LB_Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures

Keogh, Wei, Xi, Lee & Vlachos

Come, we shall learn of the indexing of shapes
Outline of Talk

• The utility of shape matching
• Shape representations
• Shape distance measures
• Lower bounding rotation invariant measures with the LB_Keogh
• Accuracy experiments
• Efficiency experiments
• Conclusions

*Paradiso Canto XVIII 85

Set forth these figures as I have conceived their shape…*
The Utility of Shape Matching I

...discovering insect mimicry, clustering petroglyphs, finding unusual arrowheads, tracking fish migration, finding anomalous fruit fly wings...
The Utility of Shape Matching II

...automatically annotating old manuscripts, mining medical images, biometrics, spatial mining of horned lizards, indexing nematodes...
Shape Representations I

For virtually all shape matching problems, rotation is the problem.

If I asked you to group these reptile skulls, rotation would not confuse you.

There are two ways to be rotation invariant:

1) Landmarking: Find the one “true” rotation
2) Rotation invariant features
Landmarking

- **Domain Specific Landmarking**
  Find some fixed point in your domain, eg. the nose on a face, the stem of leaf, the tail of a fish …

- **Generic Landmarking**
  Find the major axis of the shape and use that as the canonical alignment

The only problem with landmarking is that it does not work
Rotation invariant features

Possibilities include:
Ratio of perimeter to area, fractal measures, elongatedness, circularity, min/max/mean curvature, entropy, perimeter of convex hull and histograms

The only problem with rotation invariant features is that in throwing away rotation information, you must invariably throw away useful information
We can convert shapes into a 1D signal. Thus can we remove information about *scale* and *offset*. *Rotation* we must deal with in our algorithms...

...so it seemed to change its shape, from running lengthwise to revolving round...*

There are many other 1D representations of shape, and our algorithm can work with *any* of them.

*Dante Alighieri. The Divine Comedy Paradiso -- Canto XXX, 90.*
Shape Distance Measures

There are but three…

- Euclidean Distance
- Dynamic Time Warping
- Longest Common Subsequence
For the next ten slides, temporarily forget about rotation invariance.

Euclidean Distance works well for matching many kinds of shapes.

Euclidean Distance

Mantled Howler Monkey
*Alouatta palliata*

Red Howler Monkey
*Alouatta seniculus seniculus*
Dynamic Time Warping is useful for natural shapes, which often exhibit intraclass variability.

Is man an ape or an angel?
Matching skulls is an important problem.

LCSS can deal with missing or occluded parts.

LCSS can deal with missing or occluded parts.

This region will not be matched.
For brevity, we will only give details of Euclidean distance in this talk.

However, the main point of our paper is that the same idea works for DTW and LCSS with no overhead.

We will present empirical results that do show that DTW can be significantly better than Euclidean distance.
Given two time series \( Q = q_1 \ldots q_n \) and \( C = c_1 \ldots c_n \), the Euclidean distance between them is defined as:

\[
D(Q, C) \equiv \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2}
\]

I notice that you Z-normalized the time series first.

The next slide shows a useful optimization.
During the computation, if current sum of the squared differences between each pair of corresponding data points exceeds $r^2$, we can safely abandon the calculation.

I see, because incremental value is always a lower bound to the final value, once it is greater than the best-so-far, we may as well abandon.
Most indexing techniques work by grouping objects into logical units, and defining a lower bound distance to the units.

For example, for indexing cities we can use MBRs and the classic MIN-DIST function of Guttman.

Here we will use "wedges" as the logical unit, and LB_Keogh as the lower bound distance.
Having candidate sequences $C_1, \ldots, C_k$, we can form two new sequences $U$ and $L$:

$$U_i = \max(C_{1i}, \ldots, C_{ki})$$

$$L_i = \min(C_{1i}, \ldots, C_{ki})$$

They form the smallest possible bounding envelope that encloses sequences $C_1, \ldots, C_k$.

We call the combination of $U$ and $L$ a wedge, and denote a wedge as $W$. $W = \{U, L\}$
A lower bounding measure between an arbitrary query $Q$ and the set of candidate sequences contained in a wedge $W$, is the $LB_{Keogh}$

$$LB_{Keogh}(Q,W) = \sqrt{\sum_{i=1}^{n} \left\{ \begin{array}{ll} (q_i - U_i)^2 & \text{if } q_i > U_i \\ (q_i - L_i)^2 & \text{if } q_i < L_i \\ 0 & \text{otherwise} \end{array} \right.}$$
Generalized Wedge

- Use $W_{(1,2)}$ to denote that a wedge is built from sequences $C_1$ and $C_2$.
- Wedges can be hierarchically nested. For example, $W_{((1,2),3)}$ consists of $W_{(1,2)}$ and $C_3$.

Of course, fatter wedges mean looser lower bounds…
We are finally ready to explain our idea for rotation invariance, an idea we have sidestepped to this point. Suppose we have a shape as before...

We can create every possible rotation of the shape, by considering every possible circular shift of the time series, as shown at my left... But we already know how to index such time series by using wedges! We just need to figure out the best wedge making policy..

It sucks being a grad student
Hierarchical Clustering

Which wedge set to choose?
Once we have all possible rotations of all the objects we want to index inserted into wedges, we can simply use any LB_Keogh indexer.

Since the introduction of LB_Keogh indexing at this conference 4 years ago, at least 50 groups around the world have used/extended/adapted the idea, making this work easily reimplementable.

What are the disadvantages of using LB_Keogh?

There are Nun
"LB_Keogh has provided a convincing lower bound" T. Rath
"LB_Keogh can significantly speed up DTW". Suzuki
"LB_Keogh is the best...". Zhou & Wong

"LB_Keogh offers the tightest lower bounds". M. Cardle.
"LB_Keogh makes retrieval of time-warped time series feasible even for large data sets". Muller et. al.

"LB_Keogh can be effectively used, resulting in considerably less number of DTW computations." Karydis
"exploiting LB_Keogh, we can guarantee indexability". Bartolini et. al.
"LB_Keogh, the best method to lower bound..." Capitani.
"LB_Keogh is fast, because it cleverly exploits global constraints that appear in dynamic programming" Christos Faloutsos.

By using the LB_Keogh framework, we can leverage off the wealth of work in the literature
All our Experiments are Reproducible!

People that do irreproducible experiments should be boiled alive

Agreed! All our data is publicly available

www.cs.ucr.edu/~eamonn/shape/
We tested on many diverse datasets

...and I recognized the face

...as a fish dives through water

Leaf of mine, in whom I found pleasure

...the shape of that cold animal which stings and lashes people with its tail

*Purgatorio -- Canto IX, Purgatorio -- Canto XXIII, Purgatorio -- Canto XXVI, Paradiso -- Canto XV 88
<table>
<thead>
<tr>
<th>Name</th>
<th>Classes</th>
<th>Instances</th>
<th>Euclidean Error (%)</th>
<th>DTW Error (%)</th>
<th>Other Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face</td>
<td>16</td>
<td>2240</td>
<td>3.839</td>
<td><strong>3.170</strong>{3}</td>
<td></td>
</tr>
<tr>
<td>Swedish Leaves</td>
<td>15</td>
<td>1125</td>
<td>13.33</td>
<td><strong>10.84</strong>{2}</td>
<td></td>
</tr>
<tr>
<td>Chicken</td>
<td>5</td>
<td>446</td>
<td>19.96</td>
<td>19.96{1}</td>
<td>20.5 Discrete strings</td>
</tr>
<tr>
<td>MixedBag</td>
<td>9</td>
<td>160</td>
<td>4.375</td>
<td>4.375{1}</td>
<td>Chamfer 6.0, Hausdorff 7.0</td>
</tr>
<tr>
<td>OSU Leaves</td>
<td>6</td>
<td>442</td>
<td>33.71</td>
<td><strong>15.61</strong>{2}</td>
<td></td>
</tr>
<tr>
<td>Diatoms</td>
<td>37</td>
<td>781</td>
<td>27.53</td>
<td>27.53{1}</td>
<td>26.0 Morphological Curvature Scale Spaces</td>
</tr>
<tr>
<td>Plane</td>
<td>7</td>
<td>210</td>
<td>0.95</td>
<td><strong>0.0</strong>{3}</td>
<td>0.55 Markov Descriptor</td>
</tr>
<tr>
<td>Fish</td>
<td>7</td>
<td>350</td>
<td>11.43</td>
<td><strong>9.71</strong>{1}</td>
<td>36.0 Fourier /Power Cepstrum</td>
</tr>
</tbody>
</table>

Note that DTW is sometimes worth the little extra effort.
Implementation details should not matter, for example the results reported should be the same if reimplemented in Ret Hat Linux.

We therefore use a cost model that is independent of hardware/software/buffer size etc. See the paper for details.

We compare to brute force, and were possible a Fourier based approach (it can’t handle DTW).
Main Memory Experiments

- Projectile point database
- Increasingly larger datasets
- One-nearest-neighbor queries

Euclidean

DTW

Number of objects in database ($m$)
Indexing Experiments

- Projectile point/Heterogenous databases
- Increasingly large dimensionality
- One-nearest-neighbor queries

![Graphs showing fraction of objects retrieved vs dimensionality for Projectile Points and Heterogeneous databases, with comparison between Euclidean and DTW Wedge metrics.](image-url)
All these are in the genus *Cercopithecus*, except for the skull identified as being either a Vervet or Green monkey, both of which belong in the Genus of *Chlorocebus* which is in the same Tribe (*Cercopithecini*) as *Cercopithecus*.

- Tribe Cercopithecini
  - *Cercopithecus*
    - De Brazza's Monkey, *Cercopithecus neglectus*
    - Mustached Guenon, *Cercopithecus cephus*
    - Red-tailed Monkey, *Cercopithecus ascanius*
  - *Chlorocebus*
    - Green Monkey, *Chlorocebus sabaeus*
    - Vervet Monkey, *Chlorocebus pygerythrus*

These are the same species
- *Bunopithecus hooloc* (Hoolock Gibbon)

These are in the Genus *Pongo*

- All these are in the family *Cebidae*
- Family *Cebidae* (*New World monkeys*)
  - Subfamily *Aotinae*
    - Aotus trivirgatus
  - Subfamily *Pitheciinae* *sakis*
    - Black Bearded Saki, *Chiropotes satanas*
    - White-nosed Saki, *Chiropotes albinasus*

*Purgatorio -- Canto XXIV 117*
Unlike the primates, reptiles require warping...

Flat-tailed Horned Lizard
*Phrynosoma mcallii*

Texas Horned Lizard
*Phrynosoma cornutum*
There is a special reason why this tree is so tall and inverted at its top.*

*Purgatorio -- Canto XXXIII 64
Petroglyph Mining

- They appear worldwide
- Over a million in America alone
- Surprisingly little known about them

Petroglyphs are images incised in rock, usually by prehistoric peoples. They were an important form of pre-writing symbols, used in communication from approximately 10,000 B.C.E. to modern times. Wikipedia

* Purgatorio -- Canto XII 6

who so sketched out the shapes there?*

.. they would strike the subtlest minds with awe*
Such complex shapes probably need DTW
Future Work: Data Mining

We did not want to work on shape data mining until we could do fast matching, that would have been ass backwards.

.. so similar in act and coloration that I will put them both to one*

* Inferno -- Canto XXIII  29
Questions?

Feel free to email us with questions.
Eamonn Keogh: Project Leader
eamonn@cs.ucr.edu

Li Wei: Lower Bounding
wli@cs.ucr.edu

Michail Vlachos: Public Nudity and Index Structures
vlachos@us.ibm.com

Sang Hee Lee: Anthropology and Primatology
shlee@ucr.edu

Xiaopeng Xi: Image Processing
xxi@cs.ucr.edu