

Video based Animation Synthesis with the Essential Graph

Adnane Boukhayma, Edmond Boyer
MORPHEO INRIA Grenoble Rhône-Alpes

Goal

Given a set of 4D models, how to generate realistic motion from user specified constraints ?

Input:



Output:



- Video based 4D models of elementary movements

- Novel, user guided shape and appearance animation

Motivation

Human animation generation:

Where:

- Digital media production :
 - Video Game industry
 - Motion Picture industry
 - Virtual Reality applications



Rise of the Tomb Raider, Cristal Dynamics



Game Of Thrones, HBO

Motivation

Human animation generation:

Where:

- Digital media production :
 - Video Game industry
 - Motion Picture industry
 - Virtual Reality applications

How:

- Physical modeling:
 - Computationally expensive
 - Model limitations



Rise of the Tomb Raider, Cristal Dynamics



Game Of Thrones, HBO

Motivation

Human animation generation:

Where:

- Digital media production :
 - Video Game industry
 - Motion Picture industry
 - Virtual Reality applications



Rise of the Tomb Raider, Cristal Dynamics



Game Of Thrones, HBO

How:

- Physical modeling:
 - Computationally expensive
 - Model limitations
- Example data Reuse:
 - Key-framing data:
 - Cost-wise expensive: 100-250\$ / character second
 - Time-wise expensive: 2-3 seconds of finished character animation per day

Motivation

Human animation generation:

Where:

- Digital media production :
 - Video Game industry
 - Motion Picture industry
 - Virtual Reality applications



Rise of the Tomb Raider, Cristal Dynamics



Game Of Thrones, HBO

How:

- Physical modeling:
 - Computationally expensive
 - Model limitations
- Example data Reuse:
 - Key-framing data:
 - Cost-wise expensive: 100-250\$ / character second
 - Time-wise expensive: 2-3 seconds of finished character animation per day

Synthetic Motion

Motivation

Human animation generation:

Where:

- Digital media production :
 - Video Game industry
 - Motion Picture industry
 - Virtual Reality applications



Rise of the Tomb Raider, Cristal Dynamics

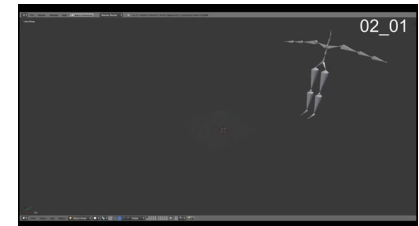


Game Of Thrones, HBO

How:

- Physical modeling:
 - Computationally expensive
 - Model limitations
- Example data Reuse:
 - Key-framing data:
 - Cost-wise expensive: 100-250\$ / character second
 - Time-wise expensive: 2-3 seconds of finished character animation per day
 - Motion capture data:
 - ✓ Real Motion

Synthetic Motion



CMU mocap dataset

Motivation

Human animation generation:

Where:

- Digital media production :
 - Video Game industry
 - Motion Picture industry
 - Virtual Reality applications



Rise of the Tomb Raider, Cristal Dynamics



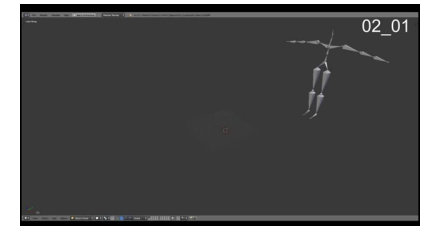
Game Of Thrones, HBO

How:

- Physical modeling:
 - Computationally expensive
 - Model limitations
- Example data Reuse:
 - Key-framing data:
 - Cost-wise expensive: 100-250\$ / character second
 - Time-wise expensive: 2-3 seconds of finished character animation per day
 - Motion capture data:
 - ✓ Real Motion

Synthetic Motion

Synthetic Shape



CMU mocap dataset



3D character animated with mocap

Motivation

Human animation generation:

Where:

- Digital media production :
 - Video Game industry
 - Motion Picture industry
 - Virtual Reality applications



Rise of the Tomb Raider, Cristal Dynamics

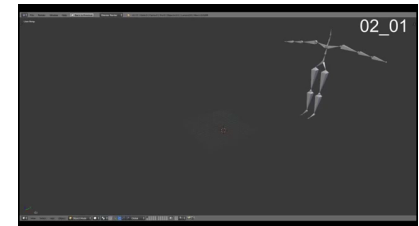


Game Of Thrones, HBO

How:

- Physical modeling:
 - Computationally expensive
 - Model limitations
- Example data Reuse:
 - Key-framing data:
 - Cost-wise expensive: 100-250\$ / character second
 - Time-wise expensive: 2-3 seconds of finished character animation per day

Synthetic Motion



CMU mocap dataset

- Motion capture data:
 - ✓ Real Motion
- **Surface capture data:**
 - ✓ Real Motion
 - ✓ Real Shape
 - ✓ Real Appearance

Synthetic Shape

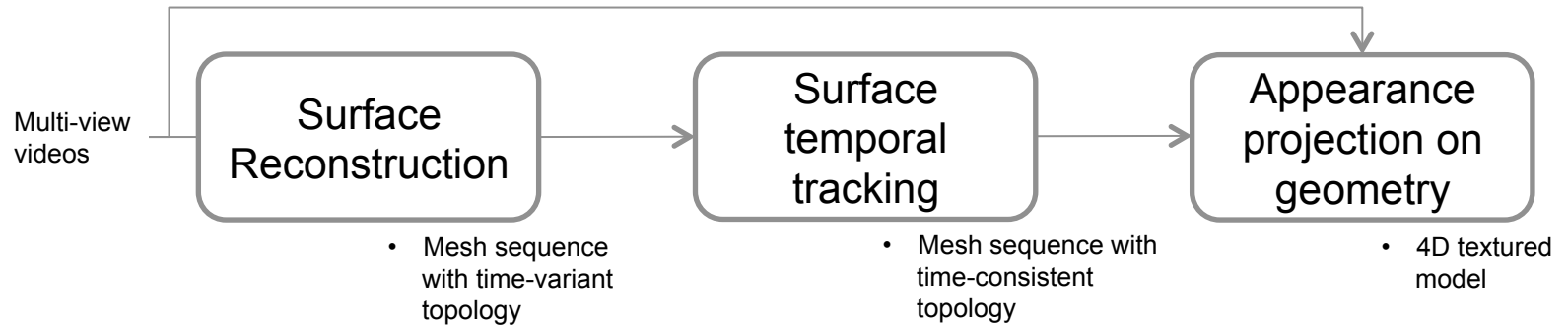


4D model, Thomas dataset

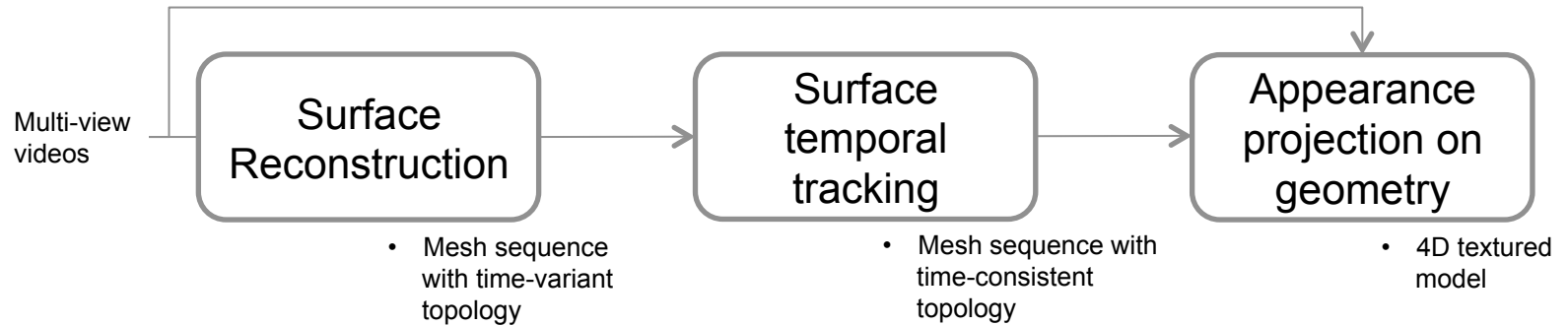


3D character animated with mocap

Surface Capture Reuse



Surface Capture Reuse



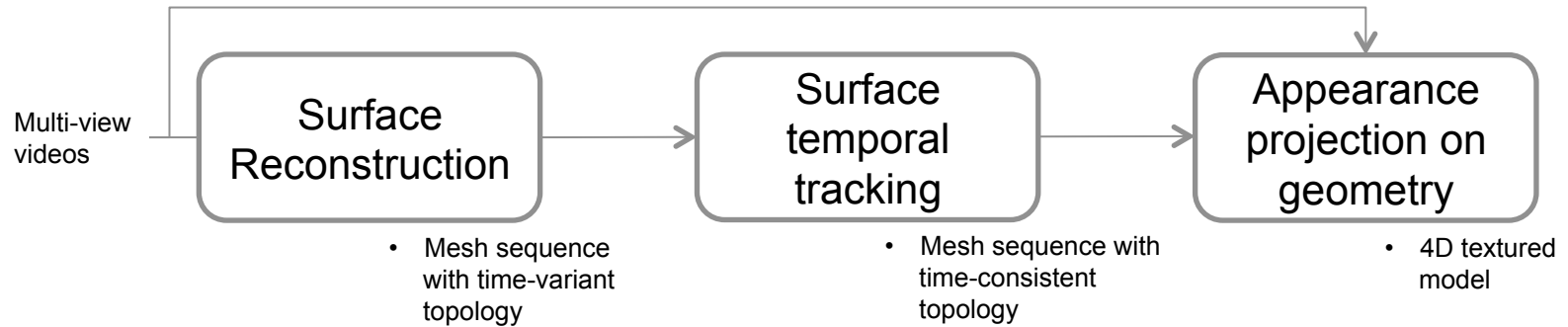
Data:

- Cyclic human movements: walk, run, jump, etc.
- Acyclic human movements: dance.



Thomas, Cathy

Surface Capture Reuse



Data:

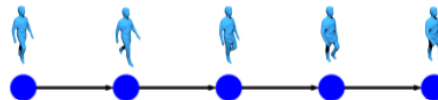
- Cyclic human movements: walk, run, jump, etc.
- Acyclic human movements: dance.



Data Reuse:

Generate continuous motion stream using **basic operations on motion segments:**

- Rigid transformations
- Concatenation
- Smooth transition generation



Thomas, Cathy

Issues and challenges

User control:

- Intuitive formulation of user defined constraints

Data organization:

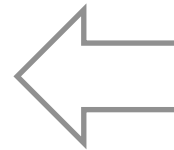
- A data structure organizing the input sequences and encoding selected transitions between them

Motion synthesis:

- Generating synthetic motion transitions.
- Concatenating real and synthetic motion segments.

Numerical realism criterion:

- Transition evaluation



Issues and challenges

User control:

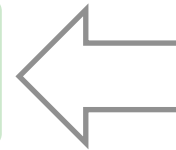
- Intuitive formulation of user defined constraints

Data organization:

- A data structure organizing the input sequences and encoding selected transitions between them

Motion synthesis:

- Generating synthetic motion transitions.
- Concatenating real and synthetic motion segments.



Numerical realism criterion:

- Transition evaluation

Challenges :

Limited data:

- Make exhaustive use of it

User perceptual acuity:

- Reliable numerical realism criterion
- Optimal results in terms of said criterion

Sensitive data:

- Robust mesh processing technique

Complex dynamics:

- Robust transition generation technique

Comparison of Graph Based Approaches

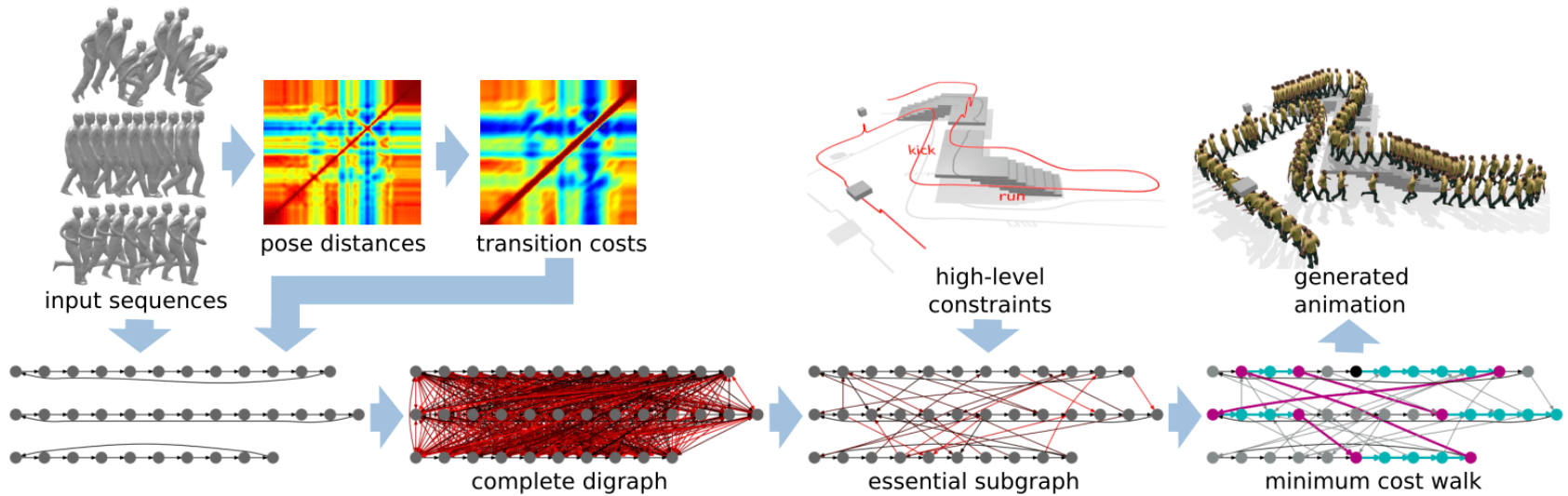
	Input data	Automatic/ supervised	Data organization
Motion Graph(kovar02)(Arikan02)	Motion Capture	automatic	Motion graph
Surface Motion Graph(Huang09)	3D Surface Capture	automatic	Motion graph
4D Parametric Motion Graph(Casas13)	4D Surface Capture	supervised	Parametric motion graph
Essential Graph(Boukhayma et. Boyer15)	4D Surface Capture	automatic	Essential graph

Contributions

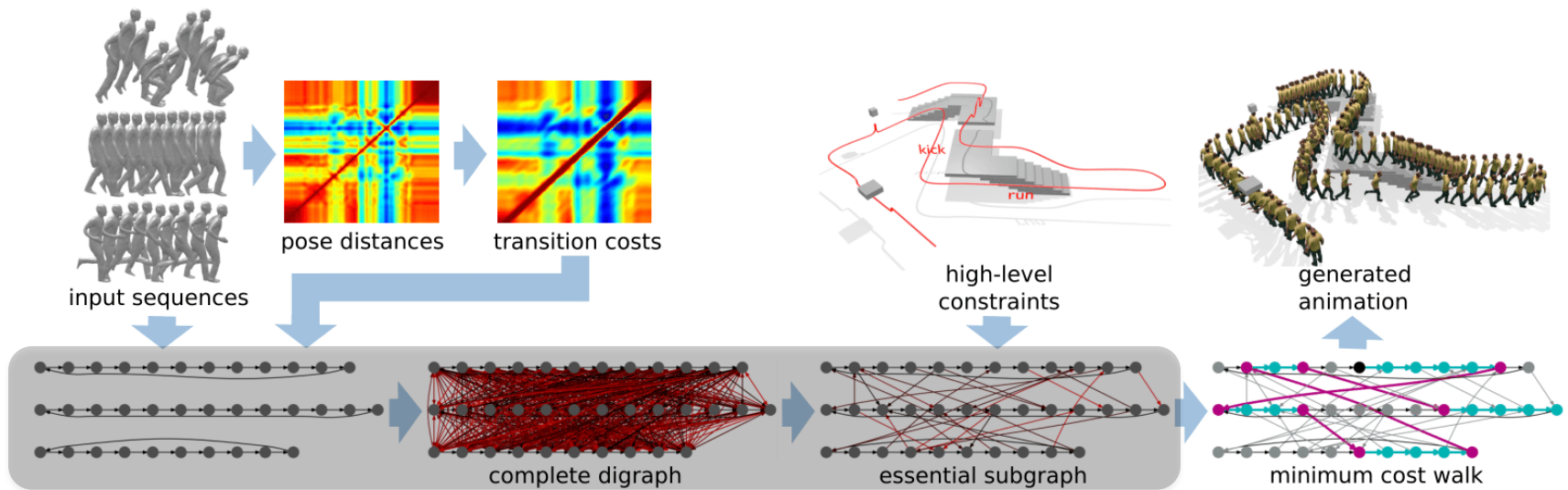
- An optimal structure for motion data organization and reuse:
The essential graph
- Improving realism in synthetic motion transitions through **dynamic time warping and variable length blended segments**
- A novel high-level constraint formulation for motion synthesis:
3D behavioral path synthesis

Approach

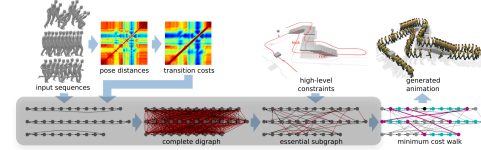
Motion synthesis pipeline :



Input Data Organization

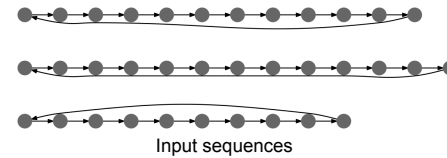


Input Data Organization

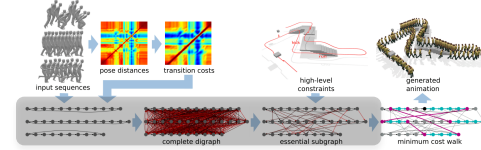


Graph structure:

- Node = frame/pose
- Edge = transition
- Edge weight = transition cost

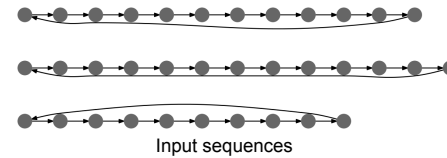


Input Data Organization

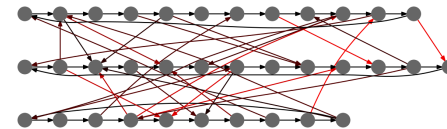


Graph structure:

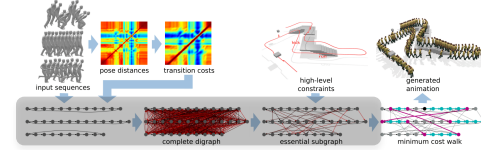
- Node = frame/pose
- Edge = transition
- Edge weight = transition cost



We need to add new transitions :



Input Data Organization

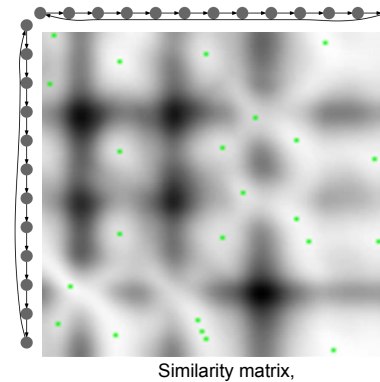
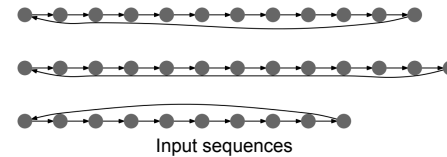


Graph structure:

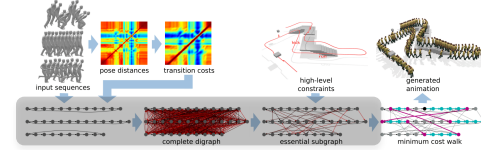
- Node = frame/pose
- Edge = transition
- Edge weight = transition cost

Motion Graph(kovar02)

- Adding edges
 - Local minima in similarity matrix
 - Threshloding



Input Data Organization

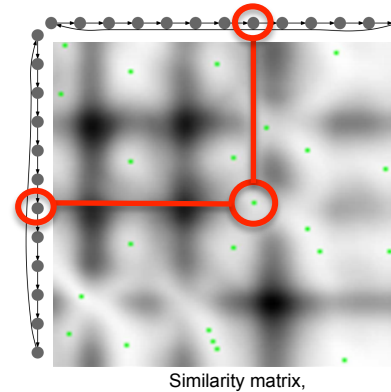
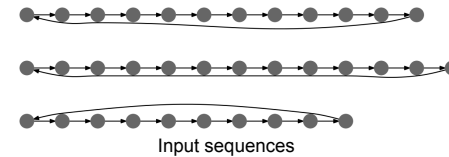


Graph structure:

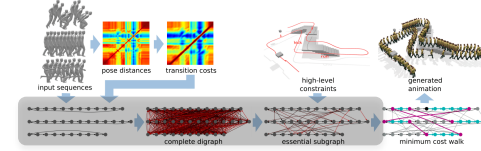
- Node = frame/pose
- Edge = transition
- Edge weight = transition cost

Motion Graph(kovar02)

- Adding edges
 - Local minima in similarity matrix
 - Threshloding



Input Data Organization

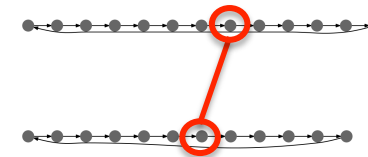
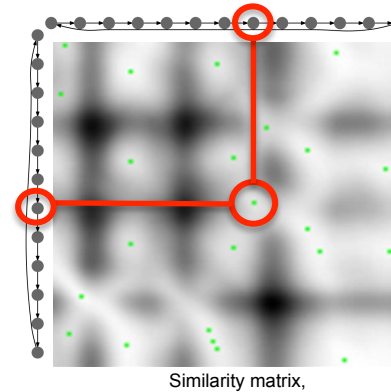
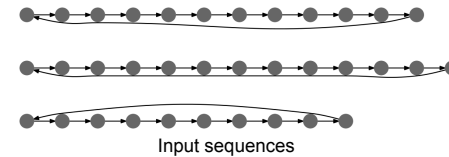


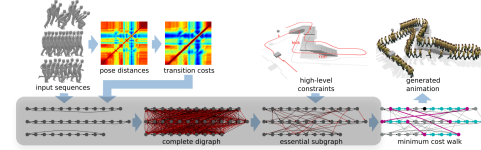
Graph structure:

- Node = frame/pose
- Edge = transition
- Edge weight = transition cost

Motion Graph(kovar02)

- Adding edges
 - Local minima in similarity matrix
 - Threshloding





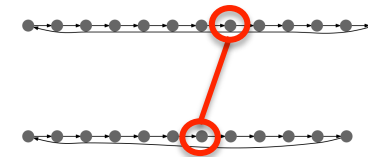
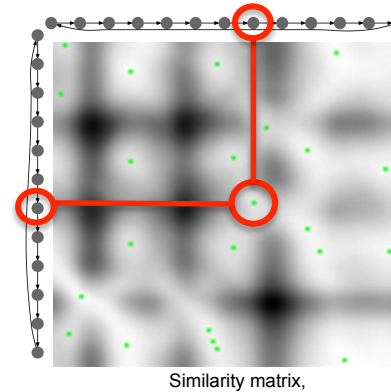
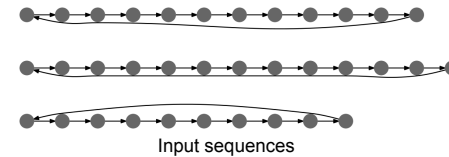
Input Data Organization

Graph structure:

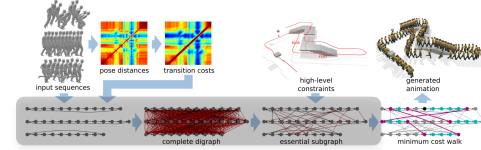
- Node = frame/pose
- Edge = transition
- Edge weight = transition cost

Motion Graph(kovar02)

- Adding edges
 - Local minima in similarity matrix
 - Threshloding
- Improvements
 - Interpolated Motion Graph(Sofanova07)
 - Well-Connected Motion Graph(Zhao08)
 - Optimization-based Motion Graph(Ren10)

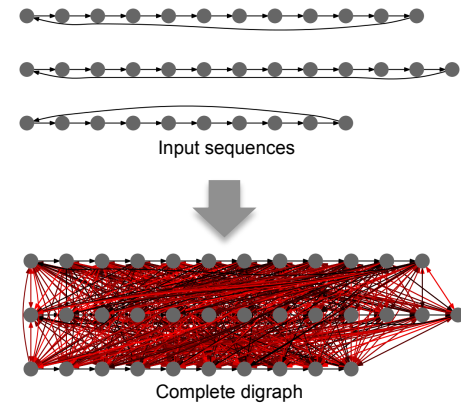


Essential graph

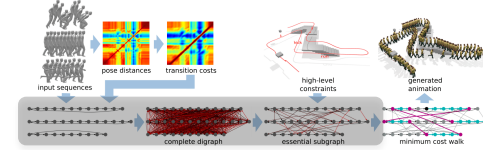


Essential Graph

- Creating a complete digraph
 - Connecting all nodes together with directed edges and transition costs as weights

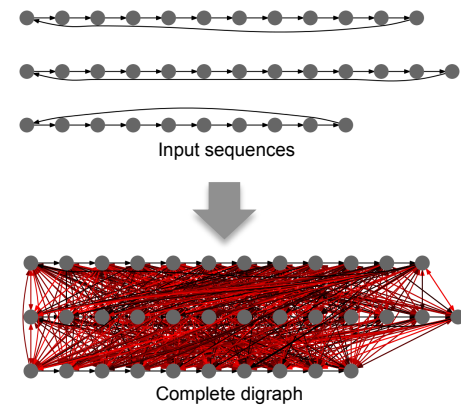


Essential graph

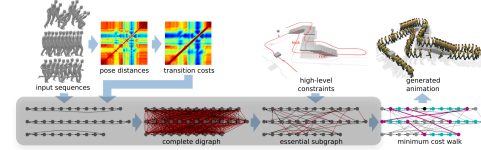


Essential Graph

- Creating a complete digraph
 - Connecting all nodes together with directed edges and transition costs as weights
- Extracting the essential sub-graph
 - For each node
 - Calculate the **shortest path tree** rooted at said node



Essential graph

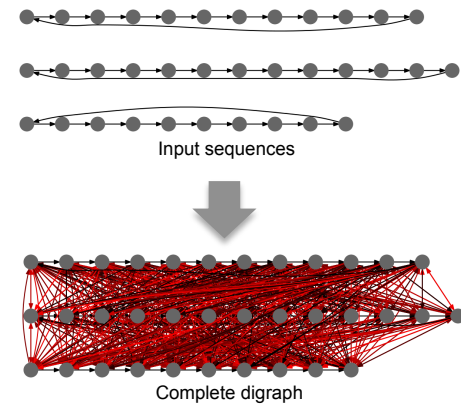


Essential Graph

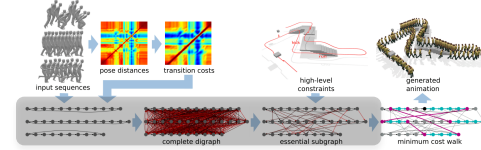
- Creating a complete digraph
 - Connecting all nodes together with directed edges and transition costs as weights
- Extracting the essential sub-graph
 - For each node
 - Calculate the **shortest path tree** rooted at said node

shortest path tree:

the path distance from the root to any vertex in the tree is the shortest path from the root to the vertex in the complete digraph



Essential graph



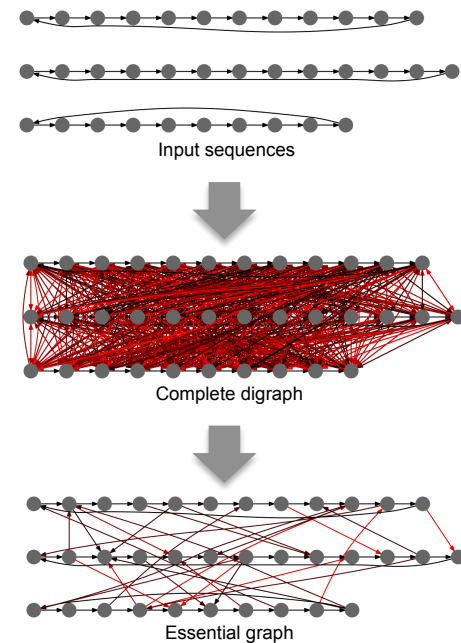
Essential Graph

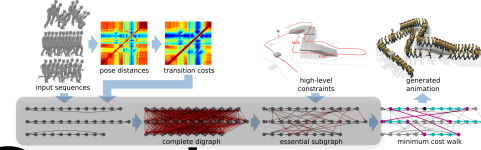
- Creating a complete digraph
 - Connecting all nodes together with directed edges and transition costs as weights
- Extracting the essential sub-graph
 - For each node
 - Calculate the **shortest path tree** rooted at said node

shortest path tree:

the path distance from the root to any vertex in the tree is the shortest path from the root to the vertex in the complete digraph

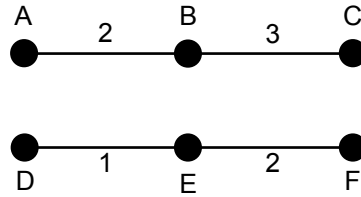
- Union of shortest path trees rooted at every node
- This structure encodes all the optimal transitions with respect to the transition cost



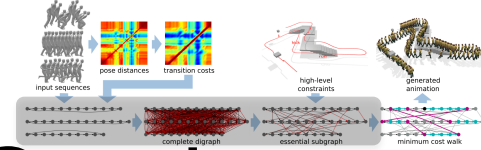


Motion Graph vs Essential Graph

Comparative case study:

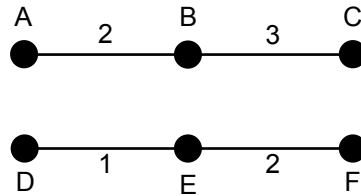


	D	E	F
A	7	5	5
B	7	2	5
C	7	4	3

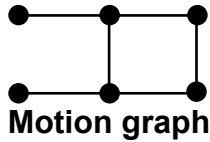


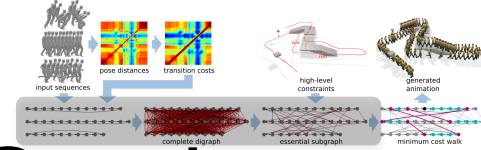
Motion Graph vs Essential Graph

Comparative case study:



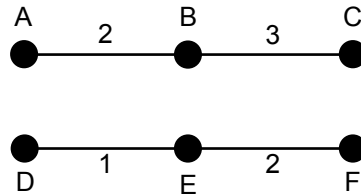
	D	E	F
A	7	5	5
B	7	2	5
C	7	4	3



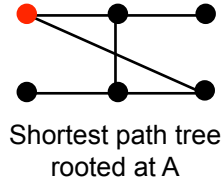
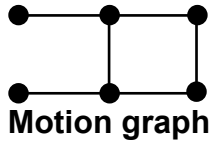


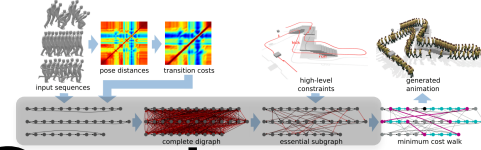
Motion Graph vs Essential Graph

Comparative case study:



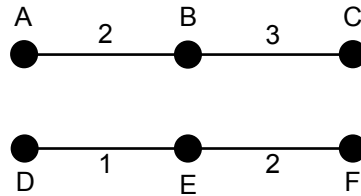
	D	E	F
A	7	5	5
B	7	2	5
C	7	4	3



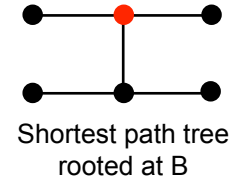
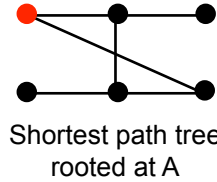
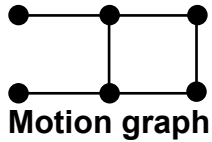


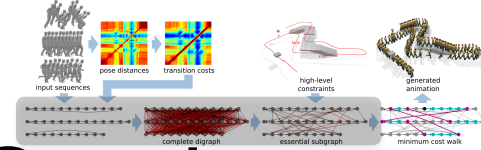
Motion Graph vs Essential Graph

Comparative case study:



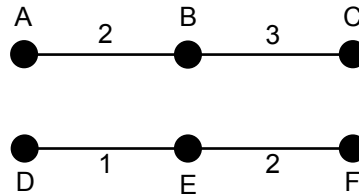
	D	E	F
A	7	5	5
B	7	2	5
C	7	4	3



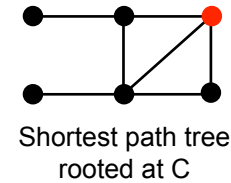
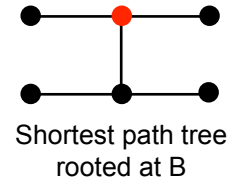
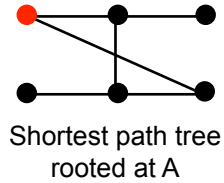
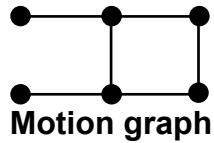


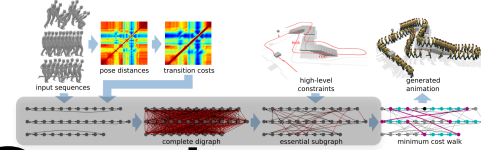
Motion Graph vs Essential Graph

Comparative case study:



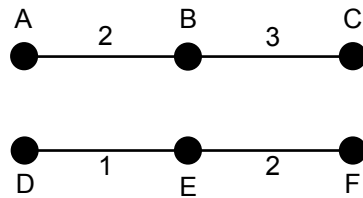
	D	E	F
A	7	5	5
B	7	2	5
C	7	4	3



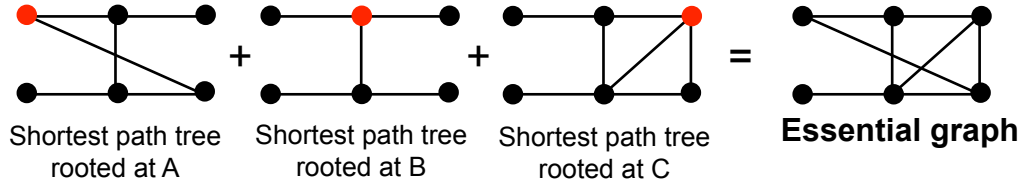
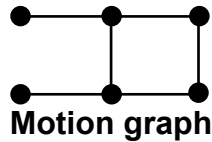


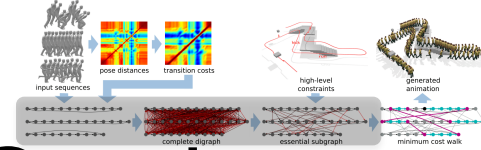
Motion Graph vs Essential Graph

Comparative case study:



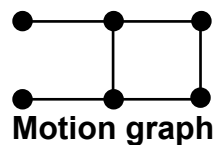
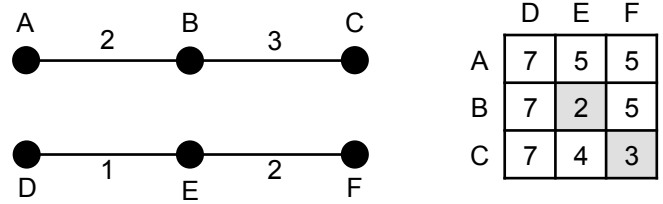
	D	E	F
A	7	5	5
B	7	2	5
C	7	4	3



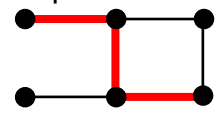


Motion Graph vs Essential Graph

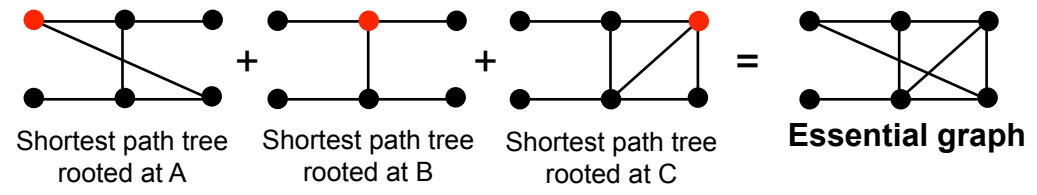
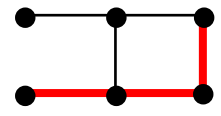
Comparative case study:

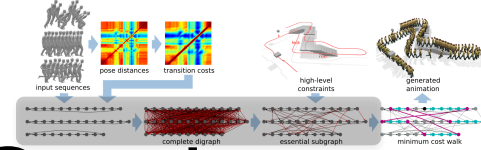


Shortest path from A to F: **6**



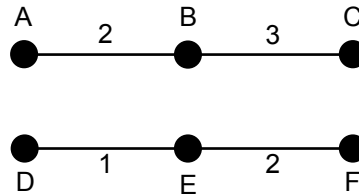
Shortest path from D to C: **6**



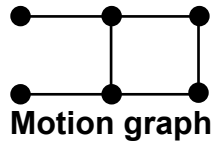


Motion Graph vs Essential Graph

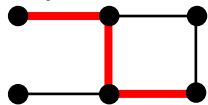
Comparative case study:



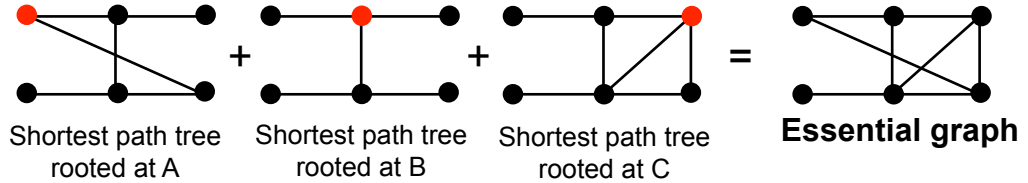
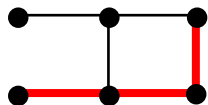
	D	E	F
A	7	5	5
B	7	2	5
C	7	4	3



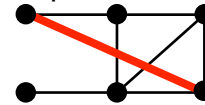
Shortest path from A to F: **6**



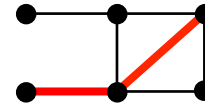
Shortest path from D to C: **6**



Shortest path from A to F: **4**



Shortest path from E to C: **5**



Input Data Organization

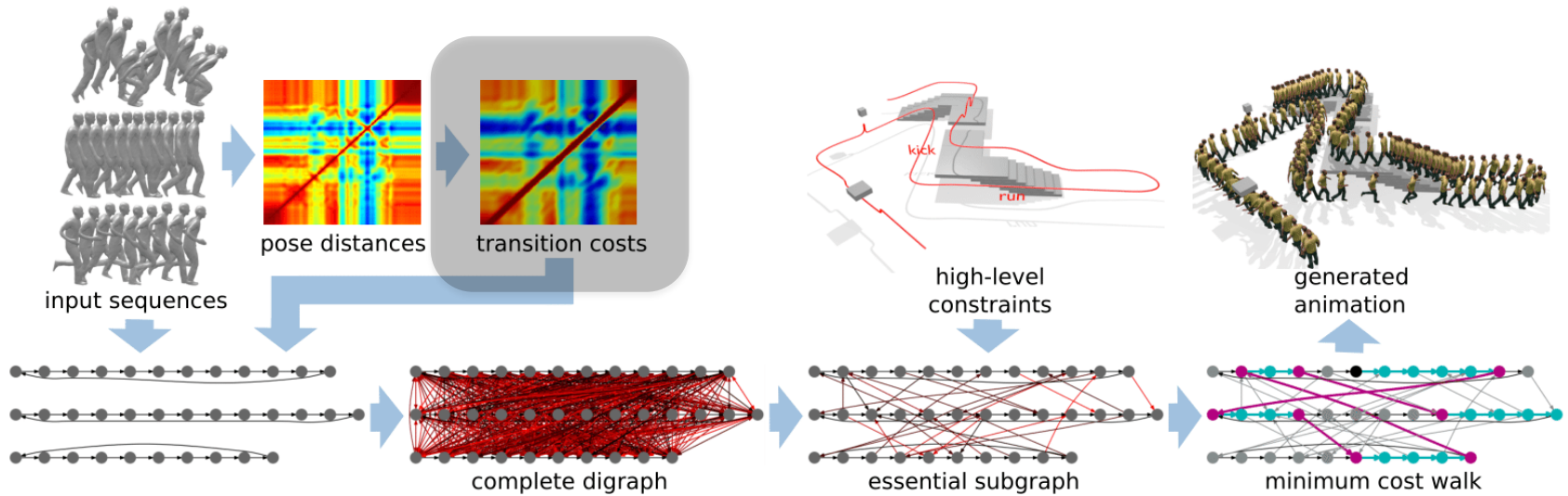


Original frames
Synthetic frames

Thomas dataset

Transition Costs

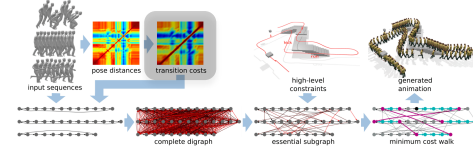
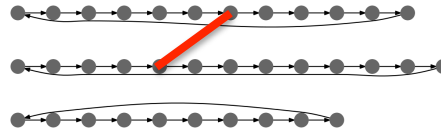
Motion synthesis pipeline :



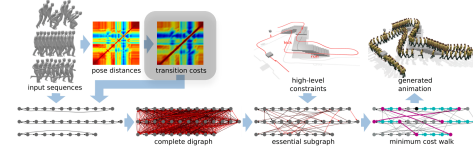
Transition Costs

Edge weight = **transition** cost in terms of **Realism criterion**:

$$\underbrace{E_{i,j}}_{\text{realism}} = \underbrace{D_{i,j}}_{\text{surface deformation}} + \alpha \underbrace{L_{i,j}}_{\text{duration}}$$

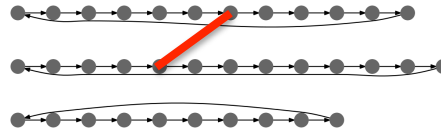


Transition Costs



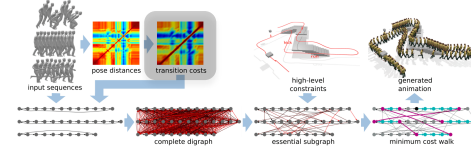
Edge weight = **transition** cost in terms of **Realism criterion**:

$$\underbrace{E_{i,j}}_{\text{realism}} = \underbrace{D_{i,j}}_{\text{surface deformation}} + \alpha \underbrace{L_{i,j}}_{\text{duration}}$$



Interpolated Transition: Source and target motion segments gradual blending

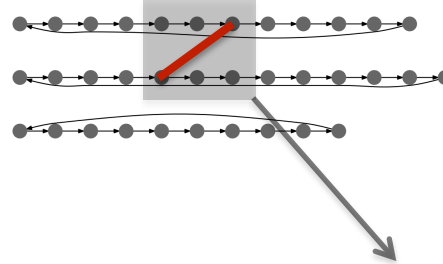
- Dynamic time warping
- Variable length blended segments



Transition Costs

Edge weight = **transition cost** in terms of **Realism criterion**:

$$\underbrace{E_{i,j}}_{\text{realism}} = \underbrace{D_{i,j}}_{\text{surface deformation}} + \alpha \underbrace{L_{i,j}}_{\text{duration}}$$



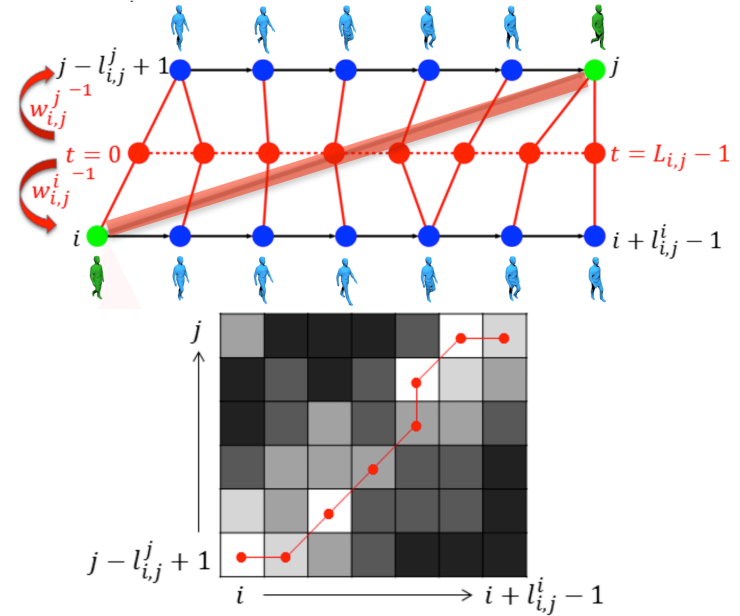
Interpolated Transition: Source and target motion segments gradual blending

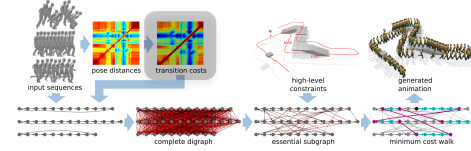
- Dynamic time warping
- Variable length blended segments

Finding **segment lengths** l^i and l^j , and **temporal warps** w^i and w^j that minimize the total **surface deformation cost**:

$$D(i, j) = \min_{l^i, l^j \in [l_{\min}, l_{\max}]} \underbrace{\min_{w^i, w^j} \sum_{t \in [0, L]} d(w^{i^1}(t), w^{j^1}(t))}_{\text{Dynamic Time Warping}}$$

Function $d(.,.)$ is a static **pose distance**.

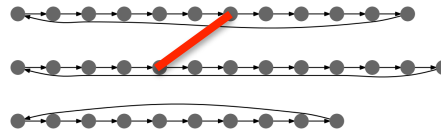




Transition Costs

Edge weight = **transition** cost in terms of **Realism** criterion:

$$\underbrace{E_{i,j}}_{\text{realism}} = \underbrace{D_{i,j}}_{\text{surface deformation}} + \alpha \underbrace{L_{i,j}}_{\text{duration}}$$



Interpolated Transition: Source and target motion segments gradual blending

- Dynamic time warping
- Variable length blended segments

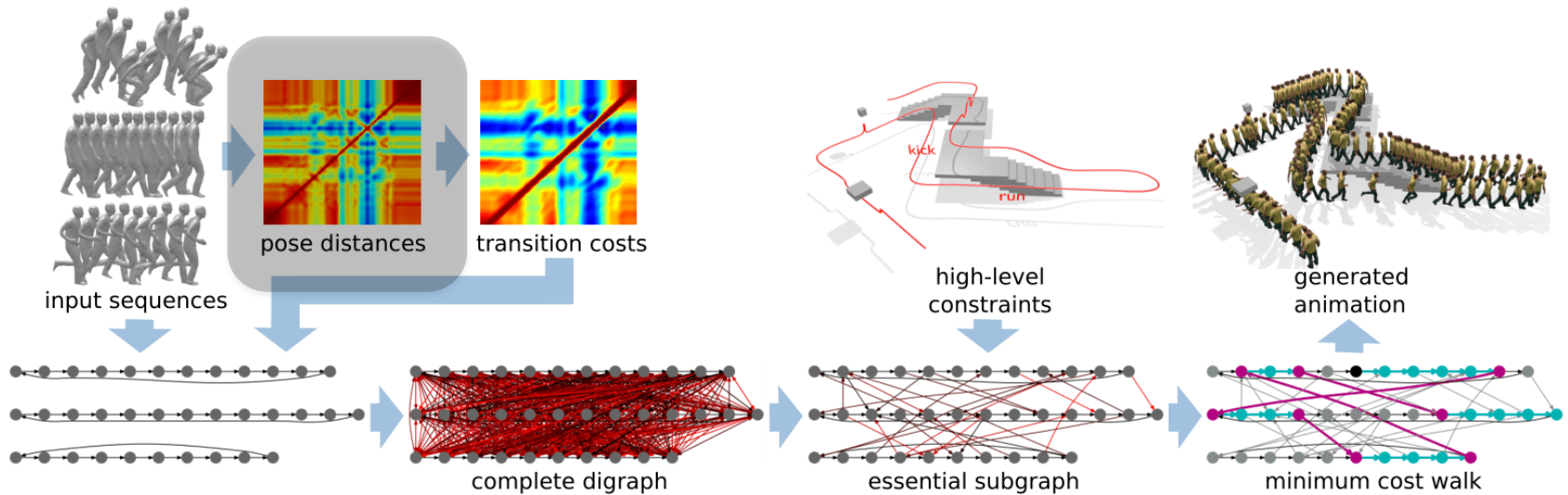


Motion synthesis level:

- **Pose interpolation** of matched frame along the transition
- Global displacements interpolation

Pose distance and interpolation

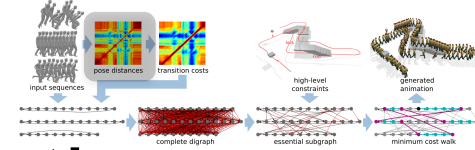
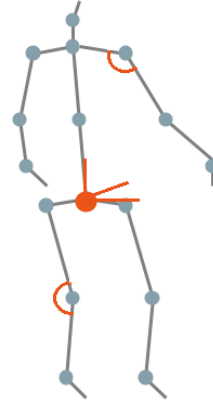
Motion synthesis pipeline :



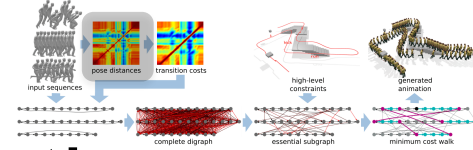
Pose distance and interpolation

Skeleton parameterization for Mo-cap data

Riemannian manifold: $SO(3)^{n_{joints}}$



Pose distance and interpolation

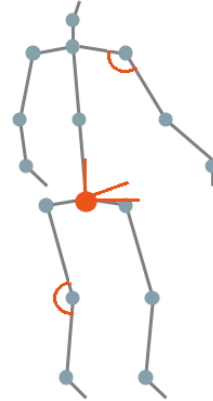


Skeleton parameterization for Mo-cap data

Riemannian manifold: $SO(3)^{n_{joints}}$

- Pose metric

$$\sum_{n_{joints}} \frac{1}{\sqrt{2}} \left\| \log(R_1^{-1} R_2) \right\|_F$$



Pose distance and interpolation

Skeleton parameterization for Mo-cap data

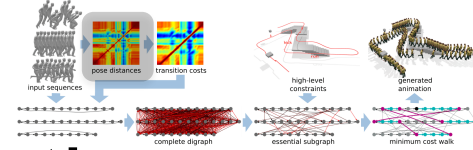
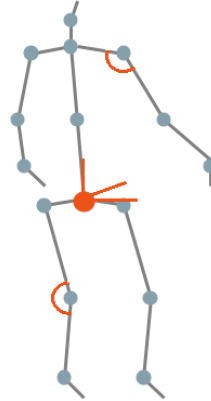
Riemannian manifold: $SO(3)^{n_{joints}}$

- Pose metric

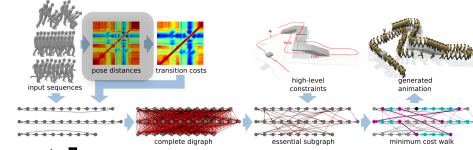
$$\sum_{n_{joints}} \frac{1}{\sqrt{2}} \left\| \log(R_1^{-1} R_2) \right\|_F$$

- Geodesic pose interpolation

$$R_1 e^{\lambda \log(R_1^{-1} R_2)}$$



Pose distance and interpolation



Skeleton parameterization for Mo-cap data

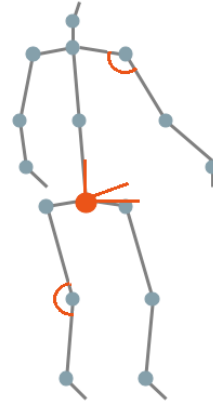
Riemannian manifold: $SO(3)^{n_{joints}}$

- Pose metric

$$\sum_{n_{joints}} \frac{1}{\sqrt{2}} \left\| \log(R_1^{-1} R_2) \right\|_F$$

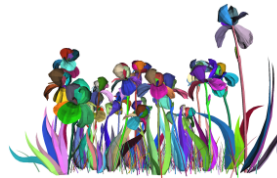
- Geodesic pose interpolation

$$R_1 e^{\lambda \log(R_1^{-1} R_2)}$$

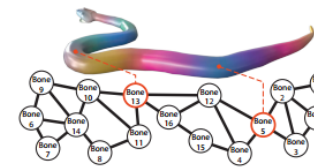


Skeleton parametrization for surfacic data:

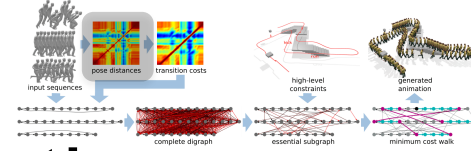
- Mesh Ensemble Motion Graph (James07)
- One-to-Many (Zheng13)



Articulated plant model, James07



LBS skinning, Zheng13



Pose distance and interpolation

Skeleton parameterization for Mo-cap data

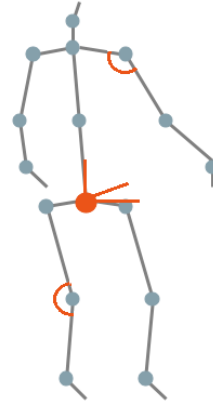
Riemannian manifold: $SO(3)^{n_{joints}}$

- Pose metric

$$\sum_{n_{joints}} \frac{1}{\sqrt{2}} \left\| \log(R_1^{-1} R_2) \right\|_F$$

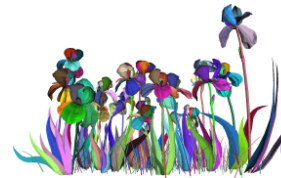
- Geodesic pose interpolation

$$R_1 e^{\lambda \log(R_1^{-1} R_2)}$$

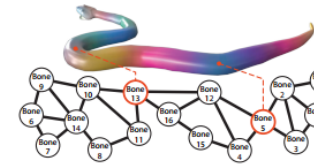


Skeleton parametrization for surfacic data:

- Mesh Ensemble Motion Graph (James07)
- One-to-Many (Zheng13)



Articulated plant model, James07

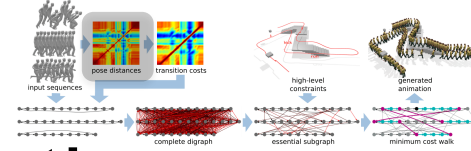


LBS skinning, Zheng13

Drawbacks

- Robust identification of an underlying skeleton structure is difficult
- Realism loss due to articulated model limitation

Pose distance and interpolation



Skeleton parameterization for Mo-cap data

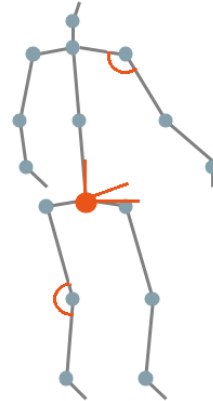
Riemannian manifold: $SO(3)^{n_{joints}}$

- Pose metric

$$\sum_{n_{joints}} \frac{1}{\sqrt{2}} \left\| \log(R_1^{-1} R_2) \right\|_F$$

- Geodesic pose interpolation

$$R_1 e^{\lambda \log(R_1^{-1} R_2)}$$

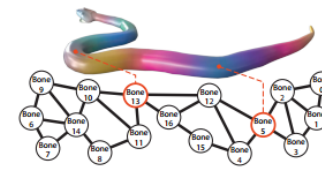


Skeleton parametrization for surfacic data:

- Mesh Ensemble Motion Graph(James07)
- One-to-Many(Zheng13)



Articulated plant model, James07



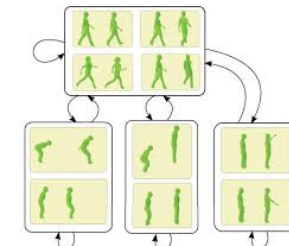
LBS skinning, Zheng13

Drawbacks

- Robust identification of an underlying skeleton structure is difficult
- Realism loss due to articulated model limitation

Surface based parametrization for surfacic data:

- Surface Motion Graph(Huang09)
- 4D Parametric Motion Graph(Casas13) (Heck07)(Shin06)

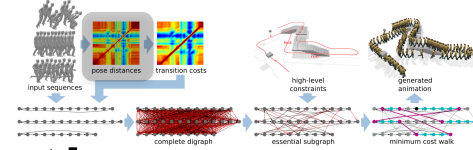


Parametric motion graph, Casas 13

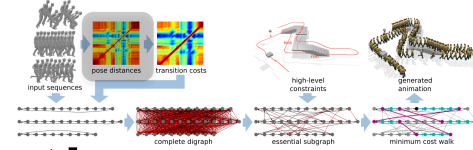
Pose distance and interpolation

Euclidean Parametrization $\mathcal{R}^{3 \times n_{vertices}}$

- Pose metric : sum of squared Euclidean distances between vertices
- pose interpolation : linear interpolation of vertices coordinates



Pose distance and interpolation

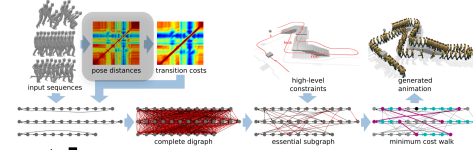


Euclidean Parametrization $\mathcal{R}^{3 \times n_{vertices}}$

- Pose metric : sum of squared Euclidean distances between vertices
- pose interpolation : linear interpolation of vertices coordinates

Drawbacks

- Rigid Alignment residual error do not account for mesh geometry
- Linear mesh interpolation introduces surface distortion



Pose distance and interpolation

Euclidean Parametrization $\mathcal{R}^{3 \times n_{vertices}}$

- Pose metric : sum of squared Euclidean distances between vertices
- pose interpolation : linear interpolation of vertices coordinates

Drawbacks

- Rigid Alignment residual error do not account for mesh geometry
- Linear mesh interpolation introduces surface distortion

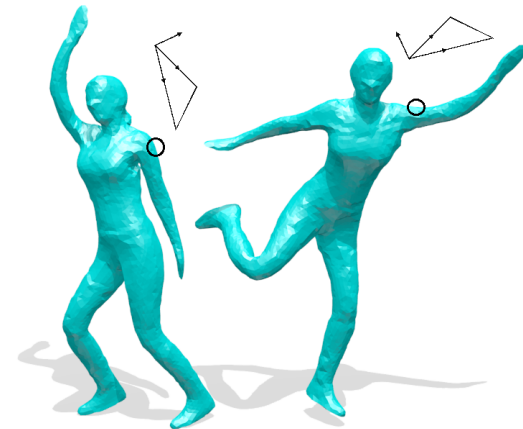
Surface Deformation based Parameterization for 3D triangular meshes:

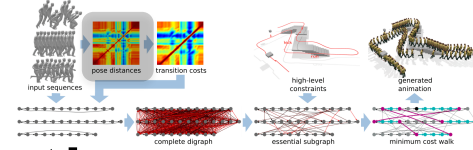
Polar decomposition of Deformation Gradients:

$$T = (v_1 - v_3, v_2 - v_3, n)^{-1} \cdot (v'_1 - v'_3, v'_2 - v'_3, n')$$

$$T = R \cdot S$$

Riemannian manifold: $(T_1, \dots, T_{m_{triangles}}) \in (SO(3) \times S_3^+)^{m_{triangles}}$





Pose distance and interpolation

Euclidean Parametrization $\mathcal{R}^{3 \times n_{vertices}}$

- Pose metric : sum of squared Euclidean distances between vertices
- pose interpolation : linear interpolation of vertices coordinates

Drawbacks

- Rigid Alignment residual error do not account for mesh geometry
- Linear mesh interpolation introduces surface distortion

Surface Deformation based Parameterization for 3D triangular meshes:

Polar decomposition of Deformation Gradients:

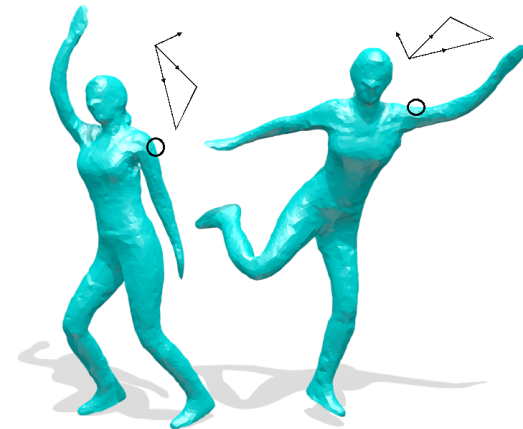
$$T = (v_1 - v_3, v_2 - v_3, n)^{-1} \cdot (v'_1 - v'_3, v'_2 - v'_3, n')$$

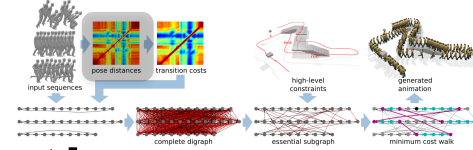
$$T = R \cdot S$$

Riemannian manifold: $(T_1, \dots, T_{m_{triangles}}) \in (SO(3) \times S_3^+)^{m_{triangles}}$

- Pose metric

$$d(.,.) = \sum_{m_{triangles}} \frac{1}{\sqrt{2}} (\|\log(R)\|_F + \|\log(S)\|_F)$$





Pose distance and interpolation

Euclidean Parametrization $\mathcal{R}^{3 \times n_{vertices}}$

- Pose metric : sum of squared Euclidean distances between vertices
- pose interpolation : linear interpolation of vertices coordinates

Drawbacks

- Rigid Alignment residual error do not account for mesh geometry
- Linear mesh interpolation introduces surface distortion

Surface Deformation based Parameterization for 3D triangular meshes:

Polar decomposition of Deformation Gradients:

$$T = (v_1 - v_3, v_2 - v_3, n)^{-1} \cdot (v'_1 - v'_3, v'_2 - v'_3, n')$$

$$T = R \cdot S$$

Riemannian manifold: $(T_1, \dots, T_{m_{triangles}}) \in (SO(3) \times S_3^+)^{m_{triangles}}$

- Pose metric

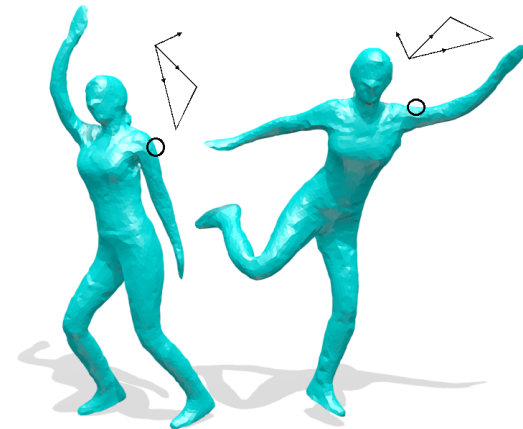
$$d(.,.) = \sum_{m_{triangles}} \frac{1}{\sqrt{2}} (\|\log(R)\|_F + \|\log(S)\|_F)$$

- Pose interpolation

- Geodesic transformation interpolation

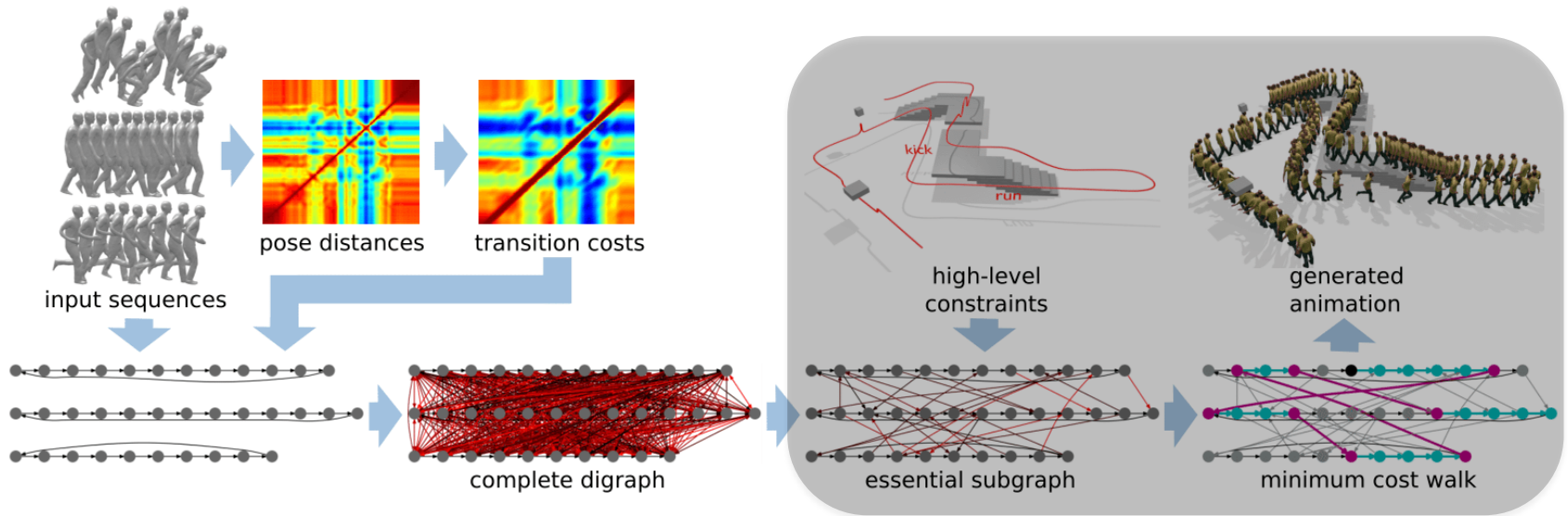
$$\tilde{T} = e^{\lambda \log(R)} \cdot e^{\lambda \log(S)}$$

- Gradient deformation with interpolated transformation
- 3D Poisson Shape reconstruction



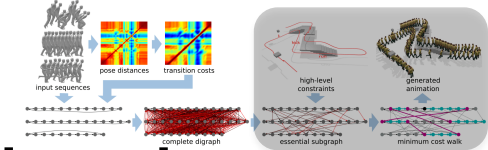
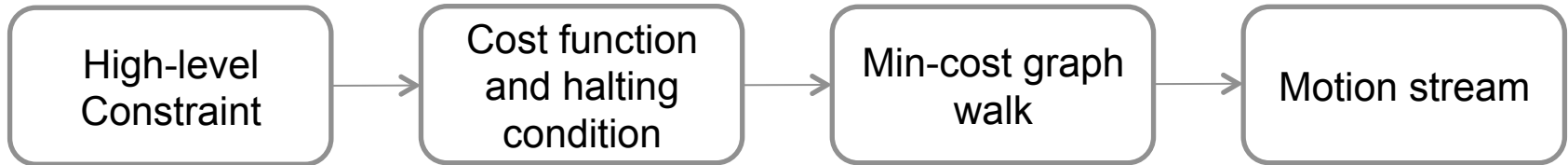
User guided animation synthesis

Motion synthesis pipeline :



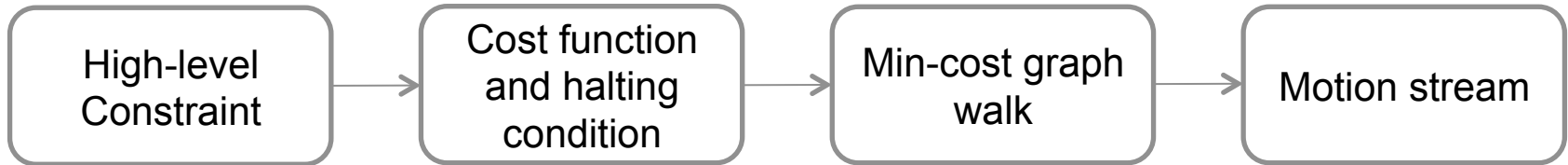
User guided animation synthesis

Motion extraction casted as a graph search problem and solved using depth first search with Branch and bound



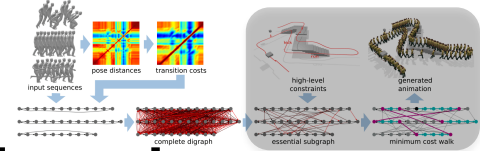
User guided animation synthesis

Motion extraction casted as a graph search problem and solved using depth first search with Branch and bound

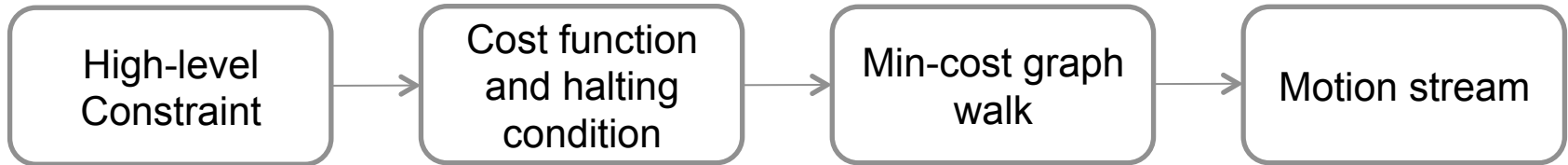


- Motion Stream
 - Concatenate original and interpolated motion segments
 - Rigid alignment at segment junctions

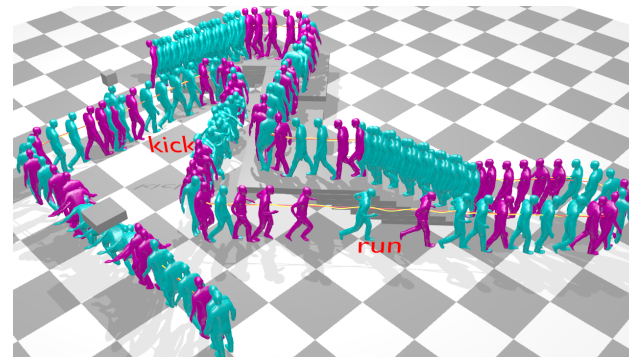
User guided animation synthesis



Motion extraction casted as a graph search problem and solved using depth first search with Branch and bound



- Motion Stream
 - Concatenate original and interpolated motion segments
 - Rigid alignment at segment junctions
- High Level constraints
 - 3d behavioral path synthesis
 - Follow a 3D path, behave as precised at some parts
 - Adapted to locomotion data
 - Pose/time constraint
 - reach specific poses at specific times
 - Adapted to unstructured acyclic motion



3D behavioral path synthesis



2D path synthesis

User guided motion synthesis



User guided motion synthesis



User guided motion synthesis



Conclusion

- Realistic shape and appearance animation generation from user specified constraints
- Contributions:
 - An optimal structure for motion data organization and reuse
 - A method for improving realism in motion transitions
 - A novel high level constraint formulation
- Limitations:
 - Foot skate
- Future work:
 - Data annotation
 - Texture interpolation
 - More Data : Motion warping, Motion transfert, Merging parts motion.

Thank You !

