Data-driven synthesis based on Motion Graphs

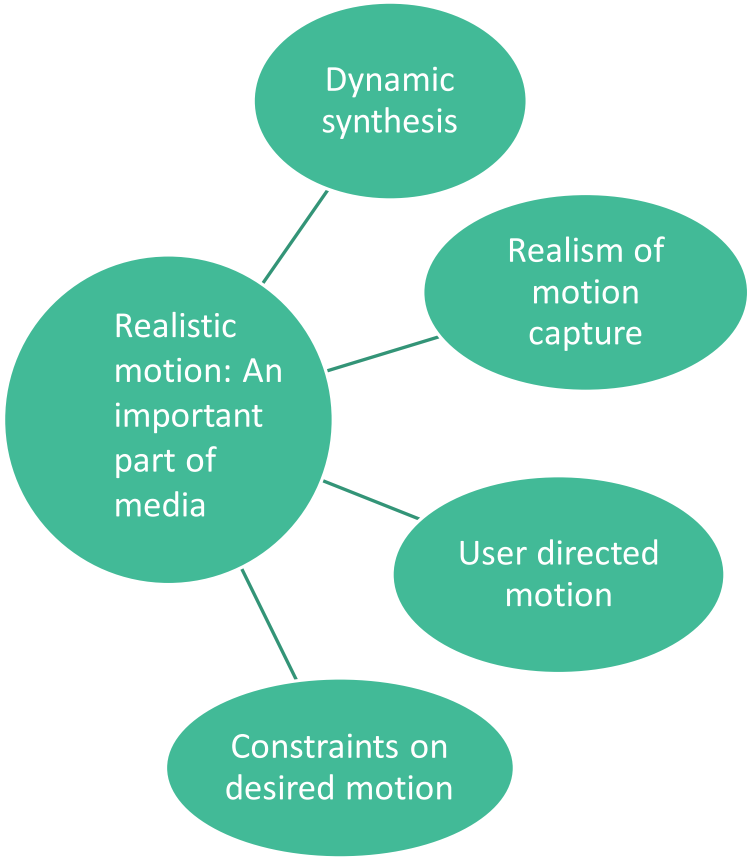
VARSHINI MUTHUKUMAR

MASTER IN VISUAL COMPUTING

MATRICULATION NO. - 2572827

Contents

Motivation


Dynamic synthesis
Realism of motion capture
Realistic motion: An important part of media
User directed motion
Constraints on desired motion

Overview:

Detect candidate transitions
- Identify similar clips
- Compute difference in poses

Create transitions
- Straightforward blending
- Join motion seamlessly

Create motion graphs
- Directed graph
- Reassemble motion clips

Motion synthesis
- Graph walk
- Select sequence of nodes

Motion capture data

Figure 1.(a)

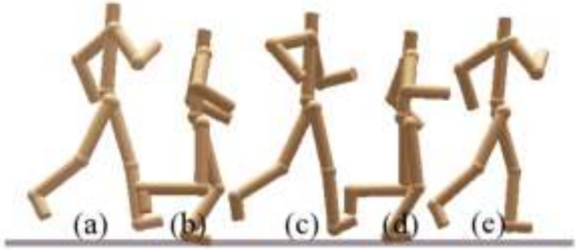


Figure 1.(b)

Sampling root position and joint rotation parameters

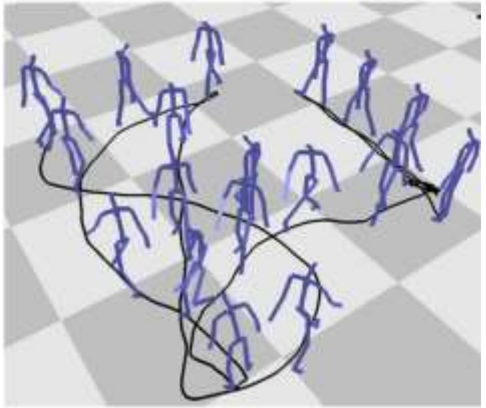


Figure 1.(c)

Figure 1.(d)

Figure 1.(a)- (d) : Example motion clips

Identify similar clips:

How to detect similarity?

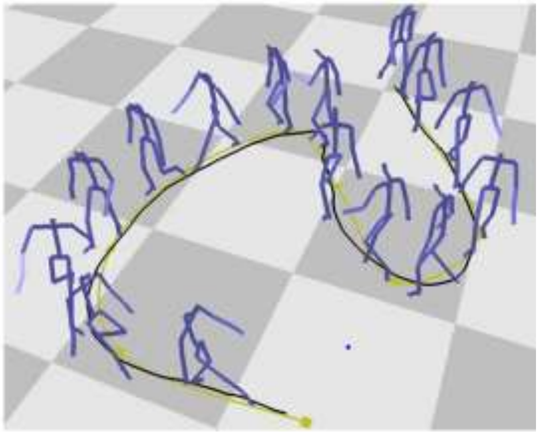


Figure 2.(a)

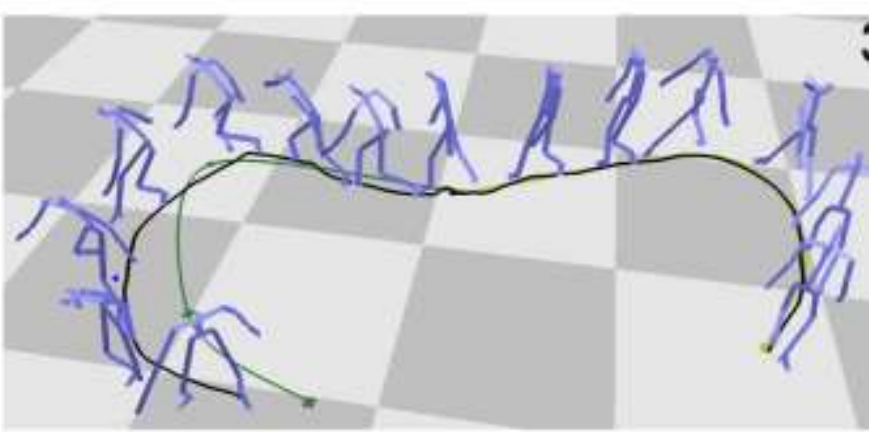


Figure 2.(b)

Figure 2.(a),(b) : Example motion clips

Identify similar clips:

How to detect similarity?

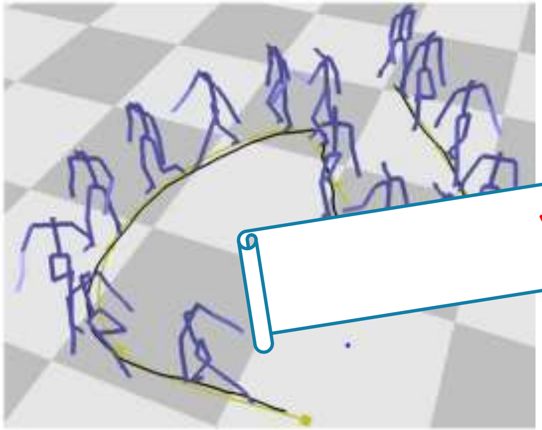


Figure 2.(a)

Vector norms

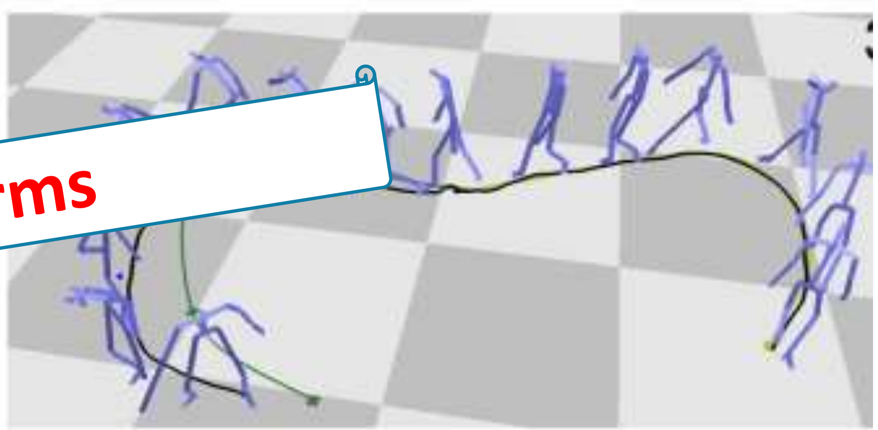


Figure 2.(b)

Figure 2.(a),(b) : Example motion clips

Identify similar clips:

But do vector norms work?

Problems:

- Fail to account for meanings of parameters

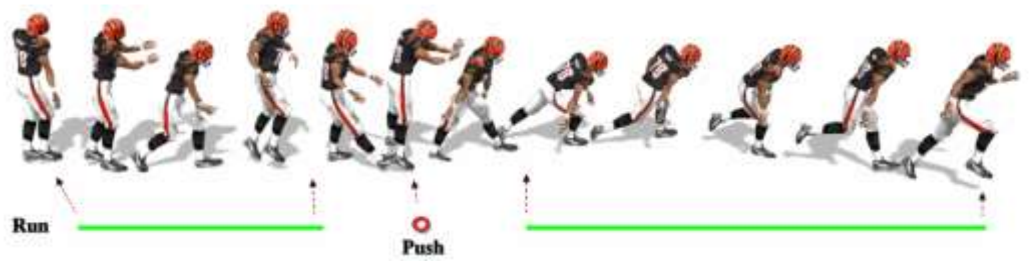


Figure 3 : Example motion clips

Figure 4 : Example motion clips – Rotating Ballerina

Identify similar clips:

But do vector norms work?

Problems:

- Fail to account for meanings of parameters
- Motion defined only up to a rigid 2D co-ordinate transformation

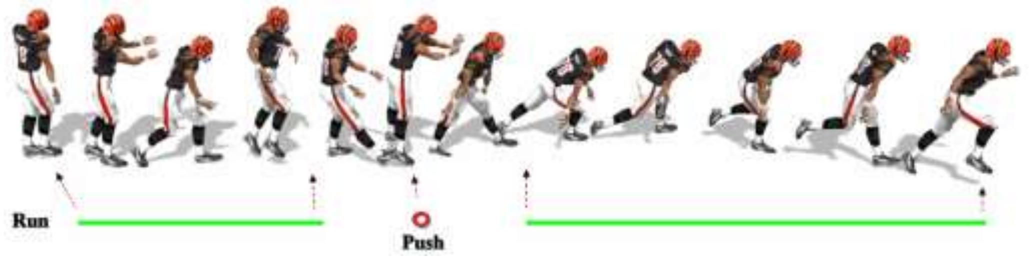


Figure 3 : Example motion clips

Figure 4: Example motion clips – Rotating Ballerina

Identify similar clips:

But do vector norms work?

Problems:

- Fail to account for meanings of parameters
- Motion defined only up to a rigid 2D co-ordinate transformation
- No information about joint velocities, accelerations

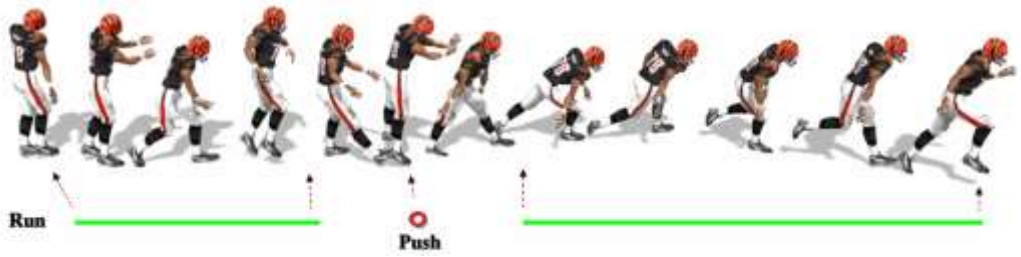


Figure 3: Example motion clips

Figure 4: Example motion clips – Rotating Ballerina

Metric driven by point cloud

Figure 5: Point cloud created by motion

Metric driven by point cloud

- Use window of frames- create point cloud
- Windows before and after the frames to be compared(Ai and Bj)
- Measure closeness of frames of animation
- Primary advantage : Incorporates derivative information
- User-defined 'k'

Figure 6: Point cloud created by motion

Figure 7 : Two example motion clips

Error Metric:

Distance between two frames Ai,Bj

$$D(A_i, B_i) = \min \sum_i w_i \; || p_i - T_{\theta, x_{0,} z_0} p_i' ||^2$$

Where,

- p_i and p_i' = points in the two point clouds
- w_i = weights
- T () = linear transformation matrix
- I = index in point cloud

Detecting candidate transitions:

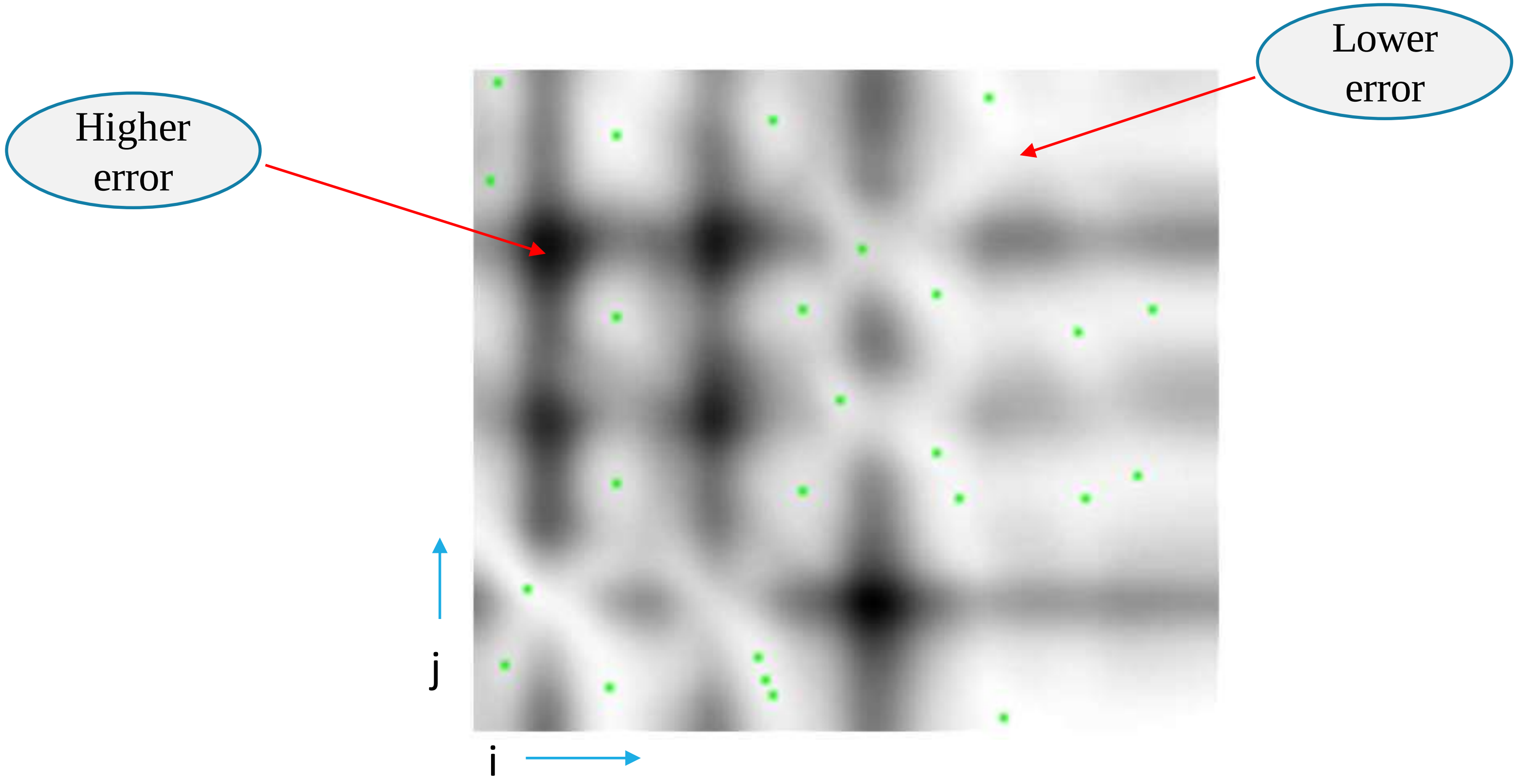


Figure 8 : Example error function for two motions

Detecting candidate transitions:

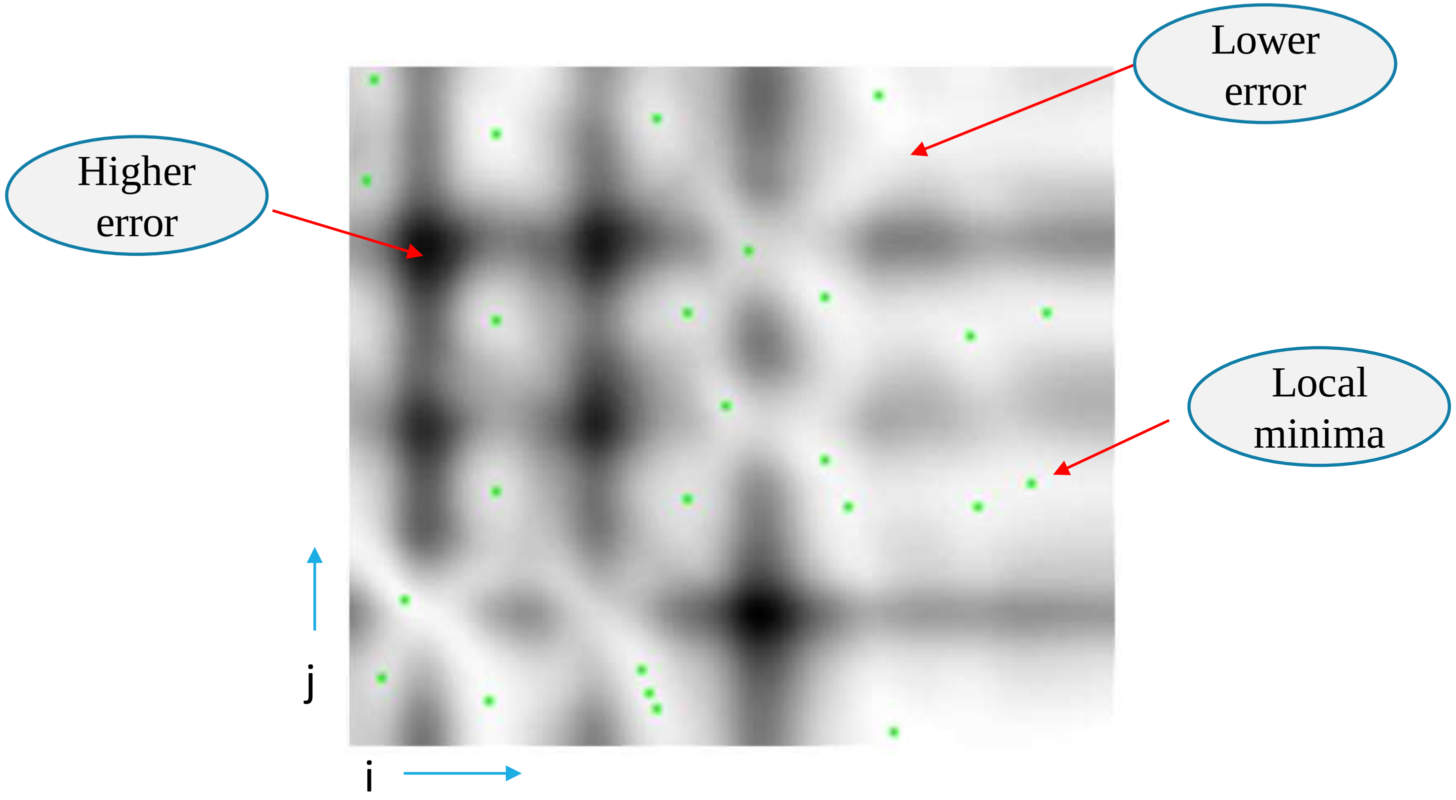


Figure 9 : Example error function for two motions

- Local minima ➡ Most opportune transition
- Choose based on a threshold
- User likes to pick this threshold

Selecting transition points

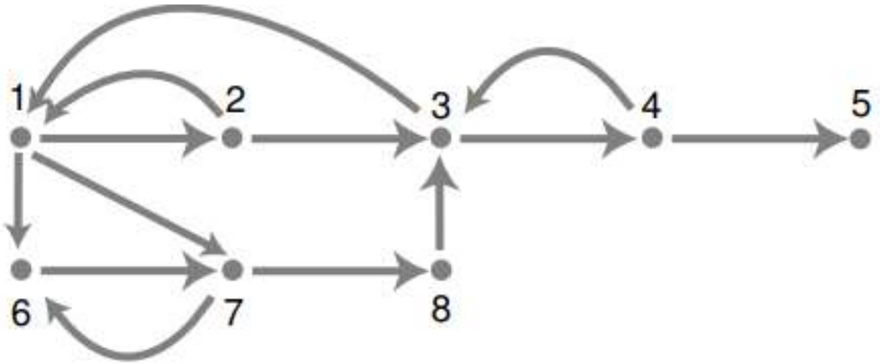
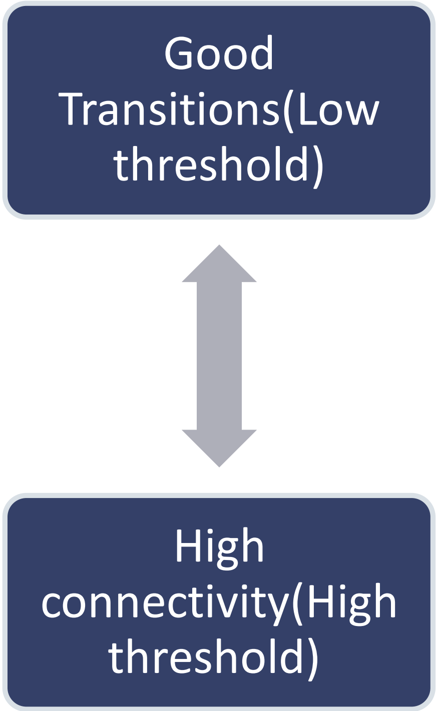


Figure 10 : A simple motion graph

Figure 11: Two example motion clips



Creating transitions : If D(A_i, B_j) meets threshold

- Blend two window of frames
- Ai to Ai+k-1with frames Bj-k+1 to Bj
- Simple linear interpolation on root positions
- Spherical interpolation on joint rotations

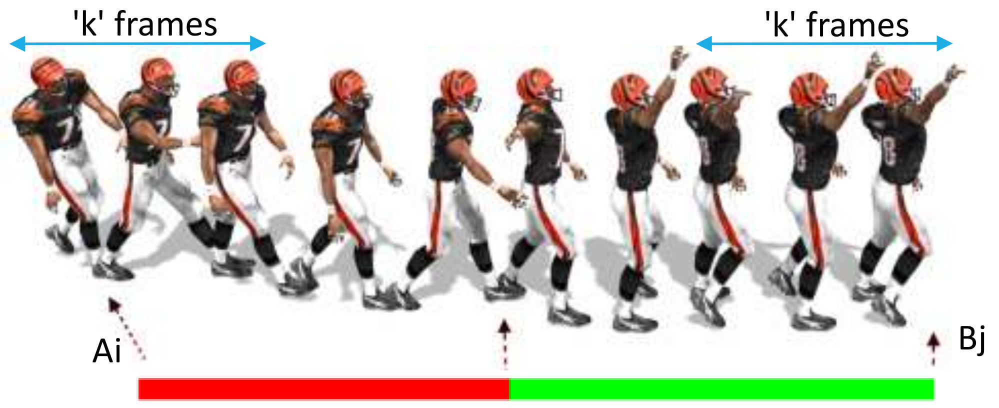


Figure 12: Two example motion clips

Creating transitions : If D(A_i, B_j) meets threshold

Steps:

1. Aligning 2D transformation to motion B
2. On frame p of transition(0≤p<k),

$$R_p = \alpha(p) R_{A_{i+p}} + [1 - \alpha(p)]\, R_{B_{j-k-1}}$$

$$q_p^i = slerp(q^i_{A_{i+p}}, q^i_{B_{j-k+1+p}}, \alpha(p))$$

- where,
  - R_p = Root position on frame *p* of transition
  - q_p^i = Rotation of the i^{th} joint P^{th}
  - R_{A_i} = Root position on frame A_i in motion A
  - R_{B_i} = Root position on frame B_i in motion B

Prevent feet sliding !!!

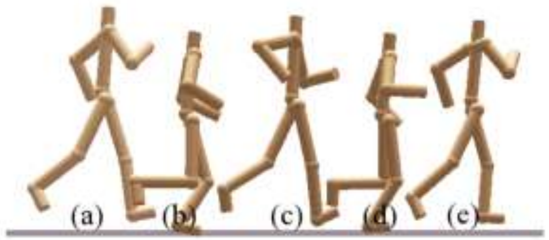


Figure 13: Example motion clip-Human walking

Creating transitions : If D(A_i, B_j) meets threshold

Steps:

1. Aligning 2D transformation to motion B
2. On frame p of transition(0≤p<k), linearly interpolate the root positions and perform spherical linear interpolation on joint rotations
3. Blend weight:

$$\alpha(p) = 2\left(\frac{p+1}{k}\right)^3 - 3\left(\frac{p+1}{k}\right)^2 + 1,\ -1 < p < k$$

- Conditions :
  - $\alpha(p) = 1, p \leq -1$
  - $\alpha(p) = 0, p \geq \mathrm{k}$
  - C1 continuity

Motion graphs

- Arrange clips in directed graph
- Key components:
  - **Dead ends**: not part of any cycle
  - **Sinks**: low connectivity
  - **Strongly connected components**: nodes part of many cycles
- Incoming label doesn't match outgoing label

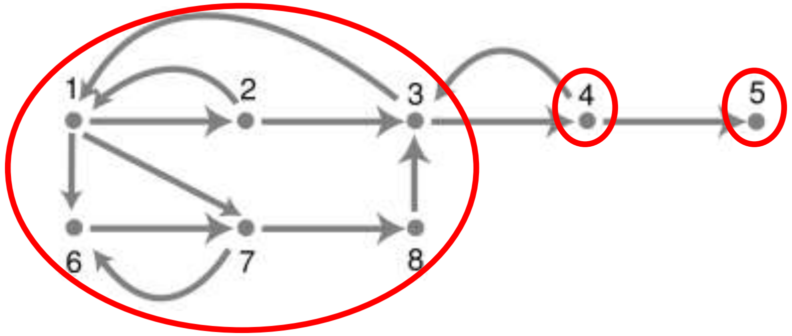


Figure 14 : A simple motion graph

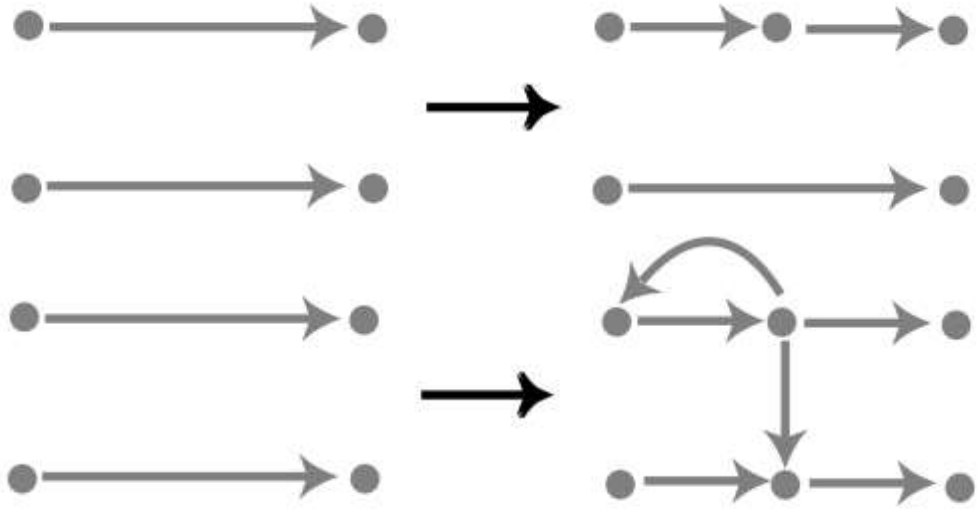


Figure 15 : Motion graphs built from initial clips

Motion graphs

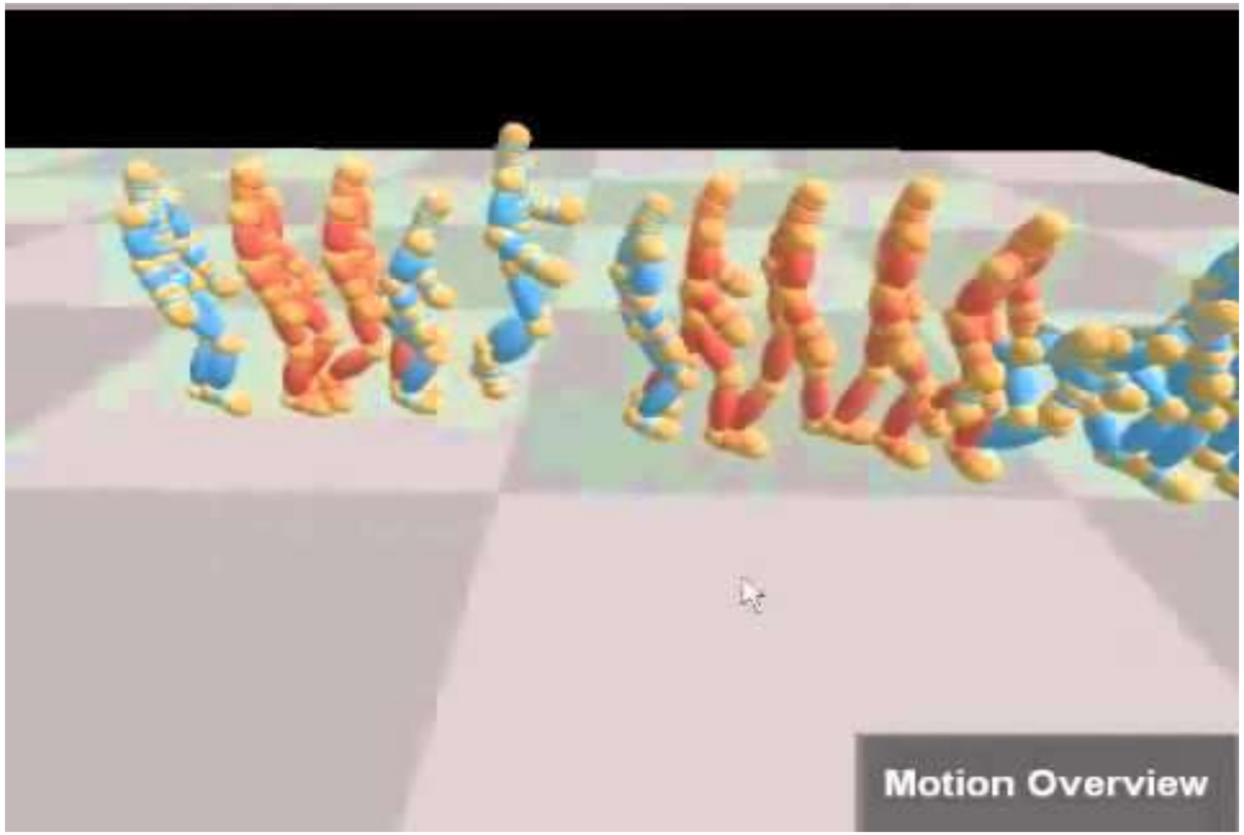


Figure 15 : Example of motion synthesis with interpolated frames

Motion extraction:

- Searching for motion ⟷ Optimizing graph walks
- Suitable metric to extract frames
- Metric used is Scalar error function : g(w,e) - Conforming to user specifications
- Total error :

$$f(w) = f([e_1, e_2, \ldots.., e_n]) = \sum_{i=1}^{n} g([e_1, e_2, \ldots.., e_n]), e_i$$

- Branch and bound to reduce number of graph walks

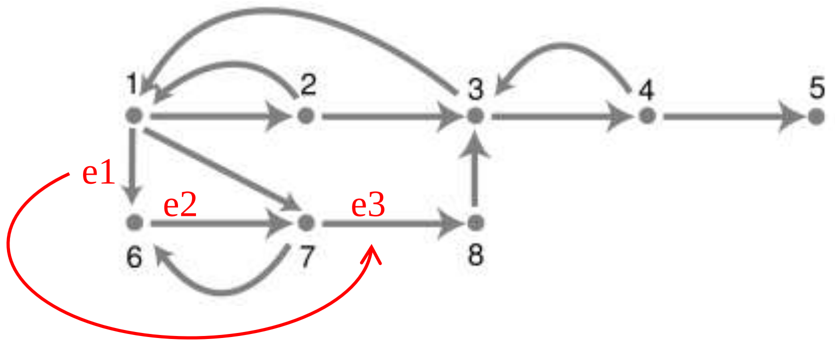


Figure 16: A simple motion graph

Motion extracted depends on 'g' !

Given starting node *N*, target edge *e,* and a transformation T'

Graph walk presenting the same result

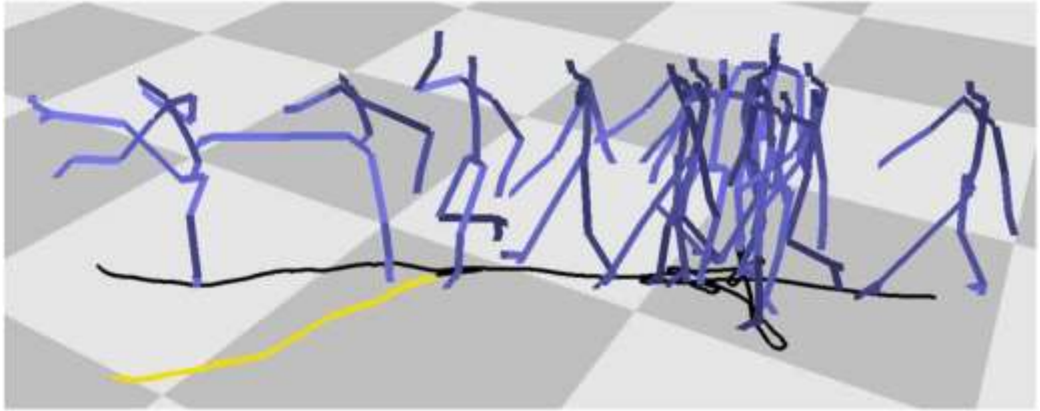


Figure 17: A generated motion using motion graph with position and orientation difference as a metric

- Key requirements in *g:*
  - Guidance through motion
  - Not be too restrictive

Path Synthesis

- Error metric – Distance between points in path
  - Given path – P and actual path P'

$$g(w,e) = \sum_{i=1}^{n} ||P'(s(e_i)) - P(s(e_i))||^2$$

  - Where,
    - i – frames in edge
    - e_i – i^{th} frame of edge
    - $s(e_i)$ - arc length at e_i
  - Halting condition: current total length of P' ≥ length of P

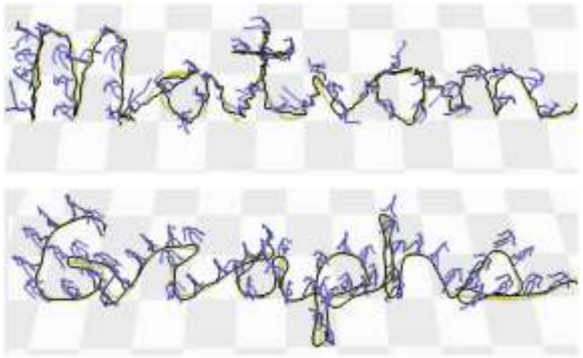


Figure 18: A generated motion using path synthesis algorithm

Character accumulates zero error by standing still

Path synthesis- with descriptive labels

- All generated motion of single type
- Confine search to relevant sub graphs
- Mixed labels on parts of path : Distinct labels for adjoining parts of path

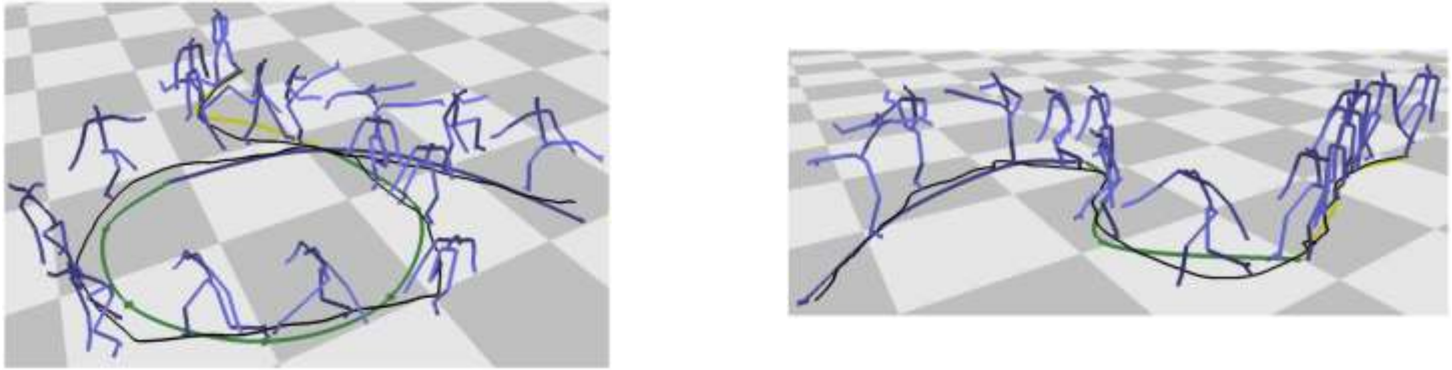


Figure 19: Motions generated with different labels on different parts of the path

Final Synthesized motion

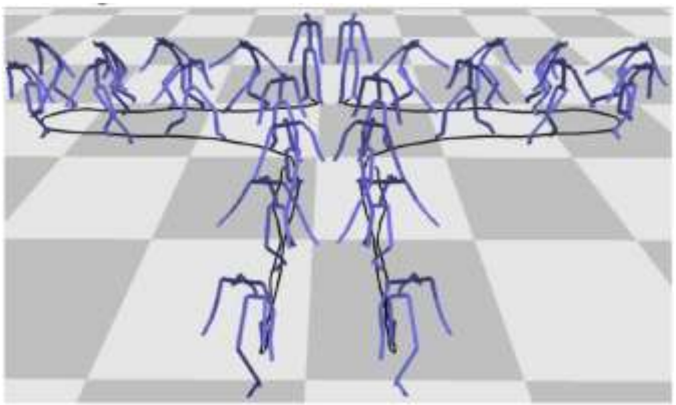


Figure 20 : Original motion clip and its reflection

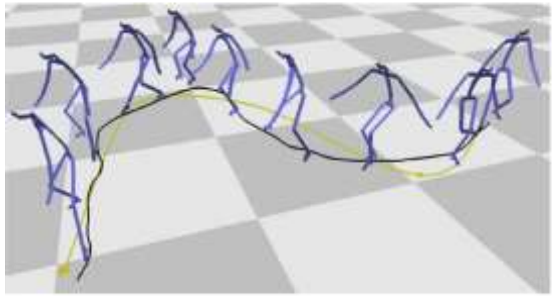


Figure 21 : Motions generated with path synthesis algorithm

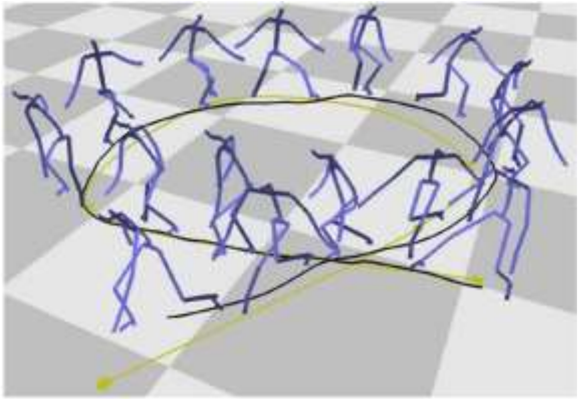


Figure 22: Motions generated with path synthesis algorithm

Final Synthesized motion

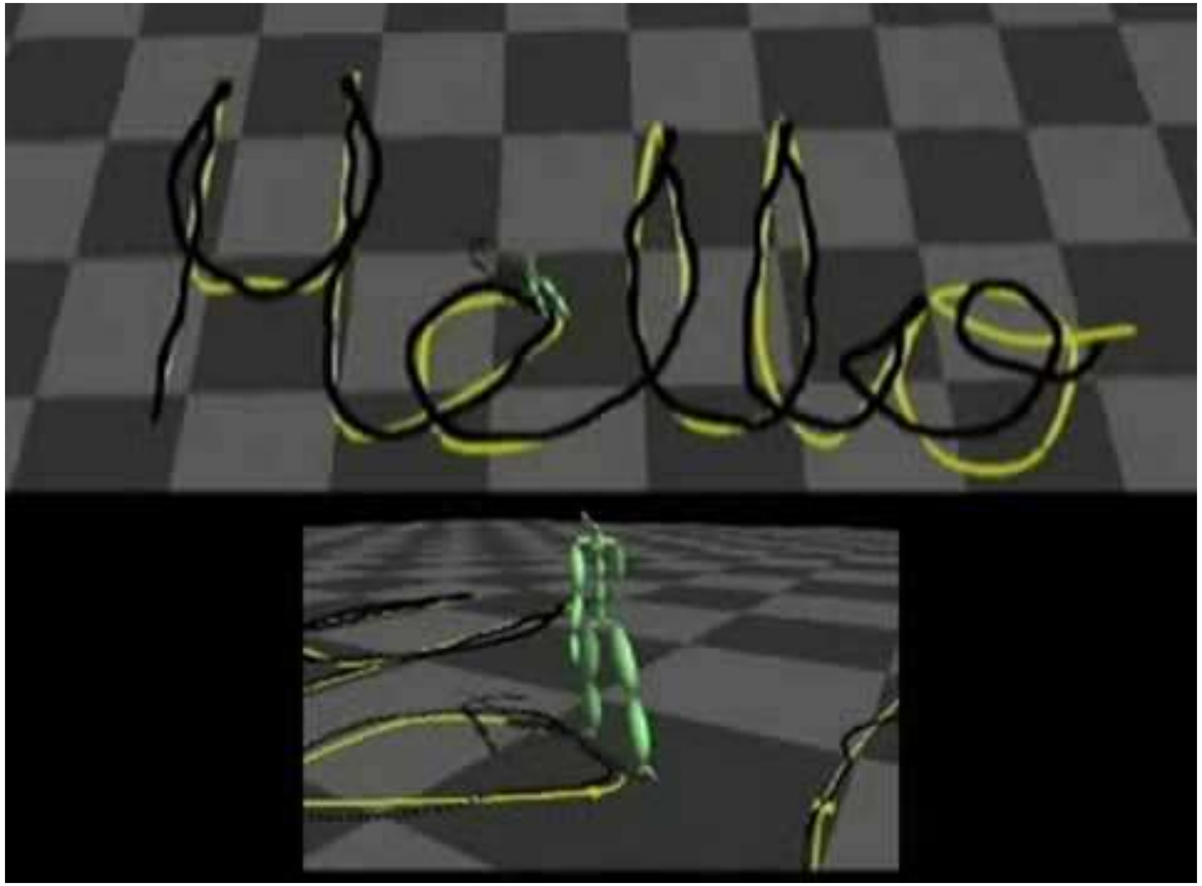


Figure 23 : Results: Motion generated with Kovar et al.. 2002 algorithm

Applications

- **Interactive control:** User control over character
- **High-Level Key Framing:** define subsections with required action types
- **Motion Dumping:** Animate non-player characters and interactive environments
- **Crowds:** Practical tool for crowd generation

Figure 27 : Interactive motion synthesis

Figure 24 : Randomly walking crowd

Figure 25 : Motion dumping in games

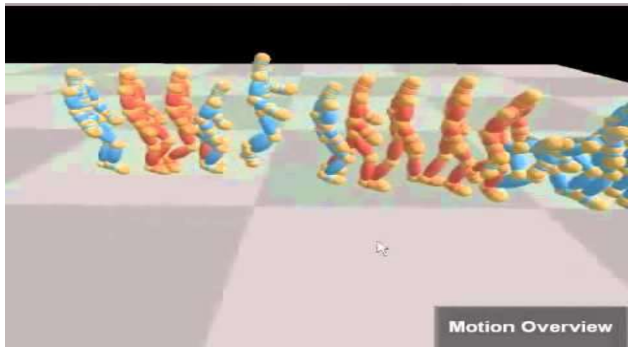
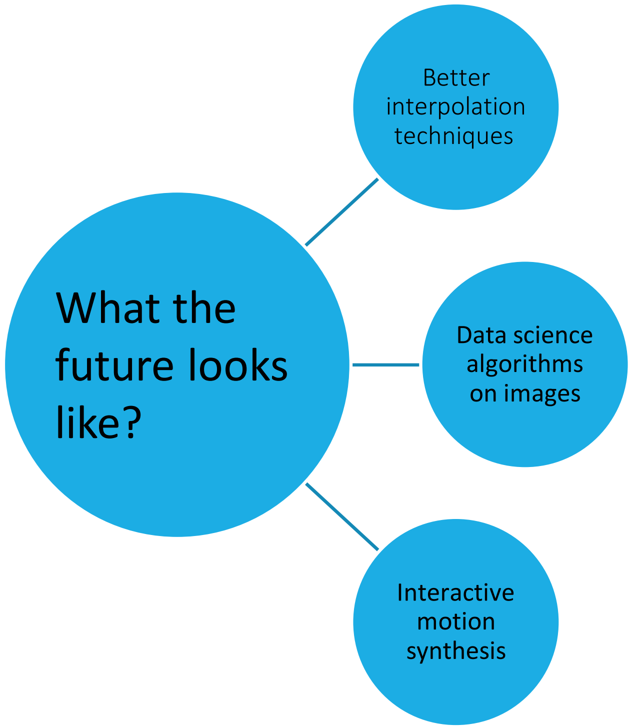


Figure 26 : High level key framing

Conclusion and Future Scope

- Framework for generating realistic, controllable motion
- Encapsulate connections in the database automatically
- Comparing every pair of F frames involves $O(F^2)$
- Major limitation here : Thresholds are user-specified



UNIVERSITÄT DES SAARLANDES

Thank you !

References

1. Kovar, Lucas, Michael Gleicher, and Frédéric Pighin. "Motion graphs." ACM SIGGRAPH 2008 classes. ACM, 2008
2. Arikan, Okan, David A. Forsyth, and James F. O'Brien. "Motion synthesis from annotations." *ACM Transactions on Graphics (TOG)*. Vol. 22. No. 3. ACM, 2003.
3. John Fisher. Green Screen Crowd of People Walking. Youtube. 2018. URL: https://www.youtube.com/watch?v=zoPCD2C5BiE.
4. Kovar, Lucas, Michael Gleicher, and Frédéric Pighin. Motion Graphs - Path Fitting. Youtube. 2008. URL: https://www.youtube.com/watch?v=UxG6U0jeOu8.
5. Liming Zhao Alla Safonova. Achieving Good Connectivity in Motion Graphs. Youtube. 2011. URL: https://www.youtube.com/watch?v=XTetbc0Q8Dc.
6. GameSprout. FIFA 16 FAIL Compilation. Youtube. 2015. URL: https://www.youtube.com/watch?v=F4MBcGrUdOk&t=16s
7. Rachel Heck, and Michael Gleicher. Parametric Motion Graph. URL: https://www.youtube.com/watch?v=ZepNQHXAcCY&t=7s

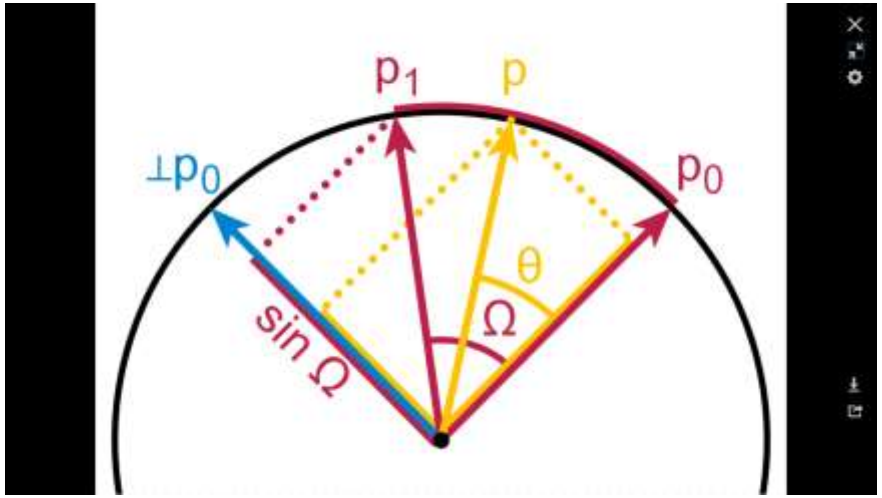
A1. Modified parametric equation in path synthesis

- Character standing still has low error
- Require atleast small progress on each frame
- Use $t(e_i)$ instead of s(e_i)

$$t(e_i) = \max(t(e_{i-1}) + s(e_i) - s(e_{i-1}), t(e_{i-1}) + \gamma_i$$

A2. Slerp

- Shorthand for spherical linear interpolation
- Rotation with uniform angular velocity around a fixed rotation axis



A2. Optimised solution of the metric

$$\theta = \arctan \frac{\sum_i w_i (x_i z'_i - x'_i z_i) - \frac{1}{\sum_i w_i} (\overline{x}\overline{z'} - \overline{x'}\overline{z})}{\sum_i w_i (x_i x'_i + z_i z'_i) - \frac{1}{\sum_i w_i} (\overline{x}\overline{x'} + \overline{z}\overline{z'})}$$

$$x_0 = \frac{1}{\sum_i w_i} (\overline{x} - \overline{x'} \cos(\theta) - \overline{z'} \sin\theta)$$

$$z_0 = \frac{1}{\sum_i w_i} (\overline{z} + \overline{x'} \sin(\theta) - \overline{z'} \cos\theta)$$

where $\overline{x} = \sum_i w_i x_i$ and the other barred terms are defined similarly.