Human Motion Analysis
Lecture 8: Shape and pose

Raquel Urtasun

TTI Chicago

May 3, 2010

Materials used for this lecture

- B. Allen, B. Curless and Z. Popovic. Articulated Body Deformation from Range Scan Data, , ACM SIGGRAPH 2002.
- B. Allen, B. Curless and Z. Popovic. The space of human body shapes: reconstruction and parameterization from range scans, ACM SIGGRAPH 2003.
- D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers. SCAPE: Shape Completion and Animation of People, ACM SIGGRAPH 2004.
- R. Urtasun, PhD. Thesis, Chapters 4, 5 and 6.
- Some slides provided by Luca Ballan.

Contents of today's lecture?

We will look into generative approaches to pose estimation. We will focus on:

- shape priors
- pose priors

The problem of human pose estimation

- The goal is given an image **I** to estimate the 3D location and orientation of the body parts **y**.

Pose estimation

- **Generative approaches:** focus on modeling

$$p(\phi|\mathbf{I}) = \frac{p(\mathbf{I}|\phi)p(\phi)}{p(\mathbf{I})}$$

- **Discriminative approaches:** focus on modeling directly

$$p(\phi|\mathbf{I})$$

Today we will talk about generative approaches.
Later in the class we will cover discriminative approaches.

Generative approaches

Generative approach models

$$p(\phi|\mathbf{I}) = \frac{p(\mathbf{I}|\phi)p(\phi)}{p(\mathbf{I})}$$

Types of generative approaches:

- **Bayesian approaches:** focus on approximating $p(\phi|\mathbf{I})$, usually via sampling (e.g., particle filter).
- **Optimization or energy-based techniques:** focus on computing the MAP or ML estimate of $p(\phi|\mathbf{I})$.

Generative approaches

Generative approach models

$$p(\phi|\mathbf{I}) = \frac{p(\mathbf{I}|\phi)p(\phi)}{p(\mathbf{I})}$$

Types of generative approaches:

- **Bayesian approaches:** focus on approximating $p(\phi|\mathbf{I})$, usually via sampling (e.g., particle filter).
- **Optimization or energy-based techniques:** focus on computing the MAP or ML estimate of $p(\phi|\mathbf{I})$.

Common to all of them is the need to model

- **Image likelihood:** $p(\mathbf{I}|\phi)$
- **Priors:** $p(\phi)$

Generative approaches

Generative approach models

$$p(\phi|\mathbf{I}) = \frac{p(\mathbf{I}|\phi)p(\phi)}{p(\mathbf{I})}$$

Types of generative approaches:

- **Bayesian approaches:** focus on approximating $p(\phi|\mathbf{I})$, usually via sampling (e.g., particle filter).
- **Optimization or energy-based techniques:** focus on computing the MAP or ML estimate of $p(\phi|\mathbf{I})$.

Common to all of them is the need to model

- **Image likelihood:** $p(\mathbf{I}|\phi)$
- **Priors:** $p(\phi)$

In general $p(\mathbf{I})$ is assumed constant and ignored. The different trackers then depend on the different modeling choices and optimization procedures.

Generative approaches

Generative approach models

$$p(\phi|\mathbf{I}) = \frac{p(\mathbf{I}|\phi)p(\phi)}{p(\mathbf{I})}$$

Types of generative approaches:

- **Bayesian approaches:** focus on approximating $p(\phi|\mathbf{I})$, usually via sampling (e.g., particle filter).
- **Optimization or energy-based techniques:** focus on computing the MAP or ML estimate of $p(\phi|\mathbf{I})$.

Common to all of them is the need to model

- **Image likelihood:** $p(\mathbf{I}|\phi)$
- **Priors:** $p(\phi)$

In general $p(\mathbf{I})$ is assumed constant and ignored. The different trackers then depend on the different modeling choices and optimization procedures.

In the next lectures we will look at ...

Priors: $p(\phi)$

- Joint limits
- Shape priors
- Pose priors
- Dynamical priors
- Physics

Likelihood models: $p(\mathbf{I}|\phi)$

- Monocular tracking: 2D-3D correspondences, silhouettes, edges, template matching, etc.
- Multi-view tracking: stereo, visual hull, etc.

In the next lectures we will look at ...

Priors: $p(\phi)$

- Joint limits
- Shape priors
- Pose priors
- Dynamical priors
- Physics

Likelihood models: $p(\mathbf{I}|\phi)$

- Monocular tracking: 2D-3D correspondences, silhouettes, edges, template matching, etc.
- Multi-view tracking: stereo, visual hull, etc.

Note that I have defined ϕ as a general quantity, not just the pose.

In the next lectures we will look at ...

Priors: $p(\phi)$

- Joint limits
- Shape priors
- Pose priors
- Dynamical priors
- Physics

Likelihood models: $p(\mathbf{I}|\phi)$

- Monocular tracking: 2D-3D correspondences, silhouettes, edges, template matching, etc.
- Multi-view tracking: stereo, visual hull, etc.

Note that I have defined ϕ as a general quantity, not just the pose.

Shape representations

Shape Primitives

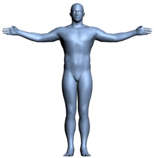
3D Sticks

3D Cylinders

2D Primitives

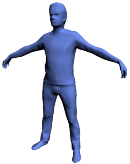
...

Meshes



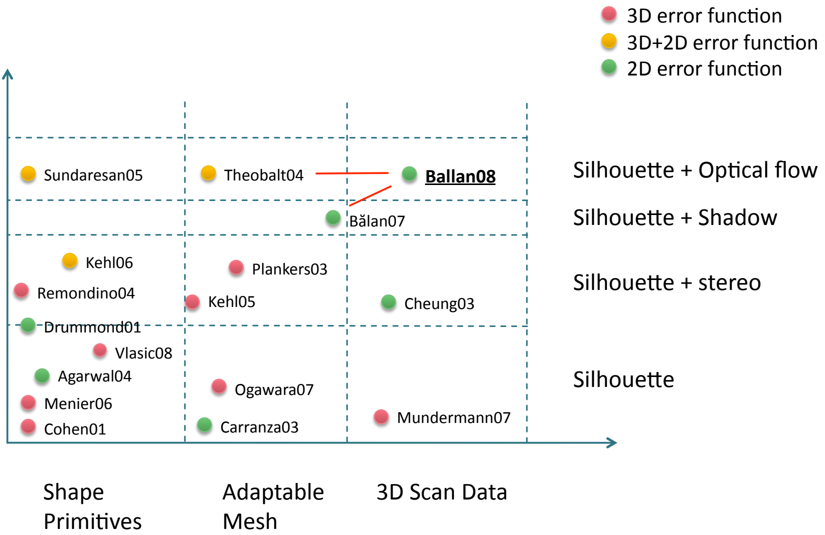
Generic Adaptable Model

- Modify length/size of each limb
- Retrieve from a database (SCAPE)



3D Scan data

Likelihood vs shape

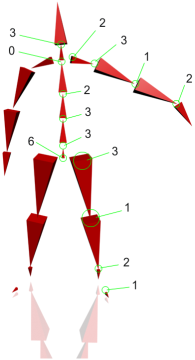


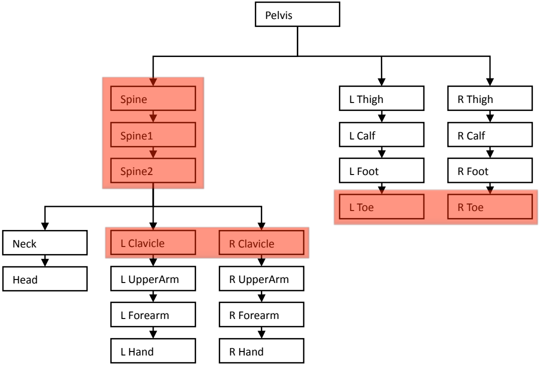
Shape representations cover in the class

- Skeleton
- Simple primitives: cylinders, cones, truncated cones, ellipsoids
- Superquadrics
- Implicit surfaces
- Scan mesh
- Allen et al. models
- SCAPE model

Skeleton representation

- Human body as a kinematic tree, where joints are connected by segments of fix length.
- Simplest representation.





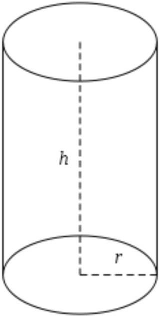
Simple primitives I

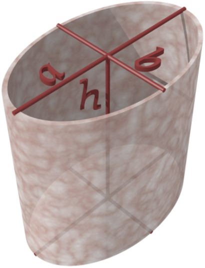
- A **cylinder** can be expressed as

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{a}\right)^2 = 1$$

- An **elliptic cylinder** can be expressed as

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$$





Simple primitives II

- A **cone** is a three-dimensional geometric shape that tapers smoothly from a flat, usually circular base to a point called the apex or vertex
- A cone with its apex cut off by a plane parallel to its base is called a **truncated cone** or **frustum**.

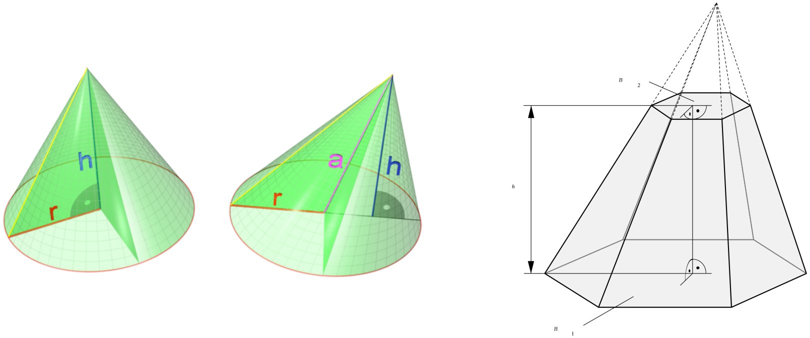


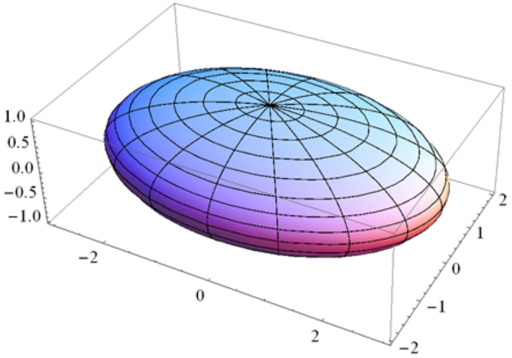
Figure: (Left) Right circular cone. (Center) Oblique circular cone. (Right) frustum.

Simple primitives III

- An **ellipsoid** is a type of quadric surface that is a higher dimensional analogue of an ellipse

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

- If the axis are not aligned, it's represented as $\mathbf{x}\mathbf{A}\mathbf{x}^T = 1$



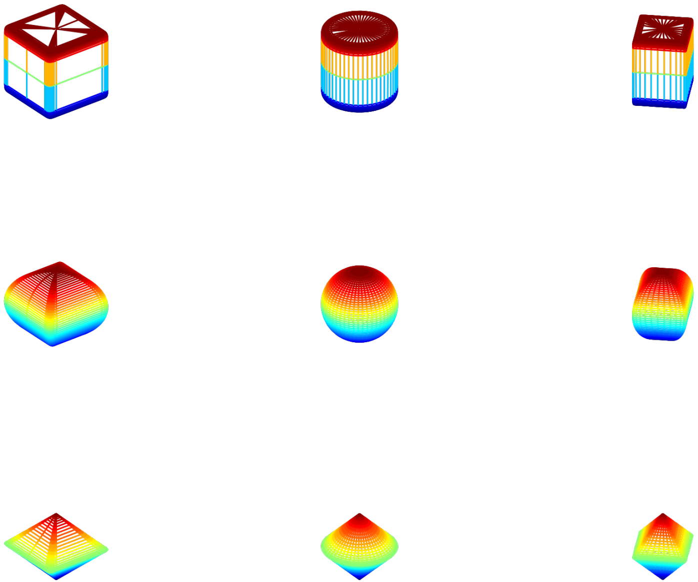
Superquadrics

- Superquadrics are a family of geometric shapes defined by formulas that resemble those of elipsoids and other quadrics, except that the squaring operations are replaced by arbitrary powers.

$$\frac{|x|^r}{a} + \frac{|y|^s}{b} + \frac{|z|^t}{c} \leq 1$$

  with $r, s, t \in \Re^+$, and $a, b, c \in \Re$.
- The superquadrics include many shapes that resemble cubes, octahedra, cylinders, lozenges and spindles, with rounded or sharp corners.
- Superellipsoids are a special case when $r = s = t$.

Superquadrics II



Superquadrics representing humans

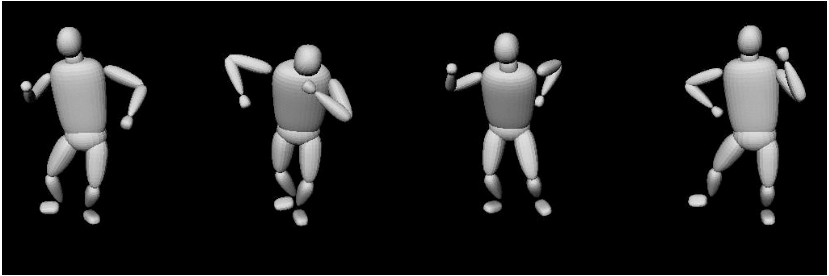


Figure: humans represented using superquadrics (Sminchisescu03)

Implicit surfaces

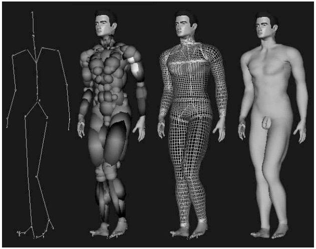
- The skin metaball surface $\mathcal{S}$ is a generalized algebraic surface that is defined as a level set of the summation over n 3D densities of primitives

$$F(x,y,z) = \sum_{i=1}^{n} f_i(x,y,z) \quad \text{with} \quad f_i(x,y,z) = \exp(-2d_i(x,y,z))$$

with d_i the distance to the i-th primitive.

- The implicit surface is defined by the level set

$$\mathcal{S} = \{[x,y,z] \in \Re^3 | F(x,y,z) = L\}$$



SCAN

Home-made 3D Body Scanner
(**< 2000 Euro**)

Shape: Silhouettes + Stereo
Texture: Wavelet blending

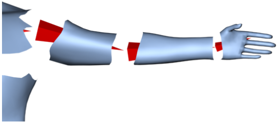
Shape: 500k faces -> 13k faces
Texture: 6000x3500 pixels

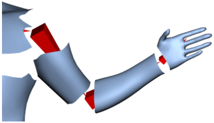
Textured model

Wireframe model

Deformations

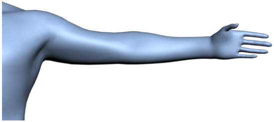
Split the surface in small pieces which moves rigidly attached each to only one bone

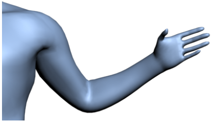




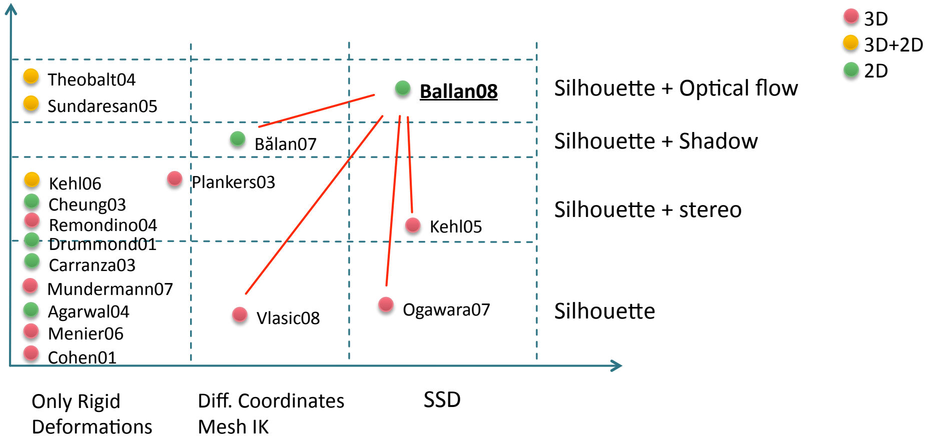
Deal with non-rigid deformation

- Skeletal Subspace Deformation
- Pose space deformation





Likelihood vs deformation



Skinning or Skeleton-Subspace Deformation (SSD)

- The position of a control vertex $\mathbf{v}_j$ on the deforming surface of an articulated object lies in the subspace defined by the rigid transformations of that point

$$\hat{\mathbf{v}}_j = \sum \alpha_{j,k} L_k(\mathbf{v}_j)\mathbf{v}_j = \sum \alpha_{j,k} L_k^{\delta} (L_k^0)^{-1} L_p^0 \mathbf{v}_j$$

where L_p^0 is the transform from the surface containing $\mathbf{v}_j$ to the world system, L_k^0 is the transform from the stationary skeletal frame k to the world system, and L_k^{δ} expresses the moving skeletal frame k in the world system.

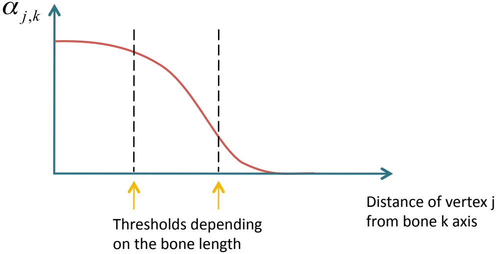
Skinning or Skeleton-Subspace Deformation (SSD)

- The position of a control vertex $\mathbf{v}_j$ on the deforming surface of an articulated object lies in the subspace defined by the rigid transformations of that point

$$\hat{\mathbf{v}}_j = \sum \alpha_{j,k} L_k(\mathbf{v}_j)\mathbf{v}_j = \sum \alpha_{j,k} L_k^{\delta}(L_k^0)^{-1} L_p^0 \mathbf{v}_j$$

  where L_p^0 is the transform from the surface containing $\mathbf{v}_j$ to the world system, L_k^0 is the transform from the stationary skeletal frame k to the world system, and L_k^{δ} expresses the moving skeletal frame k in the world system.

- The matrix $\alpha = \{\alpha_{j,k}\}$ is the normalized non-linear distance between the vertex j and the bone k axis



Pose space deformation (PSD)

- In SSD the deformation is restricted to the indicated subspace. Extreme in the case of twist.

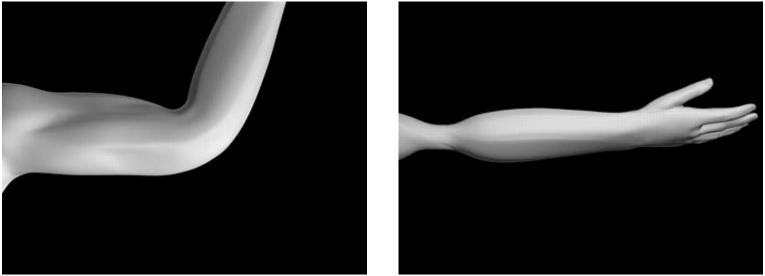


Figure: Problems of SSD (Lewis et al. 03)

Pose space deformation (PSD)

- In SSD the deformation is restricted to the indicated subspace. Extreme in the case of twist.
- SSD does not permit direct manipulation
- The solution of PSD is the identification of an appropriate space for defining deformations.
- The deformation is defined as

$$\bar{\mathbf{v}}_j = \mathbf{v}_j + f_{interp}(joints, parameters)$$

Comparison SSD vs PSD

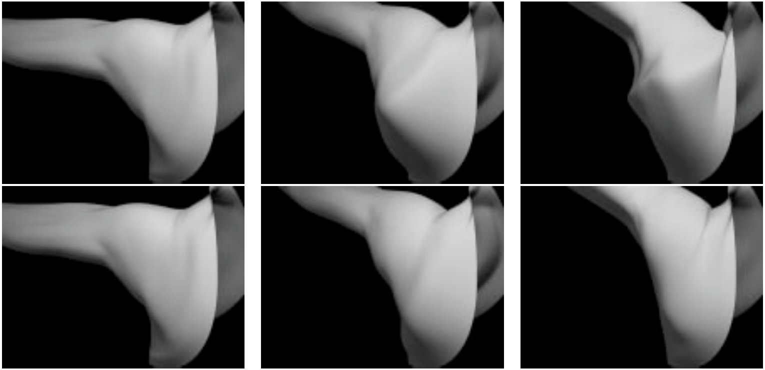


Figure: Comparison of (Top) SSD with (Bottom) PSD. (Lewis et al. 03)

Comparison SSD vs PSD

Figure: Comparison of (Left) SSD with (Right) PSD. (Lewis et al. 03)

Articulated Body Deformations from Range Scan Data

- GOAL: body parts are scanned in a set of key poses, and then animations are generated by smoothly interpolating among these poses using scattered data interpolation techniques.

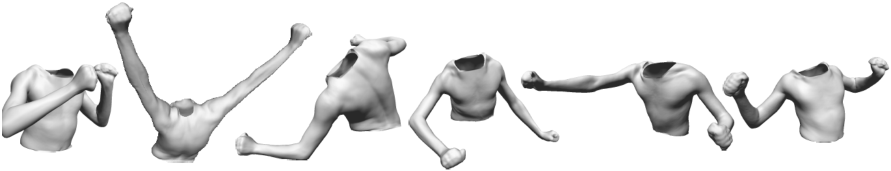


Figure: Articulated Body Deformations from Range Scan Data (Allen et al. 02)

Problems of Articulated Deformations from Scan Data

- To create compelling animations by observation we need more than just a single scan.
- In order to establish a domain for interpolation, we must discover the pose of each scan.
- Interpolation techniques require a one-to-one correspondence between points on the scanned surfaces, but the scanned data consists of unstructured meshes with no such correspondence.
- Range scans are frequently incomplete because of occlusions and grazing angle views. Thus, we are faced with the challenge of filling holes in the range data.
- Due to the combinatorics of the problem, we cannot capture a human body in every possible pose. Thus, we must blend between independently posed scans.

Approach of Allen et al 02

- Using markers placed on the subject during range scanning, we reconstruct the pose of each scan.
- We then create a hole-filled, parameterized reconstruction at each pose using displacement-mapped subdivision surfaces.
- Lastly, we create shapes in new poses using scattered data interpolation and spatially varying surface blending.

Determining the pose

- A skeleton is fitted by first identifying the markers and then minimizing

$$\min_{\mathbf{m},\mathbf{q},\mathbf{k}} \sum_{i=1}^{P} \sum_{j=1}^{m} ||\mathbf{o}_{ij} - \mathbf{c}_j(\mathbf{m}_j, \mathbf{q}_i, \mathbf{k})||_2^2$$

with $\mathbf{c}_j$ the estimated position of the markers, $\mathbf{o}_{ij}$ the observed position, $\mathbf{m}_j$ is the local position, $\mathbf{q}_i$ is the pose, and $\mathbf{k}$ are the kinematics.

Determining deformations

- Create a subdivision surface that approximates the real surface

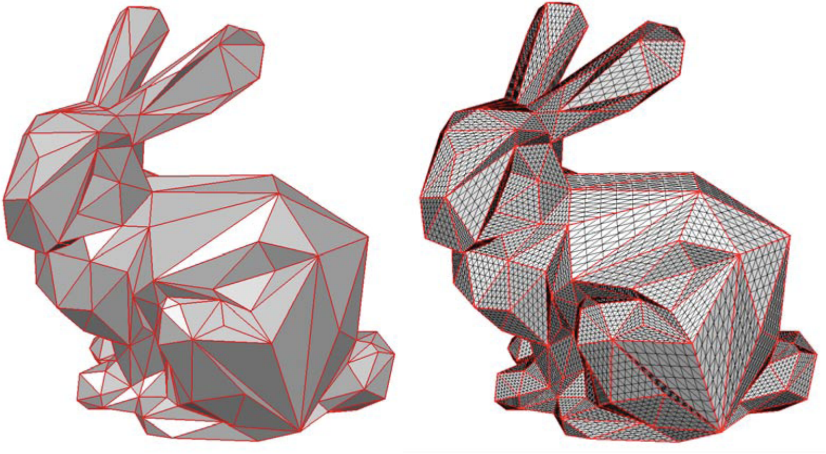


Figure: Displaced Subdivision Surfaces (Lee et al. 00)

Determining deformations

- Create a subdivision surface that approximates the real surface
- **Displaced subdivision surfaces** consist of a template subdivision surface, T, and a displacement map d that describes the final surface S by displacing the template along the normal, $\mathbf{n}$, to the template surface

$$S(\mathbf{u}, \mathbf{q}) = T(\mathbf{u}, \mathbf{q}) + d(\mathbf{u}, \mathbf{q})\mathbf{n}(\mathbf{u}, \mathbf{q})$$

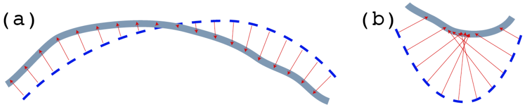


Figure: Displaced Subdivision Surfaces (Allen et al. 02)

Determining deformations

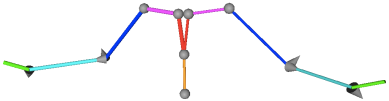
- Create a subdivision surface that approximates the real surface
- **Displaced subdivision surfaces** consist of a template subdivision surface, T, and a displacement map d that describes the final surface S by displacing the template along the normal, $\mathbf{n}$, to the template surface

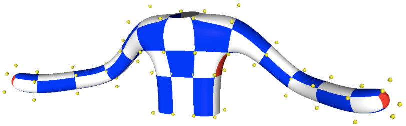
$$S(\mathbf{u}, \mathbf{q}) = T(\mathbf{u}, \mathbf{q}) + d(\mathbf{u}, \mathbf{q})\mathbf{n}(\mathbf{u}, \mathbf{q})$$

- Unlike standard displaced subdivision surfaces, the displacements are based on multiple example shapes

$$d(\mathbf{u}, \mathbf{q}) = \sum_{i=1}^{n} w_i(\mathbf{u}, \mathbf{q}) d_i(\mathbf{u})$$

  with $d_i(\mathbf{u})$ the displacement map of the i-th example, $w_i(\mathbf{u}, \mathbf{q})$ the scattered data interpolation weights





Determining deformations

- Create a subdivision surface that approximates the real surface
- **Displaced subdivision surfaces** consist of a template subdivision surface, T, and a displacement map d that describes the final surface S by displacing the template along the normal, $\mathbf{n}$, to the template surface

$$S(\mathbf{u},\mathbf{q}) = T(\mathbf{u},\mathbf{q}) + d(\mathbf{u},\mathbf{q})\mathbf{n}(\mathbf{u},\mathbf{q})$$

- Unlike standard displaced subdivision surfaces, the displacements are based on multiple example shapes

$$d(\mathbf{u},\mathbf{q}) = \sum_{i=1}^{n} w_i(\mathbf{u},\mathbf{q}) d_i(\mathbf{u})$$

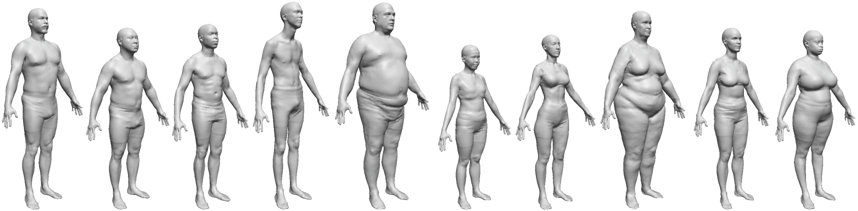
  with $d_i(\mathbf{u})$ the displacement map of the i-th example, $w_i(\mathbf{u},\mathbf{q})$ the scattered data interpolation weights

- Hole filling in 3D and refitting.

Results: Interpolation between novel poses

Space of human body shapes (Allen et al. 03)

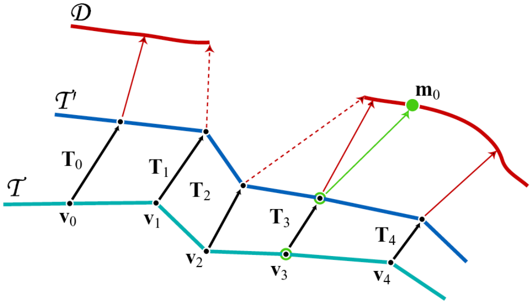
- Use the CAESAR dataset.



- Aligned the meshes to a template by using local affine transformations of each template vertex.
- Use an objective function that is the combination of smoothness, alignment and markers that help avoid local minima.
- Applications: model the space of shapes (PCA), texture transfer, etc.

Local parameterization and matching I

- Fit the template surface $\mathcal{T}$ to a scanned surface $\mathcal{D}$, each represented with a triangular mesh.
- We assume that each vertex $\mathbf{v}_i$ in the template can suffer an affine transformation $\mathbf{T}_i \in \Re^{4\times4}$.
- This results in 12 dof per vertex.
- We wish to find the set of transformations that move all points in $\mathcal{T}$ to $\mathcal{T}'$, so that $\mathcal{T}'$ is close to $\mathcal{D}$.



Local parameterization and matching II

- We solve for the local transformation by $\min_{\mathbf{T}} \alpha E_d + \beta E_s + \gamma E_m$
- The **data error** E_d is defined as

$$E_d = \sum_{i=1}^{n} w_i dist^2(\mathbf{T}_i \mathbf{v}_i, \mathcal{D})$$

  with *dist* the distance between a transformed vertex $\mathbf{T}_i \mathbf{v}_i$ and a mesh $\mathcal{D}$.
- The **smoothness error** E_s is computed as

$$E_s = \sum_{i,j \in \mathcal{E}} ||\mathbf{T}_i - \mathbf{T}_j||_F^2$$

  where $|| \cdot ||_F$ is the Frobenious norm, and $\mathcal{E}$ is the set of neighboring vertices.
- The **marker error** E_m is

$$E_m = \sum_{i=1}^{m} ||\mathbf{T}_{\kappa_i} \mathbf{v}_{\kappa_i} - \mathbf{m}_i||_2^2$$

  with $\mathbf{m}_i$ the position of the observed markers.

Local parameterization and matching II

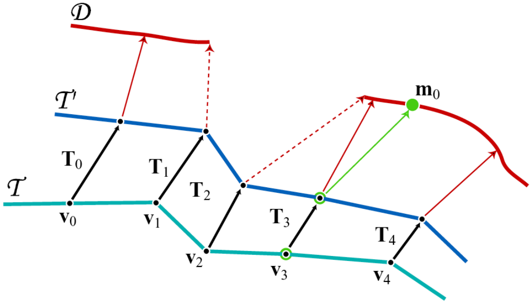
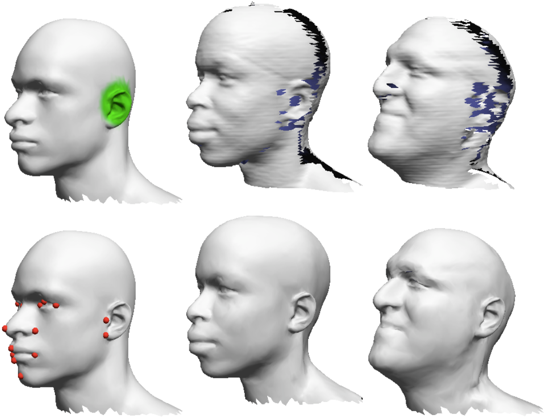


Figure: The data error, indicated by the red arrows. The dashed red arrows do not contribute to the data error because the nearest point on $\mathcal{D}$ is a hole boundary. The marker error penalizes distance between the marker points on the transformed surface and on $\mathcal{D}$ (here $\mathbf{v}_3$ is associated with m_0). (Allen et al. 03)

Hole filling

- Robust estimator that uses 0 weight for holes and outliers, only smoothness is used. Use a confidence value fort he matching
- The user specifies regions difficult to match, e.g., ear. The system favors the template over those areas.



Applications: Texture transfer

- Because the parameterization is consistent we can transfer texture.

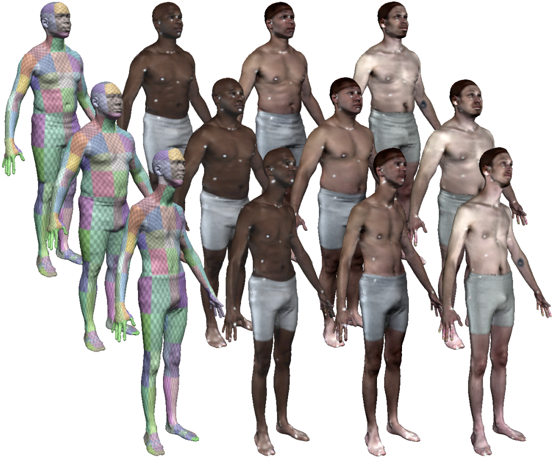


Figure: Allen et al 03

Applications: Morphing

- We can morph between any two subjects by taking linear combinations of the vertices.

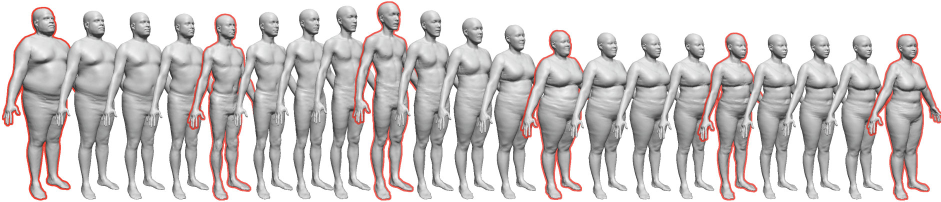


Figure: Allen et al 03

Applications: Shape matching

- Shape model is created using PCA.
- The basis are used to fit new shape.

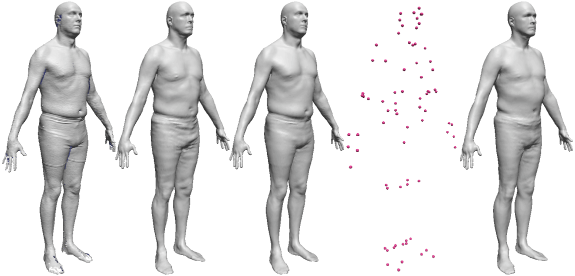


Figure: A scanned mesh that was not included in the data set previously, and does not resemble any of the other scans. (b) A surface match using PCA weights and no marker data. (c) Using (b) as a template surface, we get a good match to the surface using our original method without markers. (d) Next, we demonstrate using very sparse data; in this case, only the 74 marker points. (e) A surface match using PCA weights and no surface data (Allen et al 03)

Applications: Skeleton transfer

- Manually create a skeleton and skinning for one character, and automatically transfer the skeleton

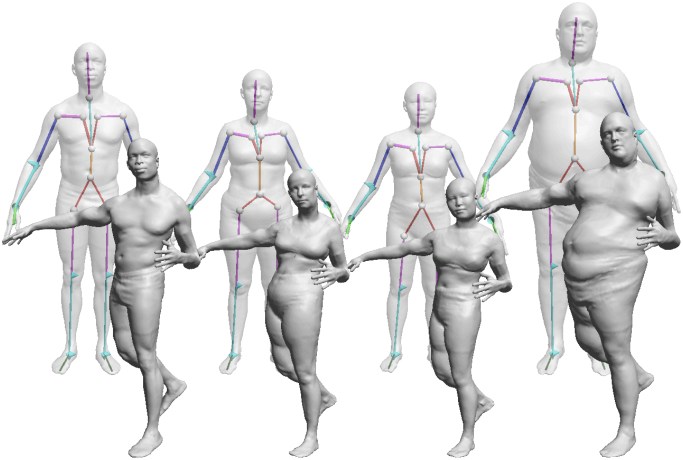


Figure: Allen et al 03

Applications: Feature based synthesis

- Principal component analysis helps to characterize the space of human body variation, but it does not provide a direct way to explore the range of bodies with intuitive controls, such as height, weight, age, and sex.
- We relate several variables simultaneously by learning a linear mapping between the controls and the PCA weight.

$$\mathbf{M}[f_1, \cdots, f_l, 1]^T = \mathbf{p}$$

  with f_i are feature values of an individual, and $\mathbf{p}$ are the corresponding PCA weights.
- Assembling all the feature together and solving the linear system we have

$$\mathbf{M} = \mathbf{P}\mathbf{F}^\dagger$$

  with $\mathbf{F}^\dagger$ the pseudoinverse of $\mathbf{F}$.
- By adding $\Delta\mathbf{p} = \mathbf{M}\Delta\mathbf{f}$ to the PCA weights of that individual, we can edit their features, e.g., making them gain or lose weight.

Applications: Feature based synthesis

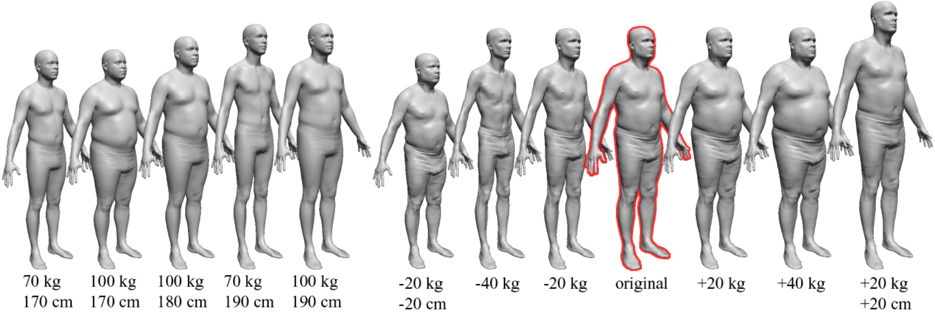


Figure: The left part of this figure demonstrates feature-based synthesis, where an individual is created with the required height and weight. On the right, we demonstrate feature-based editing. The outlined figure is one of the original subjects, after being parameterized into our system. The gray figures demonstrate a change in height and/or weight. Allen et al 03

Video

SCAPE model

- SCAPE: Shape Completion and Animation for PEople (Angelov et al. 04).
- A data driven approach for building two models: **pose** and **shape**.
- The models of shape and pose can be combined to produce 3D surface models with realistic muscle deformations of different people in different poses.
- The **pose deformation** component of our model is acquired from a set of dense 3D scans of a single person in multiple poses.
- Deformation is decoupled into rigid (skeleton) and non-rigid components (e.g., flexing of the muscles).
- The deformation is local and only depends on adjacent body parts, and thus remains low-dimensional.

SCAPE model

- SCAPE: Shape Completion and Animation for PEople (Angelov et al. 04).
- A data driven approach for building two models: **pose** and **shape**.
- The models of shape and pose can be combined to produce 3D surface models with realistic muscle deformations of different people in different poses.
- The **pose deformation** component of our model is acquired from a set of dense 3D scans of a single person in multiple poses.
- Deformation is decoupled into rigid (skeleton) and non-rigid components (e.g., flexing of the muscles).
- The deformation is local and only depends on adjacent body parts, and thus remains low-dimensional.
- The **shape variation** is acquired from a set of 3D scans of different people in different poses.
- This shape variation is represented in terms of PCA, and it does not model deformations due to pose.

SCAPE model

- SCAPE: Shape Completion and Animation for PEople (Angelov et al. 04).
- A data driven approach for building two models: **pose** and **shape**.
- The models of shape and pose can be combined to produce 3D surface models with realistic muscle deformations of different people in different poses.
- The **pose deformation** component of our model is acquired from a set of dense 3D scans of a single person in multiple poses.
- Deformation is decoupled into rigid (skeleton) and non-rigid components (e.g., flexing of the muscles).
- The deformation is local and only depends on adjacent body parts, and thus remains low-dimensional.
- The **shape variation** is acquired from a set of 3D scans of different people in different poses.
- This shape variation is represented in terms of PCA, and it does not model deformations due to pose.

SCAPE acquisition pipeline

- 2 datasets are generated: 70 poses of a particular subject for the pose, and $37 + 8$ people for the shape.
- The meshes are hole-filled
- One of the meshes in the pose data is selected as template mesh.
- To put the mesh into correspondences a small number of markers (4-10) are hand specified.
- An algorithm of Correlated Correspondences which minimizes deformation and matches similar-looking surface regions is used to create additional correspondences (140-200).
- These makers are used to bring the mesh into correspondences.

SCAPE acquisition pipeline

- 2 datasets are generated: 70 poses of a particular subject for the pose, and $37 + 8$ people for the shape.
- The meshes are hole-filled
- One of the meshes in the pose data is selected as template mesh.
- To put the mesh into correspondences a small number of markers (4-10) are hand specified.
- An algorithm of Correlated Correspondences which minimizes deformation and matches similar-looking surface regions is used to create additional correspondences (140-200).
- These makers are used to bring the mesh into correspondences.
- A skeleton is recovered using the fact that vertices of the same skeleton are spatially contiguous, and have similar motions across scans.
- The location of the rigid parts and the articulated skeleton is recovered.

SCAPE acquisition pipeline

- 2 datasets are generated: 70 poses of a particular subject for the pose, and 37 + 8 people for the shape.
- The meshes are hole-filled
- One of the meshes in the pose data is selected as template mesh.
- To put the mesh into correspondences a small number of markers (4-10) are hand specified.
- An algorithm of Correlated Correspondences which minimizes deformation and matches similar-looking surface regions is used to create additional correspondences (140-200).
- These makers are used to bring the mesh into correspondences.
- A skeleton is recovered using the fact that vertices of the same skeleton are spatially contiguous, and have similar motions across scans.
- The location of the rigid parts and the articulated skeleton is recovered.

SCAPE acquisition pipeline

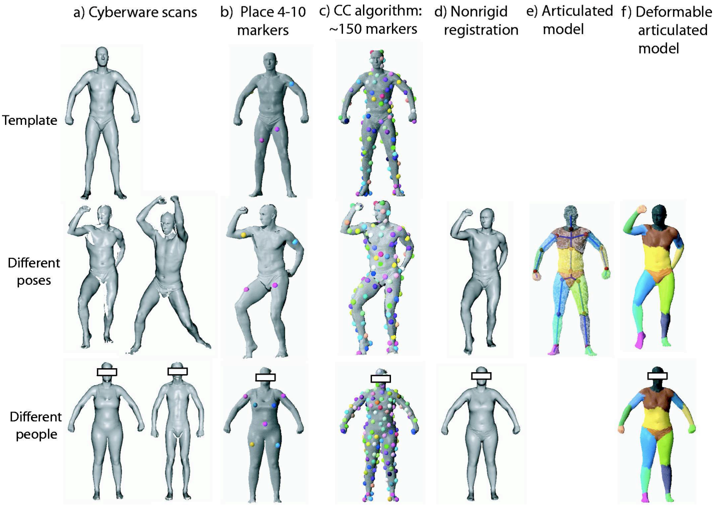
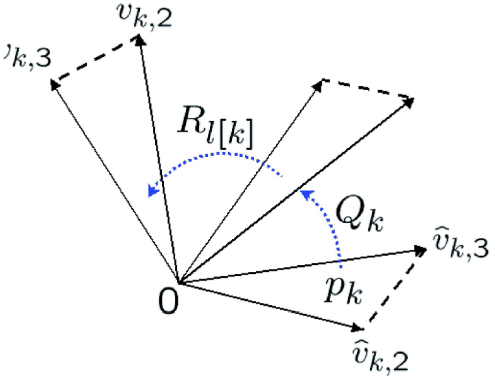


Figure: Angelov et al. 04

SCAPE local parameterization

- Let triangle $\mathbf{p}_k = [x_{k,1}, x_{k,2}, x_{k,3}]$. Deformations are apply to local coordinates of the triangle $\hat{v}_{k,j} = x_{k,j} - x_{k,1}$.
- Every triangle can have a linear transformation $\mathbf{Q}_k^i \in \Re^{3\times3}$, which induces a non-rigid pose deformation, and a rotation $\mathbf{R}_l^i$, which is constant for all the points that belong to body part l.



SCAPE pose deformation

- The local deformation matrices are learned by solving

$$\min_{\mathbf{Q}_1^i,\cdots,\mathbf{Q}_P^i} = \sum_k \sum_{j=2}^{3} ||\mathbf{R}_{l(k)}^i \mathbf{Q}_k^i \hat{v}_{j,k} - v_{j,k}^i||_2^2 + \alpha \sum_{j,k \in \mathcal{E}} \delta_{l(j),l(k)} ||\mathbf{Q}_j^i - \mathbf{Q}_k^i||_2^2$$

- The pose deformation model is learned by learning a linear regressor from the difference of twist of the two adjacent joints to the transformation matrices $\mathbf{Q}_k^i$.
- Given the $\mathbf{R}_l^i$ and the transformations $\mathbf{Q}_k^i$ obtained from the regressor, a mesh can be synthesized by solving

$$\min_{\mathbf{y}_1,\cdots\mathbf{y}_m} \sum_k \sum_{j=2}^{3} ||\mathbf{R}_{l(k)}^i \mathbf{Q}_k^i \hat{v}_{j,k} - (y_{j,k} - y_{1,k})||_2^2$$

with $\mathbf{y}_j$ the j-th point in the mesh.

Poses modeled with SCAPE

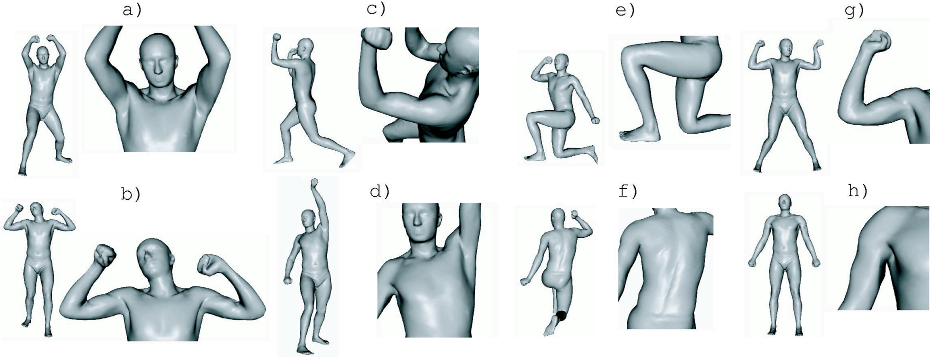


Figure: Examples of poses captured with Angelov et al. 04

Modeling body shape deformations

- The body shape deformation is modeled using an additional deformation matrix $\mathbf{S}_k^i$ such that a vertex can be computed as

$$v_{j,k}^i = \mathbf{R}_{l(k)}^i \mathbf{S}_k^i \mathbf{Q}_k^i \hat{v}_{j,k}$$

- The deformations are learned by minimizing

$$\min_{S^i} \sum_k \sum_{j=2}^{3} ||\mathbf{R}_{l(k)}^i \mathbf{S}_k^i \mathbf{Q}_k^i \hat{v}_{j,k} - v_{j,k}^i||_2^2 + \gamma \sum_{j,k \in \mathcal{E}} \delta_{l(j),l(k)} ||\mathbf{S}_j^i - \mathbf{S}_k^i||_2^2$$

- A model of shape deformations $\mathbf{S}^i \in \Re^{9 \times N}$ is learned using PCA, such that

$$\mathbf{S}^i = \mathbf{U}\beta^i + \boldsymbol{\mu}$$

- Recall that in the pose deformation model $\mathbf{R}^i$ and $\mathbf{Q}_k^i$ have already been estimated.

Shapes modeled with SCAPE

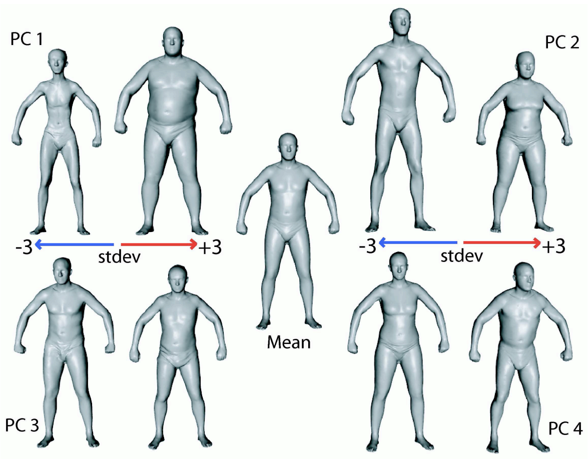
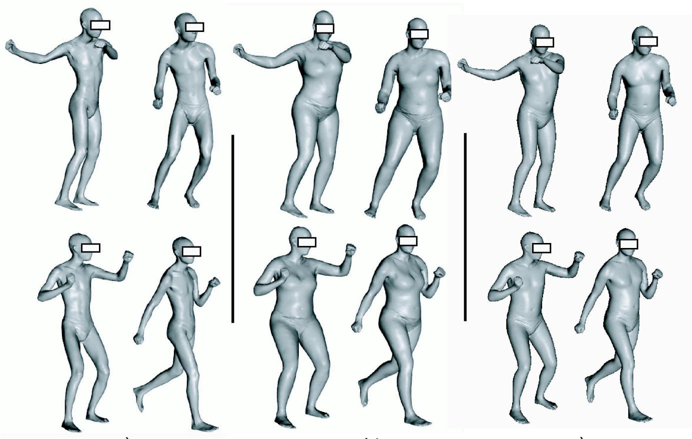


Figure: Examples of shapes captured with Angelov et al. 04

Deformation transfer

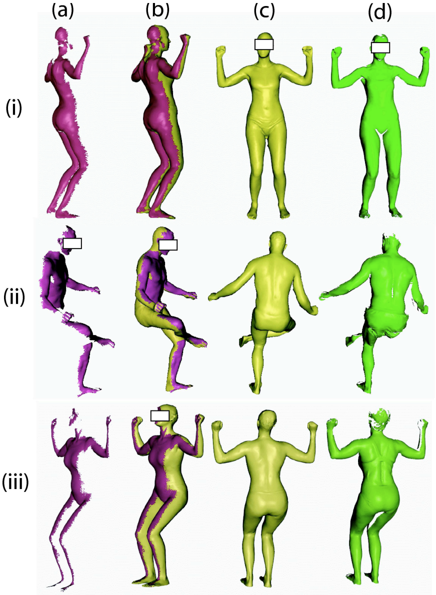
- Given the rotations $\mathbf{R}$ and the coefficients $\boldsymbol{\beta}$, we can solve for a mesh

$$\min_{\mathbf{y}_1,\cdots\mathbf{y}_m} \sum_k \sum_{j=2}^{3} ||\mathbf{R}_{l(k)}\mathbf{S}_k(\beta)\mathbf{Q}_k(\mathbf{R})\hat{v}_{j,k} - (y_{j,k} - y_{1,k})||_2^2$$

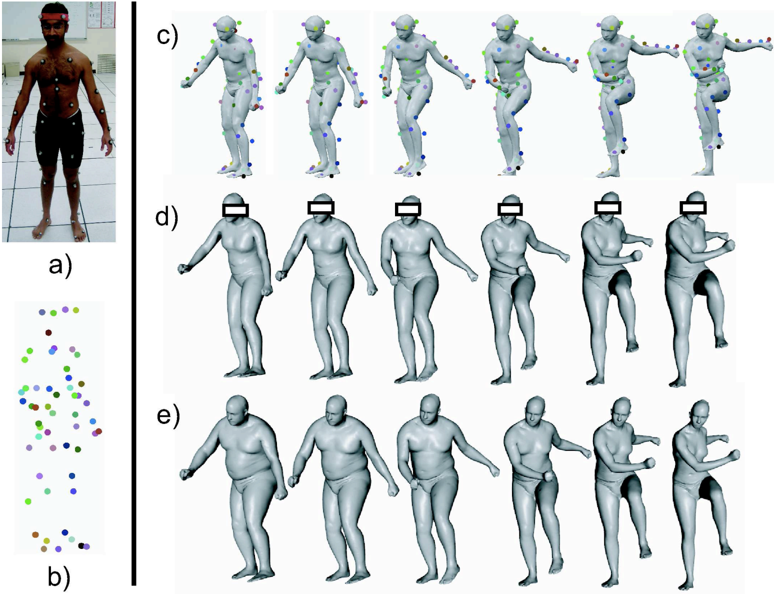


a) b) c)

Applications: Shape completion



Applications: animation from mocap



Video

Generative tracking

Priors: $p(\phi)$

- Joint limits
- Shape priors
- Pose priors
- Dynamical priors
- Physics

Pose priors

Briefly described pose priors based on dimensionality reduction

- Linear priors: PCA
- Non linear priors: GPLVM

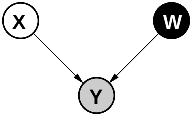
Briefly described motion priors

- Non linear models: GPDM
- Spatio-temporal linear models

Linear pose priors: PCA

Probabilistic PCA

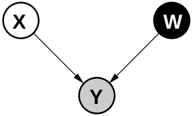
- Linear-Gaussian relationship between latent variables and data.
- **X** are 'nuisance' variables.



$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

Probabilistic PCA

- Linear-Gaussian relationship between latent variables and data.
- **X** are 'nuisance' variables.
- Latent variable model approach:

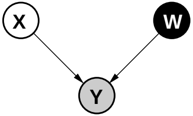


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

Linear pose priors: PCA

Probabilistic PCA

- Linear-Gaussian relationship between latent variables and data.
- $\mathbf{X}$ are 'nuisance' variables.
- Latent variable model approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.

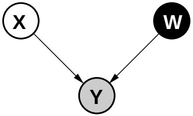


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

Linear pose priors: PCA

Probabilistic PCA

- Linear-Gaussian relationship between latent variables and data.
- **X** are 'nuisance' variables.
- Latent variable model approach:
  - Define Gaussian prior over *latent space*, **X**.
  - Integrate out nuisance *latent variables*.



$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{X}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

Linear pose priors: PCA

Probabilistic PCA

- Linear-Gaussian relationship between latent variables and data.
- $\mathbf{X}$ are 'nuisance' variables.
- Latent variable model approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out nuisance *latent variables*.

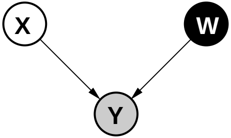
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{X}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0},\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}\right)$$

Probabilistic PCA Solution

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999b)



$$p(\mathbf{Y}|\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}\right)$$

Probabilistic PCA Solution

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999b)

$$p(\mathbf{Y}|\mathbf{W}) = \prod_{j=1}^{D} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$$

$$\log p(\mathbf{Y}|\mathbf{W}) = -\frac{N}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\right) + \mathrm{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $N^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ and the corresponding eigenvalues are Λ_q,

$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{R}^{\mathrm{T}}, \quad \mathbf{L} = \left(\Lambda_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Linear Pose priors: PCA

Two ways to construct the prior

- Assume a deterministic mapping: use the mean prediction and optimize directly in the latent space

$$\mathbf{y} \approx \mathbf{W}\mathbf{x} = \mathbf{U}_q \mathbf{L} \mathbf{R}^T \mathbf{x}$$

  In the generative tracking, the state is then $\phi = \mathbf{x}$. The latent space is typically called the PCA weights.

- Create a density model in the pose space (Salzmann et al. 10)

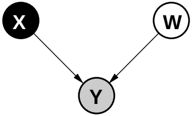
$$-\log p(\mathbf{y}) = ||\mathbf{y}\mathbf{U}_q \mathbf{L}^{-1/2}||_2^2$$

  the state then becomes $\phi = \mathbf{y}$.

Non linear pose models : GPLVM

Dual Probabilistic PCA

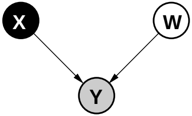
- Define *linear-Gaussian relationship* between latent variables and data.
  - **Novel** Latent variable approach:



$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
  - **Novel** Latent variable approach:
  - Define Gaussian prior over *parameters*, **W**.

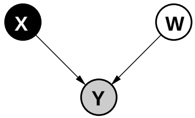


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{N}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Non linear pose models : GPLVM

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
    - **Novel** Latent variable approach:
    - Define Gaussian prior over *parameters*, **W**.
    - Integrate out *parameters*.



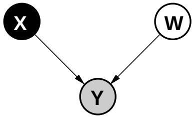
$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p\left(\mathbf{W}\right) = \prod_{i=1}^{D} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

Non linear pose models : GPLVM

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
  - **Novel** Latent variable approach:
  - Define Gaussian prior over *parameters*, **W**.
  - Integrate out *parameters*.



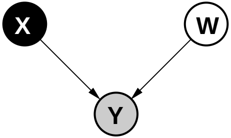
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

$$p(\mathbf{W}) = \prod_{i=1}^{D} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{D} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0},\mathbf{X}\mathbf{X}^{\mathrm{T}}+\sigma^2\mathbf{I}\right)$$

Non linear pose models : GPLVM

Dual Probabilistic PCA Max. Likelihood Soln (Lawrence, 2004)



$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{D} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^{\mathrm{T}} + \sigma^2\mathbf{I}\right)$$

Dual Probabilistic PCA Max. Likelihood Soln (Lawrence, 2004)

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{D} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0},\mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\mathrm{T}} + \sigma^2\mathbf{I}$$

$$\log p(\mathbf{Y}|\mathbf{X}) = -\frac{D}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\right) + \mathrm{const.}$$

If $\mathbf{U}'_q$ are first q principal eigenvectors of $D^{-1}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$ and the corresponding eigenvalues are Λ_q,

$$\mathbf{X} = \mathbf{U'}_q\mathbf{L}\mathbf{R}^{\mathrm{T}}, \quad \mathbf{L} = \left(\Lambda_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999b)

$$p(\mathbf{Y}|\mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$$

$$\log p(\mathbf{Y}|\mathbf{W}) = -\frac{N}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $N^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ and the corresponding eigenvalues are Λ_q,

$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{R}^{\mathrm{T}}, \quad \mathbf{L} = \left(\Lambda_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Equivalence of Formulations

The Eigenvalue Problems are equivalent

- Solution for Probabilistic PCA (solves for the mapping)

$$\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\mathbf{U}_q = \mathbf{U}_q\Lambda_q \qquad \mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{V}^{\mathrm{T}}$$

- Solution for Dual Probabilistic PCA (solves for the latent positions)

$$\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{U}'_q = \mathbf{U}'_q\Lambda_q \qquad \mathbf{X} = \mathbf{U}'_q\mathbf{L}\mathbf{V}^{\mathrm{T}}$$

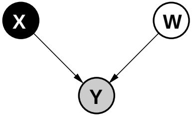
- Equivalence is from

$$\mathbf{U}_q = \mathbf{Y}^{\mathrm{T}}\mathbf{U}'_q\Lambda_q^{-\frac{1}{2}}$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach:
  - Define Gaussian prior over *parameteters*, **W**.
  - Integrate out *parameters*.



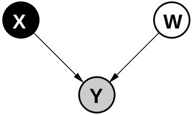
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

$$p(\mathbf{W}) = \prod_{i=1}^{D} N\left(\mathbf{w}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{D} N\left(\mathbf{y}_{:,j}|\mathbf{0},\mathbf{X}\mathbf{X}^{\mathrm{T}}+\sigma^2\mathbf{I}\right)$$

Dual Probabilistic PCA

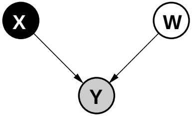
- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.



$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{D} N\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^{\mathrm{T}} + \sigma^2\mathbf{I}\right)$$

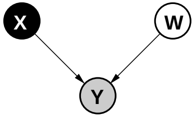
Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.



$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{D} N\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$\mathbf{K} = \mathbf{X}\mathbf{X}^{\mathrm{T}} + \sigma^2\mathbf{I}$$



Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{D} N\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

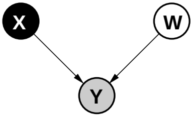
$$\mathbf{K} = \mathbf{X}\mathbf{X}^{\mathrm{T}} + \sigma^2\mathbf{I}$$

This is a product of Gaussian processes with linear kernels.

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.



$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{D} N\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$\mathbf{K} = ?$$

Replace linear kernel with non-linear kernel for non-linear model.

This is called the Gaussian Process Latent Variable Model (GPLVM)

Non linear pose models : GPLVM

Two ways to construct the prior

- Assume a deterministic mapping: use the mean prediction and optimize directly in the latent space

$$\mathbf{y} \approx \mu = \mathbf{Y}^T \mathbf{K}^{-1} \mathbf{k}_*(\mathbf{x})$$

  In the generative tracking, the state is then $\phi = \mathbf{x}$.

- Use the full probabilistic model

$$-\log p(\mathbf{y}|\mathbf{x}) = \frac{||\mathbf{y} - \boldsymbol{\mu}||_2^2}{2\sigma^2(\mathbf{x})} + \frac{d}{2} \log(\sigma^2(\mathbf{x}))$$

  the state then becomes $\phi = [\mathbf{x}, \mathbf{y}]$, with

$$\sigma^2(\mathbf{x}) = k_{*,*} - \mathbf{k}_* \mathbf{K}^{-1} \mathbf{k}_*$$

Non linear motion priors: GPDM

- We now how an additional prior over dynamics

$$-\log p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \frac{||\mathbf{x}_t - \hat{\boldsymbol{\mu}}||_2^2}{2\hat{\sigma}^2(\mathbf{x})} + \frac{d}{2}\log(\hat{\sigma}^2(\mathbf{x}))$$

with

$$\begin{aligned} \hat{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_{t-1}) &= \mathbf{X}_{out}^T \hat{\mathbf{K}}^{-1} \hat{\mathbf{k}}_* \\ \hat{\sigma}(\ \mathbf{x}_{t-1}) &= \hat{k}_{*,*} - \hat{\mathbf{k}}_* \hat{\mathbf{K}}^{-1} \hat{\mathbf{k}}_* \end{aligned}$$

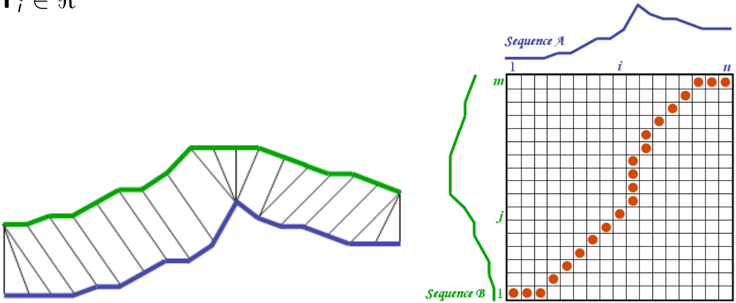
where $\hat{\mathbf{K}}$ is computed from $\mathbf{X}_{in}$.

Linear motion priors: spatio-temporal PCA

- Given a set of training sequences, if we can dynamic time warp them and set a canonical sampling (M samples), we can produce a set of examples

$$\mathbf{Y}_i = [\mathbf{y}_1, \cdots, \mathbf{y}_M]$$

with $\mathbf{Y}_i \in \Re^{D \cdot M}$

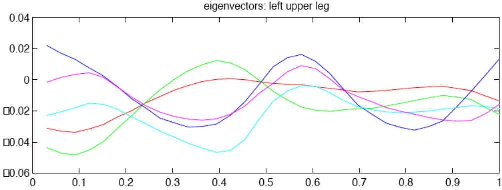


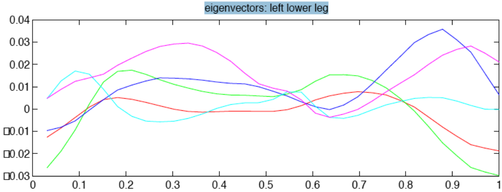
Spatio-temporal eigenvectors

- We can then learn from N spatio-temporal examples a linear PCA model by creating a matrix

$$\mathbf{Y} = [\mathbf{Y}_1, \cdots, \mathbf{Y}_N]$$

with $\mathbf{Y} \in \Re^{N \times D \cdot M}$, where the basis are spatio-temporal.





Spatio-temporal single motion latent space

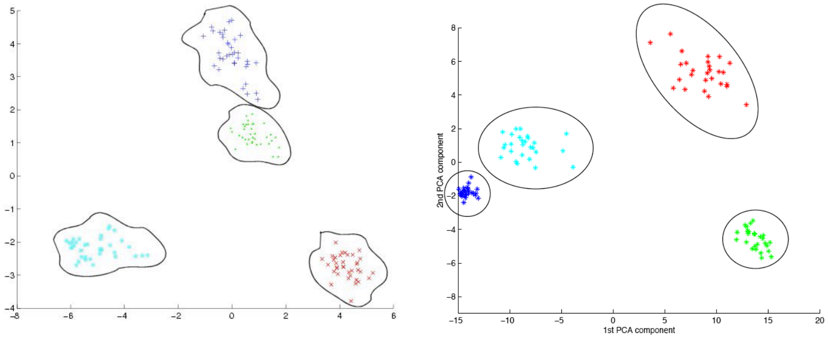


Figure: Spatio-temporal latent space for (left) walking, (right) running (Urtasun et al. 04)

Spatio-temporal multiple motion latent space

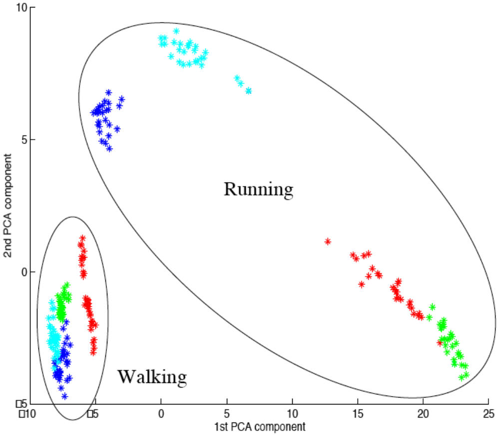


Figure: Spatio-temporal latent space for multiple motions (Urtasun et al. 04)

Motion priors

- Assume a deterministic mapping, and solve for a set of poses at the same time.

$$\mathbf{y}_* \approx \mathbf{U}\mathbf{L}\mathbf{R}^T\mathbf{x}$$

- **Constant style:** assume a single latent variable is enough to model the style

$$\phi = [\mathbf{x}, \mathbf{t}_1, \cdots, \mathbf{t}_P]$$

  where P is the length of the new motion, and t_i represents the phase of the motion at the i-th frame.

- **Varying style:** The style is changing, e.g., there is a transition from walking to running. The state is then augmented by

$$\phi = [\mathbf{x}_1, \cdots, \mathbf{x}_P, \mathbf{t}_1, \cdots, \mathbf{t}_P]$$

More?

- We learn how to create shape and motion priors.
- If you want to learn more, look at the additional material.
- Otherwise, do the research project on this topic!
- Next week we will look into image likelihoods and physics