Discrete Morphology with Line Structuring Elements

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

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

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. 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 [6]. 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 [10, 7].

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 [2], grid patterns on stamped metal sheets [9], and structure orientation estimation [8]. We also use line SEs in RIA Morphology [5].

2 Implementations of the Line SE

In this section we examine the various possible implementations of the line SE. All these approaches are compared in Section 3. 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 (unless explicitly stated otherwise).

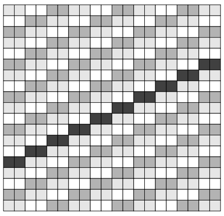

Fig. 1. A Bresenham line across the image can be tiled so that each pixel in the image belongs to a single line. Along these lines it is possible to perform an operation.

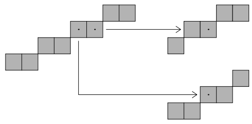

Fig. 2. The problem with a Bresenham line is that each pixel along the line is embedded in a differently shaped neighborhood.

2.1 Basic Discrete Lines: Bresenham Lines

Bresenham lines [1] are formed by steps in the eight cardinal directions of the grid, and are the basic discrete lines. The most simple implementation of a morphological operation with a line SE uses a Bresenham line segment as the SE.

To efficiently implement a dilation with a line segment, the recursive algorithm proposed by van Herk [10] can be applied to a Bresenham line crossing the image [7], 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. The problem is that, for neighboring pixels, the configuration of the SE 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, 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. This should not pose a significant problem for band-limited images. All shapes used 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.

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.

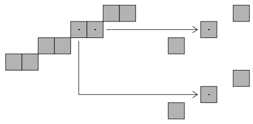

Fig. 3. A periodic line is defined by only those pixels that fall exactly on the continuous line.

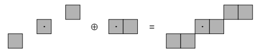

Fig. 4. By dilating a periodic line segment with a small SE, it is possible to join up the SE.

2.2 Periodic Lines

Periodic lines were introduced by Jones and Soille [3] 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.

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 3.

2.3 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. We used cubic convolution [4] to implement the skews. This method is a good compromise between accuracy, computational cost and window size.

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. Morphological operations do not

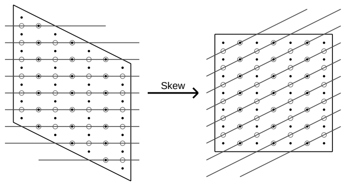

Fig. 5. After skewing the image, horizontal lines correspond to lines under a certain angle with respect to the image data. Some of the original image samples fall exactly on these lines (·), but most samples used (∘) lie in between original grid points.

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

2.4 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. On these computed samples the operation is performed. This can be implemented with one skew for each column (or row) of 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 by adding columns to the image when skewing. This also enables the creation of sub-pixel segment lengths. We have not corrected for the number of samples along the line segment in the comparison below.

2.5 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 σ with a very small error [11]. 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

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

where ℓ 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.[1] Again, setting σ to 1 is enough to obtain a correctly sampled SE.

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. 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 $|x| \leq \ell/2 + \sigma$.

3 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.
- Method 3: with periodic lines.
- Method 4: along interpolated lines across the image.
- Method 5: with true interpolated lines.
- Method 6: with an approximately band-limited line segment as SE.

Figure 6 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 of the shape used in the operation.

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

[1] Of course, generating line segments in higher-dimensional images is trivial: y needs to be substituted by a vector.

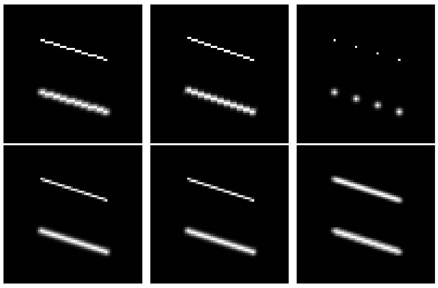

Fig. 6. Sample dilation with different implementations of the line segment structuring element. This gives an of about the shape of the structuring element used. Top row, from left to right: methods 1, 2 and 3. Bottom row: methods 4, 5 and 6.

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 7). 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 ℓ is smaller or equal to the length of these segments. The result should be 0 for any other parameter of the SE, but in practice will decrease slowly when moving away from the correct parameters. This is due to the smoothness of the line segments in the input image.

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

– All methods produce a similar result, with the exception 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. For method 1, there is also a stail-like dependency on the angle, because it is discrete as wll.
– 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 effect is angle-dependent (graphs not shown).

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.

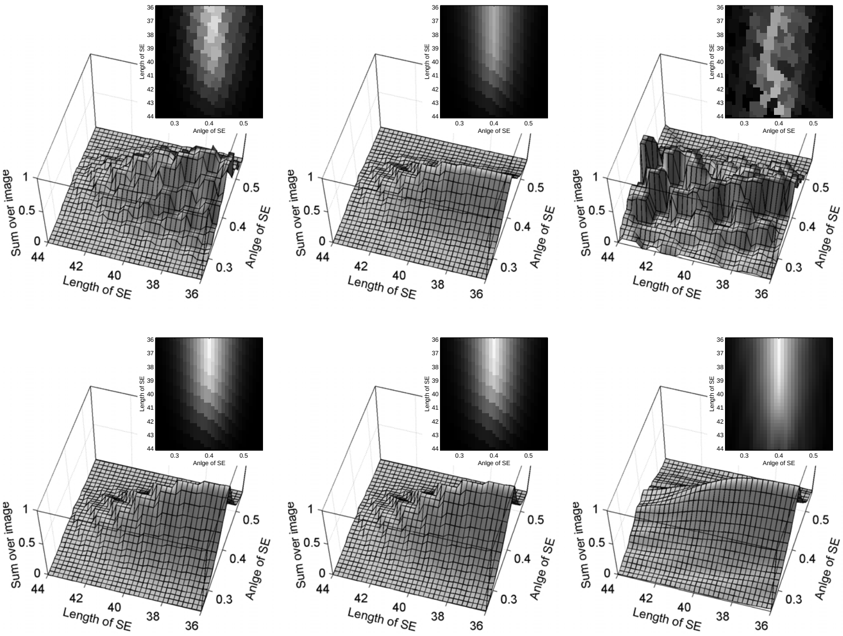

Fig. 7. Comparison of different implementations of the opening with a line segment structuring element. See text for details. The input image has line segments of length 40 pixels, under an angle of 0.4 rad. Top row, from left to right: methods 1, 2 and 3. Bottom row: methods 4, 5 and 6.

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 Figure 8. As seen in these graphs, increasing the length increases the angluar selectivity of the filter. Also, for each orientation there is a different set of possible lengths. This does not happen with the grey-value morphology (graphs not shown).

4 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.

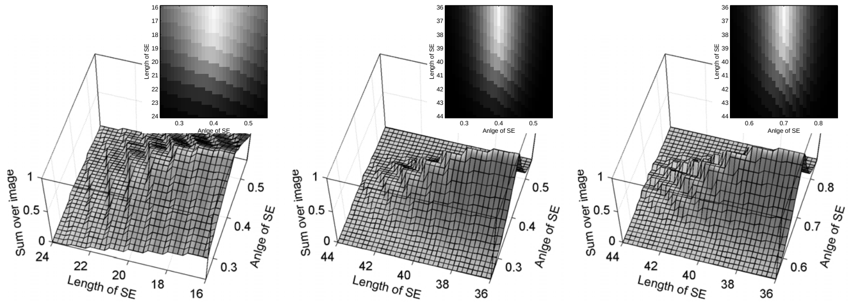

Fig. 8. Evaluation of method 4 (opening along an interpolated line). These graphs were obtained by changing the angle and length of the line segments in the input image. From left to right: 0.4 rad, 20 pixels; 0.4 rad, 40 pixels; and 0.7 rad, 40 pixels.

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. J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBS Systems Journal*, 4(1):25–30, 1965.
2. 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.
3. R. Jones and P. Soille. Periodic lines: Definition, cascades, and application to granulometries. *Pattern Recognition Letters*, 17(10):1057–1063, 1996.
4. R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.
5. 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.
6. G. Matheron. *Random Sets and Integral Geometry.* Wiley, New York, 1975.
7. 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.
8. P. Soille and H. Talbot. Directional morphological filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1313–1329, 2001.
9. 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.
10. M. van Herk. A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognition Letters*, 13:517–521, 1992.
11. L. J. van Vliet. *Grey-Scale Measurements in Multi-Dimensional Digitized Images.* PhD thesis, Pattern Recognition Group, Delft University of Technology, Delft, 1993.