ELLIPTICAL DISTANCE TRANSFORMS AND THE OBJECT SPLITTING PROBLEM

H. Talbot, B. C. Appleton
CSIRO Mathematical and Information Sciences
Locked Bag 17, North Ryde, Australia 1670
[Hugues.Talbot, Ben.Appleton]@csiro.au

Abstract The classical morphological method to separate fused objects in binary images is to use the watershed transform on the complement of the distance transform of the binary image. This method assumes roughly disk-like objects and cannot separate objects when they are fused together beyond a certain point.

In this paper we revisit the issue by assuming that fused objects are unions of ellipses rather than mere disks. The problem is recast in terms of finding the constituent primary grains given a boolean model of ellipses.

To this end, we modify the well-known pseudo-Euclidean distance transform algorithm to generate arbitrary elliptical distance transforms to reduce the dimension of the problem and we present a goodness-of-fit measure that allows us to select ellipses.

The results of the methods are given on both synthetic sample boolean models and real data.

Keywords: Binary objects segmentation, blob splitting, watershed.

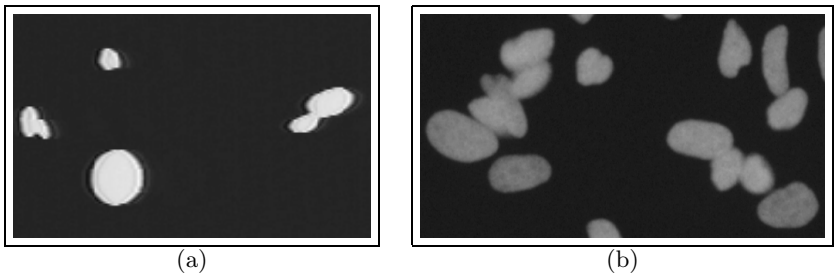
1. Introduction

In this paper we revisit the classic problem of separating fused binary objects. This problem occurs in a variety of real-world situations and is typically difficult to solve except in the simplest situations.

Figure 1 shows a couple of examples of such images. The classic morphological approach to solving such problems is to compute the watershed line on the complement of the distance transform [8]. In fact this method is so well-known that it is often incorrectly referred to as *the* watershed method [13].

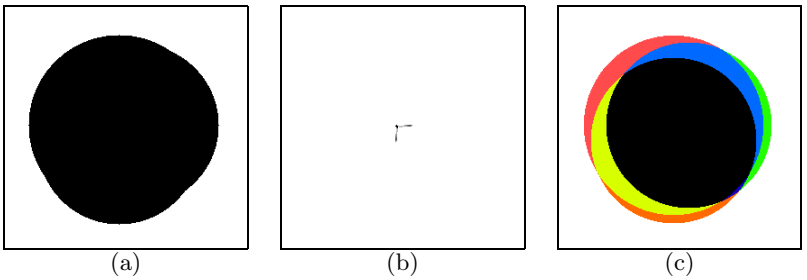
The assumption behind this method is that objects are circular and do not overlap too much. The fact that real-world objects are rarely perfectly circular implies some sensitivity to noise on the boundary of the objects. To correct for these a filtering of the distance transform is often required [3]. Other methods have also been proposed [16]. The degree of overlap that this method can cope

 ISBN 0 643 06804 X



(a) (b)

Figure 1. Two examples of fused objects: (a) glass fibres and (b) eucariot cell nuclei.



(a) (b) (c)

Figure 2. The bisector function can separate disks fused to an arbitrary degree. (a) input image; (b) bisector function – the darker pixels indicate the centres of constituent disks ; (c) derived segmentation.

with is well known: if the fused objects are disks and their centres lie on the same side of the disecting line between the two disks, then this method cannot separate them [3].

Many methods have been proposed to solve the overlap problem. A natural idea is to exploit the concavities, for example using the convex hull to find them and somehow link them together. This works well in the case of the union of a small number of simple shapes but breaks down quickly otherwise. The conditional bisector [12] and later the bisector function [17] were proposed to essentially solve the overlap problem. The latter method in particular can separate any number of disks fused to any degree, up to the digital resolution, as shown on Fig. 2. The bisector function is difficult to use when objects are not perfect disks. Its use often comes as a second pass after the standard watershed-based method, which is less than ideal [15].

A natural way to extend the method to more general shapes might be to use unions of ellipses instead. Unfortunately the dimensionality of the problem grows significantly: whereas disks only have three degrees of freedom (X and Y position plus radius), ellipses have 5 (X and Y position, orientation, major and minor axis lengths). For this reason, Hough-like methods [2] do not perform

very well, although the literature on this topic is abundant (see for example [6, 7, 9]).

In this paper we propose a method that takes as a model a union of ellipses, and uses a series of distance transforms to decrease the dimensionality of the problem. In the next section we introduce the general elliptical distance transform.

2. Elliptical distance transform

Many algorithms have been proposed to compute the Euclidean distance transform (EDT), see for example [11]. Here our goal is to compute an elliptical version of this transform, i.e: a Euclidean distance map that is stretched by a constant amount in a given direction. In contrast to the EDT which is essentially parameter-less (apart from a constant overall factor), the LDT has two parameters: an angle α and a stretch factor σ. The latter corresponds to the ratio of the major axis over the minor axis.

A continuous way of computing the elliptical distance transform (LDT) would be to contract the binary set in the direction α by the factor $1/\sigma$, to apply the ordinary EDT on the contracted set and then to stretch the resulting transform by σ in the same direction α. An obvious similar scheme can be imagined with a first stretch of the binary set in the direction $\pi/2 + \alpha$ and a contraction of the result in the same direction. However these methods do not translate well to the discrete case because of interpolation problems.

Here we propose a simple but efficient approximate scheme with the same error conditions as the Euclidean Danielsson case [5] that performs a direct computation of the LDT.

2.1 Priority queue-based EDT and LDT algorithm

The idea behind both the Danielsson method and the queue-based EDT algorithm is to propagate a vector, not a value. In two dimensions this translates to propagating two values which are the X and Y coordinates of the vector to the nearest point on the boundary.

The pseudo-code shown in Fig. 3 is valid for both the EDT and the LDT computation. For the EDT, the priority function `P` is $\texttt{P(C)} = \sqrt{\texttt{X(C)}^2 + \texttt{Y(C)}^2}$. The priority function values are written to the output image (see line 15), so the actual values of `P` are important, not just their ordering.

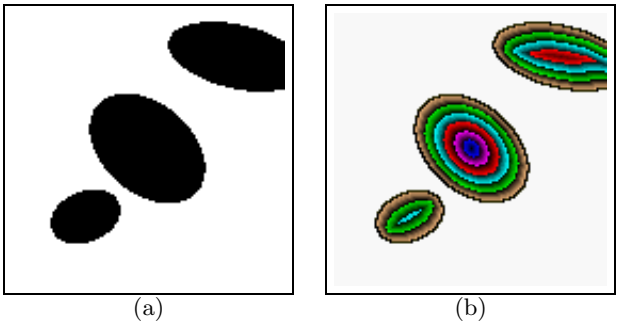
In the EDT case, this algorithm has performance equivalent to the Danielsson algorithm as each pixel within the binary set is considered to be queued on average 3 times. It can be shown that the error configurations are the same as in Danielsson's algorithm (they are artifacts of the 8-connected propagation). For our present purpose the errors are small and are not worth correcting.

One interesting property of the Fig. 3 algorithm is that `P` can be set to a wide variety of interesting functions. If set to the following function:

$$\texttt{P(C)} = \rho\sqrt{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2}, \tag{1}$$

```
- Input binary image
  Fill X image with 0
  Fill Y image with 0
  Fill output image O with 1
  Empty priority queue
~
- Scan the binary image in raster order, enqueue the points belonging to the sets
  which are 8-connected to the background with priority 1.
  For each enqueued location, set the X and Y image to the respective
  components of the direction to the nearest pixel in the background.
~
- While the priority queue is not empty; do
      - Dequeue lowest priority pixel C
      - if O(C) (the output image value at this pixel) is 1, then
          - Compute P(C), the priority of this pixel, from X(C) and Y(C),
            i.e P(C) = P(X(C), Y(C))
          - Set O(C) (the output image value at this position) to P(C).
          - For each point dC in the 8-neighborhood of this pixel; do
              - If O(C+dC) is 1 ; then
                  - if X(C+dC) or Y(C+dC) is not 0, compute P1 = P(X(C+dC),Y(C+dC))
                    else set P1 to +infinity
                  - compute P2 = P(C+dC) where '+' denotes the 2-D vector addition.
                  - If P2 < P1 then
                        - enqueue dC with priority P2
                        - set X(C+dC) to X(C) + x(dC), where x(A) is the X-component of A
                        - set Y(C+dC) to Y(C) + y(dC), where y(A) is the Y-component of A
                        - set O(C+dC) to 1
                  + end if
              + end if
          + end for
      + end if
  + end while
```

Figure 3. Pseudo-code for both the EDT and the LDT. See text for the formulation of the priority function `P`.



(a) (b)

Figure 4. (a) Input image ; (b) elliptical distance transform obtained with the algorithm from Fig. 3 with Eq. 1 as the priority function. Parameters are $\alpha = -40^\circ$ and $a/b = 1.5$ (also those of the middle ellipse).

where $\rho = \sqrt{(\mathtt{X(C)}^2 + \mathtt{Y(C)}^2)}$, $\rho\sin(\theta) = \mathtt{Y(C)}$ and $\rho\cos(\theta) = \mathtt{X(C)}$ simultaneously, then `P` defines an elliptical distance transform (LDT) of angle α and stretch factor $\sigma = a/b$. Equation 1 is a standard 2-D harmonic oscillator equation.

3. Searching for ellipses

Let us assume that a binary image I is composed of a union of ellipses (i.e: a boolean model where the primary grains are ellipses). What we are looking for is a kind of minimal set of maximal ellipses, that in some sense best represent the given image with a minimal number of "best covering" ellipses. If we were to probe the image with an a-priori set of ellipses as structuring elements, we would have to do a search on all five of the parameters of an ellipse, which would be a near-impossible task.

Fortunately, using the LDT of section 2 simplifies the search: by definition, computing the skeleton based on a given elliptical distance map (LDM) finds all the centres of maximal ellipses for the binary set that have the orientation and eccentricity of the underlying LDT [10]. To find all the possible maximal ellipses of I, it is therefore sufficient to vary the orientation and aspect ratio of the LDT.

However we still need a way to select the "best fitting" ellipses among all of those generated by the family of LDTs.

3.1 Normalization of the LDMs

In Eq. 1 different parameters yield elliptical distance maps (LDMs) which are not normalized. A simple solution is to weight the `P` function (whose values create the LDM) so that in the resulting LDM, a given value corresponds to the square root of the area of the ellipse with the same $\sigma = a/b$ and α parameters centered at that point and touching the border of the object. The resulting priority function is:

$$\texttt{P(C)} = \rho\sqrt{\frac{\pi(\cos^2(\theta-\alpha) + \sigma^2\sin^2(\theta-\alpha))}{\sigma}} \tag{2}$$

which is no more difficult to compute than Eq. 1.

3.2 Area maximization

Given this normalization, a simple goodness-of-fit measure could be to apply the LDT with different parameters and compute the point-wise maximum between them. The resulting transform, the maximum LDT (MLDT) indicates, at each point, which is the largest centered ellipse (in terms of area) that can fit in the binary image I.

$$\mathrm{MLDT}(I) = \bigvee_{\alpha,e} \mathrm{LDT}_{\alpha,e}(I)$$

By looking for regional maxima on the MLDT, one should be able to detect maximal ellipses that cover a large proportion of individual objects in the binary set. Indeed, for sets which consist of ellipses that do not overlap too much, this procedure works quite well, as shown on Fig 5(a) and (b).

Unfortunately this method falls short of our goal because for any reasonable degree of overlap, one can always fit a large ellipse in the "neck" between objects. When this happens the MLDT becomes unusable, as shown on Fig. 5(c)

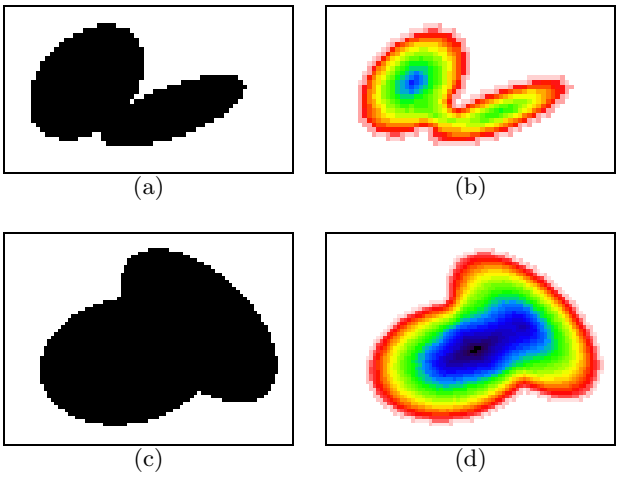


Figure 5. Examples of MLDTs in the case of ellipses that do not overlap too much ((a) and (b)), and when they do overlap significantly ((c) and (d)).

and (d). For this situation, which is the more interesting one, a better goodness-of-fit measure must be found.

3.3 Tangent ellipses

In the continuous case a maximal ellipse will be tangent to the boundary of the binary set on at least two points. If the binary set is a boolean model of ellipses, then each constituent ellipse must be tangent to the set on a non-negligible subset of the border. A good measure-of-fit for candidate ellipses might then be to test for this property. In particular poor-fit maximal ellipses will be tangent to the set only on a small number of points.

In the discrete case, one can reproduce this idea by simply counting the number of pixels on the border of a candidate ellipse that intersect the border of I. A goodness-of-fit score might then be the ratio of intersecting pixels vs. the total number of pixels in the border of the candidate ellipse. Unfortunately this number is unreliable due to discretization effects, in particular for small candidate ellipses.

A better measure in the discrete case might be to measure the *distance* of the border of a candidate ellipse to the border of the binary image I. To do this, the Euclidean distance map of I can be pre-computed. For each candidate ellipse, the distance of each of its border pixels to the border of I is recorded. The average distance is a reasonable score for single ellipses, but a more robust one is the k-percentile distance (e.g., if k=50%, the median distance), given that overlapping ellipses are to be expected and that even for good-fit ellipses a portion of their contour might be far from the border of I. Figure 6 illustrates the concept.

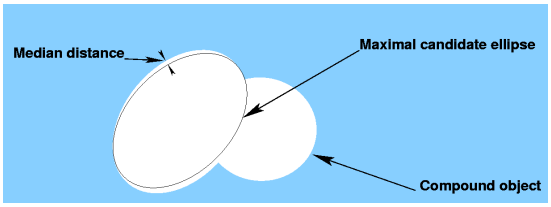


Figure 6. There is a good fit for the candidate ellipse if some proportion of the ellipse boundary is close to the object boundary.

3.4 Algorithm

Combining all these ideas, the proposed approach is the following:

1 The EDM from the contour of the binary image is computed into image C.

2 A number of LDMs of the binary image are computed, with different values of α and σ.

3 For each LDM, special points are considered as candidate centres of ellipses. These need not include all the points of the LDM skeleton. In our application we only considered regional maxima of the LDM, but ridge points on the distance map [1] or end points on the skeleton [4] are reasonable candidates.

4 For each special point, a candidate ellipse is generated with the corresponding parameters (position, axis lengths and orientation). Each pixel of the circumference of the generated ellipse is associated with its distance from the boundary of the binary image. This is provided by a simple interpolated lookup in image C. The pixel distances are sorted and a given percentile is taken as the measure m of goodness-of-fit (for example 50% yields the median distance).

For display purposes it is convenient to compute a measure which is high for a good fit. The measure that we use is $s = 100/m$. We make sure that s is never 0 even for a perfect fit. We call s the *score* of a ellipse fit. The following section shows some results obtained with this algorithm.

4. Results

4.1 Artificial data

In Fig. 7 we present the result of the algorithm on a sample boolean process instance of ellipses. Figure 7(a) is the input image, Fig. 7(b) shows the raw scores, with higher values being darker. This image clearly shows high scores near the centres of each ellipse component. Figure 7(c) is constructed by overlaying all the ellipses detected in Fig. 7(b), starting with the lower scores and finishing with the higher scores. This way high score ellipses overwrite lower

Table 1. Comparison of counting methods for image in Fig. 8(a)

Count method	Manual	Ellipse high acc.	Ellipse Low acc.	Unfilt. wshed	Filt. wshed
Number of cells	104	98	98	93	86
Total errors	0	7	10	15	18
Over-segmentation errors	0	1	2	2	0
Under-segmentation errors	0	6	8	13	18

scores. The result shows an image which is a near-perfect segmentation of Fig. 7(a). Figure 7(d) is obtained by removing the flat zones with small area from Fig. 7(c), labelling each different ellipse with a different colour and overlaying the scores greater than 100 in a different colour again. These scores were dilated once for clarity. The dilation also allows nearby high score pixels to join.

This image shows that one can obtain a perfect segmentation of the input image in this case either by selecting high score pixels or by using the overlapping ellipse method.

For comparison, Fig. 7(e) shows the Euclidean distance transform of the input image, and Fig. 7(f) shows the result of the watershed segmentation on this function. Note the under and over segmentation. Generally in the watershed method the distance transform is filtered to avoid excessive oversegmentation as in this case.

4.2 Real data

Figure 8(a) shows a subset of a thresholded image of human cell nuclei. Fig 8(b) is its segmentation by the overlapping ellipses method presented above. In spite of the low resolution, the results look reasonable. However to evaluate the results more fully we need numerical data.

We have performed a careful manual count of the cells together with 4 automated counts. The first automated count was done using the proposed method (using overlapping ellipses), with a high accuracy grid search (361 grid points[1]), the second count used the same method but with a low accuracy grid search (16 points[2]). The third counting was done using the unfiltered watershed method, and the fourth count with a watershed on a filtered distance transform (with the removal of the last plateaus, which is often the recommended method [14]). The results are presented in Table 1.

In this table it is apparent that the ellipse methods have a significantly smaller total error rate and provides a better estimate than either watershed-based method, even with the lower accuracy grid search.

In the complete image (not shown here because it is too large), the error rates are not available but the counts are shown in Table 2. Because of the

[1] $\delta\alpha$ was 5° and $\delta\sigma$ was 0.2. σ varied from 1.0 to 3.0.
[2] $\delta\alpha$ was 30°, $\delta\sigma$ was 0.5 and σ varied from 1.0 to 2.0

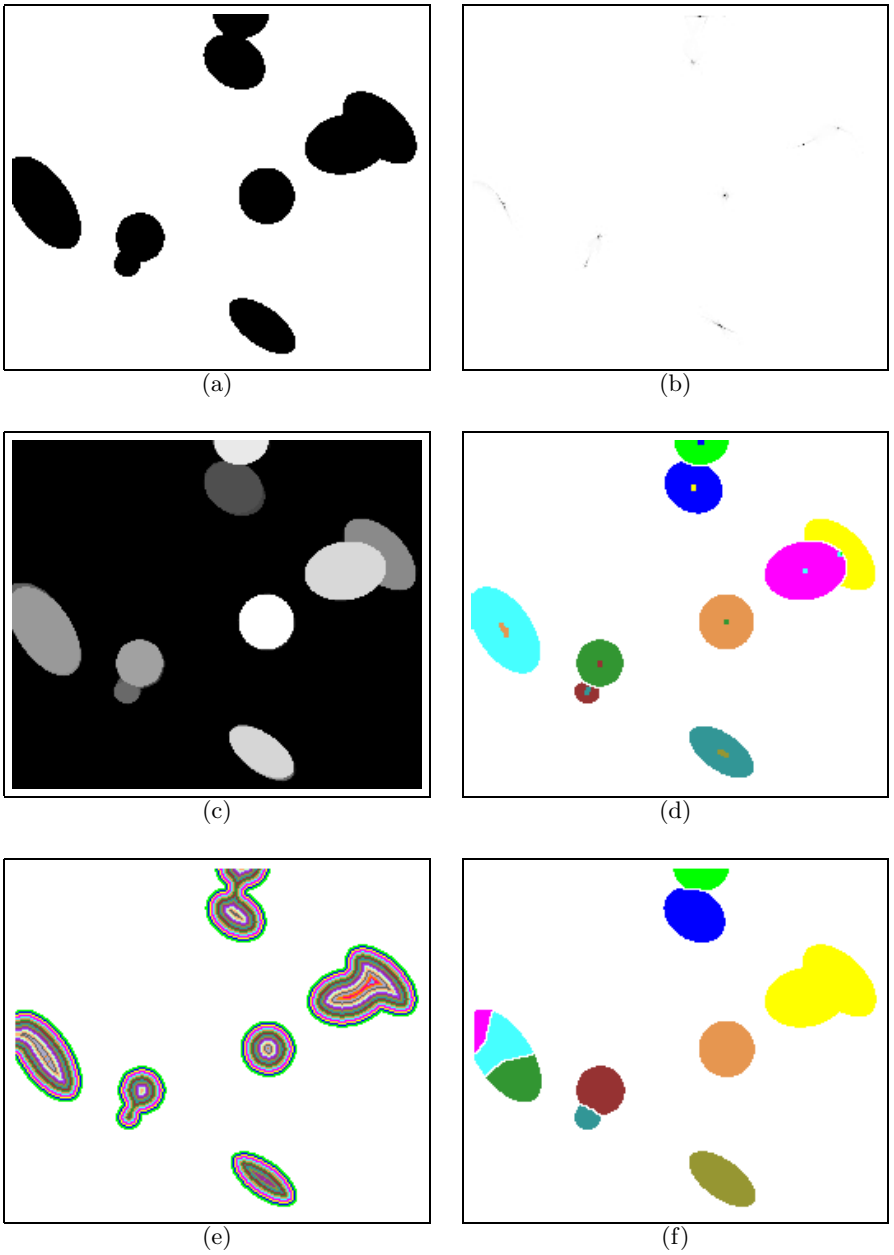


Figure 7. Result of the algorithm on synthetic data: (a) binary input image ; (b) raw scores, the higher the score the darker; (c) raw ellipses drawn from the scores, with the higher score ellipses drawn last ; (d) final result – each ellipse is drawn with a distinctive colour. Score pixels higher than 25% of the maximum are overlayed and dilated once. For comparison, (e) is the Euclidean distance transform of (a) and (f) is the watershed segmentation based on this distance transform.

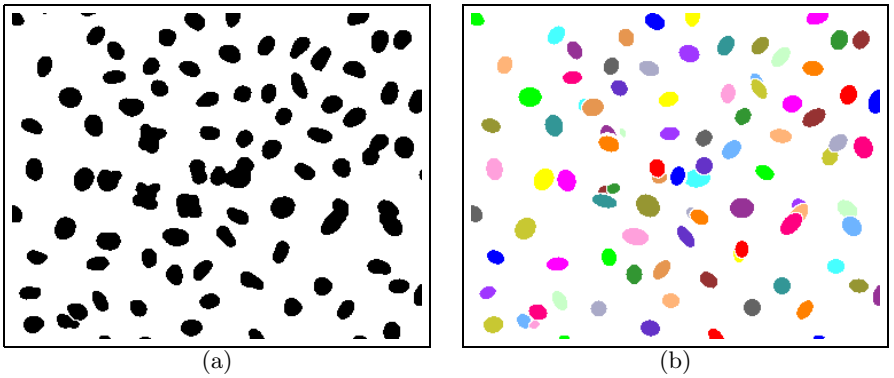


Figure 8. Result of the algorithm on real data. (a) original image ; (b) segmentation using the overlapping ellipses method.

Table 2. Comparison of counting methods for complete cell image.

Count method	Manual low	Manual high	Ellipse high accur.	Ellipse low accur.	Filtered Watershed
Number of cells	798	838	816	847	722

difficulty in counting such a large number of cells manually, both a lower and a higher bound are given. The manual counts were done by a third party.

The estimate provided by the high-accuracy ellipse method is very good, the low-accuracy ellipse method provides a slight overestimate of the cell numbers, whereas the filtered watershed method under-represents the number of cells significantly.

5. Discussion

The ellipse-based model has more degrees of freedom than the usual disk-based ones for separating fused objects. As a consequence it is more flexible and can provide better visual and numerical results as shown in the previous section.

Our method shows that the ellipse based segmentation model is tractable, however it can be quite slow. Each point of the grid search requires the computation of a distance transform. Each is quite reasonable and efficient, but a fine parameter search can require the computation of several hundreds of them. In contrast the watershed method (for example) requires the computation of only one distance transform.

The proposed method can therefore be 1 to 3 orders of magnitude slower than the benchmark method. for example the computation of the high accuracy result in Fig. 8(b) which is a 400×300 pixels image required 4 minutes on a common Pentium III/500MHz (the lower accuracy result only took 12 seconds). The computation on the complete image which is 1280×1024 pixels required

40 minutes. The lower accuracy result still took 2 minutes to complete on the whole image. Because of this fact the proposed method may be impractical for some purposes in its present state.

In future work, we plan to research ways to improve the computing time of this method and find out what degree of accuracy in the grid search is necessary to guarantee good results.

6. Conclusion

We have presented a novel way to solve the fused binary object problem using a union of ellipses model. The proposed method uses elliptical distance transforms to reduce the dimensionality of the problem, which reduces to a search for eccentricity and orientation. A goodness of fit measure for candidate ellipses was proposed.

The method is able to deliver good results on real data, showing significant improvement on the benchmark watershed-based method, but presents a dilemma in terms of accuracy vs. speed. A high accuracy result can require an impractically long computing time.

References

[1] C. Arcelli and G. S. di Baja. Ridge points in Euclidean distance maps. *Pattern Recognition Letters*, 13(4):237–243, April 1992.

[2] D. Ballard and C. Brown. *Computer Vision*. Prentice-Hall, New Jersey, 1982.

[3] S. Beucher and L. Vincent. Introduction aux outils morphologiques de segmentation. In *Traitement d'image en microscopie à balayage et en microanalyse par sonde électronique*, pages F1–F43, Paris, March 1990. ANRT, groupement microanalyse et MEB.

[4] L. Calabi and W. Hartnett. Shape recognition prairie fires, convex deficiencies and skeletons. *American Mathematical Monthly*, 75(4), 1968.

[5] P.-E. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[6] R. Duda and P. Hart. Use of the hough transform to detect lines and curves in pictures. *Comm. of the A.C.M.*, 15:11–15, 1972.

[7] J. Illingworth and J. Kittler. a survey of the hough transforms. *Computer Vision, Graphics, and Image Processing*, (44):87–116, 1988.

[8] C. Lantuéjoul. Skeletonization in quantitative metallography. In R. M. Haralick and J. C. Simon, editors, *Issues of Digital Image Procesing*. NATO, Sijthoff and Noordhoff, 1980.

[9] V. Leavers. The dynamic generalized hough transform. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 3(56):381–398, 1992.

[10] G. Matheron. Example of topological properties of skeletons. In J. Serra, editor, *Image Analysis and Mathematical Morphology*, volume 2, Theoretical Advances, pages 217–238. Academic Press, London, 1988.

[11] A. Mehnert and P. Jackway. On computing the exact euclidean distance transform on rectangular and hexagonal grids. *Journal of Mathematical Imaging and Vision*, 11(3):223–230, 1999.

[12] F. Meyer. *Cytologie quantitative et morphologie mathématique*. PhD thesis, Ecole des Mines de Paris, 1979.

[13] J. C. Russ. *The Image Processing Handbook*. CRC Press, 2nd edition, 1995. ISBN 0-8493-2516-1.

[14] P. Soille. *Morphological Image Analysis, principles and applications*. Springer, 1999.

[15] H. Talbot. *Analyse morphologique de fibres minérales d'isolation*. PhD thesis, Ecole des Mines de Paris, October 1993.

[16] H. Talbot and I. Terol Villalobos. Binary image segmentation using weighted skeletons. In *Image algebra and morphological image processing III*, volume 1769, pages 147–155, San Diego, CA, SPIE, July 1992.

[17] H. Talbot and L. Vincent. Euclidean skeleton and conditional bisectors. In *Visual Communications and Image Processing'92*, volume 1818, pages 862–873, Boston, SPIE, November 1992.