Journal of
open research software

Wagner, T and Lipinksi, H 2013 IJBlob: An ImageJ Library for Connected Component Analysis and Shape Analysis. *Journal of Open Research Software* 1:e6, DOI: http://dx.doi.org/10.5334/jors.ae

SOFTWARE METAPAPER

IJBlob: An ImageJ Library for Connected Component Analysis and Shape Analysis

Thorsten Wagner[1] and Hans-Gerd Lipinski[1]

[1] Biomedical Imaging Group, Dortmund University of Applied Sciences and Arts, Dortmund, Germany

The IJBlob library is a free ImageJ library for connected component analysis. Furthermore, it implements several contour based shape features to describe, filter or classify binary objects in images. Other features are extensible by the IJBlob extension framework. Because connected component labeling is a fundamental operation in many image processing pipelines (e.g. pattern recognition), the library could be useful for many ImageJ projects. The library is written in Java and the recent release is available at https://code.google.com/p/ijblob.

Keywords: connected component analysis, shape analysis, image processing, bioinformatics, imagej

Funding statement
This paper was supported by Grants of the German Federal Ministry of Education and Research (BMBF, FKZ 17PNT026).

(1) Overview

Introduction

Connected Component Analysis is a central part of many image processing pipelines. After object segmentation (e.g. by grey tone thresholding), the image is divided into foreground (object) and background pixels. For further processing, the connected regions of foreground pixels (called "blobs") often have to be "extracted." This process of labeling regions of connected foreground pixels is called "connected component labeling." IJBlob implements the contour labeling algorithm described by Chang et al. [1] and each region is described by its outer contour and inner contours (holes). This is useful since as soon as each object is defined individually by its contours, its properties can be easily described and processed separately. For this purpose, IJBlob provides some shape features to describe a blob like circularity, thinness ratio, elongation, orientation, Feret's diameter, fractal box dimension, and the number of holes and region-based moments. It is possible to extend the provided features by its own features using the IJBlob extensions framework. With the inbuilt filter functionality, the connected region in an image scene can be filtered by the inbuilt and extended features. An ImageJ Plugin provides a GUI (figure 1) for this functionality and is also available at http://code.google.com/p/ijblob. It is worth noting that the plugin also works with the ImageJ [6] distribution Fiji [7] .

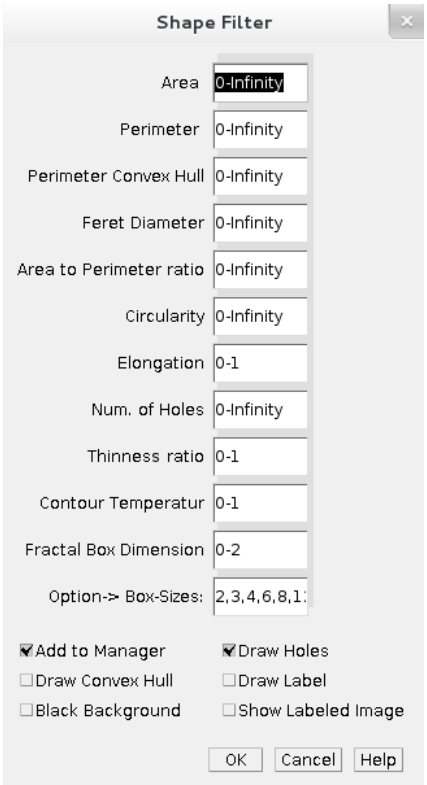

Fig. 1: The Plugin, which provides an interface for the filter functionality of the IJBlob library.

Filtering Blobs by Shape Features

If the filter is applied to a binary scene of several blobs, a resulting image is generated containing a set of remaining blobs. For each remaining blob, the features are calculated and filled into an ImageJ results table. A simple example illustrates the principal usage. In the following figure, all blobs with less than two holes should be removed.

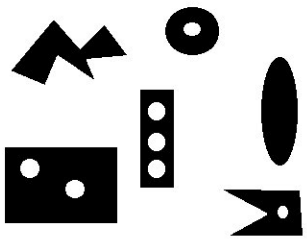

Fig. 2: The input image with some foreground objects (blobs).

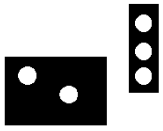

Fig. 3: The result image after all blobs with less than two holes are removed.

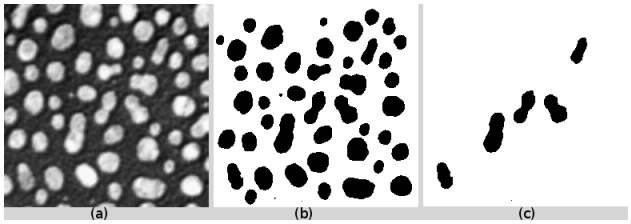

Fig. 4: Filtering of elongated blobs. The sample image "blobs.gif" of ImageJ (Fig. 4a) was segmented with a simple threshold, and the border blobs were removed (Fig. 4b). Then, the Shape Filter was parameterized for elongated objects by setting the option "Elongation" to "0.7-1" (Fig. 4c).

The option 'Num. of Holes' is therefore set to '2-infinity.' Figure 3 gives the resulting image after the filter has been applied.

Figure 4 demonstrates the filtering of elongated blobs. The ImageJ sample image (Fig. 4a) is filtered with the objective of retaining only elongated blobs. In the first step, the image is segmented by thresholding it. Second, the border objects are removed because their shape cannot be meaningfully described. Finally, the Shape Filter is applied using an elongation range of 0.7-1.

In our working group the IJBlob library is a basic part of the development of new ImageJ Plugins for (i) diffraction pattern analysis of freely diffusing, as well as sedimented (possibly agglomerated), nanoparticles [2][3], (ii) leucocyte classification (in preparation) and (iii) particle form analysis [4].

Implementation and Architecture

The library was implemented in Java (1.7) using ImageJ Version 1.47h. The connected component labeling algorithm of Chang et al. [1] is used to identify connected components. Most of the shape features are defined in Luciano's "Shape Classification and Analysis" [5]. There is direct dependence between IJBlob and ImageJ, because the library works with the internal image representation used by ImageJ. However, the image representation is simple, making it easy to develop a simple converter which allows for the use of IJBlob in other projects. The features provided by IJBlob are:

- Centre of Gravity
- Enclosed Area
- Perimeter
- Perimeter of the Convex Hull
- Circularity & Thinness Ratio
- Feret Diameter
- Area/Perimeter Ratio
- Temperature of the Outer Contour
- Fractal Box Dimensions
- Moments and Central Moments
- Eigenvalue Major & Minor Axis
- Elongation
- Orientation
- Outer Contours as Freeman-Chain-Code

Please note, that all the features are defined online (https://code.google.com/p/ijblob/wiki/BasicFeatures).

The IJBlob extension framework allows for the increase of existing features by its own features. Therefore, a class has to be derived from the CustomBlobFeature class. The class can also contain multiple features. Using the "getBlob()" method, it is possible to obtain a reference to the Blob and gain full access to the contour data and the other features of the Blob. Finally, an instance of the feature class has to be added to the Blob class using the static method "addCustomFeature(...)." The feature is now accessible using the Blob method "evaluateCustomFeature." If the return type of the custom feature methods is "Integer" or "Double," these features could be used with inbuilt filter functionality.

Quality control

Unit and functional testing have been carried out in Archlinux (64 bit), Ubuntu 13.04 (32 bit / 64 bit), Windows XP (32 bit), Windows Vista (32 bit) and Windows 7 (64 bit).

(2) Availability

Operating system

Linux, Windows XP / Vista / 7

Programming language

Java 1.7

Additional system requirements

up-article-text-paragraph

Dependencies

ImageJ 1.47

List of contributors

- Thorsten Wagner
- Hans-Gerd Lipinski

Archive

Name
Google Code

Persistent identifier
https://code.google.com/p/ijblob/downloads/list

License
GNU General Public License version 3

Publisher
Thorsten Wagner

Date published
11/02/2012

Code Repository

Name
Google Code

Identifier
http://ijblob.googlecode.com/svn/

License
GNU General Public License version 3

Date published
11/02/2012

Language
English

(3) Reuse potential

Connected component labeling is a central part of many image processing pipelines, such as object tracking, object matching or classification. Due to the broad range of possible application fields, the reuse potential seems to be high. Possible "general" fields are:

- Object tracking in image series: In segmented image series, it is often required to track an object over time [2][3] . Therefore, it is necessary to link an object from image to image. This can be done using the (i) distance (centroid to centroid), (ii) shape features or (iii) other texture based features. Imagine an image scene composed of moving rectangles and a single moving circle. It is easy to identify and track the circle using the circularity feature of its blob.
- Classification of blobs using their shape features: It is necessary to find suitable features for classification of the blobs. It is possible to use the inbuilt shape features of IJBlob or use own features via the extension framework for that purpose.
- Removing segmentation artefacts using shape features, e.g. removing blobs which are too elongated to be a "blob of interest": This can be seen as a kind of classification; one class is "objects of interest" and the other is "artefacts."
- Searching for all blobs with similar features (see the example presented in the introduction).

References

1. **Chang, F, Chen, C** and **Lu, C** 2004 "A linear-time component-labeling algorithm using contour tracing technique". *Computer Vision and Image Understanding* 93(2): 206-220, DOI:10.1016/j.cviu.2003.09.002
2. **Wagner, T, Luettmann, S O, Swarat, D, Wiemann, M** and **Lipinski, H** 2011 "Image analysis of free diffusing nanoparticles in vitro". *SPIE* 8087*:* 808726-808726-8, DOI:10.1117/12.889620
3. **Wagner, T, Wiemann, M, Schmitz, I** and **Lipinski, H** 2013 "A Cluster-Based Method for Improving Analysis of Polydisperse Particle Size Distributions Obtained by Nanoparticle Tracking". *Journal of Nanoparticles* 2013:1-9, DOI:10.1155/2013/936150
4. **Schippritt, D, Wiemann, M** and **Lipinski, H** 2010 "Sedimentation of agglomerated nanoparticles under cell culture conditions studied by image based analysis". *SPIE* 7715: 77153A-77153A-9, DOI:10.1117/12.854313
5. **da Fona Costa, L** 2009 Shape Classification and Analysis: Theory and Practice, Second Edition (Image Processing Series). CRC Press.
6. **Schneider, C A, Rasband, W S** and **Eliceiri, K W** 2012 "NIH Image to ImageJ: 25 years of image analysis". *Nature Methods* 9(7): 671-675, DOI:10.1038/nmeth.2089
7. **Schindelin, J, Arganda-Carreras, I, Frise, E, Kaynig, V, Longair, M, Pietzsch, T, Preibisch, S, Rueden, C, Saalfeld, S, Schmid, B, Tinevez, J Y, White, D J, Hartenstein, V, Eliceiri, K, Tomancak, P** and **Cardona, A** 2012 "Fiji: an open-source platform for biological-image analysis". *Nature Methods* 9(7): 676-682, DOI:10.1038/nmeth.2019

How to cite this article: Wagner, T and Lipinksi, H 2013 IJBlob: An ImageJ Library for Connected Component Analysis and Shape Analysis. *Journal of Open Research Software* 1:e6, DOI: http://dx.doi.org/10.5334/jors.ae

Published: 14 October 2013