FAST AND SUBPIXEL PRECISE BLOB DETECTION AND ATTRIBUTION

Stefan Hinz

Remote Sensing Technology
Technical University Munich, Arcisstr. 21, 80 333 Muenchen, Germany
E-mail: Stefan.Hinz@bv.tum.de; URL: http://www.RemoteSensing-TUM.de

ABSTRACT

This paper introduces an algorithm for fast and subpixel precise detection of small, compact image primitives (”blobs”). The algorithm is based on differential geometry and incorporates a complete scale-space description. Hence, blobs of arbitrary size can be extracted by just adjusting the scale parameter. In addition to center point and boundary of a blob, also a number of attributes are extracted. These describe the specific blob characteristics in more detail and, thus, allow for a subsequent classification of blobs. Several examples on real images illustrate the performance of the proposed algorithm.

1. INTRODUCTION

1.1. Motivation and Requirements

The extraction of small, compact, bright or dark images primitives is one of the basic tasks of low-level components of an image understanding system. For some applications, these so-called ”blobs” are already the desired objects to extract (eyes, particles, fiducial marks, etc.). In other systems, blobs are used as input for higher level reasoning algorithms, e.g., for facade interpretation or car detection in aerial images. To accommodate the huge variety of applications, it is clear that a blob detection algorithm must fulfill a number of general requirements, most notably:

- Reliability / noise insensitiveness: Clearly, a low-level vision algorithm should be in some way robust against under- and oversegmentation due to noise.
- Accuracy: Many applications—especially in vision metrology—need highly accurate results in sub-pixel resolution.
- Scalability: The algorithm should be scalable so that primitives of different size can be extracted.
- Speed: The algorithm should be applicable also to (near-)real-time processing.
- Few and semantically meaningful parameters for initialization: The algorithm’s parameters should be easy to understand for non-experts and the delivered results should be predictable, when changing the algorithm’s parameters.
- In addition, an important aspect of a blob detection algorithm is the capability of extracting geometric and radiometric attributes to allow for a subsequent classification of blobs.

1.2. Related Work

The huge number of approaches developed so far can roughly be grouped into following categories:

Matched filters / template matching: These approaches are quite robust against noise and they are fast in case of few transformation parameters between template and search image (e.g., only translation and rotation). Yet they become less applicable when shape deformations appear or parameters of higher order transformations must be determined. Originally, the results are delivered with pixel precision, although extensions to refine the results have been developed.

Watershed detection: Blobs are usually characterized by a quite homogeneous interior and are surrounded by an boundary edge. Such kind of structures can be extracted by so-called watershed algorithms. These algorithms assume the image to be ”grayvalue mountains” and simulate the process of rain falling onto the mountains, running down the mountain range and accumulating in valleys and basins. This process is repeated until all basins are filled and only the watersheds between different basins remain. A complementary algorithm focussing on the direct extraction of watersheds in subpixel precision is given in [1]. In order to extract both dark blobs and bright blobs, watersheds are typically extracted from the gradient amplitude image. The raining and accumulation process can be implemented very efficiently so that these algorithms can also cope with near real-time requirements. In practise, however, the bottleneck of these algorithms is the inherent noise sensitiveness which leads typically to oversegmented results. To overcome this, it would be helpful to incorporate information about shape and size of the desired primitives into the process of watershed detection. Thise is hardly feasible.

Structure tensor analysis followed by hypothesis testing of gradient directions: This method extracts blobs by detecting distinct points and analyzing their circularity (see [2] for details and [3] for improvements of the structure tensor). In a first step, potential interest points are detected by analyzing and thresholding the eigenvalues of the structure tensor in a similar way as the classical Harris-Operator. The second step consists of calculating the gradient directions in the local neighborhood of a candidate point and estimating the common intersection point of all gradients assuming circularity. Finally, a hypothesis test on the intersection's accuracy, i.e. the deviation of the gradient directions from the intersection point, decides about acceptance or rejection of the candidate point. This method combines several advantages: It can be implemented efficiently using recursive filters with noise suppression embedded; it delivers subpixel precise results for blob center points; and it needs only a few parameters to initialize. Yet, the disadvantage of this method is its limitation in extracting circular structures only. The algorithm fails as soon as the gradients do not intersect in a common point, e.g. in the case of ellipses.

Blob detection through scale-space analysis: The blob detection scheme described in [5] can be interpreted as an extension of the above method over different (Gaussian) scales. Instead of detecting features at a single scale, the complete scale space representation is analyzed for local maxima, i.e. the optimum scale. After detection of a blob center its subpixel-precise position is calculated using the same gradient intersection scheme as above. However, the optimum scale for localization is again introduced as free parameter to estimate. That particular scale, which minimizes the deviation of the gradients from their estimated intersection point, is regarded as the optimum scale for localizing the blob. From a theoretical point of view, this scheme is the most advanced one. The main drawbacks are the computational load necessary for analyzing multiple scales and the restriction to circular structures. Hence, the applicability in practise is limited to some extend.

Our work described in the sequel is mainly inspired by the classical work of Lindeberg [4, 5] and the high precision line extraction algorithm of Steger [6, 7]. Our approach attempts to keep simplicity, effectiveness, and precision of the Foerstner method [2] while extending it to more generic structures. On the other side we base the scheme on a scale space description to integrate scalability. Though, having the computational costs in mind, we refrain from searching the best scale automatically but let the user set the detection scale. Furthermore, various blob attributes are calculated to allow for a subsequent classification of blobs.

2. BLOB DETECTION

Our blob detection approach consists of two major steps. The first step comprises the extraction of potential center points of blobs in subpixel precision (Sect. 2.1). The second step outlined in Sect. 2.2 consists of reconstructing the boundary around a given point. Finally, various geometric and radiometric attributes are calculated (Sect. 2.3).

2.1. Extraction of Center Points

The extraction of blob center points is based on differential geometric considerations. A rectangle of length l, width w ($w < l$), and homogeneous area, i.e. constant contrast h, serves as basic primitive. Because of limited image resolution and the considerable amount of smoothing usually mandatory to suppress noise, an approach trying to extract and group the rectangle's lateral edges would not lead to satisfactory results. Under these conditions, a much more salient feature is the rectangle's center point that becomes a local extremum under sufficient amount of smoothing. Hence, the goal of the algorithm is first to find the center points and then to reconstruct the rectangle boundary.

Let us for a moment assume the rectangle's orientation to be known, i.e., the rectangle can be oriented along the coordinate axes x, y. Then the model function f_r for a rectangle with normalized contrast $h = 1$ is given by

$$\begin{aligned} &f_r(x,y) = 1 \quad \text{for } |x| \leq l \text{ and } |y| \leq w; \\ &f_r(x,y) = 0 \quad \text{otherwise} \end{aligned}$$

and the response r_σ after convolving f_r with a Gaussian smoothing kernel g_σ becomes

$$\begin{aligned} &r_\sigma(x,y,l,w) = \\ &(g_\sigma(x+l) - g_\sigma(x-l))(g_\sigma(y+w) - g_\sigma(y-w)) \end{aligned}$$

Because of the infinite support region of g_σ it follows that r_σ reaches the desired distinct extremum at the rectangle's center point ($x = 0, y = 0$) for all $\sigma > 0$. Another conclusion is that the 2D function r_σ results from a convolution of two separable functions: the Gaussian function g_σ and the rectangle function f_r that can be generated from two 1D bar-shaped functions (in the same way as the conventional mean filter). Consequently, r_σ is separable into two 1D functions $u_\sigma(x,l)$ and $v_\sigma(y,w)$, which are equivalent with the two profiles along both coordinate axes. What is more, also the rectangle's center point can be extracted purely from u_σ and v_σ if some requirements are met:
(1) According to [6], center points of bar-shaped profiles of width p and smoothed with g_σ can be reliably extracted by determining the curvature maximum along the profile as long as the restriction $\sigma \geq p/\sqrt{3}$ is satisfied. Thus, for a rectangle with different side lengths, either a smoothing parameter σ corresponding to $p = max(l, w)$ or two different parameters σ_l and σ_w should be chosen. Sub-pixel precision is achieved by computing the curvature maximum using a second order polynomial expanded into a Taylor series.

(2) The curvature maximum must fall into the pixel under investigation, i.e. $(x, y) \in | \pm 0.5| \times | \pm 0.5|$.
(3) The rectangle's center point is correctly reconstructed from the two profile's curvature maxima only if the maxima are not biased by different lateral contrast at the rectangle sides. To overcome this limitation, the bias must be estimated from image information and compensated. For this we adapted the algorithm described in [6].

The last question left open is how to find the rectangle's orientation before separating the analysis along and across the rectangle. The orientation can be determined by calculating the eigenvectors from the Hessian Matrix H (not to be confused with the structure tensor)

$$H(x, y) = \begin{vmatrix} r_{xx} & r_{xy} \\ r_{yx} & r_{yy} \end{vmatrix}$$

with the second partial derivatives $r_{xx};\ r_{yy};\ r_{xy} = r_{yx};$

and selecting the eigenvector (e_x, e_y) corresponding to the larger absolute eigenvalue λ_e ($\lambda_e < 0$ for bright blobs, $\lambda_e > 0$ for dark blobs). This eigenvector points along the direction o_r of the rectangle's shorter side. Since it is desirable to determine o_r in a way consistent with the extraction of the curvature maximum of the profile v_{σ_w}, the partial derivatives are estimated by convolving the image function with derivatives of g_σ using the same smoothing parameter σ_w as for analyzing v_{σ_w}.

The algorithm for center point extraction can be summarized as follows (see also Fig. 1 a) for illustration):

1. Initialization by setting σ_w, σ_l w.r.t. the expected rectangle dimensions.
2. Calculate $H(x, y)$, (e_x, e_y), and o_r using g_{σ_w}.
3. Compute curvature maximum along o_r using v_{σ_w}, analyze gradients of v_{σ_w} to determine bias and remove it.
4. Compute curvature maximum along $o_r \perp$ using u_{σ_l} and analyze gradients of u_{σ_l} to determine bias and remove it.
5. Reconstruct rectangle's center point from both profiles.

2.2. Boundary reconstruction

As a side product of the above algorithm, the blob orientation, its width and length, as well as curvature and contrast information at the center point have been determined. From these attributes the elliptical shape of the blob can be reconstructed. However, please note that this information stems from two different scales. Many applications, however, rely on attributes that refer to one single scale. Hence, we extract the blob boundary also at the smaller scale v_{σ_w}:

Since the boundary of a rectangle and of all of its smoothed versions is always convex, an appropriate model function for the boundary is given by the radius-angle function $d(\phi)$ with d being the distance from the blob center to the boundary and ϕ the corresponding angle. To reconstruct this function, we define search lines at certain angle intervals and extract the maximum grayvalue gradient along the each line. Finally, the points determined this way are linked into the boundary polygon $d(\phi)$ (see Figs. 1 b and c).

2.3. Extraction of Blob Attributes

From the features thus attained, a number of additional geometric and radiometric blob attributes are calculated, which help to further select and classify the blobs.

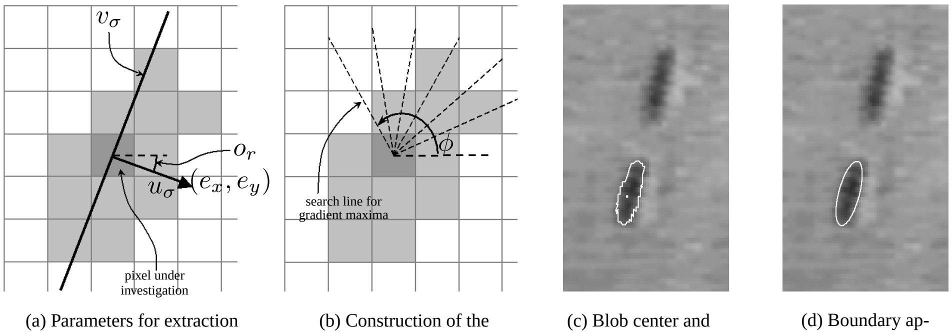
Geometric attributes are derived from the boundary polygon:

- Boundary length
- Blob area
- Geometric moments: center points, and higher order moments
- Parameters of a robustly fitted ellipse (incl. outlier removal)

Please note, that the parameters of the fitted ellipse may differ from the orientation, length and width estimated during blob detection. This is the case, for instance, if the underlying primitive is not of rectangular or elliptical shape. For specific applications these differences are valuable criteria supporting the selection of blobs.

The radiometric attributes are calculated using the (smaller) scale σ_w since the boundary is extracted at this scale:

- Mean and variance of the grayvalues of the blob interior
- Mean and variance of the gradient amplitude along the boundary, i.e. the boundary contrast statistics



(a) Parameters for extraction of center points (see text)

(b) Construction of the boundary polygon $d(\phi)$

(c) Blob center and boundary polygon

(d) Boundary approximating ellipse

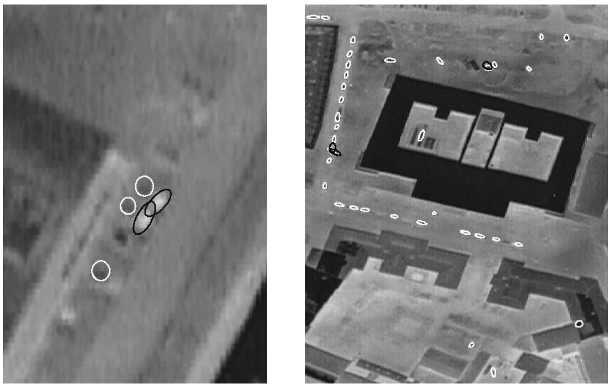
Fig. 1. Intermediate steps of blob detection.

- Asymmetry of the gradient amplitude wrt. the blob orientation, i.e., an indicator for strong contrast on one side and weak contrast on the other side of a blob.

3. RESULTS

Figures 1 c,d and 2 illustrate the application of the blob detection scheme for extracting and classifying car hypotheses using thermal aerial imagery. Passive / cold cars are imaged as dark elongated blobs and are therefore extracted by focussing on eccentrical ellipses with symmetric contrast. Contrarily, active cars are characterized as two adjacent, more or less circular blobs having different brightness. The bright spot indicates a vehicle's warm front while the dark blob corresponds to the cooler body. Hence, hypotheses for this type of cars are extracted by searching for small, near-circular ellipses with bi-polar contrast. Figure 1 c,d show the extraction of a passive car and Fig. 2 a) illustrates the extraction of two active cars. Figure 2 b) finally visualizes the hypotheses of both active and passive cars for a larger scene. As can be seen from Fig. 2 b), the majority of the hypotheses is correct and only few cars have been missed. Consequently, a high-level module could base its decision which hypotheses are to accept or reject and at which image locations additional cars have to be searched for on reliable input from the low-level module.

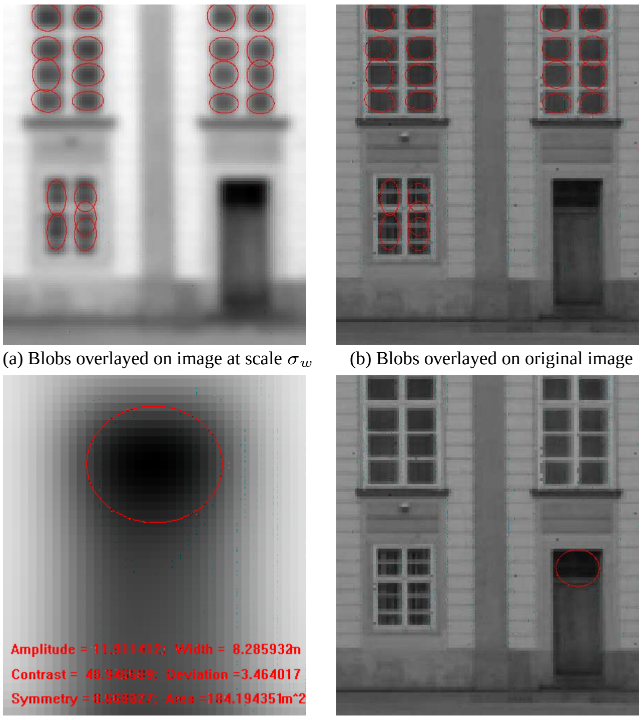
The application of our blob detection scheme as a low-level module for facade interpretation is shown in Fig. 3. In the left column, the ellipses fitted to the blob boundary are overlayed onto the smoothed image that corresponds to scale σ_w used for attribute calculation. The right column visualizes the images at original resolution. In order to extract the small windows in Figs. 3 a, b) as well as the upper part of the door in Figs. 3 c,d) two different scales have been used, and promising hypotheses have been classified using the blob attribute information.



(a) Detected near-circular blobs: bright blobs: black (and viceversa)

(b) Hypotheses for active cars (black) and passive cars (white)

Fig. 2. Detection of hypotheses for cars in airborne thermal images.


Amplitude = 11.911412; Width = 8.285932m
Contrast = 48.946689; Deviation =3.464017
Symmetry = 0.668827; Area =184.194351m^2

(a) Blobs overlayed on image at scale σ_w

(b) Blobs overlayed on original image

(c) Blob overlayed on image at scale σ_w and attribute information extracted

(d) Blob overlayed on original image

Fig. 3. Detection of blobs as primitives for facade intepretation.

4. REFERENCES

[1] C. Steger, "Subpixel-Precise Extraction of Watersheds," in *7th International Conference on Computer Vision*, 1999, pp. 884–890.

[2] W. Förstner and E. Gülch, "A Fast Operator for Detection and Precise Location of Distinct Points, Corners, and Circular Features," in *Proc. of ISPRS Intercommission Workshop, Interlaken, Switzerland*, 1987.

[3] U. Koethe, "Edge and Junction Detection with an Improved Structure Tensor," in *Pattern Recognition (DAGM 2003), Lecture Notes in Computer Science 2781*. 2003, pp. 25–32, Springer-Verlag.

[4] T. Lindeberg, *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, Dordrecht, 1994.

[5] T. Lindeberg, "Feature Detection with Automatic Scale Selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[6] C. Steger, "An Unbiased Detector of Curvilinear Structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 113–125, 1998.

[7] C. Steger, "Analytical and empirical performance evaluation of subpixel line and edge detection," in *Empirical Evaluation Methods in Computer Vision*, Kevin J. Bowyer and P. Jonathon Phillips, Eds., Los Alamitos, CA, 1998, pp. 188–210, IEEE Computer Society Press.