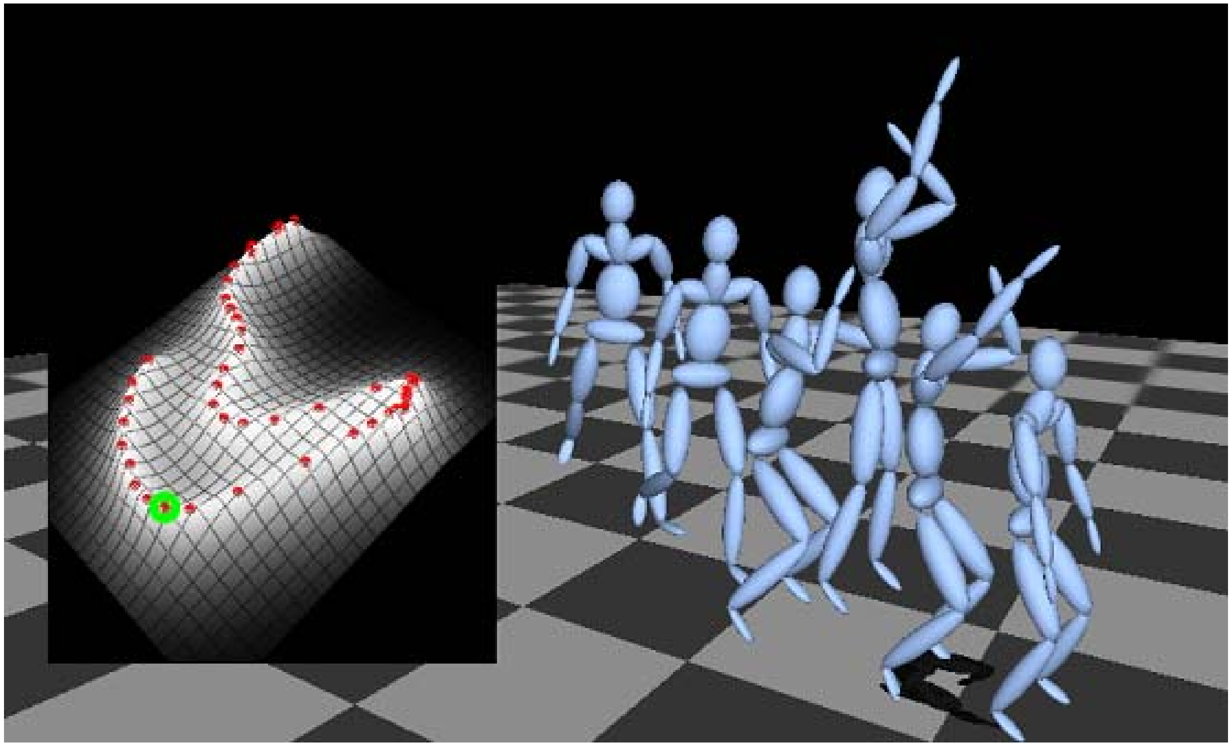
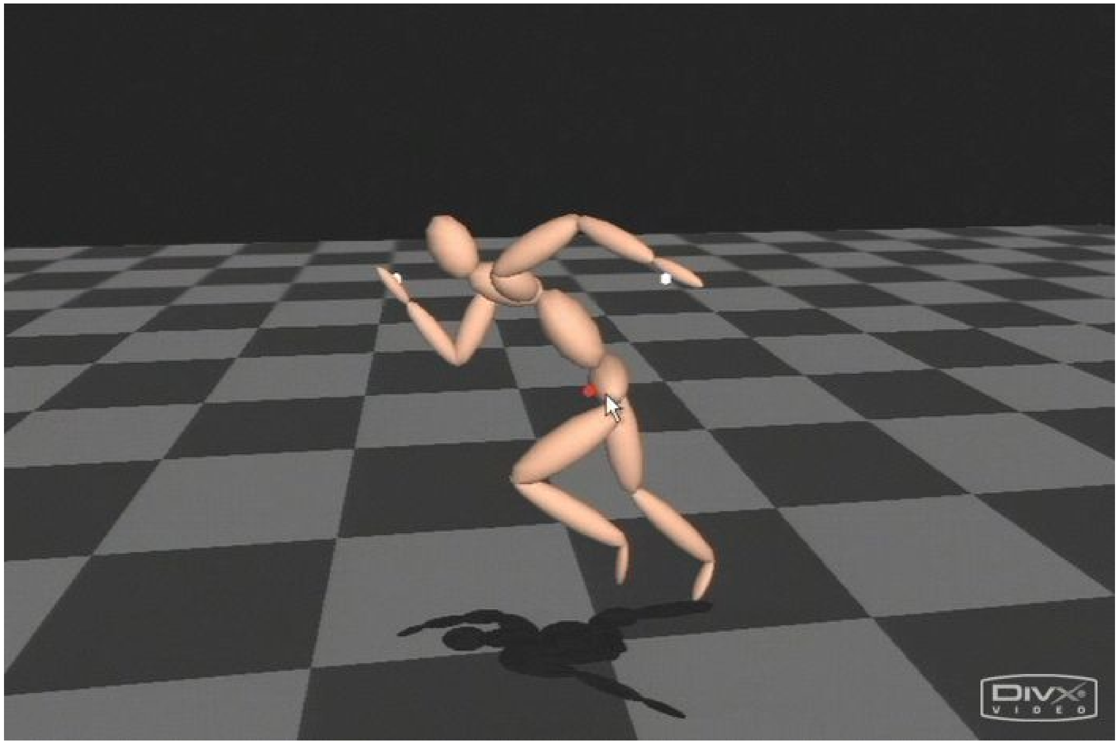
Style-based Inverse Kinematics

Keith Grochow, Steven L. Martin, Aaron Hertzmann, Zoran Popovic
SIGGRAPH'04



Presentation by
Peter Hess

Inverse Kinematics (1)



- Goal: Compute a human body pose from a set of constraints.
- Problem: The problem is inherently underdetermined!

  There are many possible solutions, but most of them don't look natural.

Inverse Kinematics (2)

- Solution: Restrain the algorithm to a certain "style" of natural looking poses ⇨ constrained optimization problem
- Previous approaches: Use a reference pose and define a distance metric
  - Mass displacement
  - Energy consumption
- Drawbacks:
  - Not every metric is good for every style.
  - Finding a good metric for a certain style is difficult.

Overview

- Scaled Gaussian Process Latent Variable Model (SGPLVM)
  - Learning
  - Synthesis
- Optimizations / Extensions
  - Annealing
  - Active Set
  - Style Interpolation
- Applications
- Performance
- Conclusions

SGPLVM: Setting (1)

Main idea

- Learning
  - Take a set of motion capture poses for training
  - Search a lowdimensonal representation of these data
  - Create a probability distribution function (PDF) in latent space
- Create new poses (synthesis)
  - For a set of constraints, use the PDF to find the pose which is most likely compared to the input.

SGPLVM: Setting (2)

- Pose $\mathbf{q}_i$:
  - Input poses are represented by skeletons
  - Joint angles, root position and root orientation
  - 42 dim
- Feature vector $\mathbf{y}_i$:
  - Joint angles from $\mathbf{q}_i$
  - Vertical orientation
  - Velocity and acceleration
  - 100+ dim

SGPLVM: Setting (3)

- Scaling matrix **W**
  - **W** = diag{w_1, ..., w_D}
  - Scales the dimensions of **y** according to their importance: **Wy**
- Latent variables $\mathbf{x}_i$
  - $\mathbf{x}_i$ are linked with $\mathbf{y}_i$ by a Gaussian Process
  - 2-3 dim
  - Lowdimensional representation of $\mathbf{y}_i$
  - Dimension reduction is only possible, because natural movements are very structured.

SGPLVM: Setting (4)

- Kernel k

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\gamma}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) + \delta_{x,x'}\beta^{-1}$$

  - Parameters α, β, γ initially unknown
  - Value of k mainly depends on distance between x and x'
  - Shows the correlation between **y** and **y'**, based on their corresponding **x** and **x'**

SGPLVM: Setting (5)

- The mapping between $\mathbf{x}_i$ and $\mathbf{y}_i$ is defined by a Gaussian Process. The likelihood for $\mathbf{y}_{i,k}$ is:

$$p(\{\mathbf{y}_{i,k}\} \mid \{\mathbf{x}_i\}, \alpha, \beta, \gamma) = \frac{1}{\sqrt{(2\pi)^N |\mathbf{K}|}} \exp(-\frac{1}{2} \mathbf{Y}_k^T \mathbf{K}^{-1} \mathbf{Y}_k)$$

- K is the covariance matrix of the feature vectors:
  - $k(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{cov}(\mathbf{y}_i, \mathbf{y}_j)$
  - distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ small $\leftrightarrow$ $\mathbf{y}_i$ and $\mathbf{y}_j$ are simillar

SGPLVM: Learning (1)

- Known:

  $\{\mathbf{q}_i\}, \{\mathbf{y}_i\}$

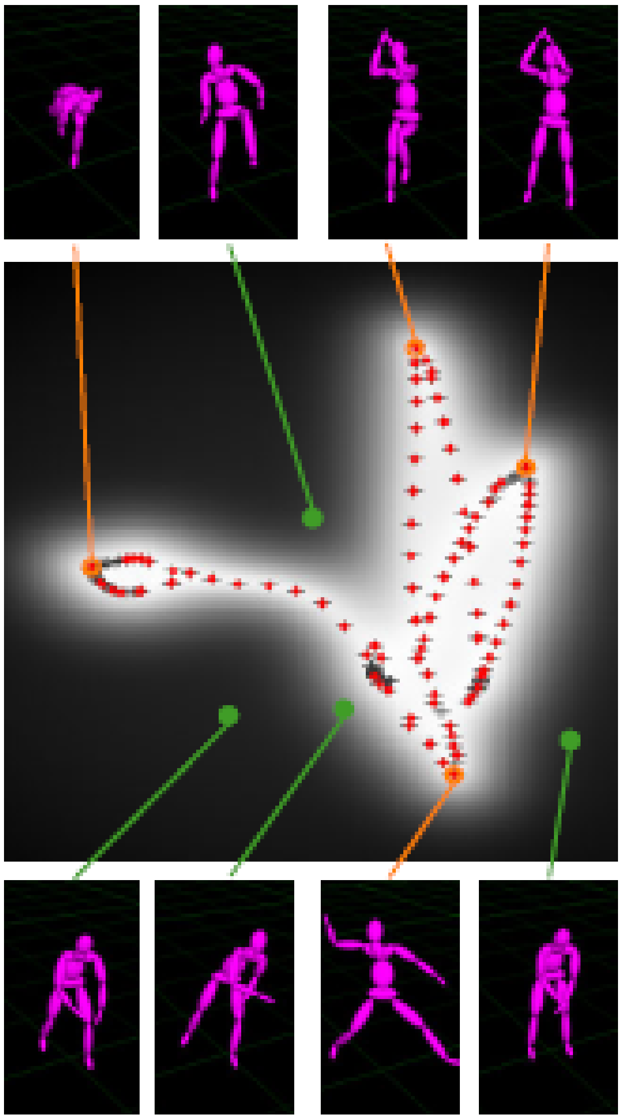
- Unknown:

  $\{\mathbf{x}_i\}, \{w_i\}, \alpha, \beta, \gamma$

- Maximize posterior probability:

$$\max_{\{\mathbf{x}_i\},\{w_i\},\alpha,\beta,\gamma} p(\{\mathbf{x}_i\},\{w_i\},\alpha,\beta,\gamma \mid \{\mathbf{y}_i\})$$

SGPLVM: Learning(2)



- Minimizing L_{GP} will learn all the model parameters you need:

$$\min_{\{\mathbf{x}_i\},\{w_i\},\alpha,\beta,\gamma} L_{GP} = \frac{D}{2}\ln|\mathbf{K}| + \frac{1}{2}\sum_k w_k^2 \mathbf{Y}_k^T \mathbf{K}^{-1}\mathbf{Y}_k + \frac{1}{2}\sum_i \|\mathbf{x}_i\|^2 + \ln\frac{\alpha\beta\gamma}{\prod_k w_k^N}$$

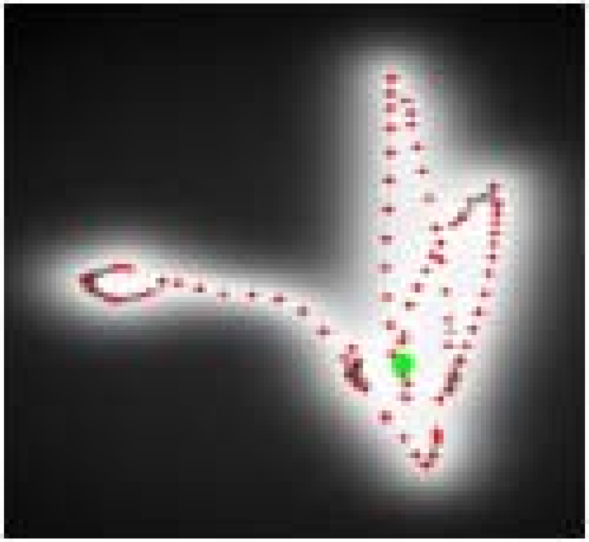
- Minimizing L_{GP} arranges $\{\mathbf{x}_i\}$, so that similar poses are nearby and dissimilar poses are far apart.

  Remember: $k(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{cov}(\mathbf{y}_i, \mathbf{y}_j)$

- Likelihood for x is high near the example poses.

SGPLVM: Synthesis (1)

For new poses $\mathbf{y}(\mathbf{q})$ the according $\mathbf{x}$ can be found by minimizing L_{IK}:



$$\min_{\mathbf{x}} L_{IK}(\mathbf{x},\mathbf{y}) = \underbrace{\frac{\left\|\mathbf{W}(\mathbf{y}-\mathbf{f}(\mathbf{x}))\right\|^2}{2\sigma^2(\mathbf{x})}}_{\text{prediction error}} + \underbrace{\frac{D}{2}\ln\sigma^2(\mathbf{x}) + \frac{1}{2}\left\|\mathbf{x}\right\|^2}_{\text{neg. likelihood of } \mathbf{x}}$$

SGPLVM: Synthesis (2)

For synthesizing new poses **y** is unknown

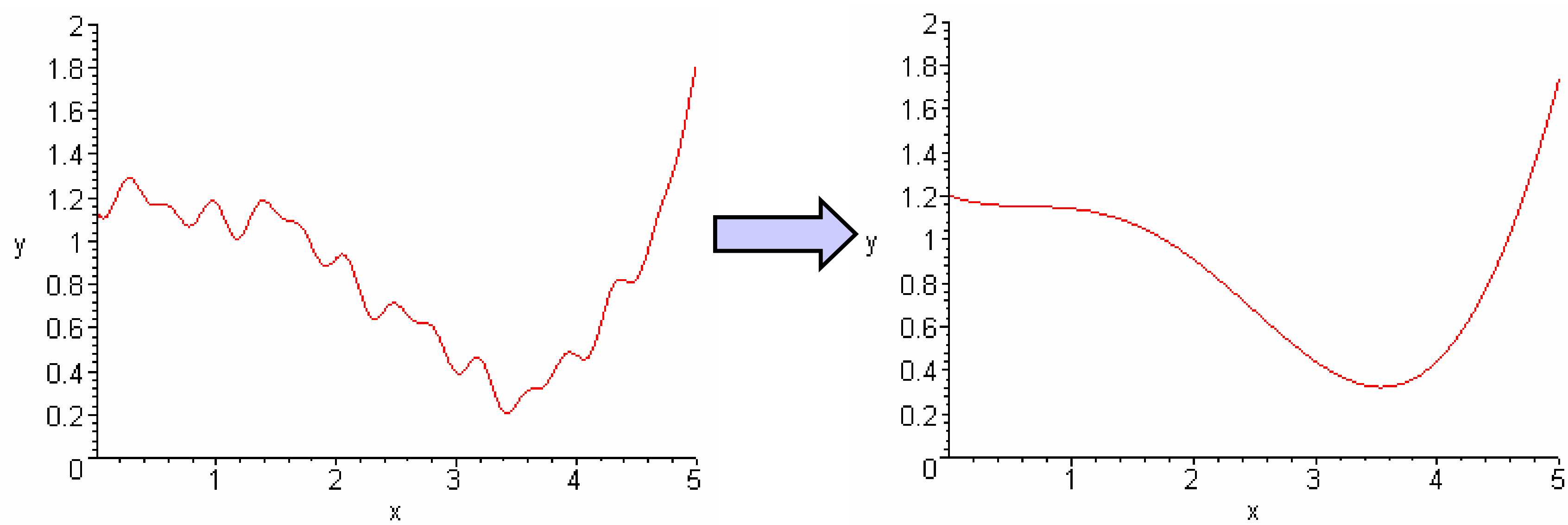
⇨ a optimize L_{IK} for **x** and **q**

$$\min_{\mathbf{x},\mathbf{q}} L_{IK}(\mathbf{x}, \mathbf{y}(\mathbf{q}))$$

$$\text{s.t.} \quad \text{Constraints}(\mathbf{q}) = 0$$

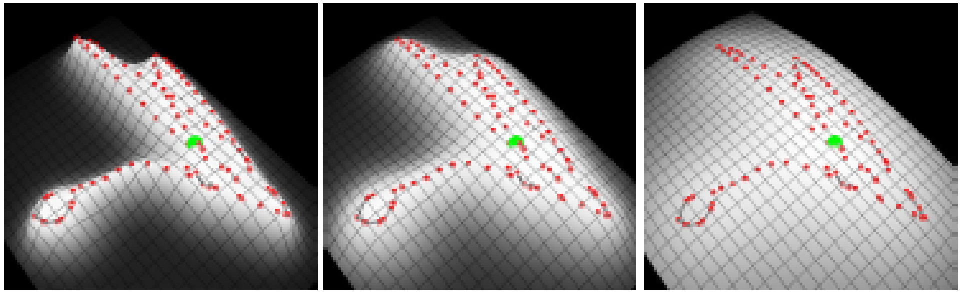
Annealing(1)

- L_{IK} is a complicated function with many local minima ⇨ the numerical optimizer may get trapped in a poor minima
- Avoid local minima by smoothing L_{IK}



Annealing(2)

- Learning:
  - use original setting:
    minimize $L_{GP} \Rightarrow \{\mathbf{x}_i\}, \{w_i\}, \alpha, \beta, \gamma$
  - Keep $\{\mathbf{x}_i\}, \{w_i\}$ fixed and add noise to $\{\mathbf{y}_i\}$:
    minimize $L_{GP} \Rightarrow \alpha', \beta', \gamma'$

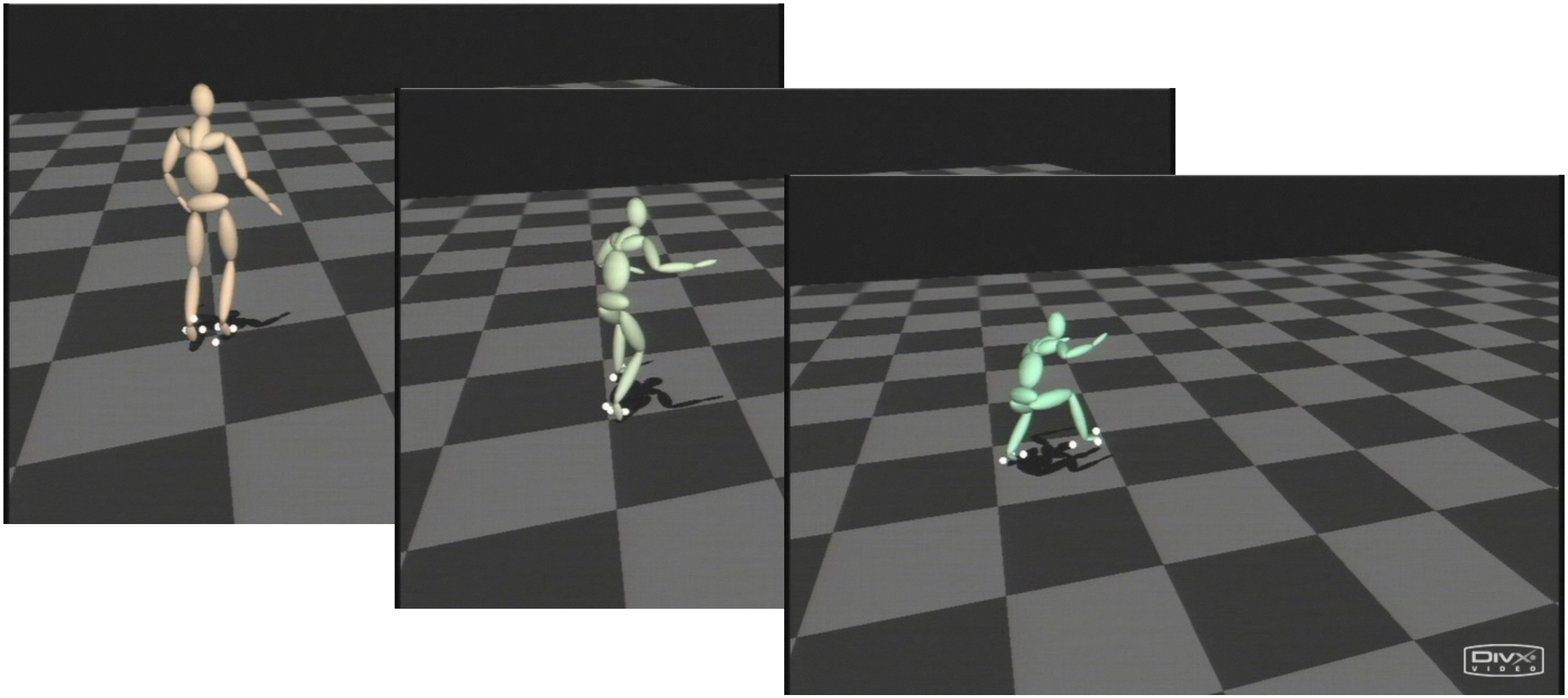


- Synthesis:
  - Optimize L_{IK}: Use α', β', γ' for first steps
  - Switch to interpolated parameters
  - Finish with using α, β, γ

Active Set

- For the optimization of L_{IK} K is needed
  ⇨ Poor scaling, because K grows quadratically with the number of learning examples.
- Simplify L_{IK} by only using a subset of all the training poses.
- The set of most representative input poses may be calculated during the learing step.

Style Interpolation



Interpolation for two styles for a given interpolation parameter s.

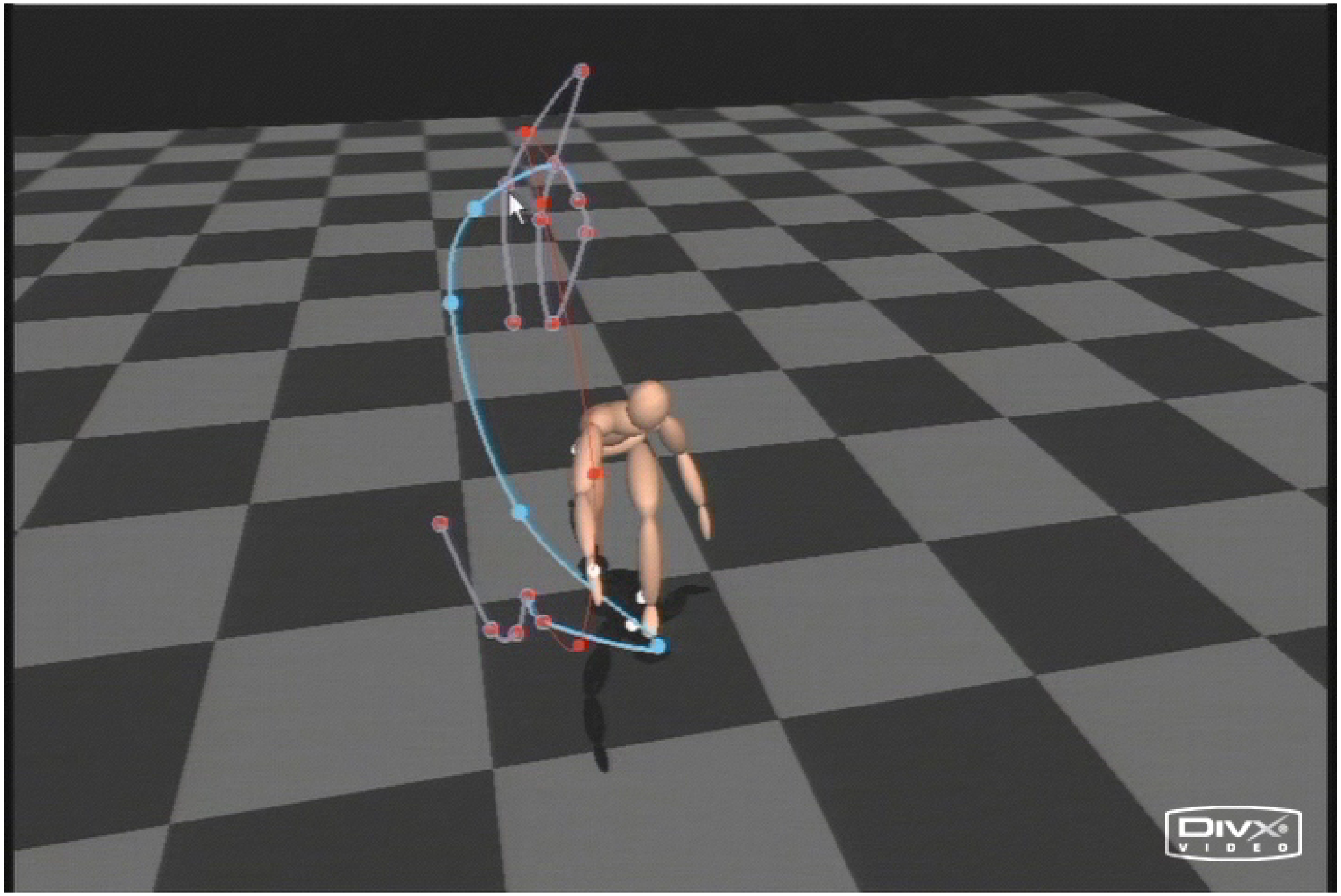
$$L_s(\mathbf{x_0},\mathbf{x_1},\mathbf{y}) = (1-s)L_{IK0}(\mathbf{x_0},\mathbf{y}) + sL_{IK1}(\mathbf{x_1},\mathbf{y})$$

Motion capturing with missing markers

Optical Motion Capturing:

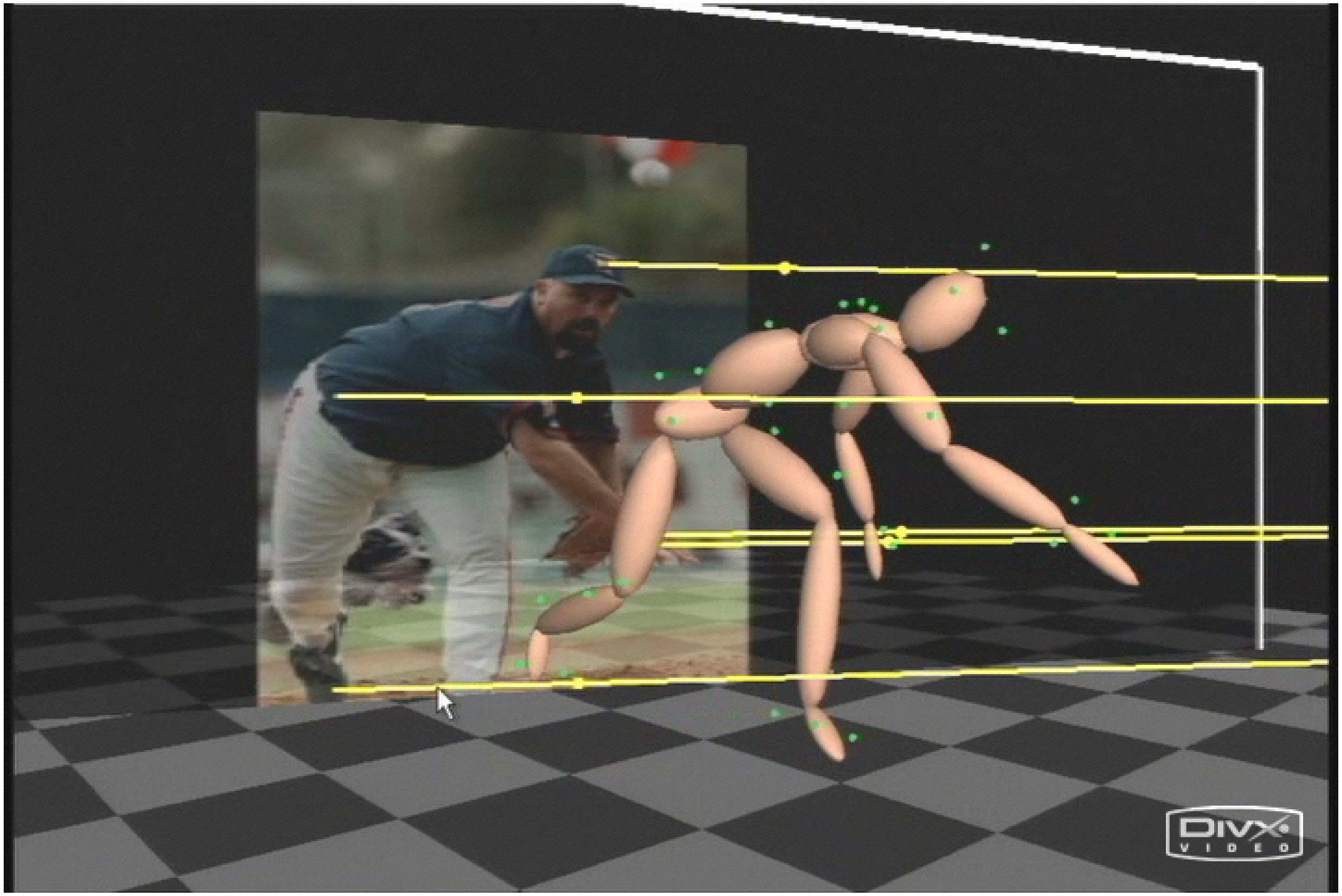
- Occluded markers can be reconstructed with style-based IK
- Works even when 50% of the markers are missing

Trajectory keyframing



Create sequence of poses by defining a trajectory for one part of the body.

Posing from 2D images



Reconstruct 3D pose from 2D image

Performance

- Computer
  - 2.8 GHz P4
- Model
  - input sequence: 500 frames
  - active set: 100 poses
  - latent space: 3D
- Performance
  - precomputation time: ?
  - synthesis: 23 fps

Conclusion(1)

Good things:

+ Realtime

+ Intuitive & easy to use

+ Usefull for different kind of problems

+ No parameter tuning (except optimizations)

Conclusion(2)

Drawbacks / possible problems:

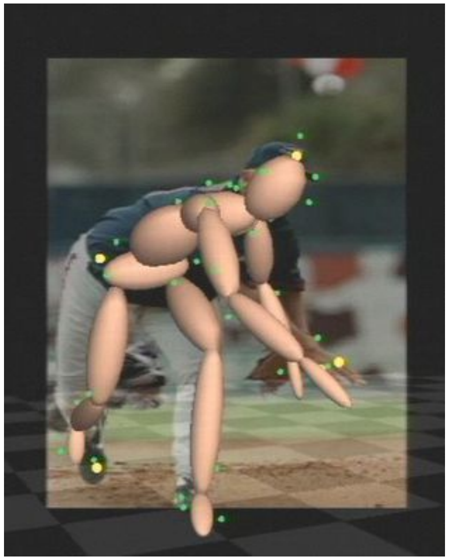
- Scaling behaviour for more complex skeletons?
- No physics
- Only indirect temporal relation between inputs

  ⇨ no keyframing
- Sensitive to input:

  How much can the input differ from the output?

Future

Now:
1 image, 1 person



?

Future:
whole film, n persons