Circular Shortest Path on Regular Grids*

Changming Sun
CSIRO Math & Information Sciences
Locked Bag 17
North Ryde, NSW 1670, Australia.
changming.sun@cmis.csiro.au

Stefano Pallottino
Dipartimento di Informatica
Università di Pisa
Corso Italia 40, 56125 Pisa, Italy.
pallo@di.unipi.it

Abstract

Shortest path algorithms have been used for a number of applications such as crack detection, road or linear feature extraction on images. In this paper, we presents several new algorithms for the extraction of a circular shortest path on regular grids such that the starting and ending positions are connected. The new algorithms we developed include multiple search algorithm, image patching algorithm, multiple backtracking algorithm, the combination of image patching and multiple back-tracking algorithm, and approximate algorithm. The typical running time of our circular shortest path extraction algorithms on a 256×256 image is in the order of 0.3 seconds on a rather slow 85MHz Sun SPARC computer. A variety of real images for crack detection on borehole data, object boundary extraction, and panoramic stereo matching have been tested, and good results have been obtained.

1. Introduction

In a weighted graph or network, it is frequently desired to find the shortest path between two nodes. The shortest path is defined as a path from one node to the other such that the sum of the weights of the arcs on the path is minimised. Most algorithms or applications in the graph framework use a labeling approach, in particular the one due to Dijkstra [4, 2].

Buckley and Yang developed a regularised shortest path extraction algorithm using dynamic programming (DP) on rectangular images for crack detection on borehole data/images and for road extraction on remote sensing images [1]. There is no constraint on the starting and ending positions of the path. A number of authors used dynamic programming technique to obtain a shortest path in a rectangular matrix for stereo disparity measurement [9, 6, 5, 7]. All these applications impose no constraints on the starting and ending positions of the shortest path.

*Research partially supported by grants: MURST and INDAM-GNAMPA of Italy.

Borehole geophysics is used in ground-water and environmental investigations to obtain information on well construction, rock lithology and fractures, permeability and porosity, and water quality [8]. The borehole acoustic televiewer resembles an optical television camera system in producing a full 360° image of the borehole walls. Fig. 3(a) shows a full 360° borehole image with cracks in it. As the image is a 360° circular image, the left and the right boundaries of the image are actually neighbouring columns. This image can be shown in a cylindrical format as in Fig. 3(b). In the example shown in the figure, a closed or circular shortest path should be extracted.

In some image analysis applications, object boundaries need to be extracted [3]. In these applications, it is necessary to make sure that the boundary extracted are closed contours. Panoramic stereo images are becoming available for 3D applications. In 360° panoramic stereo images, the left and the right columns are connected with each other. Therefore in the stereo matching process, it is necessary to take this constraint into account. This can be achieved by obtaining a circular path in the correlation coefficient matrix.

In this paper we address the issue of obtaining a circular shortest path on a regular grid for a number of applications. Section 2 gives a brief review for ordinary shortest path extraction algorithms. Section 3 presents five new methods for circular shortest path extraction on images. Section 4 shows the experimental results obtained using our circular shortest path extraction methods applied to a variety of applications. Section 5 gives concluding remarks.

2. Ordinary Shortest Path Algorithms

This section gives a brief review on ordinary shortest path extraction algorithms using dynamic programming. The problem is to find a path from the left side to the right side of an image or grid such that the cost of the path is minimum. The cost of the path is the sum of the costs along the path. A huge number of scientific papers are devoted to the shortest path problem. We refer to [4, 2] for a general view of the problem and of the most efficient algorithms proposed.

The shortest path problem in the grid is a problem on an acyclic graph. In fact, from an array of u rows and v columns, we can derive a directed graph $G = (N, A)$, where each node $i \in N$ is a pair $[h(i), k(i)]$ where $h(i)$ indicates the row and $k(i)$ indicates the column of the element represented by i. The number of nodes is $n = uv$.

An arc $(i, j) \in A$ exists if $k(j) = k(i) + 1$ and $h(j) = h(i) + \alpha$, where $\alpha = -1, 0, 1$ but the extreme cases in which $h(i) = 1$ or $= u$; the cost of arc (i, j) is set to the entry associated to the pair $[h(i), k(i)]$, i.e. all the arcs leaving node i have the same cost, and it is the value at position $[h(i), k(i)]$. With this transformation the shortest path problem on the grid is mapped into a shortest path problem on a classical graph. By using this transformation, we can see that our problem has another special characteristic: the graph G is a *stable acyclic sequential layered graph.*

Let us suppose that we have to solve the shortest path problem from the nodes in layer L_1 to the nodes in the last layer L_v. The best approach is to exploit the characteristics of the graph by analysing one layer at a time and, for each node j of that layer, by setting the optimal label value d_j by using DP technique. The resulting algorithm is

Procedure Layer_Grid():
  begin {initialising the labels of nodes of L_1}
    for each $j \in L_1$ **do**
      begin
        $d_j := 0$;
        $p(j) := nil$;
      end
    for $h := 2$ **to** v **do**
      for each $j \in L_h$ **do**
        begin {working on layer L_h}
          $d_j := \min\{d_i + c_{ij} : (i, j) \text{ enters } j\}$;
          $p(j) := \text{argmin}\{d_i + c_{ij} : (i, j) \text{ enters } j\}$
        end
  end.

The time complexity of the algorithm is linear in the number of graph arcs and therefore it is $O(uv)$.

3. Circular Shortest Path

The algorithms described in the previous section for ordinary shortest path extraction impose no constraint on the starting and ending position of the path. From now on, we assume that the image or grid is circular, that is the first and the last columns are neighbours. In this section, we will present several new algorithms for circular shortest path (CSP) extraction where the starting and the ending positions of the obtained path are connected. A circular path is a path from the first column to the last column when the starting and ending position are connected. A circular shortest path is a circular path when its cost among all the circular paths are minimum. Fig. 1 gives an example showing the different paths obtained using the ordinary shortest path and a circular shortest path extraction algorithm (to be described later). The "*" symbols in the figure indicate the positions of the shortest paths. The starting position (Ba) and the ending position (Dh) of the ordinary shortest path are not directly connected as shown in Fig. 1(a). The starting and ending positions (Da) and (Dh) of the circular shortest path are actually neighbouring points as shown in Fig. 1(b).

	a	b	c	d	e	f	g	h
A	252	245	74	13	219	171	193	242
B	*160	*83	*23	103	71	214	174	30
C	202	117	157	*90	202	102	104	235
D	161	143	231	127	*20	*20	*63	*39
E	210	63	64	192	28	36	130	170
F	235	83	129	66	233	237	41	147

(a)

	a	b	c	d	e	f	g	h
A	252	245	74	13	219	171	193	242
B	160	83	23	103	71	214	174	30
C	202	117	157	90	202	102	104	235
D	*161	143	231	127	20	*20	*63	*39
E	210	*63	*64	192	*28	36	130	170
F	235	83	129	*66	233	237	41	147

(b)

Fig. 1: *Examples showing the different paths obtained using ordinary and circular shortest path extraction. It is assumed that the column "a" and column "i" are neighbouring columns. (a) Shortest path without constraint. The starting and ending positions (Ba) and (Dh) are not neighbours. (b) Shortest path with constraint, i.e. circular shortest path. The starting and ending positions (Da) and (Dh) are neighbours.*

3.1. Multiple Search Algorithm

To find the required circular shortest path, one can run the ordinary shortest path algorithm for acyclic graphs u times (u is the number of rows in the image or grid), one for each node $[h, 1]$ of the first column as origin. Once computed the shortest paths for all the nodes of the last column, we select as the best circular shortest path the least cost path among the ones terminating at the nodes $[h-1, v]$, $[h, v]$ and $[h+1, v]$ (to satisfy the constraint that the starting and ending positions are neighbours). At the end of the u shortest path computations, we can select the path with the least cost to be our result. This is our multiple search algorithm (MSA).

This method will guarantee to find the path which satisfies our constraints. The disadvantage of this method is that the ordinary shortest path algorithm for acyclic graphs has to be run u times. Therefore it has a time complexity $O(u^2 v)$.

3.2. Image Patching Algorithm

In this subsection we will present a fast algorithm for obtaining the required circular shortest path by working with patched images. We call this the image patching algorithm (IPA).

The size of the patches could depend on the type of applications or the content of the images. Fig. 2 shows the image patching process for obtaining the circular shortest path. Patch-1 and Patch-2 are parts of the original image. Copy-of-Patch-1 and Copy-of-Patch-2 are copies of the image regions Patch-1 and Patch-2. These two copies of the local image regions are attached to the original image to build a larger image. Image patching is only performed in the X-direction of the image, as we need to find the circular shortest path from left to right of the image. If a shortest path from top to bottom of the image is needed, the patching can be done at the top and the bottom of the image. Fig. 2(a) illustrates the patching process, and Fig. 2(b) is an example of a patched image. Dark lines are drawn in Fig. 2(b) to show the image boundaries. The left side of the first dark line is the same region for Patch-2. The right side of the second dark line is the same region for Patch-1.

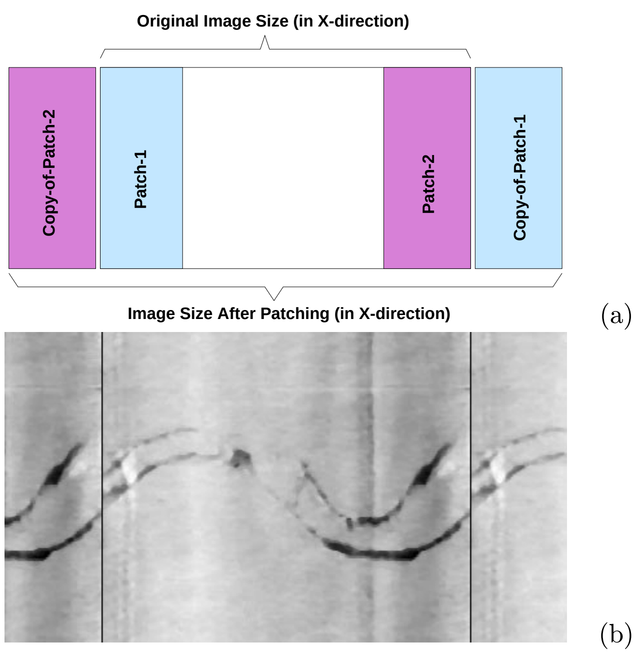


Fig. 2: *Image patching for fast circular shortest path extraction. (a) Drawings showing the patching process; (b) Illustration using a real image. Dark lines in this image are artificial. It is used to show the region boundaries.*

The image patching method does not guarantee to find the required path. However many synthetic and real image tests all produce correct results. If a circular shortest path is not found, we can iterate the process of finding circular shortest path by using a different size of the patch, or using a multiple back-tracking algorithm (MBTA) to be described in the following subsection. Or if the application is not time critical, the MSA method can be used. The main advantage of the IPA algorithm is its speed, as it only needs one run of the ordinary shortest path extraction algorithm on the patched image. The complexity of the algorithm is $O(u(v+k))$, where k is the width of the added patches.

The steps of our image patching algorithm for circular shortest path extraction are:

1. Patch the input image on the left and the right sides with portions of the input image itself (say one-eighth of the image width).
2. Perform ordinary shortest path extraction using DP on the patched image.
3. Extract the shortest path which lies inside the original image.
4. Check if the obtained path satisfy the circular constraint. If so, go to Step 5; otherwise, go to Step 1 with a different patching size, or using MBTA or MSA.
5. Display circular shortest path.

3.3. Multiple Back-tracking Algorithm

In this subsection we will present another algorithm based on the ordinary shortest path algorithm by performing multiple back-tracking. We call it the multiple back-tracking algorithm (MBTA). When carrying out the ordinary shortest path extraction using dynamic programming, we have in storage the cost value for each node and the corresponding predecessor matrix. From each node on the last column, we can back-track a path from this node to a certain node on the first column. This path has a certain cost. If the starting and the ending positions of this path are neighbours, then we say this path is a possible CSP. We back-track all the nodes on the last column, and we may find several possible CSPs. We can then choose the CSP with the minimum cost as the final result. We have found that on an image or regular grid the MBTA algorithm guarantees to find a circular path although this circular path may not be the circular shortest path.

The steps for our MBTA algorithm are:

1. Carry out ordinary dynamic programming to build the cost matrix, and the predecessor matrix.
2. Carry out back-tracking from each node on the last column and record the cost for a circular path.
3. Choose a circular path with the minimum cost as the result of this algorithm.

3.4. Combination Algorithm

The IPA algorithm provides a fast way of finding "circular" shortest paths. But the path obtained is not always circular. The MBTA algorithm guarantees to find a circular path, although this path may not be the circular shortest path. We can combine these two algorithms as the IP&MBTA algorithm to increase the chance of finding the true circular shortest path in an image or grid. This will involve running each of the IPA and MBTA algorithms once. That is using the IPA algorithm to find a path, if this path is not circular, we use the path obtained by the MBTA algorithm. If the path obtained from running the IPA algorithm is circular, we choose the path with the minimum cost from the IPA and the MBTA algorithms. Many real images tests have shown that the combination algorithm produces all the correct circular shortest path.

3.5. Approximate Algorithm

In most real cases it is enough to heuristically find a circular path, not necessarily optimum, to correctly process the image. The key point is to guarantee that the best circular path found by the heuristic is not far, in terms of "cost", from the optimal circular path. To ensure that the "sub-optimal path" is "good enough" we would limit the relative error.

More formally, let z^* (> 0) be the (unknown) cost of the optimal circular path and let $z(A)$ the cost of the best path $P(A)$ found by a given approximate algorithm A. Of course $z(A) \geq z^*$. We say that the *relative error* $E(A)$ of path $P(A)$ is: $E(A) = \frac{z(A)-z^*}{z^*}$. To define a bound on the relative error it is enough to find a so-called "lower bound" of the unknown optimal solution, i.e. a not necessarily feasible solution whose cost z' is such that $z' \leq z^*$. If that solution is a feasible circular path, then it is an optimal solution for the problem. By knowing $z(A)$ and z' we have a "threshold value": $B(A) = \frac{z(A)-z'}{z'}$, which bounds the relative error of path $P(A)$, i.e. $E(A) \leq B(A)$.

To find a lower bound for the circular path problem, it is enough to solve the shortest path problem from any node of the first column to any node of the last column. That path may not be circular, and in this case it cannot be taken as a solution; nevertheless, its cost z' is a lower bound for the optimal solution.

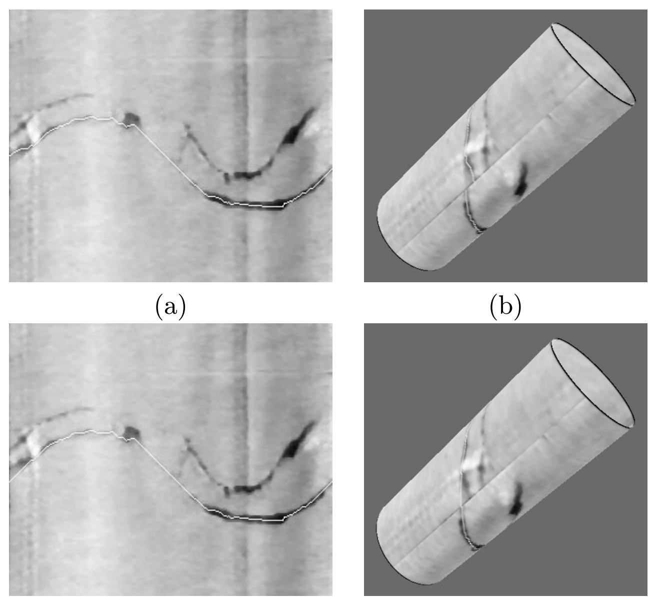
The approximate algorithm can be viewed as a special case of MSA or MBTA when the search for the circular shortest path can stop early. The predefined approximation accuracy may not always achieved even all nodes in the last column are checked in the Approximate Algorithm. This is particular true when the gap between the cost of the shortest path and the cost of the circular shortest path is large. For detailed description of all these algorithms, please see [10].

4. Experiment Results

This section shows some of the results obtained using the methods described in this paper. A variety of images have been tested, including synthetic images and different types of real images.

4.1. Borehole Data

To compare the different shapes of shortest path obtained using the ordinary and the circular shortest path algorithms, ordinary dynamic programming algorithm is applied to the borehole image shown in Fig. 3(a). The ordinary shortest path obtained is shown in Fig. 3(a). The result of our circular shortest path obtained by using the combination algorithm is given in Fig. 3(c). Notice the position difference of the shortest paths close to the left edges of Fig. 3(a) and (c). Fig. 3(b,d) show the 360° version of the flat 2D images. It is clear that the path obtained using the circular shortest path method has the same starting and ending position as shown in Fig. 3(d) while the path obtained using the ordinary shortest path extraction technique does not join up together at the starting and the ending positions as shown in Fig. 3(b).



(a) (b) (c) (d)

Fig. 3: *(a) A borehole image with a crack; The white path is obtained using an ordinary shortest path extraction algorithm without circular constraint. (b) Image in (a) shown in a cylindrical format. The starting and ending positions do not meet. (c) The white path is obtained using the algorithm developed in this paper which has the constraint that the path is circular. (d) Image in (c) shown in a cylindrical format. The starting and the ending points meet each other.*

4.2. Boundary Detection

du Buf *et al* described their first results on diatom contour extraction in [3]. In a preprocessing step initial contours are extracted using a conventional edge-following algorithm like Canny's. The object contours are extracted by using the best-fitting ellipse and a subsequent contour following in the elliptical polar-transformed image. They applied a depth-first search algorithm which evaluates the grey level changes along the path in the polar-transformed image.

Similar to du Buf *et al*'s algorithm, we obtain some initial positional information about a closed contour. In our case, however, we only need to know the approximate position of the contour. Then the input image is transformed into the polar coordinate system. Our circular shortest path algorithm is applied to this transformed image and a circular shortest path is extracted. The starting and ending positions of this obtained contour are neighbouring points. If we transform this obtained path from polar coordinate to the original Cartesian coordinate, a closed contour can be guaranteed. Fig. 4 shows two examples of finding the boundaries of an object. Fig. 4(left) are the input images. Fig. 4(middle) are the circular shortest path obtained in the polar coordinates. Fig. 4(right) show the closed object contours.

4.3. Panoramic Stereo Matching

Correlation based methods are very common for stereo matching. Usually a correlation matrix is obtained for each horizontal pair of scanlines from the left and the right stereo images, and a shortest path is obtained in this correlation matrix for disparity estimation. When performing panoramic stereo image matching, it is necessary to take the constraint that the left and the right columns of the stereo images are actually neighbours into account. We can use one of our circular shortest path extraction algorithms to obtain a CSP in a correlation matrix from panoramic stereo images. This obtained CSP will ensure that the starting and the ending position of the path are connected. The first two images in Fig. 5 are the left and the right panoramic stereo images; and the third image in Fig. 5 is the disparity map obtained using the IP&MBTA algorithm for circular shortest path extraction.

4.4. Running Times

Table 1 shows the computation time of different algorithms for obtaining circular shortest path on different images. The computer used was a rather slow 85MHz Sun SPARC. All the algorithms except the MSA are very fast and takes in the order of 0.3 seconds on 256×256 images. The timing was obtained by running the algorithms on random images several hundreds of times and taking the average.

5. Conclusions

We have developed several new algorithms for obtaining a circular shortest path on a regular grid. These algorithms have applications for borehole image analysis, object boundary detection, and panoramic stereo matching. The circular shortest path obtained in the image ensures that the starting and ending positions are connected. The five algorithms we developed are the multiple search algorithm, the image patching algorithm, the multiple back-tracking algorithm, the image patching and multiple back-tracking combined algorithm, and the approximate algorithm. The image patching algorithm is very fast although a solution is not guaranteed. The MBTA is also very fast and it is guaranteed to find a circular path, but may not be the optimal one. The combination of image patching and the multiple back-tracking algorithms gives a much higher chance and speed for finding the optimum circular shortest path. The typical running time for the image patching algorithm on a 256×256 image is in the order of 0.267 seconds on a rather slow 85MHz Sun SPARC computer. The MBTA algorithm takes about 0.227 seconds. The combination of image patching and multiple back-tracking algorithm takes about 0.312 seconds. The algorithm was shown to be fast and reliable by testing on several different types of real images.

Acknowledgement

Fig. 4(left) is from the ADIAC public data web page: `http://www.ualg.pt/adiac/pubdat/pubdat.html`. The first author thanks Moshe Sniedovich and Irina Dumitrescu for initial discussions. We are also grateful to Ernesto Martins[1], Mohan Krishnamoorthy, Michael Buckley and Houyuan Jiang for their suggestions and comments.

[1]Unfortunately, Ernesto Queiros Vieira Martins of Universidade de Coimbra (Portugal) suddenly died in November 2000, few months after his useful discussion with the authors; we dedicate this research to his memory.

Table 1: *Running times of different algorithms for circular shortest path extraction. Dynamic programming techniques are used as the ordinary shortest path extraction method on a 85MHz Sun SPARC machine.*

Image Size	Running Times (s)				
	MSA	IPA	MBTA	IP&MBTA	Approx
256×256	12.284s	0.267s	0.227s	0.312s	0.239s
512×512	389.440s	1.948s	0.989s	2.143s	0.962s

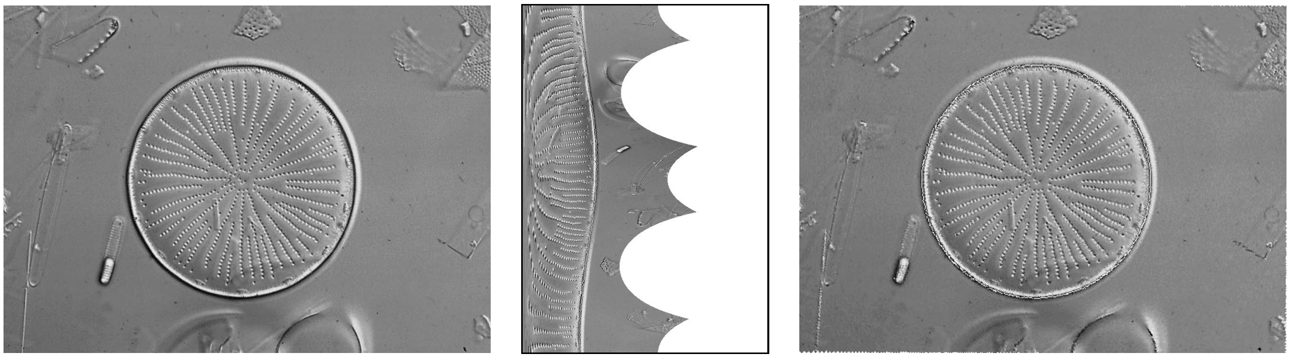


Fig. 4: *Results of boundary extraction using circular shortest path extraction. (left) Images with circular contour; (middle) Transformed image from Cartesian coordinate to polar coordinate. The white line shows the circular shortest path extracted using the patching method. (Image was rotated by 90 degrees). (right) The recovered image from the polar image. The white line is a closed contour.*

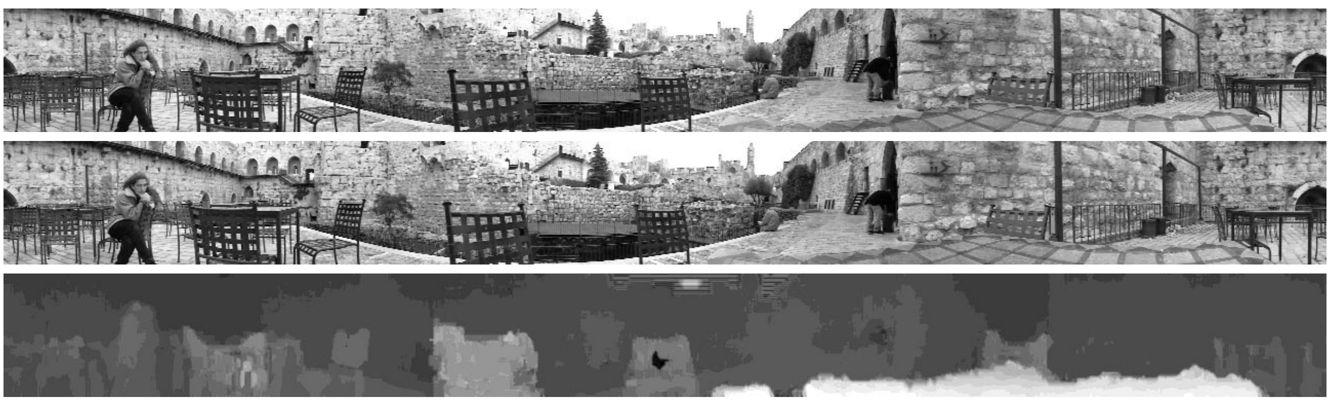


Fig. 5: *The top two images are the left and right input images. The third image gives the matching results using the method described in this paper. (Images courtesy Professor S. Peleg of The Hebrew University of Jerusalem).*

References

[1] M. Buckley and J. Yang. Regularised shortest-path extraction. *Pattern Recognition Letters*, 18(7):621–629, 1997.

[2] B. V. Cherkassky, A. V. Goldberg, and T. Radzik. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming*, 73:129–174, 1996.

[3] H. du Buf, M. Bayer, S. Droop, R. Head, S. Juggins, S. Fisher, H. Bunke, M. Wilkinson, J. Roerdink, J. Pech-Pacheco, G. Cristóbal, H. Shahbazkia, and A. Ciobanu. Diatom identification: a double challenge called ADIAC. In *Proc. 10th Int. Conf. on Image Analysis and Processing*, pages 734–739, Venice, Italy, September 27-29 1999.

[4] G. Gallo and S. Pallottino. Shortest path algorithms. *Annals of Operations Research*, 13:3–79, 1988.

[5] G. L. Gimel'farb, V. M. Krot, and M. V. Grigorenko. Experiments with symmetrized intensity-based dynamic programming algorithms for reconstructing digital terrain model. *International Journal of Imaging Systems and Technology*, 4:7–21, 1992.

[6] S. S. Intille and A. F. Bobick. Disparity-space images and large occlusion stereo. In *Proceedings of European Conference on Computer Vision*, Stockholm, Sweden, 1994.

[7] S. A. Lloyd. A dynamic programming algorithm for binocular stereo vision. *GEC Journal of Research*, 3(1):18–24, 1985.

[8] M. A. Lovell, G. Williamson, and P. K. H. (eds). *Borehole Imaging: applications and case histories*. Special Publications no. 159. Geological Society, London, 1999.

[9] C. Sun. A fast stereo matching method. In *Digital Image Computing: Techniques and Applications*, pages 95–100, Massey University, Auckland, New Zealand, December 10-12 1997.

[10] C. Sun and S. Pallottino. Circular shortest path on regular grids. Technical Report 01/76, CSIRO Mathematical and Information Sciences, May 2001.