Density-Based Shape Descriptors and Similarity Learning for 3D Object Retrieval

Ceyhun Burak Akgül[1,2]

[1]Boğaziçi University Electrical and Electronics Engineering Dept.
Istanbul, Turkey

[2]ENST Signal and Image Processing Dept. (ENST/TSI)
Paris, France

Thesis Advisers
Bülent Sankur (Boğaziçi University)
Francis Schmitt (ENST/TSI)

Information Retrieval

Information Retrieval

In text-based retrieval:

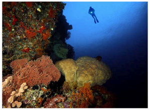
- Query is a set of *words*.
- Search results are typically based on *occurrence*.

Information Retrieval for Multimedia

Question

What if we want to look for multimedia data?

Information Retrieval for Multimedia



Search Engine

Search Results

Question

What if we want to look for multimedia data?

- **Image/Photo**

Information Retrieval for Multimedia

Question

What if we want to look for multimedia data?

- Image/Photo
- **Music**

Information Retrieval for Multimedia

→ **Search Engine** → ***Search Results***

Question

What if we want to look for multimedia data?

- Image/Photo
- Music
- **Video**

Information Retrieval for Multimedia

Question

What if we want to look for multimedia data?

- Image/Photo
- Music
- Video
- **3D Data**

Information Retrieval for Multimedia

Question

What if we want to look for multimedia data?

- Image/Photo
- Music
- Video
- **3D Data**

Information Retrieval for Multimedia

Question

What if we want to look for multimedia data?

- Image/Photo
- Music
- Video
- **3D Data**

Information Retrieval for Multimedia

Question

What if we want to look for multimedia data?

- Image/Photo
- Music
- Video
- **3D Data**

Information Retrieval for Multimedia

Question

What if we want to look for multimedia data?

- Image/Photo
- Music
- Video
- **3D Data**

Information Retrieval for Multimedia

Question

What if we want to look for multimedia data?

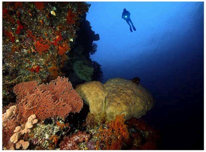
- Image/Photo
- Music
- Video
- **3D Data**

Information Retrieval for Multimedia

For text-based retrieval of multimedia data:

- Query should be given a textual description:

Information Retrieval for Multimedia



= “diving” + “underwater” + “coral” + ...

For text-based retrieval of multimedia data:

- Query should be given a textual description:

Information Retrieval for Multimedia

= "amphora with two handles"

For text-based retrieval of multimedia data:

- Query should be given a textual description:

Information Retrieval for Multimedia

= ?

For text-based retrieval of multimedia data:

- Query should be given a textual description:

Information Retrieval for Multimedia

 = ?

For text-based retrieval of multimedia data:

- Query should be given a textual description: **not always possible!**

Information Retrieval for Multimedia

 = ?

For text-based retrieval of multimedia data:

- Query should be given a textual description:
  not always possible!
- Database items should be annotated beforehand:
  extremely laborious!

Content-Based Retrieval (CBR)

In CBR

- No textual description
- No need for database annotation

Content-Based Retrieval (CBR)

In CBR

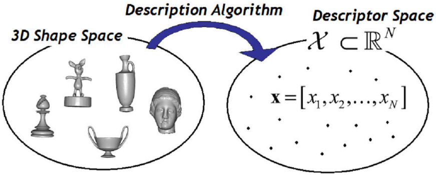
- No textual description
- No need for database annotation

Major Issues

1. How to describe the multimedia content?
   Content Description Problem
2. How to evaluate the relevance between entities?
   Similarity Problem

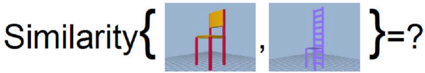
Scope and Contributions

1

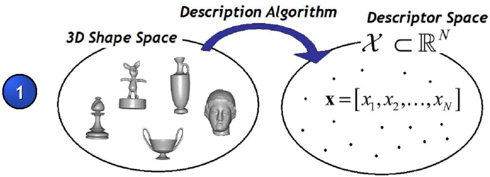


1. Content Description for **3D Objects**
2. Similarity Learning for CBR

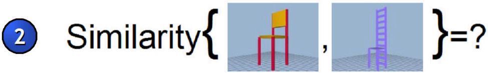
2



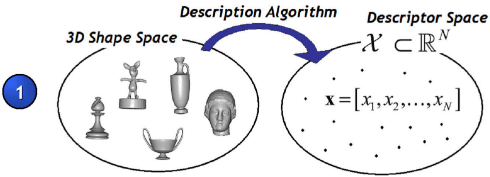
Scope and Contributions



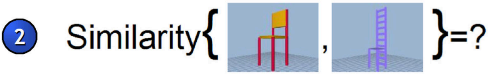
1. Content Description for **3D Objects**
   Density-Based Shape Description Framework
2. Similarity Learning for CBR



Scope and Contributions



1. Content Description for **3D Objects**
   Density-Based Shape Description Framework
2. Similarity Learning for CBR
   Similarity Score Fusion by Ranking Risk Minimization



Outline

Outline

3D Object Retrieval

Generic Retrieval Algorithm

3D Object Retrieval

Generic Retrieval Algorithm

- A **query object** Q
- **Database objects** $O_t, t = 1, \ldots, T$
- **Database descriptors** $\mathbf{f}_t$ (one for each O_t)

3D Object Retrieval

Generic Retrieval Algorithm

- A query object Q
- Database objects $O_t, t = 1, \ldots, T$
- Database descriptors $\mathbf{f}_t$ (one for each O_t)

1. Compute the descriptor $\mathbf{f}_q$ for Q

3D Object Retrieval

Generic Retrieval Algorithm

- A query object Q
- Database objects $O_t, t = 1, \ldots, T$
- Database descriptors $\mathbf{f}_t$ (one for each O_t)

1. Compute the descriptor $\mathbf{f}_q$ for Q
2. For each t, calculate a **similarity score** between Q and O_t

$$sim_t = sim(Q, O_t) = \varphi(\mathbf{f}_q, \mathbf{f}_t)$$

3D Object Retrieval

Generic Retrieval Algorithm

- A query object Q
- Database objects $O_t, t = 1, \ldots, T$
- Database descriptors $\mathbf{f}_t$ (one for each O_t)

1. Compute the descriptor $\mathbf{f}_q$ for Q
2. For each t, calculate a similarity score between Q and O_t

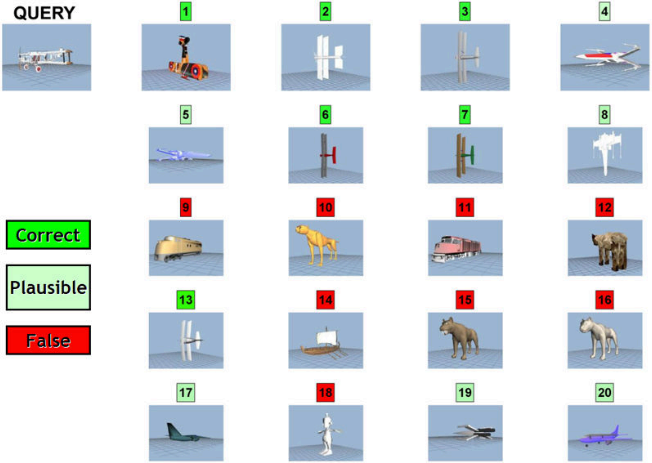
$$sim_t = sim(Q, O_t) = \varphi(\mathbf{f}_q, \mathbf{f}_t)$$

3. **Display** the database objects in **descending order of similarities**

3D Object Retrieval: *Example*

QUERY

3D Object Retrieval: *Example*



3D Object Retrieval: *Objectives*

What do we aim at?

More similar (or **relevant**) objects $\rightarrow$ Beginning of the list

3D Object Retrieval: *Objectives*

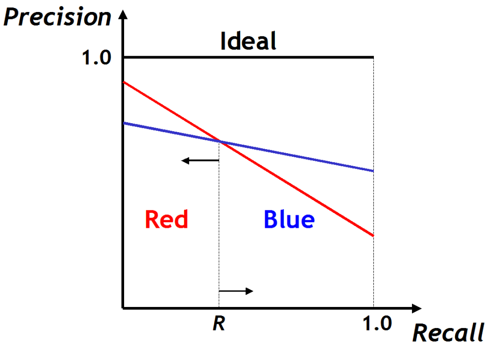
What do we aim at?

More similar (or **relevant**) objects $\rightarrow$ Beginning of the list

How do we measure the performance?

- Precision-Recall curve
- Nearest-Neighbor Score
- Discounted Cumulative Gain

3D Object Retrieval: *Performance Measures*



Definition: Precision and Recall

$$Precision = \frac{\#(\text{Relevant items in the first } K \text{ matches})}{K}$$

$$Recall = \frac{\#(\text{Relevant items in the first } K \text{ matches})}{\text{Size of the query class}}$$

3D Object Retrieval: *Performance Measures*

Nearest-Neighbor Score (NN)

- Percentage of the first correct matches
- Indicative of the classification performance
- $\mathrm{NN} \in [0, 100]$ (%)

Discounted Cumulative Gain (DCG)

- Considers the full list of retrieval results
- A user is less likely to consider elements near the end of the list.

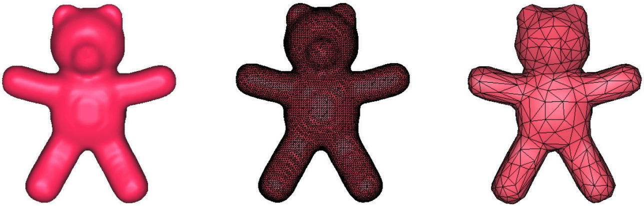
$\rightarrow$ Correct results near the front are weighted more.

- $\mathrm{DCG} \in [0, 100]$ (%)

3D Object Databases

Definition: Triangular Mesh

A **triangular mesh** is a union of triangles which approximates a continuous surface in 3D.



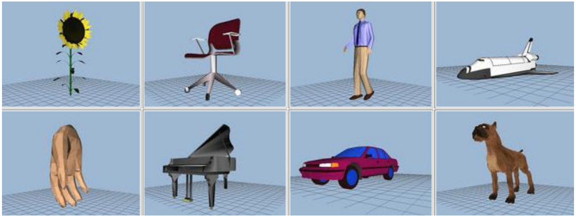
3D Object Databases

1. Princeton Shape Benchmark (PSB)
2. Sculpteur (SCU)
3. SHREC Watertight (SHREC-W)
4. Purdue Engineering Shape Benchmark

3D Object Databases: *PSB*

Princeton Shape Benchmark (PSB)

- Wide range of shape classes:
  flower, chair, human, spacecraft, piano, dog, ...
- **1814** objects in **161** classes:
  Training (907 objects, 90 classes) and Test (907 objects, 92 classes)



3D Object Databases: *SCU*

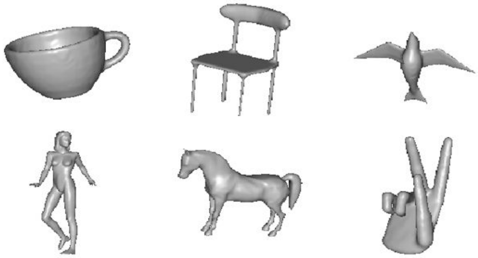
Sculpteur (SCU)

- Archaelogically valuable museum objects: **amphora, vase, pavement, statue, relievo, ...**
- **513** objects in **53** classes

3D Object Databases: *SHREC-W*

SHREC'07 Watertight (SHREC-W)

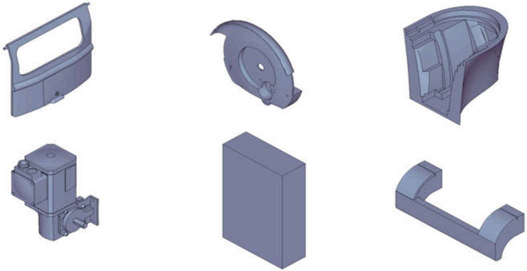
- A limited range of shape classes:
  cup, chair, bird, human, hand, spiral, ...
- Classification induced by topological equivalences
- **400** objects in **20** classes



3D Object Databases: *ESB*

Purdue Engineering Shape Benchmark (ESB)

- Engineering parts (CAD):
  bearing, gear, handle, elbow, housing, ...
- **815** objects in **45** classes



3D Object Databases: *Comparison*

Properties

	PSB	SCU	SHREC-W	ESB
#(Classes)	92	53	20	45
Resolution	Low	High	High	Medium
Watertight?	No	Yes	Yes	Yes
Smooth?	No	Yes	Yes	No

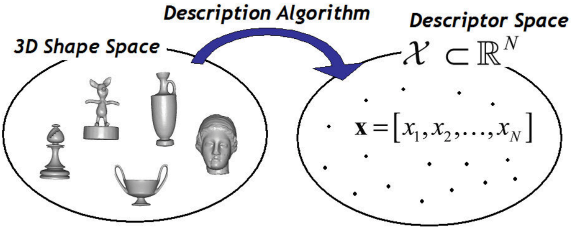
Remark

- Different 3D application domains
- Diverse semantics and shape properties

3D Shape Descriptors

Definition: Shape Descriptor

A **shape descriptor** is a **vector** or **graph-like** data structure, which encodes **geometrical** and/or **topological** shape characteristics.



3D Shape Descriptors: *Properties*

A “good” descriptor is

1. Effective
2. Efficient
3. Flexible
4. Robust
5. Invariant

3D Shape Descriptors: *Properties*

A "good" descriptor is

1. **Effective**
2. Efficient
3. Flexible
4. Robust
5. Invariant

- Captures essential shape characteristics
- Eliminates irrelevant details

→ **High retrieval performance**

3D Shape Descriptors: *Properties*

A "good" descriptor is

1. Effective
2. **Efficient**
3. Flexible
4. Robust
5. Invariant

- Fast to compute
- Low storage cost

3D Shape Descriptors: *Properties*

A "good" descriptor is

1. Effective
2. Efficient
3. **Flexible**
4. Robust
5. Invariant

Can be computed for different shape representations:

- Triangular mesh
- 3D point cloud
- Parametric surface
- Implicit surface
- Voxel ...

3D Shape Descriptors: *Properties*

A "good" descriptor is

1. Effective
2. Efficient
3. Flexible
4. **Robust**
5. Invariant

Insensitive to:

- Mesh resolution
- Mesh degeneracies
- Small shape variation
- Noise

3D Shape Descriptors: *Properties*

A “good” descriptor is

1. Effective
2. Efficient
3. Flexible
4. Robust
5. **Invariant**

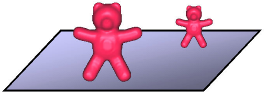
Shape is what remains after the effect of **rigid motions+scaling** are removed.

3D Shape Descriptors: *Properties*

A "good" descriptor is

1. Effective
2. Efficient
3. Flexible
4. Robust
5. **Invariant**

Translation



3D Shape Descriptors: *Properties*

A “good” descriptor is

1. Effective
2. Efficient
3. Flexible
4. Robust
5. **Invariant**

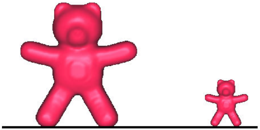
Pose Change
(Rotation and Reflection)

3D Shape Descriptors: *Properties*

A "good" descriptor is

1. Effective
2. Efficient
3. Flexible
4. Robust
5. **Invariant**

Isotropic Rescaling



3D Shape Descriptors: *Properties*

A “good” descriptor is

1. Effective
2. Efficient
3. Flexible
4. Robust
5. **Invariant**

BUT
without loss of shape information!

3D Shape Descriptors: *A Taxonomy*

1. Histogram-Based
2. Transform-Based
3. Graph-Based
4. "2D Image"-Based
5. Others

3D Shape Descriptors: *A Taxonomy*

1. **Histogram-Based**
2. Transform-Based
3. Graph-Based
4. "2D Image"-Based
5. Others

- Cord and Angle Histograms
- Shape Distributions
- Extended Gaussian Images
- 3D Hough Transform
- Shape Spectrum

3D Shape Descriptors: *A Taxonomy*

1. Histogram-Based
2. **Transform-Based**
3. Graph-Based
4. "2D Image"-Based
5. Others

Functions on the 3D grid

- 3D Fourier Transform
- Radial Cosine Transform

Functions on the unit sphere

- Angular Radial Transform
- Spherical Harmonics
- Spherical Wavelets

3D Shape Descriptors: *A Taxonomy*

1. Histogram-Based
2. Transform-Based
3. **Graph-Based**
4. "2D Image"-Based
5. Others

- Multiresolution Reeb Graphs
- Skeletal Graphs

3D Shape Descriptors: *A Taxonomy*

1. Histogram-Based
2. Transform-Based
3. Graph-Based
4. **"2D Image"-Based**
5. Others

Multiple views or projections

- Silhouette Descriptor
- Depth Buffer Images
- Lightfield Descriptor

3D Shape Descriptors: *A Taxonomy*

1. Histogram-Based
2. Transform-Based
3. Graph-Based
4. "2D Image"-Based
5. **Others**

- Spin Images
- 3D Zernike Moments
- Reflective Symmetry Descriptors

3D Shape Descriptors: *A Taxonomy*

1. **Histogram-Based**
2. Transform-Based
3. Graph-Based
4. "2D Image"-Based
5. Others

- Cord and Angle Histograms
- Shape Distributions
- Extended Gaussian Images
- 3D Hough Transform
- Shape Spectrum

Outline

1. 3D Object Retrieval
2. Density-Based Shape Description
   - Feature Design
   - Target Selection
   - Descriptor Computation
   - Additional Tools
   - Comparison Experiments
3. Statistical Similarity Learning
   - Score Fusion by Ranking Risk Minimization
   - Retrieval Protocols
   - Score Fusion Experiments
4. Conclusion and Perspectives

Density-Based Framework (DBF)

Definition: Density-Based Shape Descriptor

A **density-based descriptor** of a 3D shape is the **probability density function (pdf)** of a **local surface feature**.

Density-Based Framework (DBF)

Definition: Density-Based Shape Descriptor

A **density-based descriptor** of a 3D shape is
the **probability density function (pdf)** of a **local surface feature**.

The Premise

Similar shapes induce *similar* feature distributions.
$\Diamond$ Similarity between two shapes $\leftrightarrow$ Variation between feature pdfs

Density-Based Framework (DBF)

Earlier Approaches

Shape Distributions [Osada et al., 2002]
Cord and Angle Histograms [Paquet and Rioux, 1997]
Extended Gaussian Images [Horn, 1984; Kang and Ikeuchi, 1993]
3D Hough Transform [Zaharia and Prêteux, 2002]

Density-Based Framework (DBF)

Earlier Approaches

Shape Distributions [Osada et al., 2002]
Cord and Angle Histograms [Paquet and Rioux, 1997]
Extended Gaussian Images [Horn, 1984; Kang and Ikeuchi, 1993]
3D Hough Transform [Zaharia and Prêteux, 2002]

Key Aspects of DBF

- Scalar vs. **Multivariate** features
  $\rightarrow$ Exhaustive local characterization

Density-Based Framework (DBF)

Earlier Approaches

Shape Distributions [Osada et al., 2002]
Cord and Angle Histograms [Paquet and Rioux, 1997]
Extended Gaussian Images [Horn, 1984; Kang and Ikeuchi, 1993]
3D Hough Transform [Zaharia and Prêteux, 2002]

Key Aspects of DBF

- Scalar vs. **Multivariate** features
  $\rightarrow$ Exhaustive local characterization
- Histogram vs. **Kernel Density Estimation (KDE)**
  - More flexible
  - Smoother estimates
  - Fast thanks to Fast Gauss Transform (FGT)

Density-Based Framework (DBF)

Earlier Approaches

Shape Distributions [Osada et al., 2002]
Cord and Angle Histograms [Paquet and Rioux, 1997]
Extended Gaussian Images [Horn, 1984; Kang and Ikeuchi, 1993]
3D Hough Transform [Zaharia and Prêteux, 2002]

Key Aspects of DBF

- Scalar vs. **Multivariate** features
  $\rightarrow$ Exhaustive local characterization
- Histogram vs. **Kernel Density Estimation (KDE)**
  - More flexible
  - Smoother estimates
  - Fast thanks to Fast Gauss Transform (FGT)
- A unifying framework $\rightarrow$ **Family of descriptors**

DBF: A Three-Stage Process

Given a 3D objet O, represented by a mesh $\mathcal{M}$

DBF: A Three-Stage Process

Given a 3D objet O, represented by a mesh $\mathcal{M}$

1. **Feature Design**
   - Choose a **local surface feature** $S \in \mathcal{R}_\mathcal{S}$
   - Obtain a set of **feature observations** $\{s_k\}_{k=1}^K$ using the mesh points

DBF: A Three-Stage Process

Given a 3D objet O, represented by a mesh $\mathcal{M}$

1. **Feature Design**
    - Choose a **local surface feature** $S \in \mathcal{R}_{\mathcal{S}}$
    - Obtain a set of **feature observations** $\{s_k\}_{k=1}^{K}$ using the mesh points
2. **Target Selection**
    - Determine the **pdf evaluation points** for the feature S $\rightarrow$ **Targets:** $\overline{\mathcal{R}}_{\mathcal{S}} = \{t_n \in \mathcal{R}_{\mathcal{S}}\}_{n=1}^{N}$

DBF: A Three-Stage Process

Given a 3D objet O, represented by a mesh $\mathcal{M}$

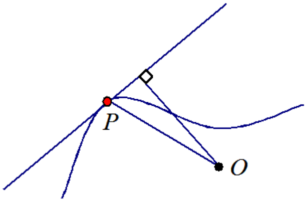
1. **Feature Design**
   - Choose a **local surface feature** $S \in \mathcal{R}_{\mathcal{S}}$
   - Obtain a set of **feature observations** $\{s_k\}_{k=1}^{K}$ using the mesh points
2. **Target Selection**
   - Determine the **pdf evaluation points** for the feature S $\rightarrow$ **Targets:** $\overline{\mathcal{R}}_{\mathcal{S}} = \{t_n \in \mathcal{R}_{\mathcal{S}}\}_{n=1}^{N}$
3. **Computation**
   - Using the observations, **estimate the feature pdf** at designated targets

$$\mathbf{f}_{S|O} = [f_S(t_1|O), \ldots, f_S(t_N|O)] \in \mathbb{R}^N$$

Outline

DBF(1): *Local Surface Features*

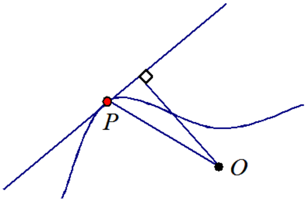
Once upon a time there was a point lying on a 3D surface...



DBF(1): *Local Surface Features*

Once upon a time there was a point lying on a 3D surface...

Zero-order
First-order
Second-order



DBF(1): *Local Surface Features*

Radial distance Radial direction Normal direction T-plane distance Shape index Alignment Torque

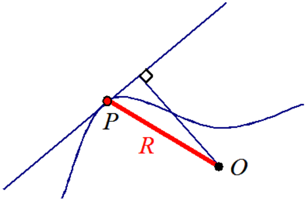
Radial Distance

$R \in [0, r_{max}]$

Scalar

Rotation-invariant

Zero-order



DBF(1): *Local Surface Features*

Radial distance *Radial direction* *Normal direction* *T-plane distance* *Shape index* *Alignment* *Torque*

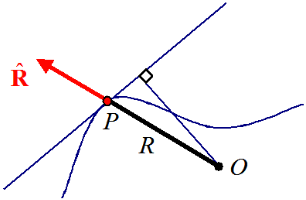
Radial Direction

$\hat{\mathbf{R}} \triangleq (\hat{R}_x, \hat{R}_y, \hat{R}_z) \in \mathcal{S}^2$

Unit-norm vector

Scale-invariant

Zero-order



DBF(1): *Local Surface Features*

Radial distance Radial direction Normal direction T-plane distance Shape index Alignment Torque

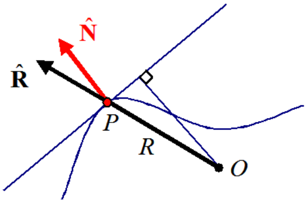
Normal Direction

$\hat{\mathbf{N}} \triangleq (\hat{N}_x, \hat{N}_y, \hat{N}_z) \in \mathcal{S}^2$

Unit-norm vector

Scale-invariant

First-order



DBF(1): *Local Surface Features*

Radial distance Radial direction Normal direction T-plane distance Shape index Alignment Torque

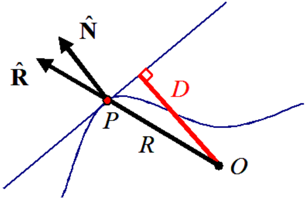
T-plane Distance

$$D = R \left| \left\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \right\rangle \right| \in [0, d_{max}]$$

Scalar

Rotation-invariant

First-order



DBF(1): *Local Surface Features*

Radial distance Radial direction Normal direction T-plane distance Shape index Alignment Torque

Shape Index

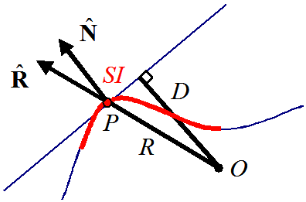
$$SI = \frac{1}{2} - \left(\frac{2}{\pi}\right) arctan\left(\frac{\kappa_1+\kappa_2}{\kappa_1-\kappa_2}\right)$$

$SI \in [0, 1]$

Scalar

Rotation and Scale-invariant

Second-order



DBF(1): *Local Surface Features*

Radial distance Radial direction Normal direction T-plane distance Shape index Alignment Torque

Shape Index

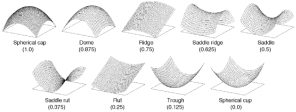
$$SI = \frac{1}{2} - \left(\frac{2}{\pi}\right) arctan\left(\frac{\kappa_1+\kappa_2}{\kappa_1-\kappa_2}\right)$$

$$SI \in [0,1]$$

Scalar

Rotation and Scale-invariant

Second-order



DBF(1): *Local Surface Features*

Radial distance Radial direction Normal direction T-plane distance Shape index Alignment Torque

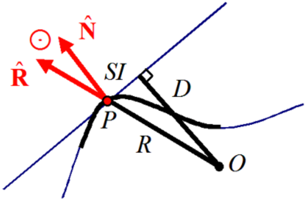
Alignment

$$A \triangleq \left| \left\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \right\rangle \right| \in [0, 1]$$

Scalar

Rotation and Scale-invariant

First-order



DBF(1): *Local Surface Features*

Radial distance Radial direction Normal direction T-plane distance Shape index Alignment Torque

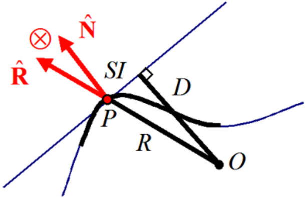
Torque

$\mathbf{C} \triangleq \hat{\mathbf{R}} \times \hat{\mathbf{N}} \in \mathcal{B}^2$

Vector

Scale-invariant

First-order



DBF(1): *Local Characterization of a 3D Surface*

Remark

Join the features → Obtain a **multivariate** local characterization

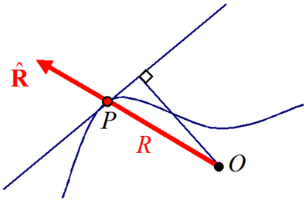
DBF(1): *Local Characterization of a 3D Surface*

Radial

$(R, \hat{\mathbf{R}})$

parametrizes the surface point

Zero-order



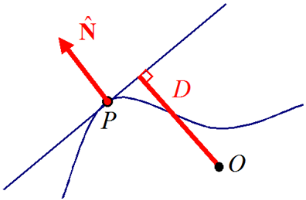
DBF(1): *Local Characterization of a 3D Surface*

T-plane

$(D, \hat{\mathbf{N}})$

parametrizes the tangent plane

First-order



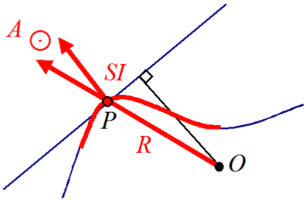
DBF(1): *Local Characterization of a 3D Surface*

"Sec-Order"

(R, A, SI)

categorical surface information

Second-order

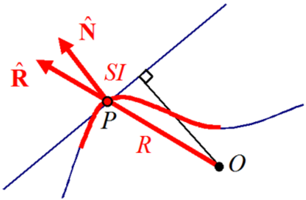


DBF(1): *Local Characterization of a 3D Surface*

Full

$(R, \hat{\mathbf{R}}, \hat{\mathbf{N}}, SI)$

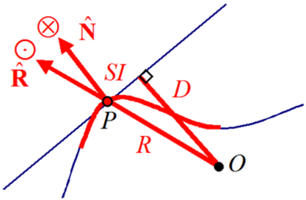
characterization up to second-order



DBF(1): *Local Characterization of a 3D Surface*

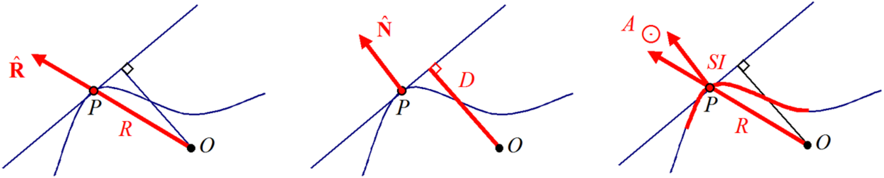
Even "More" Multivariate

$(R, \hat{\mathbf{R}}, D, \hat{\mathbf{N}}, SI, A, \mathbf{C})$



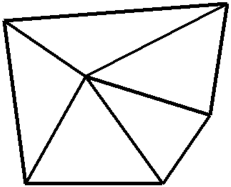
DBF(1): *Local Characterization of a 3D Surface*

Radial $(R, \hat{\mathbf{R}})$ **T-plane** $(D, \hat{\mathbf{N}})$ **Sec-Order** (R, A, SI)



DBF(1): *Feature Calculation*

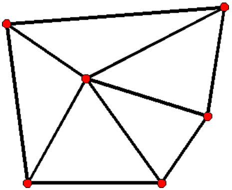
Features are calculated at mesh points.
Which ones?



DBF(1): *Feature Calculation*

Features are calculated at mesh points.
Which ones?

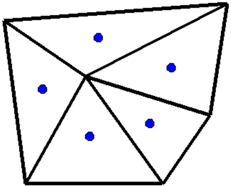
- **Vertices**



DBF(1): *Feature Calculation*

Features are calculated at mesh points.
Which ones?

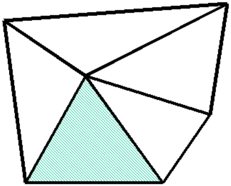
- Vertices
- **Triangle centers**



DBF(1): *Feature Calculation*

Features are calculated at mesh points.
Which ones?

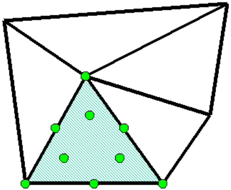
- Vertices
- Triangle centers
- **By averaging over the triangle**



DBF(1): *Feature Calculation*

Features are calculated at mesh points.
Which ones?

- Vertices
- Triangle centers
- **By averaging over the triangle**



Points given by **Simpson's approximation** to the averaging integral

DBF(1): *Effect of Feature Calculation*

DCG for the Radial $(R, \hat{\mathbf{R}})$-Descriptor

	Databases	
Feature Calculation	*PSB Training*	*SCU*
Vertex	56.0	71.3
Centroid	55.6	71.2
Simpson	**57.0**	71.3

DBF(1): *Effect of Feature Calculation*

DCG for the Radial $(R, \hat{\mathbf{R}})$-Descriptor

Feature Calculation	**Databases** *PSB Training*	*SCU*
Vertex	56.0	71.3
Centroid	55.6	71.2
Simpson	**57.0**	71.3

Facts

- **Low mesh resolution** (PSB)
  $\longrightarrow$ Simpson averaging has a positive effect
- **High mesh resolution** (SCU)
  $\longrightarrow$ All schemes are performance-wise equivalent

Outline

DBF(2): *Target Selection*

Reminder

Targets $\leftrightarrow$ Pdf evaluation points

DBF(2): *Target Selection*

- **Scalar** $S \in I = [a, b] \in \mathbb{R}$

  Radial Distance, T-plane Distance, Alignment

  1. **Uniform** Sampling **(S1)**
  2. **Equal-probability** (non-uniform) Sampling **(S2)**

DBF(2): *Target Selection*

- **Scalar** $S \in I = [a, b] \in \mathbb{R}$

  Radial Distance, T-plane Distance, Alignment

  1. **Uniform** Sampling **(S1)**
  2. **Equal-probability** (non-uniform) Sampling **(S2)**

- **Unit-norm vector** $S \in \mathcal{S}^2$

  Radial Direction, Normal Direction

  1. Octahedron subdivision **(V1)**
  2. Sampling by spherical coordinates **(V2)**

DBF(2): *Target Selection*

- **Scalar** $S \in I = [a, b] \in \mathbb{R}$

  Radial Distance, T-plane Distance, Alignment

  1. **Uniform** Sampling **(S1)**
  2. **Equal-probability** (non-uniform) Sampling **(S2)**

- **Unit-norm vector** $S \in \mathcal{S}^2$

  Radial Direction, Normal Direction

  1. **Octahedron subdivision (V1)**
  2. Sampling by spherical coordinates **(V2)**

DBF(2): *Target Selection*

- **Scalar** $S \in I = [a, b] \in \mathbb{R}$

  Radial Distance, T-plane Distance, Alignment

  1. **Uniform** Sampling **(S1)**
  2. **Equal-probability** (non-uniform) Sampling **(S2)**

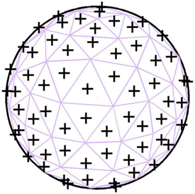
- **Unit-norm vector** $S \in \mathcal{S}^2$

  Radial Direction, Normal Direction

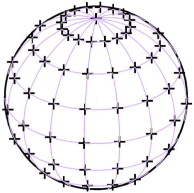
  1. Octahedron subdivision **(V1)**
  2. Sampling by **spherical coordinates (V2)**

DBF(2): *Target Selection*

Sampling the Unit-Sphere



Octahedron subdivision



Spherical coordinates

DBF(2): *Target Selection*

- **Scalar** $S \in I = [a, b] \in \mathbb{R}$
  Radial Distance, T-plane Distance, Alignment
  1. **Uniform** Sampling **(S1)**
  2. **Equal-probability** (non-uniform) Sampling **(S2)**
- **Unit-norm vector** $S \in \mathcal{S}^2$
  Radial Direction, Normal Direction
  1. Octahedron subdivision **(V1)**
  2. Sampling by spherical coordinates **(V2)**
- **General multivariate** $S = (S_1, S_2) \in \mathcal{R}_{S_1} \times \mathcal{R}_{S_2}$
  Take the Cartesian product of individual target sets

DBF(2): *Effect of Target Selection*

DCG Performance of Sampling Schemes

Sampling	Radial	T-plane
S1×**V1**	57.0	59.8
S1×**V2**	56.8	60.5
S2×**V1**	56.0	59.5
S2×**V2**	56.3	60.1

DBF(2): *Effect of Target Selection*

DCG Performance of Sampling Schemes

Sampling	Radial	T-plane
S1×**V1**	57.0	59.8
S1×**V2**	56.8	60.5
S2×**V1**	56.0	59.5
S2×**V2**	56.3	60.1

Fact

All target selection schemes lead to equivalent performances...
provided that the non-uniformity of targets is taken into account at similarity computation.

Outline

DBF(3): *Kernel Density Estimation (KDE)*

$$f_S(t_n) = \sum_{k=1}^{K} w_k \, |H_k|^{-1} \, \mathcal{K}\left(H_k^{-1}(t_n - s_k)\right)$$

DBF(3): *Kernel Density Estimation (KDE)*

$$f_S(t_n) = \sum_{k=1}^{K} w_k \, |H_k|^{-1} \, \mathcal{K}\left(H_k^{-1}(t_n - \boldsymbol{s_k})\right)$$

- **Sources** (or observations) $\{s_k\}_{k=1}^{K}$

DBF(3): *Kernel Density Estimation (KDE)*

$$f_S(t_n) = \sum_{k=1}^{K} w_k \left|H_k\right|^{-1} \mathcal{K}\left(H_k^{-1}(\boldsymbol{t_n} - s_k)\right)$$

- Sources (or observations) $\{s_k\}_{k=1}^{K}$
- **Targets** $\{t_n\}_{n=1}^{N}$

DBF(3): *Kernel Density Estimation (KDE)*

$$f_S(t_n) = \sum_{k=1}^{K} \boldsymbol{w_k} \left| H_k \right|^{-1} \mathcal{K}\left(H_k^{-1}(t_n - s_k) \right)$$

- Sources (or observations) $\{s_k\}_{k=1}^{K}$
- Targets $\{t_n\}_{n=1}^{N}$
- **Weights** $\{w_k\}_{k=1}^{K}$ *set to relative triangular areas* $\sum_k w_k = 1$

DBF(3): *Kernel Density Estimation (KDE)*

$$f_S(t_n) = \sum_{k=1}^{K} w_k \left|H_k\right|^{-1} \mathcal{K}\left(H_k^{-1}(t_n - s_k)\right)$$

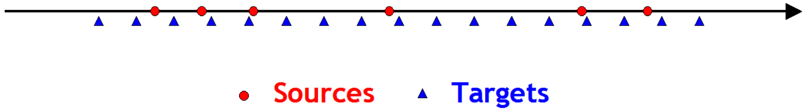
- Sources (or observations) $\{s_k\}_{k=1}^{K}$
- Targets $\{t_n\}_{n=1}^{N}$
- Weights $\{w_k\}_{k=1}^{K}$ *set to relative triangular areas* $\sum_k w_k = 1$
- **Kernel** $\mathcal{K}$ *set to Gaussian*

DBF(3): *Kernel Density Estimation (KDE)*

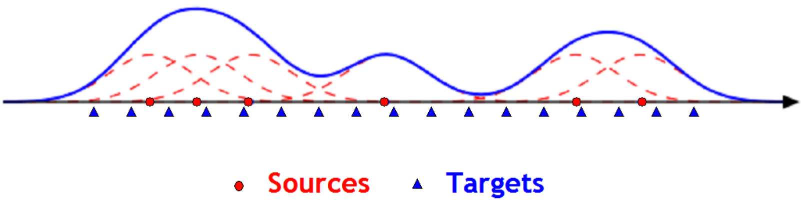
$$f_S(t_n) = \sum_{k=1}^{K} w_k \, |\boldsymbol{H_k}|^{-1} \, \mathcal{K}\left(\boldsymbol{H_k}^{-1}(t_n - s_k)\right)$$

- Sources (or observations) $\{s_k\}_{k=1}^{K}$
- Targets $\{t_n\}_{n=1}^{N}$
- Weights $\{w_k\}_{k=1}^{K}$ *set to relative triangular areas* $\sum_k w_k = 1$
- Kernel $\mathcal{K}$ *set to Gaussian*
- **Bandwidth parameter matrices** $\{H_k\}_{k=1}^{K}$

DBF(3): *Illustration of KDE*

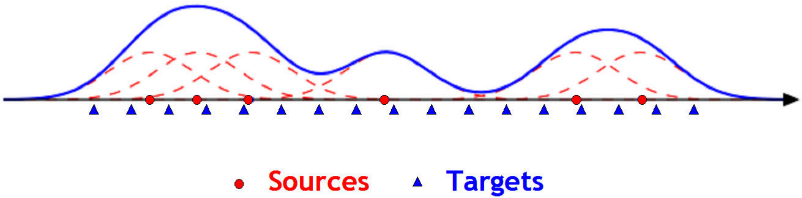


DBF(3): *Illustration of KDE*



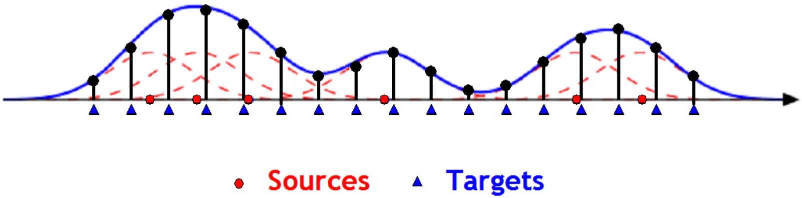
KDE places a kernel around each source...

DBF(3): *Illustration of KDE*



... and interpolates the pdf in the feature space.

DBF(3): *Illustration of KDE*



Pdf values at targets become the descriptor vector.

DBF(3): *Computational Complexity of KDE*

$$f_S(t_n) = \sum_{k=1}^{K} w_k \left|H_k\right|^{-1} \mathcal{K}\left(H_k^{-1}(t_n - s_k)\right)$$

- Direct Evaluation $\rightarrow O(KN)$

DBF(3): *Computational Complexity of KDE*

$$f_S(t_n) = \sum_{k=1}^{K} w_k \left|H_k\right|^{-1} \mathcal{K}\left(H_k^{-1}(t_n - s_k)\right)$$

- Direct Evaluation $\rightarrow O(KN)$
- When **the kernel** $\mathcal{K}$ is **Gaussian** $\rightarrow O(K+N)$
  Fast Gauss Transform (FGT)
  [Greengard and Strain, 1991; Yang et al., 2003]

Example: $K = 130000$ and $N = 1024$

- **Direct** $\rightarrow$ 125 secs
- **FGT** $\rightarrow$ 2.5 secs

DBF(3): *Effect of the Bandwidth in KDE*

DBF(3): *Effect of the Bandwidth in KDE*

Remark

The bandwidth parameter controls the smoothing behavior of KDE:

Larger bandwidth $\rightarrow$ **Smoother** estimate

DBF(3): *Effect of the Bandwidth in KDE*

Remark

The bandwidth parameter controls the smoothing behavior of KDE:

Larger bandwidth $\rightarrow$ **Smoother** estimate

- **Large** bandwidth $\rightarrow$ **Small** descriptor variation
  - Good when we want to compare similar shapes
  - Descriptors might fail to be sufficiently discriminative

DBF(3): *Effect of the Bandwidth in KDE*

Remark

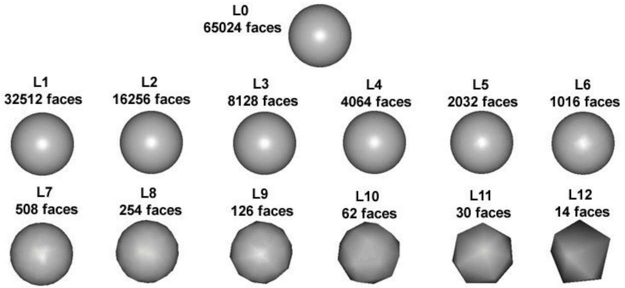
The bandwidth parameter controls the smoothing behavior of KDE:

Larger bandwidth $\longrightarrow$ **Smoother** estimate

- **Large** bandwidth $\longrightarrow$ **Small** descriptor variation
  - Good when we want to compare similar shapes
  - Descriptors might fail to be sufficiently discriminative
- **Small** bandwidth $\longrightarrow$ **Large** descriptor variation
  - Good when we want to discriminate different shapes
  - Pdf estimate might overfit the observations

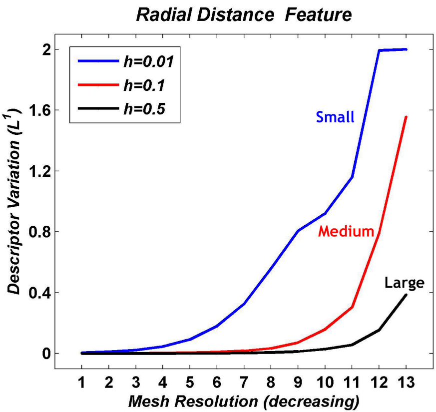
DBF(3): *Effect of the Bandwidth in KDE*

Robustness against Low Mesh Resolution



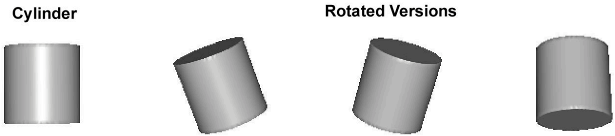
DBF(3): *Effect of the Bandwidth in KDE*

Robustness against Low Mesh Resolution



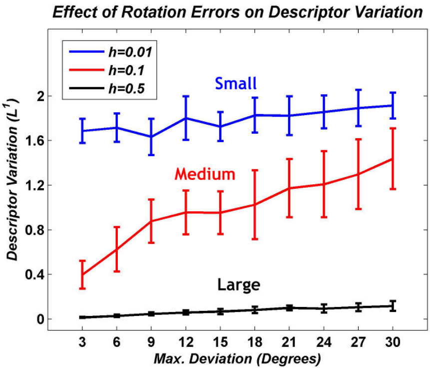
DBF(3): *Effect of the Bandwidth in KDE*

Robustness against Pose Deviations



DBF(3): *Effect of the Bandwidth in KDE*

Robustness against Pose Deviations



DBF(3): *Effect of the Bandwidth in KDE*

Fact

Variation can be rendered negligible by increasing the bandwidth.

DBF(3): *Effect of the Bandwidth in KDE*

Fact

Variation can be rendered negligible by increasing the bandwidth.

Question

Should we increase the bandwidth indefinitely?

DBF(3): *Effect of the Bandwidth in KDE*

Fact

Variation can be rendered negligible by increasing the bandwidth.

Question

Should we increase the bandwidth indefinitely?

Remark

In retrieval, we want:

- **Small** variation between **similar** shapes,
- **Large** variation between **different** shapes!

DBF(3): *Effect of the Bandwidth in KDE*

Fact

Variation can be rendered negligible by increasing the bandwidth.

Question

Should we increase the bandwidth indefinitely?

Remark

In retrieval, we want:

- **Small** variation between **similar** shapes,
- **Large** variation between **different** shapes!

Answer

Setting the bandwidth is a matter of compromise between descriptor **smoothness** vs. **discriminativeness**.

DBF(3): *Bandwidth Selection in KDE*

$$f_S(t) = \sum_{k=1}^{K} w_k \left|\boldsymbol{H_k}\right|^{-1} \mathcal{K}\left(\boldsymbol{H_k}^{-1}(t - s_k)\right)$$

Three Options

1. Triangle-level
2. Mesh-level
3. Database-level

DBF(3): *Bandwidth Selection in KDE*

$$f_S(t) = \sum_{k=1}^{K} w_k \, |\boldsymbol{H_k}|^{-1} \, \mathcal{K}\left(\boldsymbol{H_k}^{-1}(t - s_k)\right)$$

Three Options

1. **Triangle-level**
   - H_k different for each triangle on each mesh
   - $H_k \propto$ feature covariance **over the triangle**
2. Mesh-level
3. Database-level

DBF(3): *Bandwidth Selection in KDE*

$$f_S(t) = \sum_{k=1}^{K} w_k \left|\boldsymbol{H_k}\right|^{-1} \mathcal{K}\left(\boldsymbol{H_k}^{-1}(t - s_k)\right)$$

Three Options

1. Triangle-level
   - H_k different for each triangle on each mesh
   - $H_k \propto$ feature covariance over the triangle
2. **Mesh-level**
   - $H_k = H$ for a given mesh but differs from mesh to mesh
   - $H_k \propto$ feature covariance **over the mesh**
3. Database-level

DBF(3): *Bandwidth Selection in KDE*

$$f_S(t) = \sum_{k=1}^{K} w_k \left|\boldsymbol{H_k}\right|^{-1} \mathcal{K}\left(\boldsymbol{H_k}^{-1}(t - s_k)\right)$$

Three Options

1. Triangle-level
   - H_k different for each triangle on each mesh
   - $H_k \propto$ feature covariance over the triangle
2. Mesh-level
   - $H_k = H$ for a given mesh but differs from mesh to mesh
   - $H_k \propto$ feature covariance over the mesh
3. **Database-level**
   - $H_k = H$ for all meshes in the database
   - $H_k \propto$ average feature covariance **over the database**

DBF(3): *Bandwidth Selection in KDE*

DCG (%) for Possible Bandwidth Selection Strategies on PSB Training Set

	Descriptor		
Bandwidth Setting	*Radial*	*T-plane*	*Torque*
Triangle-level	35.2	-	-
Mesh-level	51.1	51.4	49.9
Database-level	**57.0**	**59.8**	**55.6**

DBF(3): *Bandwidth Selection in KDE*

DCG (%) for Possible Bandwidth Selection Strategies on PSB Training Set

	Descriptor		
Bandwidth Setting	*Radial*	*T-plane*	*Torque*
Triangle-level	35.2	-	-
Mesh-level	51.1	51.4	49.9
Database-level	**57.0**	**59.8**	**55.6**

Fact

Set the bandwidth at **database-level**

DBF: *Descriptor Properties*

Remarks

1. Effectiveness
2. Efficiency
   - Computational
   - Storage-wise
3. Flexibility
4. Robustness
5. Invariance

DBF: *Descriptor Properties*

Remarks

1. Effectiveness
2. **Efficiency**
   - **Computational** → **FGT**
   - Storage-wise
3. **Flexibility** → **Simple features + KDE**
4. **Robustness** → **Bandwidth in KDE**
5. Invariance

Outline

1. 3D Object Retrieval
2. Density-Based Shape Description
   - Feature Design
   - Target Selection
   - Descriptor Computation
   - Additional Tools
   - Comparison Experiments
3. Statistical Similarity Learning
   - Score Fusion by Ranking Risk Minimization
   - Retrieval Protocols
   - Score Fusion Experiments
4. Conclusion and Perspectives

DBF: *Additional Tools*

Exploiting the pdf structure

1. Marginalization
2. Probability Density Pruning
3. Invariance at Matching

DBF: *(1) Marginalization*

Marginalization *removes any local shape information brought by a certain feature component.*

$$
\begin{aligned}
f_{S_{\cancel{k}}|O_t} &\triangleq f\left(s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_m \middle| O_t\right) \\
&= \int_{S_k} f\left(s_1, \ldots, s_k, \ldots, s_m \middle| O_t\right) ds_k .
\end{aligned}
$$

DBF: *(1) Marginalization*

Marginalization *removes any local shape information brought by a certain feature component.*

$$f_{S_{\not k}|O_t} \triangleq f\left(s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_m | O_t\right) = \int_{S_k} f\left(s_1, \ldots, s_k, \ldots, s_m | O_t\right) ds_k .$$

Its Uses

- Feature space exploration
- Smaller feature space $\Rightarrow$ Reduced descriptor size

example: $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z) \xrightarrow{Marginalization} (R, \hat{R}_x, \hat{R}_y)$

note $\hat{R}_x^2 + \hat{R}_y^2 + \hat{R}_z^2 = 1$

DBF: *(1) Marginalization*

Feature Space Exploration by Marginalization on PSB Training

Feature	Removed	DCG	Size
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$	-	62.1	10240

DBF: *(1) Marginalization*

Feature Space Exploration by Marginalization on PSB Training

Feature	Removed	DCG	Size
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$	-	62.1	10240
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, A)$	$\hat{N}_y$	62.6	5120

DBF: *(1) Marginalization*

Feature Space Exploration by Marginalization on PSB Training

Feature	Removed	DCG	Size
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$	-	62.1	10240
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, A)$	$\hat{N}_y$	62.6	5120
$(R, \hat{R}_x, \hat{N}_x, A)$	$\hat{R}_y$	63.4	2560

DBF: *(1) Marginalization*

Feature Space Exploration by Marginalization on PSB Training

Feature	Removed	DCG	Size
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$	-	62.1	10240
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, A)$	$\hat{N}_y$	62.6	5120
$(R, \hat{R}_x, \hat{N}_x, A)$	$\hat{R}_y$	63.4	2560
$(\hat{R}_x, \hat{N}_x, A)$	R	61.5	320

DBF: *(1) Marginalization*

Feature Space Exploration by Marginalization on PSB Training

Feature	Removed	DCG	Size
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$	-	62.1	10240
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, A)$	$\hat{N}_y$	62.6	5120
$(R, \hat{R}_x, \hat{N}_x, A)$	$\hat{R}_y$	63.4	2560
$(\hat{R}_x, \hat{N}_x, A)$	R	61.5	320
$(\hat{R}_x, \hat{N}_x)$	A	58.1	64

DBF: *(1) Marginalization*

Feature Space Exploration by Marginalization on PSB Training

Feature	Removed	DCG	Size
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$	-	62.1	10240
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, A)$	$\hat{N}_y$	62.6	5120
$(R, \hat{R}_x, \hat{N}_x, A)$	$\hat{R}_y$	63.4	2560
$(\hat{R}_x, \hat{N}_x, A)$	R	61.5	320
$(\hat{R}_x, \hat{N}_x)$	A	58.1	64
$\hat{N}_x$	$\hat{R}_x$	44.8	8

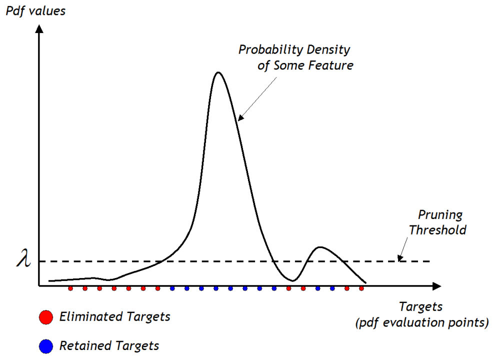
DBF: *(1) Marginalization*

Feature Space Exploration by Marginalization on PSB Training

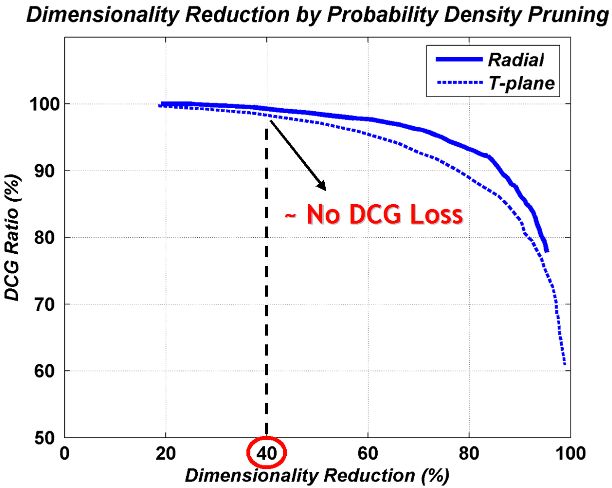
Feature	Removed	DCG	Size
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$	-	62.1	10240
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, A)$	$\hat{N}_y$	62.6	5120
$(R, \hat{R}_x, \hat{N}_x, A)$	$\hat{R}_y$	**63.4**	**2560**
$(\hat{R}_x, \hat{N}_x, A)$	R	61.5	320
$(\hat{R}_x, \hat{N}_x)$	A	58.1	64
$\hat{N}_x$	$\hat{R}_x$	44.8	8

DBF: *(2) Probability Density Pruning*

Probability Density Pruning *identifies and eliminates the targets where the pdf values are negligible.*



DBF: *(2) Probability Density Pruning*



DBF: *(3) Invariance at Matching*

Invariance: A Major Problem in Shape Matching

1. Invariance by design
2. Invariance by normalization
3. Invariance at Matching

DBF: *(3) Invariance at Matching*

Invariance: A Major Problem in Shape Matching

1. **Invariance by design**
   - Feature or descriptor is invariant by definition
   - Usually at the cost of shape information
2. Invariance by normalization
3. Invariance at Matching

DBF: *(3) Invariance at Matching*

Invariance: A Major Problem in Shape Matching

1. Invariance by design
   - Feature or descriptor is invariant by definition
   - Usually at the cost of shape information
2. **Invariance by normalization**
   - Normalize the object pose prior to descriptor computation
   - Normalization methods might fail
3. Invariance at Matching

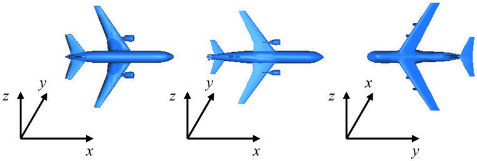
DBF: *(3) Invariance at Matching*

Invariance: A Major Problem in Shape Matching

1. Invariance by design
   - Feature or descriptor is invariant by definition
   - Usually at the cost of shape information
2. Invariance by normalization
   - Normalize the object pose prior to descriptor computation
   - Normalization methods might fail
3. **Invariance at Matching**
   - Evaluate the similarity under all possible transformations and pick the maximum
   - Costly if descriptor should be computed for every possible transformation

DBF: *(3) Invariance at Matching*

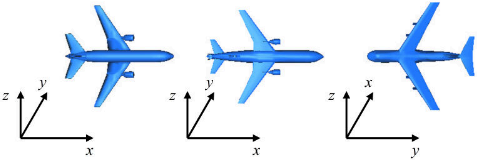
Axis Relabelings and Mirror Reflections



$$3! = 6 \text{ axis relabelings}$$
$$2^3 = 8 \text{ polarity assignments}$$
$$\Rightarrow 6 \times 8 = 48 \text{ axis configurations}$$

DBF: *(3) Invariance at Matching*

Axis Relabelings and Mirror Reflections



Given two descriptors $\mathbf{f}_1$ and $\mathbf{f}_2$
Axis changing transformations $\Gamma_i, i = 1, \ldots, 48$

1. Hold $\mathbf{f}_1$ fixed
2. For each Γ_i, calculate the similarity $sim_i(\mathbf{f}_1, \Gamma_i(\mathbf{f}_2))$
3. Pick the **maximum** sim_{i*} as the similarity between $\mathbf{f}_1$ and $\mathbf{f}_2$

DBF: *(3) Invariance at Matching*

In DBF:

No need to recompute the descriptor for every possible axis change

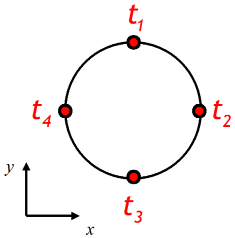
Just permute the vector entries!

DBF: *(3) Invariance at Matching*

In DBF:

No need to recompute the descriptor for every possible axis change

Just permute the vector entries!

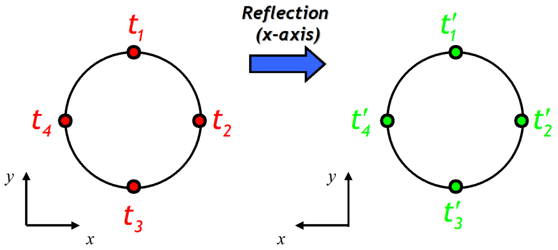


DBF: *(3) Invariance at Matching*

In DBF:

No need to recompute the descriptor for every possible axis change

Just permute the vector entries!

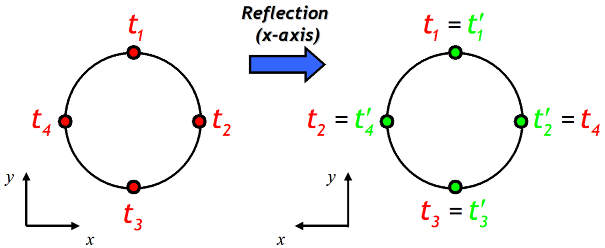


DBF: *(3) Invariance at Matching*

In DBF:

No need to recompute the descriptor for every possible axis change

Just permute the vector entries!



DBF: *(3) Invariance at Matching*

Additive DCG Gain of the Invariant Scheme

	PSB Train	PSB Test	SCU	SHREC-W	ESB
Radial	4.2	3.0	3.1	3.6	1.3
T-plane	5.1	3.6	4.2	2.3	1.7

DBF: *(3) Invariance at Matching*

Additive DCG Gain of the Invariant Scheme

	PSB Train	PSB Test	SCU	SHREC-W	ESB
Radial	4.2	3.0	3.1	3.6	1.3
T-plane	5.1	3.6	4.2	2.3	1.7

Fact

The invariant scheme improves the performance for all databases.

DBF: *Descriptor Properties*

Remarks

1. Effectiveness
2. **Efficiency**
   - **Computational** $\rightarrow$ FGT
   - Storage-wise
3. **Flexibility** $\rightarrow$ Simple features + KDE
4. **Robustness** $\rightarrow$ Bandwidth in KDE
5. Invariance

DBF: *Descriptor Properties*

Remarks

1. Effectiveness
2. **Efficiency**
   - **Computational** → FGT
   - **Storage-wise** → **Marginalization + Pruning**
3. **Flexibility** → Simple features + KDE
4. **Robustness** → Bandwidth in KDE
5. **Invariance** → **Axis relabelings and reflections**

Outline

DBF: *Comparison to Histogram-Based Peers*

Three Comparisons

1. **Cord and Angle Histograms (CAH)** [Paquet and Rioux, 1997]
2. **Extended Gaussian Images (EGI)** [Horn, 1984]
3. **3D Hough Transform (3DHT)** [Zaharia and Preteux, 2002]

DBF: *Comparison to Histogram-Based Peers (1)*

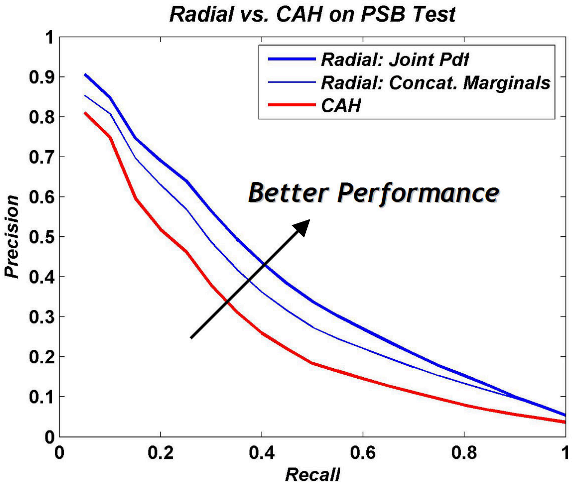
(1) Cord and Angle Histograms (CAH)

- Univariate histograms of radial distance and angles

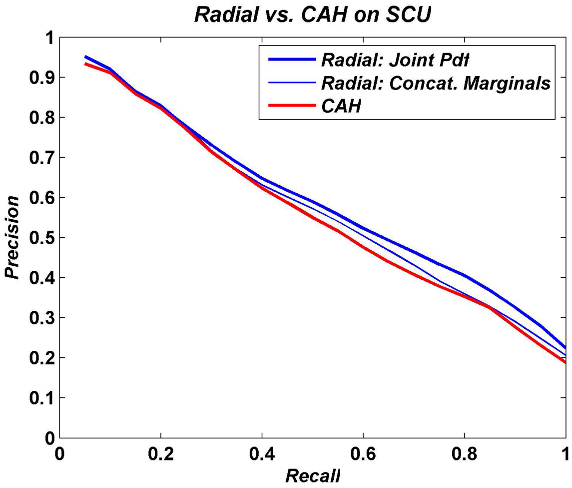
$\longrightarrow$ Univariate **pdfs** of $R, \hat{R}_x, \hat{R}_y, \hat{R}_z$ (Scalar KDE)

$\longrightarrow$ **Radial** $(R, \hat{\mathbf{R}})$**-Descriptor** (Multivariate KDE)

DBF: *Comparison to Histogram-Based Peers (1)*



DBF: *Comparison to Histogram-Based Peers (1)*



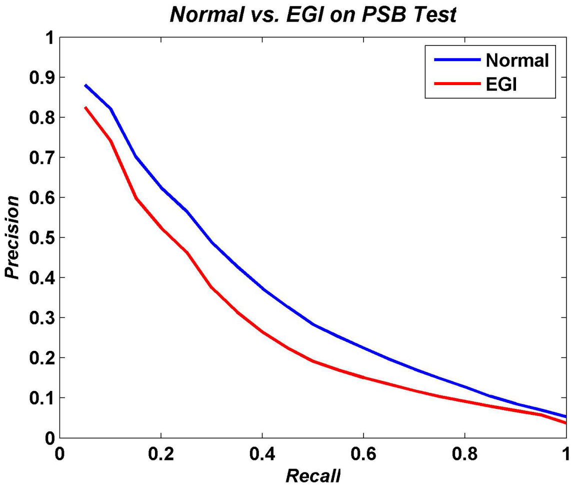
DBF: *Comparison to Histogram-Based Peers (2)*

(2) Extended Gaussian Images (EGI)

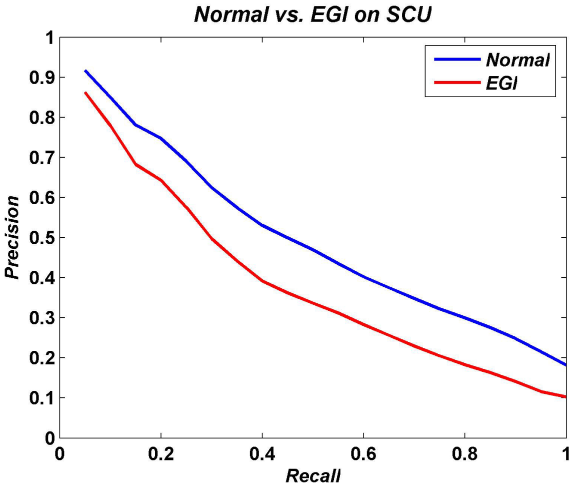
- Accumulator of the normal field

$\longrightarrow$ **Normal $\hat{\mathrm{N}}$-Descriptor**

DBF: *Comparison to Histogram-Based Peers (2)*



DBF: *Comparison to Histogram-Based Peers (2)*



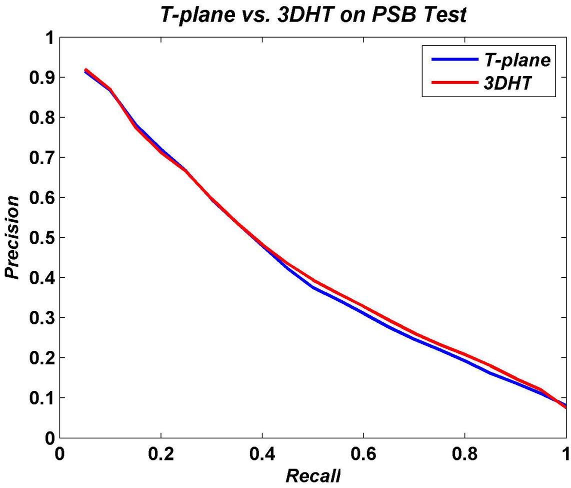
DBF: *Comparison to Histogram-Based Peers (3)*

(3) 3D Hough Transform (3DHT)

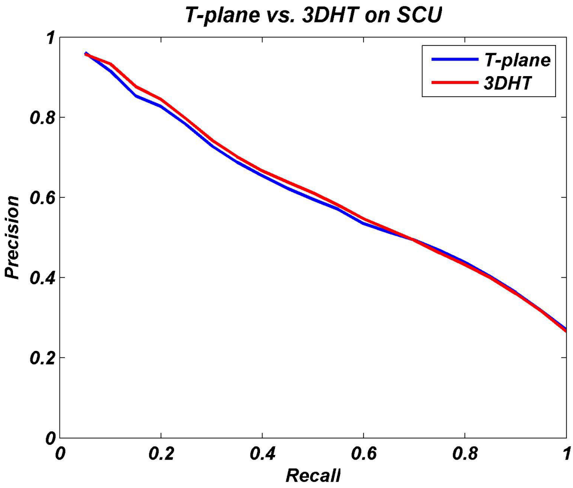
- Accumulator of tangent plane parameters

$\longrightarrow$ **T-plane $(D, \hat{\mathbf{N}})$-Descriptor**

DBF: *Comparison to Histogram-Based Peers (3)*



DBF: *Comparison to Histogram-Based Peers (3)*



DBF: *Comparison to Histogram-Based Peers*

Effectiveness Result I

Density-based descriptors perform better than or equally well as their histogram-based counterparts.

DBF: *Comparison to the State-of-the-Art*

Best Methods on PSB Test Set

DBF: *Comparison to the State-of-the-Art*

Best Methods on PSB Test Set

- Purely 3D:
  Radialized Extent Function (REXT) $\rightarrow$ **DCG** $=$ **60.1**
- Based on 2D:
  Depth Buffer Images (DBI) $\rightarrow$ **DCG** $=$ **66.3**

DBF: *Comparison to the State-of-the-Art*

Best Methods on PSB Test Set

- Purely 3D:
  Radialized Extent Function (REXT) $\rightarrow$ **DCG = 60.1**
- Based on 2D:
  Depth Buffer Images (DBI) $\rightarrow$ **DCG = 66.3**

DBF: A Few Instances

Descriptor	DCG

DBF: *Comparison to the State-of-the-Art*

Best Methods on PSB Test Set

- Purely 3D:
  Radialized Extent Function (REXT) $\rightarrow$ $\mathbf{DCG = 60.1}$
- Based on 2D:
  Depth Buffer Images (DBI) $\rightarrow$ $\mathbf{DCG = 66.3}$

DBF: A Few Instances

Descriptor	**DCG**
$(D, \hat{\mathbf{N}})$ with *Invariant-*L^1	61.4

DBF: *Comparison to the State-of-the-Art*

Best Methods on PSB Test Set

- Purely 3D:
  Radialized Extent Function (REXT) $\rightarrow$ **DCG = 60.1**
- Based on 2D:
  Depth Buffer Images (DBI) $\rightarrow$ **DCG = 66.3**

DBF: A Few Instances

Descriptor	DCG
$(D, \hat{\mathbf{N}})$ with *Invariant-*L^1	61.4
$(R, \hat{\mathbf{R}}) \oplus (D, \hat{\mathbf{N}}) \oplus (R, A, SI)$ with L^1	62.6

DBF: *Comparison to the State-of-the-Art*

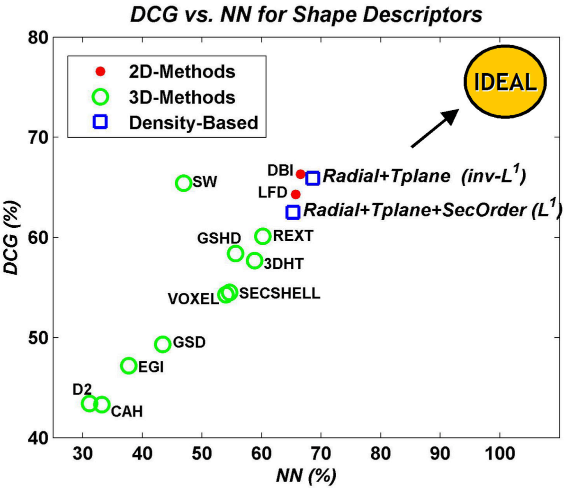
Best Methods on PSB Test Set

- Purely 3D:
  Radialized Extent Function (REXT) $\rightarrow$ **DCG** $=$ **60.1**
- Based on 2D:
  Depth Buffer Images (DBI) $\rightarrow$ **DCG** $=$ **66.3**

DBF: A Few Instances

Descriptor	DCG
$(D, \hat{\mathbf{N}})$ with *Invariant-*L^1	61.4
$(R, \hat{\mathbf{R}}) \oplus (D, \hat{\mathbf{N}}) \oplus (R, A, SI)$ with L^1	62.6
$(R, \hat{\mathbf{R}}) \oplus (D, \hat{\mathbf{N}})$ with *Invariant-*L^1	**65.9**

DBF: *Comparison to the State-of-the-Art*



DBF: *Comparison to the State-of-the-Art*

Effectiveness Result II

On **PSB**:

1. DBF is better than any other 3D method.
2. DBF is equally well as the best 2D method known (DBI).

DBF: *Performance Across Different Databases*

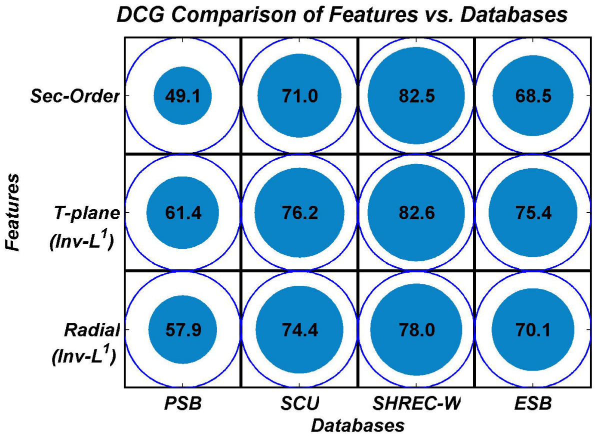
Reminder: 3D Object Databases

	PSB	SCU	SHREC-W	ESB
#(Classes)	92	53	20	45
Resolution	Low	High	High	Medium
Watertight?	No	Yes	Yes	Yes
Smooth?	No	Yes	Yes	No

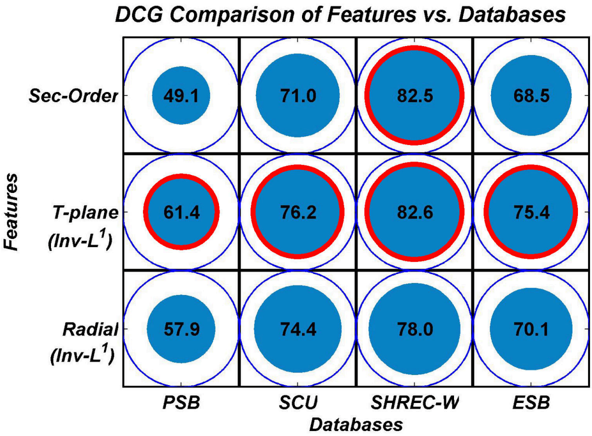
Questions

1. Given a database, which feature is the most effective?
2. Does descriptor combination work for all databases?

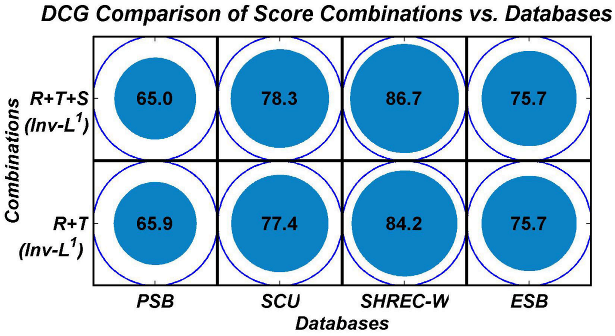
DBF: *Performance Across Different Databases*



DBF: *Performance Across Different Databases*

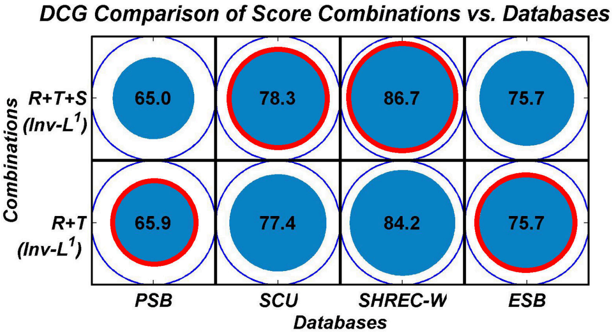


DBF: *Performance Across Different Databases*



$$R + T \equiv \textbf{Radial} \oplus \textbf{T-plane}$$
$$R + T + S \equiv \textbf{Radial} \oplus \textbf{T-plane} \oplus \textbf{Sec-Order}$$

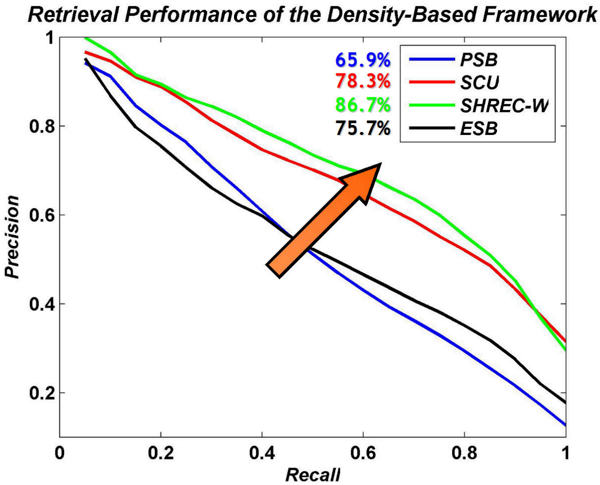
DBF: *Performance Across Different Databases*



$R + T \equiv$ **Radial** $\oplus$ **T-plane**

$R + T + S \equiv$ **Radial** $\oplus$ **T-plane** $\oplus$ **Sec-Order**

DBF: *Performance Across Different Databases*



DBF: *Performance Across Different Databases*

Effectiveness Result III

DBF generalizes well on different 3D databases.

DBF

Conclusion on DBF

1. Effectiveness
2. **Efficiency**
   - **Computational** → FGT
   - **Storage-wise** → Marginalization + Pruning
3. **Flexibility** → Simple features + KDE
4. **Robustness** → Bandwidth in KDE
5. **Invariance** → Axis relabelings and reflections

DBF

Conclusion on DBF

1. **Effectiveness** → **State-of-the-Art**
2. **Efficiency**
   - **Computational** → FGT
   - **Storage-wise** → Marginalization + Pruning
3. **Flexibility** → Simple features + KDE
4. **Robustness** → Bandwidth in KDE
5. **Invariance** → Axis relabelings and reflections

Outline

Statistical Similarity Learning (SSL)

Question

Can we boost the performance by choosing a “good” similarity?

Statistical Similarity Learning (SSL)

Question

Can we boost the performance by choosing a “good” similarity?

- **Test among possible choices and pick the best?**

Statistical Similarity Learning (SSL)

DCG (%) Values on PSB Training using Standard Similarity Measures

Descriptor	L^1	L^2	L^∞	KL	χ^2	B
Radial	**57.0**	54.7	44.4	54.4	**57.0**	**56.7**
T-plane	**59.8**	55.3	47.1	58.2	**61.1**	**59.4**

$$\boxed{L^1 \sim \chi^2 \sim Bhattacharyya(B)}$$

Statistical Similarity Learning (SSL)

Question

Can we boost the performance by choosing a “good” similarity?

- Test among possible choices and pick the best?

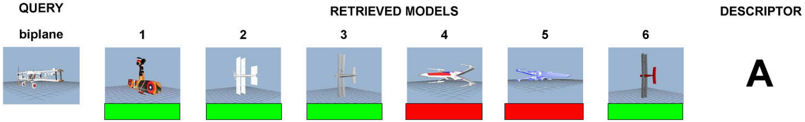
... or learn the similarities using supervision!

SSL: *Motivational Example*

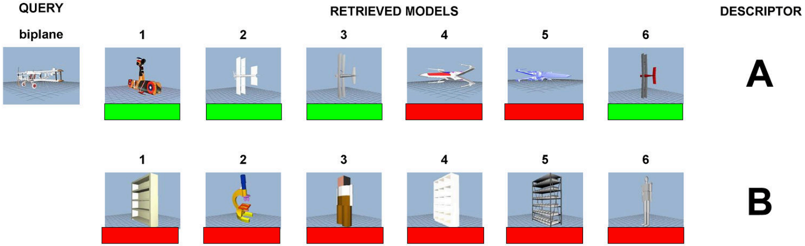
QUERY

biplane

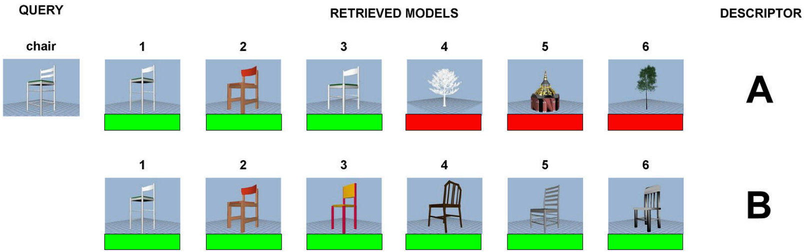
SSL: *Motivational Example*



SSL: *Motivational Example*



SSL: *Motivational Example (cont'd)*



SSL: *Score Fusion*

Motivation & Approach

◊ Different aspects of similarity $\leftrightarrow$ Different descriptors

- No single descriptor can encode all the shape information.
- No single descriptor can perform well for all types of queries.

SSL: *Score Fusion*

Motivation & Approach

◊ Different aspects of similarity $\leftrightarrow$ Different descriptors

- No single descriptor can encode all the shape information.
- No single descriptor can perform well for all types of queries.

$\rightarrow$ **Combine similarity scores** in a **supervised** manner

$\rightarrow$ **Linear similarity model**

SSL: *Score Fusion*

Our Solution

- Ranking shape instances based on their relevance to the query

$\longrightarrow$ **Empirical Ranking Risk (ERR):**
Number of misranked database shapes w.r.t. a query

- [Clémençon et al., 2006]:
Minimize a convex regularized version of ERR

SSL: *Score Fusion*

Our Solution

- Ranking shape instances based on their relevance to the query

$\longrightarrow$ **Empirical Ranking Risk (ERR):**
Number of misranked database shapes w.r.t. a query

- [Clémençon et al., 2006]:
  Minimize a convex regularized version of ERR
- Learning a **linear scoring function**
  $\Leftrightarrow$ Supervised binary classification in score difference domain
  $\longrightarrow$ Basically an **SVM-type** of learning scheme

Outline

SSL: *Score Fusion by Ranking Risk Minimization*

Notations

SSL: *Score Fusion by Ranking Risk Minimization*

Notations

- **Generic database shapes** x, x'

SSL: *Score Fusion by Ranking Risk Minimization*

Notations

- Generic database shapes x, x'
- **Query shape** q

SSL: *Score Fusion by Ranking Risk Minimization*

Notations

- Generic database shapes x, x'
- Query shape q
- **Similarity values** $s_k \triangleq sim_k(x, q), k = 1, \ldots, K$
  or more compactly: $\mathbf{s} = [s_1, \ldots, s_K] \in \mathbb{R}^K$

SSL: *Score Fusion by Ranking Risk Minimization*

Notations

- Generic database shapes x, x'
- Query shape q
- Similarity values $s_k \triangleq sim_k(x, q), k = 1, \ldots, K$
  or more compactly: $\mathbf{s} = [s_1, \ldots, s_K] \in \mathbb{R}^K$
- **Linear scoring function** $S = \langle \mathbf{w}, \mathbf{s} \rangle = \sum_k w_k s_k$

SSL: *Score Fusion by Ranking Risk Minimization*

Notations

- Generic database shapes x, x'
- Query shape q
- Similarity values $s_k \triangleq sim_k(x, q), k = 1, \ldots, K$
  or more compactly: $\mathbf{s} = [s_1, \ldots, s_K] \in \mathbb{R}^K$
- Linear scoring function $S = \langle \mathbf{w}, \mathbf{s} \rangle = \sum_k w_k s_k$
- **The sought-after weight vector $\mathbf{w} \in \mathbb{R}^K$**

SSL: *Score Fusion by Ranking Risk Minimization*

Notations

- Generic database shapes x, x'
- Query shape q
- Similarity values $s_k \triangleq sim_k(x, q), k = 1, \ldots, K$
  or more compactly: $\mathbf{s} = [s_1, \ldots, s_K] \in \mathbb{R}^K$
- Linear scoring function $S = \langle \mathbf{w}, \mathbf{s} \rangle = \sum_k w_k s_k$
- The sought-after weight vector $\mathbf{w} \in \mathbb{R}^K$
- **Relevance variable** y, e.g., in *bipartite ranking:*

$$y = 1,\ x \text{ is relevant to } q$$
$$y = -1,\ x \text{ is } \textit{not} \text{ relevant to } q$$

SSL: *Score Fusion by Ranking Risk Minimization*

Scoring function $S = \langle \mathbf{w}, \mathbf{s} \rangle$ should satisfy:

$$\begin{array}{ll} S(x,q) > S(x',q) & \text{if } x \text{ is more relevant to } q \text{ than } x', \\ S(x,q) < S(x',q) & \text{otherwise.} \end{array}$$

SSL: *Score Fusion by Ranking Risk Minimization*

Scoring function $S = \langle \mathbf{w}, \mathbf{s} \rangle$ should satisfy:

$$\begin{array}{ll} S(x,q) > S(x',q) & \text{if } x \text{ is more relevant to } q \text{ than } x', \\ S(x,q) < S(x',q) & \text{otherwise.} \end{array}$$

$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{s} \rangle > \langle \mathbf{w}, \mathbf{s}' \rangle & \text{if } y - y' > 0, \\ \langle \mathbf{w}, \mathbf{s} \rangle < \langle \mathbf{w}, \mathbf{s}' \rangle & \text{if } y - y' < 0. \end{array}$$

SSL: *Score Fusion by Ranking Risk Minimization*

Scoring function $S = \langle \mathbf{w}, \mathbf{s} \rangle$ should satisfy:

$$\begin{array}{ll} S(x,q) > S(x',q) & \text{if } x \text{ is more relevant to } q \text{ than } x', \\ S(x,q) < S(x',q) & \text{otherwise.} \end{array}$$

$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{s} - \mathbf{s}' \rangle > 0 & \text{if } y - y' > 0, \\ \langle \mathbf{w}, \mathbf{s} - \mathbf{s}' \rangle < 0 & \text{if } y - y' < 0. \end{array}$$

SSL: *Score Fusion by Ranking Risk Minimization*

Scoring function $S = \langle \mathbf{w}, \mathbf{s} \rangle$ should satisfy:

$$\begin{array}{ll} S(x,q) > S(x',q) & \text{if } x \text{ is more relevant to } q \text{ than } x', \\ S(x,q) < S(x',q) & \text{otherwise.} \end{array}$$

$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{s} - \mathbf{s}' \rangle > 0 & \text{if } y - y' > 0, \\ \langle \mathbf{w}, \mathbf{s} - \mathbf{s}' \rangle < 0 & \text{if } y - y' < 0. \end{array}$$

Let $z \triangleq sign(y - y')$ and $\mathbf{v} \triangleq \mathbf{s} - \mathbf{s}'$

SSL: *Score Fusion by Ranking Risk Minimization*

Scoring function $S = \langle \mathbf{w}, \mathbf{s} \rangle$ should satisfy:

$$\begin{array}{ll} S(x,q) > S(x',q) & \text{if } x \text{ is more relevant to } q \text{ than } x', \\ S(x,q) < S(x',q) & \text{otherwise.} \end{array}$$

$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{s} - \mathbf{s}' \rangle > 0 & \text{if } y - y' > 0, \\ \langle \mathbf{w}, \mathbf{s} - \mathbf{s}' \rangle < 0 & \text{if } y - y' < 0. \end{array}$$

Let $z \triangleq sign(y - y')$ and $\mathbf{v} \triangleq \mathbf{s} - \mathbf{s}'$

$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{v} \rangle > 0 & \text{if } z = +1, \\ \langle \mathbf{w}, \mathbf{v} \rangle < 0 & \text{if } z = -1. \end{array}$$

SSL: *Score Fusion by Ranking Risk Minimization*

Scoring function $S = \langle \mathbf{w}, \mathbf{s} \rangle$ should satisfy:

$$\begin{array}{ll} S(x,q) > S(x',q) & \text{if } x \text{ is more relevant to } q \text{ than } x', \\ S(x,q) < S(x',q) & \text{otherwise.} \end{array}$$

$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{s} - \mathbf{s}' \rangle > 0 & \text{if } y - y' > 0, \\ \langle \mathbf{w}, \mathbf{s} - \mathbf{s}' \rangle < 0 & \text{if } y - y' < 0. \end{array}$$

Let $z \triangleq sign(y - y')$ and $\mathbf{v} \triangleq \mathbf{s} - \mathbf{s}'$

$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{v} \rangle > 0 & \text{if } z = +1, \\ \langle \mathbf{w}, \mathbf{v} \rangle < 0 & \text{if } z = -1. \end{array}$$

This is the binary classification problem!

SVM-based solution

SSL: *Score Fusion by Ranking Risk Minimization*

Remark

Ranking risk in score domain

$\Leftrightarrow$

Classification error in score difference domain

Ranking Risk in score domain

$$\hat{R}(S;q) = \frac{2}{N(N-1)} \sum_{m<n} \mathbb{I}\left\{(S(x_m,q) - S(x_n,q)) \cdot (y_m - y_n) < 0\right\}$$

Classification error in score difference domain

$$\hat{R}(\mathbf{w};q) = \frac{2}{N(N-1)} \sum_{m<n} \mathbb{I}\left\{\langle \mathbf{w}, \mathbf{v}_{m,n}\rangle \, z_{m,n} < 0\right\}$$

Outline

SSL: *Retrieval Protocols*

1. Bimodal
2. Two-round (Relevance feedback)
   1. On-line version
   2. Off-line version

SSL: *Bimodal Protocol*

Database Shapes x_t

Query q

+

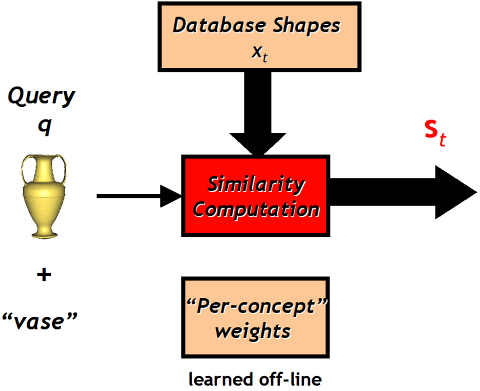
"vase"

Similarity Computation

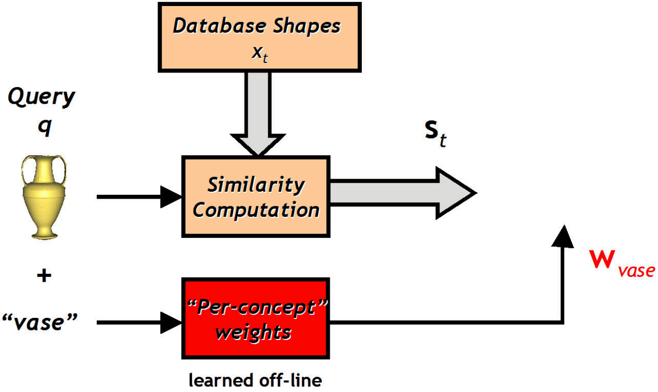
"Per-concept" weights

learned off-line

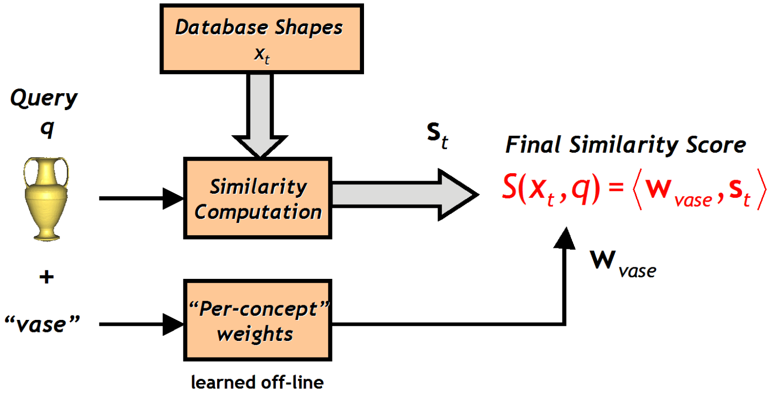
SSL: *Bimodal Protocol*



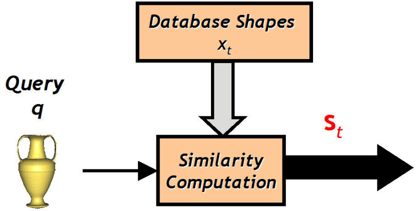
SSL: *Bimodal Protocol*



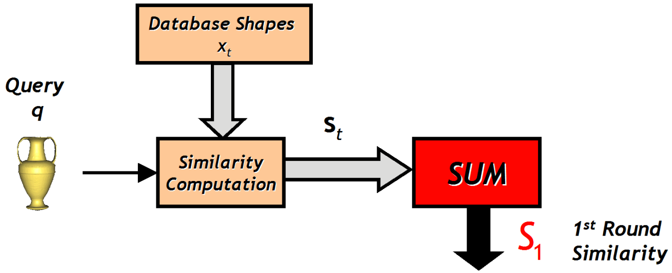
SSL: *Bimodal Protocol*



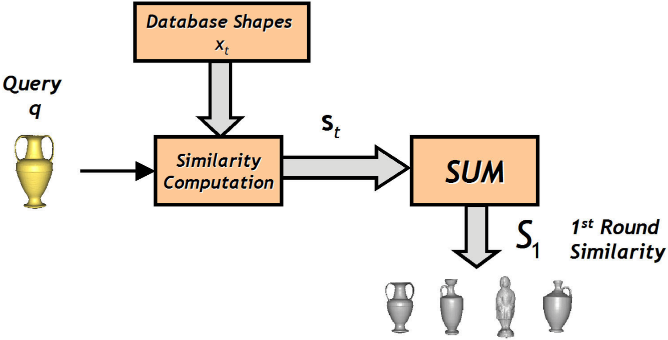
SSL: *Two-round Protocol (On-line)*



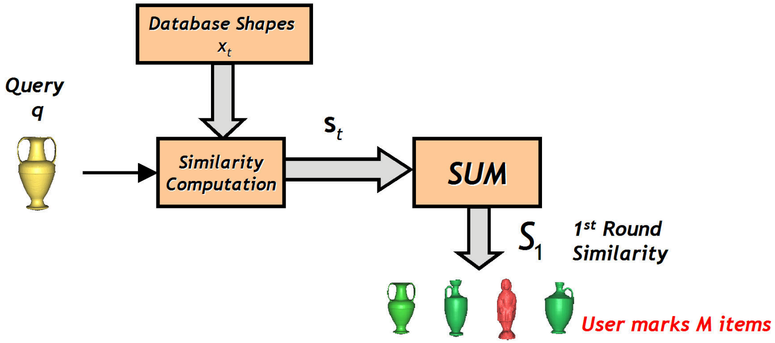
SSL: *Two-round Protocol (On-line)*



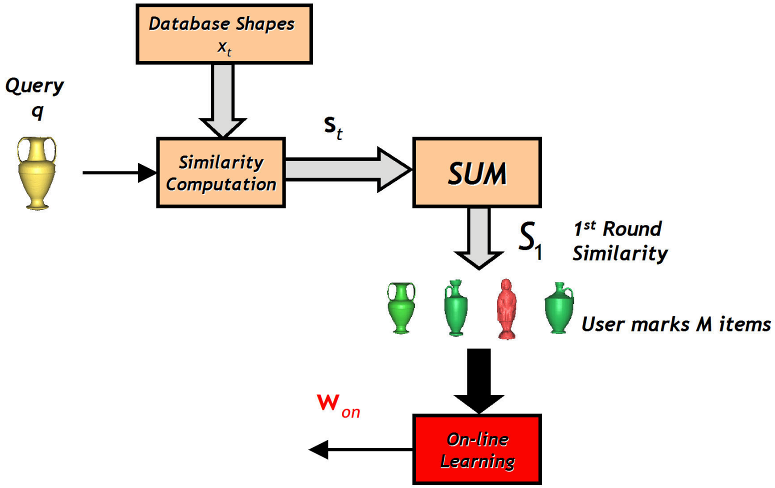
SSL: *Two-round Protocol (On-line)*



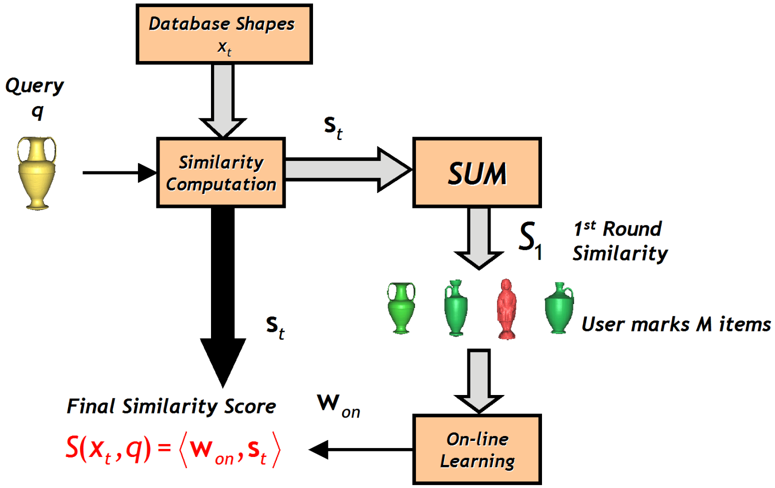
SSL: *Two-round Protocol (On-line)*



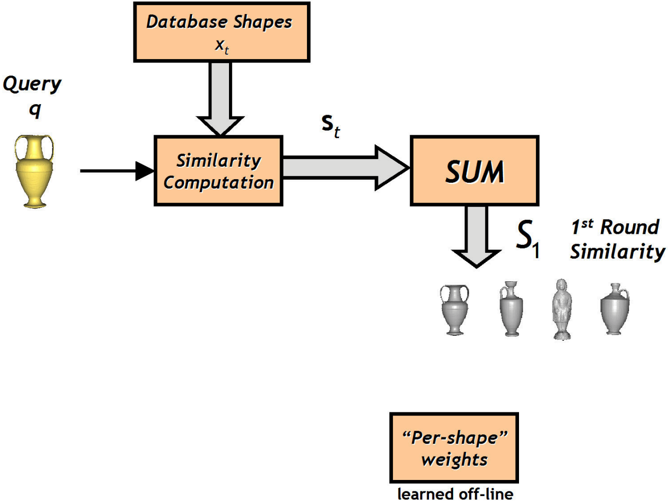
SSL: *Two-round Protocol (On-line)*



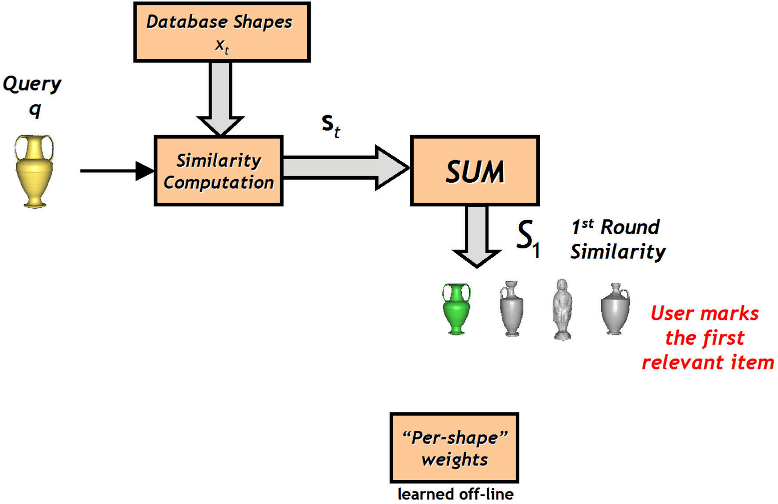
SSL: *Two-round Protocol (On-line)*



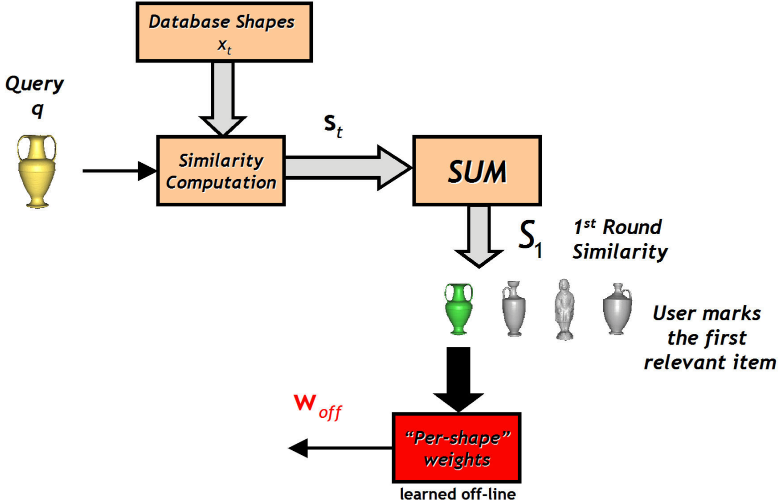
SSL: *Two-round Protocol (Off-line)*



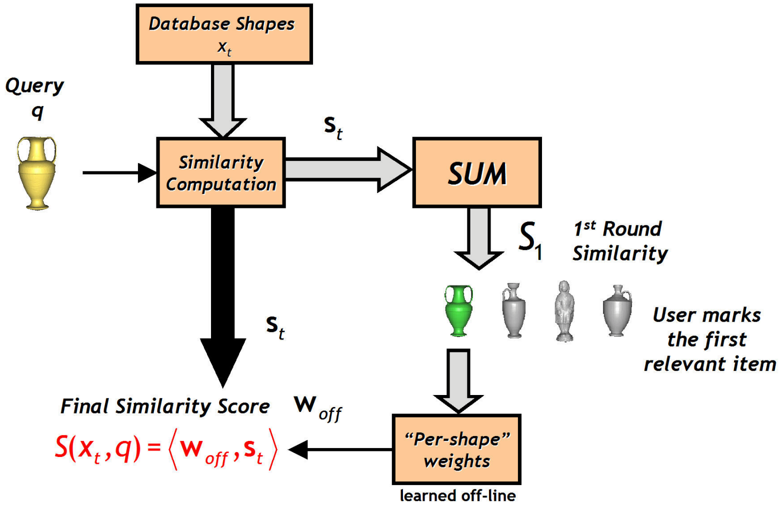
SSL: *Two-round Protocol (Off-line)*



SSL: *Two-round Protocol (Off-line)*



SSL: *Two-round Protocol (Off-line)*



Outline

SSL: *Experiments*

Database

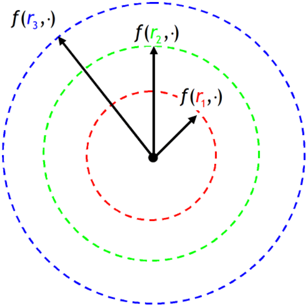
- **Princeton Shape Benchmark:** 1814 models in 161 shape concepts
- **Shape concepts:** *human, animal, tool, vehicle, household, etc.*
- **Set A:** 946 instances, **Set B:** 868 instances

	Set A	**Set B**
Bimodal	*Training*	*Test*
Two-round	*Database*	*Queries*

SSL: *Experiments*

Descriptors

- **Radial** $(R, \hat{\mathbf{R}})$-Descriptor
- **T-plane** $(D, \hat{\mathbf{N}})$-Descriptor
- **Sec-Order** (R, A, SI)-Descriptor



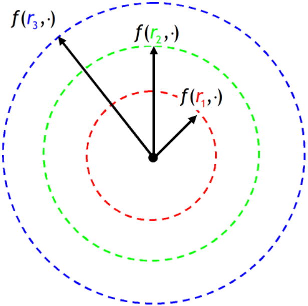
SSL: *Experiments*

Descriptors

- **Radial** $(R, \hat{\mathbf{R}})$-Descriptor
- **T-plane** $(D, \hat{\mathbf{N}})$-Descriptor
- **Sec-Order** (R, A, SI)-Descriptor

$\rightarrow$ **Radialized** descriptors

$\rightarrow$ Density **at 8 concentric shells**



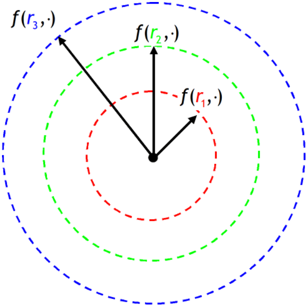
SSL: *Experiments*

Descriptors

- **Radial** $(R, \hat{\mathbf{R}})$-Descriptor
- **T-plane** $(D, \hat{\mathbf{N}})$-Descriptor
- **Sec-Order** (R, A, SI)-Descriptor

→ **Radialized** descriptors

→ Density **at 8 concentric shells**



$\mathbf{3} \times \mathbf{8} = \mathbf{24}$ descriptors in total $\rightarrow$ 24 similarity values $\Rightarrow \mathbf{s} \in \mathbb{R}^{24}$

SSL: *Performance in Bimodal Search*

DCG Performance

Score	PSB Set A	PSB Set B
SUM		
SSL		

SSL: *Performance in Bimodal Search*

DCG Performance

Score	PSB Set A	PSB Set B
SUM	61.6±28.1	
SSL	**74.9±25.2**	

SSL: *Performance in Bimodal Search*

DCG Performance

Score	PSB Set A	PSB Set B
SUM	61.6±28.1	60.6±28.1
SSL	**74.9±25.2**	**62.5±27.7**

SSL: *Performance in Bimodal Search*

DCG Performance

Score	PSB Set A	PSB Set B
SUM	61.6±28.1	60.6±28.1
SSL	**74.9±25.2**	**62.5±27.7**

Comment

Not very impressive on the Test Set!

SSL: *Performance in Bimodal Search*

DCG Performance

Score	PSB Set A	PSB Set B
SUM	61.6±28.1	60.6±28.1
SSL	**74.9±25.2**	**62.5±27.7**

Comment

Not very impressive on the Test Set!

BUT!

On Set B, *SSL* didn't work for 61 concepts (out of 161).

SSL: *Performance in Bimodal Search*

DCG Performance

Score	PSB Set A	PSB Set B
SUM	61.6±28.1	60.6±28.1
SSL	**74.9±25.2**	**62.5±27.7**
SUM+SSL	-	**64.4±23.9**

Comment

Not very impressive on the Test Set!

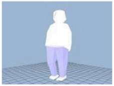
BUT!

On Set B, *SSL* didn't work for 61 concepts (out of 161).

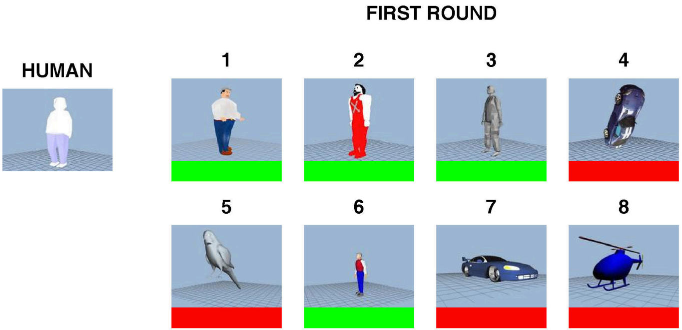
If we use the SUM rule for negatively affected concepts...

SSL: *Two-round On-line Example Query*

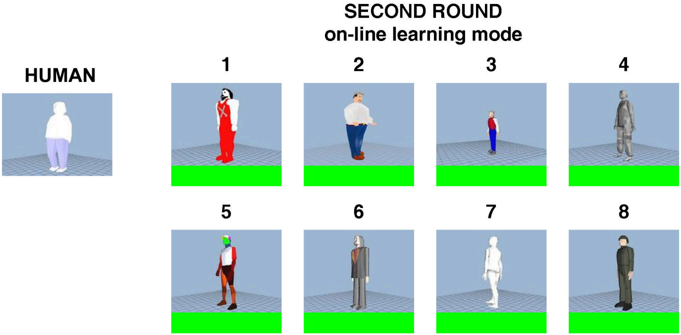
HUMAN



SSL: *Two-round On-line Example Query*

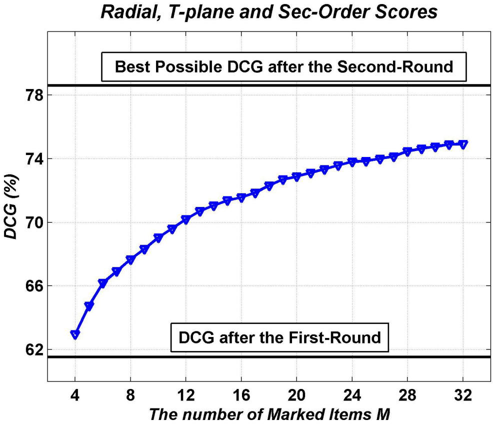


SSL: *Two-round On-line Example Query*



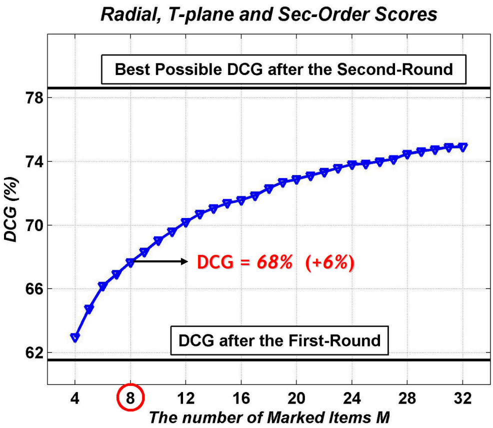
SSL: *Performance in Two-round Search (On-line)*

On-line version



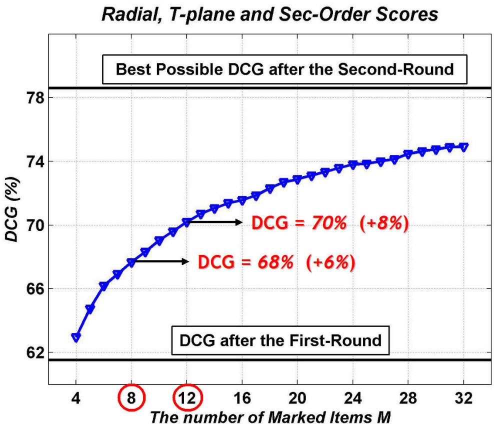
SSL: *Performance in Two-round Search (On-line)*

On-line version



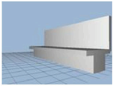
SSL: *Performance in Two-round Search (On-line)*

On-line version

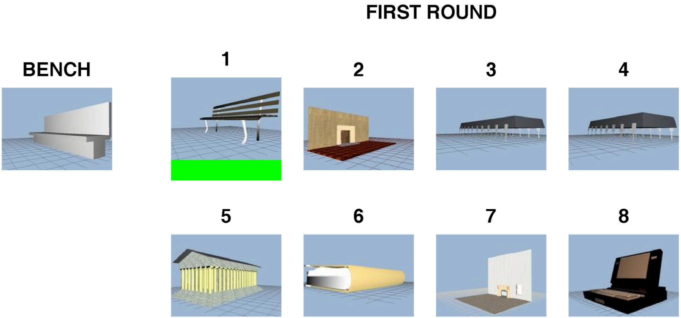


SSL: *Two-round Off-line Example Query*

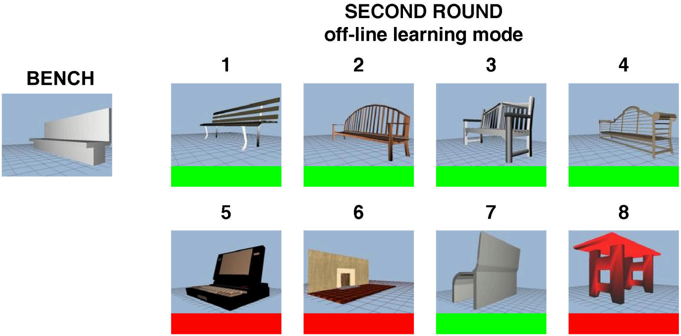
BENCH



SSL: *Two-round Off-line Example Query*

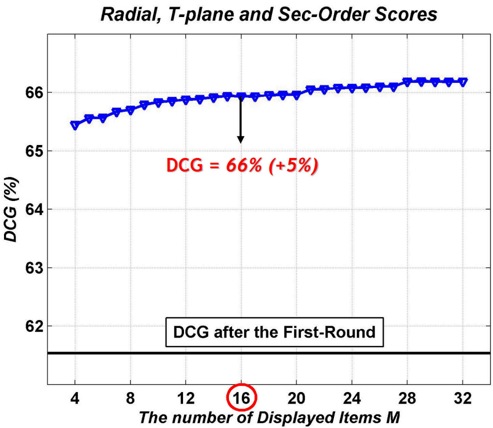


SSL: *Two-round Off-line Example Query*



SSL: *Performance in Two-round Search (Off-line)*

Off-line version



SSL

Additive DCG Gain of SSL

Two-Round On-line ($M = 8$)	Two-Round On-line ($M = 12$)	Two-Round Off-line	Bimodal
6.0	8.0	5.0	2.0-4.0

Conclusion on SSL

SSL improves retrieval effectiveness.

Outline

Contributions

Density-Based Shape Description Framework

- A family of 3D shape descriptors
- A unifying approach for histogram-based methods
- **A State-of-the-Art shape description scheme**
    - Effective
    - Efficient
    - Flexible
    - Robust
    - Invariant

Contributions

Statistical Similarity Learning

- A score fusion approach with supervision
- A first application in 3D shape retrieval
- Independent of description modality
  $\longrightarrow$ Applicable to any type of retrieval problem
- Satisfactory performance in different protocols

Perspectives

Perspectives on DBF

- Extending invariant matching to arbitrary rotations
- Parametric density estimation
- Information-theoretical analysis of local surface features

Perspectives

Perspectives on Similarity Learning

- Application to other description schemes
- Non-linear scoring functions
- DCG-based criteria