Face recognition with wave kernel signatures using a depth camera

Matthäus Brandl

Computer Vision Group
Department of Informatics
Technische Universität München

September, 14th 2012

Introduction - Problem Definition

Face Recognition:

- identify a given face
- using a database of previously trained faces

Introduction - Motivation

Applications of Face Recognition:

- access control
- surveillance
- identification

Introduction - Motivation

Applications of Face Recognition:

- access control
- surveillance
- identification

Figure: Example of identification: easy tagging in iPhoto

Introduction - Motivation

Why use point images?

Introduction - Motivation

Why use point images?

- sensors became affordable

Introduction - Motivation

Why use point images?

- sensors became affordable
- independent of illumination variations

Introduction - Motivation

Why use point images?

- sensors became affordable
- independent of illumination variations
- higher security

Introduction - Motivation

Why use point images?

- sensors became affordable
- independent of illumination variations
- higher security
- additional information

Introduction - Motivation

Why use point images?

- sensors became affordable
- independent of illumination variations
- higher security
- additional information
- possibly higher recognition rates

Introduction - Motivation

Why use point images?

- sensors became affordable
- independent of illumination variations
- higher security
- additional information
- possibly higher recognition rates

Disadvantages:

- more sensitive to facial expression variations

Introduction - Motivation

Why use point images?

- sensors became affordable
- independent of illumination variations
- higher security
- additional information
- possibly higher recognition rates

Disadvantages:

- more sensitive to facial expression variations
- not available in many day-to-day situations

Introduction - Topics of this Talk

Motivation: just covered

Introduction - Topics of this Talk

Motivation: just covered

Approach:

- data acquirement
- face detection
- face normalization
- face recognition

Introduction - Topics of this Talk

Motivation: just covered

Approach:

- data acquirement
- face detection
- face normalization
- face recognition
  - calculation of the geometry descriptor
  - determination of points of interest
  - signature extraction at points of interest
    $\Rightarrow$ concise face description

Introduction - Topics of this Talk

Motivation: just covered

Approach:

- data acquirement
- face detection
- face normalization
- face recognition
  - calculation of the geometry descriptor
  - determination of points of interest
  - signature extraction at points of interest
    $\Rightarrow$ concise face description

Results

Introduction - Topics of this Talk

Motivation: just covered

Approach:

- data acquirement
- face detection
- face normalization
- face recognition
  - calculation of the geometry descriptor
  - determination of points of interest
  - signature extraction at points of interest
    $\Rightarrow$ concise face description

Results

Conclusion

Problem Statement

Input:

- point image of a face
- face database

Problem Statement

Input:

- point image of a face
- face database

Output:

Face Enrollment:

- database with added face

Problem Statement

Input:

- point image of a face
- face database

Output:

Face Enrollment:

- database with added face

Face Recognition:

- identification of the face

Problem Statement

Input:

- point image of a face
- face database

Output:

Face Enrollment:

- database with added face

Face Recognition:

- identification of the face

Challenges:

- expression variations
- pose variations
- occlusions

Overview of the Approach

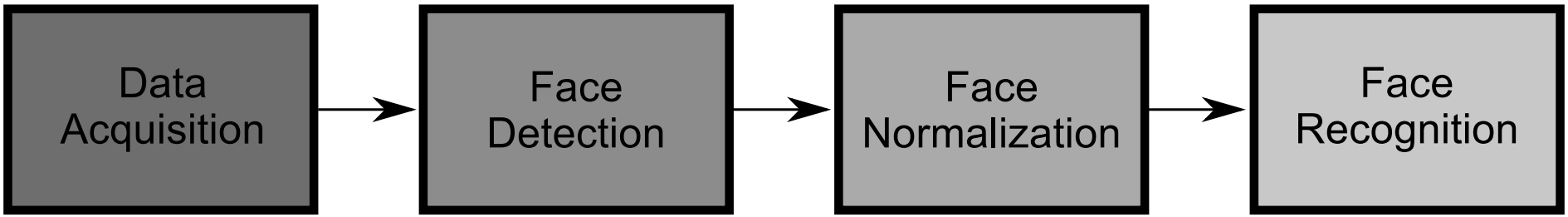
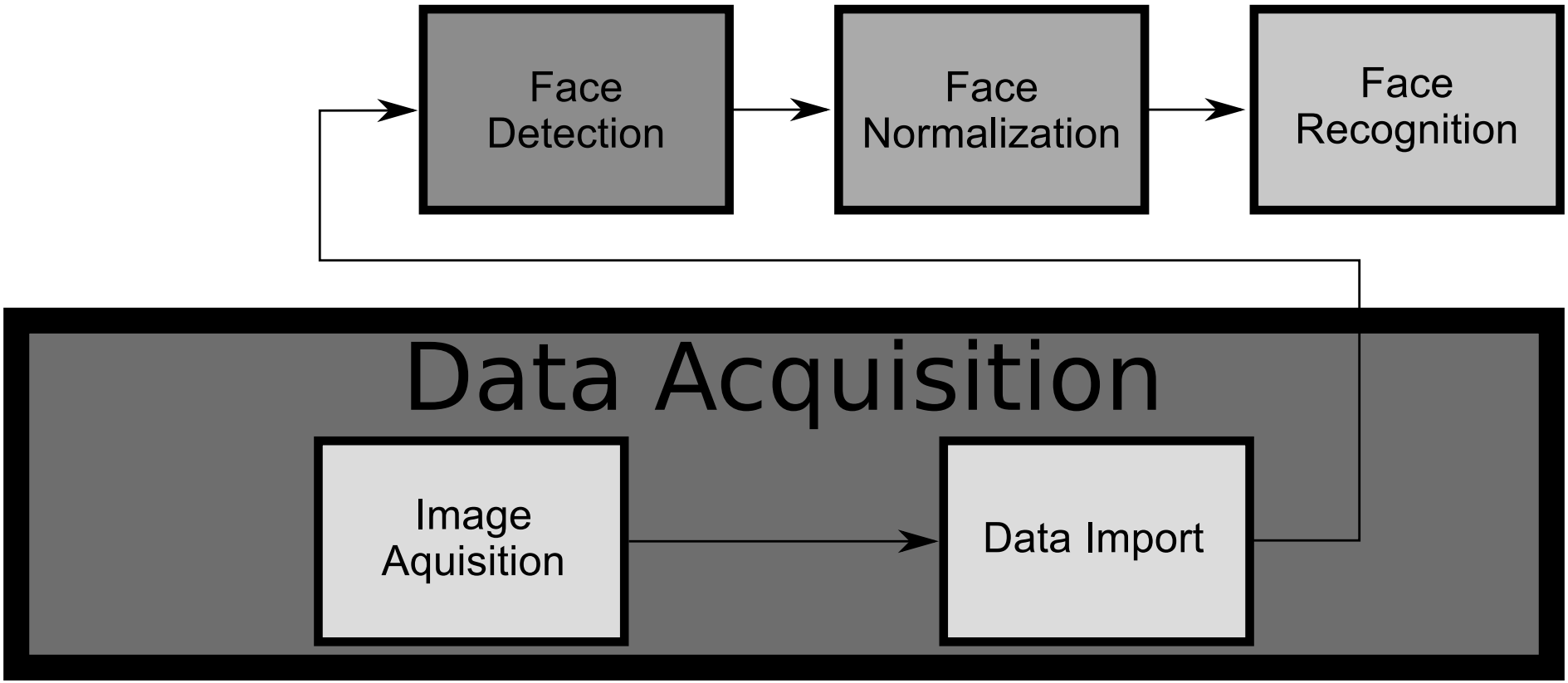


Figure: Data flow of our recognition approach

Data Acquisition - Overview



Data Acquisition - The Kinect depth camera

Data Acquisition - The Kinect depth camera

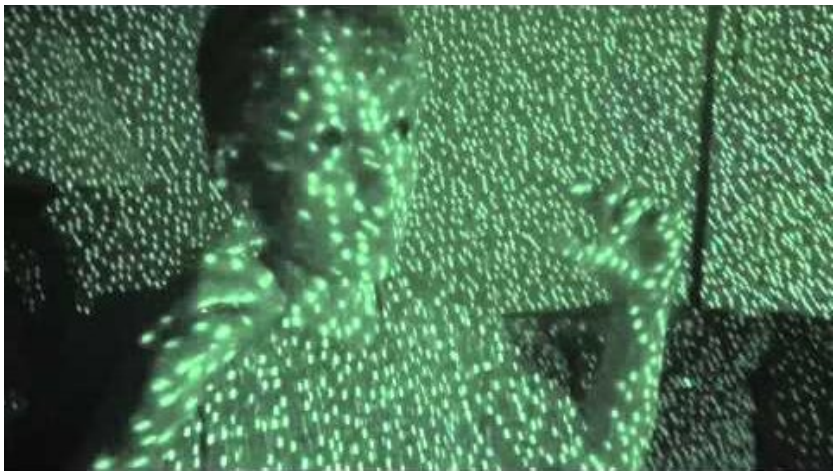
Working principle:

- disparity approach
- structured light is projected into environment
- depth is inferred from speckle deviation

Data Acquisition - The Kinect depth camera

Working principle:

- disparity approach
- structured light is projected into environment
- depth is inferred from speckle deviation



Point Images - Definition

- array of width w and height h
- each array element is a point in $\mathbb{R}^3$

Point Images - Definition

- array of width w and height h
- each array element is a point in $\mathbb{R}^3$
- all points are given in the same coordinate system

Point Images - Definition

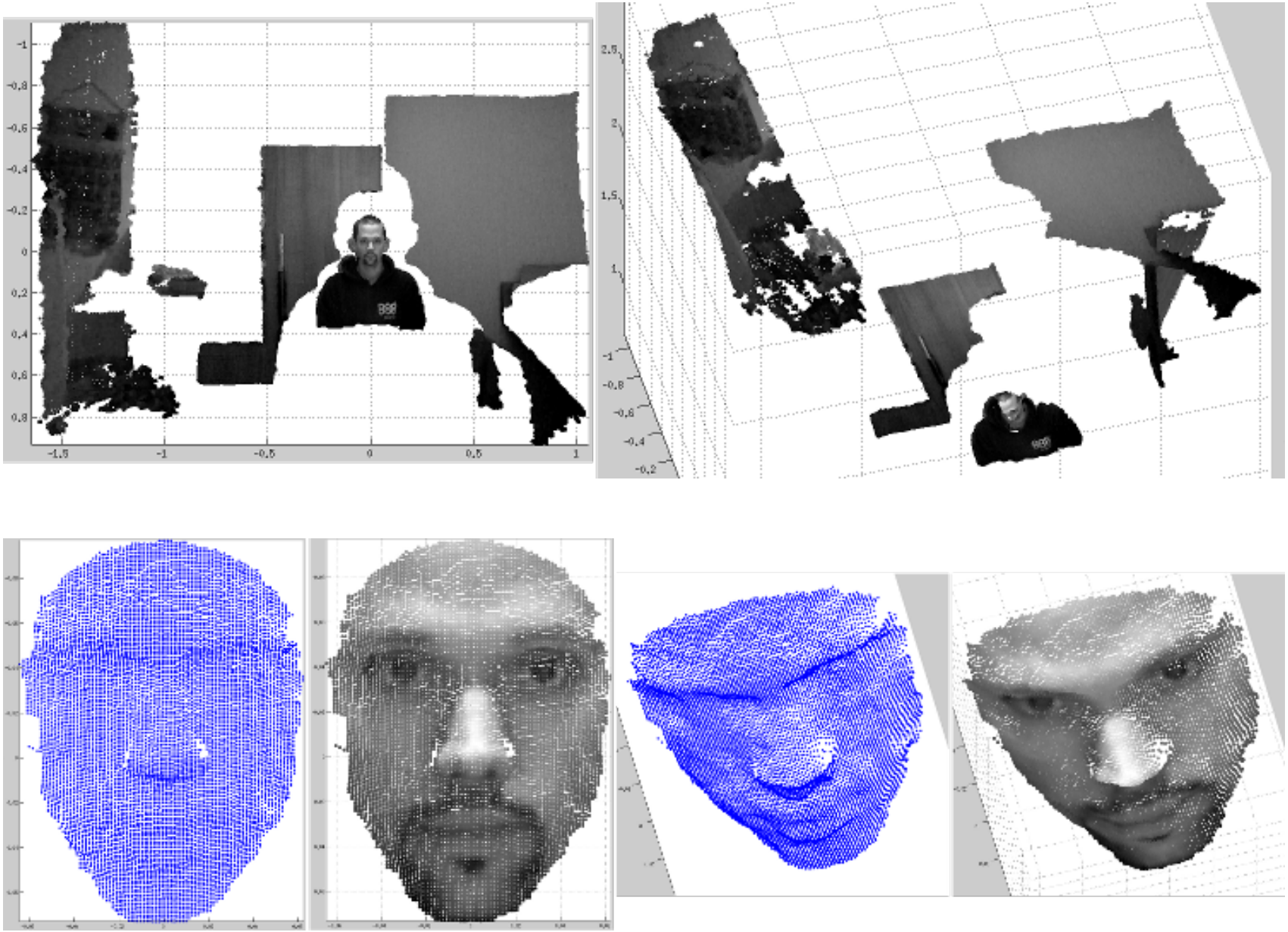
- array of width w and height h
- each array element is a point in $\mathbb{R}^3$
- all points are given in the same coordinate system
- not acquirable points are marked with a NaN vector

Point Images - Definition

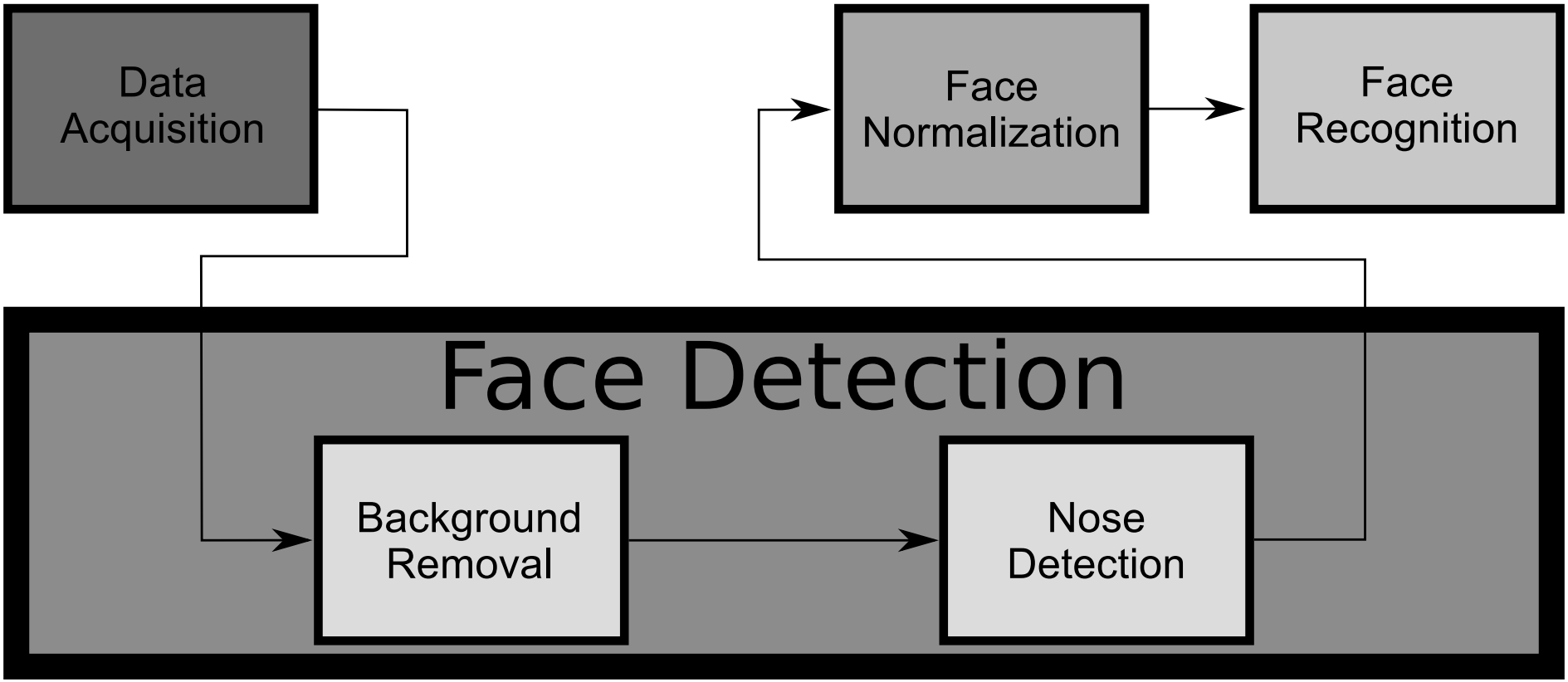
- array of width w and height h
- each array element is a point in $\mathbb{R}^3$
- all points are given in the same coordinate system
- not acquirable points are marked with a NaN vector

$$\Rightarrow \quad \mathcal{I}_P \in \left(\mathbb{R}^3 \cup (\perp \perp \perp)^T\right)^{w \times h}$$

Point Images - Examples



Face Detection - Overview



Face Detection - Background Removal

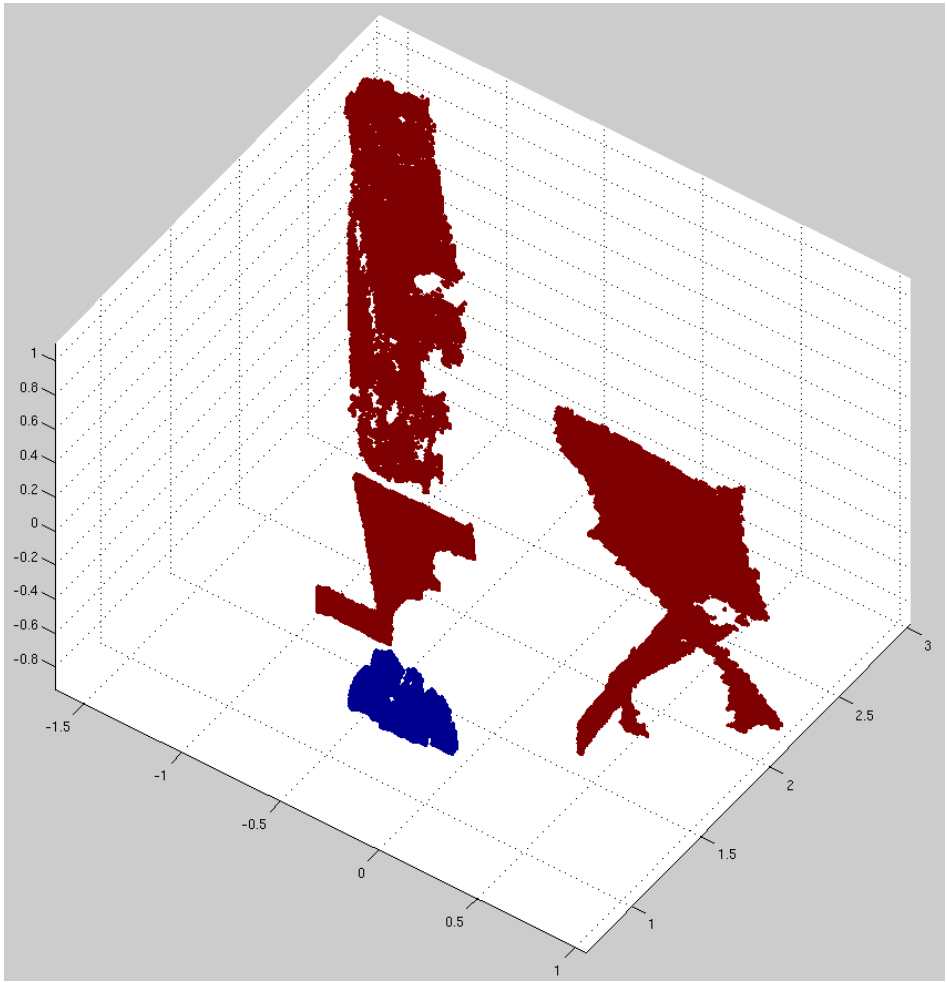


Figure: Segmented torso

Face Detection - Nose Detection

Nose Detection:

1. nose tip candiates: locally convex
2. inner eye corner candidates: locally concave

Face Detection - Nose Detection

Nose Detection:

1. nose tip candiates: locally convex
2. inner eye corner candidates: locally concave
3. geometric candidate filtering scheme

Face Detection - Nose Detection

Nose Detection:

1. nose tip candiates: locally convex
2. inner eye corner candidates: locally concave
3. geometric candidate filtering scheme
4. verification of remaining candidates using a model face

Face Detection - Nose Detection

Nose Detection:

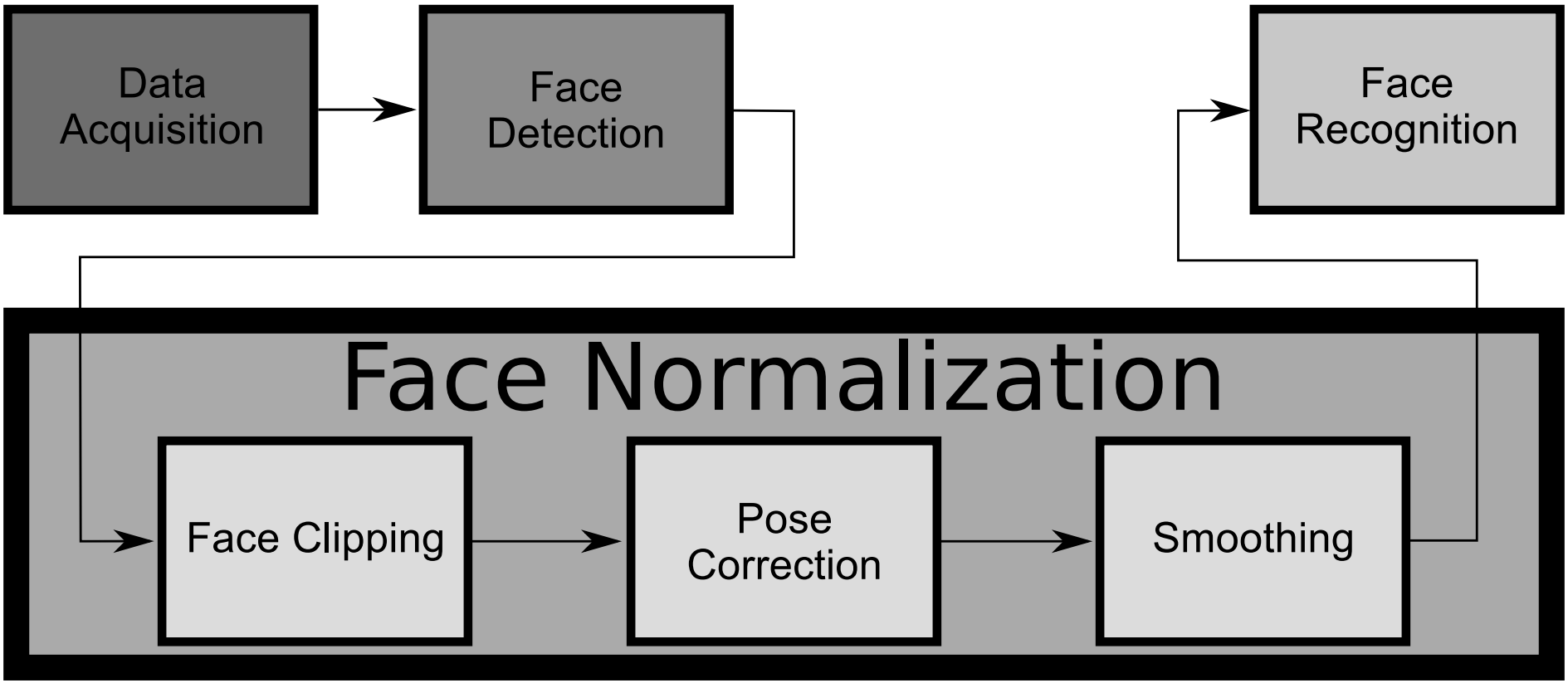
1. nose tip candiates: locally convex
2. inner eye corner candidates: locally concave
3. geometric candidate filtering scheme
4. verification of remaining candidates using a model face
5. choose nose tip candidate with least error

Face Detection - Nose Detection

Nose Detection:

1. nose tip candiates: locally convex
2. inner eye corner candidates: locally concave
3. geometric candidate filtering scheme
4. verification of remaining candidates using a model face
5. choose nose tip candidate with least error
6. unless it exceeds a threshold

Face Normalization - Overview



Face Normalization

Face Clipping:

- clip all points farther from the nose tip than a threshold

Pose Correction:

Smoothing:

Face Normalization

Face Clipping:

Pose Correction:

- translate nose tip to a defined position
- determine pose using the nose tip and the inner eye corners
- rotate the face into a defined configuration

Smoothing:

Face Normalization

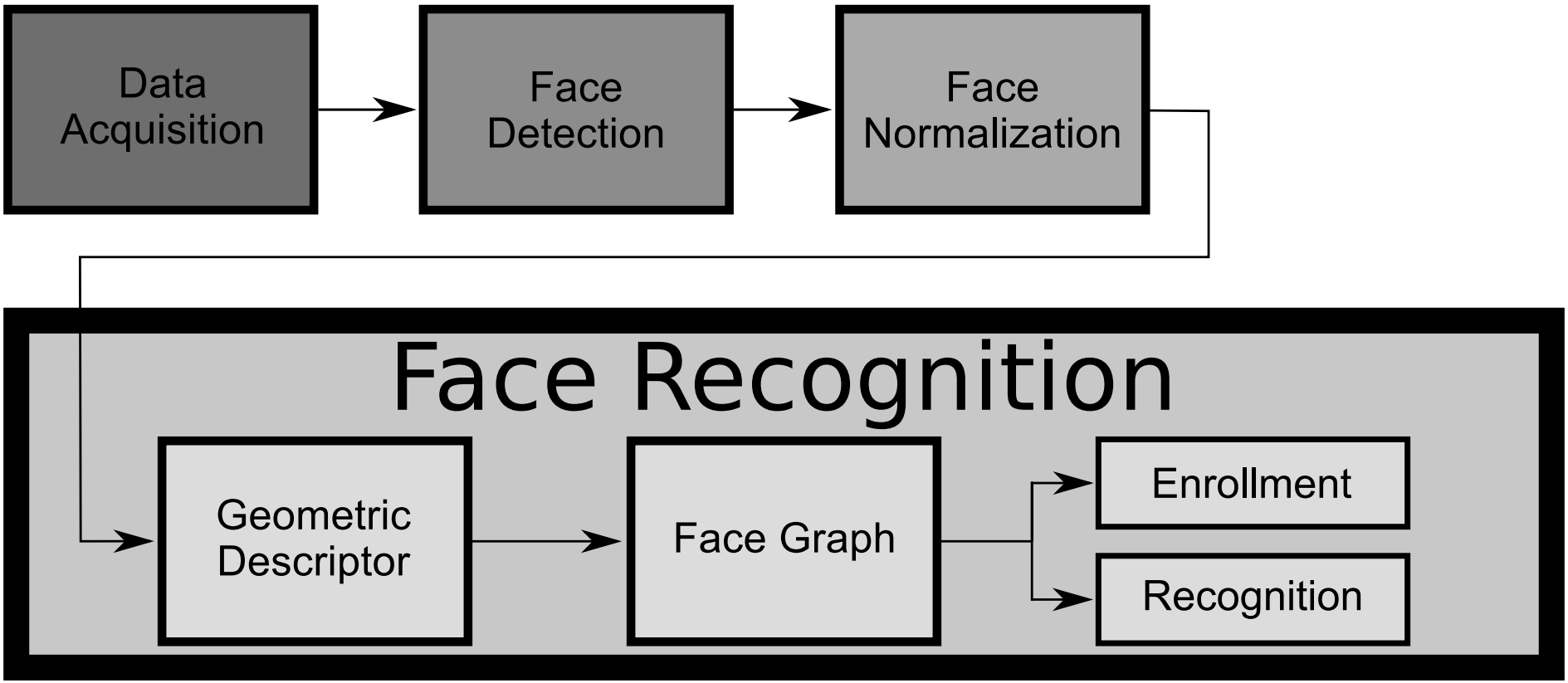
Face Clipping:

Pose Correction:

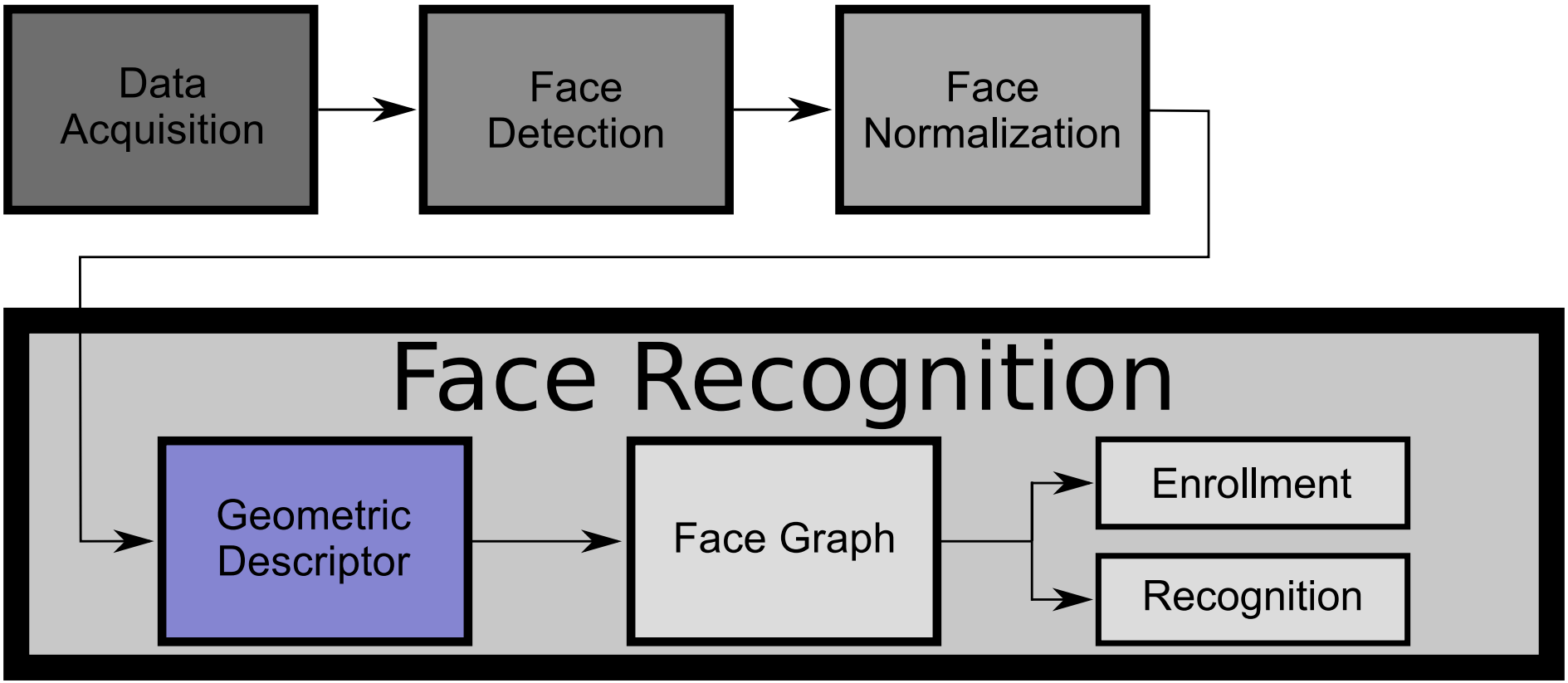
Smoothing:

- apply a Binomial filter to smoothen the depth values

Face Recognition - Overview



Face Recognition - Geometry Descriptor



Wave Kernel Signature - Derivation

Basic idea:

- quantum particle on a surface $\mathcal{X}$
- location is unknown
- approximate energy E is measured at time $t = 0$

Wave Kernel Signature - Derivation

Basic idea:

- quantum particle on a surface $\mathcal{X}$
- location is unknown
- approximate energy E is measured at time $t = 0$

$$\Rightarrow \text{energy probability distribution } f_E^2(x, t)$$

Wave Kernel Signature - Derivation

Energy probability distribution:

- governed by wave function $\psi(x, t)$

Wave Kernel Signature - Derivation

Energy probability distribution:

- governed by wave function $\psi(x, t)$

Wave function:

- solution to the Schrödinger equation $\frac{\partial \psi}{\partial t}(x, t) = i\Delta\psi(x, t)$, where Δ is the Laplace-Beltrami operator of surface $\mathcal{X}$

Wave Kernel Signature - Derivation

The Laplace-Beltrami operator can be decomposed into:

- eigenfunctions ϕ_k
- "eigenenergies" E_k

Wave Kernel Signature - Derivation

The Laplace-Beltrami operator can be decomposed into:

- eigenfunctions ϕ_k
- "eigenenergies" E_k

Using them the wave function can be given as:

$$\psi_E(x,t) = \sum_{k=0}^{\infty} \exp(iE_k t)\,\phi_k(x)\,f_E(E_k)$$

Wave Kernel Signature - Definition

The WKS is introduced as:

$$\mathrm{WKS}(E, x) = \lim_{T \to \infty} \frac{1}{T} \int_0^T |\psi_E(x, t)|^2 \, dt$$

$\mathrm{WKS}(E, x)$ is

- over time average probability
- to measure the quantum particle
- with initial energy E
- at point $x \in \mathcal{X}$ on the surface

Wave Kernel Signature - Definition

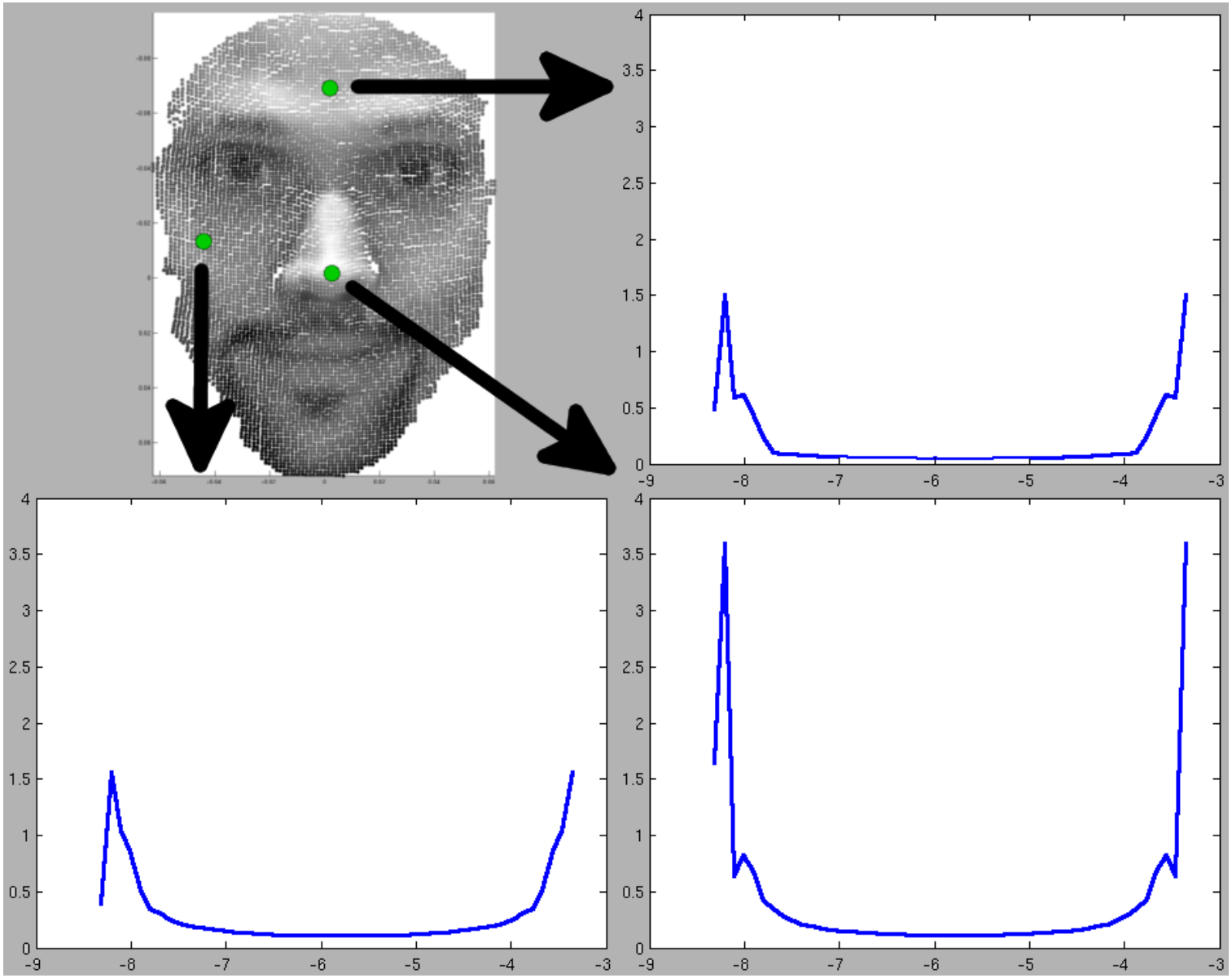
Definition (Wave Kernel Signature)

The WKS for a point $x \in \mathcal{X}$ and initial energy E is given as

$$\mathrm{WKS}(E, x) := \sum_{k=0}^{\infty} \phi_k(x)^2 \, f_E(E_k)^2$$

where f_E is the energy probability distribution for the initial energy E.

Wave Kernel Signature - Example



Wave Kernel Signature - Implementation

Finite approximation L of Laplace-Beltrami operator Δ is needed

Wave Kernel Signature - Implementation

Finite approximation L of Laplace-Beltrami operator Δ is needed

Idea:

- convert point image into a graph
- use weighted Graph Laplacian as approximation

Wave Kernel Signature - Implementation

Finite approximation L of Laplace-Beltrami operator Δ is needed

Idea:

- convert point image into a graph
- use weighted Graph Laplacian as approximation

Pro:

- easy to understand
- straightforward to implement

Wave Kernel Signature - Implementation

Finite approximation L of Laplace-Beltrami operator Δ is needed

Idea:

- convert point image into a graph
- use weighted Graph Laplacian as approximation

Pro:

- easy to understand
- straightforward to implement

Contra:

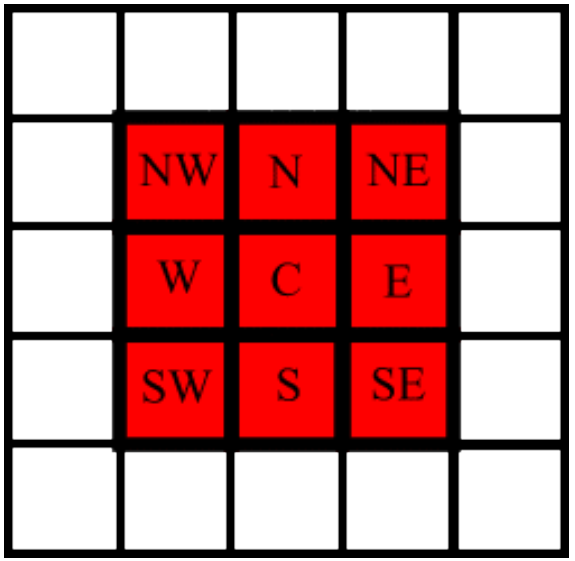
- does not converge to Δ for increasingly finer discretizations

Implementation - Conversion into a Graph

1. create a graph vertex for every element of the point image

Implementation - Conversion into a Graph

1. create a graph vertex for every element of the point image
2. add edges between each vertex and the vertices of its 8-connected-neighborhood



Implementation - Conversion into a Graph

1. create a graph vertex for every element of the point image
2. add edges between each vertex and the vertices of its 8-connected-neighborhood
3. add edge weights depending on the vertex distances

$$w_{ij} = \exp\left(-\frac{\|v_i - v_j\|^2}{t}\right)$$

Implementation - Weighted Graph Laplacian

Definition (Adjacency Matrix)

$$A(i,j) = \begin{cases} w_{ij} = \exp\left(-\frac{\|v_i - v_j\|^2}{t}\right), & \text{if } \{v_i, v_j\} \in E \\ 0, & \text{else} \end{cases}$$

Implementation - Weighted Graph Laplacian

Definition (Adjacency Matrix)

$$A(i,j) = \begin{cases} w_{ij} = \exp\left(-\frac{\|v_i - v_j\|^2}{t}\right), & \text{if } \{v_i, v_j\} \in E \\ 0, & \text{else} \end{cases}$$

Definition (Degree Matrix)

$$D(i,j) = \begin{cases} \sum_{k=1}^{n} A(i,k), & \text{if } i = j \\ 0, & \text{else} \end{cases}$$

Implementation - Weighted Graph Laplacian

Definition (Adjacency Matrix)

$$A(i,j) = \begin{cases} w_{ij} = \exp\left(-\frac{\|v_i - v_j\|^2}{t}\right), & \text{if } \{v_i, v_j\} \in E \\ 0, & \text{else} \end{cases}$$

Definition (Degree Matrix)

$$D(i,j) = \begin{cases} \sum_{k=1}^{n} A(i,k), & \text{if } i = j \\ 0, & \text{else} \end{cases}$$

Definition (Graph Laplacian)

$$L = D - A$$

Laplacian Eigenfunctions

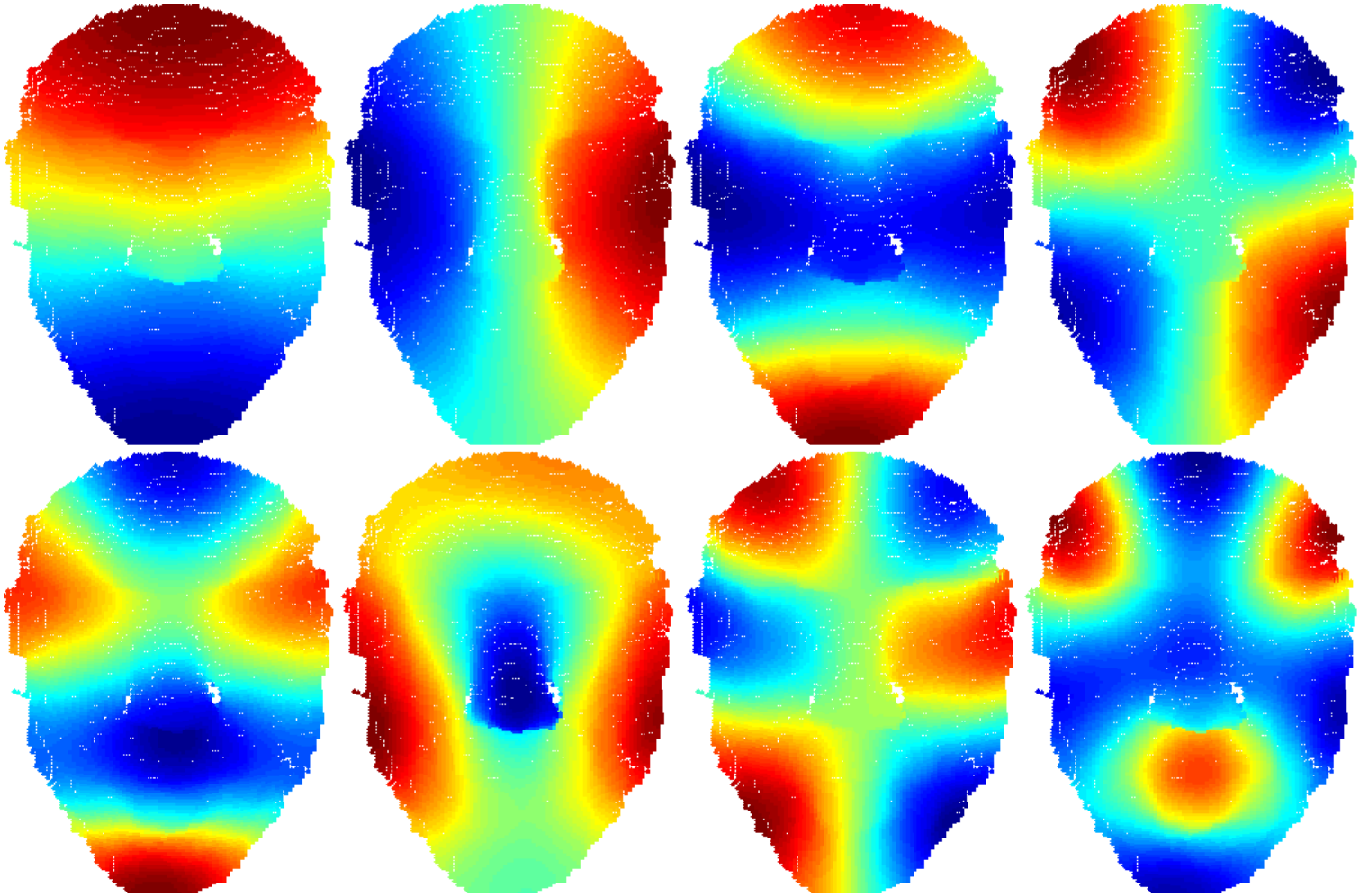


Figure: Corresponding eigenfunctions of the 8 least non-zero eigenenergies

Wave Kernel Signature - Examples

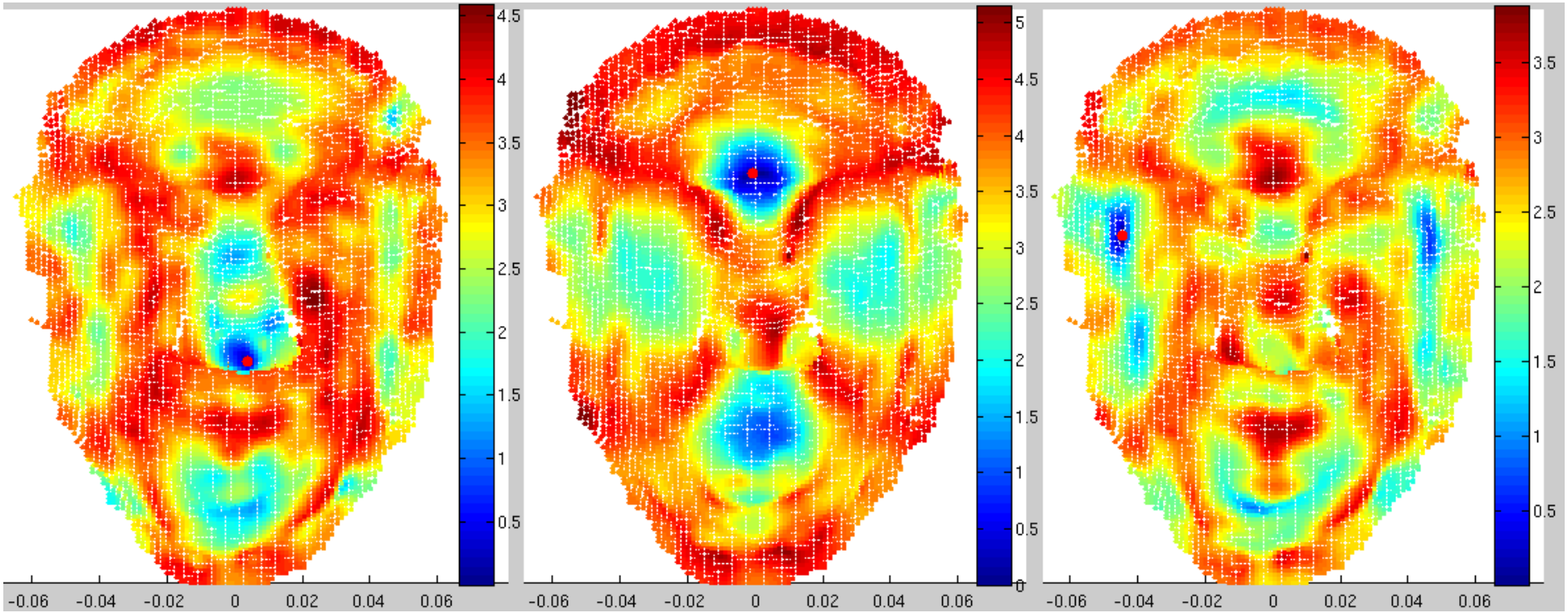
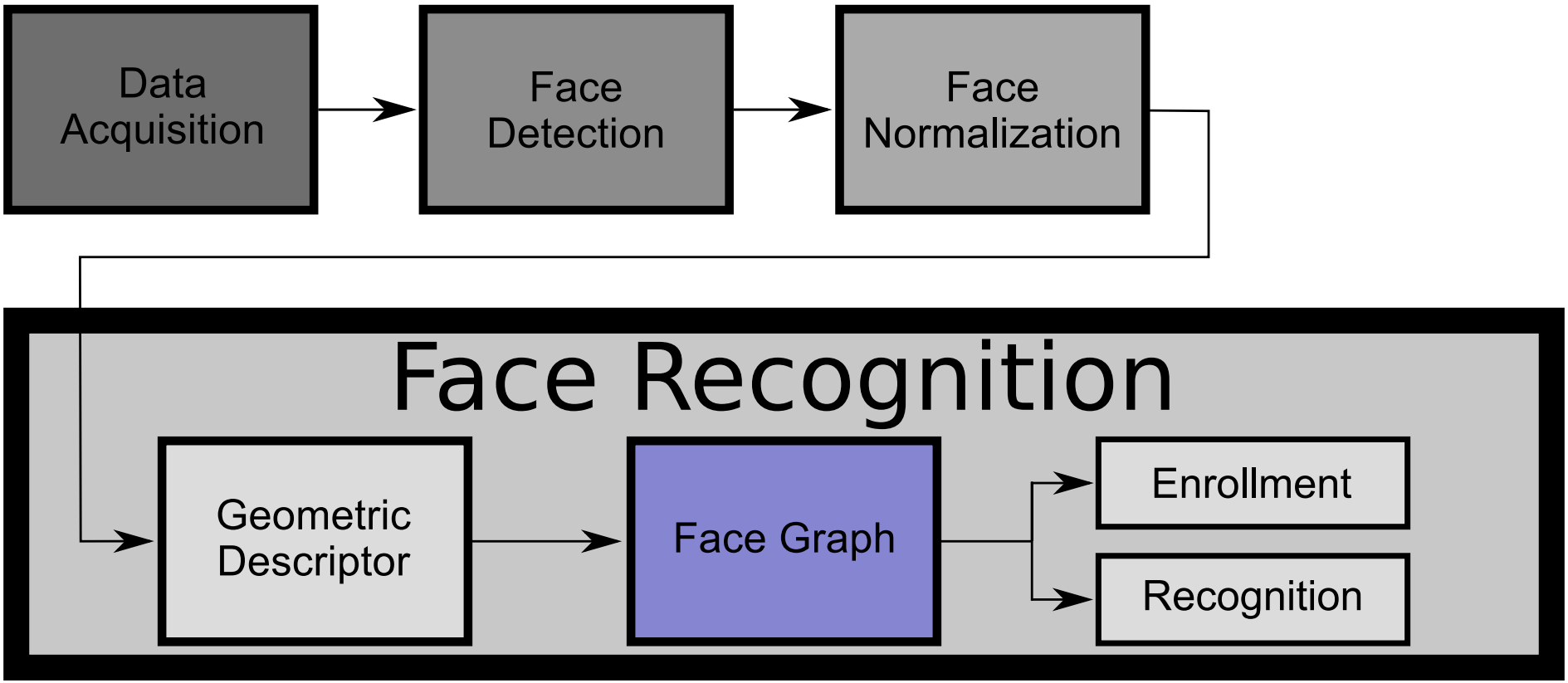
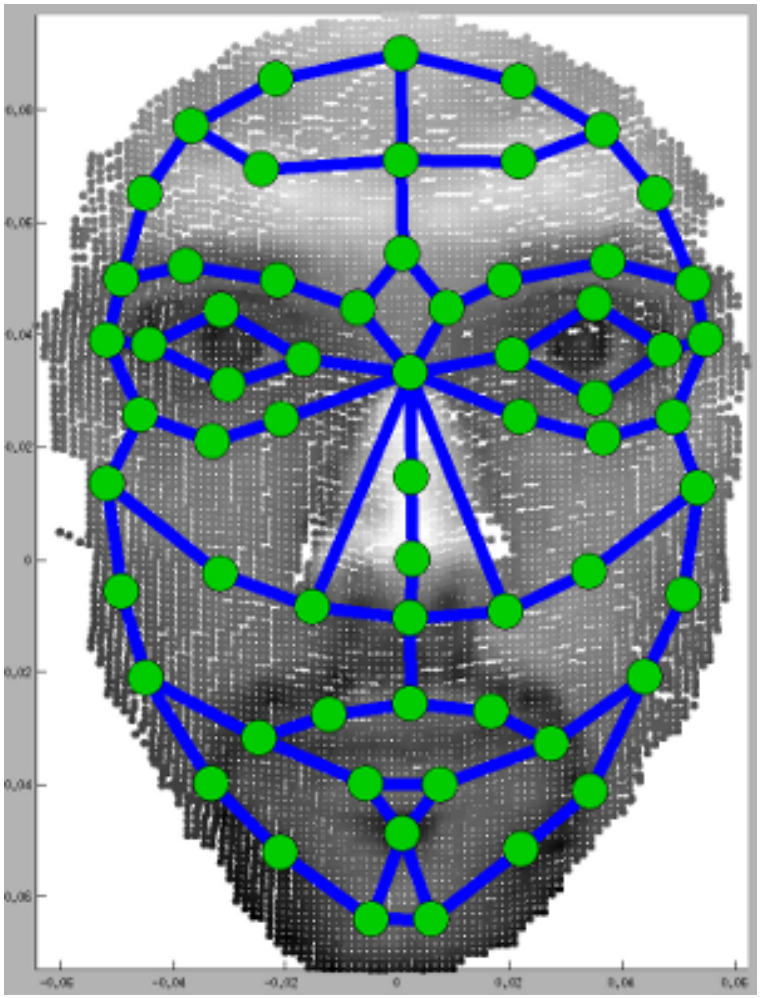


Figure: Signature distance of several points

Face Recognition - Face Graph



Face Graph - Model



Face Graph - Matching

1. match vertices to a new face using ICP

Face Graph - Matching

1. match vertices to a new face using ICP
2. determine nearest neighbor of every model graph vertex

Face Graph - Matching

1. match vertices to a new face using ICP
2. determine nearest neighbor of every model graph vertex
3. make them the corresponding vertices of the new graph

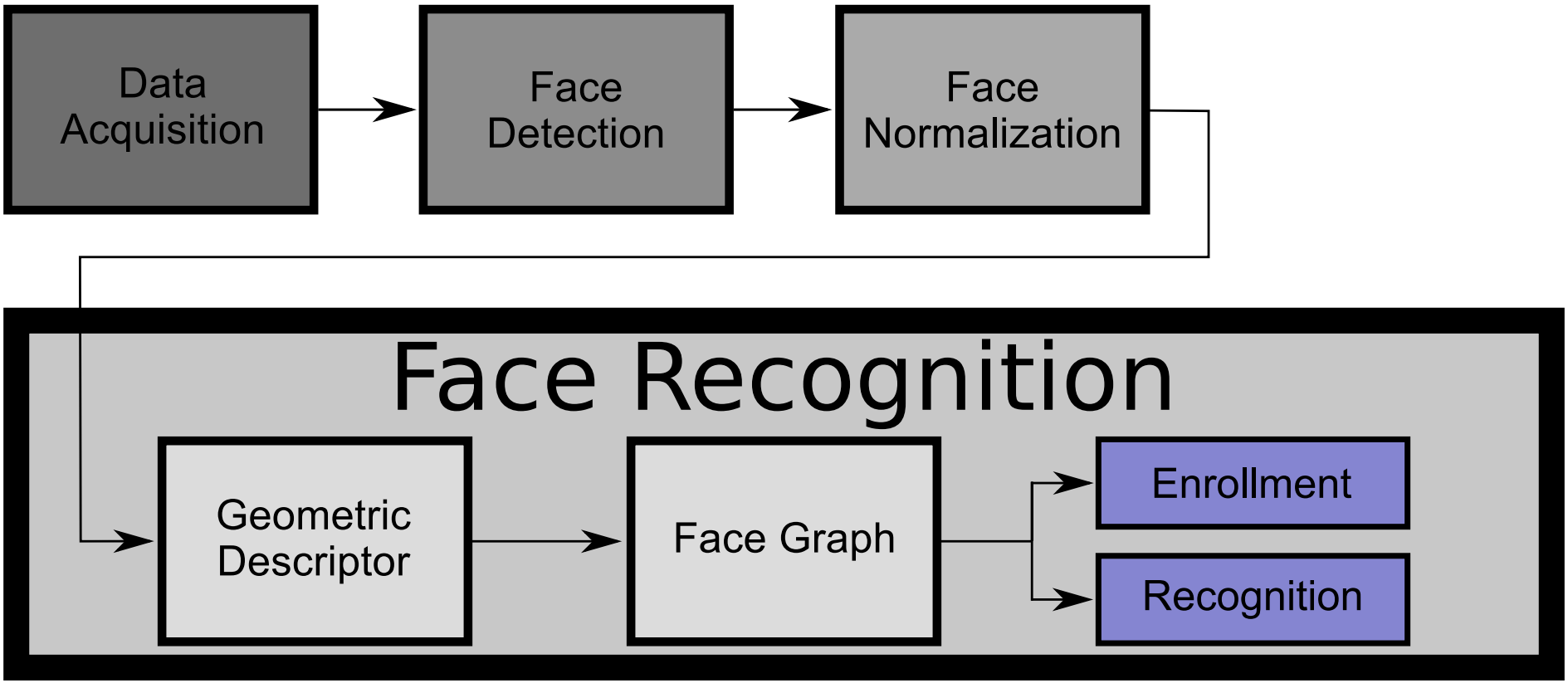
Face Graph - Matching

1. match vertices to a new face using ICP
2. determine nearest neighbor of every model graph vertex
3. make them the corresponding vertices of the new graph
4. connect the new vertices as the model graph vertices

Face Graph - Matching

1. match vertices to a new face using ICP
2. determine nearest neighbor of every model graph vertex
3. make them the corresponding vertices of the new graph
4. connect the new vertices as the model graph vertices
5. manual graph correction

Face Recognition - Enrollment and Recognition



Face Recognition - Enrollment and Recognition

1. extract the WKS at every graph vertex
   $\Rightarrow$ concise face description

Face Recognition - Enrollment and Recognition

1. extract the WKS at every graph vertex
   $\Rightarrow$ concise face description
2. enrollment: store the signatures in a database

Face Recognition - Enrollment and Recognition

1. extract the WKS at every graph vertex
   $\Rightarrow$ concise face description

3. recognition: compare the signatures with those in the database

Face Recognition - Enrollment and Recognition

1. extract the WKS at every graph vertex
   $\Rightarrow$ concise face description

3. recognition: compare the signatures with those in the database
4. select the entry with the least signature distance

$$d_{\mathrm{WKS}}\left(x, x'\right)^2 = \int_{e_{min}}^{e_{max}} \left(\mathrm{WKS}\left(x, e\right) - \mathrm{WKS}\left(x', e\right)\right)^2 de$$

Face Recognition - Experimental Setup

- we recorded point images of 54 persons, a total of 2061 images
- one image per person used for training
- three images per person used for testing

  $\Rightarrow$ total of 157 recognition runs

Face Recognition - Experimental Results

Rank 1 recognition rate:	94	out of	157	59.9%
Rank 3 recognition rate:	114	out of	157	72.6%
Rank 6 recognition rate:	130	out of	157	82.8%

Table: Overall recognition rates

Face Recognition - Experimental Results

Rank 1 recognition rate:	94	out of	157	59.9%
Rank 3 recognition rate:	114	out of	157	72.6%
Rank 6 recognition rate:	130	out of	157	82.8%

Table: Overall recognition rates

Compilation of the graph Laplacian:	2.79	seconds	30.0%
Eigendecomposition of the graph Laplacian:	6.19	seconds	66.6%
Calculation of the WKS:	0.24	seconds	1.0%
Transformation into the SIWKS:	0.15	seconds	1.6%
Recognition:	0.07	seconds	0.8%
Total	*9.29*	*seconds*	

Table: Run-times

Conclusions

Conclusions:

- recognition rate of 59.9%
- by a factor of 30 above the expected value of random choice (=1.9%)

Conclusions

Conclusions:

- recognition rate of 59.9%
- by a factor of 30 above the expected value of random choice (=1.9%)
- neutral and non-neutral expressions are recognized with equal rates
  $\Rightarrow$ WKS is well suited to handle expression variations

Conclusions

Conclusions:

- recognition rate of 59.9%
- by a factor of 30 above the expected value of random choice (=1.9%)
- neutral and non-neutral expressions are recognized with equal rates
  $\Rightarrow$ WKS is well suited to handle expression variations
- method is stable under mild occlusions and mild pose variations

Conclusions

Conclusions:

- recognition rate of 59.9%
- by a factor of 30 above the expected value of random choice (=1.9%)
- neutral and non-neutral expressions are recognized with equal rates
  $\Rightarrow$ WKS is well suited to handle expression variations
- method is stable under mild occlusions and mild pose variations
- scales very good with database size
  theoretical runtime for 1000 DB entries is 1.4 seconds

Conclusions - Future Work

Possible Future Work:

- replace graph Laplacian with *Point Cloud Data Laplacian*

Conclusions - Future Work

Possible Future Work:

- replace graph Laplacian with *Point Cloud Data Laplacian*
- improve model graph matching

Conclusions - Future Work

Possible Future Work:

- replace graph Laplacian with *Point Cloud Data Laplacian*
- improve model graph matching
- weighted signature distance

Questions?

Thank you for your attention

Any questions?