

Detecting Digital Intersections Using Line Approximation^{*}

Szilvia Szeghalmy, Henrietta Tomán, András Hajdu

Faculty of Informatics, University of Debrecen, POB 12, Debrecen, H-4010, Hungary
e-mail: {szeghalmy.szilvia, toman.henrietta, hajdu.andras}@inf.unideb.hu

Abstract

In this paper, we investigate how digital line or curve segments enclosed by intersection points can be classified as valid or invalid intersections by matching pairs of intersecting lines onto the enclosed segment. Starting from the classic digital geometric definition of intersection points, we analyze whether the enclosed segment between any two connected intersection points can be considered as an intersection. To make this decision, we match intersecting lines and calculate their Hausdorff and chamfer distance from the enclosed segment, and from the curve parts leaving the intersection, respectively.

As preprocessing steps, we fill in holes that may appear between intersecting curves, and perform morphological thinning to be able to approximate with naive lines having 1- pixel width. From the enclosed segment we can calculate the minimal and maximal slope for possible matching lines. We also examine how these results can be used in detecting intersections of retinal vessels in digital fundus images.

Keywords: image processing, retinal image, digital line, line approximation.

1. Introduction

The literature of digital geometry contains a plenty of results regarding digital lines and curves (e.g. see [1] as a reference). However, much less results have been published on digital line and curve intersections. While the intersection of two Euclidean lines with different slopes is a single point, the intersection of two such digital lines can be a single discrete point, a set of discrete points or even empty on rectangular grids.

^{*}This work was supported in part by the Janos Bolyai grant of the Hungarian Academy of Sciences, and by the TECH08-2 project DRSCREEN- Developing a computer based image processing system for diabetic retinopathy screening of the National Office for Research and Technology of Hungary (contract no.: OM-00194/2008, OM-00195/2008, OM-00196/2008).

In Figure 1 we can observe how digital intersections may degenerate with containing more than 1 pixel.



Figure 1: Digital intersections. (a) non-degenerated case, (b) degenerated case.

Results presented in this field include an enumeration algorithm of the intersection pixels using quasiaffine applications [2]; a definition of the set of intersection pixels/voxels of two digital lines/planes using an unimodular matrix based on the arithmetical definition of a discrete line/plane [3]; the investigation of minimal carrier, connectivity and periodicity properties [4].

In many application fields, it has a great importance to differentiate between true intersections and enclosed segments connecting intersection points. We can mention e.g. digital curve compression, where a possible approach [5] is to assign an abstract curve graph to the object to be compressed, after which graph traversal algorithms can be executed. If intersections are not detected properly, the parity of the degree of the vertices may change which will ruin the performance of the graph algorithms that are very sensitive for this issue. Moreover, in several fields of medical imaging the investigation of the vascular system may lead to the recognition of diseases. Such a field is the early screening of diabetic retinopathy, where the proper mapping of the vascular system has prominent role. For example, a proper traversal of the vessels gives information about the rate how the thickness changes, or about the artery/vein (A/V) ratio. To reach these aims some empirical efforts have been made to detect intersections to be able to track a vessel (see e.g. [6]), however, proper theoretical analysis has not been executed yet. Figure 2 shows a naive example to see the difference between an intersection and an enclosed segment connecting two intersections.

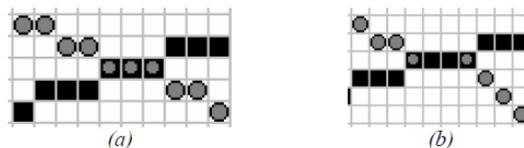


Figure 2: Cases to be differentiated. (a) intersection, (b) non-intersection (enclosed segment).

In this paper, we make the first step for an exhausting analysis of the intersection of retinal vessels with detecting them through matching intersecting digital lines to intersection candidates (enclosed segments). Since the intersection of digital lines may not be simply 8-connected [7] (that is, holes may occur within an

intersection), we fill in all the holes in the input set. Filling holes gives us the possibility to perform a morphological thinning [8] which makes the input image (vascular system) 1- pixel wide. In this representation we can locate junction (intersection) points using classic definitions. Since the thinning step does not take effect on the length of the enclosed segments between junction points, we can properly estimate the slope of digital lines containing the enclosed segment. Considering this allowed slope interval, we can match digital lines containing the enclosed section. The enclosed segment is accepted as an intersection, if we are able to find a pair of digital lines having minimal Hausdorff distance from the enclosed section, together with minimal chamfer distance from the inbound segments. The whole approach is depicted in Figure 3.

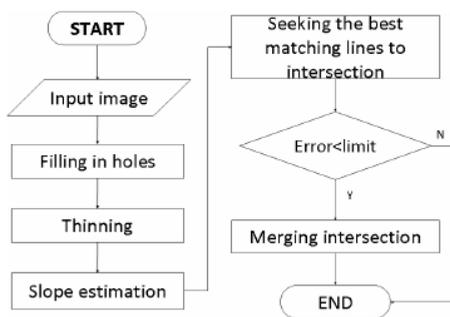


Figure 3: The flow chart of the approach.

The rest of the paper presents the details regarding the elements of our approach together with some results for artificial (line drawings) and real (retinal vessels) input.

2. Basic concepts and results

In this section we give some basic definitions (see e. g. [9]) that will be needed further.

Digital line: A digital line with parameters (a,b,c) is a set of pixels with integer coordinates (x,y) for which: $-1/2 \leq ax + by + c \leq 1/2$. This digital line has the slope $-a/b$ (with $b = 0$ for the vertical line). The points of the digital line e will be denoted by $(x, e(x))$.

Digital intersection: The digital lines e and f intersect each other if they have 8-connected pixels. The point $p(x_0, y_0)$ is called the beginning (end) of the intersection if p is a junction point (p has at least three 8-neighbors) and $x_0 < x$ ($x_0 > x$) holds for any other junction points $q(x, y)$.

Hole within an intersection: Let (x_1, y_1) and (x_2, y_2) be the respective coordinates of the beginning and end of the intersection of the lines e and f . If there exists an $x_1 < x < x_2$, such that $|e(x) - f(x)| > 1$, then we have a hole within the intersection

of e and f . Moreover, if there exists an $x_1 < x < x_2$, such that $|e(x) - f(x)| > 2$, then we call the hole as Z-shaped.

Length of staircases: It is well-known (see e.g. [7]) that any digital line consists of 8-connected horizontal line segments (staircases) having length difference at most 1.

3. Morphological preprocessing

To have a 1- pixel wide input for further processing, it is a usual approach to thin the input. In Figure 4 we can see the result of the thinning step for a digital intersection.

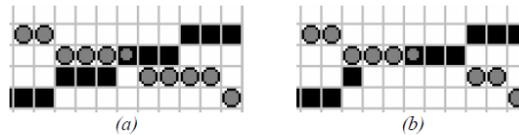


Figure 4: Thinning the input image. (a) before thinning, (b) after thinning.

However, in the case of intersecting digital lines/curves, even holes may appear within the intersection. So to be able to perform a successful thinning, holes should be filled in first.

3.1. Filling in holes

Since holes may appear not only at intersections, but also at other parts of the input, first we have to investigate what types of holes may appear between intersecting lines. Then, as a natural consequence, the filling algorithm will be restricted only to such types of holes. Figure 5a and 5b illustrate cases, when holes should or should not be filled in, respectively.



Figure 5: Holes in the input data. (a) a hole to be filled in, (b) a hole not to be filled in.

From [7] we know that if the slopes of the lines have different sign, then no hole can appear within their intersection. With the following two statements we perform a more detailed analysis to be able to describe holes that may appear within intersections. Thus, based on the shape of the hole, we can decide whether it should be filled in or not.

Proposition 1: *Let e and f be two intersecting lines with the same sign for their slope and containing staircases of length at most n . Then there may exist a 1- pixel wide hole of length at most n within their intersection. (See Figure 6a)*

Proposition 2: *A Z-shaped hole may occur only in that case if the two intersecting lines have the same sign for their slope and they consist of staircases of different length. (See Figure 6b)*

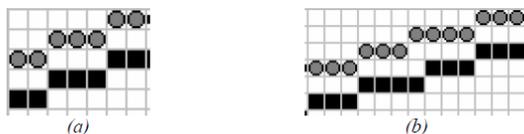


Figure 6: Hole within an intersection. (a) hole of 1- pixel width, (b) Z-shaped type hole.

As additional statements, we have the following ones:

- the beginning of a Z-shaped hole coincides with the beginning of the staircases of the intersecting lines, and its end with the end of them,
- other types of holes than discussed in Proposition 1 and 2 cannot occur inside an intersection. For example, such a hole cannot contain a 2×2 subhole.

The natural adoption of these results in the hole filling procedure is to fill in holes starting from junction points and having the shapes described in Proposition 1 and 2. Any other types of holes should remain unfilled in a line drawing input.

3.2. Thinning

After executing hole filling in the above discussed way, we perform thinning to be able to detect the beginning and end pixels of the intersections using the classic definition of junction points. The thinning procedure has to be selected to have a strict 8-connected, 1- pixel wide output. The literature of such thinning methods is quite rich, our implementation was based on [10].

Since we estimate the slope of the possibly intersecting lines from the enclosed section, we have to check whether thinning has influence on the length of this segment or not.(See Figure 7)

Proposition 3: *Thinning does not change the length of the digital intersections. The only trivial exception is the intersection of a horizontal and vertical line.*

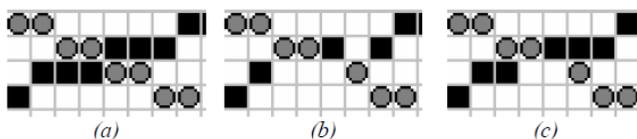


Figure 7: Thinning. (a) original intersection, (b) strict 8-neighborhood, (c) 4-neighborhood.

4. Slope estimation from intersection

After finishing thinning, we are ready to locate junction points based on the number of their pixel neighbors. Having the junction points located in this way, we consider all the segments enclosed by two junction points as intersection candidates. To test whether an enclosed segment is an intersection or not, we match pairs of lines that go through the enclosed segment and are also close to the segments leaving the intersection. However, to reduce the number of possible pairs of lines, we can estimate their slopes from the length of the enclosed segment considering it linear. Namely, if the intersection candidate is enclosed by the junction points $p(x_1, y_1)$ and $q(x_2, y_2)$, then the following inequalities hold:

$$-1/2 \leq ax_1 + by_1 + c \leq 1/2 \quad (1)$$

$$-1/2 \leq ax_2 + by_2 + c \leq 1/2 \quad (2)$$

where a, b, c are scalars, and the Euclidean line e passing through p and q has the following equation: $ax + by + c = 0$. Without the loss of generality we may assume that $x_2 > x_1$.

Now, from (1) and (2) we can give a minimal (s_{min}) and maximal (s_{max}) slope value for the lines going through the enclosed segment. Namely, we have

$$s_{min} = (y_2 - y_1 - 1)/(x_2 - x_1), \quad s_{max} = (y_2 - y_1 + 1)/(x_2 - x_1) \quad (3)$$

Also note that the largest slope interval is achieved with considering the beginning and end point of the enclosed section, so considering an intermediate point of the enclosed segment will not enlarge the possible slope domain.

The above slope estimation needs some further corrections, since it is possible that the beginning and end points of the enclosed segment will not belong to either of the lines (see Figure 7b). Thus the minimal and maximal slope estimation should be further loosened to cover these possibilities, as well. By recalling the issues we discussed in section 3 for thinning, the following completion is to be added.

After thinning, the line with maximum slope may pass the points $p'(x_1, y_1-1)$ and $q'(x_2, y_2+1)$ instead of the original junction points $p(x_1, y_1)$ and $q(x_2, y_2)$. In this case we get:

$$s_{max} = (y_2 - y_1 + 3)/(x_2 - x_1) \quad (4)$$

for the maximal slope. In a similar way, the points $p''(x_1, y_1+1)$ and $q''(x_2, y_2-1)$ may belong to the line with minimal slope so we have:

$$s_{min} = (y_2 - y_1 - 3)/(x_2 - x_1) \quad (5)$$

These extreme values for the slope of the line considering the junction points $p(x_1, y_1)$ and $q(x_2, y_2)$ cannot be further loosened because of the definition of the junction point.

In Figure 8 an example can be seen, when thinning changes the primary slope value.

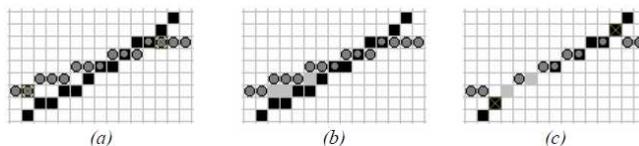


Figure 8: Maximal slope change caused by thinning. (a) original intersection, (b) result of filling in holes, (c) result of thinning.

Thus we have to match all the pairs of lines having slope values from the $[s'_{min}, s'_{max}]$ interval.

5. Matching pairs of lines

As the next step, we match pairs of lines having allowed slope onto the intersection. As a precise center, we consider the midpoint $((x_1 + x_2)/2, (y_1 + y_2)/2)$ of the segment enclosed by the $p(x_1, y_1)$ beginning and $q(x_2, y_2)$ end points. If p and q are very close together, that is their distance is less than 3, we immediately merge them with classifying the enclosed segment as an intersection.

To measure the goodness of matching of a line e onto the intersection candidate, we investigate both the matching of the line onto the enclosed segment and the matching of the line onto the segments leaving the intersection. More precisely, the goodness of matching is calculated based on $D_C(e)$ and $D_H(e)$, where:

- $D_H(e)$ is the Hausdorff distance [11] of the enclosed segment and the part of e having the same horizontal coordinates, that is, the part of e expected to contain the enclosed segment,
- $D_C(e)$ is the chamfer distance [12] of e and the intersection candidate. This chamfer distance is calculated in the following way: we create the distance map of the input image and sum its distance values along a segment of e of length L pixels centered at the point $((x_1 + x_2)/2, (y_1 + y_2)/2)$.

The actual matching process means the scan of the allowed slope domain for the best matching. First, we consider the line e having the slope s'_{min} . Then, if $D_H(e)$ is above a certain threshold T_H or $D_C(e)$ is above a certain threshold T_C we increase the slope of e with 1° . Otherwise, we start an iterative process to find the other line f by letting its slope start from s'_{max} and decrease it with 1° in every step. The slope of e and f can never be closer than a certain threshold V . In this way, we will find the best matching line pair for a fixed sloped e defined as $D_C(e) + D_C(f)$ and we go on with increasing the slope of e with 1° within the allowed slope domain. Note that this approach is faster than the brute force traversal of the complete slope domain. As a final decision, the candidate is accepted as an intersection if $D_C(e) + D_C(f) < T$ holds for the best matching pair of lines.

6. Experimental results

To check how the algorithm works, we executed tests based on natural data considering intersections of retinal vessels. Our main interest was to see whether line matching is feasible in this scenario or not.

In most of the cases we can encounter with vascular systems of types shown in Figure 9.

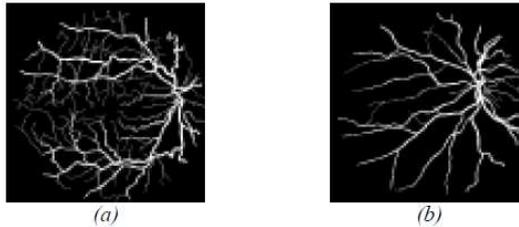


Figure 9: Vascular system of the retina; (a) macula centered, (b) optic disc centered.

If the image is centered at the macula (Figure 9a) then the vessels have higher curvature, while for an image centered at the optic disc (Figure 9b) they are more linear. As a consequence, for images centered at the macula contains more intersecting curve segments of larger curvature, and thus, the line matching approach gives less reliable result than for images centered at the optic disc. Since in our system we intend to process images of both types, we did not make difference between them and calculated an overall performance. Namely, our algorithm classified properly the 72% of the 130 manually segmented true intersections. To raise this figure we plan to improve our algorithm with considering quadratic interpolation (e.g. circular segments) instead of the linear one considered above.

In our experiments we have found the following setup of the parameters to be optimal: length of line segments matched $L = 40$; maximal allowed Hausdorff distance $T_H = 4$; maximal allowed chamfer distance $T_C = 240$; minimal slope difference between the lines $V = 10^\circ$; maximal chamfer distance of line pair $T = 240$.

Also note that in our experiments we used the distance map based on $\langle 3,4 \rangle$ chamfering [13], that is, the distance of 4-neighbouring pixels is equal to 3, while the distance of two diagonally neighboring pixels is 4.

In Figure 10 we can see an example for proper classification of retinal vessel intersection.

Also note that vessel segmentation algorithms usually return a hole-free vascular system. If not, small holes should be filled in, however, these holes are not so specific than the ones discussed in section 3.1.

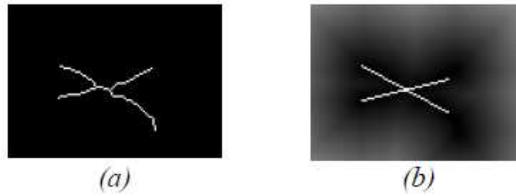


Figure 10: Detect retinal vessel intersections; (a) input image, (b) proper classification using matching intersecting lines.

7. Conclusion

In this paper we analyzed 1- pixel wide segments enclosed by junction points to see whether the segment is an intersection or not. We investigated the possible types of holes that may appear inside a line intersection. To locate the beginning and end of the intersection candidates, we filled in such holes and performed a thinning of the original set. We gave estimation for the slope of digital lines that may pass such a segment. To decide whether the segment is an intersection or not, we matched pairs of lines onto the intersection candidate and segments radiating from here using Hausdorff and chamfer distances. Our results look promising in detecting vessel intersections and also in making generalizations using higher order interpolations in matching.

References

- [1] Klette, R., and A. Rosenfeld, *Digital Geometry*, Morgan Kaufmann, San Francisco, 2004.
- [2] Jacob, M.A., *Applications Quasi-affines*, Ph.D. thesis, Universite Louis Pasteur, Strasbourg, France, 1993.
- [3] I. Debled, J.P. Reveilles, “A new approach to digital planes”, in: *SPIE Internat. Symposium on Photonics and Industrial Applications—Technical conference vision geometry 3*, 1994.
- [4] I. Sivignon, F. Dupont, and J.M. Chassery, “Digital Intersections: minimal carrier, connectivity, and periodicity properties”, *Graphical Models 66(4)*, 2004, pp. 226-244.
- [5] A. Hajdu, and I. Pitas, “Piecewise linear digital curve representation and compression using graph theory and a line segment alphabet”, *IEEE Trans. on Image Processing 17/2*, 2008, pp. 126-133.
- [6] K. Rothaus, P. Rhiem, X. Jiang, “Separation of the retinal vascular graph based upon structural knowledge”, *Image and Vision Computing 27/7*, 2009, pp. 864-875.
- [7] Reveilles, J.P., *Geometrie discrete, calcul en nombres entiers et algorithmique*, Ph.D. thesis, Universite Louis Pasteur, Strasbourg, France, 1991.
- [8] Serra, J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

- [9] S. Pham, "Parallel, overlapped, and intersected digital straight lines", in: *Proceedings of The Visual Computer*, 1988, pp. 247-258.
- [10] Gonzalez, R.C., R.E. Woods, *Digital image processing*, Prentice Hall, Upper Saddle, River, New Jersey, 2002.
- [11] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the Hausdorff distance", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15/9, 1993, pp. 850-863.
- [12] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching." in *Proc. 5th Int. Joint Conf. Artificial Intelligence*, Cambridge, MA, 1977, pp. 659-663.
- [13] G. Borgefors, "Distance transformations in digital images", *Computer Vision, Graphics, and Image Processing* 34/3, 1986, pp. 344-371.