Danielsson's algorithm generalization: an experiment

R. Juan

Report LSI–92–29–R

Danielsson's Algorithm Generalization: An Experiment

Robert Juan*
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

July 1992

*While on leave in Computer Sciences Department, Purdue University. Partially supported by a NATO Scientific Programme fellowship and by a fellowship of the Spanish Ministerio de Educación y Ciencia

Danielsson's Algorithm Generalization: An Experiment

Robert Juan*
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

July 1992

Abstract

This writing deals with some issues arised while attempting to generalize Danielsson's algorithm for computing a exact Euclidean distance map on the vertices of a topologically compatible triangulation of a two-dimensional plane polygonal domain.

1 Introduction

It is well know that distances measure is central to many applications of digital picture processing [8], [9] where several algorithms for computing such distances with reasonable approximations of the Euclidean distance have been proposed. [2]. [3]. [5]. [6]. [10].

All of these algorithms share two characteristics: the need for a rectangular regular grid over which to compute de distance map and, the fact that the computed map is both approximate and not Euclidean –for exemple, triangular inequality does not necessarily hold and the map is orientation dependent. These limitations preclude us from applying directly those algorithms in computational geometry and solid modelling. Hence, to develop algorithms to compute exact Euclidean distance maps is of interest.

We tried to develop an algorithm for computing a two-dimensional discret exact Euclidean distance map. Given a set of points, hereafter called vertices, which define a topologically compatible Delaunay triangulation over a

*While on leave in Computer Sciences Department, Purdue University. Partially supported by a NATO Scientific Programme fellowship and by a fellowship of the Spanish Ministerio de Educación y Ciencia

two-dimensional domain, we are asked to find for each vertex in the triangulation the nearest vertex in the domain boundary. This algorithm should be viewed as a generalization of Danielsson's algorithm. Conceptually, it has the same logical structure [2], [3]: the distance map should be produced in two domain scans, top-to-bottom (forward pass) and bottom-to-top (backward pass). Vertices in the triangulation should behave like pixels of a picture while the monotonicity induced by the regular rectangular grid in Danielsson's algorithm is achieved through the use of a data structure that stores the vertices sorted.

Montanari proved in [5] the existence of minimal paths that guarantee the two-pass convergence of the general dynamic programming problem to which the map distance computation is equivalent [1]. We will see that these paths do not always exist on a constrained Delaunay triangulation so the generalization is not possible.

2 Definitions

The basic definitons that we will need are the following.

Definition 2.1 Any vertex v in a triangulation T, that is not a boundary vertex is a interior vertex.

Definition 2.2 Given a triangulation T of a set of vertices, two vertices v_i, v_j in T are neigbours of each other if the straight-line segment connecting them is a edge of T. The set of all of the neighbours of a vertex v, $\mathcal{N}(v)$, defines the neighbourhood of v.

The set of vertices in $\mathcal{N}(v)$ for which the y coordinate value is greater than or equal to that of v is denoted by $\mathcal{N}^+(v)$. Similarly, $\mathcal{N}^-(v)$ denotes the vertices in $\mathcal{N}(v)$ with y coordinate value less than or equal to that of v.

Definition 2.3 A triangulation is topologically compatible with the domain on which it is defined if no triangle cuts across the domain boundary, i.e. it accurately represents the topology of the domain, [7].

In order to generalize those concepts used in [5] we need the following definitions.

Definition 2.4 A *path* in a triangulation T is a set of vertices $\{v_1, \ldots, v_p\}$ and edge set $\{(v_i, v_{i+1}) : i = 1, \ldots, p-1\}$ such that each vertex is shared by no more than two edges.

Definition 2.5 A *monotone path* in a triangulation T is a path such that its vertices have either non increasing or non decreasing y coordinate values.

Definition 2.6 A *bi-tonic path* in a triangulation T is a path such that there is a vertex v with $v \neq v_1$ and $v \neq v_p$ which splits the path into two monotone

paths.

3 The algorithm

A intuitive generalization of Danielsson's algorithm [2], [3], can be written as a *scan-line algorithm* [4] in which the monotonicity implied by the regular grid is provided by scanning the triangulation vertices in a *bucked-sorted* list [4]. For each vertex v, the sets $N^+(v)$ and $N^-(v)$ are stored in two lists.

Assumed that the triangulation T is held in a static variable and that there is at least one interior vertex, the main algorithm visits each vertex in the triangulation and gives initial value to the data structures. Then performs the forward and backward domain scans.

```
algorithm Exact_euclidean_distances_on_a_triangulation ()

    for each vertex v in T do
        if Boundary_vertex (v) then
            Initialize_boundary_vertex (v)
        else
            Initialize_interior_vertex (v)
        endif
    enfor
    Forward_pass ()
    Backward_pass ()
endalgorithm
```

The forward pass starts to scan the domain from the vertices with the greatest value of y coordinate. The current distance map value is updated for all those vertices belonging to the current entry in the bucket-sorted list: $\{v_{y1}, \ldots, v_{ym}\}$. The initial distance from the domain boundary to a given vertex is either infinity, when the vertex is visited for the first time, or the last updated value.

```
procedure Forward_pass ()

    for y = 1 to nrows do
        for v = v_y1 to v_ym do
            dmin := infinity
            First_neighbour_forward (v, nv)
            while (nv ≠ nil)  do
                Update_reference_value (dmin, v, bp)
```

```
            Next_neighbour_forward (v, nv)
        endwhile
    endfor
endfor
endprocedure
```

The algorithm for the backward pass scans the domain just in the oposite way than the forward pass.

```
procedure Backward_pass ()

    for y = nrows to 1 step -1 do
        for v = v_ym downto v_y1 do
            dmin := v.dx * v.dx + v.dy * v.dy
            First_neighbour_backward (v, nv)
            while (nv ≠ nil)   do
                Update_reference_value (dmin, v, bp)
                Next_neighbour_backward (v, nv)
            endwhile
        endfor
    endfor
endprocedure
```

4 Why the algorithm is incorrect

Obviously, the algorithm will be correct if after the backward pass each interior vertex has been associated with its nearest boundary vertex. In order to prove this, we previously should prove that every interior vertex is connected to its nearest boundary vertex at least through either a monotone path or two different bi-tonic paths of edges in T. One bi-tonic path should have descending-ascending sequence with respect increasing values of x coordinate, and the other bi-tonic path should have ascending-descending sequence. All the vertices in these paths should have the same nearest boundary vertex. These paths should play the same roll as minimal paths play in [5].

Unfortunately, the existence of these paths is easily disproved. See Fig. 1 showing a constrained Delaunay triangulation where $v_1^{'}$ and $v_2^{'}$ are respectively the nearest boundary vertices of v_1 and v_2 and there are no such paths connecting the interior vertex v_j and its nearest boundary vertex v_i. Furthermore, in the case illustrated in Fig. 1 vertex v_j will never be associated to its nearest

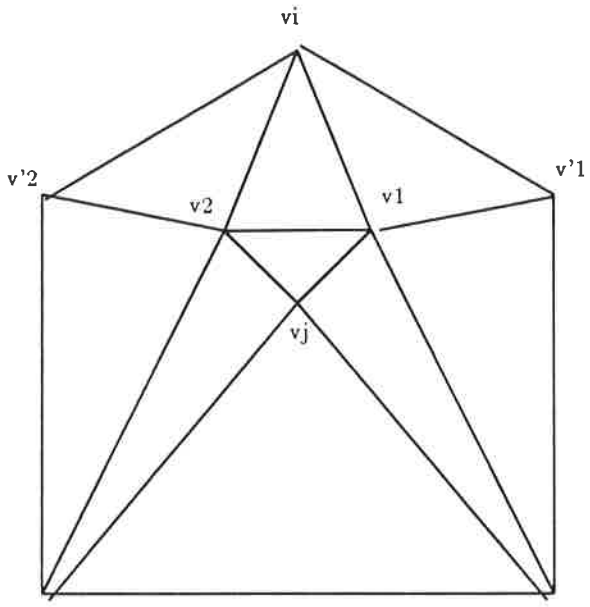

Figure 1: Disproving the existence of paths.

boundary vertex v_i even if we let the algorithm to iterate an unlimited number of times.

5 Conclusions

We have shown that Danielsson's algorithm can not be extended in a straightforward way to constrained Delaunay triangulations. The conditions in Danielsson's algorithm, derived from the use of a regular grid, allowed the existence of minimal paths with properties that make the general dynamic programmming iterative algorithm to converge in two passes over the grid. For more general grids, there are no such minimal paths so, we need to go back to the general iterative algorithm. Furthermore, a triangulation has been presented in which the general iterative algorithm, as can be derived from the algorithm we have formulated here, does not converge to the true solution.

References

[1] R. Bellman and S. Dreyfus. *Applyed Dynamic Programming.* Princeton University Press, Princenton, N.J. 1962, pp. 229–231.

[2] G. Borgefors. Distance Transformations in Arbitrary Dimensions. *Computer Vision, Graphics, and Image Processing.* **27**, 321–345 (1984).

[3] P.-E. Danielsson. Euclidean Distance Mapping. *Computer Vision, Graphics, and Image Processing.* **14**, 227–248 (1980).

[4] J.D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes. *Computer Graphics. Principles and Practice.* Addison Wesley, 2nd Edition. 1990.

[5] U. Montanari. A Method for Obtaining Skeletons Using a Quasi-Euclidean Distance. *Journal of the Association for Computer Machinery*, Vol. 15, No. 4, October 1968, pp. 600–624.

[6] U. Montanari. Continuous Skeletons from Digitized Images. *Journal of the Association for Computer Machinery*, Vol. 16, No. 4, October 1969, pp. 534–549.

[7] M.A. Price, T.K.H. Tam, C.G. Amstrong and R.M. McKeag. Computing the Branch Points of the Voronoi Diagram Using a Point Delaunay Triangulation Algorithm. Draft manuscript. January 1991.

[8] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, 2nd ed., Vol. 2. Academic Press, New York, 1982

[9] A. Rosenfeld and J. Pfaltz. Sequential Operations in Digital Picture Processing. *Journal of the ACM*, **13**, 471–494, 1966.

[10] A. Rosenfeld and J. Pfaltz. Distance Functions on Digital Pictures. *Pattern Recognition*, **1**(1), 1968.