Robust Marker Trajectory Repair for MOCAP using Kinematic Reference

Maksym Perepichka
maksym@perepichka.com
Concordia University
Montreal, Quebec, Canada

Daniel Holden
daniel.holden@Ubisoft.com
Ubisoft La Forge
Montreal, Quebec, Canada

Sudhir P. Mudur
sudhir.mudur@concordia.ca
Concordia University
Montreal, Quebec, Canada

Tiberiu Popa
tiberiu.popa@concordia.ca
Concordia University
Montreal, Quebec, Canada

Motivation

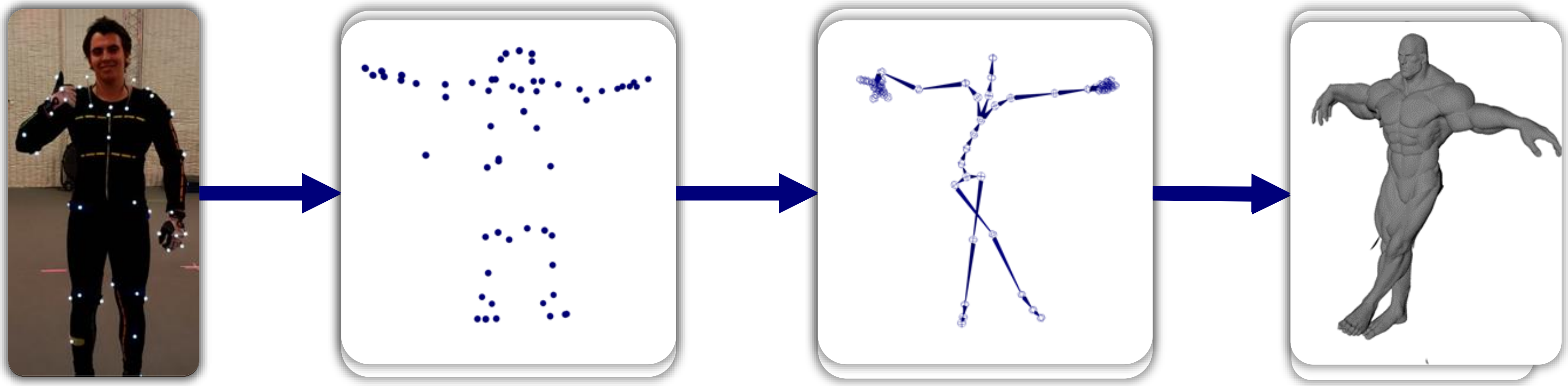
Getting Animation Data for Games

- Keyframing
- MOCAP
  - Markerless
  - Markers
    - Active
    - **Passive**

Challenges

Ideal MOCAP Pipeline

Actual MOCAP Pipeline



MOCAP Studio **Marker Data** **Kinematic Solution** **Delivery**

- Markers often have erroneous gaps
  - Occlusion, Swapping, Noise, Trajectory Errors
- Erroneous gaps require MOCAP artist cleanup
  - Time and Expertise
  - Bottleneck for production

Solutions

Marker Level Solutions

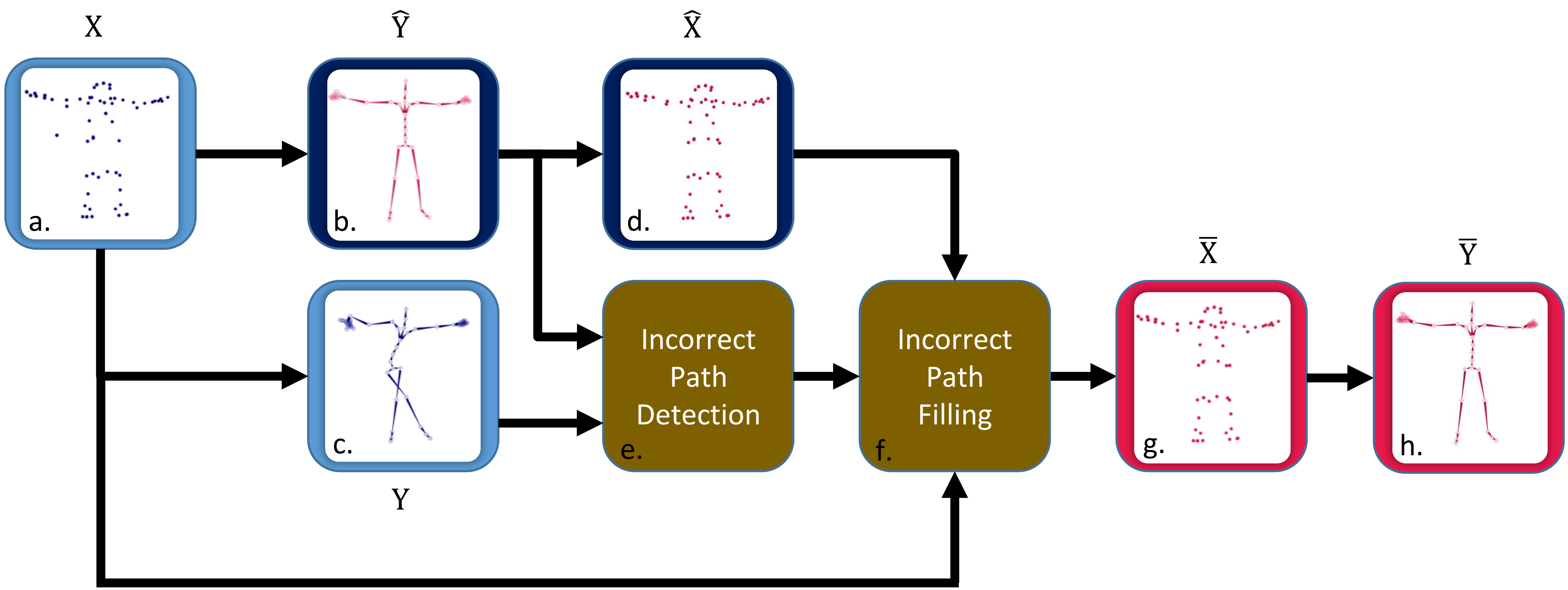
- Solve markers individually
- Marker PCA *[Liu and McMillan 2006; Federolf 2013; Gløersen and Federolf 2016]*
- Data-driven *[Baumann et al.2011; Hsu et al.2004; Wang et al.2014; Zhang and van de Panne 2018; Kucherenko et al. 2018]*
- Doesn't take into account kinematic validity

Solutions

Kinematic Level Solutions

- Solve directly
- Kinematically valid, robust to noise
- Commercial Solvers: *[Vicon Software]*
- Data-driven: *[Ren et al.2005; Kim and Rehg 2008; Shen et al.2012; Fragkiadaki et al.2015; Mall et al. 2017; Holden 2018]*
- Hard to integrate → Marker paths are gone

Our Solution



a. Raw Marker Data X

b. *[Holden 2018]** Solver $\hat{Y}$

c. Commercial Kinematic Solution Y

d. Reconstructed Markers $\hat{X}$

e. Incorrect Path Detection

f. Incorrect Path Filling

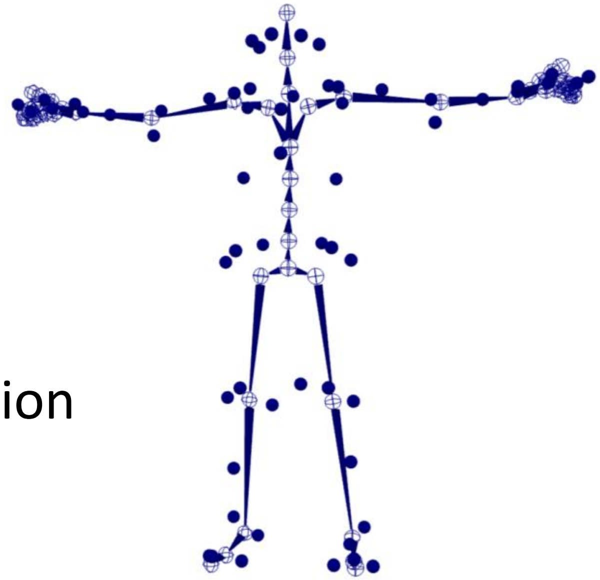
g. Fixed Marker Paths $\bar{X}$

h. Final Kinematic Solution $\bar{Y}$

Solution: Marker Reconstruction

- Reconstruct markers from $\hat{\mathbf{Y}}$ using LBS

$$\mathrm{LBS}(\hat{\mathrm{Y}}, \mathrm{Z}) = \sum_{i=0}^{j} \mathbf{w}_i \odot (\hat{\mathrm{Y}}_i \otimes \mathbf{Z}_i)$$



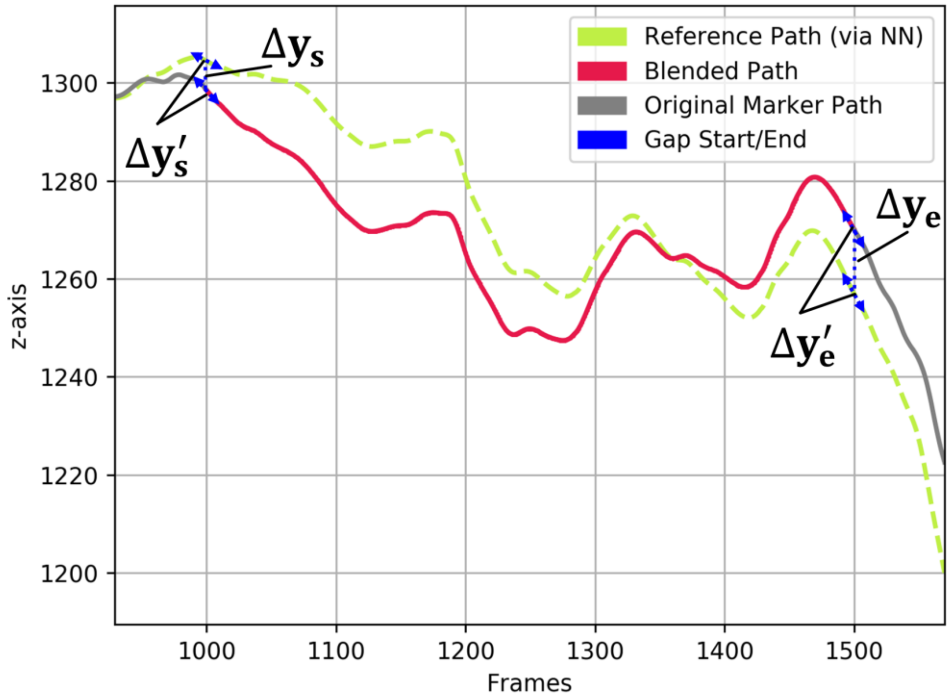
- New markers coined $\hat{\mathbf{X}}$
  - Marker representation of $\hat{\mathbf{Y}}$ kinematic solution
  - “Clean” set of markers

- Issues:
  - Missing small detail
  - Offset from original paths $\mathbf{X}$

Solution: Gap Filling

Fix Erroneous Frames in $\mathbf{X}$ using $\hat{\mathbf{X}}$

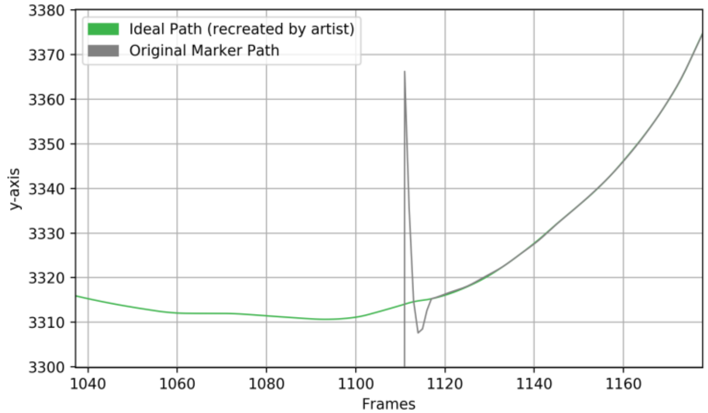
- Get $\Delta y_s, \Delta y_s'$ and $\Delta y_e, \Delta y_e'$
- Fit degree 5 polynomial
- Subtract polynomial from original path

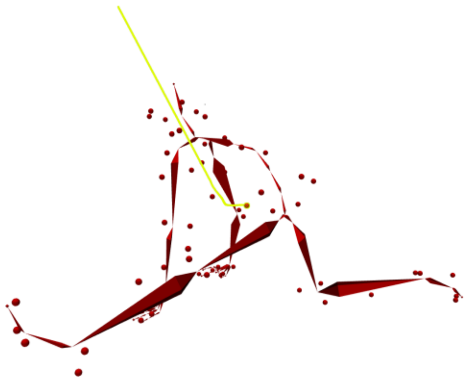


Solution: Gap Filling

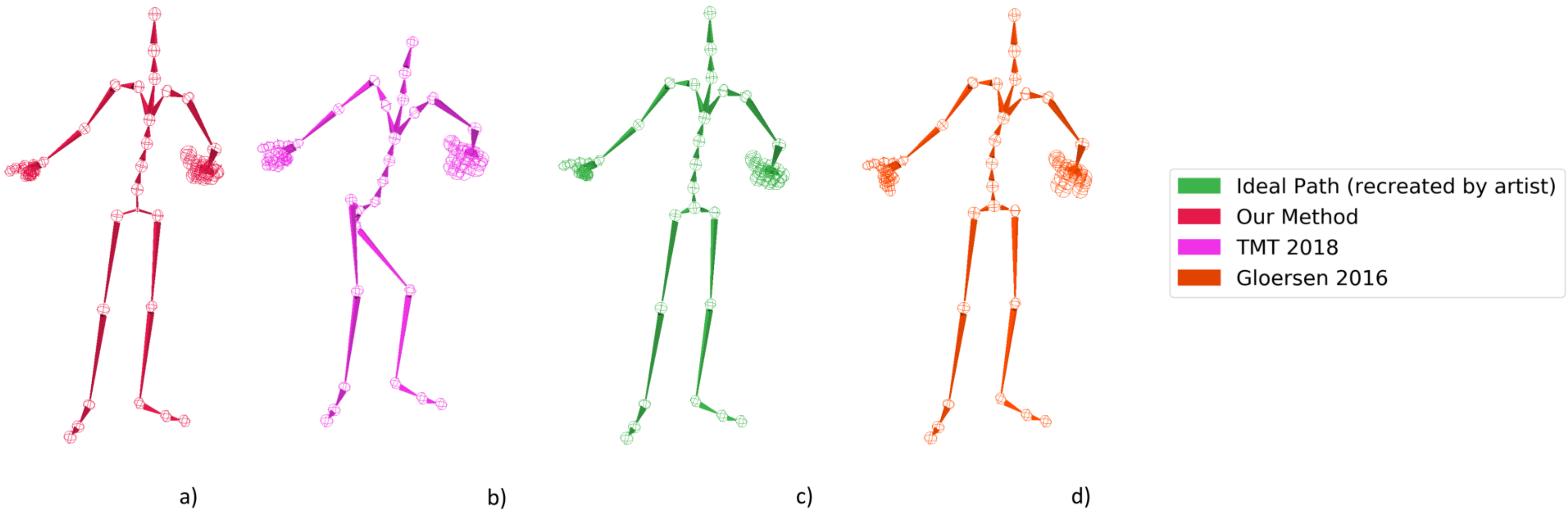
Handle Degenerate Cases

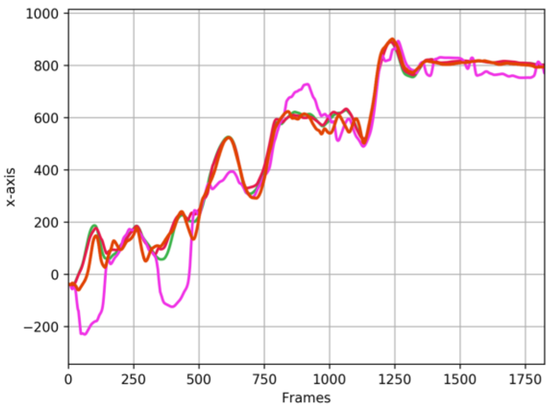
- Pad start/end frames
- Search for best start/end frames for gaps
- Clamp velocity differences

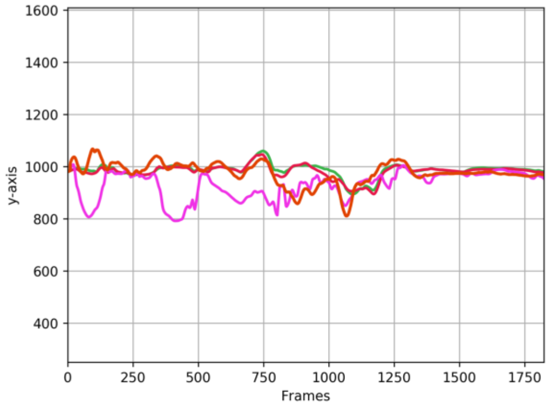


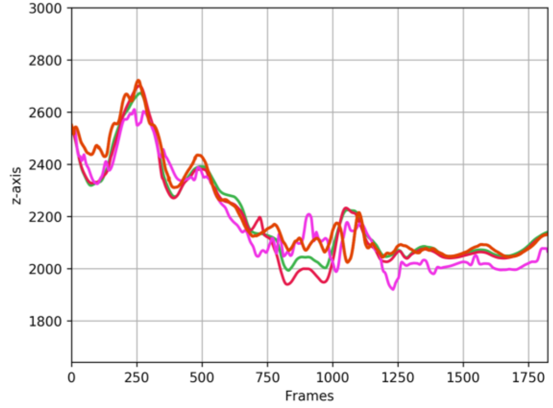


Results

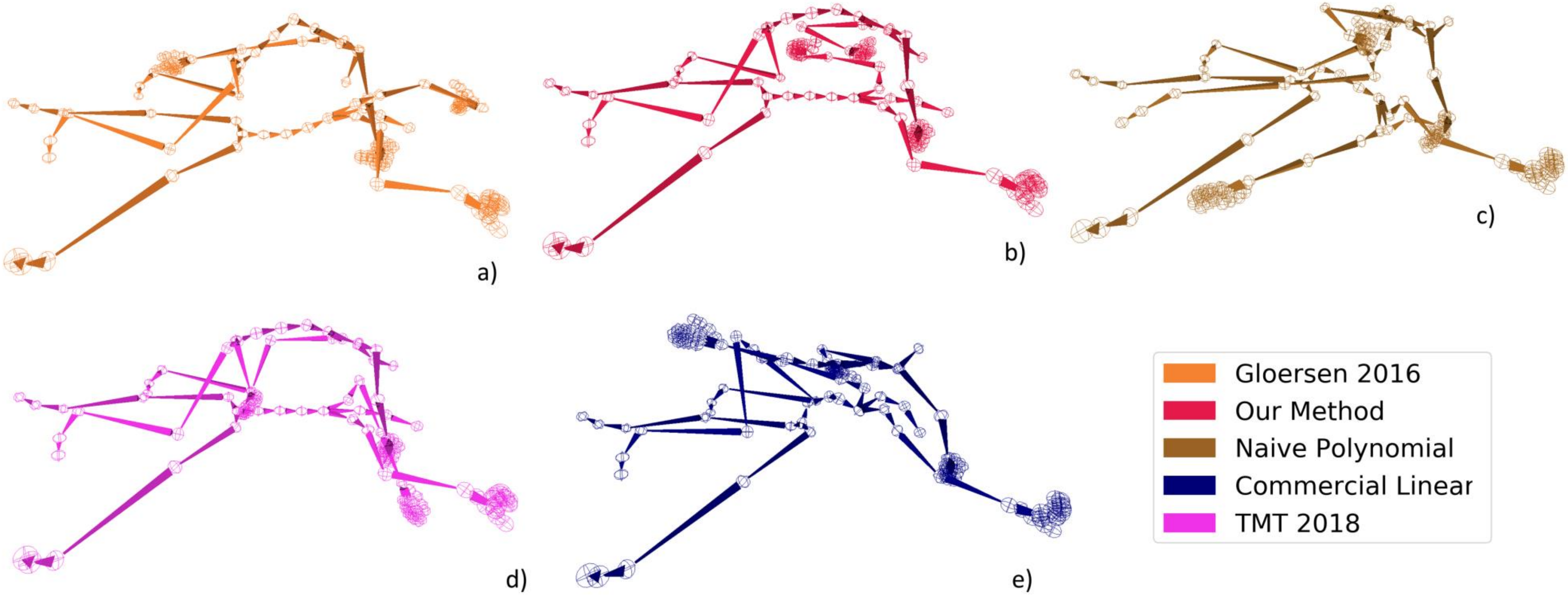

Ideal Path (recreated by artist)
Our Method
TMT 2018
Gloersen 2016
a)
b)
c)
d)


x-axis
Frames
1000
800
600
400
200
0
−200
0
250
500
750
1000
1250
1500
1750


y-axis
Frames
1600
1400
1200
1000
800
600
400
0
250
500
750
1000
1250
1500
1750

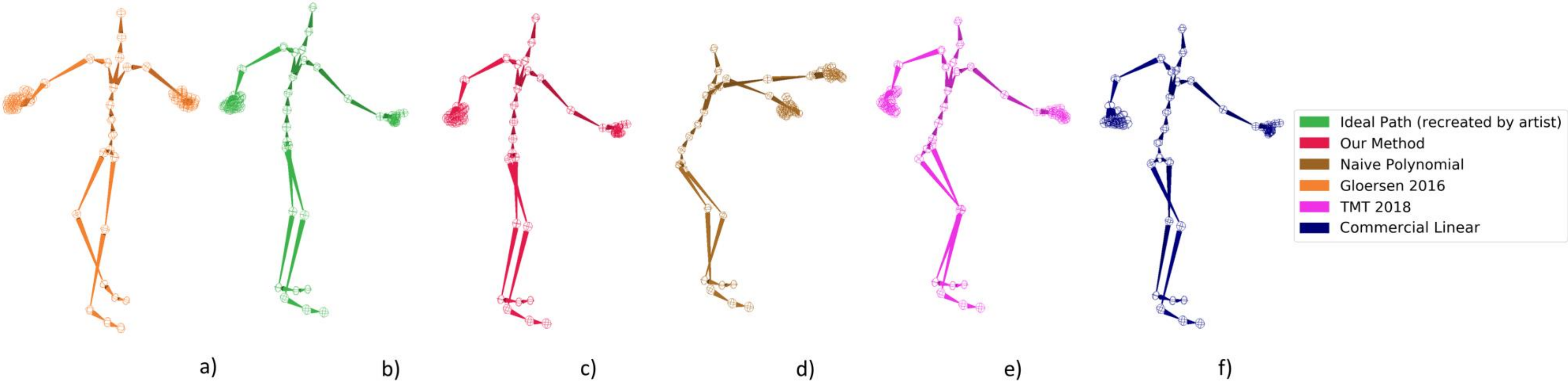

z-axis
Frames
3000
2800
2600
2400
2200
2000
1800
0
250
500
750
1000
1250
1500
1750

Results



a)
b)
c)
d)
e)
Gloersen 2016
Our Method
Naive Polynomial
Commercial Linear
TMT 2018

Results



Method	Δ_{pos}	Δ_{rot}
Original	1.841	35.220
Naive Polynomial	10.678	42.93
Commerical	1.053	22.101
Gloersen 2016	0.716	23.827
TMT 2018	0.603	8.038
Holden 2018	1.337	19.604
Ours	**0.288**	**4.727**

Results

- TMT 2018
- Our Method
- Naive Polynomial
- Commercial Linear
- Gloersen 2016
- Ideal Path (recreated by artist)

Results

- Performance metrics on 10,920 frames of animation at 120FPS

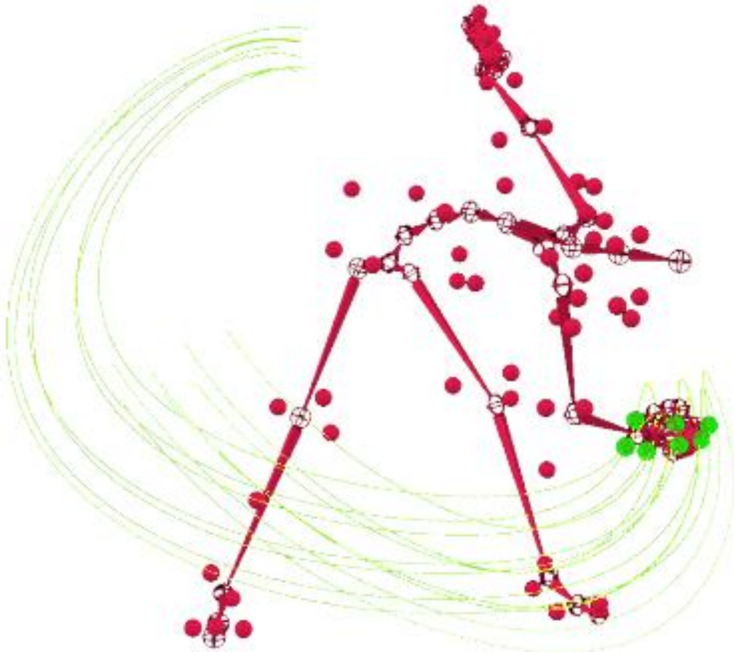
Method	FPS	Time (%)
NN Preprocessing	4029	2
NN Evaluation	1339	6
Filtering	5527	1
IK Retargeting	108	74
Marker Reconstruction	2278	4
Gap Detection	10749	1
Gap Preprocessing	904	9
Marker Filling	2620	3
Total	81	100

Future

- Limitations:
  - Depends on performance of robust kinematic solver
  - Degree 5 polynomial
  - Ambiguous results when large number of markers missing

- Future work:
  - Use different kinematic solver
  - Use different IK process
  - Improve on blending → interialization?
  - Integrate with markerless system

Q&A



Appendix: Improved Kinematic Solver

- Based on *[Holden 2018]*
- Improve kinematic solver to remove issues such as foot sliding
- Augment dataset using local perturbations
  - Sampled on entire dataset of training data
- Increase correlations between missing markers

Appendix: Erroneous Interval Detection

- Compare kinematic solutions: Positions and Rotation
- Check against threshold
  - Positions → 10cm max
  - Rotation → 30 degrees max
- Determine erroneous markers

Appendix: Erroneous Interval Detection

Algorithm 1 Given two kinematic solutions, an erroneous one and a clean one, determine the set of missing markers paths

Function *DetectBad* $(\mathrm{Y} \in \mathbb{R}^{n \times j \times 3 \times 4}, \hat{\mathrm{Y}} \in \mathbb{R}^{n \times j \times 3 \times 4})$

// Get differences between two kinematic solutions

$\Delta \mathrm{P} \in \mathbb{R}^{n \times j} \leftarrow ||\hat{\mathrm{Y}} - \mathrm{Y}||$

$\Delta \mathrm{R} \in \mathbb{R}^{n \times j} \leftarrow \mathrm{Angle}(\hat{\mathrm{Y}}\mathrm{Y}^{\top})$

// Get joints surpassing allowed threshold

$$\mathbf{J} \in (0,1)^{n \times j} \leftarrow \begin{cases} 1, & \text{if } \Delta\mathbf{P} - \Delta\mathrm{P}_{\max} >= 0 \\ & \text{or } \Delta\mathbf{R} - \Delta\mathrm{R}_{\max} >= 0 \\ 0, & \text{otherwise} \end{cases}$$

// Get markers associated with joints surpassing threshold

$$\hat{\mathbf{w}} \in (0,1)^{m \times j} \leftarrow \begin{cases} 1, & \text{if } \mathbf{w} > 0 \\ 0, & \text{otherwise} \end{cases}$$

$\mathbf{M} \in (0,1)^{n \times m} \leftarrow \mathbf{J}\,\hat{\mathbf{w}}$

return M

End

Appendix: Gap Filling

Algorithm 2 Given a gap in marker data, fill the gap by referring to a reference marker path

Function *GapFill* ($X \in \mathbb{R}^{n\times m\times 3}, \hat{X} \in \mathbb{R}^{n\times m\times 3}, M \in \mathbb{N}^{g\times 3}$)

// Copy over marker paths

$\bar{X} \in \mathbb{R}^{n\times m\times 3} \leftarrow X$

// Loop over all gaps in a clip

for i ... g **do**

// Get start, end and marker indices

$s, e, m \leftarrow M_{i,1}, M_{i,2}, M_{i,3}$

// Get differences in positions

$$\Delta y_s \in \mathbb{R}^3 \leftarrow \begin{cases} \hat{X}^m_{s-1} - X^m_{s-1}, & \text{if } s > 1 \\ \hat{X}^m_{e+1} - X^m_{e+1}, & \text{if } e < n \\ 0 & \text{otherwise} \end{cases}$$

$$\Delta y_e \in \mathbb{R}^3 \leftarrow \begin{cases} \hat{X}^m_{e+1} - X^m_{e+1}, & \text{if } e < n \\ \hat{X}^m_{s-1} - X^m_{s-1}, & \text{if } s > 1 \\ 0 & \text{otherwise} \end{cases}$$

// Get differences in velocities

$$\Delta y'_s \in \mathbb{R}^3 \leftarrow \begin{cases} \frac{(\hat{X}_{s-1}-\hat{X}_{s-3})}{2} - \frac{(X_{s-1}-X_{s-3})}{2}, & \text{if } s > 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\Delta y'_e \in \mathbb{R}^3 \leftarrow \begin{cases} \frac{(\hat{X}_{e+3}-\hat{X}_{e+1})}{2} - \frac{(X_{e+3}-X_{e+1})}{2}, & \text{if } e < n \\ 0 & \text{otherwise} \end{cases}$$

// Clamp velocity differences

$\Delta y'_s \leftarrow \text{AbsClamp}(\Delta y'_s, \Delta_{\max})$

$\Delta y'_e \leftarrow \text{AbsClamp}(\Delta y'_e, \Delta_{\max})$

// Fit cubic polynomial using constraints

$\mathcal{P} \leftarrow \text{HermiteSpline}(\Delta y_s, \Delta y_e, \Delta y'_s, \Delta y'_e)$

// Subtract polynomial from reference

$\bar{X}^m_{s\ldots e} \leftarrow \hat{X}^m_{s\ldots e} - \mathcal{P}_{s\ldots e}$

end for

return $\bar{X}$

End