Graz University of Technology

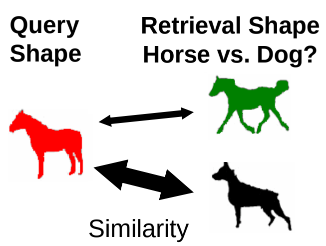
Beyond Pairwise Shape Similarity Analysis

Peter Kontschieder

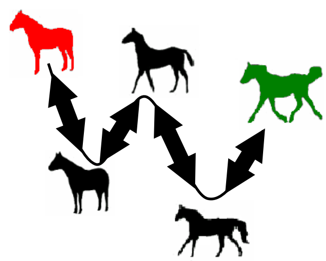
Institute for Computer Graphics and Vision, Graz University of Technology

Abstract This work considers two major applications of shape matching algorithms: (a) query-by-example, i.e. retrieving the most similar shapes from a database and (b) finding clusters of shapes, each represented by a single prototype. Our approach goes beyond pairwise shape similarity analysis by considering the underlying structure of the shape manifold, which is estimated from the shape similarity scores between all the shapes within the input database. We propose a modified mutual *k*NN graph as underlying representation and demonstrate its performance for the task of shape retrieval. We further describe an efficient, unsupervised clustering method which uses the modified mutual *k*NN graph for initialization. Experimental evaluation proves the applicability of our method, e.g. by achieving the highest ever reported retrieval score of 93.40% on the well known MPEG-7 database.

Motivation



Possibly incorrect retrieval as result of simple, pairwise shape matching



Find & exploit intra-class geodesic paths on shape manifold

Manifold Normalization & Analysis for Retrieval

1. Normalization by Local Scaling

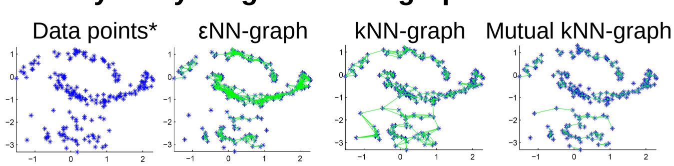
Let W be a $N \times N$ affinity matrix which we derive from a given distance matrix A by

$$w_{ij} = e^{-\frac{a_{ij}^2}{\sigma_{ij}^2}}$$

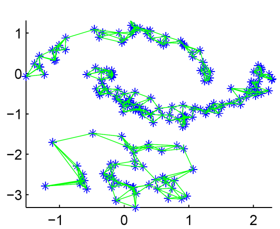
where $\sigma_{ij} = \sigma_i \sigma_j$ with $\sigma_i = a_{iK(i)}$

and $K(i)$ is the Kth NN of object i

2. Analysis by Neighborhood graphs



Proposed Modified Mutual kNN-graph



Two vertices (v_i, v_j) are connected only, if

$v_j \in kNN(v_i)$ and

$v_i \in ckNN(v_j)$

$c \geq 1$ … asymmetry coefficient

*von Luxburg, U. A Tutorial on Spectral Clustering, Technical Report 2006.

Shape Clustering and Prototype Identification

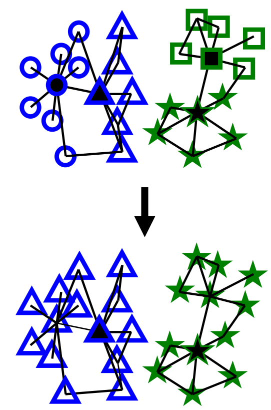
1. Initialization from Modified Mutual kNN graph

   Choose vertices with high degree as potential cluster prototypes v_{C_i}

2. Iteratively check intra- and inter-class distances for possible mergings

Blue elements … Class 1

Green elements … Class 2



Intra-class distance:

$$Intra(C_i) = \max_{e \in C_i} w(e)$$

Inter-class distance:

$$Inter(C_i, C_j) = d_{shortest}(v_{C_i}, v_{C_j})$$

Inter-class upper boundary:

$$Inter_{bnd}(C_i, C_j) = \min(Intra(C_i) + T\tau(C_i), Intra(C_j) + T\tau(C_j))$$

Where $\tau(C)$ is a self-scaling threshold function.

Merging criterion:

$$Inter(C_i, C_j) \leq Inter_{bnd}(C_i, C_j)$$

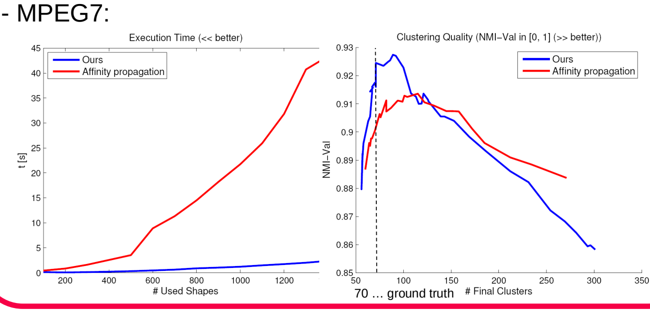
Results

Highest ever reported retrieval scores on MPEG7 and KIMIA99 databases

Algorithm	Graph Transduction ECCV 2008	Locally Densified Spaces, CVPR 2009, [2]	**Our Method**
MPEG7 scores +	91.00 %	93.32 %	**93.40 %**
Execution times	6444 s	n. A.	**3.2 s**
KIMIA99 scores	99.79 %	n. A.	**100.00 %**

Clustering results compared to Affinity Propagation [1]

- 100% correct clustering results on KIMIA99 database
- MPEG7:



References

[1] B. J. Frey and D. Dueck, Clustering by Passing Messages Between Data Points, Science, vol. 315, pp. 972-976, 2007.

[2] X. Yang, S. Köknar-Tezel, L. J. Latecki, Locally Constrained Diffusion Process on Locally Densified Distance Spaces with Applications to Shape Retrieval, CVPR 2009.