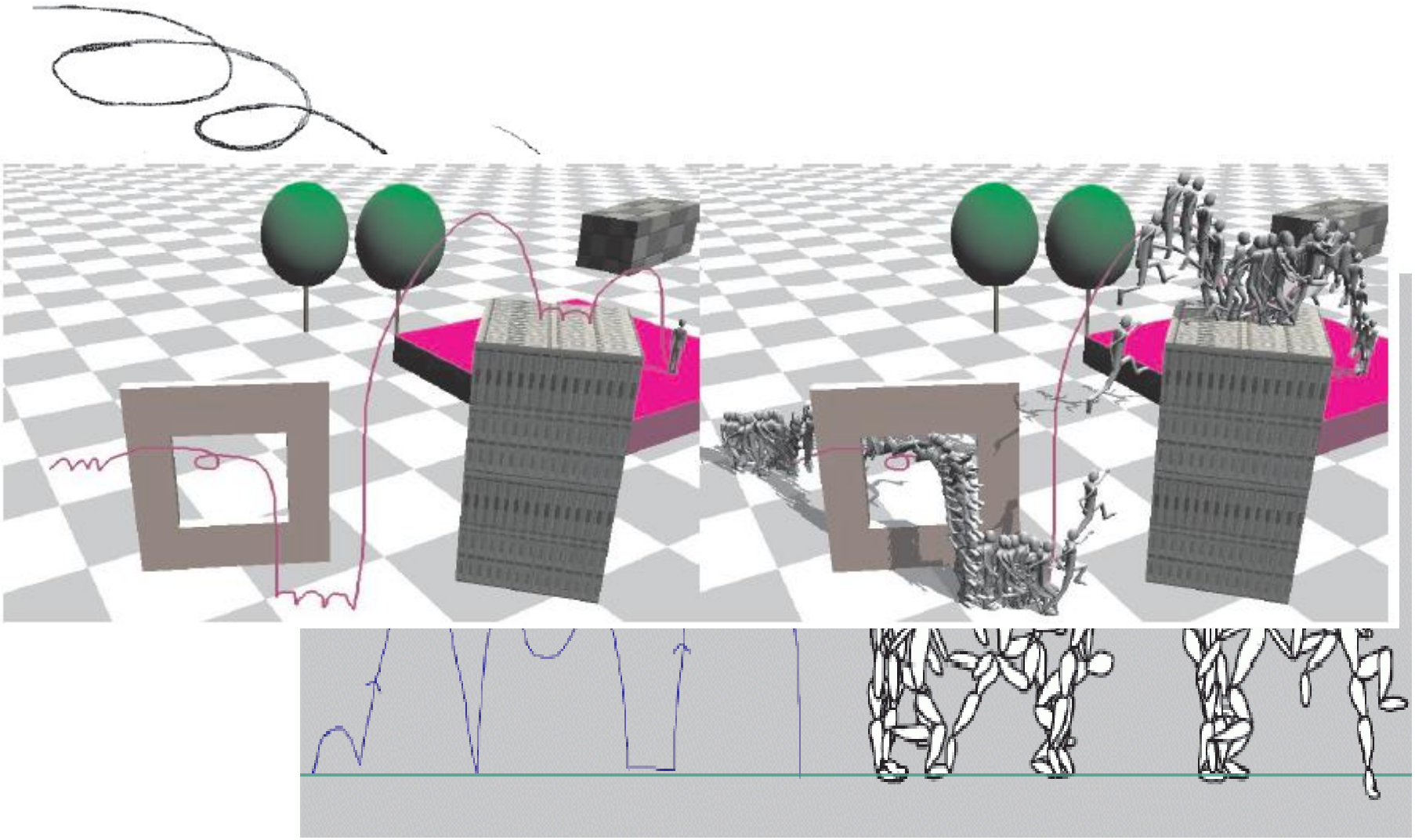
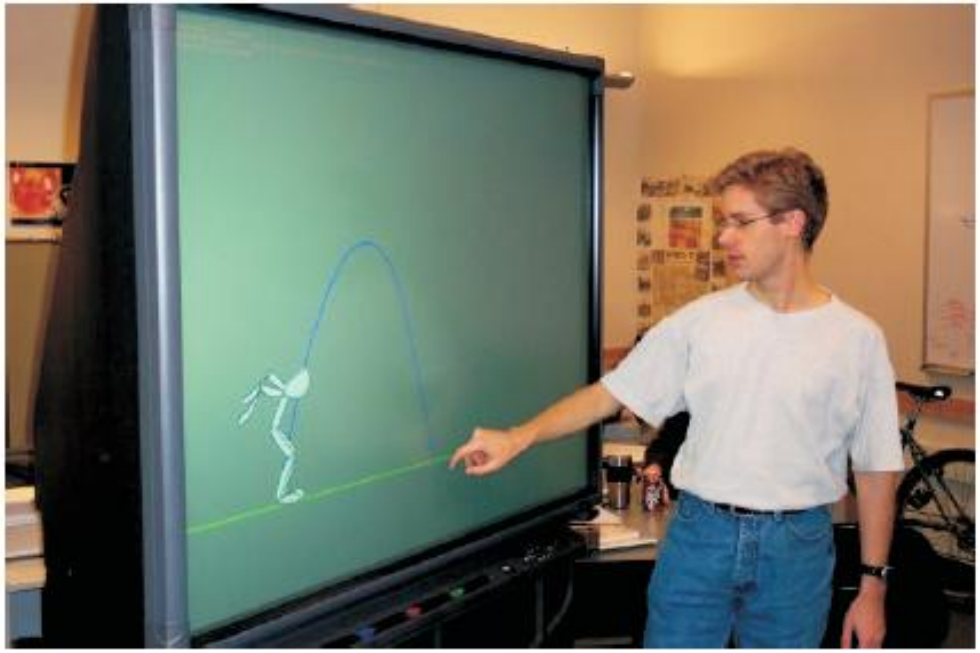
Motion Doodles : An Interface for Sketching Character Motion

Matthew Thorne , David Burke ,
Michiel van de Panne

Overview





SMARTboard

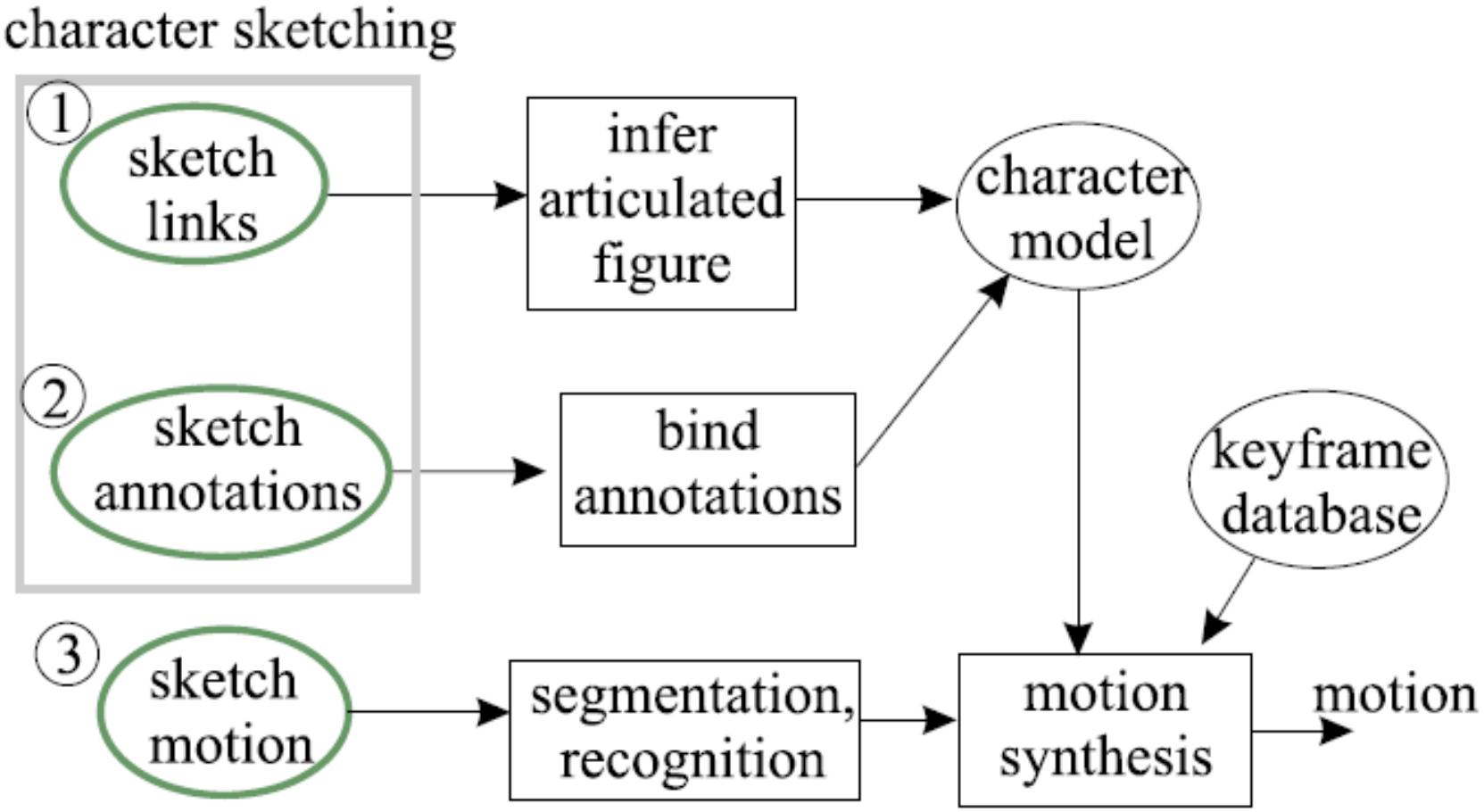
Tablet PC

Outline

- Introduction
- Contributions
- Algorithm
  - Character Sketching
  - Motion Sketching
- Conclusion

Introduction

- Interactive animation system
- Draw simple character
  - Body , head , arms , legs , feet
- Easy “draw” motion
- Gestures are highly visual in nature

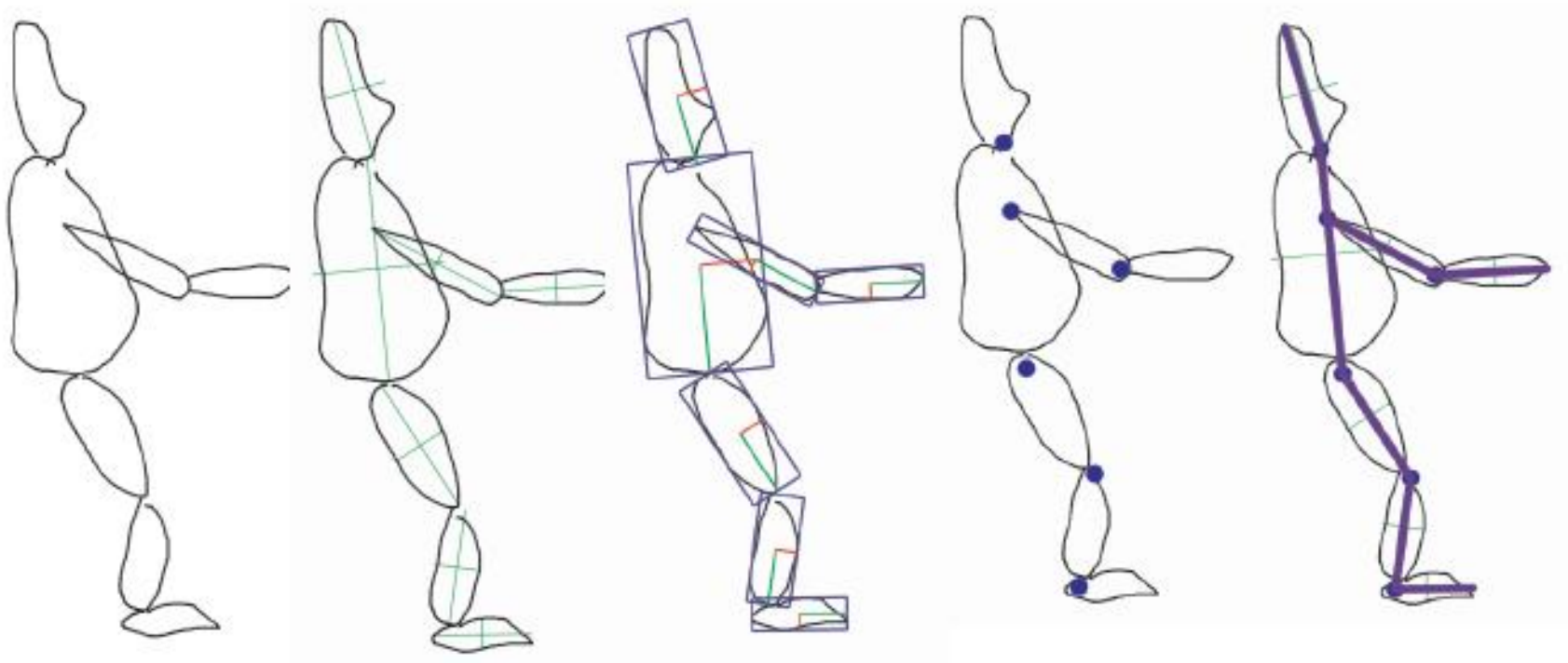

character sketching
1
sketch links
2
sketch annotations
3
sketch motion
infer articulated figure
bind annotations
segmentation, recognition
character model
keyframe database
motion synthesis
motion

Contribution

- Design a set of continuous gestures for sketching
  - Motions , locations , timing
  - Put onto variety of display device
- Easy to sketch a 2D character and then draw a variety of animated motion

Algorithm – Character Sketching

- Assumptions
  - Draw side view
  - 7 links : head , torso , upper arm , lower arm , upper leg , lower leg , foot
  - Each link is one continuous stroke
  - Draw links in any order , links can intersect
  - Surface detail : thumb , pot-belly , nose



7 sketched links

→ Compute major and minor axes

→ OBB (Oriented bounding box)

→ Compute joint location

→ Compute skeleton

Constraints make it simpler :

-- Individual links known to be a link

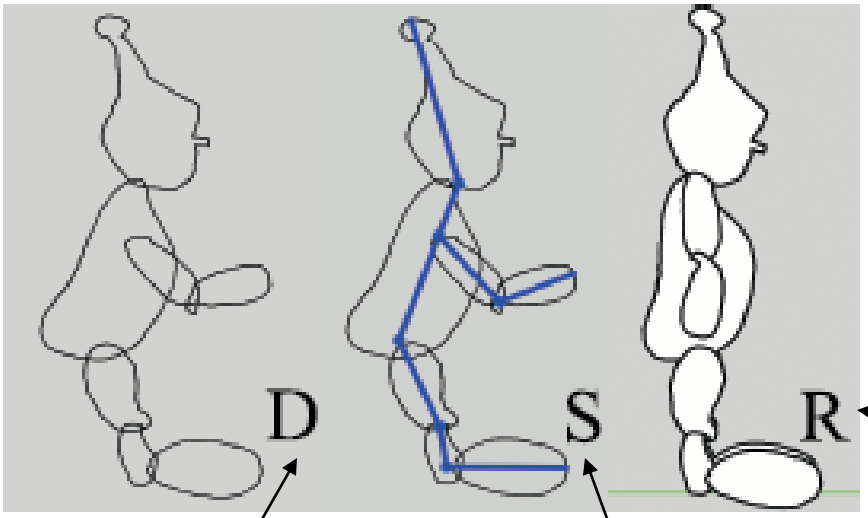
-- connectivity of the links is known in advance

1. Wait for seven links to be sketched
2. Fit oriented bounding boxes to all links
3. For each link i
4. For each major-axis end-point on link i, P_i^1 and P_i^2:
5. Search all links $j \neq i$, for the closest point, P_j
6. If links i and j are not aligned
7. create joint J_n at intersection of major axes of i and j
8. else
9. create joint J_n at midpoint of P_iP_j
10. Identify and remove all duplicate joints
11. Identify links based on connectivity
12. Create duplicate arm and leg segments.

如果兩個major axes 在20度之內 => 接近平行

Torso 有三個 joint ,head 只有一個 joint 且連結在torso上。

fail => report to user



Fail :

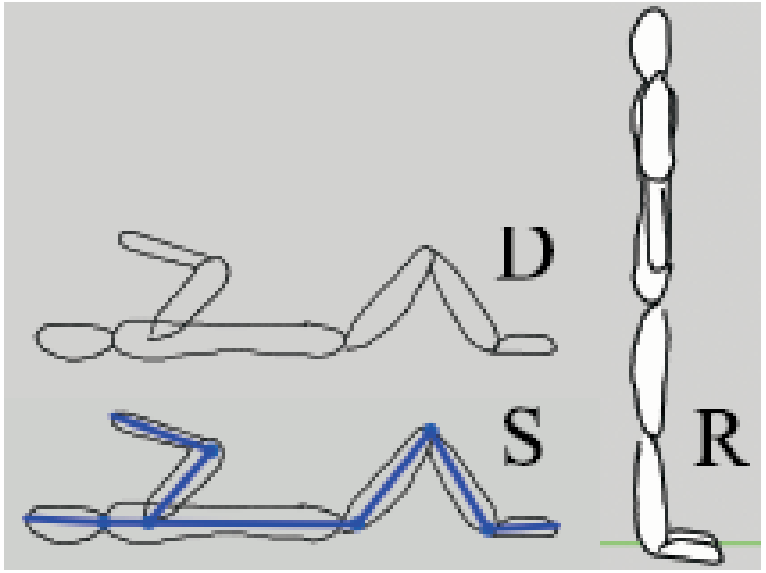
1. arm downwards parallel to the torso
2. hands are located close to the head , knees , or feet

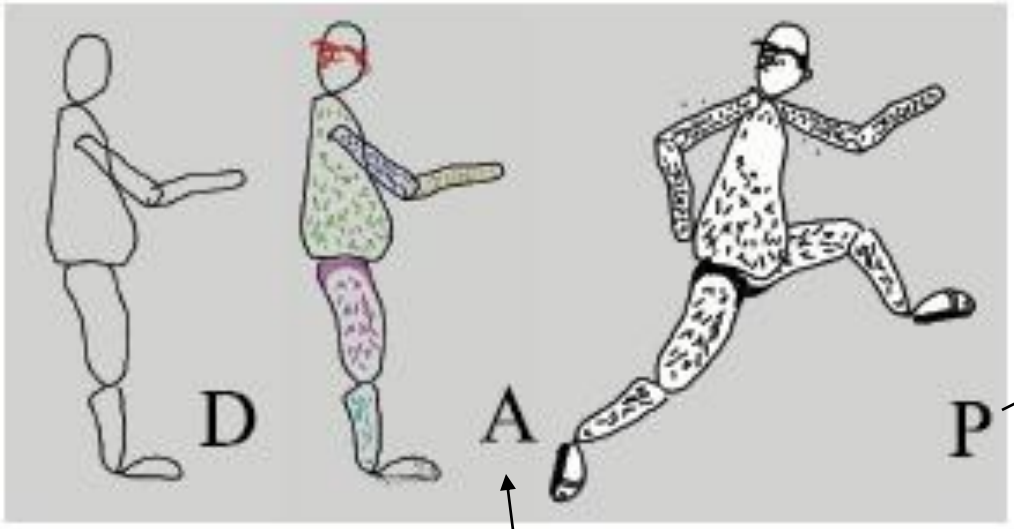
R : reference pose (initial at animation start) => all bones vertical , feet horizontal

D : original drawing

S : inferred joints and the fitted skeleton

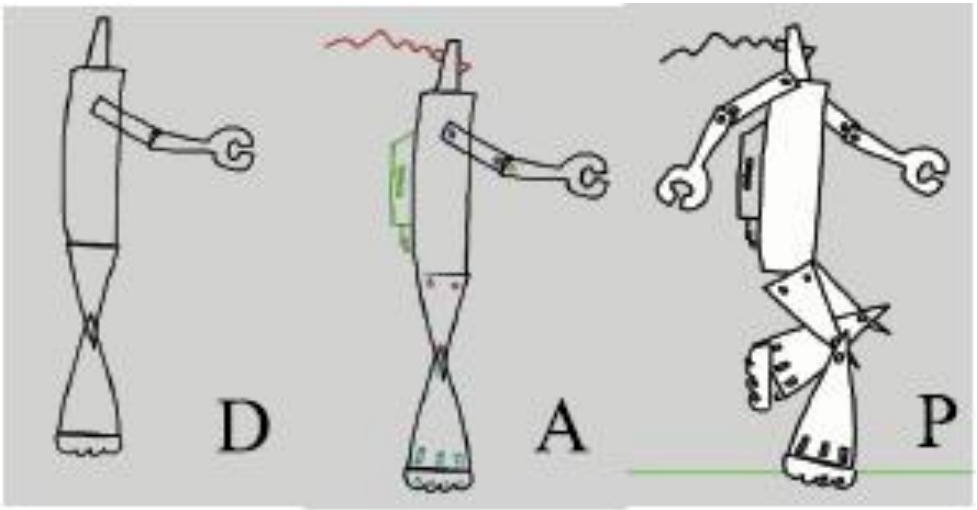
Currently , user cannot refine the skeletons





P : an animated pose

A : annotations



Decorate character : eyes , ears , hands , hair , a hat , a nose , shoes

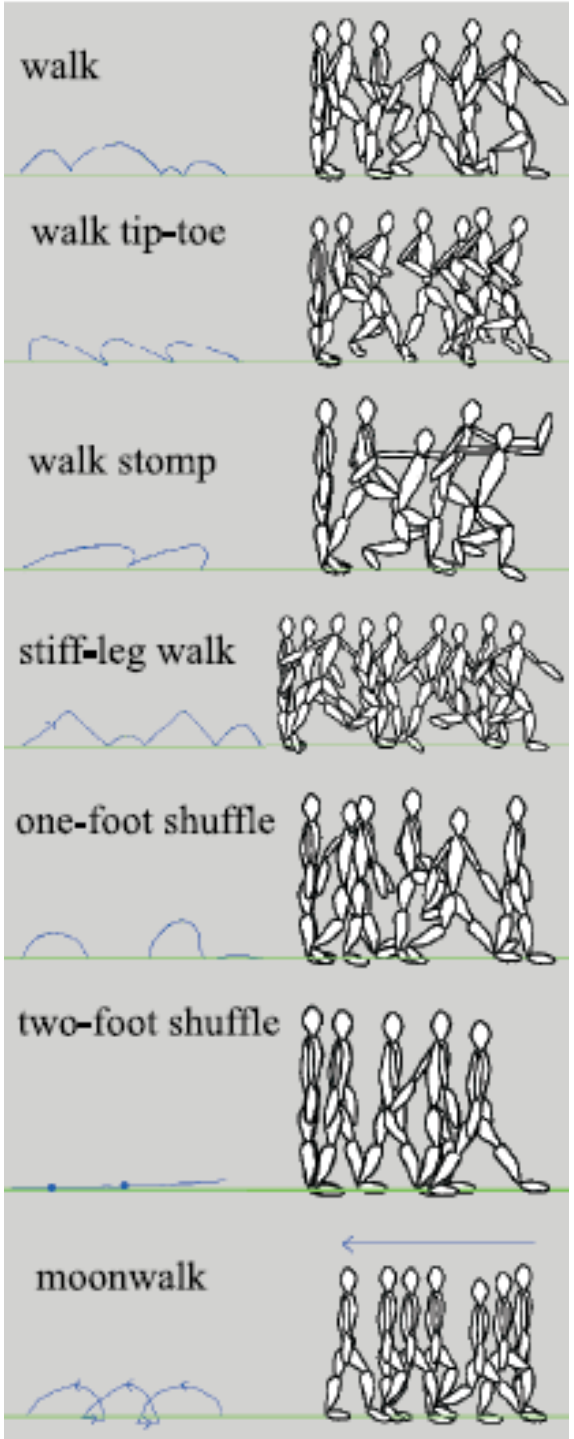
Limitation : cannot cross multiple links

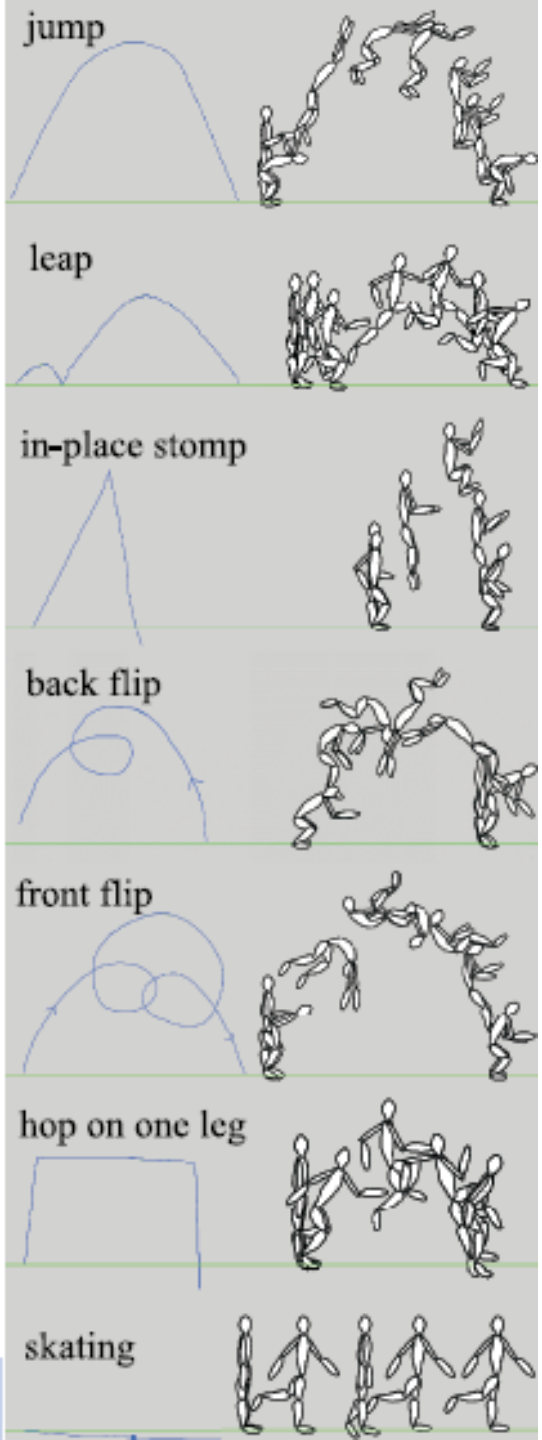
Motion Sketching

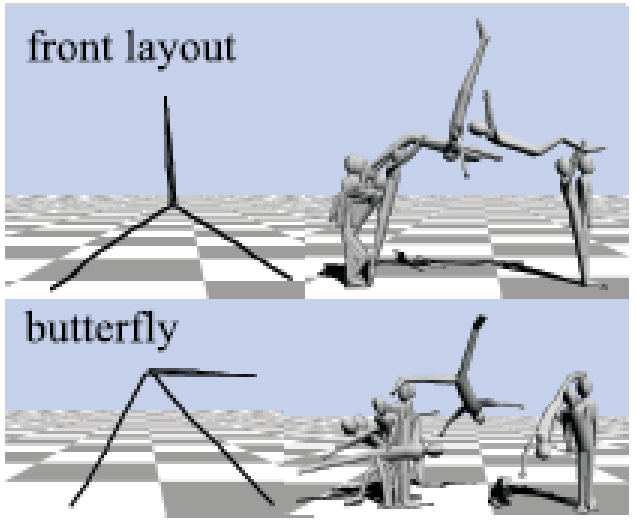
- Motion sketch needs:
  - Type of Motion
  - Spatial location of Motion
  - Extent of Motion
  - Timing of Motion
- Cursive gesture

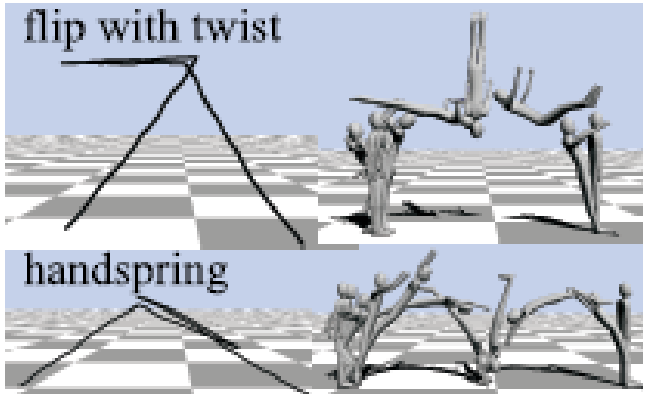
Gesture vocabulary design principles

1. Gesture should be cursive => smoothly flow
2. Limited number easy-to-draw gestures
3. Reminiscent , can extend
4. Allow forwards or backwards
5. Similar motion ⇔ similar gesture
6. Allow for the generation of stylistic variations


walk
walk tip-toe
walk stomp
stiff-leg walk
one-foot shuffle
two-foot shuffle
moonwalk


jump
leap
in-place stomp
back flip
front flip
hop on one leg
skating

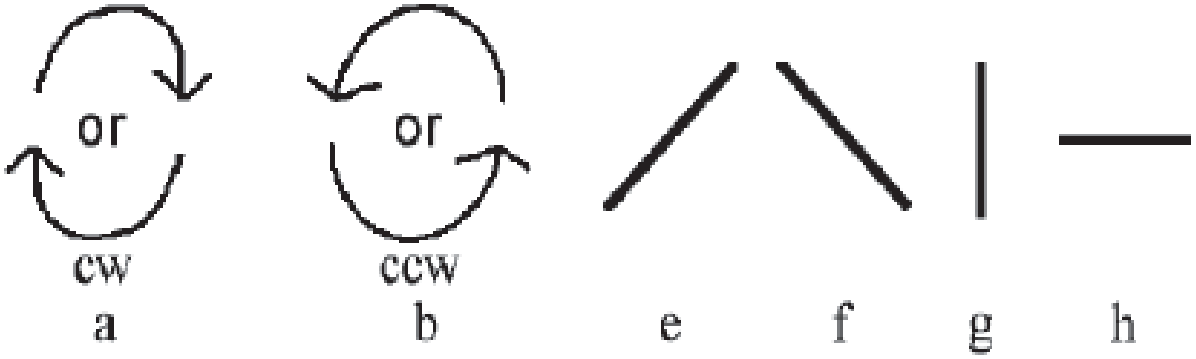

front layout
butterfly


flip with twist
handspring

Sketch Segmentation

- 3 stages
  1. Tokenization stage
     - Produce a corresponding list of token
  2. Parsing stage
     - Identify the set of admissable gestures
  3. Identification stage
     - Identify the specific motions to be generated

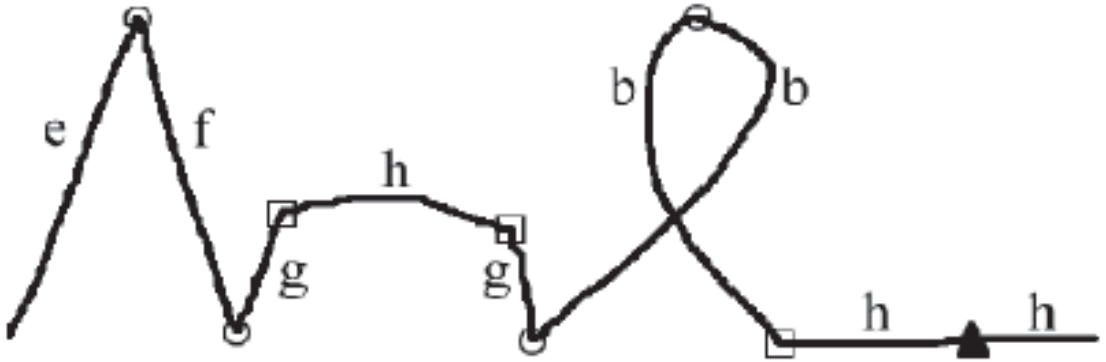
tokens



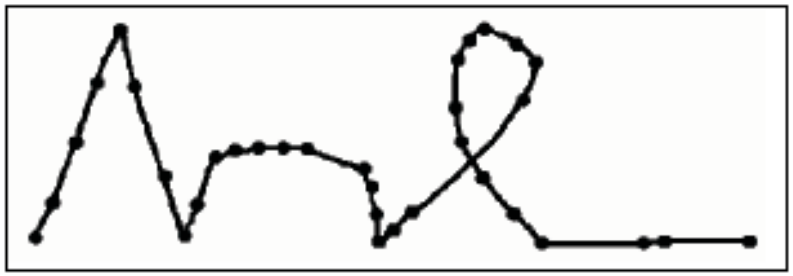
Use six token

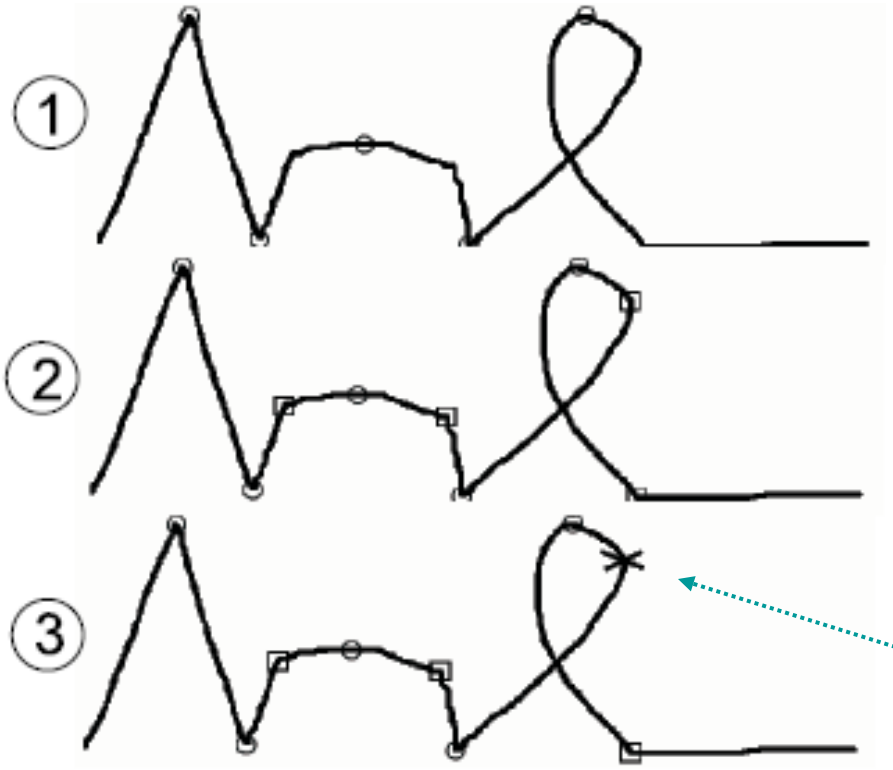
Tokenization stage

example token labeling



input:
time-stamped
points



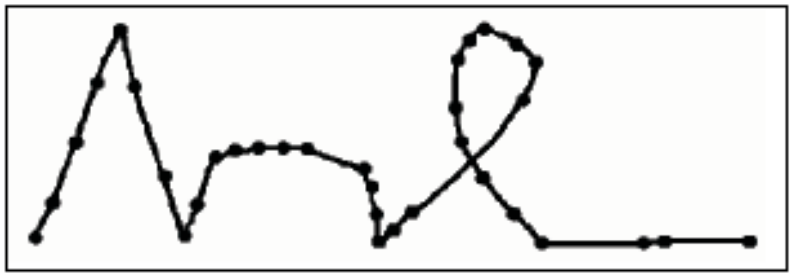


Segmentation on vertical direction (mark at the point of direction changed)

Corner detection

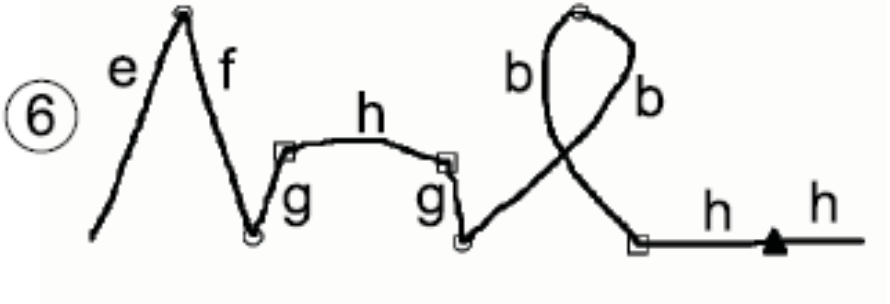
Refine the corner

input:
time-stamped
points



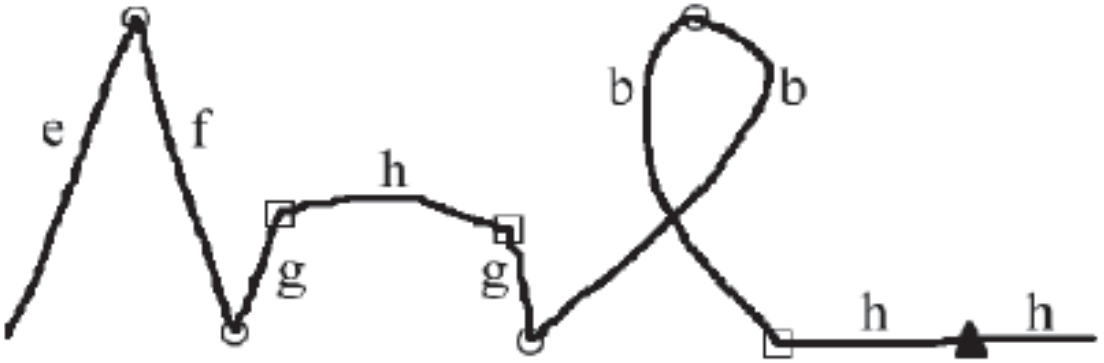
Specify either straight or curve
(line : segment ratio r<1.2)
Merge colinear segments

Identify pauses (add point)



Token assignment
straight(<30 degree) => g, h
straight(>30 degree) => e, f
curve => a, b

example
token
labeling



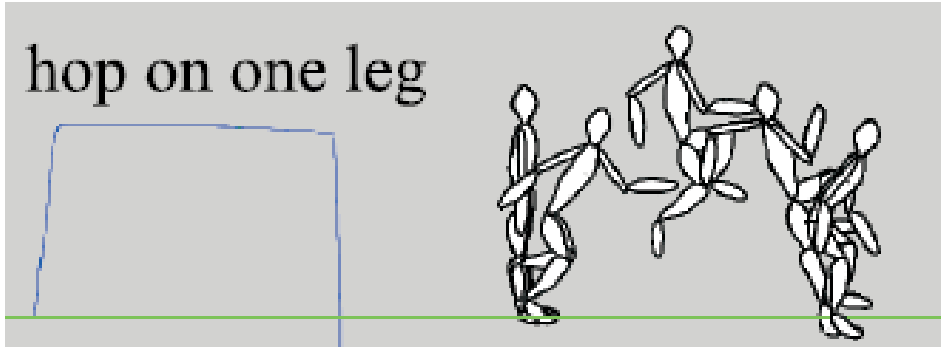
Parsing stage

Regular Expression !

gesture
identification

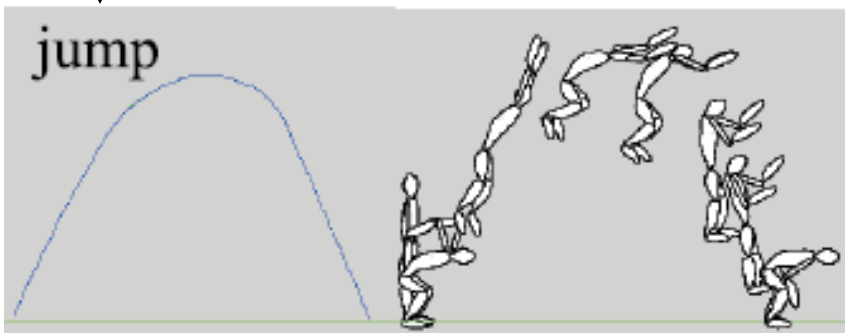
ARC	UP LOOP	DOWN LOOP	SPIKE	HAT	LINE
aa, af, ag, ag, ea, ga	bb, gb, fb, eb, bg, bf, be	b(bb)*b	ef, eg, ee, gf, gg, gf, ff, fe, fg	(f|e|g|a|b)hh * (f|g|e|a|b)	e, f, g, h

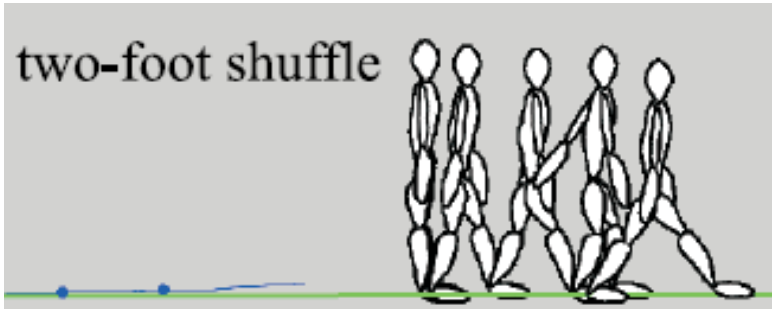
Unique interpretation =>

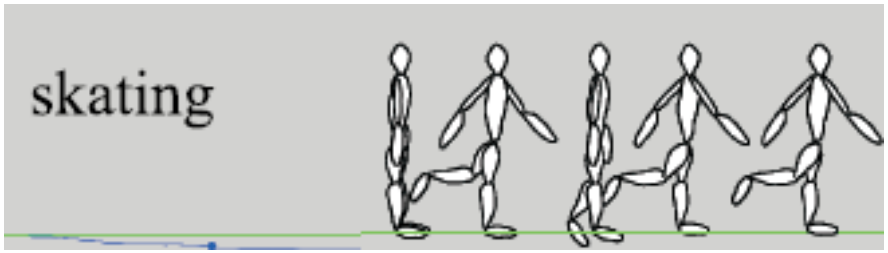


$h > h_{walk}$

頂端傾斜的程度





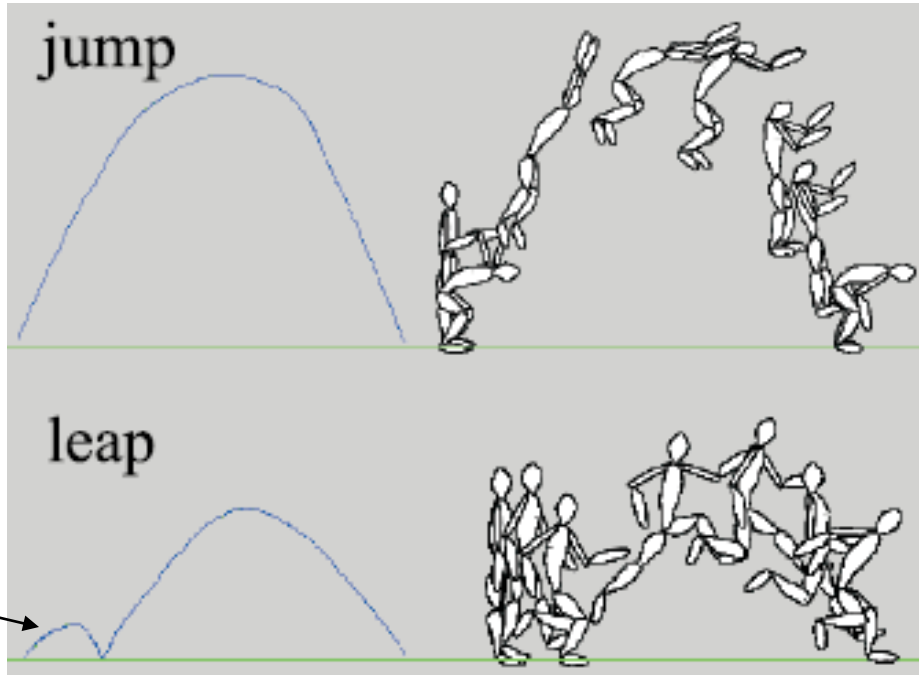


Step < $d_{maxstep}$

Step > $d_{maxstep}$

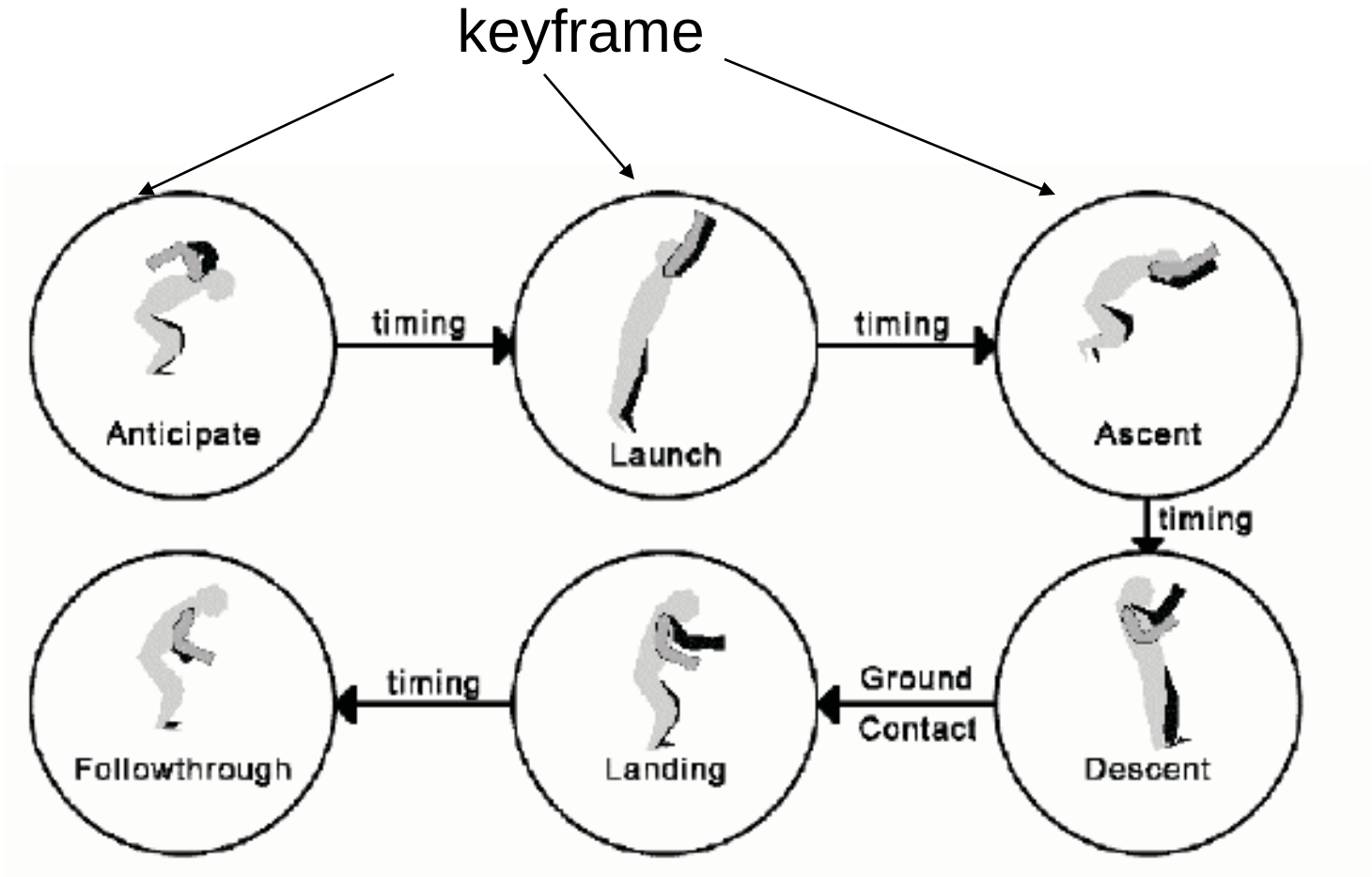
Pen remains stationary for more 0.5s => standing posture

Walk before jump => leap



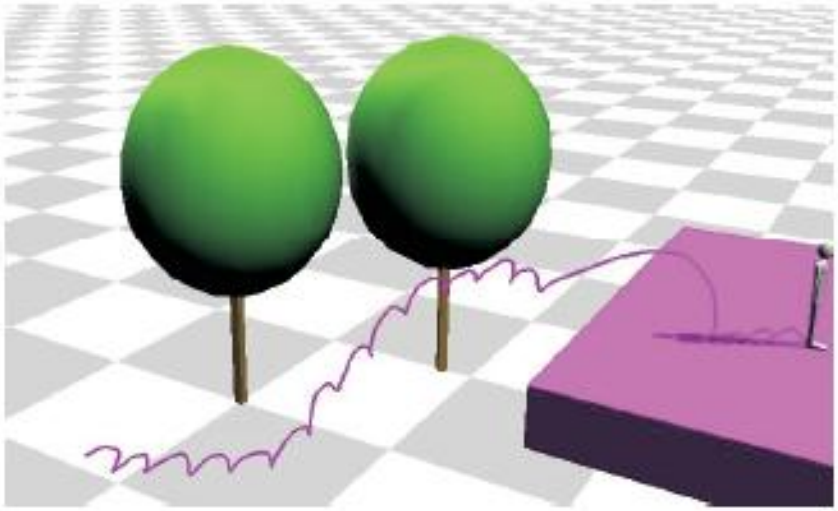
Output Motion Synthesis

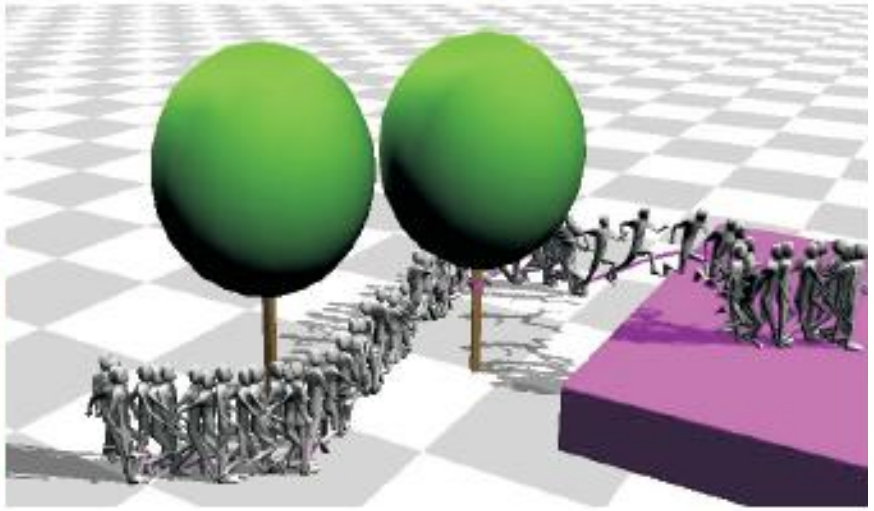
- More information
  - Duration , height , rotation
- Parameterized keyframe-based motion synthesis
  - Keyframe database
  - Keyframe interpolator (Catmull-Rom)
  - Inverse-kinematics solver
  - A means to position the center-of –mass at a specify point

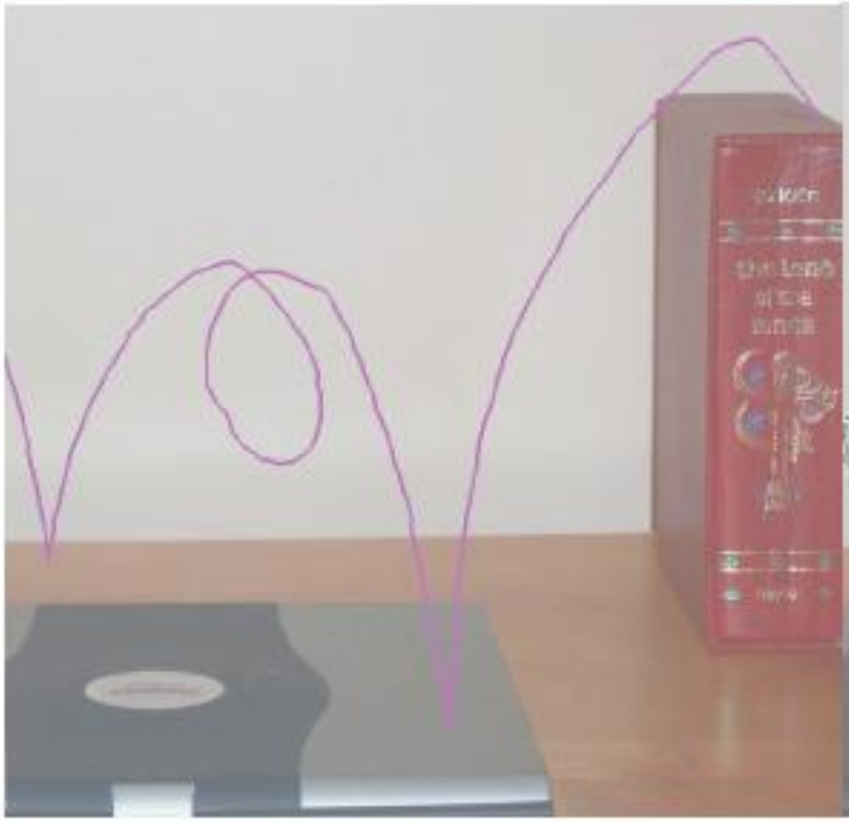

keyframe
Anticipate
timing
Launch
timing
Ascent
timing
Descent
Ground
Contact
Landing
timing
Followthrough

Sketching in 3D environments

- Set camera such that it covers the desired workspace
- 3D character modelled in advance
- Find start and end points of each gesture
  - Find the vertical plane embeds the 3D start and end points , process like in 2D
  - Using vertical direction , corner metric
- Limitations

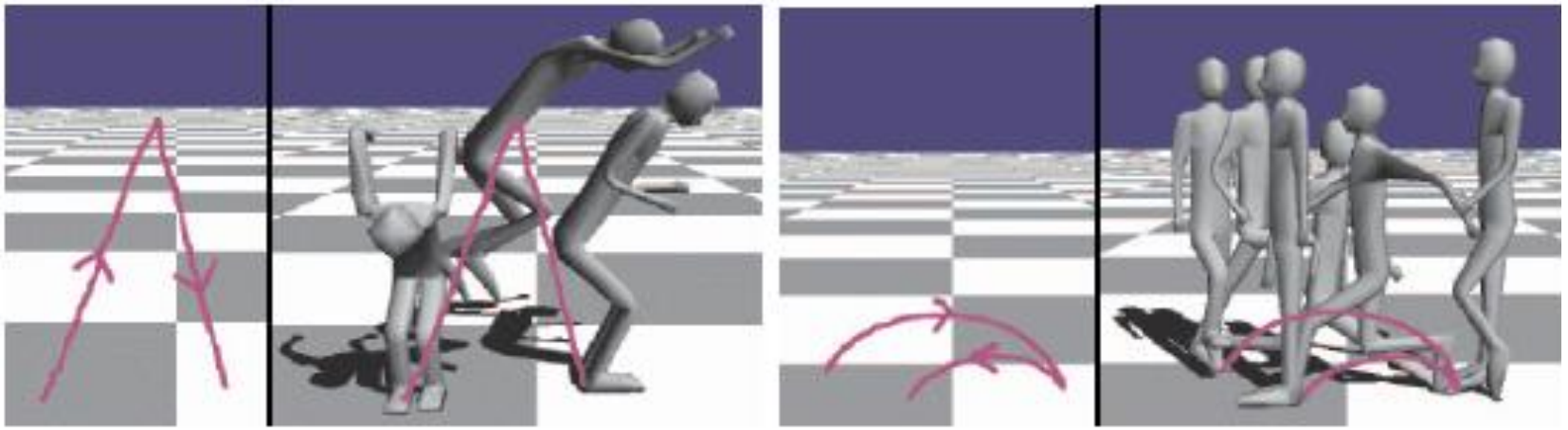






Limitations

- Motions directly towards camera or away
- Ambiguity of gestures
  - In-place stomps v.s. slides
- Direction
  - Forward or backward



Conclusion

- Highly-accesible means for users to create a certain class of character animations
- User rapidly learned the gesture vocabulary and enjoy it
- Add a new motion => create a new gesture that can be identified