Shape Matching for 3D Retrieval and Recognition

Ivan Sipiran and Benjamin Bustos

PRISMA Research Group
Department of Computer Science
University of Chile

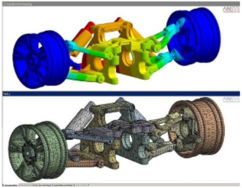
SIBGRAPI 2013 Tutorial , Arequipa - Perú, August 5, 2013

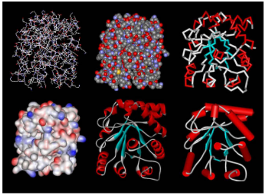
Outline

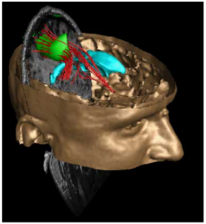
3D collections

3D collections

3D applications

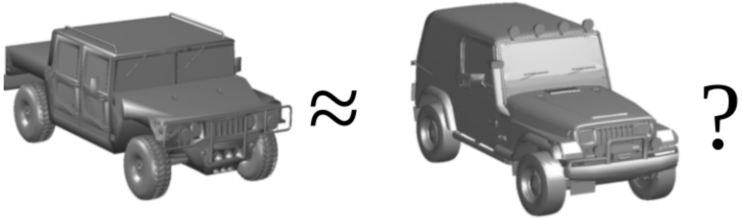






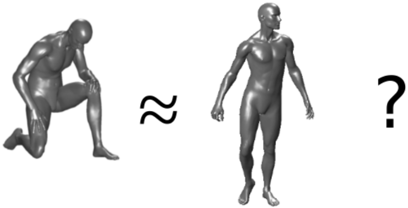
3D as media

- The same problem as other media
  - Representation
  - Storage
  - Analysis
  - Processing
- Content-based matching or ...

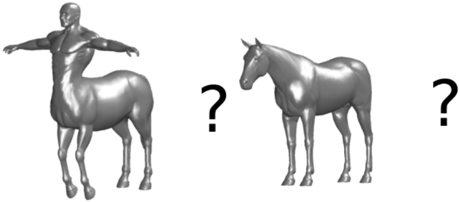


The problem with matching

Non-rigid matching

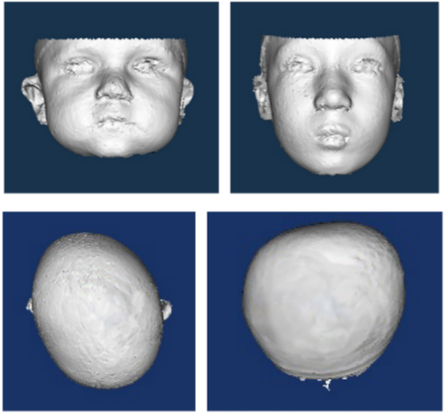


Partial matching



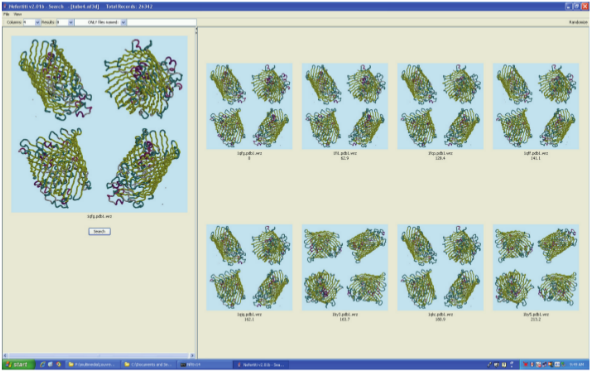
Craniofacial research

- 3D features to detect anomalies (Atmosukarto et al. 2010)



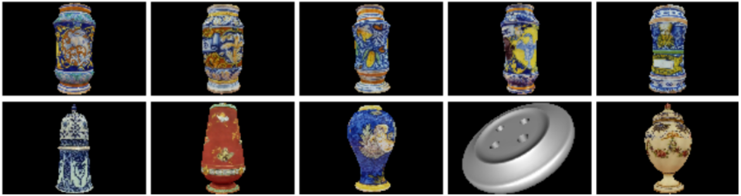
3D protein retrieval and classification

- Searching for similar structures (Paquet and Viktor, 2008)



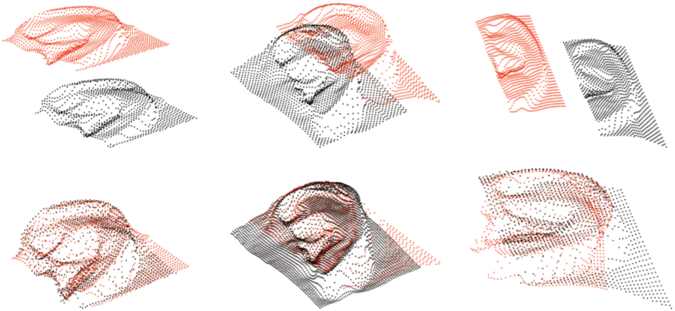
3D retrieval for museums

- 3D retrieval for navigation (Goodall et al. 2004)



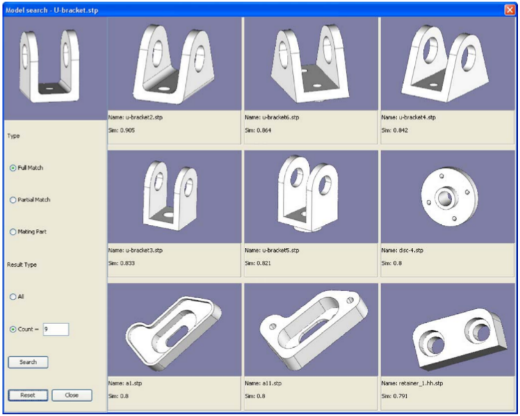
Human ear recognition in 3D

- 3D features to represent an ear (Chen and Bhanu, 2009)



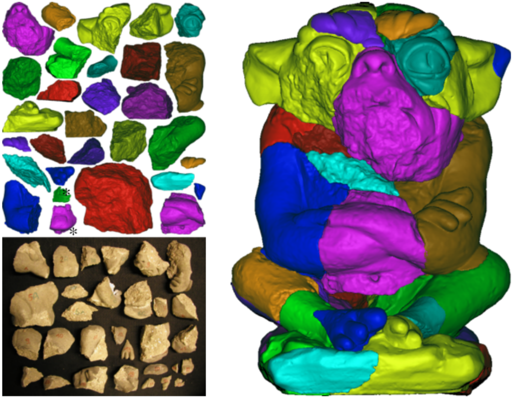
CAD/CAM

- Manufacturing and production (You and Tsai, 2010)



Archeology

- Matching for reconstruction (Huang et al. 2006)

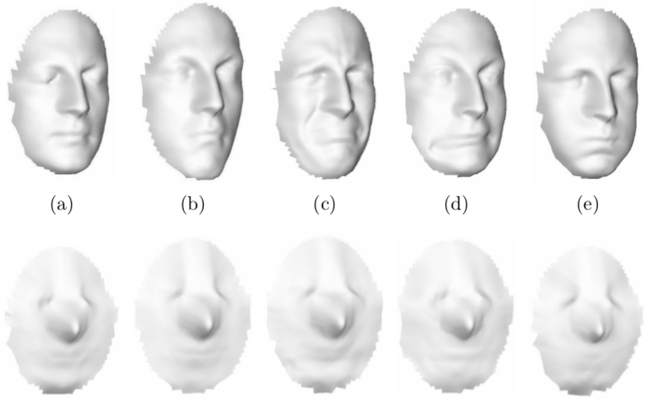


3D video sequences

- Characterize a motion (Huang et al., 2010)

3D face recognition

- Gesture-invariant representation (Bronstein et al. 2005)

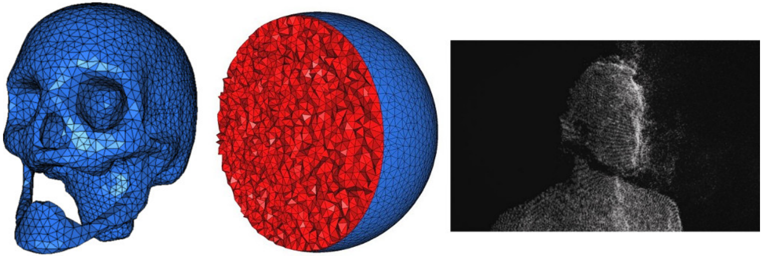


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

3D Representations

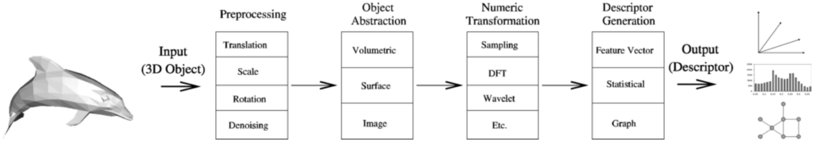
- Triangular meshes (in this tutorial)
- Volumes
- Point cloud



Outline

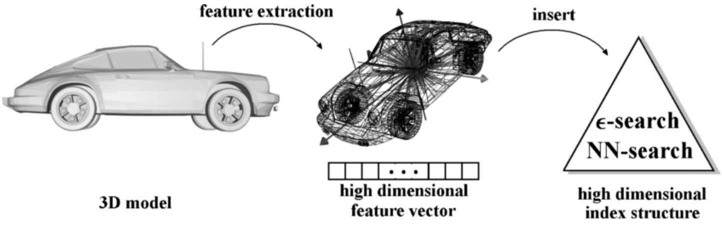
The global approach

- Transform a 3D object into a numeric/symbolic representation
    - Feature vectors
    - Graphs
- Compare two objects through their representations



The global approach

- Feature vector approach has been extensively studied
    - Scalability

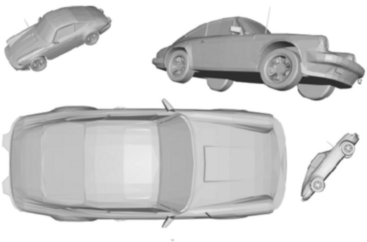


Depth-buffer descriptor

- Image-based descriptor (Vranic 2004)
  - Pose normalization
  - Depth-buffer construction
  - Fourier transformations
  - Selection of coefficients

Depth-buffer descriptor

- Pose normalization - Typical procedure
  - Translate the center of mass to the origin of the coordinate system
  - Rotate according to the largest spread
  - Scale to common size



Depth-buffer descriptor

- Pose normalization - Continuous PCA
  - Let $f : \mathbb{T} \rightarrow \mathbb{M}$ be a function on the set of triangles $\mathbb{T}$ in $\mathbb{R}^3$.
  - Let us define an operator for the function f on the set $\mathbb{T}$,

$$I_f(T_i) = \int\int_{v \in T_i} f(v)ds$$
$$= 2S_i \int_0^1 d\alpha \int_0^{1-\alpha} f(\alpha p_{A_i} + \beta p_{B_i} + (1-\alpha-\beta)p_{C_i})d\beta$$

  - In addition

$$I_f(I) = \sum_{i=1}^{m} I_f(T_i) = \int\int_{v \in I} f(v)ds$$

Depth-buffer descriptor

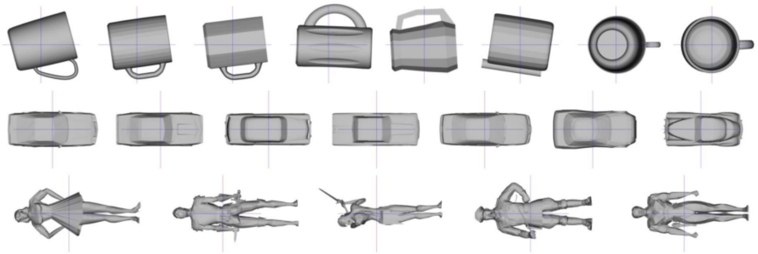
- Pose normalization - Continuous PCA
  - When $f(v) = 1$, $I_f(I)$ is the surface area.
  - When $f(v) = v$, $I_f(I) = m_I$ is the center of mass.
  - When $f(v) = (v - m_I)(v - m_I)^T$, $I_f(I)$ evaluates to the covariance matrix

$$C_I = \frac{1}{S}\int\int_{v\ inI}(v - m_I)(v - m_I)^T ds$$

$$= \frac{1}{12S}\sum_{i=1}^{m}(f(p_{A_i}) + f(p_{B_i}) + f(p_{C_i}) + 9f(g_i))S_i$$

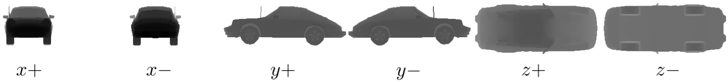
Depth-buffer descriptor

- Pose normalization
  - With the continuous covariance matrix C_I, PCA can be applied as usual



Depth-buffer descriptor

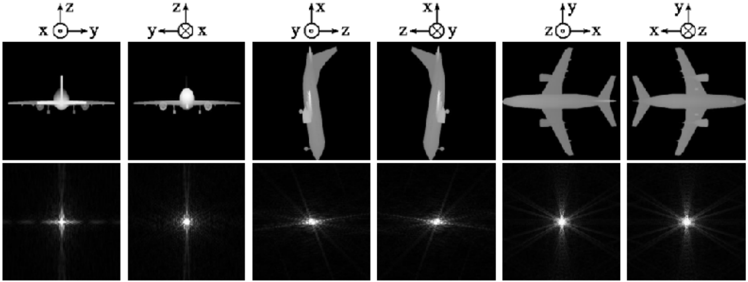
- Construction
  - Project the object into the faces of a bounding rectangle



Depth-buffer descriptor

- Fourier transformation

$$\hat{f}_{pq} = \frac{1}{\sqrt{MN}} \sum_{a=0}^{M-1} \sum_{b=0}^{N-1} f_{ab} exp(-j2\pi(pa/M + qb/N))$$



Depth-buffer descriptor

- Selection of coefficients
  - As depth-buffers are real, coefficient posses the symmetry property.
  - Select coefficients whose indices satisfy

$$|p - N/2| + |q - N/2| \leq k \leq N/2$$

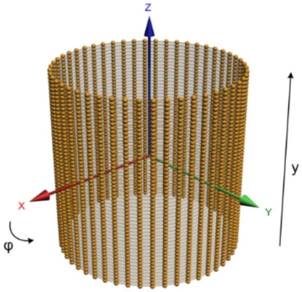
for some natural number k

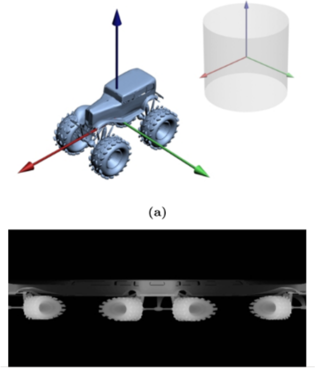
PANORAMA descriptor

- Image-based descriptor (Papadakis et al. 2009)
  - Pose normalization (Continuous PCA)
  - Cylindrical projection
  - Fourier and Wavelet transformations

PANORAMA descriptor

- Cylindrical projection





PANORAMA descriptor

- Fourier coefficients
- Haar and Coiflet wavelets (features computed on sub-images of the DWT)
  - Mean

$$\mu = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} I(x, y)$$

  - Standard deviation

$$\sigma = \sqrt{\frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (I(x, y) - \mu)^2}$$

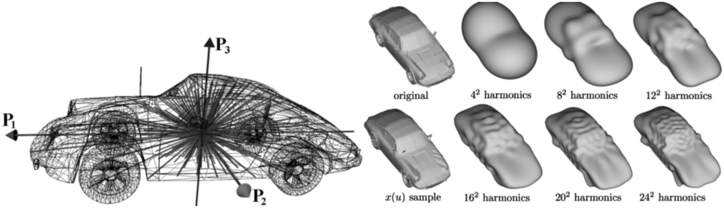
PANORAMA descriptor

- Fourier coefficients
- Haar and Coiflet wavelets (features computed on sub-images of the DWT)
    - Skewness

$$\beta = \frac{\frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (I(x,y) - \mu)^3}{\sigma^3}$$

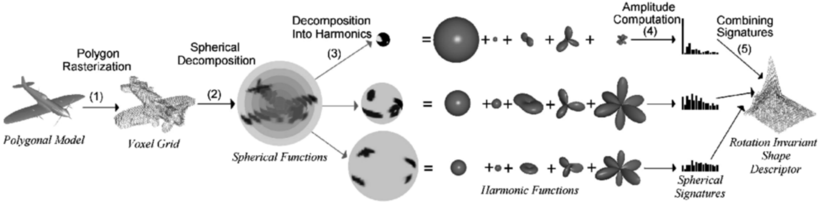
Other approaches

- Ray-based feature vector (Vranic 2004)



Other approaches

- 3D harmonics (Funkhouser et al. 2003)



Other approaches

- SHREC 2009 Generic Shape Retrieval: Competition with 20+ algorithms (Godil et al. 2009)

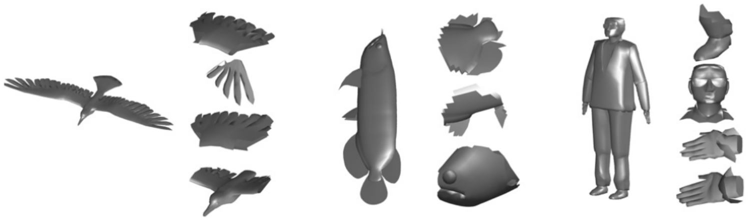
PARTICIPANT	METHOD	NN	FT	ST	E	DCG
Akgül	DBFc8	0.825	0.433	0.550	0.383	0.748
(sect. 5.6)	DBFc10	0.825	0.443	0.574	0.398	0.757
	DBFc12	0.813	0.449	0.578	0.406	0.759
Bustos	DSR_segment	0.863	0.561	0.696	0.49	0.825
(sect. 5.5)	DSR_nosegment	0.85	0.546	0.691	0.479	0.819
	Entropy_123_6_segment	0.838	0.526	0.663	0.464	0.803
	Entropy_6789_6_segment	0.838	0.528	0.668	0.467	0.805
	W1_segment	0.838	0.528	0.666	0.466	0.806
Chaouch (sect. 5.1)	MDLA	0.963	0.730	0.848	0.602	0.917
Daras	3D_shape_impact	0.8	0.447	0.567	0.396	0.749
(sect. 5.3)	Compact_multiview	0.8	0.49	0.626	0.437	0.771
	Compound_SID_CMVD	0.875	0.558	0.69	0.487	0.83
Furuya	BF-SIFT	0.850	0.483	0.624	0.433	0.777
(sect. 5.6)	MR-SPRH-UDR	0.875	0.550	0.703	0.491	0.824
Lian	SHD+GSMD	0.875	0.597	0.733	0.514	0.85
(sect. 5.2)	RECT+SHD+GSMD	0.925	0.633	0.778	0.542	0.875
	RECT+SHD+GSMD+MR	0.925	0.724	0.844	0.595	0.904
Napoléon	Run1	0.900	0.522	0.665	0.459	0.814
(sect. 5.4)	Run2	0.950	0.615	0.701	0.502	0.864
	Run3	0.950	0.639	0.771	0.540	0.882
	Run4	0.900	0.550	0.662	0.465	0.826
	Run5	0.887	0.570	0.709	0.497	0.838

Global + Local approach

- Trying to take advantage of the local information in shapes (Sipiran et al. 2013)

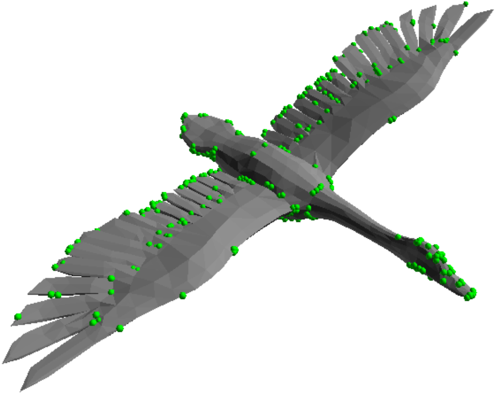
Global + Local approach

- We need discriminative and robust partitions
- Local features-based approach



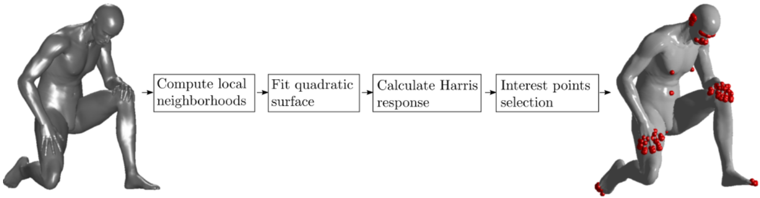
Data-aware 3D partitioning

- Step 1: Detection of keypoints
  - Harris 3D algorithm (Sipiran and Bustos, 2011)



Data-aware 3D partitioning

- Harris 3D algorithm
  - Pipeline



Data-aware 3D partitioning

- Harris algorithm
  - Extension of the well-known method for images
  - Harris algorithm
    - Autocorrelation function

$$e(x, y) = \sum_{x_i, y_i} W(x_i, y_i)[I(x_i + \triangle x, y_i + \triangle y) - I(x_i, y_i)]^2$$

where $I(.,.)$ denotes the image function and (x_i, y_i) are the points in the Gaussian function W centered on (x, y), which defines the neighborhood area in analysis.

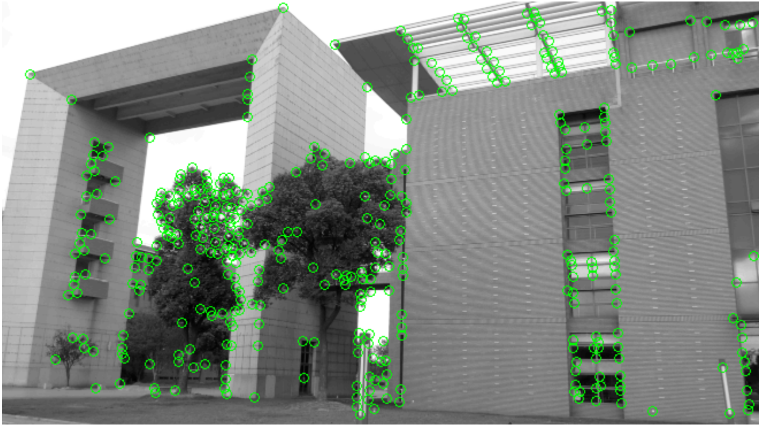
Data-aware 3D partitioning

- Harris 3D algorithm
  - Using a Taylor expansion

$$e(x,y) = \vec{S}\begin{bmatrix} \sum_{x_i,y_i} W.I_x^2 & \sum_{x_i,y_i} W.I_x.I_y \\ \sum_{x_i,y_i} W.I_x.I_y & \sum_{x_i,y_i} W.I_y^2 \end{bmatrix}\vec{S}^T$$
$$= \vec{S}E(x,y)\vec{S}^T$$

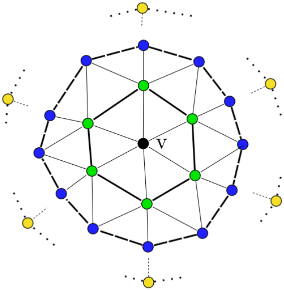
where $\vec{S} = [\triangle x \ \triangle y]$ is a shift vector, I_x and I_y denote the partial derivatives in x and y, and along with W are evaluated in (x_i, y_i) points.

- Harris algorithm



Data-aware 3D partitioning

- Harris 3D algorithm
  - Extension for 3D meshes in not trivial due to the lack of a regular neighborhood topology.
  - How to compute a neighborhood around a vertex?
    - Adaptive neighborhood



Data-aware 3D partitioning

- Harris 3D algorithm
  - Good choice: neighborhood dependent of the local structure

$$ring_k(v) = \{w \in V' \text{ such that } |shortest\_path(v, w)| = k\}$$

$$d_{ring}(v, ring_k(v)) = max_{w \in ring_k(v)} \|v - w\|_2$$

$$radius_v = \{k \in \mathbb{N} \text{ such that } d_{ring}(v, ring_k(v)) \geq \delta \text{ and } d_{ring}(v, ring_{k-1}(v)) < \delta\}$$

Data-aware 3D partitioning

- Harris 3D algorithm
  - Translate the neighborhood, v_i should be the origin
  - PCA to normalize the spread of the points. Optimally, points are well distributed in plane XY.
  - Fit a quadratic surface

$$z = f(x, y) = \frac{p_1}{2}x^2 + p_2xy + \frac{p_3}{2}y^2 + p_4x + p_5y + p_6$$

  - Function $f(x, y)$ is similar to an image

Data-aware 3D partitioning

- Harris 3D algorithm
  - In order to deal with local changes: smoothing

$$A = \frac{1}{2\sigma^4\pi}\int_{\mathbb{R}^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \cdot \left(\frac{\partial f(x,y)}{\partial x}\right)^2 dxdy$$

$$B = \frac{1}{2\sigma^4\pi}\int_{\mathbb{R}^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \cdot \left(\frac{\partial f(x,y)}{\partial y}\right)^2 dxdy$$

$$C = \frac{1}{2\sigma^4\pi}\int_{\mathbb{R}^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \cdot \left(\frac{\partial f(x,y)}{\partial x}\right)\left(\frac{\partial f(x,y)}{\partial y}\right) dxdy$$

Data-aware 3D partitioning

- Harris 3D algorithm
  - Evaluate the integrals to obtain the terms

$$A = \frac{p_4^2}{\sigma^2} + p_1^2 + p_2^2$$

$$B = \frac{p_5^2}{\sigma^2} + p_2^2 + p_3^2$$

$$C = \frac{p_4 p_5}{\sigma^2} + p_1 p_2 + p_2 p_3$$

Data-aware 3D partitioning

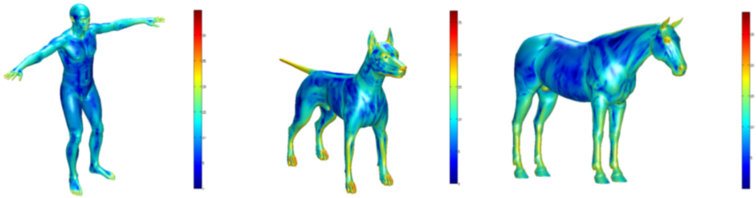
- Harris 3D algorithm
  - The autocorrelation matrix is then

$$E = \begin{pmatrix} A & C \\ C & B \end{pmatrix}$$

  - Now we can evaluate the Harris operator for each vertex in the mesh, as usual.
  - To detect keypoints, we can select, for instance, the top 1% vertices with the highest response.

Data-aware 3D partitioning

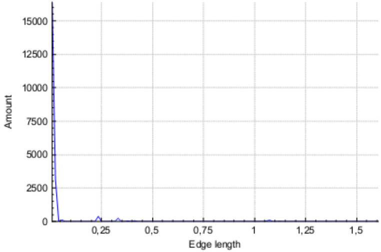
- Harris 3D algorithm
  - Saliency plot



Demo 1: Harris keypoints

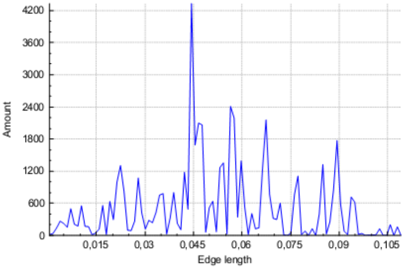
Data-aware 3D partitioning

- Step 1: Detection of keypoints
  - Meshes with bad triangulation
  - Control of resolution to improve triangulations (Johnson and Hebert, 1998)



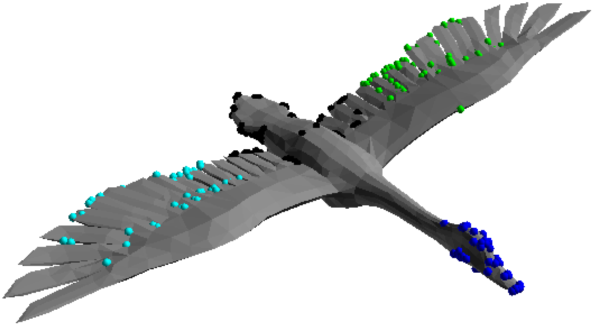
Data-aware 3D partitioning

- Step 1: Detection of keypoints
  - Algorithm controls the edge lengths



Data-aware 3D partitioning

- Step 2: Adaptive clustering of keypoints in Euclidean space
  - Near points: same clustering
  - Far points: different cluster



Adaptive clustering in $\mathbb{R}^n$

- Input: $P \in \mathbb{R}^n$, inter-cluster threshold R, intra-cluster threshold S, minimum number of elements N
  - For each $p \in P$, if p belongs to some existing cluster C_i, insert p into C_i
  - If p does not belong to any cluster, create a new cluster
  - For each cluster C_i, if $|C_i| < N$, then remove cluster, update centroid otherwise.
  - Repeat until satisfying some stop criterion

Data-aware 3D partitioning

- Step 3: Partitioning and description
  - Extract the patch enclosed by a sphere containing a cluster
  - We use a kd-tree to efficiently search vertices in the enclosing sphere
  - An object is represented as

$$S_O = \{(s_O, P_O) | s_O \in \mathbb{R}^n \text{ and } P_O = \{p_O^1, p_O^2, \ldots, p_O^m\}, p_O^i \in \mathbb{R}^n\}$$

where s_O is a global descriptor of the entire shape, and p_O^i is a global descriptor for a part.

Data-aware 3D partitioning

- Matching
  - Given two objects O and Q, with their representations

$$S_O = \{(s_O, P_O) | s_O \in \mathbb{R}^n \text{ and } P_O = \{p_O^1, p_O^2, \ldots, p_O^m\}, p_O^i \in \mathbb{R}^n\}$$

$$S_Q = \{(s_Q, P_Q) | s_Q \in \mathbb{R}^n \text{ and } P_Q = \{p_Q^1, p_Q^2, \ldots, p_Q^k\}, p_Q^i \in \mathbb{R}^n\}$$

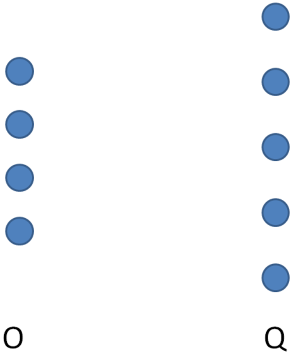
  - The distance is a linear combination

$$d(S_O, S_Q) = \mu \|s_O - s_Q\| + (1 - \mu) d(P_O, P_Q)$$

  - How to evaluate $d(P_O, P_Q)$ if it involves a many-to-many matching?

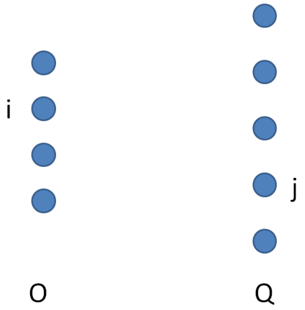
Data-aware 3D partitioning

- Matching



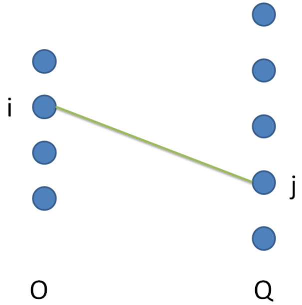
Data-aware 3D partitioning

- Matching



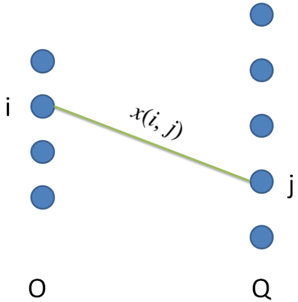
Data-aware 3D partitioning

- Matching



Data-aware 3D partitioning

- Matching



Data-aware 3D partitioning

- The correspondence can be formulated as a binary variable

$$x(i,j) = \begin{cases} 1, & \text{if } p_O^i \text{ matches } p_Q^j \\ 0 & \text{otherwise.} \end{cases}$$

- The problem is to find the best x

$$f(x) = \sum_{i,j} \|p_O^i - p_Q^j\|_2 . x(i,j)$$

- The optimum can be used to formulate a distance

$$d(P_O, P_Q) = \frac{f(x^*)}{min(|P_O|, |P_Q|)}$$

Data-aware 3D partitioning

- Matching is solved with integer programming

$$\min_{x} C^T x \text{ such that } \begin{cases} Ax \leq b \\ A_{eq}x = b_{eq} \\ x \text{ is binary} \end{cases}$$

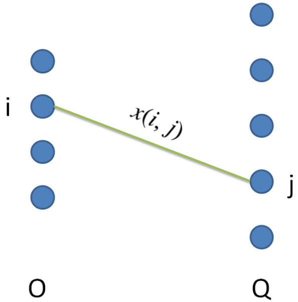
where $C(i,j) = \|p_O^i - p_Q^j\|_2$.

Data-aware 3D partitioning

- Linear approach is not geometrically consistent
- Let us introduce a geometric constraint for parts

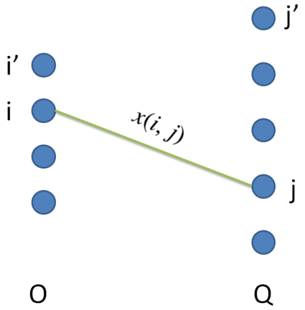
Data-aware 3D partitioning

- Matching



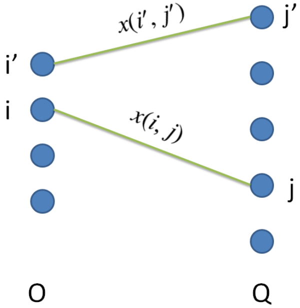
Data-aware 3D partitioning

- Matching



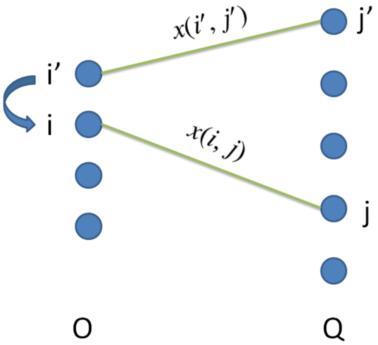
Data-aware 3D partitioning

- Matching



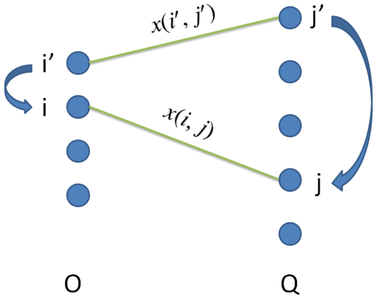
Data-aware 3D partitioning

- Matching



Data-aware 3D partitioning

- Matching



Data-aware 3D partitioning

- Quadratic programming

$$f(x) = \alpha \sum_{i,j,i',j'} |d_S^O(i,i') - d_S^Q(j,j')| x(i,j) x(i',j') + \beta \sum_{i,j} \|p_O^i - p_Q^j\|_2 . x(i,j)$$

- Now, we consider the inter-distance between parts

Data-aware 3D partitioning

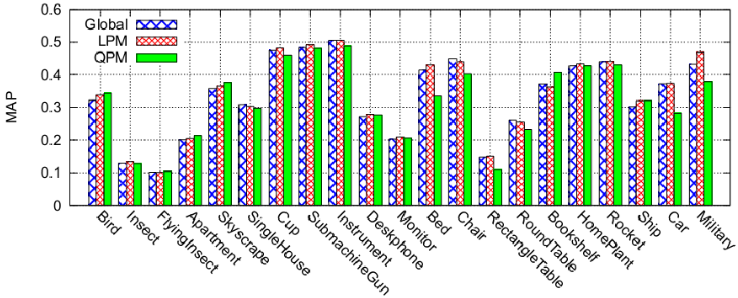
- Matching is solved with quadratic integer programming

$$\min_x \frac{1}{2}x^T Dx + C^T x \text{ such that } \begin{cases} Ax \leq b \\ A_{eq}x = b_{eq} \\ x \text{ is binary} \end{cases}$$

where $D(\{i,j\},\{i',j'\}) = |d_S^Q(i,i') - d_S^Q(j,j')|$.

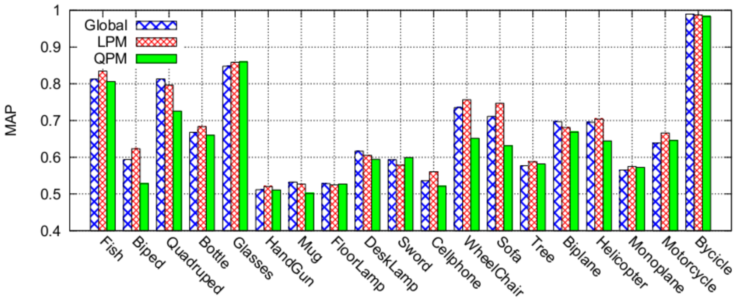
Data-aware 3D partitioning

- Class-by-class



Data-aware 3D partitioning

- Class-by-class



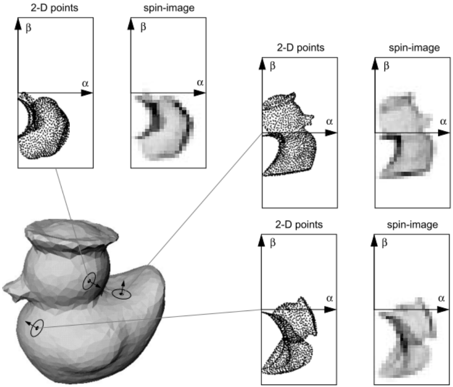
Data-aware 3D partitioning

- High variability inside classes
- Difficult problem for representations

Outline

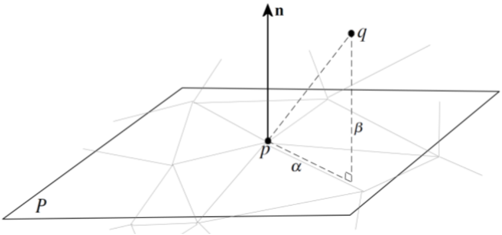
Spin images

- Robust local descriptor (Johnson 1997)
- It is based on how points are distributed on a surface



Spin images

- A local basis is constructed from
  - An oriented point *p*
  - The normal *n*
  - The tangent plane *P* through *p* and perpendicular to *n*



Spin images

- Any point q can be represented in this basis

$$S_O : \mathbb{R}^3 \rightarrow \mathbb{R}^2$$

$$S_O(q) \rightarrow (\alpha, \beta) = (\sqrt{\|q-p\|^2 - (\vec{n}\cdot(q-p))^2}, \vec{n}\cdot(q-p))$$

- The coordinate of q in the spin image is computed from (α, β)

Spin images

- Computing positions

$$i = \left\lfloor \frac{\frac{W * bin}{2} - \beta}{bin} \right\rfloor$$

$$j = \left\lfloor \frac{\alpha}{bin} \right\rfloor$$

Spin images

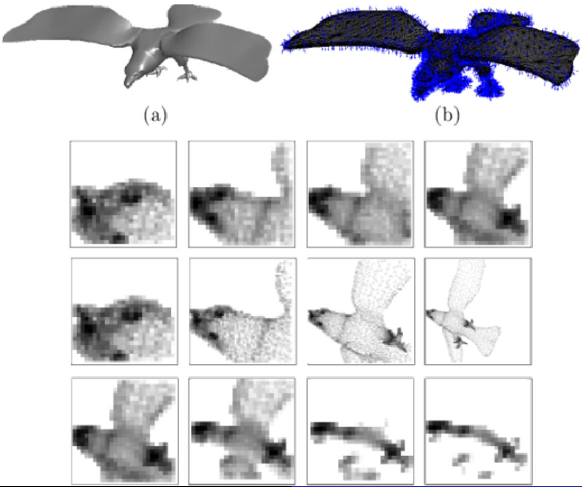
- Accumulation is performed using bilinear weights

$$I(i,j) = I(i,j) + (1-a)(1-b)$$
$$I(i,j+1) = I(i,j+1) + (1-a)b$$
$$I(i+1,j) = I(i+1,j) + a(1-b)$$
$$I(i+1,j+1) = I(i+1,j+1) + ab \quad (1)$$

where

$$a = \frac{\alpha}{bin} - j$$
$$b = \frac{\frac{W*bin}{2} - \beta}{bin} - i \quad (2)$$

Spin images



Demo 2: Spin images

Spin images

- Matching
  - Given two spin images with N bins, we compute the cross-correlation

$$R(P,Q) = \frac{N\sum p_i q_i - \sum p_i \sum q_i}{\sqrt{(N\sum p_i^2 - (\sum p_i)^2)(N\sum q_i^2 - (\sum q_i)^2)}}$$

  - Similarity takes into account the variance to avoid the dependency of cross-correlation to the overlap

$$C(P,Q) = (atanh(R(P,Q)))^2 - \lambda\left(\frac{1}{N-3}\right)$$

Spin images

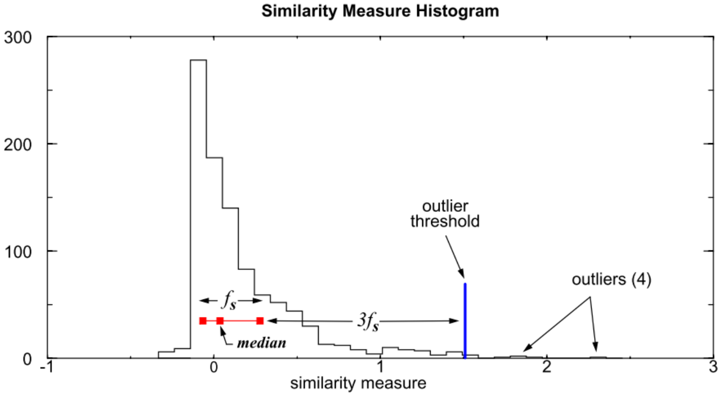
$C(P, Q)$ has a high value if two spin images are highly correlated and a large number of pixels overlap.

Spin images

- Matching
  - For each shape, a number of random spin images are computed and stored.
  - Given a spin image, the matching method computes the similarity to every stored spin images.
  - We only need to determine a set with the highest values (extreme outliers of the similarity histogram)

Spin images

Set of candidates



Spin images

- Filtering of correspondences
  - Correspondences with similarity less than the half of the maximum similarity
  - Given two correspondences $C_1 = (s_1, m_1)$ and $C_2 = (s_2, m_2)$, the geometric consistency is defined as

$$d_{gc}(C_1, C_2) = 2\frac{\|S_{m_2}(m_1) - S_{s_2}(s_1)\|}{\|S_{m_2}(m_1) + S_{s_2}(s_1)\|}$$

$$D_{gc}(C_1, C_2) = max(d_{gc}(C_1, C_2), d_{gc}(C_2, C_1))$$

where $S_O(p)$ denotes the spin map function of point p using the local basis of point O.

Spin images

- Filtering of correspondences
  - Geometric consistency involves position and normals.
  - D_{gc} is small if C_1 and C_2 are geometrically consistent.
  - Discard correspondences which are not consistent with at least a quarter of the complete list of correspondences.

Spin images

- Final step: searching a transformation
  - A group measure is defined

$$w_{gc}(C_1, C_2) = \frac{d_{gc}(C_1, C_2)}{1 - \exp(-(\|S_{m_2}(m_1)\| + \|S_{s_2}(s_1)\|)/2)}$$

$$W_{gc}(C_1, C_2) = max(w_{gc}(C_1, C_2), w_{gc}(C_2, C_1))$$

  - And a measure between a correspondence C and a group $\{C_1, C_2, \ldots, C_n\}$

$$W_{gc}(C, \{C_1, C_2, \ldots, C_n\}) = max_i(W_{gc}(C, C_i))$$

Spin images

Algorithm to generate groups

- For each correspondence $C_i \in L$, initialize a group $G_i = \{C_i\}$
- Find a correspondence $C_j \in L - G_i$, such that $W_{gc}(C_j, G_i)$ is minimum. If $W_{gc}(C_j, G_i) < T_{gc}$ then update $G_i = G_i \cup \{C_j\}$. T_{gc} is set between zero and one. If T_{gc} is small, only geometrically consistent correspondences remains. A commonly used value is 0.25.
- Continue until no more correspondences can be added.

Spin images

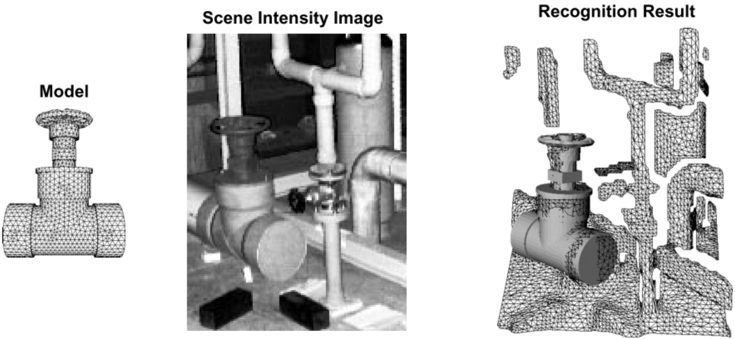
- The grouping algorithm generates n groups
- For each group of correspondences $\{(m_i, s_i)\}$ a rigid transformation T is calculated by minimizing the following error using least squares method

$$E_T = \min_T \sum \|s_i - T(m_i)\|^2$$

where $T(m_i) = R(m_i) + t$, R and t are the rotation matrix and the translation vector, representing the rotation and position of the viewpoint s_i in the coordinate system of m_i.

Spin images

A final step could involve an Iterative Closest Point algorithm for refinement.



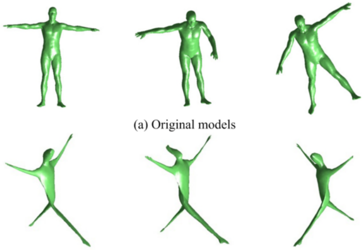
Outline

Non-rigid shapes

- Models with non-rigid transformations
- Several approaches
  - Canonical embedding
  - Spectral theory (Tutorial 4: Spectral geometry methods in shape analysis at 15:00)

Canonical embedding

- Find a canonical pose for models and compare them as in global matching (Elad and Kimmel 2003)
- Goal: "Unroll" the object
- Approach: Multi-dimensional scaling



(a) Original models

(b) Results of Least Square MDS

Canonical embedding

- MDS
  - Given a shape (X, d_X) where d_X is the geodesic distance

$$f : (X, d_X) \rightarrow (\mathbb{R}^m, d_{\mathbb{R}^m})$$

  - Map f converts points on the surface onto points in some Euclidean space
  - Hard to find an exact f.
  - In matching: as non-rigid shapes preserve geodesic distances, the embedding should be similar.

Canonical embedding

- MDS
  - Find a minimum-distortion embedding

$$f = \underset{f:X \rightarrow \mathbb{R}^m}{\arg\min} \sum_{i>j} \left| d_{\mathbb{R}^m}(f(x_i), f(x_j)) - d_X(x_i, x_j) \right|^2$$

  - The SMACOF algorithm is a gradient descent. It does not guarantee a global minimum

Demo 3: Canonical embedding

Introduction to Spectral Analysis

Heat diffusion on $\mathbb{R}^n$ is governed by the heat equation

$$\left(\Delta + \frac{\partial}{\partial t}\right) u(x;t) = 0; u(x;0) = u_0(x)$$

under some boundary condition.

- $u(x;t)$ is the heat distribution at point x at time t.
- $u_0(x)$ is the initial heat distribution
- Δ is the Laplacian

Introduction to Spectral Analysis

For a surface X, function u is defined on points of X, and the heat diffusion equation is

$$\left(\Delta_X + \frac{\partial}{\partial t}\right) u(x;t) = 0; u(x;0) = u_0(x)$$

- Δ_X is the Laplace-Beltrami operator

Introduction to Spectral Analysis

The Laplacian eigenvalue problem (the Helmholtz equation)

$$\Delta_X \phi = -\lambda \phi$$

where λ is an eigenvalue of the Laplacian, and ϕ is its corresponding eigenfunction.
Eigenfunctions are related to the Fourier basis functions.

Introduction to Spectral Analysis

The Laplace-Beltrami operator Δ_X has a discrete set of eigenvalues and eigenvectors.

$$\Delta_X \phi = \lambda \phi$$

where $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots$

- $\lambda_0 = 0$ and ϕ_0 constant if X has a boundary.
- Orthogonal eigenvectors

$$\phi_i . \phi_j = \int_X \phi_i \phi_j = 0, i \neq j$$

Introduction to Spectral Analysis

- Laplace-Beltrami operator Δ_X is invariant to isometric transformations
- Eigenvalues and eigenvectors are also invariant (Reuter 2006)

Reuter proposed to represent a non-rigid shape with a small set of eigenvalues: ShapeDNA

Demo 2: ShapeDNA

Shape Google

- Represent a 3D model as a quantized vector of spectral descriptors (Bronstein et al. 2010)
- The fundamental solution of heat equation is the heat kernel, represented as

$$K_t(x, y) = \sum_{i=0}^{\infty} \exp(-\lambda_i t) \vec{v}_i(x) \vec{v}_i(y)$$

where λ_i and $\vec{v}_i$ are the eigenvalues and eigenvectors of the Laplace-Beltrami operator, respectively.

Shape Google

- A representation for a point can be obtained (Sun et al. 2009)

$$K_t(x,x) = \sum_{i=0}^{\infty} \exp(-\lambda_i t)\vec{v}_i(x)^2$$

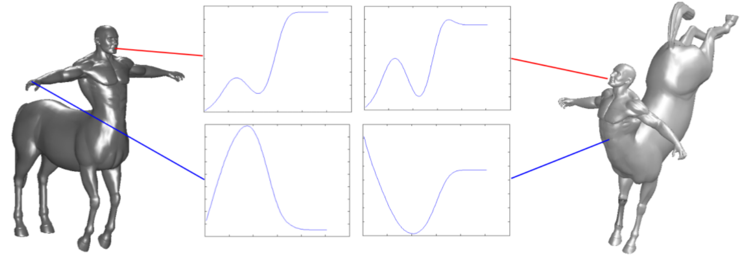
- Using values for t, we can get a descriptor which is called Heat Kernel Signature

$$p(x) = (p_1(x), \ldots, p_n(x))$$

$$p_i(x) = c(x)K_{\alpha^{i-1}t_0}(x,x)$$

Shape Google

- Heat Kernel Signatures (Bustos and Sipiran, 2012)



Shape Google

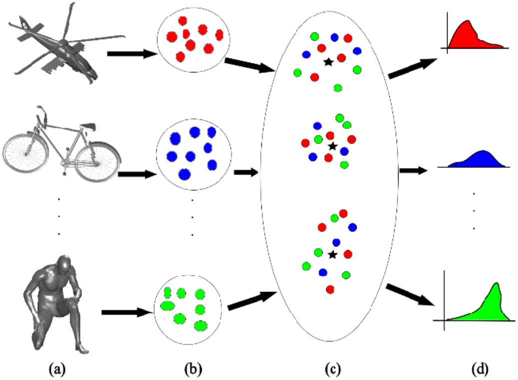
- Heat kernel signatures are sensitive to scale
- A scale-invariant variant has been proposed (Bronstein and Kokkinos, 2010)
  - It uses discrete derivatives and Fourier coefficients for removing the scale dependency of HKS

Shape Google

- Procedure
  - Compute a descriptor for each vertex in a mesh
  - Given the entire collection of descriptors, perform a k-means clustering to find a dictionary
  - Quantize the descriptors of a shape using the dictionary

Shape Google

- Process



Shape Google

- Apply k-means clustering to find a dictionary $M = \{m_1, m_2, \ldots, m_k\}$
- For each point x on a mesh with its descriptor $p(x)$, the feature distribution

$$\theta(x) = (\theta_1(x), \ldots, \theta_k(x))^T$$

is a vector with elements

$$\theta_i(x) = c(x) \exp\left(\frac{-\|p(x) - m_i\|_2}{2\sigma^2}\right)$$

Shape Google

- The Bag of Feature of a shape S is

$$f(S) = \sum_{x \in S} \theta(x)$$

- The distance between two shapes S and T is

$$d(S, T) = \|f(S) - f(T)\|$$

Shape Google

- Spatial information is lost during the quantization process
- Consider pairs of descriptors with a weighting factor

$$F(S) = \sum_{x \in S} \sum_{y \in S} \theta(x)\theta^T(y)K_t(x, y)$$

- $F(S)$ is a matrix.

Signature Quadratic Form Distance for Retrieval

- Unlike bag of features, this approach is local for defining the signatures
- Signature Quadratic Form Distance (Beecks et al. 2010)
  - Final representation only depends on the object information
  - It is possible to measure the distance between objects with representations of different sizes.

SQFD for Retrieval

- Object is represented as a set of features

$$F = \{f_i\}$$

- Let us suppose the existence of a local partitioning

$$F : C_1, \ldots, C_n$$

- The signature is defined as

$$S^P = \{(c_i^P, w_i^P), i = 1, \ldots, n\}$$

where $c_i^P = \frac{\sum_{f \in C_i} f}{|C_i|}$ and $w_i^P = \frac{|C_i|}{K}$ represent the centroid of i-th cluster and a weight, respectively.

SQFD for Retrieval

- Given two signatures

$$S^P = \{(c_i^P, w_i^P), i = 1, \ldots, n\}$$

$$S^Q = \{(c_j^Q, w_j^Q), j = 1, \ldots, m\}$$

- SQFD is defined as

$$SQFD_{f_S}(S^P, S^Q) = \sqrt{(w^P| - w^Q) \cdot A_{f_S} \cdot (w^P| - w^Q)^T}$$

SQFD for retrieval

- $A_{f_S} \in R^{(n+m)\times(n+m)}$ is the similarity matrix defined as

$$a_{ij} = \begin{cases} f_S(c_i^P, c_j^P) & \text{if } i \leq n \text{ and } j \leq m \\ f_S(c_{i-n}^Q, c_j^P) & \text{if } i > n \text{ and } j \leq m \\ f_S(c_i^P, c_{j-n}^Q) & \text{if } i \leq n \text{ and } j > m \\ f_S(c_{i-n}^Q, c_{j-n}^Q) & \text{if } i > n \text{ and } j > m \end{cases}$$

- The similarity function f_S can be
  - Minus: $f\_(c_i, c_j) = -d(c_i, c_j)$
  - Gaussian: $f_g(c_i, c_j) = exp(-\alpha d^2(c_i, c_j))$
  - Heuristic: $f_h(c_i, c_j) = \frac{1}{\alpha + d(c_i, c_j)}$

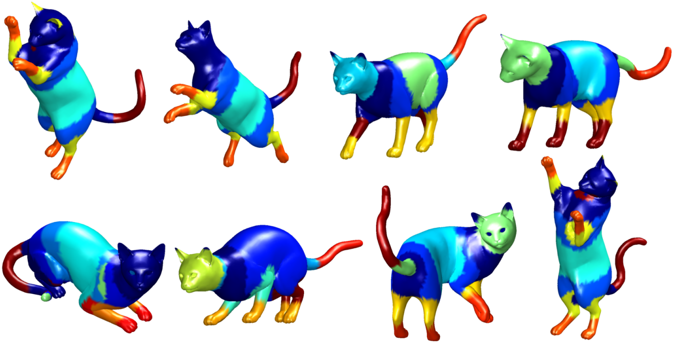
SQFD for Retrieval

- Three approaches for computing the signatures in 3D meshes
  - All descriptors of an objects
  - Descriptors of keypoints
  - Geodesic clusters
- Adaptive clustering for computing the local partitioning
- We use the Heat Kernel Signatures as descriptors

SQFD for Retrieval

- All vertices

$$FS(S) = \left\{ \frac{hks(v_i)}{\|hks(v_i)\|} | v_i \in S, i = 1, \ldots, n \right\}$$

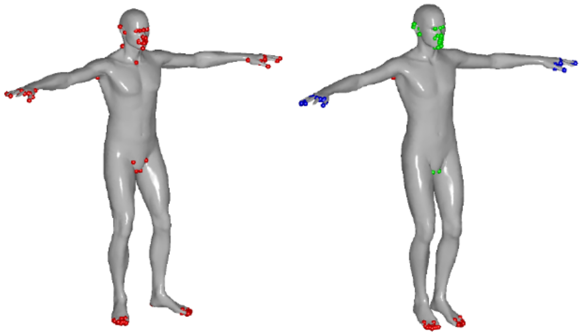


Demo 5: Signatures with all vertices

SQFD for Retrieval

- Descriptors of keypoints

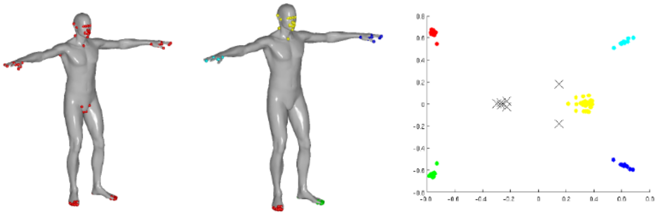
$$FS_{IP}(P) = \left\{ \frac{hks(v)}{\|hks(v)\|} | v \in IP(P) \right\}$$



Demo 6: Signatures with keypoints

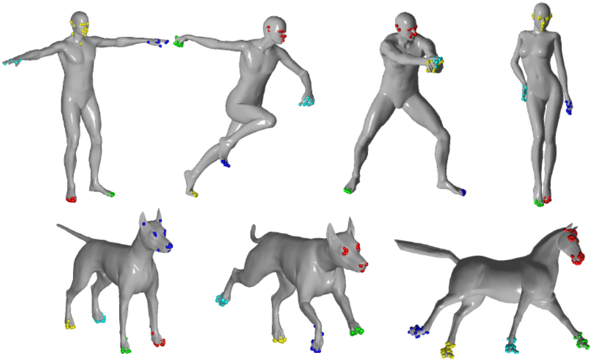
SQFD for Retrieval

- Geodesic clusters
  - Compute the MDS of the keypoints in $\mathbb{R}^2$
  - Perform an adaptive clustering
  - One signature for each geodesic cluster



SQFD for Retrieval

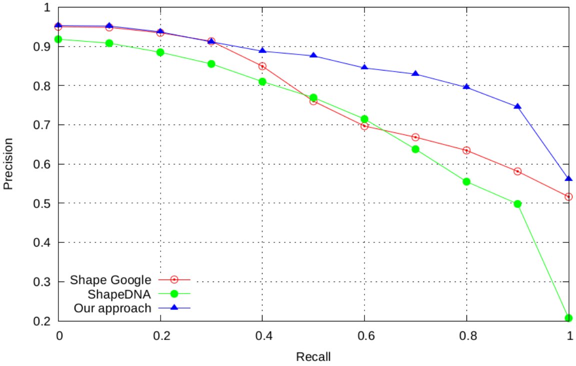
- Examples of geodesic clusters



Demo 7: Signatures with geodesic clusters

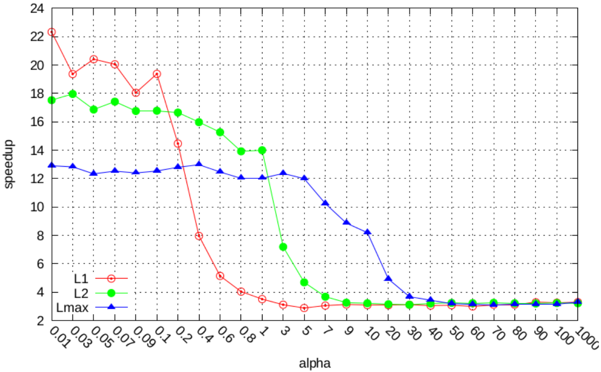
SQFD for Retrieval

- To evaluate this approach, we built a dataset with 5604 models



SQFD for Retrieval

- SQFD is indexable with metric access methods
- We used pivot tables to avoid the linear scan



SQFD for Retrieval

- SQFD is indexable with metric access methods
- We used pivot tables to avoid the linear scan

Method	Query time
ShapeDNA	0.01
Shape Google	0.1330
Total	0.9479
Keypoint	0.0252
Cluster	1.1842

Other approaches

- A recent comparison evaluated state-of-the-art methods (Lian et al. 2013)

Shape Retrieval Contests (SHREC)

- Competitions started in 2006
- To date: 40+ tracks presented
- Each track has a dataset and evaluation tools
- Good initiative to evaluate algorithms and make comparisons with the state of the art

SHREC Examples

- CAD models (2008)[1]
  - Using the ESB benchmark

Stats for all run files

Mean Average Precision(relevant)		
Rank	RunFile	Value
1	U_Yama_A Run 1	0.79965734
2	U_Yama_B Run 1	0.47644576
3	TNepolean Run 2	0.4330686
4	TNepolean Run 1	0.39915675
5	Asim Run 1	0.35762513
6	X_Li Run 1	0.32791287

Mean First Tier(relevant)		
Rank	RunFile	Value
1	U_Yama_A Run 1	78.166664%
2	U_Yama_B Run 1	44.282406%
3	TNepolean Run 2	41.62454%
4	TNepolean Run 1	38.54358%
5	Asim Run 1	32.961426%
6	X_Li Run 1	31.491108%

Mean Second Tier(relevant)		
Rank	RunFile	Value
1	U_Yama_A Run 1	39.739418%
2	U_Yama_B Run 1	27.32075%
3	TNepolean Run 2	26.433836%
4	TNepolean Run 1	24.579538%
5	Asim Run 1	21.038078%
6	X_Li Run 1	19.132729%

Mean Dynamic Average Recall		
Rank	RunFile	Value
1	U_Yama_A Run 1	0.7943641
2	U_Yama_B Run 1	0.5675939
3	TNepolean Run 2	0.50698084
4	TNepolean Run 1	0.47764158
5	Asim Run 1	0.44011146
6	X_Li Run 1	0.42887306

Mean Normalized Cumulated Gain @5		
Rank	RunFile	Value
1	U_Yama_A Run 1	0.79111105
2	U_Yama_B Run 1	0.6577777
3	TNepolean Run 2	0.5555556
4	TNepolean Run 1	0.52
5	Asim Run 1	0.49777776
6	X_Li Run 1	0.48888898

Mean Normalized Cumulated Gain @10		
Rank	RunFile	Value
1	U_Yama_A Run 1	0.7883333
2	U_Yama_B Run 1	0.51199293
3	TNepolean Run 2	0.48215166
4	TNepolean Run 1	0.44985005
5	Asim Run 1	0.4206437
6	X_Li Run 1	0.39493832

Mean Normalized Cumulated Gain @25		
Rank	RunFile	Value
1	U_Yama_A Run 1	0.7886773
2	U_Yama_B Run 1	0.50129616
3	TNepolean Run 2	0.4817584
4	TNepolean Run 1	0.4206269
5	Asim Run 1	0.409007
6	X_Li Run 1	0.3634691

Mean Normalized Cumulated Gain @50		
Rank	RunFile	Value
1	U_Yama_A Run 1	0.82316583
2	U_Yama_B Run 1	0.5586469
3	TNepolean Run 2	0.5507752
4	TNepolean Run 1	0.5017806
5	Asim Run 1	0.4520872
6	X_Li Run 1	0.40770996

Mean Normalized Cumulated Gain @100		
Rank	RunFile	Value
1	U_Yama_A Run 1	0.87145066
2	TNepolean Run 2	0.67413086
3	TNepolean Run 1	0.6479456
4	U_Yama_B Run 1	0.64153767
5	Asim Run 1	0.5727366
6	X_Li Run 1	0.48382616

[1] Available in: https://engineering.purdue.edu/PRECISE/shrec08

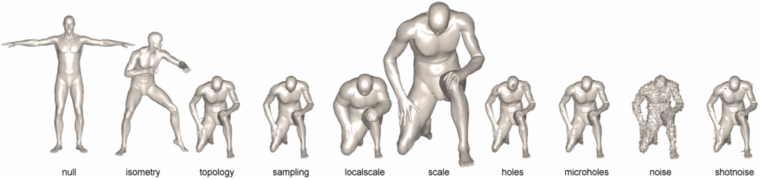
SHREC Examples

- Generic shape retrieval (2009)[2]
  - 720 objects organized in 40 classes, 22 algorithms evaluated

[2] http://www.itl.nist.gov/iad/vug/sharp/benchmark/shrecGeneric/

SHREC Examples

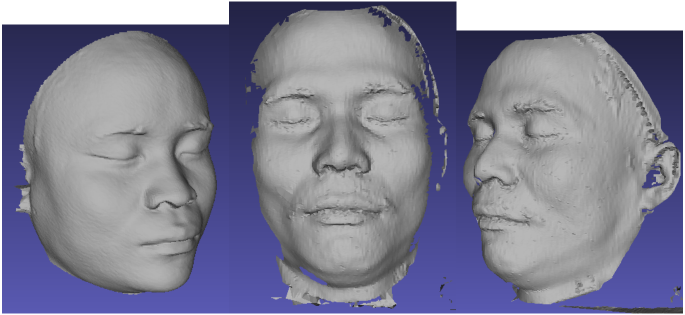
- Feature detection and description (2010)[3]
  - Three shapes, 9 transformations in 5 levels of strength.
  - Goal: measure the repeatability of local features



[3]Available in: http://tosca.cs.technion.ac.il/book/shrec_feat2010.html

SHREC Examples

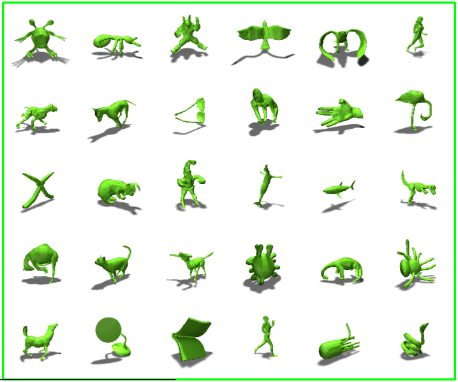
- Face scans (2010) [4]
  - Training set: 60 models
  - Test set: 650 scans



[4] Available in: http://give-lab.cs.uu.nl/SHREC/shrec2011/faces/index.php

SHREC Examples

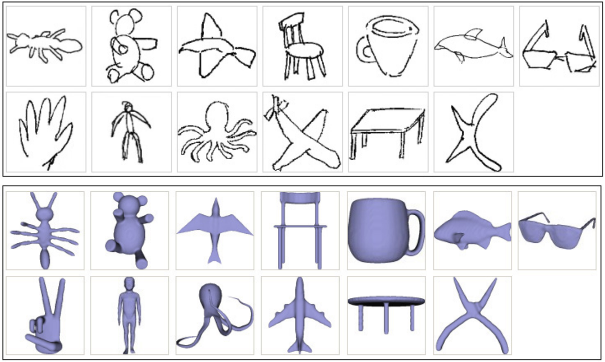
- Non-rigid retrieval (2011) [5]
  - 600 objects with non-rigid transformations



[5] Available in: http://www.itl.nist.gov/iad/vug/sharp/contest/2011/NonRigid/

SHREC Examples

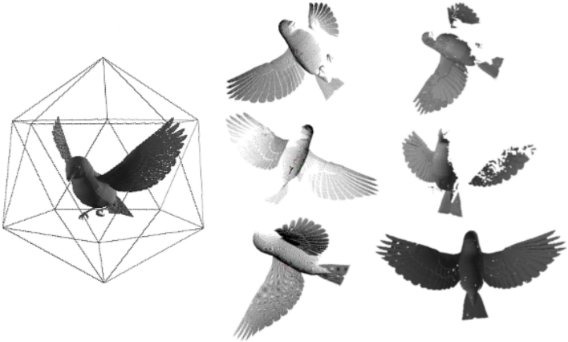
- Sketch-based 3D models retrieval (2012) [6]
    - 400 3D models, 250 hand-drawn sketches



[6] Available in: http://www.itl.nist.gov/iad/vug/sharp/contest/2012/SBR/

SHREC Examples

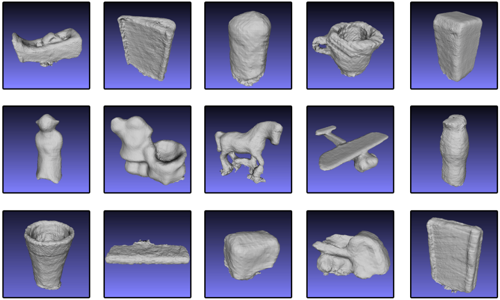
- Large-scale partial shape retrieval (2013) [7]
  - 360 models, 7200 partial queries



[7] Available in: http://dataset.dcc.uchile.cl/

SHREC Examples

- Retrieval of Objects Captured with Low-Cost Depth-Sensing Cameras (2013) [8]
  - 192 models captured with Kinect



[8] Available in:http://3dorus.ist.utl.pt/research/BeKi/index.html

Final remarks

- Good balance of theory and practice in solutions
- Current methods will be useful tools for supporting the emergence of massive 3D data
- There is still room for improvements (efficiency, scalability, robustness)

Future trends

- Not-so-local features
- How to deal with missing data? For instance, due to occlusions
- Representations: point clouds

Tutorial material

- Demo's shapes belongs to the TOSCA dataset (A. Bronstein and M. Bronstein and R. Kimmel, 2008)
- Slides and matlab codes will be available soon on http://users.dcc.uchile.cl/ isipiran/

Thank You

Thank You!
Questions please