Computer Graphics

Shape retrieval II

Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

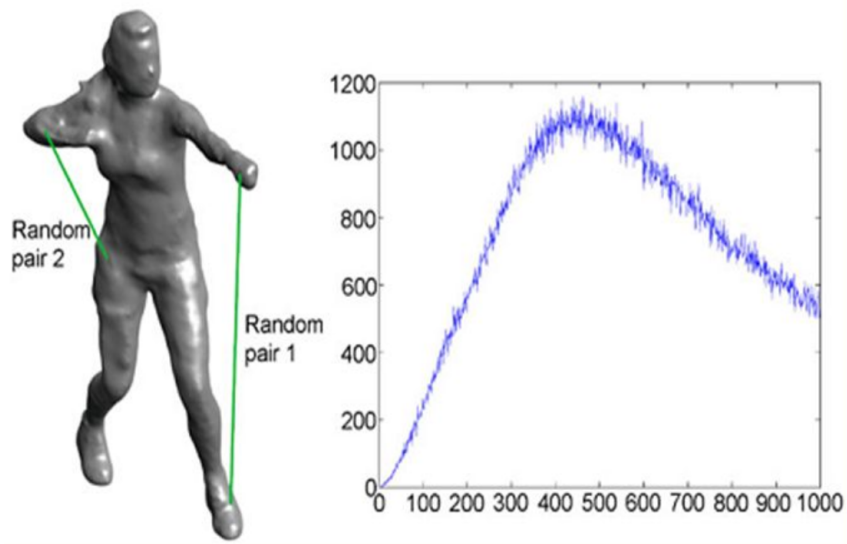
2nd semester a.y. 2018/2019 · May 08, 2019

Motivation: A 3D version of Google

Recap: General approach

Baseline algorithm:

- Compute a global descriptor for each shape (e.g., aggregate local descriptors)
- Search for nearest neighbors in descriptor space



Recap: Laplace-Beltrami operator

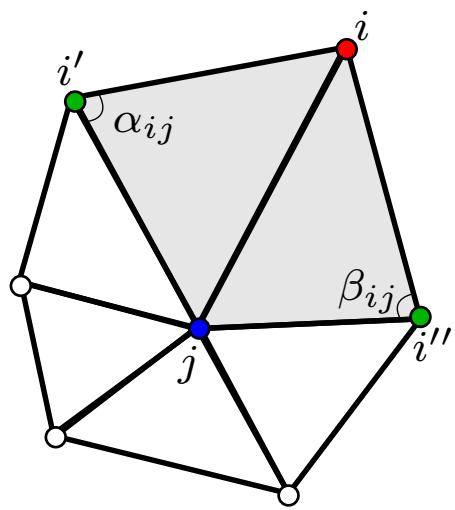
$$\mathbf{L} = \mathbf{M}^{-1}\mathbf{S}$$

$\mathbf{M} \in \mathbb{R}^{n \times n}$ is the mass matrix $\mathbf{M} = \mathrm{diag}(\mathbf{a})$ with area elements

$$\mathbf{a}_i = \frac{1}{3} \sum_{T_j : v_i \in T_j} A(T_j)$$

$\mathbf{S} \in \mathbb{R}^{n \times n}$ is the stiffness matrix with values

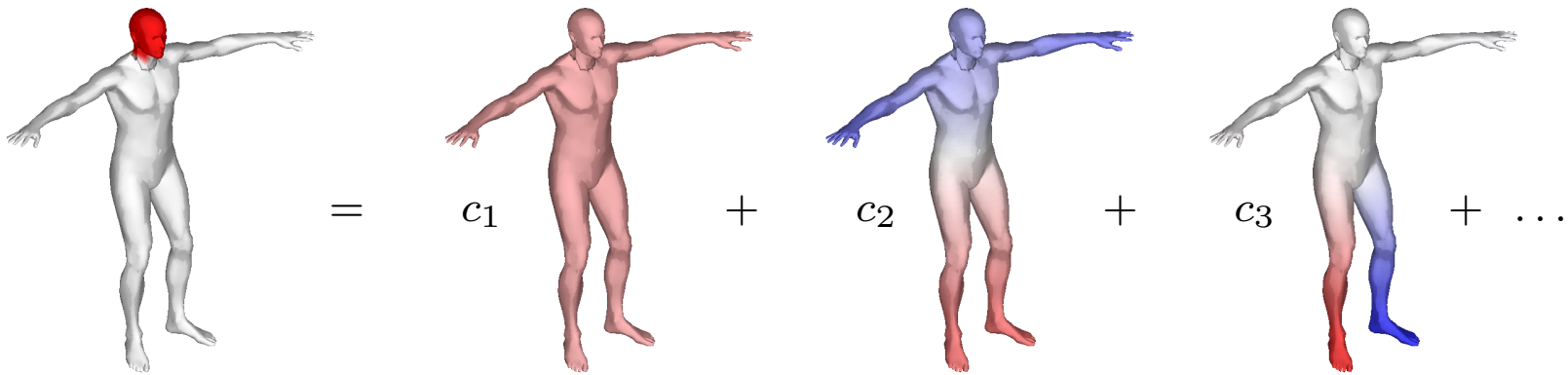
$$\mathbf{S}_{ij} = \begin{cases} -\frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij}) & \text{if } e_{ij} \in E \\ -\sum_{k \neq i} s_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$



Recap: Laplacian eigenfunctions

$$\Delta\phi_i = \lambda_i\phi_i$$

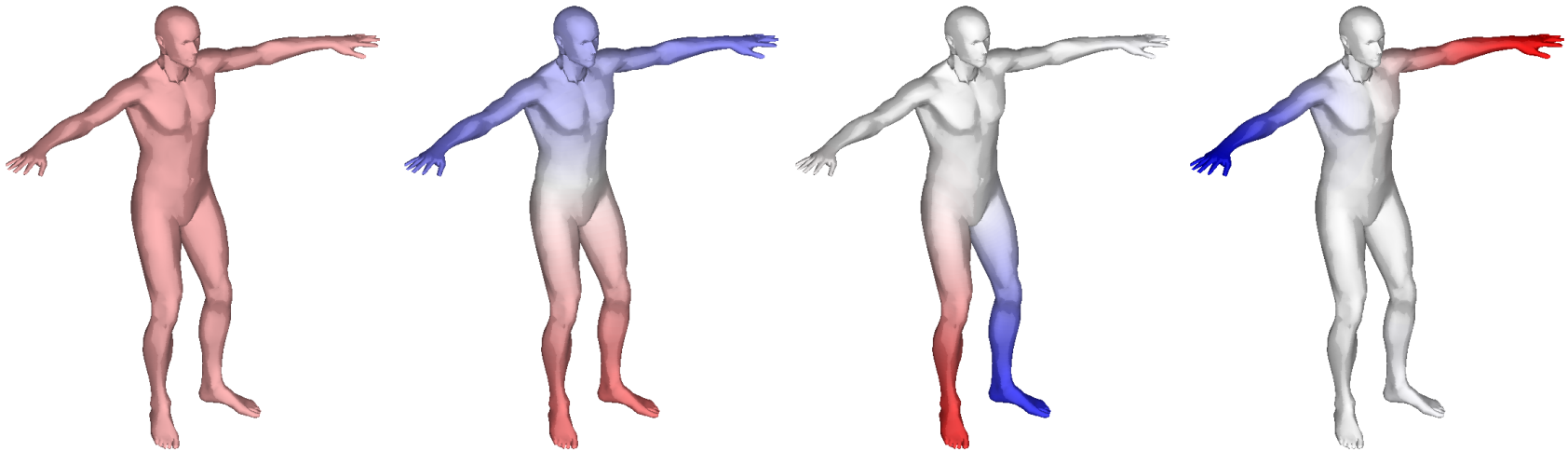
The eigenfunctions $\{\phi_i\}_{i=1,\ldots,k}$ form an orthogonal basis for the vector space of functions $f : \mathcal{X} \to \mathbb{R}$



Recap: Laplacian eigenfunctions

The Laplacian is invariant to isometries

ϕ_1 ϕ_2 ϕ_3 ϕ_4

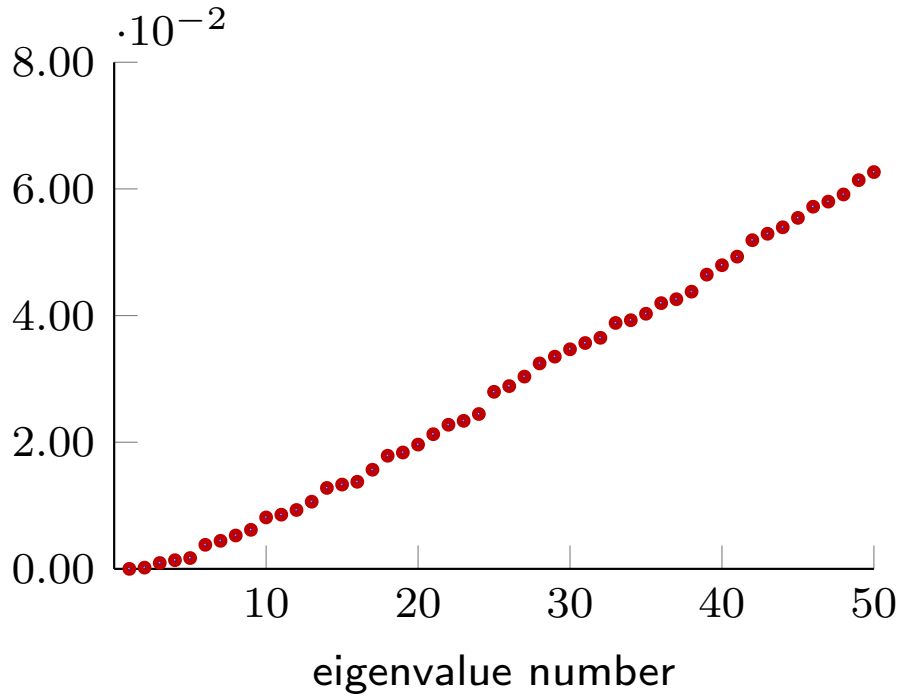


ψ_1 ψ_2 ψ_3 ψ_4

Recap: Laplacian eigenvales

The eigenvalues of the Laplacian have a canonical ordering and the first eigenvalue is always 0:

$$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \cdots \to \infty$$

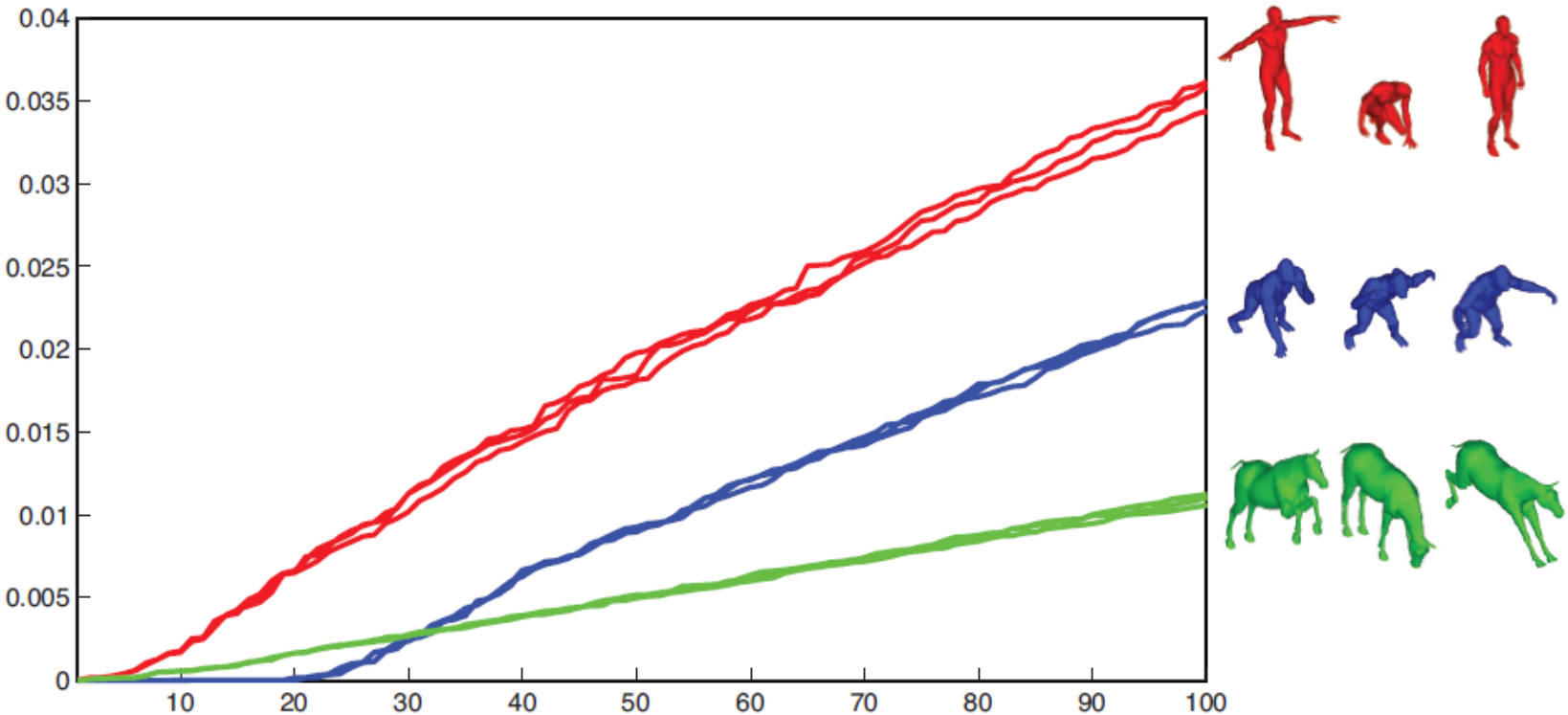


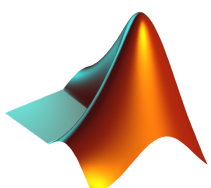
Weyl's law:

$$\lambda_i \approx \frac{\pi}{\int_{\mathcal{X}} da} i \qquad \text{for } i \to \infty$$

Recap: Shape DNA

We can use the Laplacian spectrum as a global shape descriptor





Recap: Global Point Signature (GPS)

The Laplacian eigenfunctions can be directly used to define local point descriptors:

$$\mathrm{GPS}(x_i) = \left(\frac{\phi_1(x_i)}{\sqrt{\lambda_1}}, \frac{\phi_2(x_i)}{\sqrt{\lambda_2}}, \ldots, \frac{\phi_k(x_i)}{\sqrt{\lambda_k}}\right) \in \mathbb{R}^k$$

A single descriptor for the entire shape can be obtained by weighted averaging:

$$\mathrm{GPS}(\mathcal{X}) = \int_{\mathcal{X}} \mathrm{GPS}(x)dx \Longrightarrow \mathbf{H} = (\mathbf{1}^\top \mathbf{M}\mathbf{G})^\top$$

Heat diffusion

For a surface $\mathcal{X} \in \mathbb{R}^3$ the diffusion of heat is described by the heat equation:

$$\frac{\partial}{\partial t} u(x,t) = -\Delta u(x,t)$$
$$u(x,0) = u_0(x)$$

Heat diffusion

For a surface $\mathcal{X} \in \mathbb{R}^3$ the diffusion of heat is described by the heat equation:

$$\frac{\partial}{\partial t} u(x,t) = -\Delta u(x,t)$$
$$u(x,0) = u_0(x)$$

For flat shapes, the Laplacian is defined as:

$$\Delta u(x,t) = \frac{\partial^2 u(x,t)}{\partial x_1^2} + \frac{\partial^2 u(x,t)}{\partial x_2^2}$$

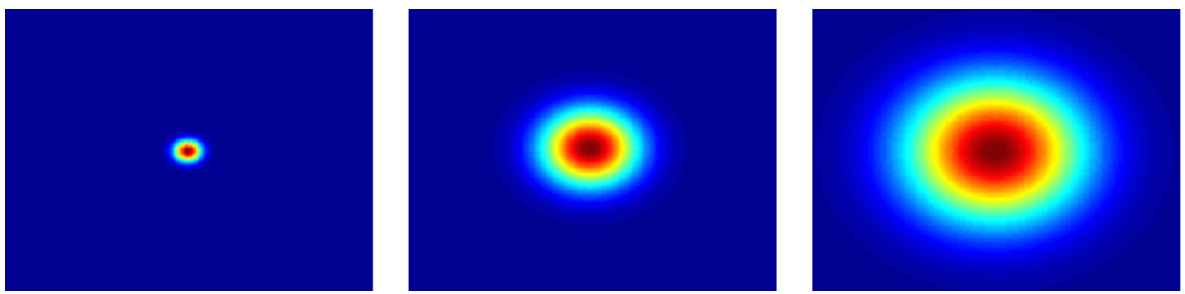
Heat diffusion

For a surface $\mathcal{X} \in \mathbb{R}^3$ the diffusion of heat is described by the heat equation:

$$\frac{\partial}{\partial t} u(x,t) = -\Delta u(x,t)$$
$$u(x,0) = u_0(x)$$

For flat shapes, the Laplacian is defined as:

$$\Delta u(x,t) = \frac{\partial^2 u(x,t)}{\partial x_1^2} + \frac{\partial^2 u(x,t)}{\partial x_2^2}$$



$t = 0$ $t = 1.3$ $t = 3.7$

Heat kernel

$$\frac{\partial}{\partial t}u(x,t) = -\Delta u(x,t)$$
$$u(x,0) = u_0(x)$$

A solution is obtained by

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y)u_0(y)dy$$

Heat kernel

$$\frac{\partial}{\partial t}u(x,t) = -\Delta u(x,t)$$
$$u(x,0) = u_0(x)$$

A solution is obtained by

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y)u_0(y)dy$$

The function $k_t : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called the heat kernel of $\mathcal{X}$

Heat kernel

$$\frac{\partial}{\partial t}u(x,t) = -\Delta u(x,t)$$
$$u(x,0) = u_0(x)$$

A solution is obtained by

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y)u_0(y)dy$$

The function $k_t : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called the heat kernel of $\mathcal{X}$

- $k_t(x,y)$ describes the amount of heat transferred from point x to point y in time t

Heat kernel

$$\frac{\partial}{\partial t}u(x,t) = -\Delta u(x,t)$$
$$u(x,0) = u_0(x)$$

A solution is obtained by

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y)u_0(y)dy$$

The function $k_t : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called the heat kernel of $\mathcal{X}$

- $k_t(x,y)$ describes the amount of heat transferred from point x to point y in time t
- It is a property of the manifold $\mathcal{X}$ and does not depend on the initial distribution $u_0(x)$

Dirac initialization

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y) u_0(y) dy$$

Assume the initial distribution is a Dirac delta at point $z \in \mathcal{X}$

Dirac initialization

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y) u_0(y) dy$$

Assume the initial distribution is a Dirac delta at point $z \in \mathcal{X}$

By definition, the Dirac distribution δ_z satisfies the sampling property

$$\int_{\mathcal{X}} f(x) \delta_z(x) dx = f(z)$$

Dirac initialization

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y) u_0(y) dy$$

Assume the initial distribution is a Dirac delta at point $z \in \mathcal{X}$

By definition, the Dirac distribution δ_z satisfies the sampling property

$$\int_{\mathcal{X}} f(x) \delta_z(x) dx = f(z)$$

With this initialization, the solution to heat equation is simply

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y) \delta_z(y) dy =$$

Dirac initialization

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y) u_0(y) dy$$

Assume the initial distribution is a Dirac delta at point $z \in \mathcal{X}$

By definition, the Dirac distribution δ_z satisfies the sampling property

$$\int_{\mathcal{X}} f(x) \delta_z(x) dx = f(z)$$

With this initialization, the solution to heat equation is simply

$$u(x,t) = \int_{\mathcal{X}} k_t(x,y) \delta_z(y) dy = k_t(x,z)$$

Note that z is fixed, so here $k_t(x,z)$ is a function of x

Heat kernel: Spectral decomposition

If $u_0(x) = \delta_y(x)$, it can be proven that:

$$u(x,t) \approx \sum_{i=0}^{k} e^{-\lambda_i t} \phi_i(x) \phi_i(y)$$

Heat kernel: Spectral decomposition

If $u_0(x) = \delta_y(x)$, it can be proven that:

$$k_t(x, y) \approx \sum_{i=0}^{k} e^{-\lambda_i t} \phi_i(x) \phi_i(y)$$

Recall that this also defines the heat kernel between x and y

Heat kernel: Spectral decomposition

If $u_0(x) = \delta_y(x)$, it can be proven that:

$$k_t(x, y) \approx \sum_{i=0}^{k} e^{-\lambda_i t} \phi_i(x) \phi_i(y)$$

Recall that this also defines the heat kernel between x and y

In matrix notation, the heat kernel can be written as a $n \times n$ matrix

$$\mathbf{K}_t = \mathbf{\Phi} \text{diag}(e^{-\lambda_i t}) \mathbf{\Phi}^\top$$

Heat kernel: Spectral decomposition

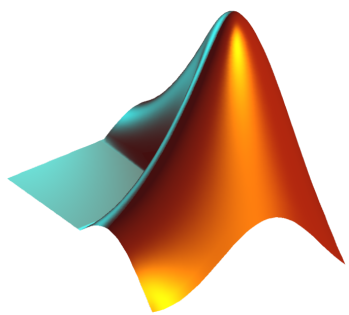
If $u_0(x) = \delta_y(x)$, it can be proven that:

$$k_t(x, y) \approx \sum_{i=0}^{k} e^{-\lambda_i t} \phi_i(x) \phi_i(y)$$

Recall that this also defines the heat kernel between x and y

In matrix notation, the heat kernel can be written as a $n \times n$ matrix

$$\mathbf{K}_t = \mathbf{\Phi} \mathrm{diag}(e^{-\lambda_i t}) \mathbf{\Phi}^\top$$



Heat Kernel Signature (HKS)

If $T : \mathcal{X} \to \mathcal{Y}$ is an isometry, then

$$k_t^{\mathcal{X}}(x, y) = k_t^{\mathcal{Y}}(T(x), T(y))$$

and vice-versa

Heat Kernel Signature (HKS)

If $T : \mathcal{X} \to \mathcal{Y}$ is an isometry, then

$$k_t^{\mathcal{X}}(x, y) = k_t^{\mathcal{Y}}(T(x), T(y))$$

and vice-versa

We use this property to define a local descriptor based on the heat kernel;

Consider the diagonal of the heat kernel:

$$k_t(x, x) = \sum_{i=0}^{k} e^{-\lambda_i t} \phi_i(x)^2$$

Sun, Ovsjanikov and Guibas 2009

Heat Kernel Signature (HKS)

If $T : \mathcal{X} \to \mathcal{Y}$ is an isometry, then

$$k_t^{\mathcal{X}}(x, y) = k_t^{\mathcal{Y}}(T(x), T(y))$$

and vice-versa

We use this property to define a local descriptor based on the heat kernel;

Consider the diagonal of the heat kernel:

$$k_t(x, x) = \sum_{i=0}^{k} e^{-\lambda_i t} \phi_i(x)^2$$

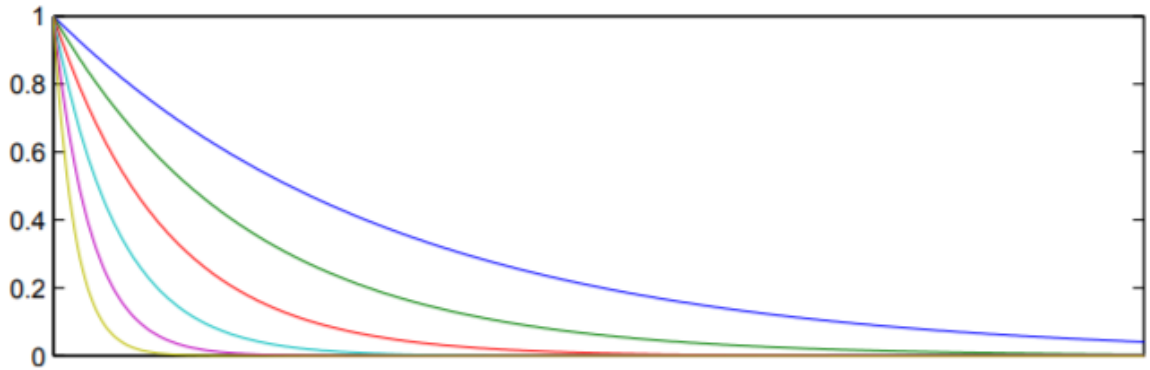
The heat kernel signature is then defined as:

$$\mathrm{hks}(x) = (k_{t_1}(x, x), \ldots, k_{t_T}(x, x)) \in \mathbb{R}^T$$

Sun, Ovsjanikov and Guibas 2009

Heat Kernel Signature (HKS)

$$\mathrm{hks}(x) = (k_{t_1}(x,x), \ldots, k_{t_T}(x,x)) \in \mathbb{R}^T$$



If $\mathcal{X}$ and $\mathcal{Y}$ are isometric, corresponding points $(x, y) \in \mathcal{X} \times \mathcal{Y}$ are expected to have similar signatures

Sun, Ovsjanikov and Guibas 2009

Visualizing descriptors: GPS

We can visualize each descriptor dimension individually

GPS dimension: 1

Visualizing descriptors: GPS

We can visualize each descriptor dimension individually

GPS dimension: 6

Visualizing descriptors: GPS

We can visualize each descriptor dimension individually

GPS dimension: 19

Visualizing descriptors: HKS

We can visualize each descriptor dimension individually

HKS dimension: 1

Visualizing descriptors: HKS

We can visualize each descriptor dimension individually

HKS dimension: 30

Visualizing descriptors: HKS

We can visualize each descriptor dimension individually

HKS dimension: 80

Visualizing descriptor repeatability

We can also visualize distance in descriptor space

GPS

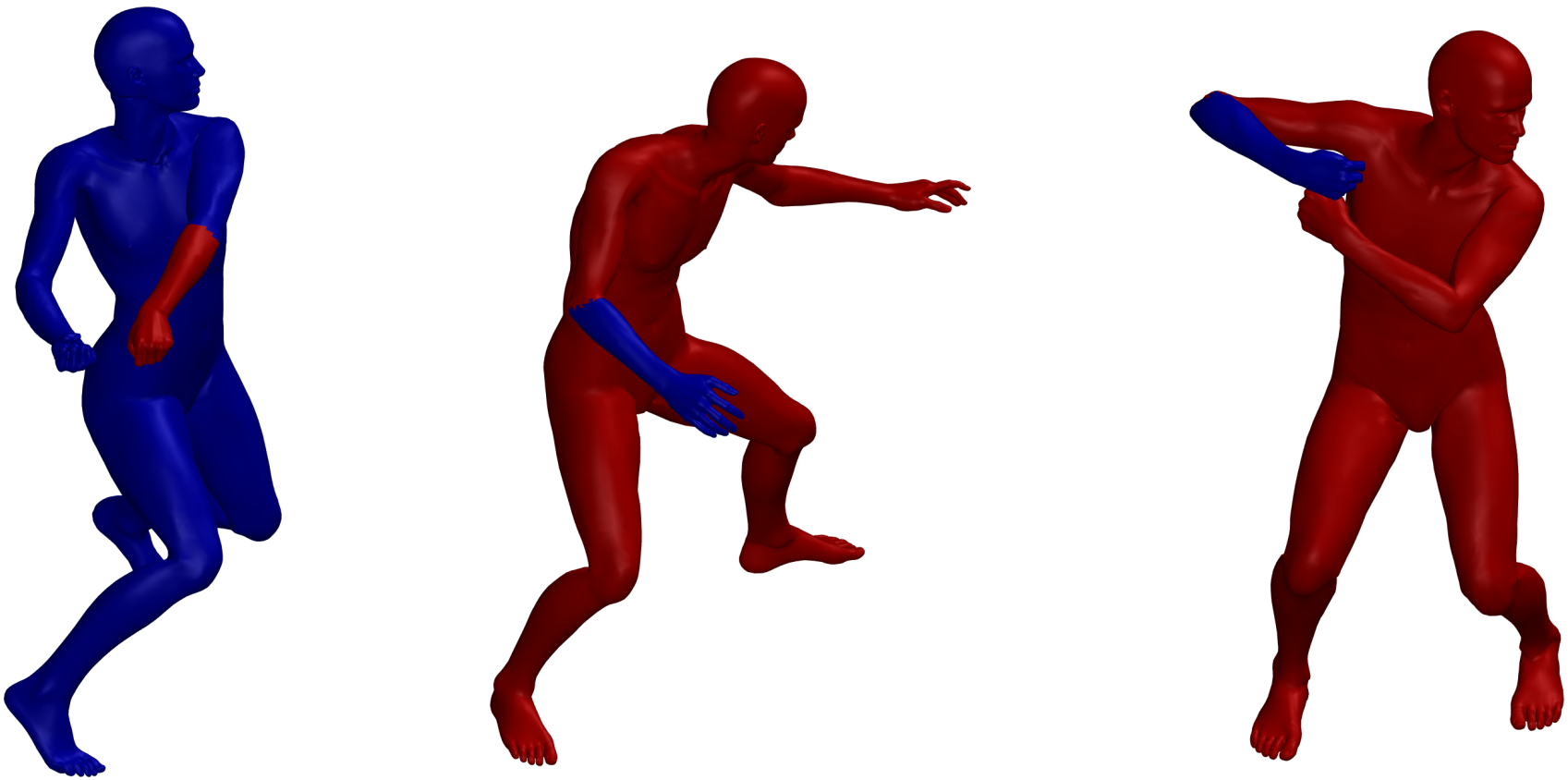
Visualizing descriptor repeatability

We can also visualize distance in descriptor space

HKS

Shape segmentation

By running k-means in descriptor space, one gets a segmentation into regions of the surface



GPS, $k = 2$

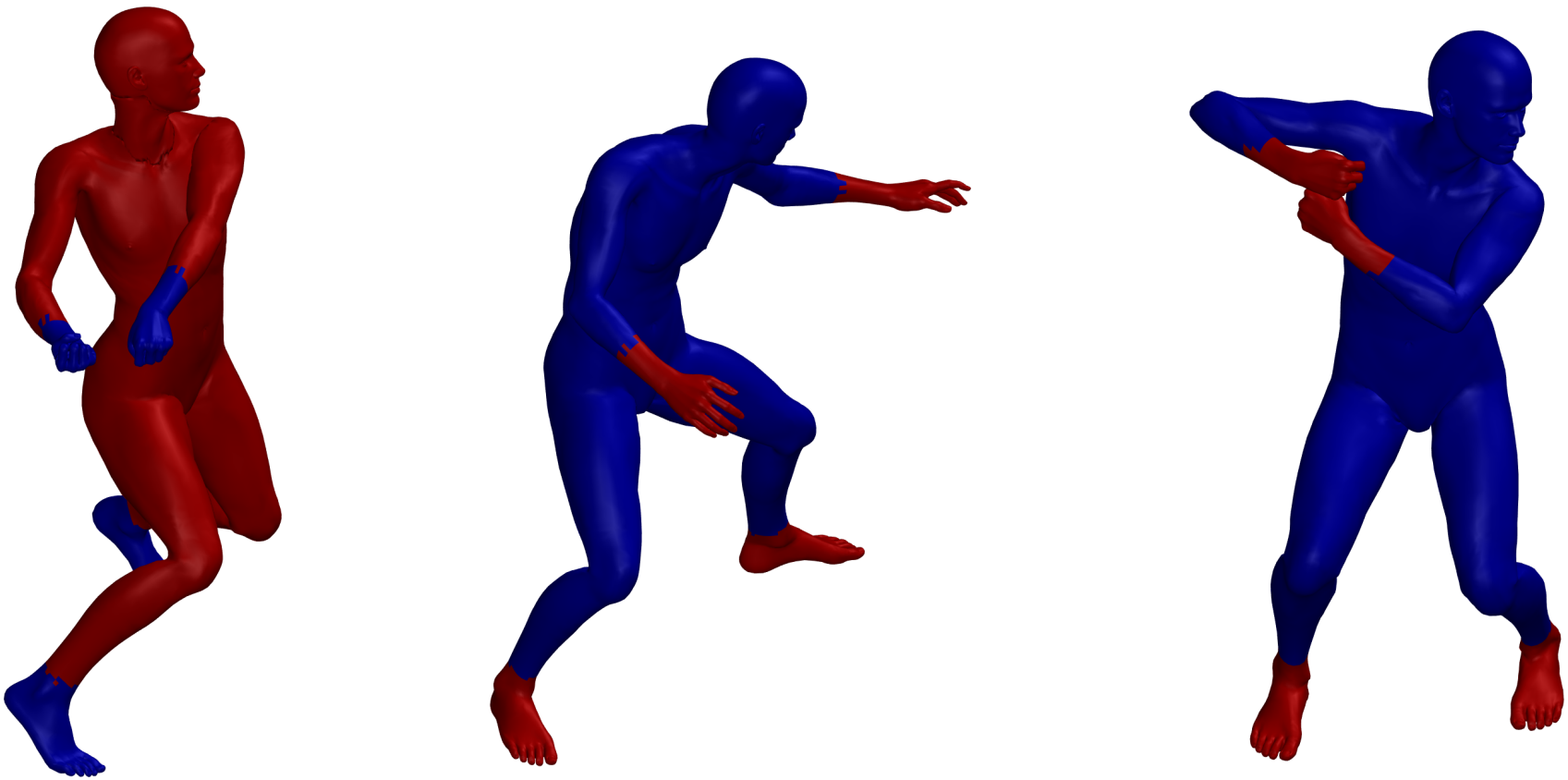
Shape segmentation

By running k-means in descriptor space, one gets a segmentation into regions of the surface

GPS, $k = 10$

Shape segmentation

By running k-means in descriptor space, one gets a segmentation into regions of the surface



HKS, $k = 2$

Shape segmentation

By running k-means in descriptor space, one gets a segmentation into regions of the surface

HKS, $k = 10$

Partial shape retrieval

Using shape segmentation we can search for individual regions

Partial shape retrieval

Using shape segmentation we can search for individual regions

- Segment the shapes into regions

Partial shape retrieval

Using shape segmentation we can search for individual regions

- Segment the shapes into regions
- Compute a global descriptor for each region

Partial shape retrieval

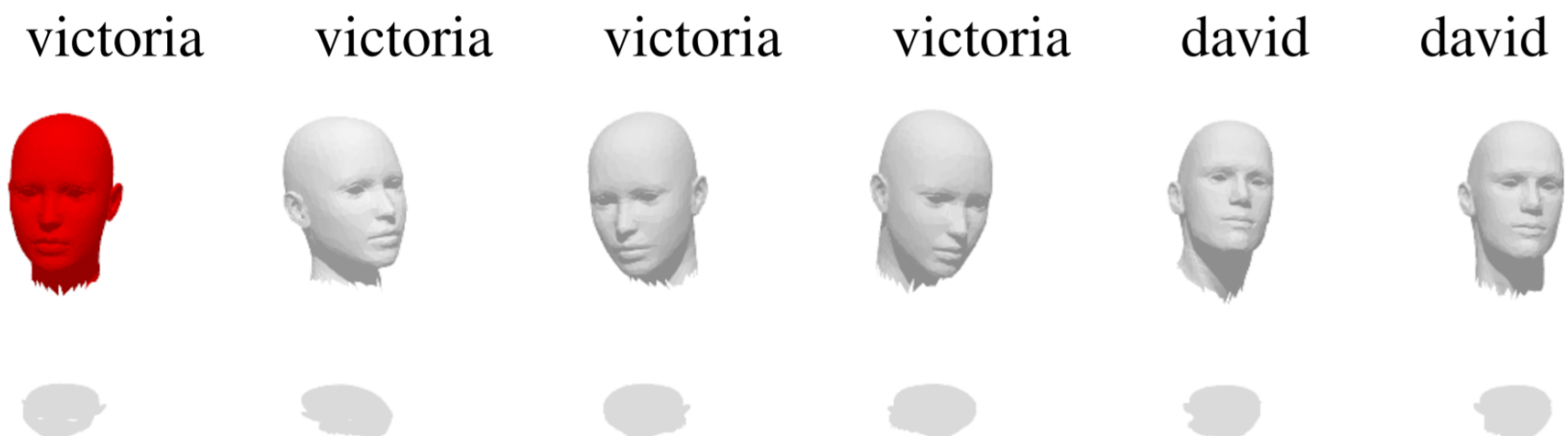
Using shape segmentation we can search for individual regions

- Segment the shapes into regions
- Compute a global descriptor for each region
- Compute a ranking for the most similar regions

Partial shape retrieval

Using shape segmentation we can search for individual regions

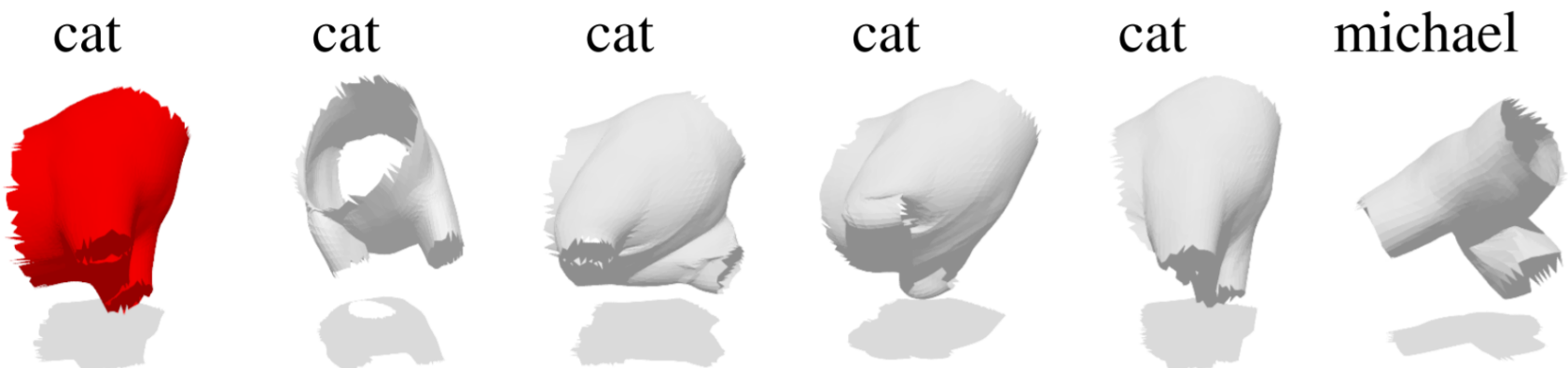
- Segment the shapes into regions
- Compute a global descriptor for each region
- Compute a ranking for the most similar regions



Partial shape retrieval

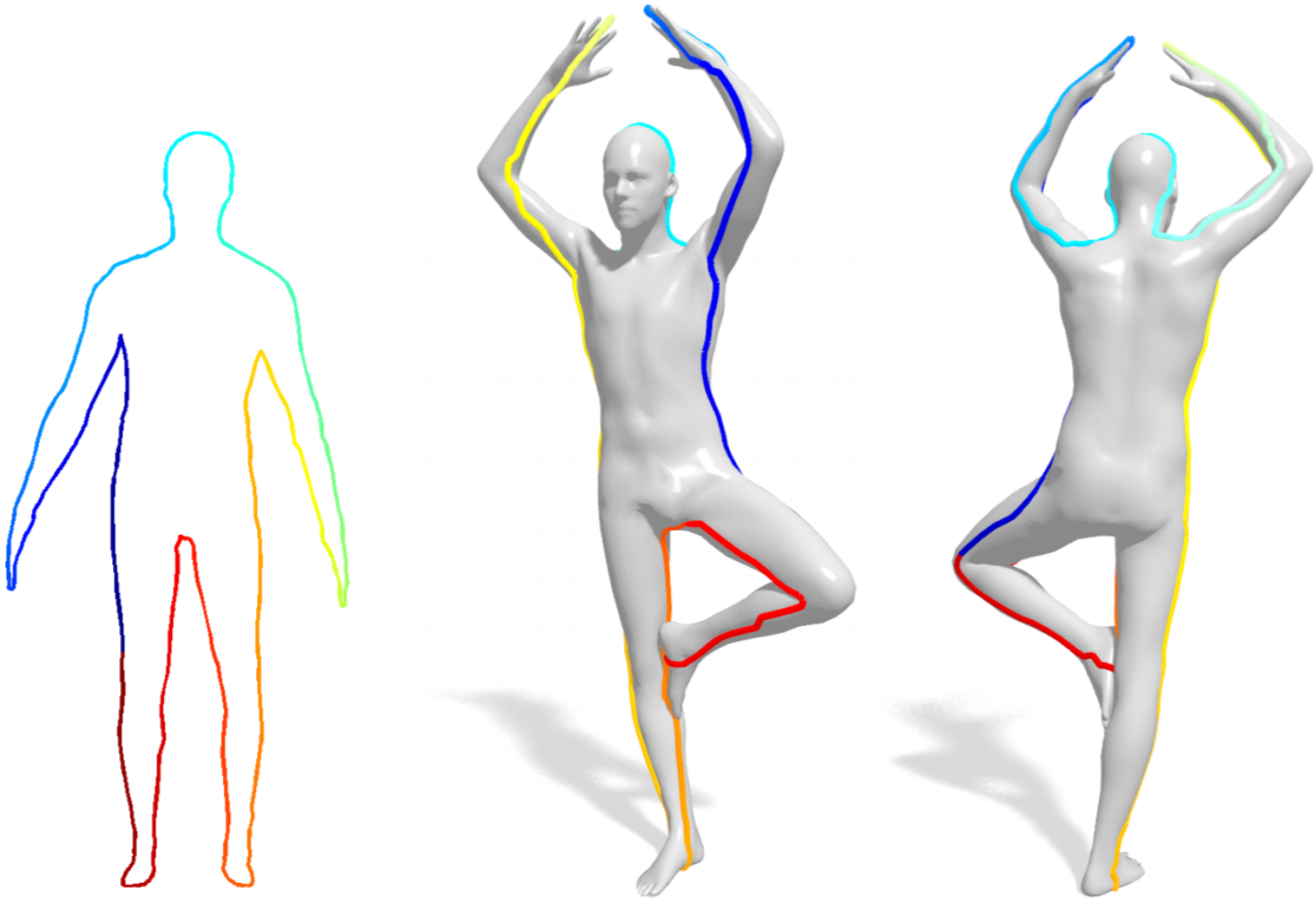
Using shape segmentation we can search for individual regions

- Segment the shapes into regions
- Compute a global descriptor for each region
- Compute a ranking for the most similar regions



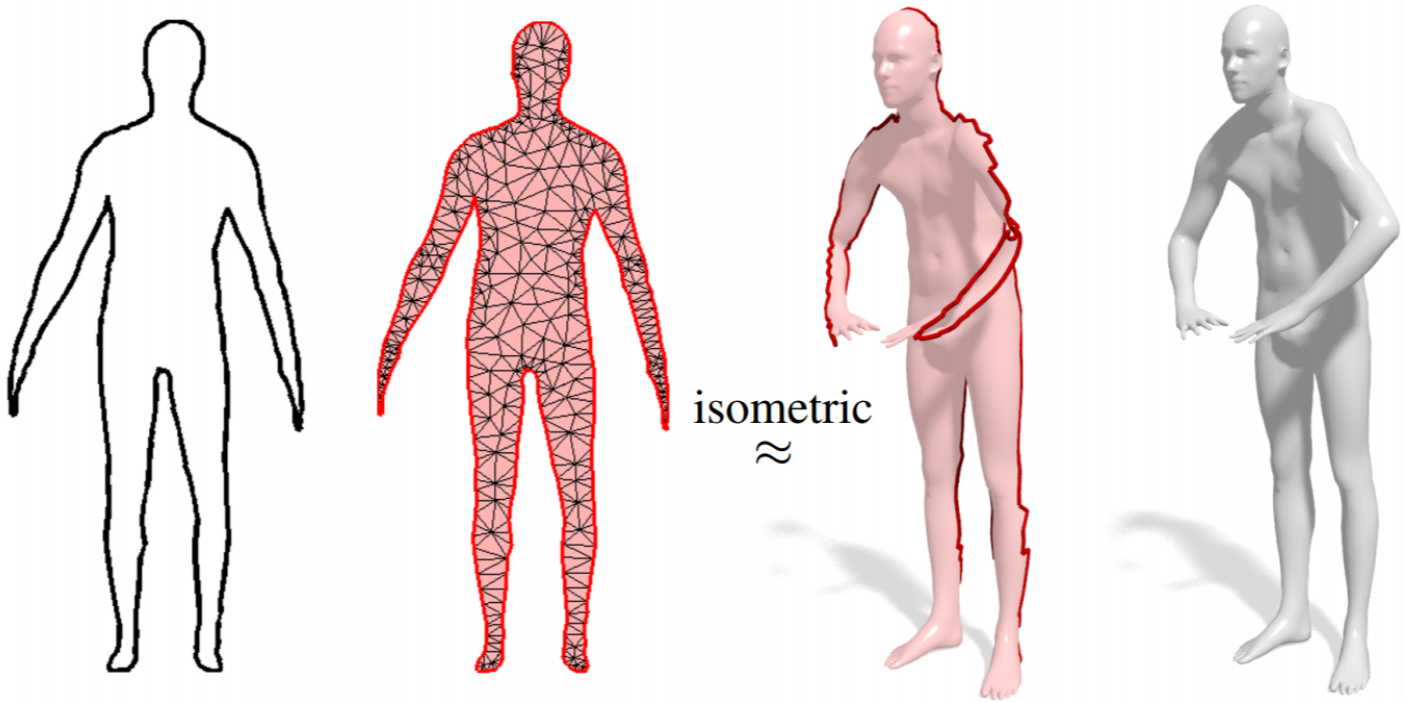
Shape-from-sketch retrieval

Simplest setting: the sketch is a silhouette



Shape-from-sketch retrieval

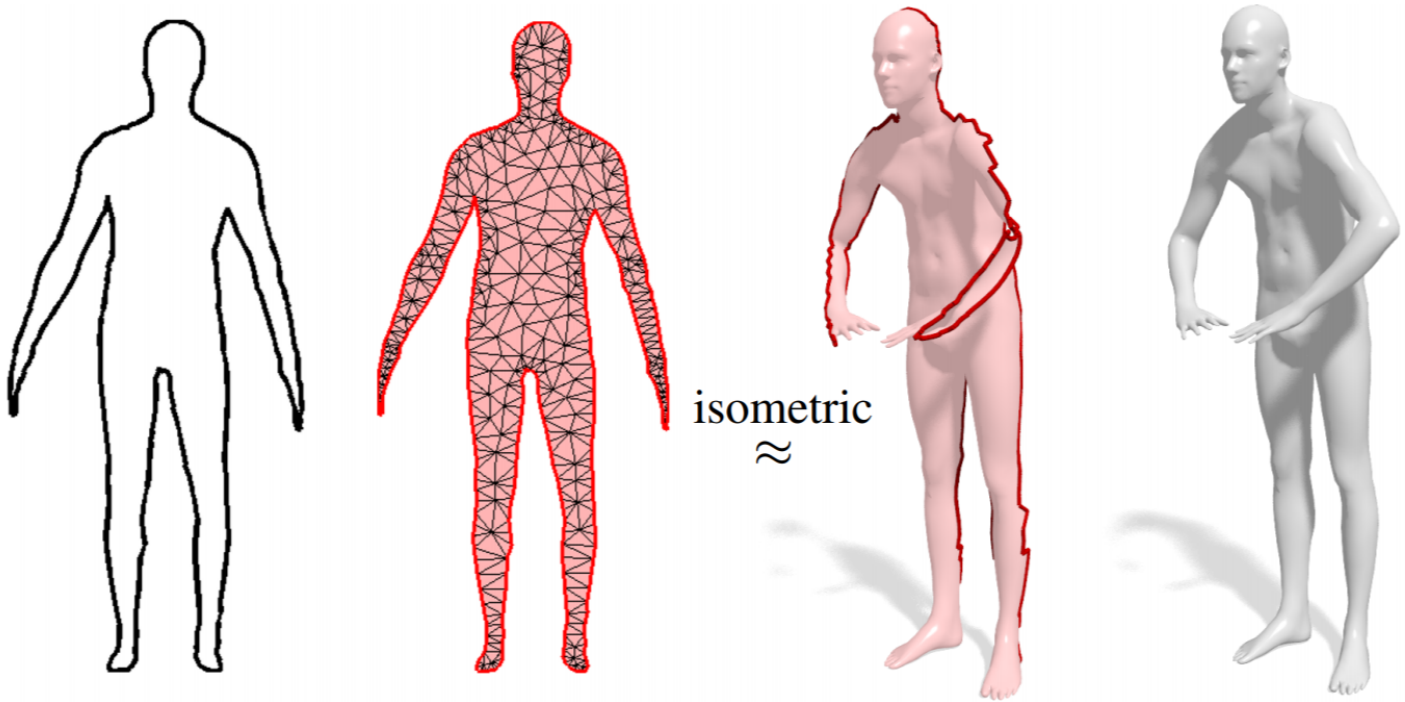
Idea: treat the silhouette as a flat shape



Lähner, Rodolà, Schmidt, Bronstein, Cremers. "Efficient globally optimal 2d-to-3d deformable shape matching". CVPR 2016

Shape-from-sketch retrieval

Idea: treat the silhouette as a flat shape

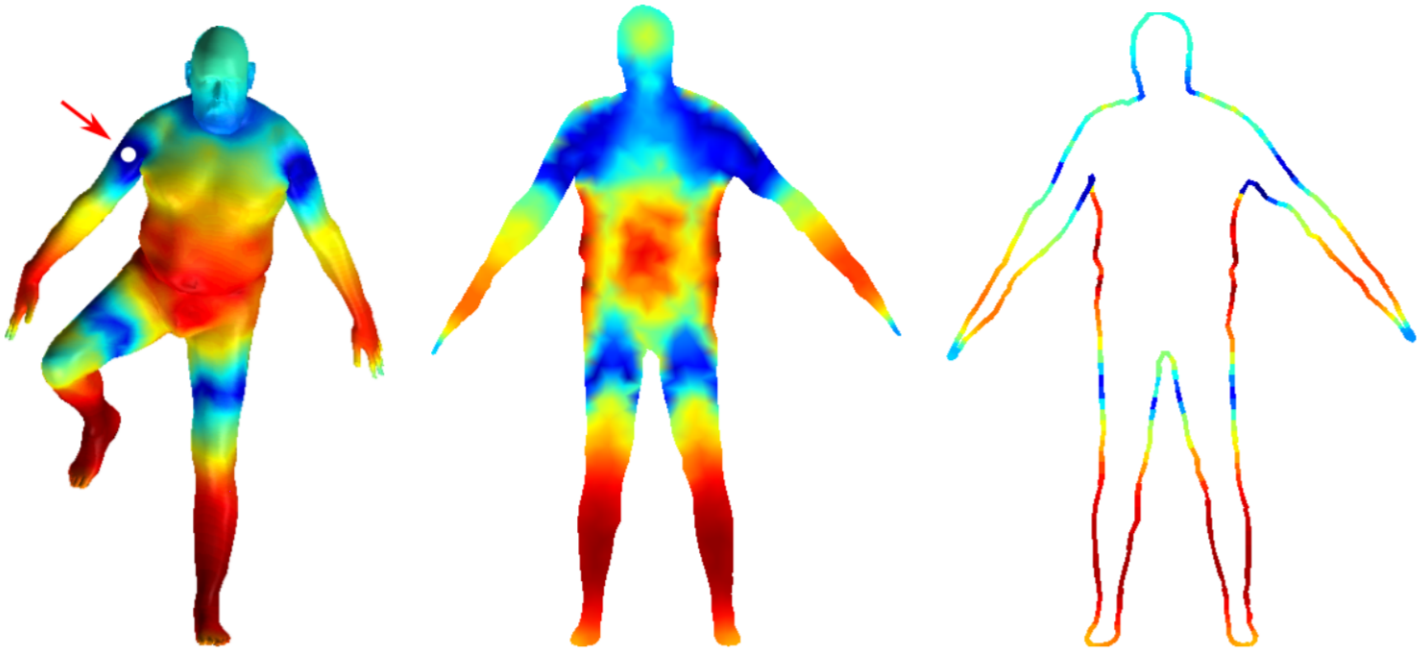


We can compute descriptors in the flat case as we did on surfaces

Lähner, Rodolà, Schmidt, Bronstein, Cremers. "Efficient globally optimal 2d-to-3d deformable shape matching". CVPR 2016

Shape-from-sketch retrieval

Distance in HKS descriptor space



Recall that these descriptors are deformation invariant

Lähner, Rodolà, Schmidt, Bronstein, Cremers. "Efficient globally optimal 2d-to-3d deformable shape matching". CVPR 2016