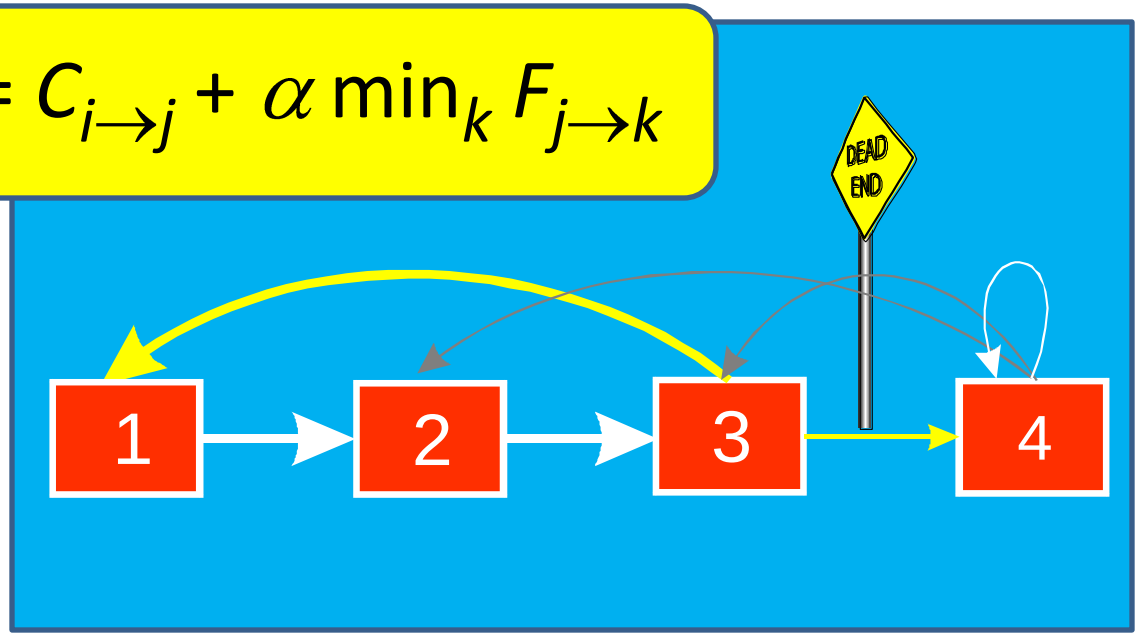
- $F_{i \to j} = C_{i \to j} + \alpha \min_k F_{j \to k}$



Future cost

- Propagate future transition costs backward
- Iteratively compute new cost

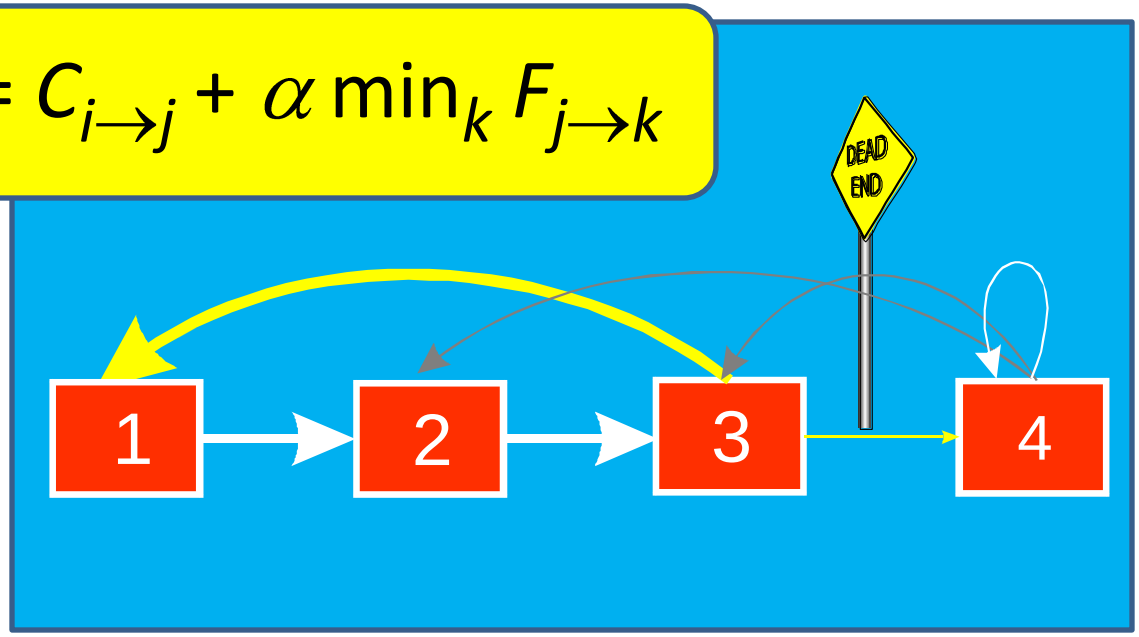
- $F_{i \to j} = C_{i \to j} + \alpha \min_k F_{j \to k}$



Future cost

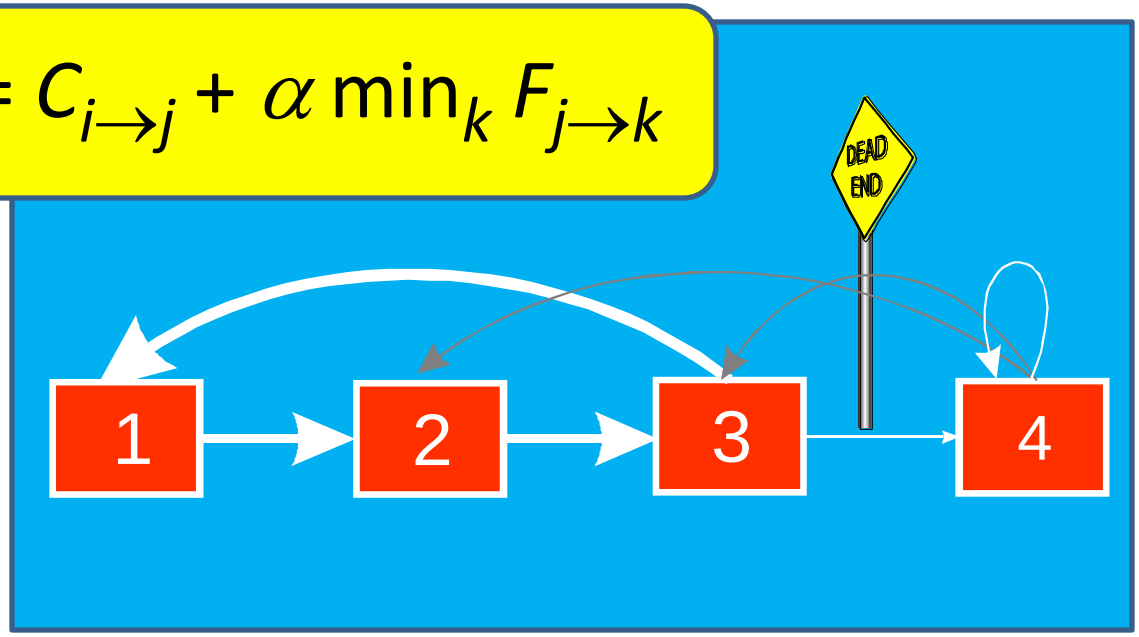
- Propagate future transition costs backward
- Iteratively compute new cost

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



Future cost

- Propagate future transition costs backward
- Iteratively compute new cost
  - $F_{i \to j} = C_{i \to j} + \alpha \min_k F_{j \to k}$
- Q-learning



Future cost – effect

Finding good loops

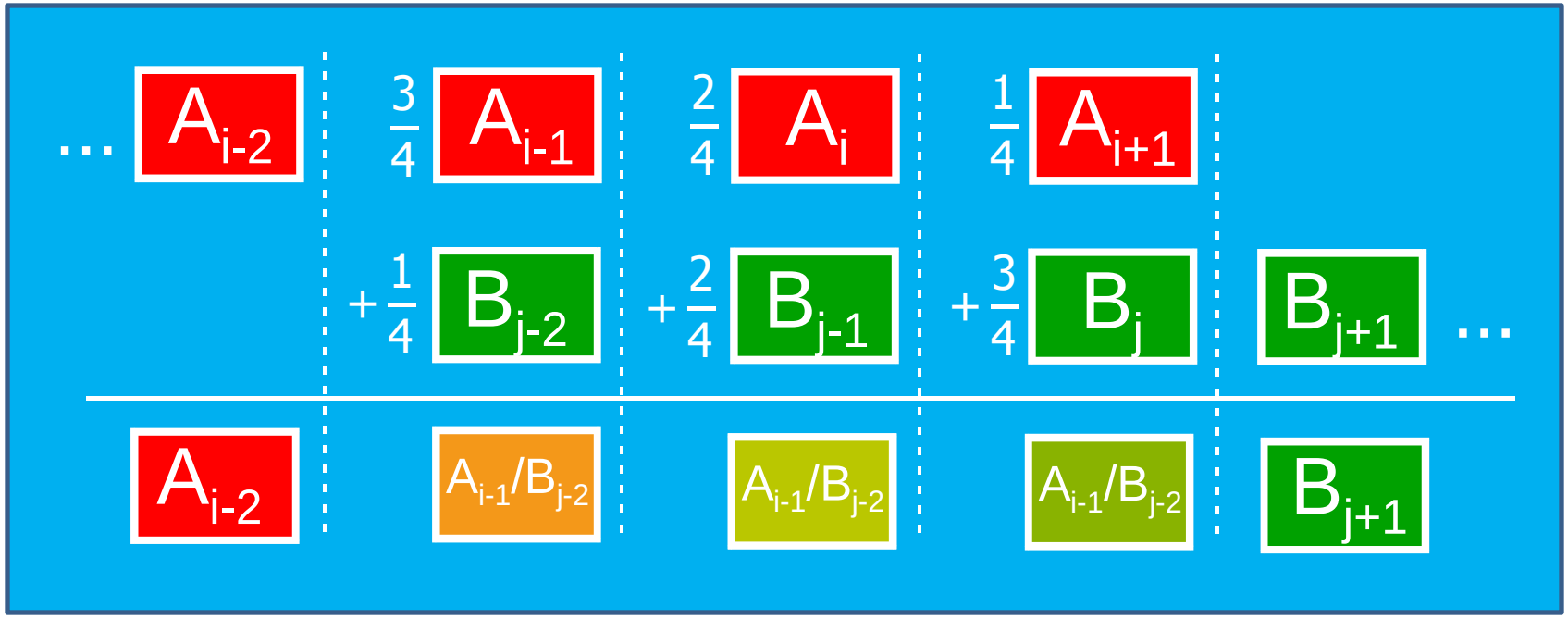
- Alternative to random transitions
- Precompute set of loops up front

Visual discontinuities

- Problem: Visible “Jumps”

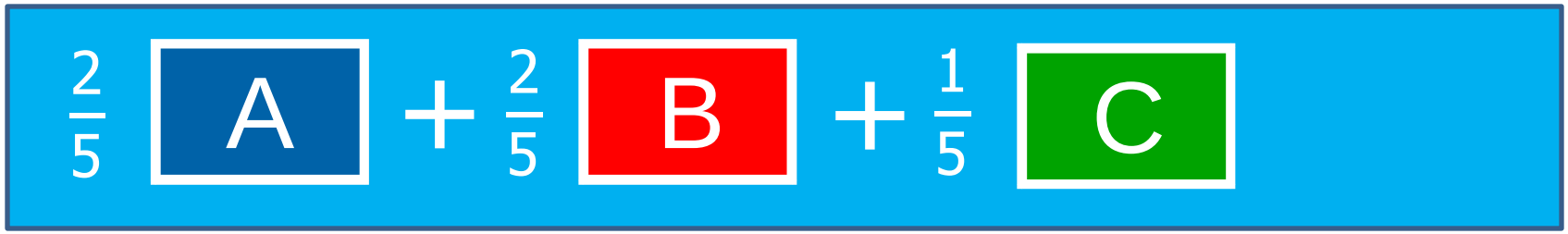
Crossfading

- Solution: Crossfade from one sequence to the other.



Morphing

- Interpolation task:

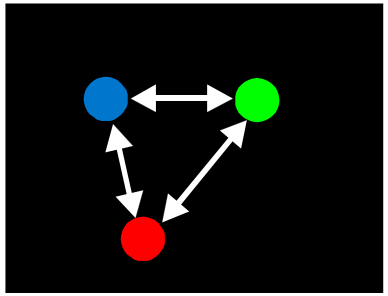


Morphing

- Interpolation task:

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

- Compute correspondence between pixels of all frames

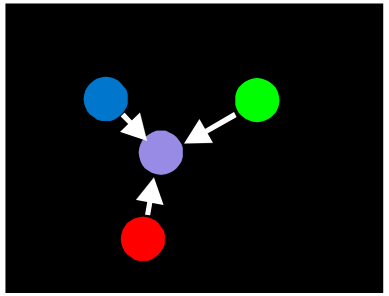


Morphing

- Interpolation task:

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

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



Results – crossfading/morphing

Results – crossfading/morphing

Jump Cut Crossfade Morph

Crossfading

Frequent jump & crossfading

Video portrait

- Useful for web pages

Video portrait – 3D

- Combine with IBR techniques

Region-based analysis

- Divide video up into regions

- Generate a video texture for each region

Automatic region analysis

User selects target range, S

slow variable fast

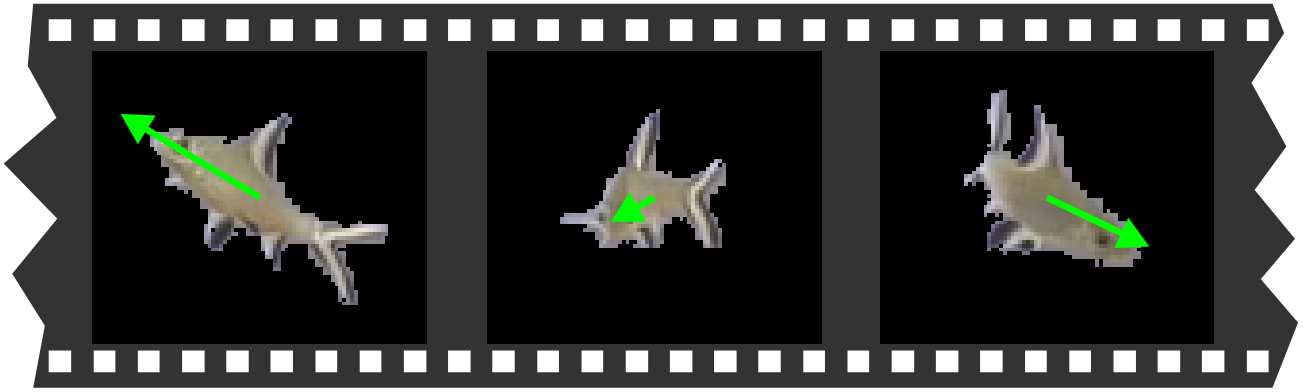
$$P_{ij} \propto \exp\left(-\frac{D''_{i+1,j}}{\sigma} - w * \text{distance}(j, S) \right)$$

Video-based animation

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

Video sprite extraction

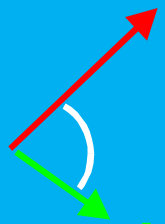
Blue screen matting and velocity estimation



Video sprite control

Augmented transition cost:

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

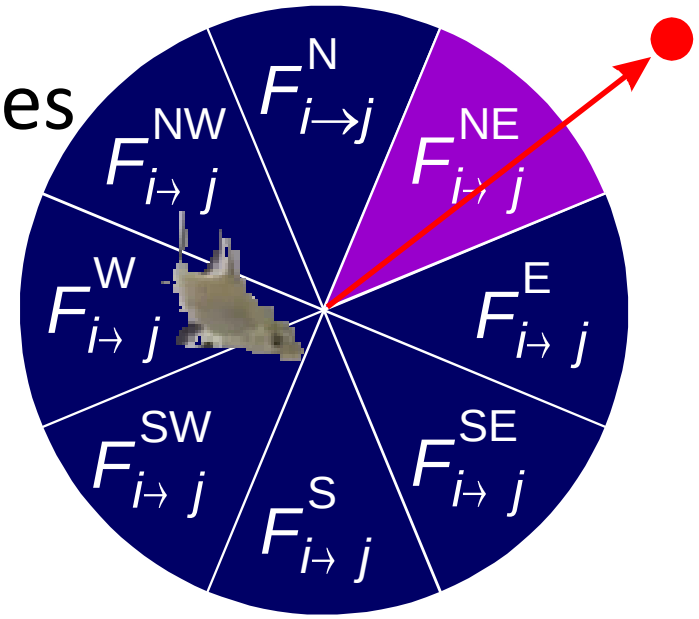


vector to mouse pointer

velocity vector

Video sprite control

- Need future cost computation
- Precompute future costs for a few angles.
- Switch between precomputed angles according to user input
- Continued in VideoSprites



Interactive fish

Discussion

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

Discussion

- Some things are relatively easy

Discussion

- Some are hard

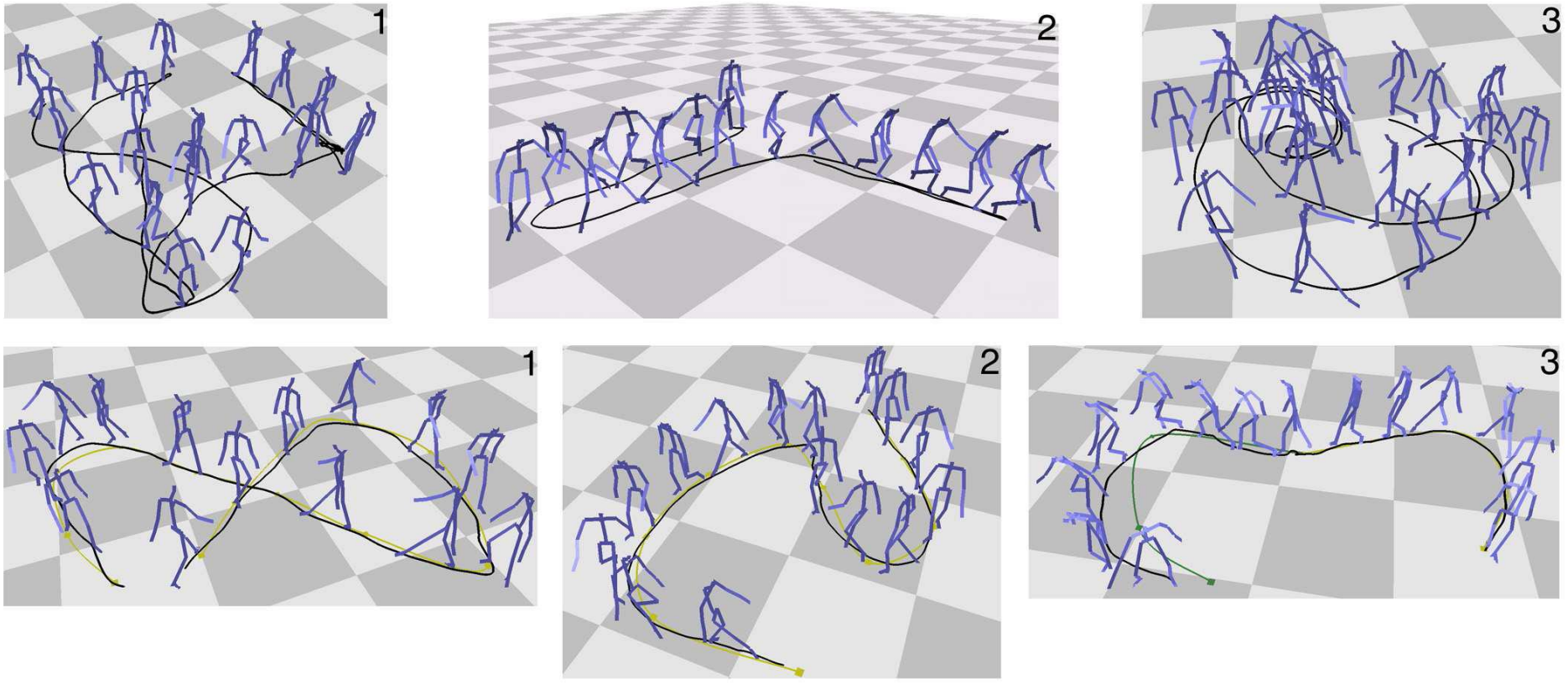
Siggraph “2000” example

VideoTextures for Motion Capture

- Resulted in 4 papers at SIGGRAPH 2002
  - Motion Graphs, by Kovar et al. <link>
  - Interactive Motion Generation from Examples, by Arikan & Forsyth <link>
  - Interactive Control of Avatars Animated with Human Motion Data, by Lee et al. <link>
  - Motion capture assisted animation: Texturing and synthesis, by Pullen and Bregler

Motion Graphs

Kovar, Gleicher, Pighin, SIGGRAPH 2002

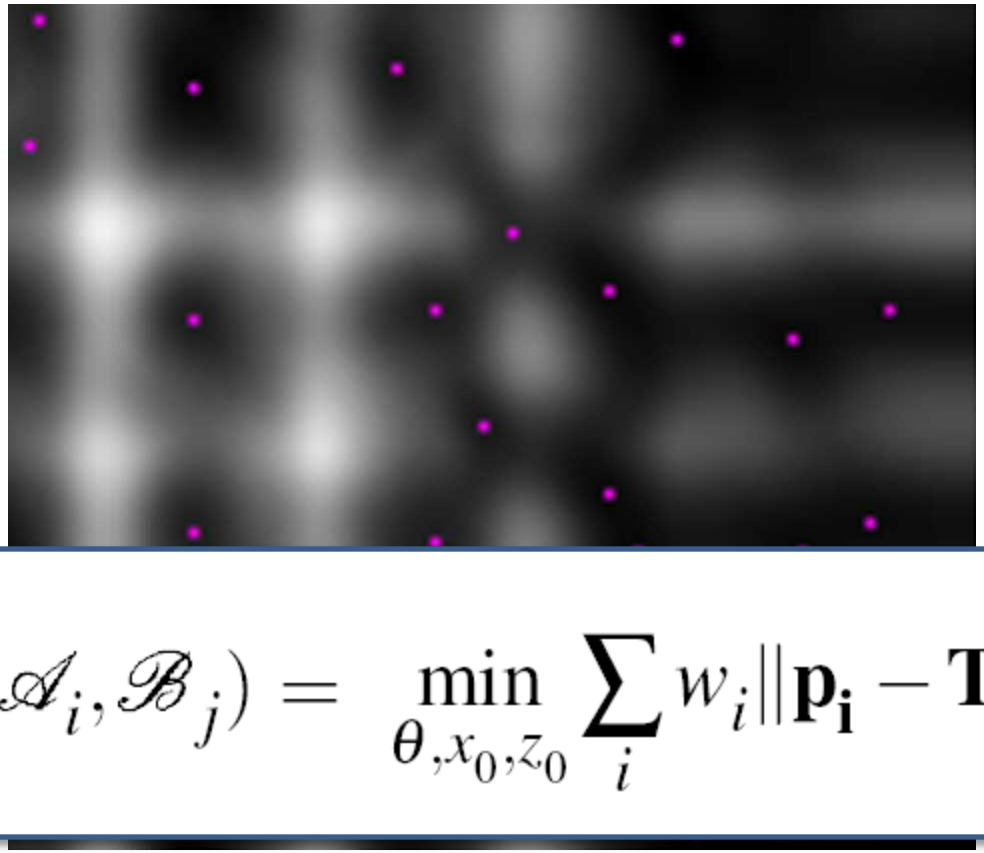


Top: Real data;

Bottom: Synthesized to match yellow line

Motion Graphs

Kovar, Gleicher, Pighin, SIGGRAPH 2002



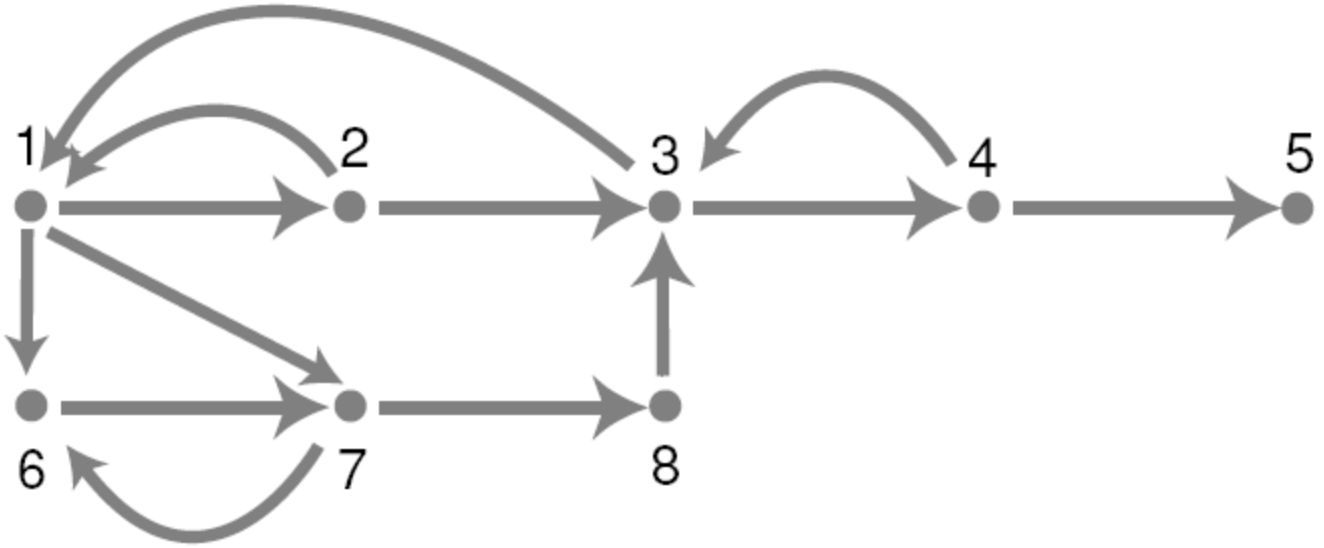
$$D(\mathscr{A}_i, \mathscr{B}_j) = \min_{\theta, x_0, z_0} \sum_i w_i \|\mathbf{p_i} - \mathbf{T}_{\theta, \mathbf{x_0}, \mathbf{z_0}} \mathbf{p'_i}\|^2$$

Distance Matrix of Mocap frames:

Based on point-cloud over 1/3 sec, ground-plane transform T

Motion Graphs

Kovar, Gleicher, Pighin, SIGGRAPH 2002



Plain path-fitting video

Multi-style path-fitting video

Interactive control video

Remaining Challenges of Video-Textures-for-Mocap

- Is this the right distance metric?
- How to interpolate poses?
- How long should the transition be?
- Pose vs. style?
- What to capture?

Flow-based Video Synthesis and Editing

K. Bhat et al. SIGGRAPH 2004

<Main video>

Video control using particle systems video

Adding video texture to CG scene <video>

Chemical Brothers' "Star Guitar" Directed by Michel Gondry

http://youtube.com/watch?v=qUEs1BwVXGA

Star Guitar (local copy)

Making of Star Guitar (1)

Making of Star Guitar (2)

“Hand” Made Videos by Guillaume Reymond