TOWARDS THE GENERALISATION OF DISTANCE TRANSFORM FOR LINE PATTERNS

Sergey V Ablameyko* and Tony P Pridmore+

*Institute of Engineering Cybernetics

Belarussian Academy of Sciences

Surganov Str. 6, 220012 Minsk

Belarus

+School of Engineering Information Technology,

Sheffield Hallam University

Harmer Building, City Campus, Sheffield S1 1WB

United Kingdom

Abstract

The notion of Generalised Distance Transform for Line Patterns (DTLP) is introduced in the paper. The Generalised DTLP models distance transforms on line data as the application of a combination function to values derived from the lengths of a set of paths to feature points from the pixel of interest. Previous distance transforms for line patterns are assessed within the proposed framework. Algorithms computing transformations defined by the general form are presented and discussed. Transforms based upon a variety of combination functions and distance-related values have been implemented; the results of their application are considered.

1. Introduction

One of the basic and well developed tools in low-level image processing is a distance transform. The distance transform (DT) is an operation that takes a binary image as its input and generates a grayscale image as output. Each pixel in grayscale image has a value which is equal to the distance to the closest set pixel (object or background) in the initial binary image [3]. The Distance Transform of Line Patterns (DTLP) [9] is a particular case of DT where the input image is assumed to be a binary line pattern, i.e. to contain objects with one pixel wide almost everywhere. Data of this type is typically generated via edge detection or by thinning binarised images of, for example, line drawings. In the DTLP the ends of the data point strings are considered to be features, the remainder of the line data are non-feature pixels. The effect of the DTLP is to assign to each pixel on the line pattern some measure of the distance to an end point. There are two main forms of DTLP, which they term Type I and Type II. The latter measures the distance to the nearest end point; the former, given simple figures at least, records the distance to the furthest such feature.

More recently, a modified DTLP has been proposed [7] in which both terminations and intersections are considered to be features. This algorithm produces, as a side effect, data structures describing feature points (terminations, intersections) and the branches (strings of 2-connected pixels) which link them.

The DTLP is not widely developed because it does not adequately reflect the structure of the line data. The line patterns used in [9] are line structures; they comprise fairly clean, curvilinear segments which intersect at often noisy, multi-pixel nodes. These patterns may be thought of as graphs: a distance transform for line data will only be of use if the measures it supplies give useful information about the structure of those graphs. Although valuable and interesting work has been done on the extraction of symbolic representations (tables, networks, etc.) which make explicit the graph structure of line patterns we do not propose to extend that work here. Instead we want to show that much of this information can be stored in a (generalised) distance map.

In general, there are many paths from each element of a line pattern to an appropriate feature pixel. A major issue to be addressed during the interpretation of line data is which paths are important and which are not. It is our belief that the role of a distance transform should be to provide information which supports higher level systems in making this decision. The distance maps generated by the type I and II DTLPs however, effectively ignore the graph structure of their data, choosing instead to provide information about only one path from each point - the longest or shortest to a feature point. The process intended to support decision making implicitly makes the decision. The modified DTLP clearly takes more account of the structure of its data; though it too performs a segmentation, this time explicitly. Recall that it was their consideration of this segmentation which lead them to question the value of their distance map.

In this paper, we propose a general form of the distance transformation for line patterns. This views previous workers' selection of the maximum or minimum length path as particular instances of the application of a combination function to values derived from the lengths of a set of paths to feature points from the pixel of interest. Algorithm implementing members of the set of distance transforms defined by the general form is shortly mention. Transformations employing a variety of combination functions and distance-related measures have been implemented.

2. Definition of the Generalised Distance Transform for Line Patterns

We seek transforms which provide information about all relevant paths from a given pixel to appropriate feature pixels and propose a generalised form of the distance transform for line patterns as follows.

A central feature of our generalised transform is the combination function

$$Cxy(gxy1, gxy2, gxy3, \ldots, gxyn)$$

where each $gxyi = G(lxyi)$ and $lxyi = L(pxyi)$ i.e. $gxyi = G(L(pxyi))$

In the above, $lxyi = L(pxyi)$ is a measure of the length of a path pxyi from a data point at pixel (x,y) to some appropriate feature pixel and $gxyi = G(lxyi)$ is a function of that length. In traditional distance transforms $G(lxyi) = lxyi$ in general, however, G(lxyi) may measure any useful path property.

Each path pxyi comprises an ordered list of pixels

$$pxyi = ((x1, y1), (x2, y2), (x3, y3), \ldots\ldots, (xm, ym))$$

such that $x1 = x$, $y1 = y$, (xm, ym) is a feature pixel and adjacent members of the list are neighbours within the input line pattern. More formally, we define a binary valued feature detection operator F and a binary valued neighbourhood test N and require that

$$F(xm, ym) = 1 \quad \text{and} \quad N(xi, yi, xi+1, yi+1) = 1 \quad \text{for } 1 <= i < m$$

In general, the set of paths

$$Pcxy = \{pxyi : 1 \leq i \leq n\}$$

contributing to

$$Cxy(G(L(pxyi)) : 1 \leq i \leq n)$$

is a subset of the set of all paths Pxy to a feature pixel from (x, y). Membership of this subset is determined by the binary valued selection function Sxy(pxy). Hence

$$Pcxy = \{pxyi : pxyi \quad Pxy \;\&\&\; Sxy(pxyi) = 1\}$$

The proposed scheme, which we refer to hereafter as the Generalised Distance Transform for Line Patterns (GDTLP) admits many transformations, a given transform T being defined by

$$T = \{Cxy, Sxy, G, L, N, F\}$$

Care must be taken when selecting the components for a given T as many of the admissible transforms will provide no useful information about the structure of the line pattern.

Output from the GDTLP is a generalised form of distance map, which we term the Generalised Distance Map for Line Patterns; the GDMLP. In its pure form this comprises a (probably real-valued) array containing at each data pixel the value of $Cxy(G(L(pxyi)) : 1 \leq i \leq n)$

determined by T. Obstacle/background pixels we assume to contain zeros. Some transformations may compute useful information as a stepping stone to Cxy(G(L(pxyi))). To enable this to be retained we further allow each pixel of the GDMLP to be associated with a pointer to an optional secondary data structure. The utility, size and content of these secondary structures also varies with T.

3. GDTLP and its relations with other DTLP types

In DTLP Type II an initialisation phase sets the distance value of background pixels to 0, termination points to 1 and all other pixels to invalid. Pixels with an invalid distance only receive a value when at least one of their neighbours holds a valid distance; in these circumstances the value assigned is the minimum of all the valid neighbours + 1. This may be expressed within our generalised model as follows:

N(xi, yi, xj, yj) = 1 if (xi, yi) and (xj, yj) exhibit 4-, 8- or mixed connectivity, 0 otherwise

L((x1, y1), (x2, y2), (x3, y3), ..., (xk, yk)) = k

i.e. L(pxyi) provides a simple count of the number of pixels in pxyi

G(lxyi) = lxyi

F(x,y) = 1 if exactly 1 point (xk, yk) st N(x, y, xk, yk) = 1,
0 otherwise

and

Sxy(p) = 1 i.e. all paths are (effectively) considered.

while

Cxy(L(pxyi) : pxyi ∈ Pcxy) = MIN(L(pxyi) : pxyi ∈ Pcxy)

i.e. the shortest path to a feature point is selected and its length reported.

The Modified DTLP is very similar, differing only in its definition of F. In the MDTLP

F(x,y) = 1 if :
exactly 1 point (xk, yk) st N(x, y, xk, yk) = 1 or
more than 2 points (xk, yk) st N(x, y, xk, yk) = 1 or

$Nc8(x,y) < 2$ and $Nc8(x,y)$ is the connectivity number

$F(x,y) = 0$ otherwise

Type I DTLP employs the same initialisation as their Type II transform. Thereafter, invalid distances are replaced only when the pixel concerned has not more than one neighbour with an invalid distance value. The pixel is then assigned the maximum of the valid neighbours + 1. In terms of our model, L, G, N and F are as for the Type II transform and

$$Cxy(L(pxyi) : pxyi \in Pcxy) = MAX(L(pxyi) : pxyi \in Pcxy)$$

i.e. the longest member of Pcxy is selected and its length recorded. The definition of Pcxy is, however complex. Assume that a given data point (x,y) has r neighbours

$$(x1, y1), (x2, y2), (x3, y3), \ldots\ldots, (xr, yr)$$

and partition the set of all paths from (x,y), Pxy into r subsets Pixy such that

$$Pixy = \{pxyj : pxyj \in Pxy \text{ and } (xi, yi) \in pxyj\}$$

i.e. all paths in Pixy must pass through (xi, yi) and $N(x, y, xi, yi) = 1$.

Now let $Limax = Lmax(Pixy)$ be the length of the longest path in each Pixy, defined according to L, and

$$PMAXxy = Pixy \text{ such that } Limax = MAX(L1max, L2max, L3max, \ldots, Lrmax)$$

then

$$Sxy(pxy) = \begin{cases} 1 \text{ iff } pxy \in (Pxy - PMAXxy) \\ 0 \text{ otherwise} \end{cases}$$

The complexity of the path selection function implicit in the Type I DTLP may be one reason for its infrequent use by the image analysis community, though, as Toriwaki et al point out, the transform has a number of interesting properties.

5. Discussion

There exist many papers devoted to the description and decomposition of complex binary patterns. The papers [2,5,8] are devoted to the description and decomposition of thinned binary patterns: line patterns. The representation produced is usually symbolic - a network or table of segments - and intended to support recognition Such methods are usually inspired by a particular applied task; vectorisation of maps, for example. As a result, they typically produce descriptions which are only really suited to that one task. A more general approach to object decomposition is described in [1]. Their algorithm produces a table making explicit both branches and loops and supplies a hierarchical object description based upon various properties of the table elements.

Our approach differs in that we attempt to store as much useful information as possible in a generalised distance map. This map and the Generalised DTLP which produces it are both very flexible; the component functions of the GDTLP may be varied to provide a transform and map appropriate to the task at hand.

The GDTLP extends the DTLPs and MDTLP in two ways. First, it considers properties other than simple length. In this it owes much to the algorithm of [1]. Second, it recognises that there usually exist many paths from any given pixel to a feature and allows measurements of the properties of those paths to be combined in a flexible manner. It is not yet clear which of these generalisations is the most valuable. What is needed now is experience of applying these transforms to real problems.

Although much remains to be done in this area, we believe that the GDTLP will prove to be a valuable aid to the interpretation, and particularly segmentation, of line patterns. We see GDTLPs providing contextual information to knowledge-based, application-specific systems such as ANON [6]. Some such systems incorporate low level processes which are loosely related to the GDTLP, though these typically lack any theoretical underpinning.

GDTLPs may also form the basis of more general segmentation schemes. In [1], authors propose a hierarchical description based upon the areas of pattern segments. Their description can be tuned to particular applications by considering different segment properties and their descriptive process can be thought of as operating on a Generalised Distance Map for Line Patterns.

Throughout this paper we have concentrated on the interpretation of thinned binary data and assumed that the ambiguities to be resolved are between segments rather than pixels. That is, the question to answer is which segments should form part of an object, not which pixels should be part of a segment. In other circumstances, e.g. in certain types of edge data, the latter question is equally important. Many systems, often based on line tracking techniques, have been developed to address this problem. These trackers typically suffer from a lack of global contextual information in the same way as do higher level, knowledge-based interpretation systems. They too would benefit from the information supplied by an appropriate (G)DTLP.

References

1.Ablameyko, S., C. Arcelli and G. Sanniti di Baja, Finding and ranking basic structures on complex line patterns, Proc. SSPR Workshop, Israel, 1994.

2.Ablameyko, S., V. Bereishik, N. Paramonova, A. Marcelli, S. Ishikawa and K. Kato, Vectorisation and representation of large-size 2D line drawing images, Journal of Visual Communication and Image Representation, 5, pp 245-254, 1994

3.Borgefors, G., Distance transformations in arbitrary dimensions, Computer Vision, Graphics and Image Processing, 27, pp 321-345, 1984.

4.Joseph, S.H. and T.P Pridmore, Knowledge-directed interpretation of mechanical engineering drawings, IEEE Trans. PAMI, 14, pp 928-940, 1992.

5.Pavlidis, T., A vectorizer and feature extractor for document recognition, Computer Vision, Graphics and Image Processing, 35, pp 111-127, 1986.

6.Pridmore T. and S.Ablameyko, Generalised distance transform for line patterns, Submitted to Pattern Recognition, 1995.

7.Ragnemalm, I. and S. Ablameyko, On the distance transform of line patterns, 8th SCIA, Thomso, Norway, 1993.

8.Suzuki, S., Graph-based vectorization method for line patterns, in Proc IEEE Conference on Computer Vision and Pattern Recognition, Ann Arbor, Michigan, pp 616-621, 1988.

9.Toriwaki, J, N. Kato and T. Fukumura, Parallel local operations for a new distance transformation of a line pattern and their applications, IEEE Trans. Systems, Man and Cybernetics, 9, 10, 1979