SHERPA – Technical Details

This document provides a supplement to the SHERPA manual and contains descriptions of the methods, algorithms and measures employed, going into detail for those topics which are not explained comprehensively within the manual.

Most of the image processing functions are realized using OpenCV 2.4.2 (itseez 2012), which is wrapped for .NET by Emgu CV 2.4.2 (Emgu 2012).

1 Image formats

Valid image file formats are TIFF, JPG, PNG and BMP, containing 8bit gray level or 24bit color data.

2 Segmentation

Segmentation is preceded by noise reduction using a Gaussian or a median filter, realized via OpenCV function “cvSmooth”.

Segmentation using Otsu’s method (Otsu 1979), histogram equalization (Bradski and Kaehler 2008, p. 186ff.) plus Otsu’s method, Canny edge detector (Canny 1986) and adaptive thresholding (Bradski and Kaehler 2008, p. 138) is realized using the according OpenCV functions. The implementation of the robust automated threshold selector is based on (Lehmann 2006), using ITK 4.2 (Kitware and al. 2012).

3 Contour extraction

Object contours are derived from the segmented images using the OpenCV “findContours” function, which employs (Suzuki and Keiichi 1985). Since only the outer contours are used for analysis, the topological information is discarded.

4 Contour optimization

Morphological operators are applied using the OpenCV “morphologyEx” function, the structuring element is disc shaped.

5 Shape measures

Contour area and perimeter are calculated by the appropriate OpenCV functions. The rotation angle φ is calculated based on central moments:

$$\varphi = 0.5 \cdot atan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$$

with μ_{pq} = central moments of order $(p + q)$

The correct sign of φ is determined by using the .NET function “Math.Atan2”.

Width and height are calculated as the largest distance between contour points along the rotation angle resp. perpendicular to it. The enclosing rectangle is calculated accordingly.

6 Template Matching

Matching between templates and objects is based on elliptic Fourier descriptors. SHERPA's implementation uses the algorithms from (Claude 2008) and the corresponding errata (Claude 2010) for calculating the descriptors and their normals (in SHERPA referred to as "invariants"). The matching value is calculated by summing up the squared distances between the coefficients for all harmonics of index 2 and higher.

A set of contour points is selected according to the number of harmonics to be used for Fourier analysis. This points are distributed at equal distances along the contour perimeter in clockwise direction. For jagged shapes this might result in a set of points of varying Euclidian distances, but for smoother contours this approach works well. Starting point (index 0) is the leftmost point with respect to the major axis.

For the template having the lowest matching value (= best match) according to Fourier analysis, in addition Hu invariants are calculated using the OpenCV function "HuMoments" and matched against the Hu invariants of the object shape by employing the OpenCV function "matchShapes" with comparison method "CV_CONTOUR_MATCH_I1".

7 Quality indicators

The standard deviation for the inner part of the object area is calculated using the OpenCV function "AvgSdv", masking out the outer part of the object. The mask is derived by applying morphological erosion to the object shape using a 3x3 rectangular structuring element until the resulting area is below 50% of the original area.

Contour smoothness is calculated as ratio between the perimeters of the smoothed contour and the original contour. The contour is smoothed by applying a Gaussian filter to the x and y coordinates of its points. The "formfactor" is described below.

8 Heuristic descriptors

Heuristic descriptors are calculated according to:

Formfactor $$= \frac{4\pi \cdot Area}{Perimeter^2}$$ (Russ 2011)

Rectangularity $$= \frac{Area\ of\ object}{Area\ of\ enclosing\ rectangle}$$ (du Buf and Bayer 2002)

Compactness $$= \frac{\sqrt{\frac{4}{\pi} \cdot Area}}{max.\,Diameter}$$ (Russ 2011)

Ellipticity	$$= \begin{cases} 16\,\pi^2\, I_1 & if\ I_1 \le \frac{1}{16\,\pi^2} \\ \frac{1}{16\,\pi^2\, I_1} & otherwise \end{cases}$$ $$with\ I_1 = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4}$$	(Rosin 2003)
Triangularity	$$= \begin{cases} 108\, I_1 & if\ I_1 \le \frac{1}{108} \\ \frac{1}{108\, I_1} & otherwise \end{cases}$$ $$with\ I_1 = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4}$$	(Rosin 2003)
Roundness	$$= \frac{4 \cdot Area}{\pi \cdot max.Diameter^2}$$	(Russ 2011)
Convexity by perimeter	$$= \frac{Perimeter\ of\ convex\ hull}{Perimeter\ of\ object}$$	(Russ 2011)
Convexity by area	$$= \frac{Area\ of\ object}{Area\ of\ convex\ hull}$$	(Zunic and Rosin 2002)

9 Absolute convexity measures

All convexity measures are based on comparison between the object shape and its convex hull. The convex hull is calculated using the OpenCV function “convexHull”, which employs an algorithm developed by Sklansky (Sklansky 1982).

The “Convexity Defection Factor” (CDF) (Kloster 2013) calculates the mean ratios of area resp. perimeter:

$$CDF = 0{,}5 \cdot \left(\frac{A_H}{A_O} + \frac{P_O}{P_H}\right)$$

with A_H = convex hull area; A_O = object area; P_H = convex hull perimeter; P_O = object perimeter

The “Percent Concave Area Fraction” (PCAF) (Nafe and Schlote 2002) describes the ratio between the area of object and convex hull as:

$$PCAF = 100 \cdot \frac{A_H - A_O}{A_O}$$

with A_H = convex hull area; A_O = object area

For the “Convex Hull Maximum Distance Factor” (Kloster 2013) each convexity defect’s maximum distance between contour and convex hull is calculated (see Fig. 1). For the distances larger than $\sqrt{2} \cdot$ pixelwidth the squares of the distances are summed up to the CHMDF:

$$CHMDF = \sum D_{max}{}^{2} : \{\forall\, D_{max} > \sqrt{2}\}$$

with D_{max} = maximum distance between object contour and convex hull within each convexity defect

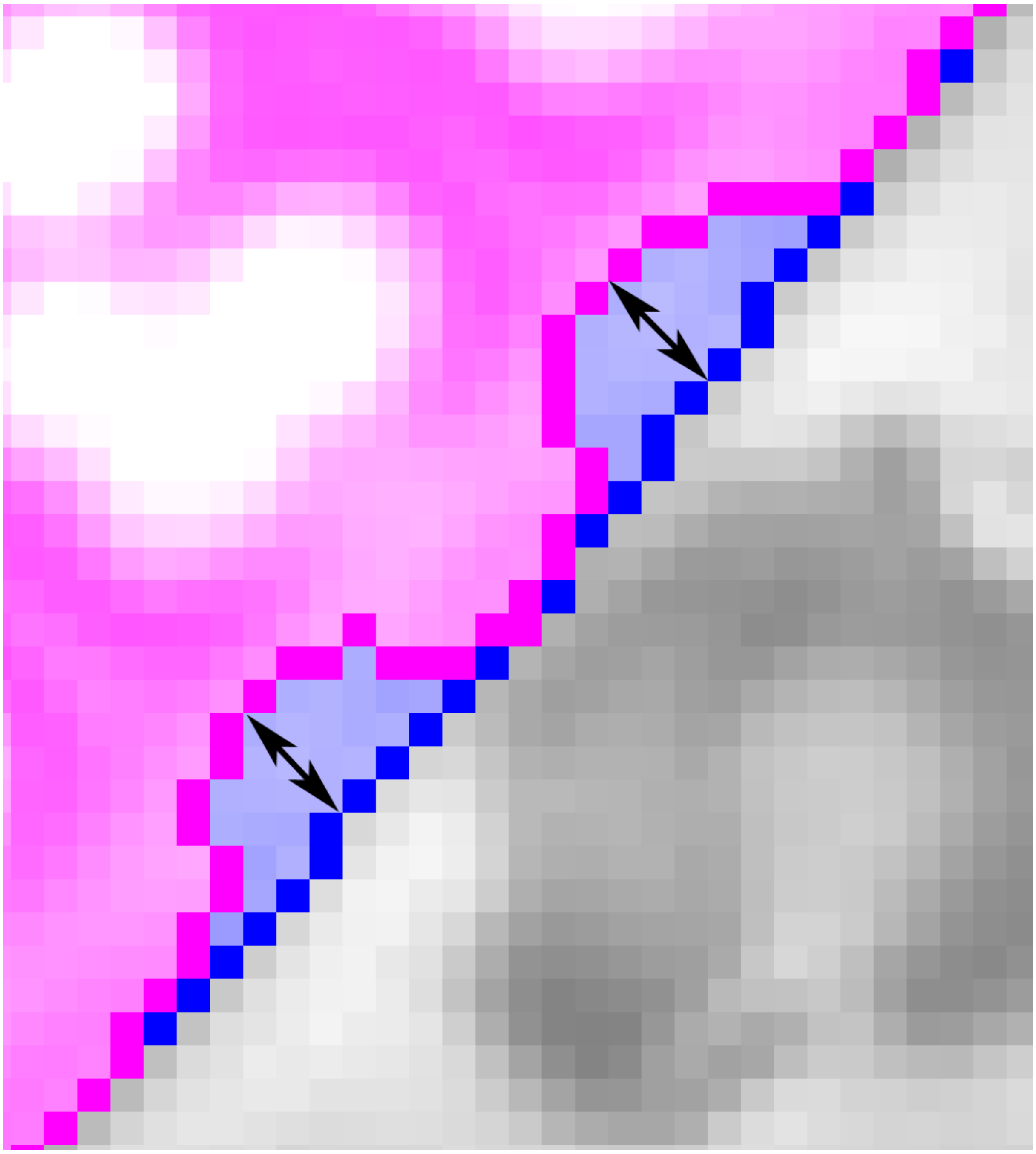

Fig. 1: Illustration of the CHMDF. The object area is highlighted red, the convex hull blue, resulting in purple for their intersection and blue for the convexity defects. Black arrows depict the maximum distance between object contour and convex hull within each convexity defect.

10 Relative convexity measures

The relative convexity measures compare CDF, PCAF and the heuristic descriptor compactness of object and best matching template:

CDF-Match $= \begin{cases} R_{CDF} & if\ R_{CDF} \leq 1 \\ \frac{1}{R_{CDF}} & otherwise \end{cases}$

$$with\ R_{CDF} = \frac{template\ CDF}{object\ CDF}$$

PCAF-Match $= \begin{cases} R_{PCAF} & if\ R_{PCAF} \leq 1 \\ \frac{1}{R_{PCAF}} & otherwise \end{cases}$

$$with\ R_{PCAF} = \frac{template\ PCAF}{object\ PCAF}$$

Compactness-Match $= \begin{cases} R_{Comp} & if\ R_{Comp} \leq 1 \\ \frac{1}{R_{Comp}} & otherwise \end{cases}$

$$with\ R_{Comp} = \frac{template\ compactness}{object\ compactness}$$

11 Ranking

Ranking depends on the quality and the convexity indicators. The indicators are calculated according to the respective thresholds, values exceeding the appropriate range cause a penalty and thus a higher (= worse) ranking value.

The convexity indicators are calculated depending on the type of convexity analysis (use / force convexity or none of these).

Please consult the SHERPA manual for more information on this topic.

12 Export

Analysis results can be exported to a set of CSV files ("Comma Separated Values", text files which can be opened e.g. by Excel) and TIFF image files (which can be opened by most graphics programs). Depending on the features selected for export, following files will be created:

filename.csv	Basic data for all contours, similar to the view in SHERPA's contour area, plus some additional data. The index number listed is used for naming files specific to a single contour ("xxxxxx" below).

filename.settings.csv	Settings used for segmentation and analysis.
filename.xxxxxx.XY.csv	Coordinates of the contour points
filename.xxxxxx.XY_EFA.csv	Coordinates of the set of points used for elliptic Fourier analysis
filename.xxxxxx.EFDs.csv	Coefficients from elliptic Fourier analysis and their normals/invariants
filename.xxxxxx.cropped.tif	Cutout of the image data showing the object. The coordinates of the top left point with respect to the original image is listed in "filename.csv".
filename.xxxxxx.contour.tif	Image showing the object shape
filename.xxxxxx.convexHull.tif	Image showing the convex hull
filename.xxxxxx.all.tif	Image showing a combination of the upper three plus the enclosing rectangle

The data for template files can be exported accordingly apart from the image data.

13 Literature

Bradski, G. and A. Kaehler (2008). Learning OpenCV : Computer Vision with the OpenCV Library. Sebastopol, O'Reilly.

Canny, J. (1986). "A Computational Approach to Edge Detection." IEEE Transactions on Pattern Analysis and Machine Intelligence **30**: 125-147.

Claude, J. (2008). Morphometrics with R. New York, Springer Science+Business Media, LLC.

Claude, J. (2010). "Morphometrics with R - Errata 1.81." from http://www.isem.univ-montp2.fr/recherche/files/2012/01/Morphometrics_errata1.81.pdf.

du Buf, H. and M. M. Bayer (2002). Automatic Diatom Identification. New Jersey, London, Singapore, Hong Kong, World Scientific Publishing Co. Pte. Ltd.

Emgu (2012). "Emgu CV, a cross platform .NET wrapper for the OpenCV image processing library, Version 2.4.2." from http://www.emgu.com.

itseez (2012). "OpenCV (Open Source Computer Vision Library) Version 2.4.2." from http://opencv.org/.

Kitware and e. al. (2012). " Insight Segmentation and Registration Toolkit (ITK), Version 4.20." from http://www.itk.org/.

Kloster, M. (2013). Digitale Bildsignalverarbeitung in der Bioinformatik: Methoden zur Segmentierung und Klassifizierung biologischer Merkmale am Beispiel ausgewählter Diatomeen. Emden, University of Applied Sciences Emden/Leer. **Master of Engineering**.

Lehmann, G. (2006). "Robust Automatic Threshold Selection." The Insight Journal **2006** (July - December).

Nafe, R. and W. Schlote (2002). "Methods of Shape Analysis of two-dimensional closed Contours - A biologically important, but widely neglected Field in Histopathology." Electronic Journal of Pathology and Histology **8**(2): 1-18.

Otsu, N. (1979). "A Threshold Selection Method from Gray-Level Histograms." IEEE Transactions on Systems, Man and Cybernetics **9**(1): 62-66.

Rosin, P. L. (2003). "Measuring shape: ellipticity, rectangularity, and triangularity." Machine Vision and Applications **14**(3): 172-184.

Russ, J. C. (2011). The Image Processing Handbook, Sith Edition. Boca Raton, London, New York, CRC Press.

Sklansky, J. (1982). "Finding the convex hull of a simple polygon." Pattern Recognition Letters **1**(2): 79-83

Suzuki, S. and A. Keiichi (1985). "Topological Structural Analysis of Digitized Binary Images by Border Following." Computer Vision, Graphics, and Image Processing **30**: 32-46.

Zunic, J. and P. L. Rosin (2002). A Convexity Measurement for Polygons. Proceedings of the British Machine Vision Conference 2002, BMVC 2002. Cardiff, UK.