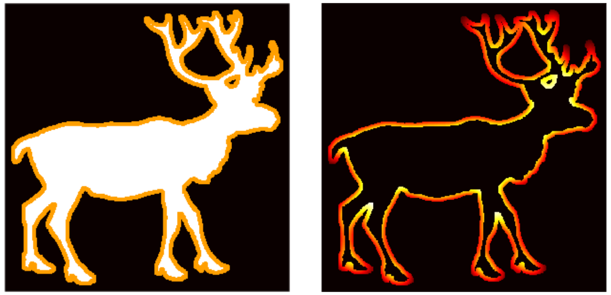
Analysis of 3D Shapes (IN2238)

Frank R. Schmidt
Matthias Vestner

Summer Semester 2016

2D Shape



If a 2D shape S stems from a 2D object O with a connected 1D-boundary, we usually represent it by two different functions

$$c\colon \mathbb{S}^1 \to \mathbb{R}^2 \qquad\qquad f\colon \mathbb{S}^1 \to \mathbb{R}^k,$$

where

- c describes a **specific parametrization** of the boundary of a **specific object** O corresponding to the shape S.
- f describes the **shape features**, which help to describe the shape S and not the object O.

Matching of Two 2D Shapes

Given two shapes S_1 and S_2 with their representations $c_1, c_2 \colon \mathbb{S}^1 \to \mathbb{R}^2$ as well as $f_1, f_2 \colon \mathbb{S}^1 \to \mathbb{R}^k$, we define the matching energy for an arbitrary matching function $m \colon \mathbb{S}^1 \to \mathbb{S}^1$ as

$$E_{S_1,S_2}(m) = \int_{\mathbb{S}^1} \mathrm{dist}_{\mathcal{F}}(f_1(t), f_2 \circ m(t)) \cdot \sqrt{\dot{c}_1(t)^2 + \frac{\mathrm{d}}{\mathrm{dt}}(c_2 \circ m)(t)^2} \mathrm{dt}.$$

This leads to a semi-metric $\mathrm{dist}(\cdot,\cdot)$ of shapes

$$\mathrm{dist}(S_1, S_2) = \min_{m \in \mathrm{Diff}(\mathbb{S}^1)} E_{S_1,S_2}(m),$$

where $\mathrm{Diff}(\mathbb{S}^1)$ describes the set of diffeomorphic mappings $m \colon \mathbb{S}^1 \to \mathbb{S}^1$.

Classification Task

A common task in computer vision is known as **classification**. Here, we assume a set $\mathcal{O}$ of possible **observations**, a smaller class $\mathcal{C}$ of different **classes** and a **classifier**

$$\Phi: \mathcal{O} \to \mathcal{C}$$

that assign to each observation its unique class.

In general, neither $\mathcal{O}$ nor Φ are known. The goal of classification is to estimate Φ by providing a certain amount of classified observations.

$(S_i, \ell_i)_{i \in \mathcal{I}}$ contains observations $S_i \subset \mathbb{R}^N$ and class labels $\ell_i \in \mathcal{C}$. Estimating Φ can be formulated as finding a function $\hat{\Phi}: \mathbb{R}^N \to \mathcal{C}$ such that $\hat{\Phi}(S_i) = \ell_i$. Estimating Φ based on certain observations is called **training**.

Most of the known classification methods use the canonical metric on $\mathbb{R}^N$ in order to **train the classifier**.

Classification of Shapes

If we want to classify shapes, we have more information about the observation space, *i.e.*, **shape space**. Instead of using the metric of some embedding space, we can directly use the **semi-metric** of the shape space.

In fact, it is common to use a simplified classifier framework in order to evaluate shape distances. Given a shape dataset $\mathcal{S}$, one computes in a first step for each pair $(S_i, S_j) \in \mathcal{S}^2$ their shape distance.

In a second step, we **retrieve** for each shape S_i its k nearest neighbors $\mathcal{S}_i^k$, *i.e.*, the k shapes of a given dataset that have the smallest distance with respect to S_i.

Evaluating how well these k nearest neighbors coincide with the perceptual shape class gives us a measure on how well the shape distance measures the practical shape similarity that we like to model.

MPEG-7 CE-Shape-1 Part-B

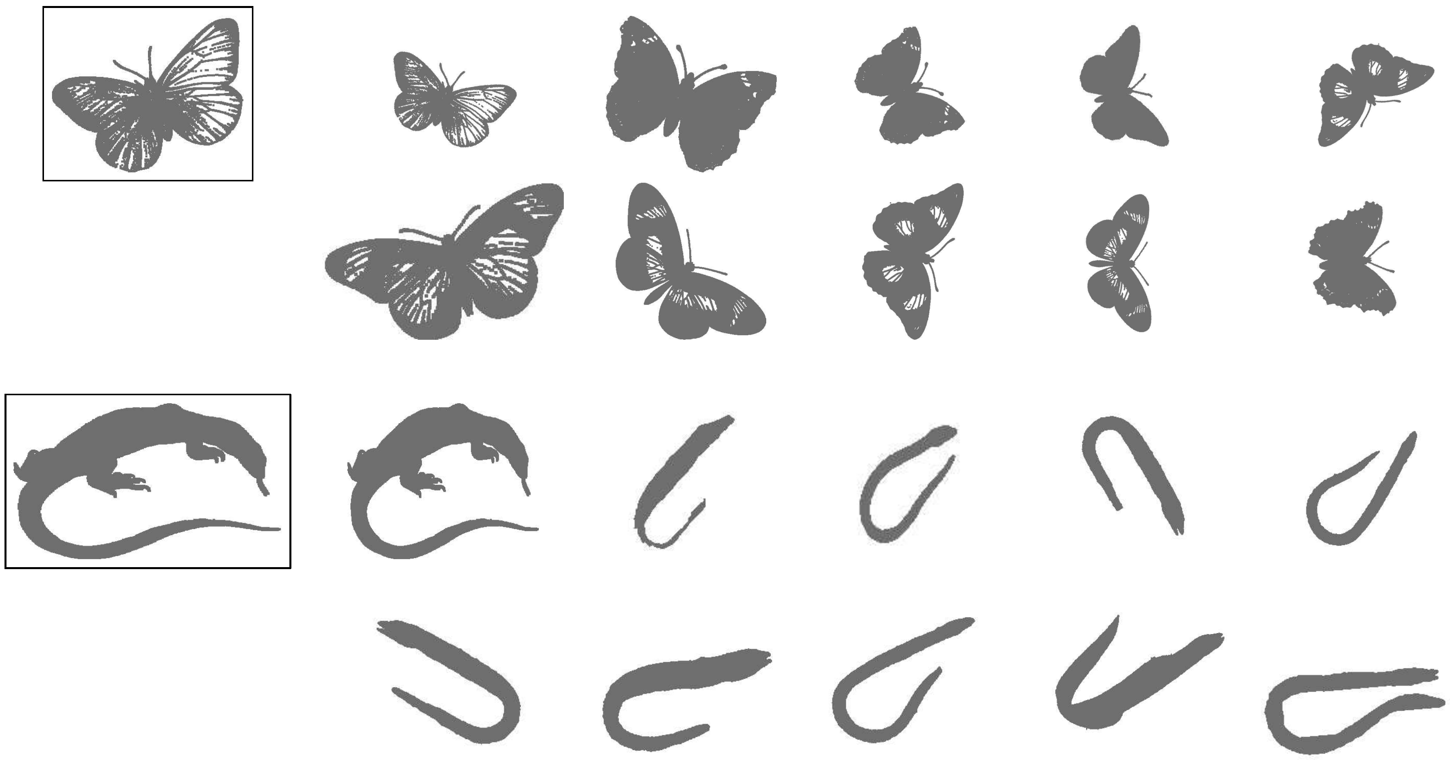
MPEG-7 is a standard to describe multimedia content. In contrast to MPEG-1, MPEG-2 and MPEG-4 it does not introduce a new encoding scheme. Instead it provides meta-information.

One of the **core experiments** was with respect to shape. The **CE-Shape-1** contains 4 different databases. From particular interest is **Part-B**.

This database contains 70 shape classes with 20 objects contained in each class. This provides us with a very big database of 1400 shapes. Computing all pairwise distances results in about **2 million** shape matching tasks.

The **bull's eye test** with respect to the MPEG7 shape database asks for the computation of the 40 nearest neighbors. The **recall** of a specific shape is the ratio of correctly retrieved shapes and 20. The **bull's eye score** is the average recall for all 1400 objects. At this point, every method achieved a score below 90%.

Distance-Driven Retrieval



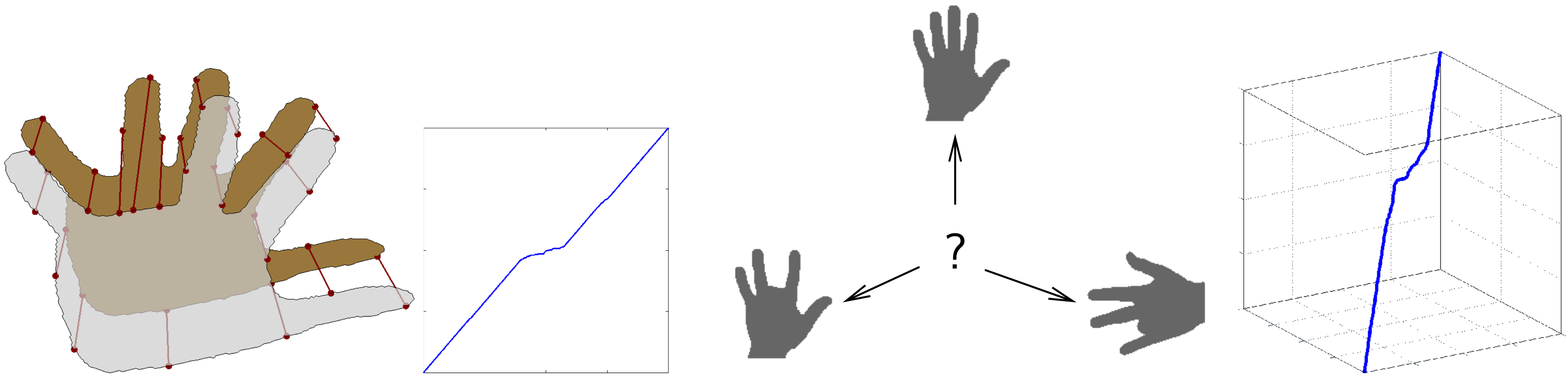
Lack of Knowledge

If we compare two objects of the same shape class, we might miss some vital information of the whole shape class.

For that reason, we would expect a better shape matching if we consider multiple shapes at the same time in order to obtain more information.

This leads us to the problem of **simultaneous shape matching**.

Outline

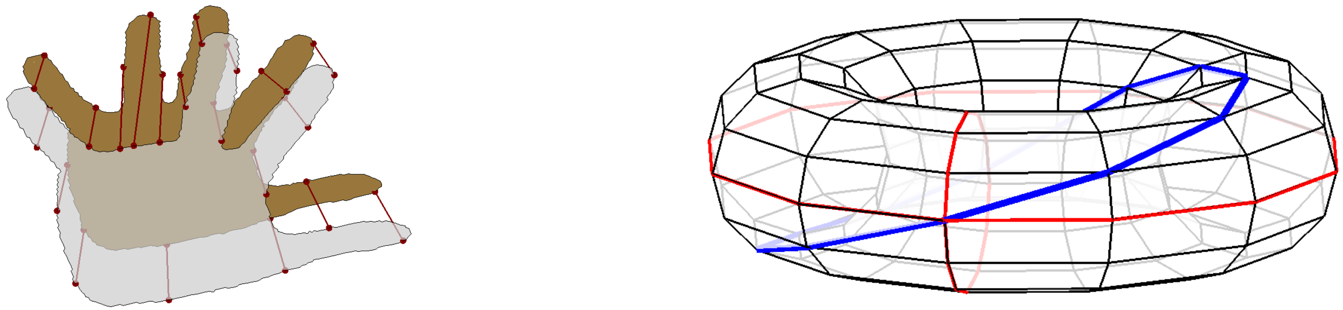


Given two shapes (discretized by N points each) and an initial match, we can compute the optimal matching in $\mathcal{O}(N^2)$.

We want to extend this idea to multiple shapes. This can be cast as finding a shortest path in a higher-dimensional graph.

After finding such a matching path, we like to use it in order to define a **mean shape** of multiple shapes.

Simultaneous Shape Matching

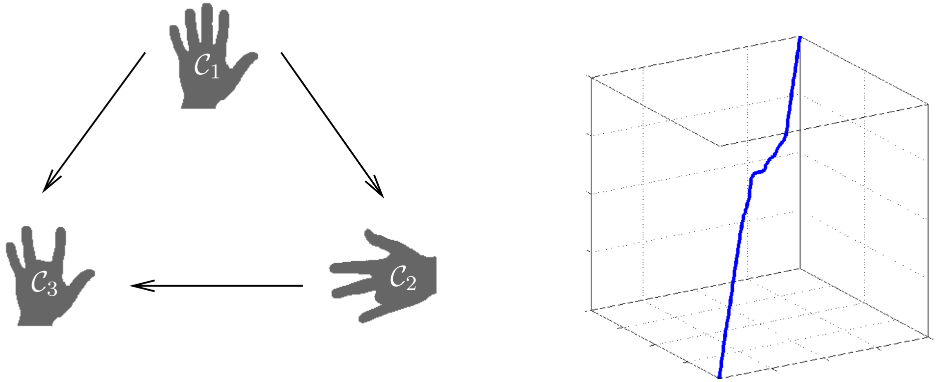


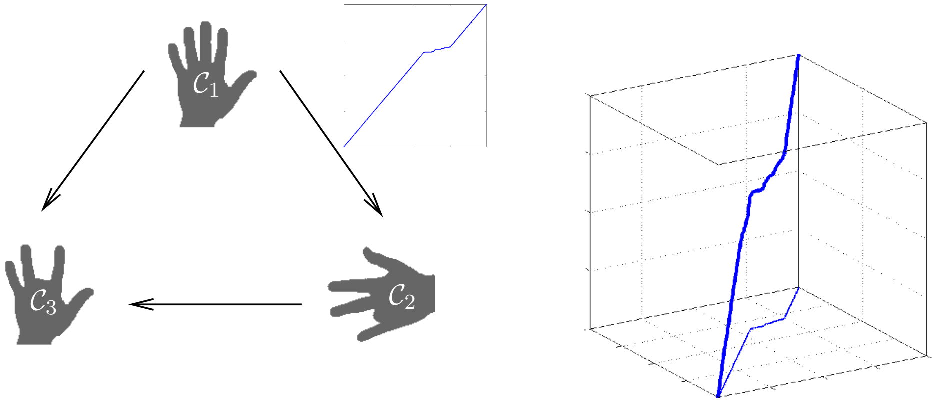
For d different shapes, the optimal matching $m = (m_1, \ldots, m_d) : \mathbb{S}^1 \to (\mathbb{S}^1)^d$ shall minimize the energy functional:

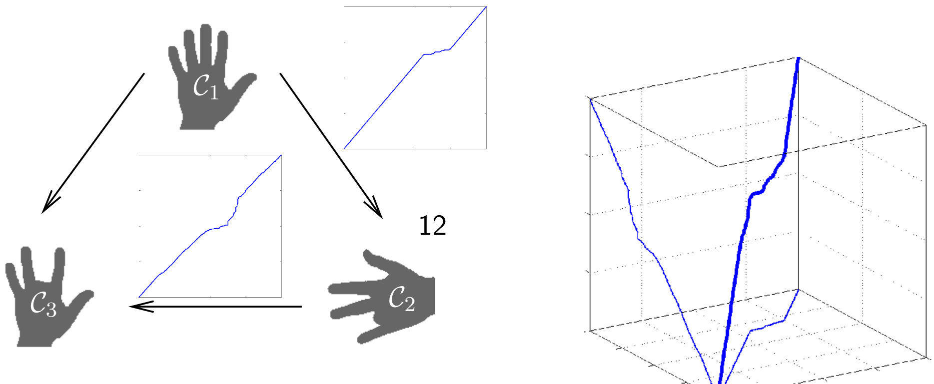
$$E(m) := \sum_{i,j=1}^{d} E_{S_i,S_j}(m_{ij}), \qquad m_{ij} = m_j \circ m_i^{-1}$$

and can be represented as a loop on the d-dimensional torus $\mathbb{S}^1 \times \ldots \times \mathbb{S}^1$.

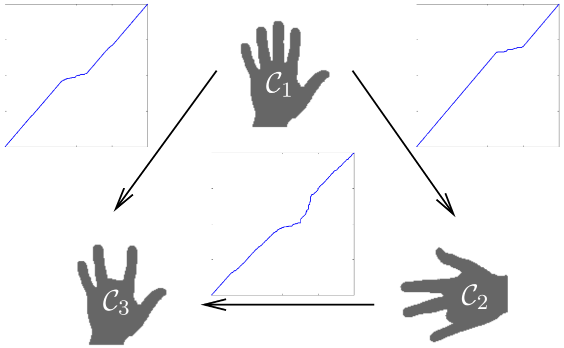
Relationship to 2-Matching

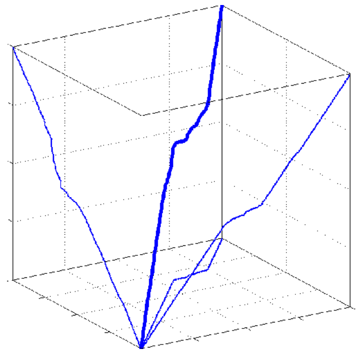






Running Time of d-Matching





Given an initial match of d shapes (discretized by N nodes each), results in a graph of N^d nodes and a running time for the optimal path of $\mathcal{O}(N^d)$.

Testing also all N^{d-1} initial matches leads us to a running time of $\mathcal{O}(N^{2d+1})$.

We were only able to reduce this running time for $d = 2$, because the resulting graph is planar and we can apply a binary search of the graph's domain.

Approximating Energy

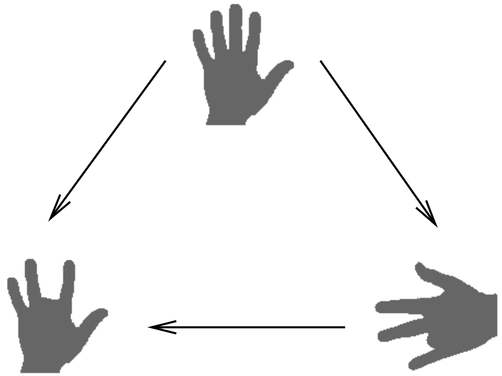
Instead of minimizing

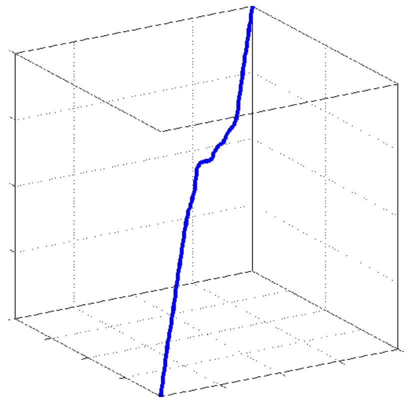
$$E(m_1, \ldots, m_d) = \sum_{i,j=1}^{d} \underbrace{E_{S_i,S_j}(m_{ij})}_{\text{pairwise matching}} \qquad\qquad m_{ij} := m_j \circ m_i^{-1}$$

we would like minimize the following energy

$$\hat{E}(m_{11}, \ldots, m_{dd}) = \sum_{i,j=1}^{d} \Big[\underbrace{E_{S_i,S_j}(m_{ij})}_{\text{pairwise matching}} + \gamma \cdot \int_{\Gamma(m_{ij})} \sum_{k=1}^{d} \underbrace{\|(m_{ij} - m_{kj} \circ m_{ik})(s)\|}_{\text{consistency costs}} \mathrm{ds} \Big]$$

Minimizing the Approximative Energy

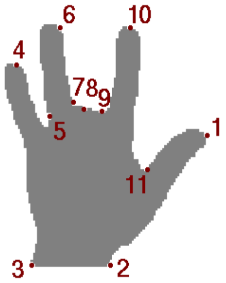


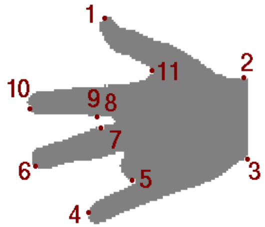


1. Start with the 2-matchings m_{ij} that minimize E_{S_1,S_2}.
2. Minimize the functional $\hat{E}(\cdot)$ with respect to m_{ij} for each m_{ij} until convergence.

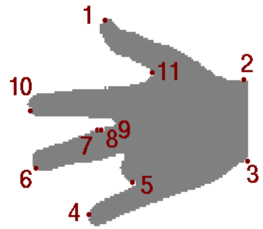
- Instead of $\mathcal{O}(N^d)$ graph nodes, we need only $\mathcal{O}(d^2 \cdot N^2)$.
- Instead of $\mathcal{O}(N^{2d-1})$ steps, every iteration needs only $\mathcal{O}(d^2 \cdot N^2 \log(N))$.

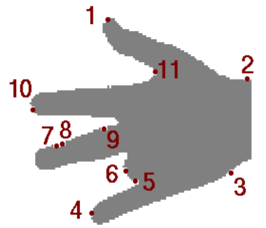
Results of the Approximation





$\gamma = 10^{-6}$





$\gamma = 10^{-4}$

Mean

If we take n samples $x_1, \ldots, x_n \in \mathbb{R}^n$ of an n-dimensional vector space, the mean is usually defined as

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

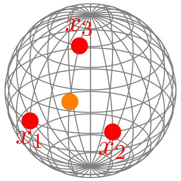
Interestingly, it is equivalent to the following formulation

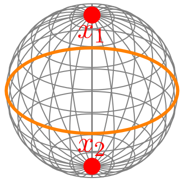
$$\mu = \operatorname*{argmin}_{x^* \in \mathbb{R}^n} \sum_{i=1}^{n} \|x_i - x^*\|^2$$

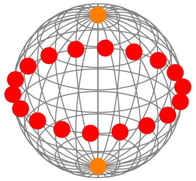
Thus, we can define a mean μ of n samples $x_1, \ldots, x_n \in X$ of a metrical or even semi-metrical space X as

$$\mu = \operatorname*{argmin}_{x^* \in X} \sum_{i=1}^{n} \operatorname{dist}(x_i, x^*)^2$$

Karcher Mean







Karcher was the first who introduced this mean for manifolds.
Note that a manifold M becomes a metric via

$$\mathrm{dist}(x, y) = \min_{\substack{c\colon [0,1]\to M,\\ c(0)=x, c(1)=y}} \mathrm{length}(c)$$

The Karcher mean is well-defined if the samples are close to one another.

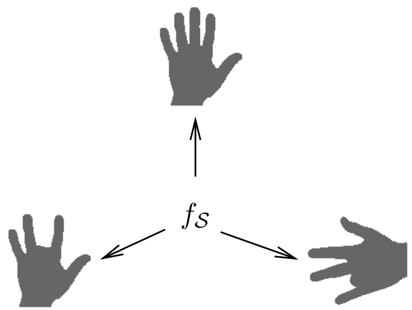
For samples that are widely spread on a manifold, there might be multiple Karcher means that minimize the involved energy function.

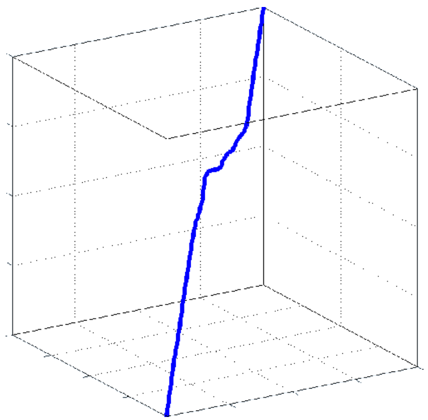
Matching-Driven Mean

We like to define the mean of a collection $\mathcal{S} = \{S_1, \ldots, S_d\}$ of shapes with respect to the (approximative) simultaneous shape matching functions $(m_{ij})_{i,j=1,\ldots,d}$

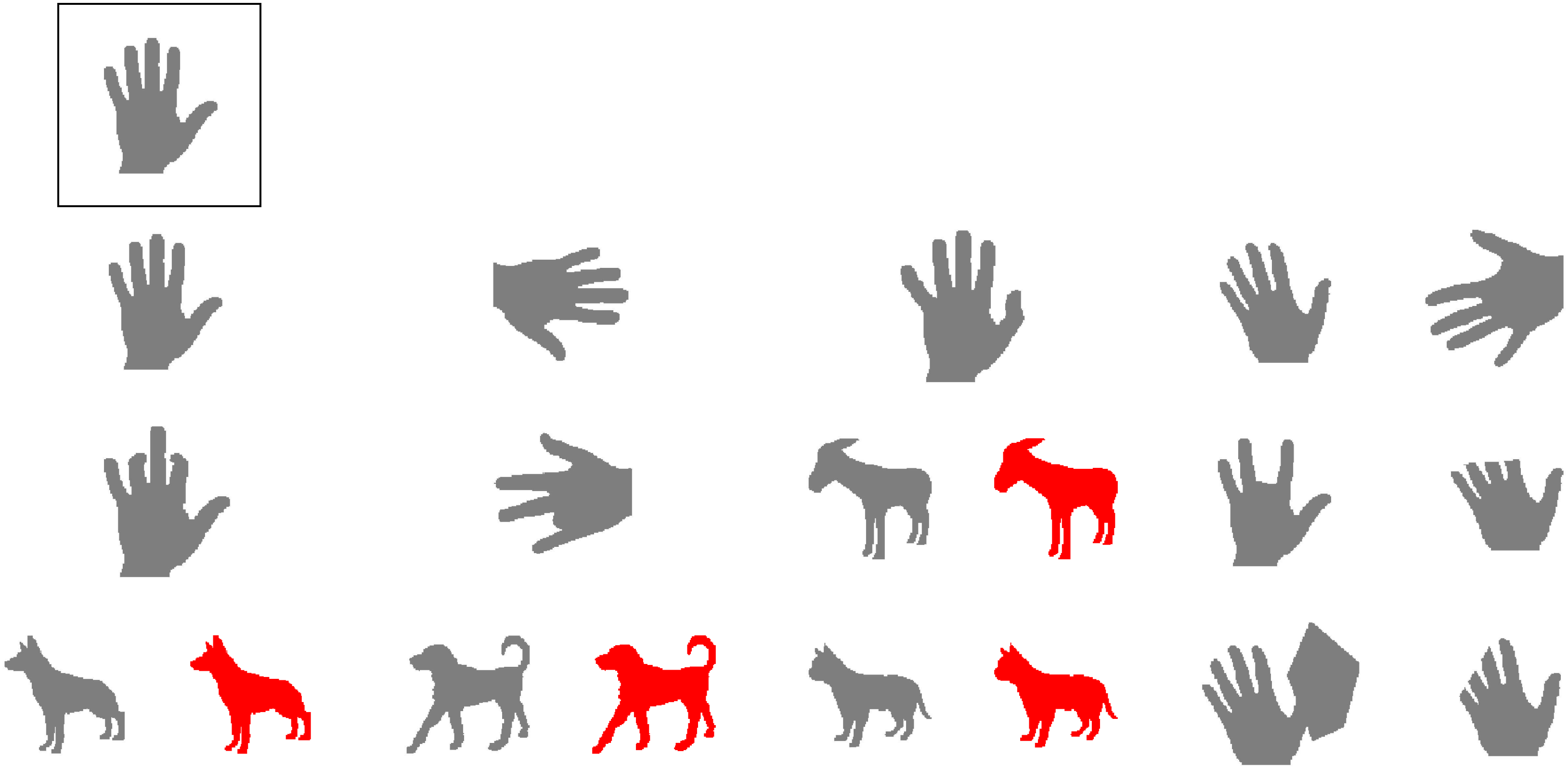
The feature representation $f_{\mathcal{S}}$ of this collection shall minimize the energy

$$f^* \mapsto \sum_{i=1}^{d} \int_{\Gamma(m_{1i})} \mathrm{dist}_{\mathcal{F}} \left(f^*(s_1), f_i(s_2)\right)^2 \mathrm{ds}.$$

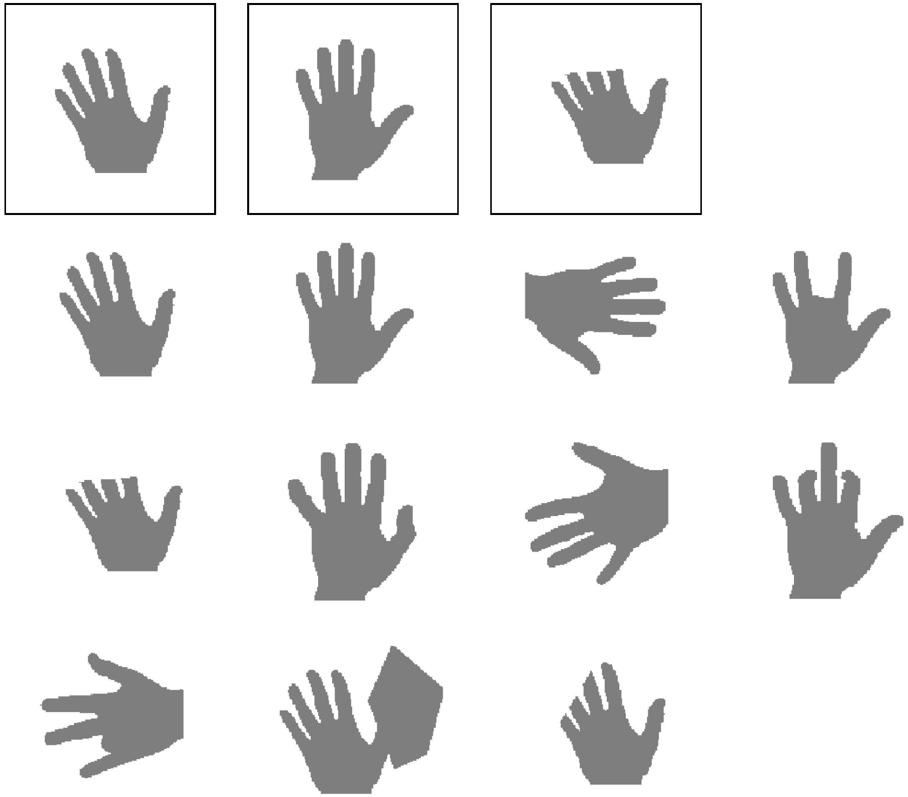




Shape Retrieval



Mean Shape Retrieval



Literature

MPEG7

- MPEG7 CE Shape-1 Part-B, `http://www.cis.temple.edu/~latecki/TestData/mpeg7shapeB.tar.gz`
- Belongie et al., *Shape Matching and Object Recognition Using Shape Context*, 2002, IEEE TPAMI (24) 24, 509–521.
- Bai et al., *Learning Context Sensitive Shape Similarity by Graph Transduction*, 2009, IEEE TPAMI (32) 5, 861–874.

Mean

- Karcher, *Riemannian center of mass and mollifier smoothing* , 1977, Comm. Pure Appl. Math. 30 (5), 509–541.
- Schmidt et al., *Intrinsic Mean for Semimetrical Shape Retrieval via Graph Cuts*, 2007, Pattern Recognition (Proc. DAGM), LNCS 4713, 446–455.