Blob Detection

Anne Kaspers
Biomedical Image Sciences
Image Sciences Institute
UMC Utrecht

Abstract

For many high vision purposes, detecting low-level objects in an image is of great importance. These objects, which can be 2D or 3D, are called blobs. Blobs appear in different ways depending on their scale and can be detected using local operations in a multi-scale representation of the image. This paper describes several blob detection methods and applications and tries to make a fair comparison without performing experiments. It shows that blobs can be defined and localized in different ways and that each method has its own strength and shortcomings.

1 Introduction

Automatic detection of blobs from image datasets is an important step in analysis of a large-scale of scientific data. These blobs may represent organization of nuclei in a cultured colony, homogeneous regions in geophysical data, tumor locations in MRI or CT data, etc. This paper presents several approaches for blob detection and applications.

Before going into detail on blob detection, first some definitions of a blob are given. Lindeberg [10] defines a blob as being a region associated with at least one local extremum, either a maximum or a minimum for resp. a bright or a dark blob. Regarding the image intensity function, the spatial extent of a blob is limited by a saddle point, a point where the intensity stops decreasing and starts increasing for bright blobs and vice versa for dark blobs. A blob is represented as a pair consisting of one saddle point and one extremum point.

Hinz [8] just describes a blob as a rectangle with a homogeneous area, i.e. a constant contrast, which becomes a local extremum under sufficient amount of scaling.

Rosenfeld et al. [13] defines a blob as a crossing of lines perpendicular to edge tangent directions, surrounded by 6 or more directions, like in the following picture:

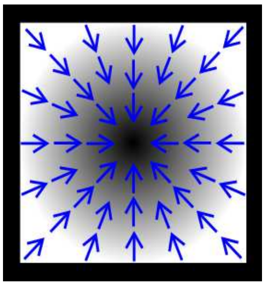


Fig. 1. A blob surrounded by 8 different directions taken from www.wikipedia.org

A third definition of a blob is given by Damerval [4], describing blobs as the largest modulus maxima of the continuous wavelet transform (CWT, see Appendix) along some maxima of interest. The CWT is able to construct a time-frequency representation, offering a good localization of frequencies and time (scale). The exact meaning of modulus maxima and maxima of interest is explained later in the section of the concerning method.

To this point only 2D blob definitions are mentioned. Yang and Parvin [17] give a definition of a 3D blob as being elliptic features in scale-space portioned by a convex hull (boundary of the minimal convex set containing a set of voxels belonging to a blob).

Blobs occur in many shapes and places. For instance, blobs can be found in an image of sunflowers, zebra fish neurons or in an image of a hand. Below, a number of example images are shown.

Fig. 2. A Sunflower field (taken from [10])

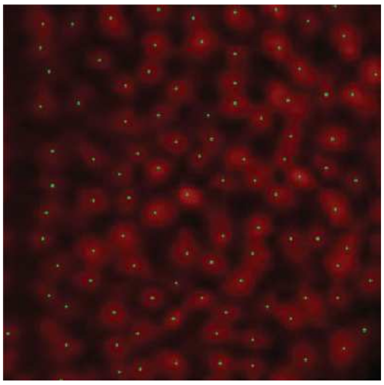


Fig. 3. A microscopic zebra fish image overlaid by green dots corresponding to blobs (taken from [9])

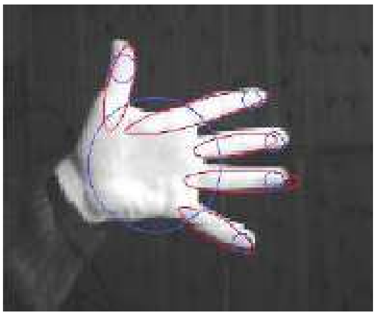


Fig. 4. A hand overlaid by a multiple of blobs and ridges (taken from 15])

Hand and finger blobs are defined in a different way than sunflower blobs. For different kinds of blobs, different detection methods are needed. These methods must fulfill a number of requirements:

- reliability / noise insensitiveness: a low-level vision method must be robust to noise,
- accuracy: in vision metrology highly accurate results are needed,
- scalability: blobs of different sizes should be detected,
- speed: it should be near to real-time processing,
- few and semantically meaningful parameters: it should be easy to understand for adjustment,
- capability of extracting geometric and radiometric attributes; this is needed for blob classification

The organization of the paper is as follows: In the next section, a number of distinctive blob detection methods will be presented and in the following section applications of blob detection methods will be described. After that, the methods and applications will be evaluated using (some of) the requirements mentioned above. The discussion section concludes the paper.

2 Methods

2.1 Prerequisites

A majority of the blob detection methods are based on the scale-space representation. The main purpose of the scale-space representation is to understand image structure at all levels of resolution simultaneously and link images at successive scales. Scale-space is derived by applying a smoothing kernel -like the Gaussian – (see Appendix)on the image with a scale parameter depending on the amount of smoothing of finer image structures. A blob with linked extrema over scale is called a scale-space blob. The difference in scale where a blob shows up and disappears is called the scale-space lifetime of a blob. The scale at which a maximum over scales is attained will be assumed to give information about how large a blob is.
Blob detectors can be based on image gradients (contrast), eigenvalues or templates. Gradient magnitudes are used to detect blob outlines, eigenvalues to specify the length, width and orientation of blobs and templates to match a blob to a predefined shape. More details will follow in the method sections.

2.2 Template Matching

A fast and robust method for detecting blobs is template matching [16]. A template of a blob is moved over the search image and blobs are detected where the template matches a part of the image. The following steps are performed:

1. Overlay the template on the initial image position (0,0).
2. Calculate the sum of squared differences (SSD) or the sum of absolute differences (SAD) for the overlaid area and store it in a correlation matrix.
3. Move on to the next image position and repeat step 2 until the final image position is reached.

Bright spots in the correlation image correspond to probable blob locations. By defining a threshold,

an exact number of blobs and exact locations can be used as result.
When the template covers pixels outside the image, those values could be calculated by mirroring or extrapolation (see Appendix). Otherwise, the template positions could be restricted to positions with template coverage within the image.
Small templates can be used to detect primitive blobs while large templates can detect specifically shaped blobs. To get a more flexible blob detection, multiple templates could be designed.

2.3 Watershed Detection

The watershed method [19] assumes an image to be grey value mountains and simulates the process of rain falling onto the mountains, running down the mountain range and accumulating in basins. This process is repeated until all basins are filled and only the watersheds between different basins remain. These watersheds correspond to bright blobs, whereas dark blobs can also be obtained by taking the gradient amplitude image. This flooding process is performed on the gradient image, i.e. the basins should emerge along the edges. Normally, this algorithm will lead to an oversegmentation of the image, especially for noisy image material, e.g. medical CT data. Either the image must be pre-processed or the regions must be merged on the basis of a similarity criterion afterwards.

2.4 Spoke Filter

An early (single scale) blob detector which detects blobs of various sizes is the spoke filter, also called Adaptive Spatial Erosion Filter, proposed by Minor and Sklansky [18]. It is applied as follows:

1. Apply edge filters to extract local edge elements of all (8) orientations.
2. Mark pixels as “interior”, which lie within a certain distance of an edge element on a line perpendicular to the edge tangent direction.
3. Mark *spoke crossings* as being interior pixels marked by edge elements of different orientations.
4. Mark blobs as being crossings marked by 6, 7 or all 8 directions.

By varying the distance, blobs of various sizes can be detected.

This method is extended to multi-scale by Rosenfeld et al. [13]. He defines an intensity pyramid as a set of fine to coarse resolution images. At each level, the spoke filter is applied to detect blobs. Alternatively, for each image in the intensity pyramid, the edge elements can be calculated and summed for all images. For the summed gradient image, step 2 to 4 of the spoke filter algorithm can be followed to detect blobs at multiple scales.

2.5 Automatic scale selection

Most blob detection applications are based on Lindeberg’s method for automatic scale selection [11]. The principle for automatic scale selection is: in the absence of other evidence, assume that a scale level, at which some combination of normalized derivatives assumes a local maximum over scales, reflects the size of the corresponding blob. Scale levels are obtained by Gaussian smoothing. The Gaussian function meets the requirement that no details are generated when resolution decreases and it provides simpler pictures at coarse scales [5].
Lindeberg presents two combinations as basic blob detectors to be used in Gaussian scale-space: the Laplacian and the Monge-Ampère operator.
The Laplacian operator is defined as the trace of the Hessian matrix, which is the square matrix of second-order partial derivatives of the image function (see Appendix). By multiplying the trace with a scale parameter, the Laplacian operator can be used to detect scale-space maxima.
The Monge-Ampère operator is defined as the scale-normalized determinant of the Hessian matrix. The scale parameter is multiplied twice to obtain scale invariance.
Lindeberg claims, that maxima over scales have a nice behavior under rescalings of the intensity pattern: if an image is rescaled with a constant factor, than the scale at which the maximum is assumed, will be multiplied with the same factor. This guarantees that image operations transform with size variations.
In practice, blobs may be detected at coarse scales, and the localization properties may not be the best. Therefore a coarse-to-fine approach is needed to compute more accurate localization estimates.

2.6 Sub-pixel precise blob detection

Hinz [8] presents such an approach for accurate localization, build upon Gaussian scale-space. The blob is defined as a rectangle with constant contrast, which becomes a local extremum under sufficient Gaussian smoothing. Because this approach is applied to images containing equally aligned objects, like windows or cars, a rectangle is used as basic model of the blob. The basic idea of the algorithm is as follows:

1. Initialize the expected rectangle orientations of the shorter and larger side.
2. Calculate the Hessian matrix, the eigenvector and the direction of the rectangle's shorter side using Gaussian smoothing with an 1D kernel in the expected orientation of the shorter side of the rectangle.
3.
    a. Compute the curvature maximum along the direction of the rectangle's shorter side using the profile along the direction of the larger side.
    b. Analyze the gradients of the used profile to determine bias and remove it.
4.
    a. Compute the curvature maximum along the direction of the rectangle's larger side using the profile along the direction of the shorter side.
    b. Analyze the gradients of the used profile to determine bias and remove it.
5. Reconstruct the rectangle's center point from both profiles.

In addition, this method provides a way to construct the boundary of the blob and approximate the boundary by an ellipse. For blob classification, it can extract attributes like the blob's boundary length, area, geometric moments and the parameters of the fitted ellipse.

2.7 Effective maxima line detection

Damerval [4] presents a method where connected curves of modulus maxima at different scales - called maxima lines - are effectively selected, to divide blobs from noise. The selection of maxima lines is performed by the following steps.

1. Compute the 2D Gaussian scale-space.
2. Compute modulus maxima at every scale.
3. Connect modulus maxima in adjacent scales that are close to each other and have the same sign (plus or minus) to obtain maxima lines.
4. Remove maxima lines that consist of coefficients that increase on average when scale decreases; they associate to noise.
5. Remove maxima lines that do not cross at least 5 integer scales; they associate to white noise.
6. Compute the global maximum for each maxima line and remove maxima lines which deviate at scales larger than the global maximum scale; they associate to blob structures outside the blob boundary.
7. Join maxima lines that cross in scale-space; the blob location is given by the cross point and its characteristic scale by the median of the global maximum scales of all joined maxima lines.

2.8 Confidence Measurement

Forssén and Granlund [6] present a rather complex multi-scale method to extract blobs from an image. It is not based on linear Gaussian smoothing, as the previous methods. The full algorithm is complex and can be summarized as follows:

1. The image is first converted into channel images using a set of windowed cosine kernel functions.
2. For each of the images, a low-pass pyramid is generated
3. Because the filter sums up to 1, a threshold of 0.5 is used to obtain binary confidence values, resulting in a clustering pyramid.
4. The image is pruned by deleting similar clusters that lie on top of each other.
5. The pixels left in the pyramid are used as seeds for region growing resulting in a region image.
6. For all regions, the raw moments of order 0 to 2 are computed to approximate blobs by ellipses.

This results in an image of ellipses with different sizes and orientations, overlapping each other.

2.9 Summary

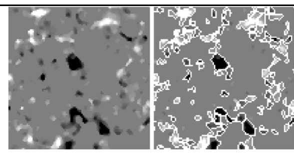
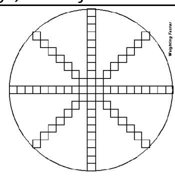
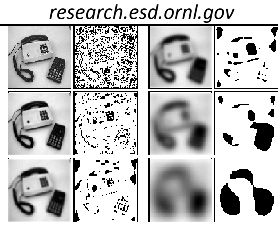
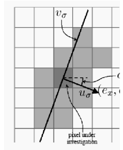
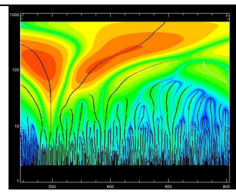
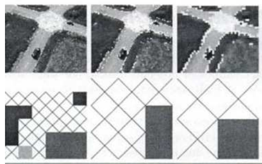
Method	Blob Definition	Blob/Noise Threshold	Multi-scale	Automatic	Speed	Accuracy	Image
Template matching	An exact shape on a template	A threshold on the SSD or SAD	No	Semi-automatic: the template must be designed, but the matching step is automatic	Very fast	High	 *Detection of specific case, taken from Mathworks.com*
Watershed detection	A watershed between different basins	An image intensity threshold	No	Yes	Fast	High	 *Detected elements in a space image, taken from aanda.org*
Spoke filter	A crossing of lines perpendicular to edges	A minimum of 6 crossings	No, but can be exten-ded to multi-scale	Yes	Fast	High	 *A spoke crossing, taken from research.esd.ornl.gov*
Automatic scale selection	A local maximum of a normalized derivative over scales	A threshold on the scale-space lifetime and operator response	Yes	Mostly, only the number of blobs must be set manually	Quite slow (multi-scale)	Reasonable	 *Levels in scale-space [10]*
Sub-pixel precise blob detection	A small, compact image primitive, defined as a rectangle with constant contrast	A blob has to turn in a local extremum under sufficient amount of smoothing	Yes	Mostly, only the detection scale must be set manually	Quite Slow	Very high	 *Detected rectangle center point and orientation [8]*
Effective maxima line detection	Modulus maxima connected in scale-space	A Maxima line selection step	Yes	Yes	Quite Slow	Reasonable	 *Wavelet maxima lines [4]*
Confidence measure-ment	A homo-geneous region with a clear boundary	The confidence and a threshold on area size	Yes	Mostly, only the minimum area size must be set manually	Quite Fast	Reasonable	 *Part of a clustering pyramid [4]*

Table 1. Summary of Blob Detection Methods

3 Applications

3.1 Nodule detection

A medical application for nodule detection is presented in *Multi-scale Nodule Detection in Chest Radiographs* by Schilham et al. [14], where pulmonary nodules are detected in thorax x-ray images using a Computer Aided Diagnosis (CAD) scheme.
The multi-scale blob detection is performed on a lung image, using the Laplacian and gamma-normalized derivatives. Detected blobs are represented using spatial and geometrical features which are used to train a classifier.

3.2 Lesion detection

In *Detection and Characterization of Unsharp Blobs by Curve Evolution* by Gerig et al. [7], an application, based on geometric heat flow (see Appendix) is represented for detecting lesions in MR images of Multiple Sclerosis patients. These lesions are visible in MR images as unsharp blobby structures. Analysis of level curves of the image intensity function makes it possible to identify and characterize these structures. In this way, lesions are identified as collections of concentric, simple closed curves. A so-called onion-skin model is performed, where segmentation is formulated as search for concentric curves organized as layers of the onion-skin. The detected lesions are stored using two parameters: their center and their radial intensity function.

3.2.1 2D Nuclei detection

An application for 2D nuclei detection is presented in *Detection of Blob Objects in Microscopic Zebrafish Images Based on Gradient Vector Diffusion* by Li et al. [9], where nuclei, cells and neurons need to be detected in microscopic zebrafish images.
The method consists of three key steps: diffusion of gradient vector field with elastic deformable model, computing the flux image and finally post-procesing, using a threshold and non-maximum suppression.
In the first step, the direction of image gradients provides important information for blob detection. In the second step, the flux of the diffused gradient field is computed, where blobs are detected as points where the divergence is negative and diffused gradient flow is directed to, comparable with the spoke filter (but much more complicated). Finally the flux image is reversed, because blobs correspond to local minima, and non-maximum suppression and a threshold is used to detect candidate blobs.

Another 2D nuclei detection method for counting cell nuclei in tissue sections is presented in *Automated Tool for the Detection of Cell Nuclei in Digital Microscopic Images: Application to Retinal Images* by Byun et al. [2]. It is based on scale-space blob detection, for it uses a Laplacian of Gaussian (LoG) filter. Its purpose is to count death cells in the inner nuclear layer of retina images.
The method mainly consists of two steps:

1. Blob detector design. The LoG filter is designed such that the diameter of the filter is proportional to the average diameter of nuclei in the image. After filtering the given retinal image, local maxima will correspond to blob centers.
2. Searching for local maxima. The minimum distance between blob centers is used as the search radius for the local maxima.

This method consists of two parameters which have to be manually set: the cell size and the minimal distance between cells.

3.2.2 3D Nuclei detection

In *CHEF: Convex Hull of Elliptic Features for 3D Blob Detection* by Yang and Parvin [17], an application for detecting 3D nuclei is given, based on Gaussian smoothing. Blobs are classified as elliptic features in scale-space and then grouped into 3D connected components, divided by their convex hulls. This method called CHEF (Convex Hull of Elliptic Features) is applied to a database of multi-cellular systems for detailed quantitative analysis.
CHEF is applied as follows (see next page):

1. The scale and the number of planes of the convex hull (N) are selected.
2. The original image is convolved with the Gaussian kernel at the selected scale.
3. The bright elliptic features are computed using a Hessian matrix to detect and classify each point in the image. A negative (positive) Hessian classifies a bright (dark) elliptic feature.
4. The bright elliptic features are decomposed into disjoint connected components.
5. For each component, the convex hull is computed.
6. Components are merged if they overlap. If no merging appears, then stop; otherwise go to step 4.

For construction of the convex hull, the size of the known objects of interest is the key parameter; it is determined empirically.

3.3 Human body part detection

An application for body part detection is presented in *Blob Analysis of the Head and Hands: A Method for Deception Detection* by Lu et al. [12]. It is applied on 2D video images and it tries to capture the location and movements of head and hands to identify behavioral states.
The position, size and angle of head and hands are tracked using color analysis, eigenspace-based shape segmentation and Kalman filters. Hand and face regions are detected as blobs using a 3D Look-Up-Table (LUT) of skin color samples. Regions that are incorrectly identified because their color matches to the skin color, are disregarded through fine segmentation and comparing the subspaces of the face and hand candidates.

Another application for body part detection is given in *Real-Time Markerless Human Body Tracking Using Colored Voxels and 3-D Blobs* by Caillette and Howard [3]. The system first reconstructs a 3D voxel-based representation of a person from images of multiple cameras, and then matches a kinematic model in 3D space. Voxels are classified from pixel-samples taken inside the 2D projections onto the surface planes. To attain real-time performance, a measure of the distance to the background model is computed for each 2D sample. Voxels are then classified from statistics on these distances across the available views, as discarded, subdivided or retained as belonging to the foreground.

3.4 Summary

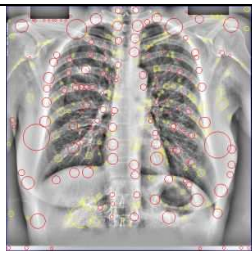
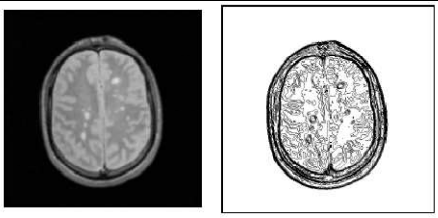
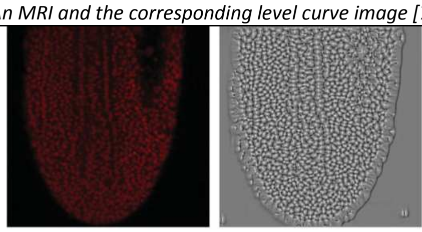
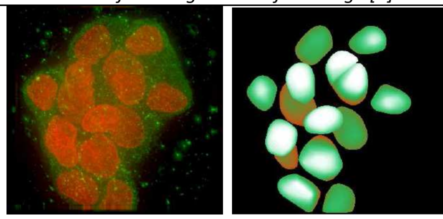
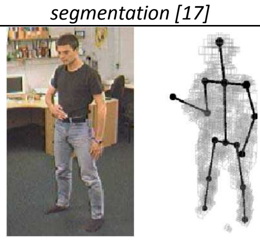
Application	Basic Method	Extension	Image
Nodule detection	Laplacian of Gaussian	Pre-processing: contrast enhancement with a local normalization filter Post-processing: blob boundary extraction by multi-scale edge focusing	 *Detected nodules, taken from isi.uu.nl/Research/Gallery/noduledetection*
Lesion detection	Geometric heat flow	Pre-processing: selecting samples and discretizing spatial and scale parameters.	 *An MRI and the corresponding level curve image [7]*
2D Nuclei detection	Laplacian of Gaussian	Method: construction of a flux image Post-processing: reversing flux image, non-maximum suppression and a threshold step	 *A zebrafish image and its flux image [9]*
3D Nuclei detection	Laplacian of Gaussian	Pre-processing: selecting scale and number of planes of the convex hull Method: computing Hessian matrix and convex hulls Post-processing: splitting and merging of convex hulls	 *One view of a 3D nuclei image and the result of segmentation [17]*
Human Body part detection	Skin color sampling	2D [12] Method: eigenspace-based shape segmentation and Kalman filters 3D [3] Post-processing: splitting and merging of blobs	 *A person and its fitted model [3]*

Table 2. Summary of Blob Detection Application

4 Evaluation

4.1 Sensitivity and specificity

The quality of Blob detectors are measured and compared using sensitivity and specificity.
Sensitivity is measured by dividing the number of true positive blob candidates by the sum of true positive and false negative blob candidates. A sensitivity of 100% means that the detector recognizes all blobs that are actually there in the image.
Specificity is measured by dividing the number of true negative blob candidates by the sum of true negative and false positive blob candidates. 100% specificity means that all detected blob candidates are in fact blobs.

By plotting the sensitivity vs. 1 – the specificity, a ROC curve is created, for which an optimal combination of sensitivity and specificity can be chosen.
To be able to measure the sensitivity and specificity, a golden standard is needed. Only in the multi-scale nodule detection method, a golden standard was available. So the blob detectors need to be evaluated in another way.

4.2 Experiments

Another way of comparing the quality of blob detectors is by performing all detectors on the same set of images. This is however not feasible for two reasons.
First, there are different blob definitions, as mentioned in the introduction. Therefore one detector looks for other kind of blobs, for example spots, while another detector looks for specific template blobs. Comparing these results would lead to nothing.
Second, performing experiments are not part of the literature study. Hence, comparison can only be done based on existing experiments and what the authors claim.
For the 7 blob detector, the pros and cons, which are claimed by the authors, in the following table:

Method	Pros	Cons
Template matching	• Exact blob matching possible • Very fast • Easy to implement	• Restricted blob definition • Manual template design • Single-scale • Translation, rotation and scaling only possible by designing extra templates
Watershed detection	• Works good for blobs with homogenous contrast and a clear edge • Near real-time performance • Automatic • Can extract bright and dark blobs	• Very sensitive to noise, which leads to oversegmented results • Single-scale • No shape and size information available
Spoke filter	• Works good for blobs of different sizes with clear consistently oriented edges • Extension to multi-scale possible • Extension works well for blobs with non-homogeneous background • Fast • Automatic • Single-scale is easy to implement	• Very dependent on blob boundary orientation; works only for perfect boundaries with edge directions crossing in the middle of the blob • Only perfect circular blobs are detected
Automatic scale selection	• Works well for blobs of all sizes • Invariant under scaling, translation and rotation • Insensitive to noise because of significance measurement and scale-space lifetime threshold • Size measurement possible • Automatic	• Works not well for blobs with unsharp boundaries • No accurate extraction of the blob position and size • Rather slow because multiple scales are analyzed • Only circular blobs are detected
Sub-pixel precise blob detection	• Works well for circular and elliptic blobs • Sub-pixel precise detection • Extraction of several useful blob attributes possible • Blob classification possible	• Scale-space parameter needs to be manually adjusted for detecting blobs of different sizes • Works not well on blobs with lots of different sizes • Difficult to implement • Slow when blobs with lots of sizes needs to be detected
Effective maxima line detection	• Works well for blobs of all sizes • Invariant under scaling, translation and rotation • Exact blob size computation possible • Robust to noise	• Same as for automatic scale selection
Confidence measurement	• Works well for homogenous blobs of all sizes • Overlapping blobs and blobs within blobs can also be detected • Proven noise insensitiveness for salt and pepper noise, white rectangular noise and 1/f noise	• Works not well for inhomogeneous blobs • Not fully evaluated • Hard to implement • Detects a lot of small blobs within bigger blobs

Table 3. Pros and cons of Blob Detection Method

For different kind of blob images, different methods would be preferable. To get a proper qualification (and comparison) of the methods, without doing actual experiments, the best detector for a set of different images is selected, based on the qualifications that are given by their authors. In the next table, the 'test' images are given, together with their best and worst blob detector. The images are all taken from en.wikipedia.org.

Image	Best	Worst
	Automatic scale selection and effective maxima line selection. They work well on circular blobs of different sizes.	Watershed detection and spoke filter. They cannot cope with different blob sizes. The watershed detector would have difficulties with the inhomogeneous center of the sunflowers. The spoke filter can only handle perfectly circular blobs, which is not the case in this real world example.
	Confidence measurement. It works well on homogenous blobs of all sizes even if they overlap.	Sub-pixel precise blob detection. The stars differ a lot in size while the scale parameter cannot be infinitely changed.
	Sub-pixel precise blob detection. It works well on blobs that are equally sized and rectangular shaped, because in the method a scale parameter is set and a rectangle serves as basic primitive.	The spoke filter. A spoke crossing has to consist of at least 6 edge directions, while a rectangle only has 4 edges.
	Template matching. The bricks are easily detected using templates. Also confidence measurement can work reasonable after merging.	Automatic scale selection an effective maxima line selection. The inhomogeneous contrast would lead to detection of lots of blobs on the brick.
	Watershed detection. It works well for detection of homogeneous blobs of all shapes and sizes.	The spoke filter. It works badly on non-circular shaped blobs. Template matching. All shapes are different and need a separate template.

Table 4. Blob images and their most and less preferable detector

With this knowledge, the following qualifications can be made:

- In most cases, the spoke filter and template matching are not preferable. They are both too restricted. The spoke filter is made to detect perfectly circular blobs, which cannot be found in real world data. Template matching detects only specific blobs, while most images contain a variety of blob shapes. Even windows on a building are difficult to detect because they are slightly different in scale and rotation.
- Watershed detection is more flexible, because it can detect blobs of all sizes, but it is very sensitive to noise. Only inhomogeneous blobs without noise (local maxima) are perfectly detected.

- Sub-pixel precise blob detection is very accurate in detecting blobs and its sizes and positions. It is robust to noise, since it smoothes the image using a Gaussian function. However, it is not flexible in detecting blobs of all sizes, since it has a scale parameter which has to be changed for each blob size.
- Automatic scale selection and effective maxima line selection work both well for circular blobs of different sizes. They are robust, because noise is filtered out based on their short scale-space lifetime. Their bottleneck is the circularity restriction.
- Most flexible is confidence measurement. It works well on homogenous blobs of all sizes even if they overlap.

5 Discussion

In this paper, a number of blob detection methods and applications described. The blob detection methods have been compared and qualified, without experiments. The blob detection applications mostly contain specific methods for their purpose. No qualification of those methods is made because only globally designed blob detection methods are relevant in this paper.

Strictly spoken, the Laplacian and Monge-Ampère operator, used in automatic scale selection and effective maxima line selection, are the basic blob detectors. Automatic scale selection and effective maxima line selection are algorithms that use those operators. However, an algorithm that detects blobs is in definition a blob detector.

Because of lack of experiments, no hard qualification of the blob detectors could be made in terms of sensitivity and specificity. Using the authors claim and the results of their experiments, a comparison was performed. There is not a detector that is best for all cases. Every case needs a consideration of which detector would be most suitable. The most flexible detector seems the confidence measurement; however it needs to be evaluated.

The concept of automatic scale selection is widely used in applications and as base for other blob detectors. It has a great advantage that blobs of all sizes are detected at their own scale, which can be used to calculate their sizes. The difficult algorithm and the lack of evaluation of confidence measurement could be a reason it is not so popular.

In future, experiments could be done on test images, like the one in the table, to make a proper comparison between the methods. For now, this paper could serve as guide to make a choice of the right blob detector for the right application.

6 References

[1] L. Bretzner and T. Lindeberg. Feature tracking with automatic selection of spatial scales. Computer Vision and Image Understanding, 71(3):385–392, 1998.

[2] J. Byun, M. R. Verardo, B. Sumengen, G. P. Lewis, B. S. Manjunath and S. K. Fisher. Automated tool for the detection of cell nuclei in digital microscopic images: application to retinal images. Mol Vis, 12:949–960, 2006.

[3] F. Caillette and T. Howard. Real-time markerless human body tracking using colored voxels and 3-D blobs. In Proc. ISMAR, 266–267, 2004.

[4] C. Damerval and S. Meignen. Blob detection with wavelet maxima lines. IEEE Signal Processing Letters, 14(1):39–42, 2007.

[5] L. M. J. Florack. Image Structure. Kluwer Academic, 1997.

[6] P.-E. Forssén and G. Granlund. Robust multi-scale extraction of blob features. In Proceedings of the 13th Scandinavian Conference on Image Analysis, LNCS 2749, 11–18, 2003.

[7] G. Gerig, G. Szekely, G. Israel and M. Berger. Detection and characterization of unsharp blobs by curve evolution. In Proc. of Information Processing in Medical Imaging, 165-176, 1995.

[8] S. Hinz, Fast and Subpixel Precise Blob Detection and Attribution. In: Proceedings ICIP'05, Genua. pp. 457–460, 2005.

[9] G. Li, T. Liu, J. Nie, L. Guo, J. Malicki, A. Mara, S. A. Holley, W. Xia and S.T. Wong. Detection of blob objects in microscopic Zebrafish images based on gradient vector diffusion. *Cytometry A*, 71:835–845, 2007.

[10] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. International Journal of Computer Vision, 11(3):283-318, 1993.

[11] T. Lindeberg. Feature detection with automatic scale selection. International Journal of Computer Vision, 30(2):79-116, 1998.

[12] S. Lu, G. Tsechpenakis, D. N. Metaxas, M. L. Jensen, and J. Kruse. Blob Analysis of the Head and Hands: A Method for Deception Detection, Hawaii International Conference on System Science (HICSS'05), 2005.
[13] A. Rosenfeld and C.-Y. Sher. Detecting image primitives using feature pyramids. Journal of Information Sciences, 107:127-147, 1998.
[14] A. M. R. Schilham, B. van Ginneken, and M. Loog, Multi-scale Nodule Detection in Chest Radiographs, Lecture Notes in Computer Science, vol. LNCS 2878, pp. 602-609, 2003.
[15] A. Shokoufandeh, S. Dickinson, C. Jonsson, L. Bretzner, and T. Lindeberg. On the Representation and Matching of Qualitative Shape at Multiple Scales. In Proceedings, European Conference on Computer Vision, 2002.
[16] M. Sonka, V. Hlavac, R. Boyle. Image Processing, Analysing, and Machine Vision. Thomson, 2008.
[17] Q. Yang and B. Parvin. Chef: Convex hull of elliptic features for 3D blob detection. In Proceedings of the International Conference on Pattern Recognition, 2002
[18] L. G. Minor and J. Sklansky. Detection and segmentation of blobs in infrared images. IEEE Tranactions on Systems Man and Cyberneteics, SMC-11(3):194–201, March 1981.
[19] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In International workshop on image processing, real-time edge and motion detection, 1979.

Appendix

A1. Wordlist

A1.1 Continuous wavelet transform

A continuous wavelet transform is used to divide a continuous-time function into wavelets. Unlike Fourier transform, the continuous wavelet transform possesses the ability to construct a time-frequency representation of a signal that offers very good time and frequency localization. In mathematics, the continuous wavelet transform of a continuous, square-integrable function x(t) at a scale a > 0 and translational value $b \in \mathbb{R}$ is expressed by the following integral

$$X_w(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (1)$$

Taken from en.wikipedia.org

where ψ(t) is a continuous function in both the time domain and the frequency domain called the mother wavelet and $*$ represents operation of complex conjugate. The main purpose of the mother wavelet is to provide a source function to generate the daughter wavelets which are simply the translated and scaled versions of the mother wavelet. To recover the original signal x(t), inverse continuous wavelet transform can be exploited.

A1.2 Extrapolation

To estimate (a function that is known over a range of values of its independent variable) values outside the known range. For example, when pixels add up with some number towards the border, values outside the border are estimated using the same addition.

A1.3 Flux

The flux is computed as:

$$Flux(\mathbf{x}) = \sum_{i=1}^{8} \langle \mathbf{N}_i, \mathbf{F}(\mathbf{x}_i) \rangle \quad (2)$$

, where $\mathbf{x}_i$ is a 8-neighbor of $\mathbf{x}$, and $\mathbf{N}_i$ is the outward normal at $\mathbf{x}_i$ on the unit circle centered at $\mathbf{x}$.

A1.4 Gaussian

The Gaussian Smoothing Operator performs a weighted average of surrounding pixels based on the Gaussian distribution. It is used to remove Gaussian noise and is a realistic model of defocused lens. The scale parameter (sigma) defines the amount of blurring. Large values for sigma will only give large blurring for larger kernel sizes.

The operator generates a kernel of values that are then applied to groups of pixels in the image. These template values are defined by 2D Gaussian Equation:

$$\frac{1}{2\pi\sigma^2} \exp\{-\frac{x^2+y^2}{2\sigma^2}\} \quad (3)$$

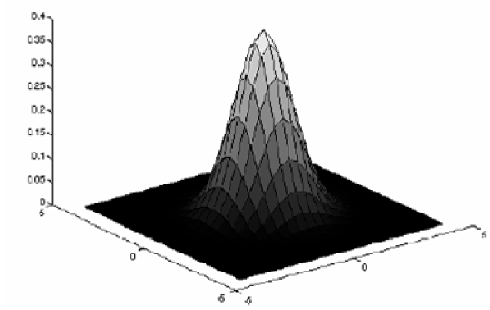


Fig. 5. The 2D Gaussian function, taken from en.wikipedia.org

A1.5 Hessian matrix

The Hessian matrix is the square matrix of second-order partial derivatives of a function; that is, it describes the local curvature of a function of many variables.
Given the real-valued function

$$f(x_1, x_2, \dots, x_n),$$

if all second partial derivatives of f exist, then the Hessian matrix of f is the matrix

$$H(f)_{ij}(x) = D_i D_j f(x)$$

where x = (x_1, x_2, ..., x_n) and Di is the differentiation operator with respect to the i_{th} argument and the Hessian becomes:

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

The Hessian is defined as the determinant of the above matrix.

A1.6 Mirroring

Copying values inside the image to their mirrored position outside the image. For example: 3, 2, 1 towards the image border would be continued with 1, 2, 3.