ICPR 2012 Tutorial

Part III. Non-Rigid 3D Shapes
Manifold-based Analysis of Deformable Shapes

Hamid Laga
University of South Australia

Ryutarou Ohbuchi
University of Yamanashi
http://www.kki.yamanashi.ac.jp/~ohbuchi
Shape = intrinsic (geometry) x extrinsic (bending)

Images courtesy of [Sumner et al. 2004]
In this talk: pose-invariant shape analysis

A similarity measure that is invariant to the shape pose
Use of properties that do not change with pose

Geodesic distances do not change (that much) with pose

Images courtesy of Bronstein et al.
Sample $N$ uniformly distributed points $P = \{p_1, ..., p_N\}$ on the shape surface

- **Rigid shapes**
  - Compute pairwise Euclidean distances
  - Make a histogram of $B$ bins
- **Invariance properties**
  - Translation, rotation
  - Scale (after normalization)

- **Non-rigid shapes**
  - Compute pairwise Geodesic distances
  - Make a histogram of $B$ bins
- **Invariance properties**
  - Translation, rotation
  - Scale (after normalization)
  - Bending.
Shape distribution revisited

Using Euclidean distances (captures the extrinsic geometry)

Using geodesic distances (captures the intrinsic geometry)
Metric spaces and canonical forms

Isometric embedding

Shape as a metric space

Canonical form

\[ d_X(x_1, x_2) = d_{R^3}(\varphi(x_1), \varphi(x_2)) \]
Intrinsic similarity = Extrinsic similarity between the canonical forms

\[ \varphi(X) \overset{?}{=} \psi(Y) \]

\[ d_{int}(X, Y) = d_{ext}(\varphi(X), \psi(Y)) \]
Analysis using the canonical forms

- **Input**
  - Two shapes $X$ and $Y$
    (Let $d_X$ the geodesic distances and $d_{R^3}$ the Euclidean distances)

- **Algorithm**
  - Sample $n$ points on $X$ and $n$ points on $Y$
  - Minimum distortion embedding

\[
\min_{\varphi: X \rightarrow R^3} \sum_{i,j} \left\| d_X(x_i, x_j) - d_{R^3}(\varphi(x_i), \varphi(x_j)) \right\|^2
\]

- Analyze the two shapes $X$ and $Y$ in the new space
  (i.e., by analyzing their embeddings)
Analysis using the canonical forms

Shape $X + a$ metric $d$

Shape $Y + a$ metric $d$

Normalization

Canonical form of $X$

Canonical form of $Y$

Descriptors

$\equiv$

Similar / not similar
Multi Dimensional Scaling (MDS)

- MDS is a distance-preserving embedding
  - Compute the $n \times n$ Gram matrix $G_X$

\[
G_X = \begin{pmatrix}
G_X \\
(n \times n)
\end{pmatrix}
= \begin{pmatrix}
X^T \\
\vdots \\
\end{pmatrix}
= \begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{pmatrix}
\]

- SVD decomposition of $G_X$

\[
G_X = VS^2V^T
\]

- Canonical form in $\mathbb{R}^d$ is computed using the first $d$ principal directions (largest eigenvectors) as

\[
Z^* = (VS)^T
\]
MDS results

Near-isometric deformations of a shape

Canonical forms (MDS-based embedding in $\mathbb{R}^3$)

Images courtesy of A. Bronstein
Spectral embedding (Jain & Zhang 07)

Input data → Encode information about pairwise affinities → Operator $A$ → Leading eigenvectors

Slide courtesy of [Levy and Zhang’10]
Spectral embedding (Jain & Zhang 07)

Fig. 2. Spectral embeddings (bottom row) of some articulated 3D shapes (top row) from the McGill shape database. The embeddings are constructed using the second, third, and fourth eigenvectors.
Applications of spectral embedding

Kmeans in the embedding domain

- (a) 3 parts.
- (b) 5 parts.
- (c) 7 parts.
- (d) 9 parts.

Mesh segmentation

Descriptors in the embedding domain

Non-rigid (isometric) shape retrieval

ICP in the embedding domain

Correspondences (isometric deformations)
Is geodesic distance the right metric?

<table>
<thead>
<tr>
<th></th>
<th>Rigid</th>
<th>Scale</th>
<th>Isometric</th>
<th>Topology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Euclidean</strong></td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Geodesic</strong></td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>
Diffusion geometry

Euclidean

Geodesic

Diffusion
Heat equation: heat propagation on a manifold $X$

\[ (\Delta_X + \frac{\partial}{\partial t}) u = 0 \]

- **Initial conditions:**
  heat distribution at time $t = 0$.

- **Boundary condition** (if $X$ has a boundary)

- **Heat Kernel** $h_t(x, y)$ is a fundamental solution of the heat equation with point heat source $x$ (heat value at point $y$ after time $t$)
Spectra of the Laplace-Beltrami

• Heat Kernel: heat value at point $y$ after time $t$
  – Can be represented in the Laplace-Beltrami eigenbasis
    \[
    h_t(x, y) = \sum_{i=0}^{\infty} e^{-t\lambda_i} \phi_i(x)\phi_i(y)
    \]

  \(\{\phi_i, \lambda_i\}_{i \geq 1}\) are eigenvalues and eigenfunctions of the Laplace-Beltrami operator \(\Delta \phi = \lambda \phi\)

  \[
  \lambda_0 = 0 < \lambda_1 < \lambda_2 < \ldots < \lambda_i < \ldots
  \]

  \[
  \langle \phi_i, \phi_i \rangle = 1, \langle \phi_i, \phi_j \rangle = 0
  \]

• Embedding using the Laplacian-Beltrami eigenfunctions
  – Eigenfunctions \(\{\phi_i\}_{i \geq 1}\) have global nature
  – More stability to local changes
  – The eigenvalues and eigenfunctions are isometry invariant
Laplace-Beltrami on triangulated surfaces

- **Laplacian**  
  \( L = (a_{ij}) \)  
  - \( i, j = 1 \ldots N \)  
  - \( N \): number of vertices

- **Discrete Graph Laplacian**  
  - uniform weights \( w_{ij} = 1 \)  
  - Discretized Laplacian

\[
w_{ij} = \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{\sqrt{A(v_i)A(v_j)}}
\]  

\[
a_{i,j} = \begin{cases} 
  w_{i,j} > 0 & \text{if } x_j \in N(x_i) \\
  -\sum_j w_{i,j} & \text{otherwise}
\end{cases}
\]
Level sets of an eigenfunction of the Laplacian

Discrete graph Laplacian

Discretized Laplace-Beltrami

Image courtesy of [Levy and Zhang’10]
First eigenfunctions of the Laplace-Beltrami
Invariance to isometries

An eigenfunction of the Laplace-Beltrami operator computed on different deformations of the shape, showing the invariance of the Laplace-Beltrami operator to isometries

[Bronstein et al.]
Shape DNA (Reuter et al. 06)

Laplace-Beltrami spectrum $\lambda_i$, $i \geq 1$
as an isometry invariant shape signature

Laplace-Beltrami spectrum

MDS plot of the mesh Shape-DNA
Diffusion Kernels

Probabilistic interpretation

- Brownian motion on $X$ starting at $x$, measurable at $C$.

$$\int_C k_t(x, z) d\mu(z) = \text{probability of the Brownian motion to be in } C \text{ at time } t$$

Diffusion kernel can be parameterized by a transfer function $K(\lambda)$

- Transition probability from $x$ to $y$ in one step

$$k(x, y) = \sum_{i=0}^{\infty} K(\lambda_i) \phi_i(x) \phi_i(y)$$

- Transition probability from $x$ to $y$ in $t$ steps

$$k_t(x, y) = \sum_{i \geq 0} K^t(\lambda_i) \phi_i(x) \phi_i(y)$$
Diffusion maps

\[ \Phi : x \rightarrow \{ K(\lambda_i)\phi_i(x) \} \in \ell^2 \]

\[ d^2(x, y) = \| \Phi(x) - \Phi(y) \|_{\ell^2}^2 = \sum_{i>0} K^2(\lambda_i)(\phi_i(x) - \phi_i(y))^2 \]
Heat Kernel-based metrics

**Heat Diffusion metric**

\[ K(\lambda) = e^{-t\lambda} \]

\[ d_t^2(x, y) = \sum_{i=0}^{\infty} e^{-2\lambda_i t} (\phi_i(x) - \phi_i(y))^2 \]

Connectivity by random walks of length \( t \)

**Heat Kernel Signature (HKS)**

\[ \Phi_t : x \rightarrow \left\{ e^{-t\lambda_1} \phi_1(x), e^{-t\lambda_2} \phi_2(x), \ldots \right\} \]

**Commute time metric**

\[ K(\lambda) = \frac{1}{\sqrt{\lambda}} \]

\[ d_{CT}^2(x, y) = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} (\phi_i(x) - \phi_i(y))^2 \]

Connectivity by random walks of any length

**Global Point Signature (GPS)**

\[ \Phi : x \rightarrow \left\{ \frac{1}{\sqrt{\lambda_1}} \phi_1(x), \frac{1}{\sqrt{\lambda_2}} \phi_2(x), \ldots \right\} \]
Invariance properties

<table>
<thead>
<tr>
<th></th>
<th>Rigid</th>
<th>Scale</th>
<th>Isometric</th>
<th>Topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Geodesic</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Heat diffusion</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Commute time</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
Can points on the surface be characterized intrinsically?

- Use histogram of distances in the GPS embeddings

$$GPS(p) = \left( \frac{1}{\sqrt{\lambda_1}} \phi_1(p), \frac{1}{\sqrt{\lambda_2}} \phi_2(p), \frac{1}{\sqrt{\lambda_3}} \phi_3(p), \ldots \right)$$

- $\phi(p)$ is the value of the eigenfunction at point $p$
- Invariance properties reflected in GPS embeddings
- Less sensitive to topology changes by using only low-frequency eigenfunctions
- Sign flips and eigenvector switching are still issues
Applications of Global Point Signatures

Non-rigid shape clustering

Segmentation (K-means in the GPS)
Heat Kernel Signatures (Sun et al. 09)

- Diagonal of the Heat Kernel

\[ k_t(x, y) = \sum_i e^{-\lambda_i t} \phi_i(x) \phi_i(y) \]

\[ \Rightarrow \ HKS_x(t) = k_t(x, x) = \sum_i e^{-\lambda_i t} \phi_i^2(x) \]

Multi-scale point descriptor

\[ p(x) = (h_t(x, x), \ldots, h_{at}^n(x, x)) \]

\[ h_t(x, x) \quad h_{at}(x, x) \quad h_{a^2 t}(x, x) \]
Heat Kernel Signatures (Sun et al. 09)

Invariant to isometric deformations

Localized sensitivity to topological noise
Applications of HKS (Sun et al. 09)

Feature selection based on the maxima of $k_t(x, x)$

Multiscale matching

Feature selection based on the maxima of $k_t(x, x)$

Multiscale matching

scaled HKS: $\frac{k_t(x, x)}{\int_M k_t(x, x) dx}$
Applications of HKS (Sun et al. 09)

Multiscale matching

(a) maxima of $k_t(x, x)$ for a fixed $t$.  
(b) $t = [0.1, 4]$  
(c) $t = [0.1, 80]$
Spatially-sensitive bag of features

HKS descriptors

Geometric words

Bag of Features

Geometric expressions

Bag of Expressions

Shape Google (Bronstein et al. 11)
• Vocabulary $\mathcal{P} = \{p_1, \ldots, p_V\}$ of size $V$
  – Compute the HKS $p(x)$ at different locations $x$ on the shapes
  – Quantize the descriptor space.

• Describing a feature point $x$

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

$$\theta_i(x) = c(x)e^{-\frac{\|p(x) - p_i\|^2}{2\sigma^2}},$$

• Spatial relationships

$$F(X) = \int_{x \times x} \theta(x)\theta^T(y)K_t(x, y)d\mu(x)d\mu(y).$$

Input: Query shape $X$; geometric vocabulary $\mathcal{P} = \{p_1, \ldots, p_V\}$, database of shapes $\{X_1, \ldots, X_D\}$.

Output: Shapes from $\{X_1, \ldots, X_D\}$ most similar to $X$.

1. if Feature detection then
2. | Compute a set $X'$ of stable feature points on $X$.
3. else
4. | $X' = X$.
5. Compute local feature descriptor $p(x)$ for all $x \in X'$.
6. Quantize the local feature descriptor $p(x)$ in the vocabulary $\mathcal{P}$, obtaining for each $x$ a distribution $\theta(x) \in \mathbb{R}^V$.
7. if Geometric expressions then
8. | Compute a spatially-sensitive $V \times V$ bag of geometric words
9. | $F(X) = \int_{x \times x} \theta(x)\theta^T(y)K_t(x, y)d\mu(x)d\mu(y)$,
10. and parse it into a $V^2 \times 1$ vector $f(X)$.
11. else
12. | Compute a $V \times 1$ bag of geometric words
13. | $f(X) = \int_X \theta(x)d\mu(x)$.
Spatially sensitive bag of features
1237 shapes from public domain
TOSCA shapes, Robert Sumner's shapes, and Princeton shapes

60 transformations per shape
Performance metrics (Bronstein et al. 10)

• **Task**
  – Match transformed shapes to a database of (469) untransformed ones
  – Each transformation contains $n$ strengths

• **Metrics**
  – Precision $P(r)$/ recall :
    • $P(r)$ : percentage of relevant shapes in the first $r$ top-ranked retrieval results
  – Mean average precision $mAP$

$$mAP = \sum_{r} P(r) \cdot rel(r),$$
Evaluated methods (Bronstein et al. 10)

- **Visual similarity**
  - Clock Matching Bag of Words (CM-BOF)
  - Geodesic Sphere-based Multiview Descriptor (GSMD)
- **Part-based BoW (PB)**
- **Shape Google**
  - SG1: HKS (using cotangent weights)
  - SG2: HKS (using FEM)
  - SG3: Scale Invariant HKS (SI-HKS)
- **Similarity Sensitive Hashing SS1**
  - Extension of Shape Google
  - Bag of Features are embedded into a Hamming space (shapes represented as binary codes)
Evaluation results

<table>
<thead>
<tr>
<th>Transform</th>
<th>Strength ≤ 1</th>
<th>Strength ≤ 3</th>
<th>Strength ≤ 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isometry</td>
<td>VS2, PB1, SG1–3, SS1</td>
<td>SG1–3, SS1</td>
<td>SG1–3, SS1</td>
</tr>
<tr>
<td>Topology</td>
<td>VS2, PB1–3, SG1–2, SS1</td>
<td>VS2, SS1</td>
<td>VS2, SS1</td>
</tr>
<tr>
<td>Holes</td>
<td>VS2, PB1, SG1–3, SS1</td>
<td>VS2, SG2–3, SS1</td>
<td>VS2, SS1</td>
</tr>
<tr>
<td>Micro holes</td>
<td>VS2, PB1–2, SG1–3, SS1</td>
<td>VS2, PB1, SG1–3, SS1</td>
<td>VS2, PB1, SG1–3, SS1</td>
</tr>
<tr>
<td>Scale</td>
<td>VS2, PB1, SS1</td>
<td>VS2, PB1, SS1</td>
<td>VS2, PB1, SS1</td>
</tr>
<tr>
<td>Local scale</td>
<td>VS1–2, PB1–2, SG1–3, SS1</td>
<td>SS1</td>
<td>SS1</td>
</tr>
<tr>
<td>Sampling</td>
<td>VS2, PB2, SG1–3, SS1</td>
<td>VS2, SG1–3, SS1</td>
<td>VS2, SG2</td>
</tr>
<tr>
<td>Noise</td>
<td>VS2, PB1–2, SG1–3, SS1</td>
<td>VS2, SG1–3, SS1</td>
<td>VS2, SG1–3, SS1</td>
</tr>
<tr>
<td>Shot noise</td>
<td>VS2, SG1–3, SS1</td>
<td>VS2, SG1–3, SS1</td>
<td>SG1–3, SS1</td>
</tr>
<tr>
<td>Partial</td>
<td>SS1</td>
<td>SS1</td>
<td>SS1</td>
</tr>
<tr>
<td>Mixed</td>
<td>VS2</td>
<td>VS2</td>
<td>VS2</td>
</tr>
<tr>
<td>Average</td>
<td>SS1</td>
<td>SS1</td>
<td>SS1</td>
</tr>
</tbody>
</table>

Table 11: Winning algorithms across transformation classes and strengths. VS1 = CM-BOF+MMR, VS2 = CM-BOF, VS3 = GSMD, PB1 = part-based bag of words with large number of visual words, PB2 = part-based bag of words with visual vocabulary computed from the training set, PB3 = part-based bag of words with visual vocabulary computed from the test set, SG1 = ShapeGoogle with HKS descriptor using cotangent weights, SG2 = ShapeGoogle with HKS descriptor using FEM, SG3 = ShapeGoogle with SI-HKS descriptor using cotangent weights, SS1 = ShapeGoogle with SI-HKS descriptor and similarity-sensitive hashing.
There is no absolute winner

• **SHREC’10 robust large-scale shape retrieval benchmark**
  
  – Different methods showed different performance across transformation classes.
  
  – On the average, ShapeGoogle using SI-HKS local descriptor and similarity sensitive hashing (SS1) showed the best performance (98:27% mAP on the full query set, second place CM-BOF with 94:33% mAP, third place SG3 with 90:79% mAP).
  
  – SS1 was also among the best in all transformation classes excepting sampling and mixed transformation.
  
  – CM-BOF and ShapeGoogle using HKS local descriptor computed with FEM discretization (SG2) showed the best robustness to sampling change.
  
  – CM-BOF showed the best performance in mixed transformations class.
  
  – The ShapeGoogle framework showed significantly better robustness to non-rigid deformations compared to other methods.
We need deformation-invariant similarity measures

Summary
What we didn’t cover in this talk

• Elastic shape analysis
  – Riemannian geometry,
  – shape spaces, and shape statistics

Continuous variability in 3D shape collections
Limitations and open problems

• Non-manifold surfaces
  – How about polygon soups, range scans, point set that undergo non-rigid deformations?

• Partial similarity

• Elastic deformations

• How about the semantics?
• Pose (bending)-invariant shape analysis
References

- **Others**
  - http://www.cs.jhu.edu/~misha/Fall07/Notes/Rustamov07.pdf

- **Datasets**
  - TOSCA dataset: http://tosca.cs.technion.ac.il/
  - Shape Retrieval Evaluation Context (SHREC)

- **Source codes**
  - TOSCA: http://tosca.cs.technion.ac.il/
Tutorial outline

• Part I. Rigid 3D shape analysis
  3D representation, invariance properties, similarity measures

• Part II. Non-rigid 3D shape analysis
  Local geometric features, deformable shapes, shape statistics

• Part III. Learning
  for better features, to capture semantics and intention

• Part IV. Querying 3D shape databases
  By text, 3D examples, (depth) images, sketches, context

• Part V. Applications
  Retrieval, classification, archaeology, biomedical, modeling