KeyPoint Subspace Acceleration and SoftCaching

Mark Meyer and John Anderson
Pixar Animation Studios
Outline

• Motivation

• Basic Algorithm
  – KeyPoint Subspace Acceleration (KPSA)
  – SoftCaching

• Results
  – Facial Articulation
  – Rendering Indirect Illumination

• Conclusions and Future Work
Motivation

High quality articulated characters pose slowly

- Expensive deformation calculations
- Performed at many points

Results are often correlated
Motivation

Character Posing Block Diagram

Animation Controls → Posing Engine → Posed Points

Often slow
Motivation

Subspace Acceleration Block Diagram

- Works well on jointed characters
- More difficult on nonlinear areas such as faces
Motivation

KeyPoint Subspace Acceleration Block Diagram

Posing the Key Points deals with non-linearities
KPSA Algorithm

Subspace Approximation

\[ f \approx \hat{f} = Ap \]

- Posed Points
- Subspace Approximation
- Subspace Matrix
- Subspace Coordinates
KPSA Algorithm

Subspace Coordinate Computation

• Pose only the key points: \( \mathbf{f}_{\text{key}} \)

• Project \( \mathbf{f}_{\text{key}} \) onto the subspace using least squares

\[
p = \underset{p}{\arg \min} \left\| \mathbf{f}_{\text{key}} - A_{\text{key}} p \right\|
\]
KPSA Algorithm

Basis Motion Computation

- Given a training set of example poses
- Perform *Principal Component Analysis* (PCA):
  - Choose the $M$ most significant basis motions

The basis motions can be transformed without affecting the subspace
KPSA Algorithm

KeyPoint Computation

PCA basis motions (4 of 85)

The basis motions are global
KPSA Algorithm

KeyPoint Computation

Varimax rotated basis motions (4 of 85)

Key Points are points with maximal motion
KPSA Algorithm

- **Preprocess:**
  1) Build the subspace
  2) Choose the Key Points

- **Runtime:**
  1) Pose the Key Points: \(f_{key}\)
  2) Compute the subspace coordinates: \(p\)
  3) Compute the subspace approximation: \(\hat{f}\)
Sources of Error

Projection Error

• Error due to the subspace not containing the pose

Cueing Error

• Error due to not finding the “closest” pose in the subspace
Subspace approximations can fail

Use the KeyPoint Error as a confidence metric:

- High Error: use the fully posed solution
- Low Error: use the KPSA solution
SoftCaching

Subspace approximations can fail

Use the KeyPoint Error as a confidence metric:

- High Error: use the fully posed solution
- Low Error: use the KPSA solution
KPSA-SC Algorithm

- **Preprocess:**
  1) Build the subspace
  2) Choose the Key Points

- **Runtime:**
  1) Pose the Key Points: \( \mathbf{f}_{key} \)
  2) Compute the subspace coordinates: \( \mathbf{p} \)
  3) Compute the subspace approximation: \( \hat{\mathbf{f}} \)
  4) SoftCache using KeyPoint error
Results - Dash

- Fully Posed
- Key Points
- KPSA

170 Key Points, 2986 Total Points, 8.7x speedup
Results - Dash

Fully Posed     KPSA

170 Key Points, 2986 Total Points, 8.7x speedup
Results - McQueen

KPSA results

70 Key Points, 2625 Total Points, 15x speedup
Results - Ratatouille

KPSA results

80 Key Points, 4150 Total Points, 20.75x speedup
Results - Global Illumination

Example Training Images

(a) Example Training Images

Fully Rendered Results

(b)

(c)

(d)

KPSA Results

200 Key Points, 160000 Total Points
Conclusions

- **KPSA - acceleration**
  - General
  - Automatic Key Point Selection
  - Soft Caching
- **Limitations**
  - Subspace quality dependent on training set
  - Acceleration determined by time to pose Key Points
Future Work

• Alternate Key Point selection algorithms
  – Incorporate posing cost into key point selection

• “Local” cache misses
  – Only fully compute locally for a cache miss
Acknowledgements

We would like to thank:

- Pixar Animation Studios
- Pixar Studio Tools
- Pixar Research
Error Plots

RMS Error for KPSA Results

RMS Error for SoftCaching Results
Error Equations

Projection Error

\[ e_{proj} = \| f - A \ p_{proj} \| \]

\[ p_{proj} = \arg \min_{p} \| f - A \ p \| \]

Cueing Error

\[ e_{cue} = \| A \ p_{proj} - A \ p \| \]