Video Effects

CS5245 Vision & Graphics for Special Effects

Leow Wee Kheng

Department of Computer Science
School of Computing
National University of Singapore

Video Effects

Two kinds of video effects:

- Video Texture
- Controlled Animation of Video Sprites

Video Texture

Video Texture

- A term coined by the authors [2].
- A continuous infinitely varying stream of images.
- Can be created from a finite video stream.

Main Idea:

- Find similar pairs of frames.
- When playing the video, jump from one of the pairs to the other.

Main Stages:

1. Analyze video: look for similar pairs of frames.
2. Generate sequence: sequence the video frames.
3. Render sequence: render the final video.

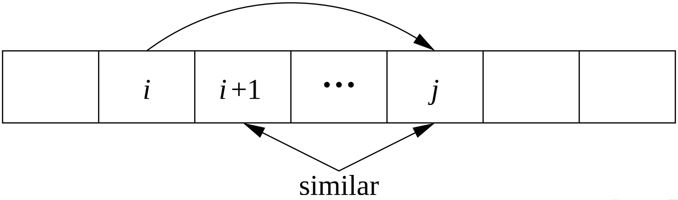
Video demo

Video Analysis

- Equalize brightness of frames.
- Stabilize video, if necessary.
- Compute difference between any two frames I_i and I_j:

$$D_{ij} = \| I_i - I_j \| . \tag{1}$$

  Can use, e.g., L_2 distance, i.e., Euclidean distance.
- Create a transition from frame i to frame j if the successor of i is similar to j, i.e., when $D_{i+1,j}$ is small.



- A simple way is to map distances to probabilities:

$$P_{ij} \propto \exp(-D_{i+1,j}/\sigma) \tag{2}$$

  and normalized probabilities so that

$$\sum_j P_{ij} = 1 \tag{3}$$

  Transit if probability is high.
- Use small σ to get good transition.

Sequence Generation

Two possible types of sequence:

- random play
- video loop

Random Play

- Start at any point before the last non-zero-probability transition.
- Repeat indefinitely:
  After playing frame i, select next frame j according to P_{ij}.

Notes:

- Next frames are selected probabilistically.
- High probability transitions are selected more often.
- Low probability transitions can still be selected.
- The sequence never repeats itself exactly because of probabilistic selection.

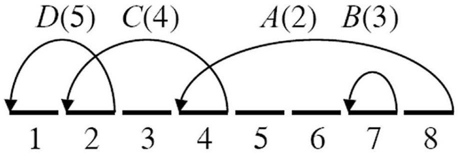
Video Loop

A short video that can be played repeatedly (loop-play) without apparent abrupt transition.

Main Ideas:

- Consider a transition $i \rightarrow j$.
- For $i \rightarrow j$ to form a cycle, need $i > j$. This is a primitive loop.
- Range of loop is $[j, i]$, cost of loop D'_{ij}.
- Can combine primitive loops into compound loops.

Example compound loops obtained from dynamic programming:



length	$A(2)$	$B(3)$	$C(4)$	$D(5)$
1		$B(3)$		
2		$B^2(6)$		$D(5)$
3		$B^3(9)$	$C(4)$	
4		$B^4(12)$		$D^2(10)$
5	$A(2)$	$B^5(15)$	$CD(9)$	$CD(9)$
6	$AB(5)$	$AB(5)$	$C^2(8)$	$D^3(15)$
		$\vdots$		

- Length = number of transitions in the loop.
- Number in brackets are costs.
- This method is used to find the list of primitive loops in the lowest-cost compound loop of a given length.

Next, schedule primitive loops found in lowest-cost compound loop.

Use $\{A, B, C, D\}$ as example:

1. Schedule the transition $i \to j$ at the end of the sequence as the first transition, i.e., A.
2. Remove scheduled transition A .
3. Now the primitive loops are divided into two continuous-range sets $\{C, D\}$ and $\{B\}$. Frame j of transition A is in the set $\{C, D\}$. Now, schedule next transition in set $\{C, D\}$ after frame j of transition A, i.e., C.
4. Repeat steps 2 and 3 until all primitive loops in the set are scheduled, i.e., D.,
5. Repeat from step 2 to schedule primitive loops in other disjoint sets, i.e., B.

The loops are scheduled in this order A, C, D, B.

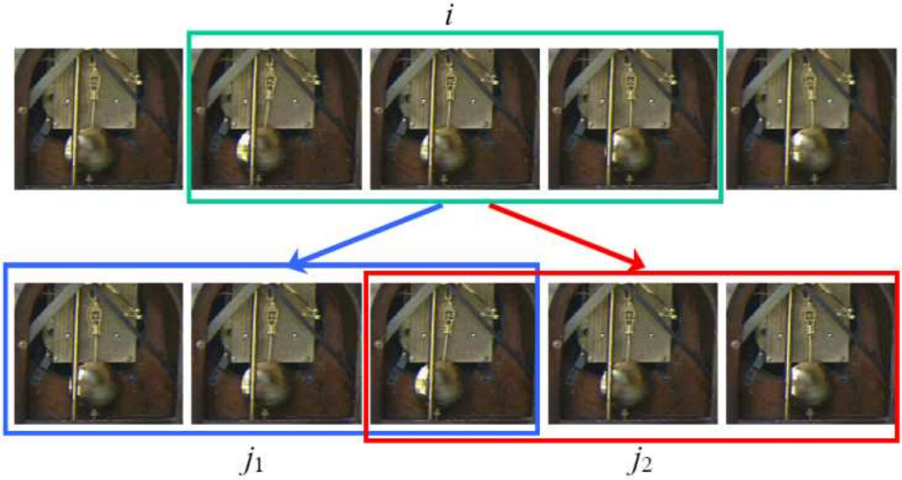
Finally, render the sequence to obtain a video loop.

Example




Preserving Dynamics

Need to preserve the dynamics of some sequences.



- In the pendulum sequence, each frame in left-to-right swing has a similar frame in the right-to-left swing.
- Transition from a frame to another similar frame in a different swing direction will create abrupt and unnatural change in pendulum's motion.

Demo:

- input sequence
- dynamics not preserved

To preserve dynamics:

- Make sure the corresponding frames and their temporally adjacent frames are similar.
- That is, match subsequences instead of individual frames.
- This is achieved by computing weighted average subsequence distance:

$$D'_{ij} = \sum_{k=-m}^{m-1} w_k \, D_{i+k,j+k} \,. \tag{4}$$

  $m = 1$ or 2.
- Compute new P'_{ij} as in the previous case.

Other details to take care of (see [2] for details):

- Avoiding dead ends: plan ahead.
- Pruning transition: select only local maxima in transition matrix P'_{ij}.

Demo: dynamics preserved

Controlled Animation of Video Sprites

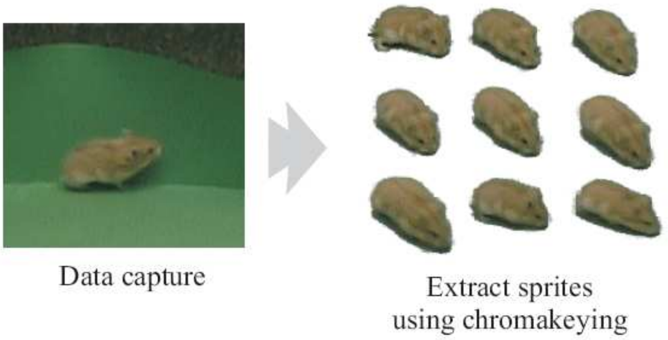
Video Sprites [1]

- Animations created by rearranging recorded video frames of a moving object.
- Uses same idea as for video texture, i.e., find similar pairs of frames that form good transitions.

Main Stages:

1. Capture natural animal motion with green screen.
2. Extract sprites using chroma keying.
3. Find good transitions by comparing all pairs of frames.
4. Find sequence of frames that show desired animation.
5. Render sequence.

Capture and Extract Sprites

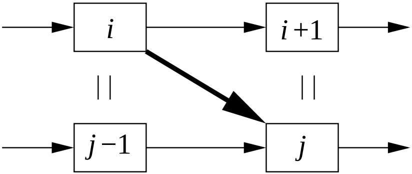


- After extracting sprites, also perform perspective correction so all sprites fall on same depth plane.
- This is to reduce size variation due to perspective projection.

Finding Good Transition

A transition from frame i to j is a good transition if

- frame i is similar to frame $j - 1$, and
- frame $i + 1$ is similar to frame j.



Cost of transition is

$$C_{ij} = D_{i,j-1} + D_{i+1,j}\ . \tag{5}$$

Train a classifier to find good transition pairs.

Sequencing Video Sprites

Let $S = \{s_1, s_2, \ldots, s_n\}$ denote the new sequence.

The cost of visual smoothness of transition $C_S(S)$ is

$$C_S(S) = \sum_{i=1}^{n-1} C_{s_i, s_{i+1}} \,. \tag{6}$$

To constrain to desired animation, add another control cost $C_C(S)$.

Then, total cost $C(S)$ is

$$C(S) = C_S(S) + C_C(S) \,. \tag{7}$$

New sequence is obtained by finding the sequence S that minimizes $C(S)$.

For details of optimization algorithm, refer to [1].

Examples

References

A. Schödl and I. Essa.
Controlled animation of video sprites.
In *Proc. ACM SIGGRAPH*, pages 121–127, 2002.

A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa.
Video textures.
In *Proc. ACM SIGGRAPH*, 2000.