2D Shape Matching: 
Inner Distance + Shape Context

Presented by Chen Feng

Dept. of Civil & Environmental Engineering @ University of Michigan

October 14, 2010
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
Problem Domain

- 2D shape recognition

Example

- handwritten recognition
Problem Domain

- 2D shape recognition
- 2D shape searching

Example

```
query  1: 0.086  2: 0.108  3: 0.109
```
Problem Domain

- 2D shape recognition
- 2D shape searching
- 2D shape classification etc.
Major Difficulties

- deformation

Example
Major Difficulties

- deformation
- articulation

Example
Related Works

Two major approaches:

- Feature-Based Methods
  using spatial arrangements of extracted features (edges, junctions)
Related Works

Two major approaches:

- **Feature-Based Methods**
  using spatial arrangements of extracted features (edges, junctions)

- **Brightness-Based Methods**
  directly using pixel brightness
Related Works

Two major approaches:

- **Feature-Based Methods**
  using spatial arrangements of extracted features (edges, junctions)

- **Brightness-Based Methods**
  directly using pixel brightness
Outline

1 Background

2 Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks

3 Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity

4 Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context

5 Experiments
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
How to define a shape?

Definition (Shape)

A shape is represented as a sequence of boundary points:

\[ P = \{p_1, \cdots, p_n\}, \quad p_i \in \mathbb{R}^2 \]
How to define a shape?

**Definition (Shape)**

A shape is represented as a sequence of boundary points:

\[
P = \{p_1, \cdots, p_n\}, \quad p_i \in \mathbb{R}^2
\]
What is Shape Context?

Definition (Shape Context)

Shape Context is a descriptor of interest point, i.e. a histogram: \[ h_i(k) = \# \{ p_j : j \neq i, x_j - x_i \in \text{bin}(k) \} \] in which bins are uniformly divided log-polar space.
What is Shape Context?

Definition (Shape Context)
Shape Context is a descriptor of interest point, i.e. a histogram:

\[ h_i(k) = \# \{ p_j : j \neq i, x_j - x_i \in \text{bin}(k) \}, \]

in which bins are uniformly divided log-polar space.
What is Shape Context?

Definition (Shape Context)

Shape Context is a descriptor of interest point, i.e. a histogram:

\[ h_i(k) = \# \{ p_j : j \neq i, x_j - x_i \in bin(k) \}, \]

in which bins are uniformly divided log-polar space.
Examples of Shape Context

[Images of shape context examples]
Matching Shape Contexts

- The cost of matching point $p_i$ on the first shape to point $q_j$ on the second shape (chi-square distance)

$$C_{ij} = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}$$ (1)
Matching Shape Contexts

- The cost of matching point $p_i$ on the first shape to point $q_j$ on the second shape (chi-square distance)
- Minimize total matching cost

$$\sum_{i} C(p_i, q_{\pi(i)}).$$  \hspace{1cm} (2)

This could be solved by Hungarian method in $O(n^3)$ time complexity. $\pi(i)$ is a permutation.
Matching Shape Contexts

- The cost of matching point $p_i$ on the first shape to point $q_j$ on the second shape (chi-square distance)
- Minimize total matching cost
  This could be solved by Hungarian method in $O(n^3)$ time complexity. $\pi(i)$ is a permutation.
Some Properties of Shape Context

- Invariant to translation and scale (normalization by mean distance of $n^2$ point pairs)
Some Properties of Shape Context

- Invariant to translation and scale (normalization by mean distance of $n^2$ point pairs)
- Can be made invariant to rotation (local tangent orientation)
Some Properties of Shape Context

- Invariant to translation and scale (normalization by mean distance of $n^2$ point pairs)
- Can be made invariant to rotation (local tangent orientation)
- Tolerant to small affine distortion (log-polar, spatial blur proportional to r)
Outline

1 Background

2 Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks

3 Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity

4 Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context

5 Experiments
Thin Plate Spline Model

- 2D generalization of cubic spline

\[ T(x, y) = (f_x(x, y), f_y(x, y)) \]

\[ f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{n} w_i r_i^2 \log r_i^2 \]  \hspace{1cm} (3)

\[ r_i = \| (x_i, y_i) - (x, y) \| \]
Thin Plate Spline Model

- 2D generalization of cubic spline
- Solved by minimizing bending energy

\[
f_{tps} = \arg \min_f E_{tps}
\]

\[
E_{tps} = \sum_{i=1}^{K} \| y_i - f(x_i) \|^2 + \lambda \int \int \left[ \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 \right] \, dx \, dy
\]
Thin Plate Spline Model

- 2D generalization of cubic spline
- Solved by minimizing bending energy

\[ f_{tps} = \arg \min_f E_{tps} \]

\[ E_{tps} = \sum_{i=1}^{K} \| y_i - f(x_i) \|^2 \]

\[ + \lambda \int \int \left[ \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 \right] \, dx \, dy \]

- Result in solving a linear system
Thin Plate Spline Model

- 2D generalization of cubic spline
- Solved by minimizing bending energy

\[ f_{tps} = \arg \min_f E_{tps} \]

\[ E_{tps} = \sum_{i=1}^{K} \| y_i - f(x_i) \|^2 \]

\[ + \lambda \int \int \left[ \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 \right] \, dx \, dy \] (4)

- Result in solving a linear system
- Tune regularization parameter \( \lambda \) to handle noisy data
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
**Similarity Measure**

**Definition**

After TPS transformation $T$ has been applied, two shapes’ similarity is measured by the weighted sum

$$D = aD_{ac} + D_{sc} + bD_{be} \quad (5)$$

- **shape context distance $D_{sc}$**

$$D_{sc}(P, Q) = \frac{1}{n} \sum_{p \in P} \arg \min_{q \in Q} C(p, T(q)) + \frac{1}{m} \sum_{q \in Q} \arg \min_{p \in P} C(p, T(q)) \quad (6)$$

the symmetric sum of shape context matching costs over best matching points
Similarity Measure

Definition

After TPS transformation $T$ has been applied, two shapes’ similarity is measured by the weighted sum

$$D = aD_{ac} + D_{sc} + bD_{be}$$  \hspace{1cm} (5)

- shape context distance $D_{sc}$
- appearance cost $D_{ac}$

$$D_{ac}(P, Q) = \frac{1}{n} \sum_{i=1}^{n} \sum_{\Delta \in Z^2} G(\Delta) \left[ I_P(p_i + \Delta) - I_Q(T(q_{\pi(i)}) + \Delta) \right]^2$$  \hspace{1cm} (7)

the sum of squared brightness differences in Gaussian windows around corresponding points
Similarity Measure

Definition

After TPS transformation $T$ has been applied, two shapes’ similarity is measured by the weighted sum

$$D = aD_{ac} + D_{sc} + bD_{be}$$

- shape context distance $D_{sc}$
- appearance cost $D_{ac}$
- transformation cost $D_{be}$
  in TPS case, it is the bending energy
Outline

1 Background
2 Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3 Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4 Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5 Experiments
Drawbacks of Shape Context

- not invariant to articulation

Example
Drawbacks of Shape Context

- not invariant to articulation
- does not capture part structure

Example
Drawbacks of Shape Context

- not invariant to articulation
- does not capture part structure

Example
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
Definition of Inner Distance

Definition (Shape)

A shape $O$ is a connected and closed subset of $\mathbb{R}^2$. 
Definition of Inner Distance

Definition (Shape)
A shape $O$ is a connected and closed subset of $\mathbb{R}^2$.

Definition (Inner Distance)
For two points $x, y \in O$, their inner distance, denoted as $d(x, y; O)$, is the length of the shortest path connecting $x$ and $y$ within $O$. 
Definition of Inner Distance

Definition (Shape)
A shape $O$ is a connected and closed subset of $\mathbb{R}^2$.

Definition (Inner Distance)
For two points $x, y \in O$, their inner distance, denoted as $d(x, y; O)$, is the length of the shortest path connecting $x$ and $y$ within $O$.

Example
Definition of Inner Distance

Definition (Shape)
A shape $O$ is a connected and closed subset of $\mathbb{R}^2$.

Definition (Inner Distance)
For two points $x, y \in O$, their inner distance, denoted as $d(x, y; O)$, is the length of the shortest path connecting $x$ and $y$ within $O$.

Note
- Multiple shortest paths appear rarely. If do, arbitrarily choose one
- Shapes are defined by their boundaries, but more on this later
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
     - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
How to Compute Inner Distance

Shortest Path Algorithm

1. Build a graph.
How to Compute Inner Distance

Shortest Path Algorithm

1. Build a graph.
   - Treat each sample point $p_i$ as a node $i$ in the graph
How to Compute Inner Distance

Shortest Path Algorithm

1. Build a graph.
   1. Treat each sample point $p_i$ as a node $i$ in the graph
   2. Add an edge between node $i$ and $j$, if and only if line segment $p_ip_j \in O$

Note: Neighboring boundary points are always connected. Hole boundary points are only used to determine $p_ip_j \in O$, not treated as nodes.
How to Compute Inner Distance

Shortest Path Algorithm

1. **Build a graph.**
   1. Treat each sample point \( p_i \) as a node \( i \) in the graph.
   2. Add an edge between node \( i \) and \( j \), if and only if line segment \( p_i p_j \in O \).

2. **Apply to the graph any standard all-pair shortest path algorithms.**
   E.g. Johnson or Floyd-Warshall’s algorithms, \( O(n^3) \) complexity.
How to Compute Inner Distance

Shortest Path Algorithm

1. Build a graph.
   1. Treat each sample point $p_i$ as a node $i$ in the graph
   2. Add an edge between node $i$ and $j$, if and only if line segment $\overline{p_ip_j} \in O$

2. Apply to the graph any standard all-pair shortest path algorithms. E.g. Johnson or Floyd-Warshall’s algorithms, $O(n^3)$ complexity.

Note

- Neighboring boundary points are always connected.
- Hole boundary points are only used to determine $\overline{p_ip_j} \in O$, not treated as nodes.
How to Compute Inner Distance

Note

- Neighboring boundary points are always connected.
- Hole boundary points are only used to determine $p_ip_j \in O$, not treated as nodes.

Example
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
Intuitions

- An articulated shape can be decomposed into rigid parts connected by very small junctions.

Example
Intuitions

- An articulated shape can be decomposed into rigid parts connected by very small junctions.
- Shortest path between landmark points can be divided into segments within each parts.
Intuitions

- An articulated shape can be decomposed into rigid parts connected by very small junctions.
- Shortest path between landmark points can be divided into segments within each parts.
- The articulation of $O$ is a rigid transformation with respect to any single part, but can be non-rigid on junctions.
Articulated Object Model

Definition (Articulated Object)

An articulated object $O \subset \mathbb{R}^2$ is composed of parts $O_i \subset \mathbb{R}^2$ and junctions $J_{ij} \subset \mathbb{R}^2$

$$O = \bigcup_{i=1}^{n} O_i \bigcup \bigcup_{i \neq j} J_{ij},$$

where, $\text{diam}(J_{ij}) = \max_{x, y \in P} \{d(x, y; J_{ij})\} \leq \epsilon$. 
Articulation Insensitivity

Theorem

Given an object $O$, $\forall x, y \in O$,

$$|d(x, y; O) - d(x', y'; O')| \leq \max\{m, m'\}\varepsilon,$$

where $m$ is the number of junctions the corresponding shortest path goes through, $(\cdot)'$ means after articulation transformation.

Example
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
Multi-Dimentional Scaling

Definition (MDS)

Given sample points \( P = \{ p_i \}_{i=1}^n \) on shape \( O \) and their inner distances \( \{ d_{ij} \}_{i,j=1}^n \), MDS finds the transformed points \( Q = \{ q_i \}_{i=1}^n \)

\[
\arg_{Q} \min S(Q) = \frac{\sum_{i<j} w_{ij} (d_{ij} - e_{ij}(Q))^2}{\sum_{i<j} d_{ij}^2},
\]

where \( w_{ij} \) are weights, \( \{ e_{ij}(Q) = \| q_i - q_j \| \}_{i,j=1}^n \)
Multi-Dimensional Scaling

Definition (MDS)

Given sample points \( P = \{ p_i \}_{i=1}^n \) on shape \( O \) and their inner distances \( \{ d_{ij} \}_{i,j=1}^n \), MDS finds the transformed points \( Q = \{ q_i \}_{i=1}^n \)

\[
\arg_{Q} \min S(Q) = \frac{\sum_{i<j} w_{ij} (d_{ij} - e_{ij}(Q))^2}{\sum_{i<j} d_{ij}^2},
\]

where \( w_{ij} \) are weights, \( \{ e_{ij}(Q) = \| q_i - q_j \| \}_{i,j=1}^n \)

Example
Outline

1 Background

2 Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks

3 Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity

4 Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context

5 Experiments
Inner Distance Shape Context

1. Replace Euclidean distance with inner distance in SC
Inner Distance Shape Context

1. Replace Euclidean distance with inner distance in SC
2. Replace relative orientation with inner angle in SC
Inner Distance Shape Context

1. Replace Euclidean distance with inner distance in SC
2. Replace relative orientation with inner angle in SC

Definition (Inner Angle)

Given two boundary point \( p, q \) and their shortest path \( \Gamma(p, q; O) \), the angle between the contour tangent at \( p \) and the direction of \( \Gamma(p, q; O) \) at \( p \) is the inner angle, \( \theta(p, q; O) \).
Inner Distance Shape Context

1. Replace Euclidean distance with inner distance in SC
2. Replace relative orientation with inner angle in SC

Definition (Inner Angle)

Given two boundary point $p, q$ and their shortest path $\Gamma(p, q; O)$, the angle between the contour tangent at $p$ and the direction of $\Gamma(p, q; O)$ at $p$ is the inner angle, $\theta(p, q; O)$

Example
Inner Distance Shape Context

1. Replace Euclidean distance with inner distance in SC
2. Replace relative orientation with inner angle in SC

Example
Matching through Dynamic Programming

- Since the contours provide ordering information, we can use $O(n^2)$ dynamic programming to solve the matching problem, instead of using $O(n^3)$ bipartite graph matching.
Matching through Dynamic Programming

- Since the contours provide ordering information, we can use $O(n^2)$ dynamic programming to solve the matching problem, instead of using $O(n^3)$ bipartite graph matching.
- By default we assume the two contours are aligned at start and end points, so it is $O(n^2)$ complexity. If not, we have to try different alignments and the complexity will be $O(n^3)$.
Matching through Dynamic Programming

- Since the contours provide ordering information, we can use $O(n^2)$ dynamic programming to solve the matching problem, instead of using $O(n^3)$ bipartite graph matching.
- By default we assume the two contours are aligned at start and end points, so it is $O(n^2)$ complexity. If not, we have to try different alignments and the complexity will be $O(n^3)$.
- But if we firstly rotate them according to the moments, then we need only try a small fixed number $k = 4$ or $8$ alignments, so still $O(kn^2) = O(n^2)$. 

Advantages

1. Better performance
2. Only two parameters to tune (non-match penalty $\tau$, alignment tries $k$)
3. Does not require appearance and transformation model.
Matching through Dynamic Programming

- Since the contours provide ordering information, we can use $O(n^2)$ dynamic programming to solve the matching problem, instead of using $O(n^3)$ bipartite graph matching.
- By default we assume the two contours are aligned at start and end points, so it is $O(n^2)$ complexity. If not, we have to try different alignments and the complexity will be $O(n^3)$.
- But if we firstly rotate them according to the moments, then we need only try a small fixed number $k = 4$ or $8$ alignments, so still $O(kn^2) = O(n^2)$.

Advantages

1. Better performance
Matching through Dynamic Programming

- Since the contours provide ordering information, we can use $O(n^2)$ dynamic programming to solve the matching problem, instead of using $O(n^3)$ bipartite graph matching.
- By default we assume the two contours are aligned at start and end points, so it is $O(n^2)$ complexity. If not, we have to try different alignments and the complexity will be $O(n^3)$.
- But if we firstly rotate them according to the moments, then we need only try a small fixed number $k = 4$ or $8$ alignments, so still $O(kn^2) = O(n^2)$.

Advantages

1. Better performance
2. Only two parameters to tune (non-match penalty $\tau$, alignment tries $k$)
Matching through Dynamic Programming

- Since the contours provide ordering information, we can use \( O(n^2) \) dynamic programming to solve the matching problem, instead of using \( O(n^3) \) bipartite graph matching.
- By default we assume the two contours are aligned at start and end points, so it is \( O(n^2) \) complexity. If not, we have to try different alignments and the complexity will be \( O(n^3) \).
- But if we firstly rotate them according to the moments, then we need only try a small fixed number \( k = 4 \) or \( 8 \) alignments, so still \( O(kn^2) = O(n^2) \).

Advantages

1. Better performance
2. Only two parameters to tune (non-match penalty \( \tau \), alignment trys \( k \))
3. Do not requires appearance and transformation model
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
Drawbacks of IDSC

Shape information is often not enough in real applications:

- Different classes may have similar shapes
Drawbacks of IDSC

Shape information is often not enough in real applications:
- Different classes may have similar shapes
- Occlusion, self-overlapping may damage shapes
Drawbacks of IDSC

Shape information is often not enough in real applications:
- Different classes may have similar shapes
- Occlusion, self-overlapping may damage shapes

Shortest Path Texture Context
A new descriptor considering texture information inside the shape · · ·
For each boundary point $p_i$, it is a three-dimensional histogram $h_i$. The first two dimensions are the same as IDSC, i.e., inner distance and inner angle. The third dimension is a measurement of the distributions of relative gradient orientation along all shortest paths ended at point $p_i$. Relative Gradient Orientation $\alpha(p_i, q; O)$.
Shortest Path Texture Context

1. For each boundary points $p_i$, it is a three-dimensional histogram $h_i$.
2. The first two dimensions are the same as IDSC, i.e. inner distance and inner angle.
Shortest Path Texture Context

1. For each boundary points $p_i$, it is a three-dimensional histogram $h_i$.
2. The first two dimensions are the same as IDSC, i.e. inner distance and inner angle.
3. The third dimension is a measurement of the distributions of relative gradient orientation along all shortest paths ended at point $p_i$.

Relative Gradient Orientation $\alpha(p, q; O)$
Outline

1. Background
2. Shape Context
   - Find Point Correspondences
   - Estimate Transformation
   - Measure Similarity
   - Drawbacks
3. Inner Distance
   - Definition
   - Computation
   - Articulation Insensitivity
4. Applications of Inner Distance
   - Articulation Invariant Signatures
   - Inner Distance Shape Context
   - Shortest Path Texture Context
5. Experiments
Parameters

- number of inner distance bins $n_d = 5$ or $8$
Experiments

Parameters

- number of inner distance bins $n_d = 5$ or $8$
- number of inner angle bins $n_{\theta} = 12$
Parameters

- number of inner distance bins $n_d = 5$ or $8$
- number of inner angle bins $n_{\theta} = 12$
- number of relative gradient orientation bins $n_r = 8$
Experiments

Parameters

- number of inner distance bins $n_d = 5$ or $8$
- number of inner angle bins $n_{\theta} = 12$
- number of relative gradient orientation bins $n_r = 8$
- number of different alignment trys $k = 4$ or $8$
Parameters

- number of inner distance bins $n_d = 5$ or $8$
- number of inner angle bins $n_\theta = 12$
- number of relative gradient orientation bins $n_r = 8$
- number of different alignment trys $k = 4$ or $8$
- non-matching penalty $\tau = 0.3$
Articulated Database

Database

(a)

(b)
Articulated Database

SC+DP Result
Articulated Database

IDSC+DP Result

![Diagram showing 2D shape matching results with IDSC+DP method. The diagram includes various shapes and their corresponding scores. The shapes are marked with red circles and outlines to highlight specific results. The scores range from 56.0 to 108.6.]
## Articulated Database

### Retrieval Result

<table>
<thead>
<tr>
<th>Distance Type</th>
<th>Top 1</th>
<th>Top 2</th>
<th>Top 3</th>
<th>Top 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_2$ (baseline)</td>
<td>25/40</td>
<td>15/40</td>
<td>12/40</td>
<td>10/40</td>
</tr>
<tr>
<td>SC+DP</td>
<td>20/40</td>
<td>10/40</td>
<td>11/40</td>
<td>5/40</td>
</tr>
<tr>
<td>MDS+SC+DP</td>
<td>36/40</td>
<td>26/40</td>
<td>17/40</td>
<td>15/40</td>
</tr>
<tr>
<td>IDSC+DP</td>
<td>40/40</td>
<td>34/40</td>
<td>35/40</td>
<td>27/40</td>
</tr>
</tbody>
</table>
MPEG7 Shape Database

Two retrieval examples
Foliage Image Retrieval

Three retrieval examples
Human Silhouettes Matching
Thank You!
Any Questions?