# Discrete Morphology with Line Structuring Elements

C.L. Luengo Hendriks, L.J. van Vliet

Pattern Recognition Group
Department of Applied Physics
Delft University of Technology
Lorentzweg 1, 2628 CJ Delft, The Netherlands
cris@ph.tn.tudelft.nl

**Keywords:** interpolation, skewing, Bresenham line, periodic line, band-limited line, image processing

## Abstract

*Discrete morphological operations with line segments are notoriously hard to implement. In this paper we study different possible implementations of the line structuring element, compare them, and examine their rotation and translation invariance in the continuous-domain sense. That is, we are interested in obtaining a morphological operator that is invariant to rotations and translations of the image before sampling.*

## 1 Introduction

Morphological operations use a structuring element (SE), which plays the role of a neighborhood or convolution kernel in other image-processing operations. Often, these SEs are composed of line segments. For example, the square, hexagon and octagon, which are increasingly accurate approximations of the disk, can be decomposed into two, three and four line segments respectively [11, 12]. Thus, it is possible to create an arbitrarily accurate approximation of a disk by increasing the number of line segments used. The advantage of using line segments instead of $N$-dimensional structuring elements is a reduction in the computational complexity. Furthermore, it is possible to implement a dilation or erosion by a line segment under an arbitrary angle with only 3 comparisons per pixel, irrespective of the length of the line segment, using a recursive algorithm [17, 13].

Adams [1] showed how to create an optimal discrete disk using dilations with line segments. These disks are only approximations of the sampled Euclidean disk. The optimality is a trade-off between accuracy and efficiency. For multi-scale closings with these disks, however, absorption does not hold. Jones and Soille [5] improved on this by using periodic lines, so that the absorption property is satisfied. Nonetheless, these SEs sacrifice accuracy to gain implementation efficiency. The actual implementation of the line structuring element is not very important, since the result is an approximation anyway.

Our reason to study the implementation of the line SE is to improve on the result of morphological operations used to detect and measure linear features in images. Examples are roads in airborne images [3, 6], grid patterns on stamped metal sheets [16], and structure orientation estimation [14, 15]. We also use line SEs in RIA Morphology [9, 10].

This paper is organized as follows. We start with an introduction to Bresenham lines, the basic discrete lines. The most simple implementation of the dilation uses a Bresenham line as SE. For efficiency purposes, one might compute regional maxima over a Bresenham line across the image (using the recursive algorithm mentioned above). The drawback is that this operation is not even translation-invariant in the discrete sense (i.e. invariant over integer pixel shifts). Jones and Soille [5] introduced periodic lines, which are studied in Section 3. Using periodic lines, it is possible to construct recursive dilations that are translation-invariant in the discrete sense. After that we introduce operations obtained by interpolating the image to obtain regional maxima over line segments (Sections 4 and 5), and a grey-value SE that implements an approximately band-limited line segment (Section 6).

All of these approaches are compared in Section 7. We then test the two best methods for rotation-invariance and translation-invariance. Note that when we talk about translation-invariance, we actually mean invariance to (sub-pixel) shifts of the sampling grid; that is, translation-invariance in the continuous-domain sense (unless explicitly stated otherwise).

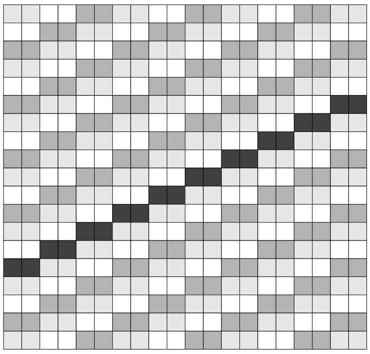

Figure 1: A Bresenham line across the image can be tiled so that each pixel in the image belongs to a single line. Along these lines it is possible to compute the dilation (or any other operation).

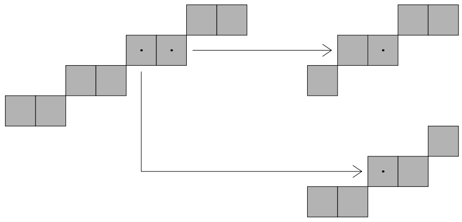

Figure 2: The problem with a Bresenham line is that each pixel along the line is embedded in a differently shaped neighborhood. Each of these neighborhoods are equally good approximations of the continuous line segment.

## 2 Basic Discrete Lines: Bresenham Lines

Bresenham [2] published an algorithm to draw a line segment of any orientation on a plotter that could only draw horizontal, vertical and diagonal lines. The algorithm combines small portions of these lines to form a line segment of any orientation. In image processing, Bresenham lines are formed by steps in the eight cardinal directions of the grid.

To efficiently implement a dilation with a line segment of any orientation, the recursive algorithm proposed by van Herk [17] can be applied to a Bresenham line crossing the image [13], as in Figure 1 (lines can be tiled to cover the whole image). This results in, at each point, the maximum over some pixels (along the line) at each side of that point. The problem is that, for neighboring pixels, the configuration of this neighborhood is different. Take as an example a line that goes up one pixel for each two that it goes right. Such a line is drawn by making one step right and one diagonally up (see Figure 2). There are two ways of starting this line (one of the two steps must be taken first), and each pixel along this line is embedded in one of two different neighborhoods. The dilation along this line will therefore be computed with two different SEs (both versions are an equally good approximation of the continuous line segment), alternated from pixel to pixel. When the image is translated horizontally by one pixel, and translated back after the operation, a different result is produced than when the operation is computed without translation.

Only the horizontal, vertical and diagonal lines can be used to compute dilations that are translation-invariant (in the discrete sense). For all other orientations, the shape of the SE changes from point to point in the image. This should not pose a significant problem for band-limited images. Both shapes used (in the example above) are equally poor approximations of the continuous line segment. The error introduced because of this outweighs the problems caused by the shape-change due to the recursive implementation.

We implemented this method by skewing the image in such a way that all pixels belonging to the Bresenham line are aligned on a row (or column, depending on the orientation of the line) of the image (that is, each column is shifted by an integer number of pixels). On this skewed image the operations can be applied along the rows, and the result must be skewed back.

Soille and Talbot [15] proposed to use the intersection of the closings (or the union of the openings) along all possible Bresenham lines of the desired angle. In the example above, where there are two possible Bresenham lines representing the same continuous line, this would be the minimum of two closings. Using this method, discrete translation-invariance is assured, but there are other problems. First of all, depending on the number of Bresenham lines that exist for the given angle, this can be more expensive than the non-recursive implementation using a Bresenham line segment as SE. Secondly, a closing in this manner uses a non-rigid line segment: because the intersection of closings is used, if any of the possible segments fits a feature, this feature is kept. This means that the line segment is allowed to “wiggle” in between the image features. Thirdly, the operation is still not translation-invariant in the continuous-domain sense. This method is not applicable for dilations or erosions (since the intersection of dilations is not a dilation and the union of dilations leads to a dilation with a thick line segment).

Another problem with the discrete line segment (whether implemented with a recursive algorithm or not) is that the length, defined by an integer number of pixels, depends on the orientation of the segment. For each orientation, there is a different set of lengths that are possible to construct.

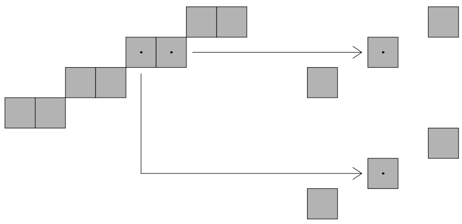

Figure 3: The problem of the Bresenham line can be solved by using only a limited number of pixels on the line. This way, each neighborhood is the same, although it is no longer connected. This is a periodic line.

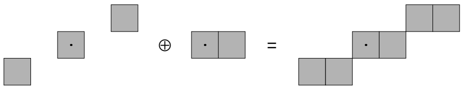


Figure 4: By dilating a periodic line segment with a small SE, it is possible to join up the SE. This limits the available lengths of the SE to multiples of the period.

## 3 Periodic Lines

Periodic lines were introduced by Jones and Soille [5] as a remedy to the (discrete) translation-invariance of the morphological operations along Bresenham lines. A periodic line is composed of only those points of the continuous line that fall exactly on a grid point, see Figure 3. These lines are thus formed of disconnected pixels, except for lines of one of the three cardinal orientations. When considering only these points, it is possible to use a recursive implementation along the periodic lines that is translation-invariant in the discrete sense. However, because of the sparseness of the points along such a line, they are not useful except in constructing more complex structuring elements. For example, by dilating a periodic line segment with a small connected segment, one creates a connected line segment, as in Figure 4. Thus, to implement a (discrete) translation-invariant dilation, one would compute a dilation with a periodic line segment, and on the result apply another dilation with a small connected line segment (which does not need to be implemented recursively because it is so small).

The drawbacks of this method are the small number of orientations for which it is useful (there are only few orientations that produce a short periodicity; for longer periodicities the line segment needed to connect the periodic line is longer as well), and the limited number of lengths that can be created (the length is a multiple of the periodicity, which depends on the orientation).

Because the result of this implementation is the same as that obtained by a direct (non-recursive) implementation using a Bresenham line segment as SE, we do not consider it separately in the comparison of Section 7.

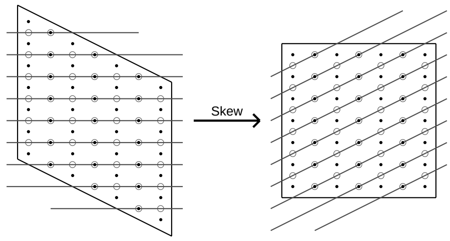


Figure 5: After skewing the image, horizontal lines correspond to lines under a certain angle with respect to the image data. Some of the original image samples fall exactly on these lines (·), but most samples used (○) lie in between original grid points. The value at these points is obtained by interpolation.

## 4 Interpolated Lines by Skewing of the Image

We mentioned above that operations along a Bresenham line can be implemented by skewing the image, applying the operation along a column (or row), and skewing the image back. In this section we consider image skews with interpolation (that is, the rows or columns of the image are not shifted by an integer number of pixels, but by a real value). See Figure 5.

The interpolation method used is an important factor in the correctness of the output. The better the method is, the smaller the error will be. We used cubic convolution [7] to implement the skews. This method is a good compromise between accuracy, computational cost and window size.[1]

The lines obtained in this way are interpolated, but have the same number of samples as the Bresenham line of the same parameters. It is expected that these result in a somewhat better translation-invariance. The mayor drawback is that the result needs to be skewed back. As stated before, morphological operations do not produce band-limited images, and therefore the results are not sampled properly. Interpolating the result of a morphological operation is questionable at best.

The reason we need to interpolate in the resulting image is that the result of the morphological operation is computed at the points along the continuous line laid across the image, and not at the grid points of the output image. There are few columns (as many as there are points in the periodic line representation for the selected orientation) with zero or integer shift.

[1] Remember that the image is not infinite in size, and therefore it is not possible to use the ideal interpolator. The window size is important because it determines the portion of the image affected by the border.

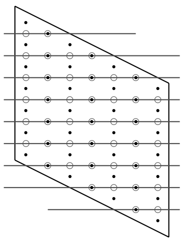

Figure 6: At the expense of some extra computations, it is possible to directly compute each of the output columns, so that the inverse skew is not required. Not having to interpolate in the result of a morphological operation is the safest way.

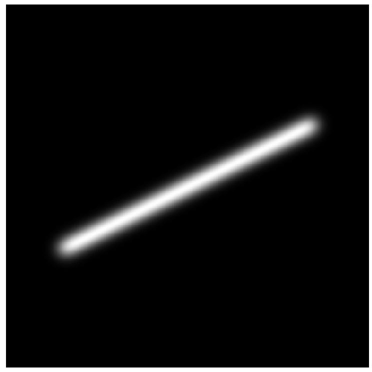

Figure 7: An approximately band-limited line segment constructed with Equation (1).

For these columns, no interpolation of the output is required, and the result is at its best.

To improve the result on the other columns, it might be interesting to sample the lines more densely before applying the morphological operation. This makes the inverse skew more accurate because the aliasing introduced by the operation will be less severe. In [8] we also used interpolation to increase the accuracy of morphological operations.

## 5 True Interpolated Lines

The interpolated lines presented above are at their best on only a few columns (or rows) of the image. It is, of course, also possible to accomplish the same accuracy for all output pixels. In this case, for each output pixel, samples along a line that goes exactly through it are computed by interpolation, as in Figure 6. On these computed samples the operation is performed.

To compute these lines somewhat efficiently, we resort again to the skew. By changing the offset of the image for the skew, it is possible to select which group of columns gets an integer shift. After performing the skew many times, different images are obtained. Each of the columns of the input image is represented with integer shift in one of these skewed images. After applying the operation on the rows of these images, the columns with integer shifts can be extracted and used to construct the output image. No interpolation needs to be performed in the output images of the operation. The number of skews that need to be computed is equal to the periodicity of the periodic line across the image.

Again, as for all discrete line segments mentioned up to now, the number of samples used in the computation of the morphological operation depends on both the length of the segment and the orientation. Line segments along the grid are the densest, and diagonal segments have the least number of samples. Thus, for some orientations it is more probable to miss a local maximum (i.e. the maximum falls in between samples) than for others. This makes the continuous-domain translation-invariance better for horizontal and vertical lines than for diagonal lines, and also has repercussions for the rotation-invariance. Ideally, one would like to sample each of these lines equally densely. To do so, it would be necessary to add columns to the image when skewing. As mentioned above, this also enables the creation of sub-pixel segment lengths, in a similar fashion to the interpolation used in [8] to increase the accuracy of the isotropic closing.

Alternatively, rotating the image instead of skewing it also alleviates this problem. However, when rotating, only a limited set of samples falls exactly on output samples, and in the worst case this happens only for the sample in the origin of the rotation. This means that a larger number of operations is required to compute the result of the operation at all output pixels.

We have not corrected for the number of samples along the line segment in the comparison below.

## 6 Band-Limited Lines

A last option when implementing morphology with discrete line segments is to use grey-value SEs, which allows to construct band-limited lines. Such a segment is rotation and translation invariant, and does not have a limited set of available lengths. The drawback is that the line is thicker, but this should not be a problem for band-limited images, since it should contain only thick lines as well.

A Gaussian function, as well as its integral, are band-limited in good approximation, and can be sampled at a rate of $\sigma$ with a very small error [18]. An approximately band-limited line segment can be generated using the error-function along the length of the segment, and using the Gaussian function in the other dimensions.

Let us define a two-dimensional image $L_{(\ell,\sigma)}$, to be used as a structuring element, by

$$L_{(\ell,\sigma)}(x,y) = A \cdot \exp\left(\frac{-y^2}{2\sigma^2}\right) \cdot \frac{1}{2}\left\{1 - \operatorname{erf}\left(\frac{\ell - 2|x|}{2\sigma}\right)\right\} \quad , \qquad (1)$$

where $\ell$ is the length of the line segment, $x$ is the coordinate axis in the direction of the segment, and $y$ is the coordinate axis perpendicular to it. Again, setting $\sigma$ to 1 is enough to obtain a correctly sampled SE. Figure 7 shows an example of such a band-limited line segment. Of course, generating line segments in higher-dimensional images is trivial: $y$ needs to be substituted by a vector. Note that the grey-value of the segment is 0, and the background has a value of $-A$. $A$ is the scaling of the SE image, and should depend on the grey-value range in the image to be processed.

It is not directly clear, however, how to scale this image $L_{(\ell,\sigma)}$. It is obvious that the height $A$ of the line segment must be larger than the range of grey-values in the image. If it is not, the edge of the image used as SE will influence the morphological operation, which is not desirable. But this height will also influence the shape of the segment. Even though the line segment is approximately band-limited for any $A$, its slopes are not invariant to this grey-value scaling. Since morphological operations can be written as an interaction between slopes [4], it follows that this scaling definitely has an influence on the result of the operation. By relating the value of $A$ to the range of grey-values in the image, the operation is invariant to grey-value scaling of the image, but not invariant to e.g. impulse noise (which increases the grey-value range), or grey-value scaling of individual objects in the image. We obtained the best results by setting $A$ just a little larger than the image grey-value range. We used the factor 1.0853, which sets the region of the SE that can interact with the image to $|x| \leq \ell/2 + \sigma$.

## 7 Comparison of Discrete Line Implementations

We have implemented the following versions of the dilation and the opening with a line segment SE:

- Method 1: with a Bresenham line segment as SE.
- Method 2: along Bresenham lines across the image (Section 2).
- Method 3: with periodic lines (Section 3).
- Method 4: along interpolated lines across the image (Section 4).
- Method 5: with true interpolated lines (Section 5).
- Method 6: with an approximately band-limited line segment as SE (Section 6).

Figure 8 shows the dilation with each of these methods applied to an image with a discrete delta pulse and a Gaussian blob. This figure gives an idea about the shape used in the operation. Methods 1 and 2 produce discrete line segments, whereas methods 4 and 5 produce line segments with grey-values that do not exist in the input image. As expected, using a periodic line produces a disjoint collection of points. Finally, method 6 produces the thickest, but also the smoothest, line segment.

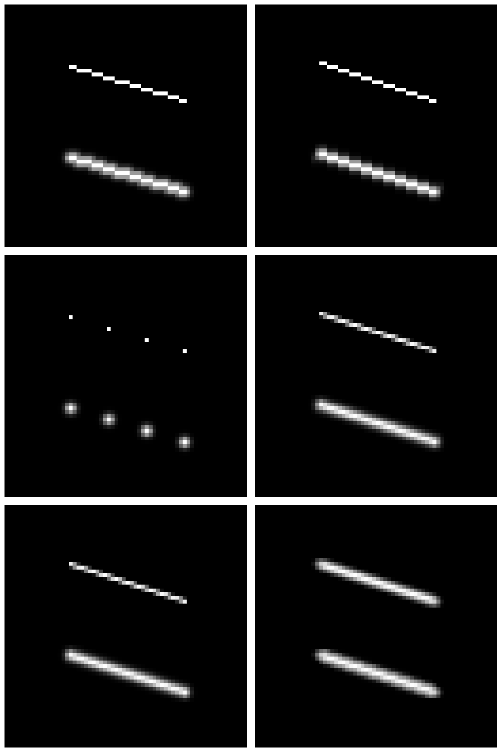

Figure 8: Sample dilation with different implementations of the line segment SE. This gives an idea about the shape of the SE used. Top row, from left to right: methods 1 and 2. Middle row: methods 3 and 4. Bottom row: methods 5 and 6.

To compare these different methods, an image was generated that contains many line segments of fixed length and orientation, but varying sub-pixel position. They were drawn using (1). Openings were applied to this image, changing both the length and orientation of the SE, and using each of the implemented methods. The result of each operation is integrated (taking the sum of the pixel values), and plotted in a graph (see Figure 9). It is expected that this results in a value of 1 for the openings in which the angle of the SE matches that of the segments in the input image, and the length $\ell$ is smaller or equal to the length of these segments. The result should be 0 for any other parameter of the SE. The more the result approximates this ideal situation, the better the specificity of the operator is.

There are a couple of things that readily come to mind when comparing these graphs:

- All methods produce a similar result, with the ex-

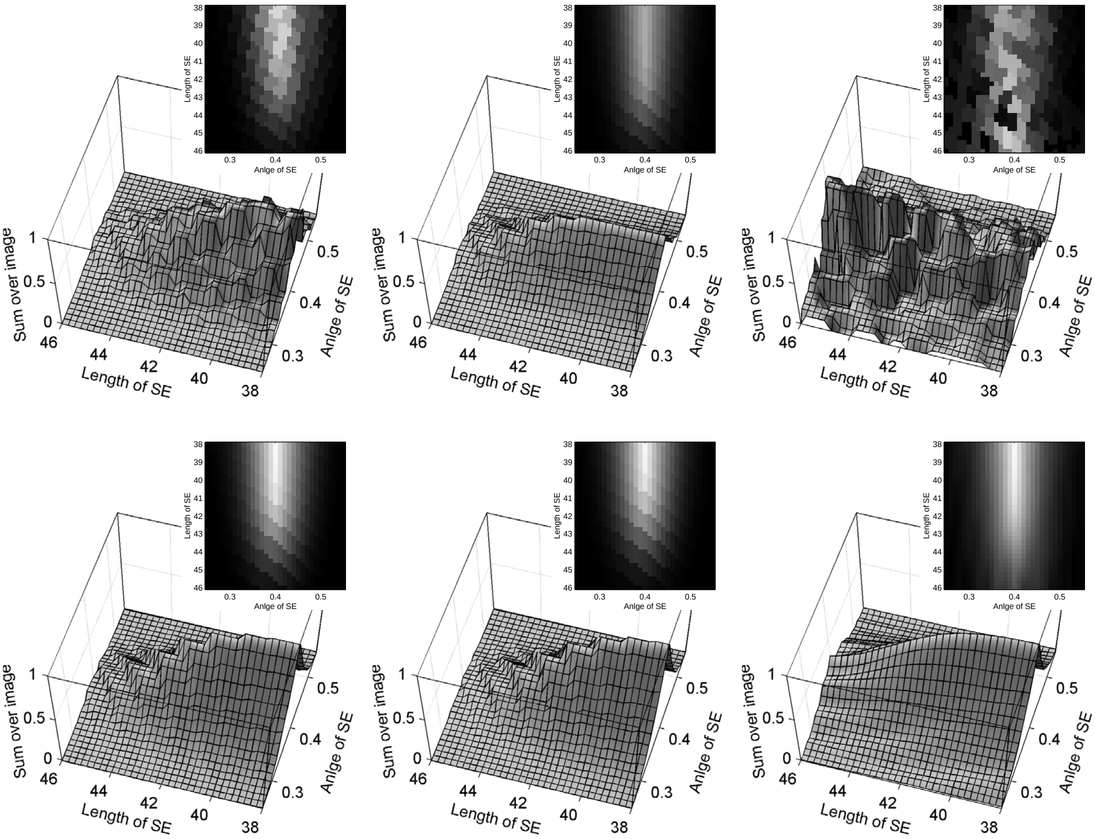


Figure 9: Comparison of different implementations of the opening with a line segment SE. See text for details. The input image has line segments of length 42 pixels, under an angle of 0.4 rad. Top row, from left to right: methods 1, 2 and 3. Bottom row: methods 4, 5 and 6.

ception of the periodic lines (method 3). This is due to the fact that the periodic line segment is disjoint, and therefore can "fit" inside two image features at once. For most of the orientations, the periodic line segment consists of only 2 points.

- The two discrete, non-interpolated implementations (methods 1 and 2), never reach values approximating 1. The interpolated and grey-value methods (methods 4, 5 and 6) reach higher values, closer to the ideal value of 1.
- The three methods that work along lines across the image (methods 2, 4 and 5) show a stair-like dependency on the length. This is because of the discretized lengths of these segments. Note that the actual length of the SE depends on the angle. This dependency is less obvious in method 1 because there both the length and the angle are discrete. When slightly changing the length, the angle changes slightly as well. This results in a less ordered pattern, which masks this dependency. In the other methods, first an angle is set, then the length is rounded to the nearest integer.
- There are very few differences between the two interpolated methods (methods 4 and 5).
- The result of the grey-value method (method 6) is very smooth, but shows some "ringing". This can be explained by the sampling of the SE and the image: morphological filtering uses the maximum or minimum value in a neighborhood, and it depends on whether a sample exactly hits such a maximum or minimum that it can be found or not. By modifying slightly the angle of the line, a different set of samples will sit close to maxima or minima (i.e. the ridge of the line).

Taking these observations into account, it can be said that the interpolated methods and the grey-value method produce results more consistent with the expectations than the discrete methods. Also, it does not appear to be necessary to use method 5, since it produces a result very similar to method 4. Method 4 is, of course, much simpler and computationally cheaper.

To further examine the interpolated method (method 4), the experiment was repeated changing the length and orientation of the line segments in the image. The results are shown in Figures 10 and 11. When changing the length, it becomes obvious that it is not possible to distinguish between lengths of 42 and 42.5 pixels. The reason is that the SE length is rounded to an integer value after skewing. This means that for each orientation, there is a different set of possible lengths, as can be seen in the graphs obtained by

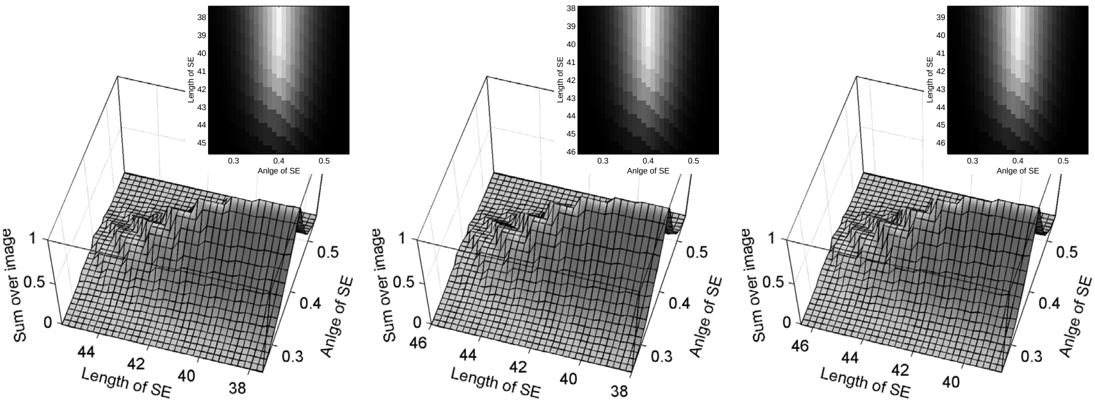


Figure 10: Evaluation of method 4 (opening along an interpolated line). These graphs were obtained by changing the length of the line segments in the input image. From left to right: 41.5, 42.0 and 42.5 pixels long.

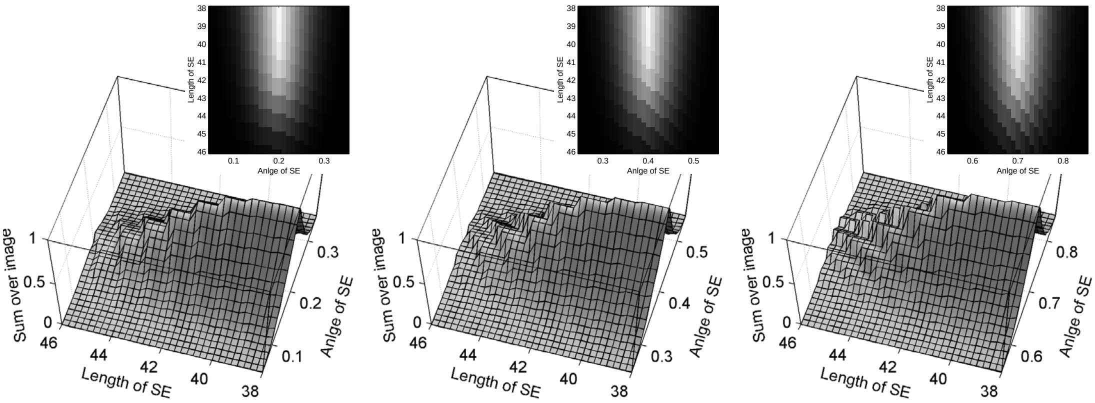


Figure 11: Evaluation of method 4 (opening along an interpolated line). These graphs were obtained by changing the angle of the line segments in the input image. From left to right: 0.2, 0.4 and 0.7 rad.

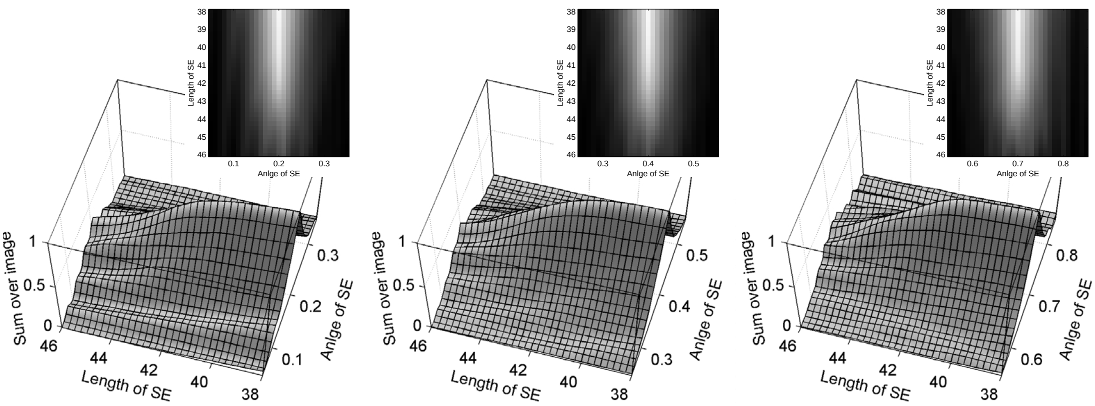


Figure 12: Evaluation of method 6 (opening using a grey-value line segment). These graphs were obtained by changing the angle of the line segments in the input image. From left to right: 0.2, 0.4 and 0.7 rad.

changing the orientation of the line segments in the image (Figure 11). The angle of the steps in these graphs change with the selected angles.

This does not happen with the grey-value morphology (see Figure 12). The only thing that changes in these graphs is the strength of the ringing effect. The smaller the angle, the larger this effect, because there will be larger sections of the ridge far away from any sample.

## 8 Conclusion

In this paper we reviewed some common methods to implement morphological operations with line structuring elements on digitized images. Besides these methods we also proposed some methods that use interpolation, under the assumption that this will increase the similarity of the operator to its continuous-domain counterpart. We also investigate the use of an approximately band-limited line segment as a grey-value structuring element.

After comparing these methods, we conclude that using interpolation indeed improves the performance of the operator. However, we also note that the available lengths are still discrete and depend on the orientation of the line segment. Using a grey-value structuring element produces satisfactory results as well, and removes the discreteness of the length and angle of the structuring element.

## References


[1] R. Adams. Radial decomposition of discs and spheres. *CVGIP: Graphical Models and Image Processing*, 55(5):325–332, 1993.

[2] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBS Systems Journal*, 4(1):25–30, 1965.

[3] J. Chanussot and P. Lambert. An application of mathematical morphology to road network extractions on SAR images. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 399–406, Dordrecht, 1998. Kluwer Academic Publishers.

[4] L. Dorst and R. van den Boomgaard. Morphological Signal Processing and the Slope Transform. *Signal Processing*, 38:79–98, 1994.

[5] R. Jones and P. Soille. Periodic lines: Definition, cascades, and application to granulometries. *Pattern Recognition Letters*, 17(10):1057–1063, 1996.

[6] A. Katartzis, V. Pizurica, and H. Sahli. Applications of mathematical morphology and Markov random field theory to the automatic extraction of linear features in airborne images. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 405–414, Boston, 2000. Kluwer Academic Publishers.

[7] R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

[8] C. L. Luengo Hendriks and L. J. van Vliet. Morphological scale-space to differentiate microstructures of food products. In *Proceedings of the 6th Annual Conference of ASCI*, pages 289–293, Lommel, Belgium, 2000.

[9] C. L. Luengo Hendriks and L. J. van Vliet. Segmentation-free estimation of length distributions using sieves and RIA morphology. In *Proceedings Third International Conference, Scale-Space 2001*, LNCS 2106, pages 389–406. Springer, Berlin, 2001.

[10] C. L. Luengo Hendriks and L. J. van Vliet. A rotation-invariant morphology for shape analysis of anisotropic objects and structures. In *Proceedings 4th International Workshop on Visual Form, IWVF4*, LNCS 2059, pages 378–387. Springer, Berlin, 2001.

[11] G. Matheron. *Random Sets and Integral Geometry*. Wiley, New York, 1975.

[12] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[13] P. Soille, E. J. Breen, and R. Jones. Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):562–567, 1996.

[14] P. Soille and H. Talbot. Image structure orientation using mathematical morphology. In *Proceedings of the 14th International Conference on Pattern Recognition*, volume 1, pages 1467–1469. Los Alamitos, 1998.

[15] P. Soille and H. Talbot. Directional morphological filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1313–1329, 2001.

[16] A. Tuzikov, P. Soille, D. Jeulin, H. Bruneel, and M. Vermeulen. Extraction of grid patterns on stamped metal sheets using mathematical morphology. In *Proceedings of the 11th International Conference on Pattern Recognition*, volume 1, pages 425–428, The Hague, 1992.

[17] M. van Herk. A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognition Letters*, 13:517–521, 1992.

[18] L. J. van Vliet. *Grey-Scale Measurements in Multi-Dimensional Digitized Images*. PhD thesis, Pattern Recognition Group, Delft University of Technology, Delft, 1993.