Defuzzification Based on Feature Distance Minimization

Fuzzy Sets and Fuzzy Techniques
Lecture 11
Nataša Sladoje

Outline
Introduction
- Fuzzy sets in image processing
- Defuzzification
Defuzzification by feature distance minimization
- Feature distance minimization
- Defuzzification at increased spatial resolution
- Scale space approach
- Extension to 3D

Fuzzy sets in image processing
In the digitization process, fuzzy representations preserve information about the original object better than crisp representations.

- In cases where fuzzy image analysis tools are not yet developed, while crisp analogues exist.

Defuzzification
- to generate a crisp representation of a fuzzy set.

Sometimes crisp is required
Reasons to use a crisp representation may be:
- to provide easier and less subjective interpretation of images, especially if the dimension of the image is higher than 2;
- in cases when fuzzy image analysis tools are not yet developed, while crisp analogues exist.

Defuzzification by \( \alpha \)-cutting
Thresholding of the membership function.

An example:
A part of a microscopy image of a bone implant. Original and a fuzzy segmented bone region. \( \alpha \)-cut at \( \alpha = 0.25 \) \( \alpha \)-cut at \( \alpha = 0.5 \) \( \alpha \)-cut at \( \alpha = 0.75 \) Which \( \alpha \) to choose?
Defuzzification by minimizing point-wise difference

\[ \mu = 0.2 \quad \mu = 0.4 \quad \mu = 0.6 \quad \mu = 0.8 \quad \mu = 1 \]

Stability?

Outline

Introduction
- Fuzzy sets in image processing
- Defuzzification

Defuzzification by feature distance minimization
- Feature distance minimization
- Defuzzification at increased spatial resolution
- Scale space approach
- Extension to 3D

Defuzzification by feature distance minimization

Defuzzification formulated as an optimization problem

\[
\text{Definition: An optimal defuzzification of a fuzzy set } A \text{ is a crisp set } C \text{ (on the reference set } X) \text{ such that the distance between } A \text{ and } C \text{ is minimal:}
\]

\[
D(A) \in \{ C \in P(X) \mid d(A, C) = \min_{B \in P(X)} [d(A, B)] \}.
\]

Find the crisp set closest to the given fuzzy set.
- Which distance to use?
- How to perform the optimization?

Feature distance

- Extract both local and global “features” of the sets.
- Make a vector representation of each set using the extracted features.
- Measure the distance between two sets in this high dimensional feature space.

The feature distance between sets \(A\) and \(B\), with feature representations \(\Phi(A)\) and \(\Phi(B)\), is

\[
d^p(A, B) = d(\Phi(A), \Phi(B)).
\]

We use the Minkowski distance, with \(p=1\), as distance measure in the feature space.

\[
d_p(x, y) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}
\]

Examples of defuzzification based on different features

original synthetic fuzzy image

memberships

gradient

perimeter

area

centroid

all features

A combination of many features provides a good defuzzification result.

Minimizing the distance

- An analytic solution is, in general, not know.
- \(2^n\) configurations – exhaustive search is not an option.
- Some heuristics must be applied.

Simulated annealing has shown to provide a good trade off between speed and performance.

The optimal \(\alpha\)-cut provides a fast alternative (and a good starting point for the simulated annealing).
Defuzzification by feature distance minimization

<table>
<thead>
<tr>
<th></th>
<th>Perimeter</th>
<th>Area</th>
<th>Centroid</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy object</td>
<td>91.46</td>
<td>182.78</td>
<td>(13.66,21.08)</td>
<td>0</td>
</tr>
<tr>
<td>α-cut at α=0.5</td>
<td>92.75</td>
<td>176.00</td>
<td>(13.57,21.18)</td>
<td>0.1271</td>
</tr>
<tr>
<td>α-cut at optimal</td>
<td>93.30</td>
<td>182.00</td>
<td>(13.68,21.10)</td>
<td>0.1177</td>
</tr>
<tr>
<td>Sim. annealing</td>
<td>91.48</td>
<td>182.00</td>
<td>(13.65,21.08)</td>
<td>0.1053</td>
</tr>
</tbody>
</table>

Improved stability from global features

<table>
<thead>
<tr>
<th></th>
<th>μ=0.2</th>
<th>μ=0.4</th>
<th>μ=0.6</th>
<th>μ=0.8</th>
<th>μ=1</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Membership based defuzzification" /></td>
<td><img src="image2" alt="Feature based defuzzification (simulated annealing)" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Improved stability from global features

<table>
<thead>
<tr>
<th></th>
<th>μ=0.2</th>
<th>μ=0.4</th>
<th>μ=0.6</th>
<th>μ=0.8</th>
<th>μ=1</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3" alt="Stability?" /></td>
<td><img src="image4" alt="Better stability" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Improved stability from global features

<table>
<thead>
<tr>
<th></th>
<th>μ=0.2</th>
<th>μ=0.4</th>
<th>μ=0.6</th>
<th>μ=0.8</th>
<th>μ=1</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image5" alt="Feature based defuzzification (simulated annealing)" /></td>
<td><img src="image6" alt="Feature based defuzzification (simulated annealing)" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Additional constraints in the defuzzification

<table>
<thead>
<tr>
<th></th>
<th>μ=0.2</th>
<th>μ=0.4</th>
<th>μ=0.6</th>
<th>μ=0.8</th>
<th>μ=1</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image7" alt="Feature based defuzzification (simulated annealing)" /></td>
<td><img src="image8" alt="With topological constraints" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Defuzzification at increased spatial resolution

- Features of an original crisp continuous object can be estimated with a higher precision from a fuzzy representation, than from a crisp representation at the same spatial resolution.
- We use that information to find a crisp representative at an increased spatial resolution.
Distance between sets at different resolution

De-couple the feature representation from the image resolution.
- Global features
  - By proper rescaling of the feature values.
- Local features
  - By comparing individual pixels of the lower resolution image, with blocks of pixels of the higher resolution image.

Membership value – area of a block of pixels

The membership value of a fuzzy pixel corresponds to the area of a block of \( r \times r \) crisp sub-pixels.

High resolution defuzzification

<table>
<thead>
<tr>
<th>( \mu = 0.2 )</th>
<th>( \mu = 0.4 )</th>
<th>( \mu = 0.6 )</th>
<th>( \mu = 0.8 )</th>
<th>( \mu = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>12\times12</td>
<td>24\times24</td>
<td>48\times48</td>
<td>96\times96</td>
<td>192\times192</td>
</tr>
</tbody>
</table>

Fuzzy disk
12 x 12 pixels
8 bits per pixel = 1152 bits
32 times more information!

Defuzzified set
192 x 192 pixels
Crisp disk
192 x 192 pixels
1 bit per pixel = 36,864 bits

High resolution reconstruction disk vs. octagon

Fuzzy objects, 16 x 16 pixels,
Defuzzified objects, 256 x 256 pixels,
Crisp original objects, 256 x 256 pixels
Realistic example: Blood vessels

A fuzzy segmentation, and defuzzifications, of a slice in a 3D MRA of a human aorta, where it separates into the two iliac arteries.

Motivation for a scale space approach

- Example with four discrete fuzzy disks

The method can “transport” area from one part of the image to another.

Meso-scale features

- Meso-scale is scale in-between the local (pixel-size) and the global (the whole object) scale.
- Characteristics of a fuzzy set often have different importance at different scales.
- Our first choice of feature to observe at meso-scale is area (volume in 3D case).

Feature vector representation for scale space defuzzification

- Features included in the vector representation of a 2D fuzzy set are
  - Area, at a number of scales (including pixel-size, and global)
  - Perimeter
  - Centroid
- Resolution pyramid representation of sets is utilized for storing areas of blocks of pixels.
- Comparison of meso-scale areas reduces to a comparison of the corresponding pixels in the sets at corresponding levels of the resolution pyramid.
- All measures are appropriately rescaled.

Defuzzification of 4 circles

Four fuzzy disks

A. Defuzzification based on local and global features

B. Defuzzification using scale space approach

Perimeter | Area   | Centroid | Membership | Meso-scale | Dist 1 | Dist 2
---|---|---|---|---|---|---
A. | 0.0030 | 0.0000 | 0.0000 | 0.0057 | 0.3289 | 0.0952 | 0.5785
B. | 0.0015 | 0.0031 | 0.0000 | 0.0057 | 0.1758 | 0.1353 | 0.3413

Contribution of different features to the distance measure. Dist1 without meso-scale, Dist2 with meso-scale features.

Scale space defuzzification of 3D fuzzy sets

- Defuzzification method for 2D fuzzy sets is straightforwardly generalized to the 3D case.
- The features included in the 3D defuzzification are local, meso-scale, and global volumes, obtained by iterative grouping of blocks of 2x2x2 voxels, together with surface area and centroid.
- Larger search space puts additional demands on the efficiency of the optimization method.
- Simulated annealing with re-annealing is a reasonable option, if started from an optimal $\alpha$-cut.
Instead of a conclusion…

A better solution to a crisp problem is found by looking first in a larger space of fuzzy sets which has different (usually less) constraints and therefore allows the algorithms more freedom and reduces errors caused by forced crisp answers at intermediate steps.