Similarity Search in Fuzzy Object Databases

Diana Uskat, Tobias Emrich, Andreas Züfle, Klaus Arthur Schmid, Thomas Bernecker, Matthias Renz
Institute for Informatics, Ludwig-Maximilians-Universität, Munich
{uskat,emrich,zuefle,schmid,bernecker,renz}@dbs.ifi.lmu.de
Satellite Imagery

Analysis of depicted objects based on pixel information.

Precision highly dependent on resolution and image quality.

Which pixels belong to objects? What happens in ambiguous cases?

Satellite Photos © Space Imaging
Medical Imagery

Interpretation of medical images.

Example: brain cell.

Actual picture may be interpreted in different ways, depending on which pixels are flagged to be part of object.
Similarity Search on Fuzzy Object Databases.

- Distance measures dependent on cut-off (alpha value)
- Possibly different alphas for each objects
- Alpha specified at query time
Fuzzy Objects

• Object A in d-dimensional space
• Represented by set of spatial points
• Each point in a associated with membership probability $\mu_A \in [0, 1]$
• $\mu_A$ is membership function of A: Indicates probability of point belonging to A
Alpha-Cut

Support Set of $A$:
$A_S = \{ a \in A \mid \mu_A(a) > 0 \}$

Kernel Set of $A$:
$A_k = \{ a \in A \mid \mu_A(a) = 1 \}$

Alpha-Cut of $A$:
$A_\alpha = \{ a \in A \mid \mu_A(a) \geq \alpha \}$
Rubber Duck from ALOI dataset
Object Partitioning via concentrical shells
Different Alpha-Cuts

\( \alpha = 0.1 \)

\( \alpha = 0.5 \)

\( \alpha = 0.9 \)

Alternative feature extraction methods: e.g., sectors, grids, ...
Workflow: what we actually do

- Characterisation of each image in database
- Similarity defined by distance of histograms
- Necessary for quick pruning: Efficient approximation of alpha-cuts
- Needs to work in query time
Approximation of $\alpha$-density function, per shell

Actual distribution
Experimental Evaluation: Approximation

- **Precision** vs **Number of Separations**
  - Two graphs showing the performance of `APPoly3` and `APSigmoid` algorithms.
  - `APPoly3` graph shows a steady increase in precision with more separations.
  - `APSigmoid` graph also shows an increase, but at a slower rate.

- **Recall** vs **Number of Separations**
  - Similar to the precision graphs, showing the performance of `APPoly3` and `APSigmoid`.
  - `APPoly3` graph shows a steady increase in recall.
  - `APSigmoid` graph also increases, but the rate is lower than `APPoly3`.

These graphs illustrate the comparative performance of the two algorithms under varied conditions of separation in fuzzy object databases.
Bounding of $\alpha$-density function, per shell

Aim: upper and lower bounds for monotonically decreasing function.

$f^\wedge (\alpha)$ with seven samples

Naive bounds: Can be wrong
Bounding of $\alpha$-density function, per shell

Better: box bounding

Actual bounds (cut at zero)
fkNN Multistep Algorithm

Find k-nearest neighbours for query object from large image database
fkNN Multistep Algorithm

Determine exact alpha-cut for query image.

1. Sequential scan of image database, determine upper and lower bound values for given alpha, for each sector / shell.

   Shell 1: 67
   Shell 2: 48
   Shell 3: 32
   ...

2. Determine exact alpha-cut for query image.

   Shell 1: [42,88]
   Shell 2: [31,65]
   Shell 3: [12,22]
   ...

Contribution 3:
Multistep fkNN (cont.)
fkNN Multistep Algorithm

3 Rank candidates by lower bound distance, ascending.

4 Pruning step: Prune all whose lower bound is > than upper bound of k-th

5 Expansion step: Expand all candidates in between
Experimental Evaluation: Upper and Lower Bounds

![Graphs showing time vs. database size for different search methods.]

- **SeqScan**
- **APPoly**
- **APSigmoid**
- **MSLinBound**
- **MSLinBoundC**
- **MSPreAlphaCut**

The left graph shows the time in ns (nanoseconds) for different database sizes, while the right graph shows a different set of methods with a different scale for time.
Thank you for your attention.