Determination of gradient and curvature constrained optimal paths

(Revised Jan, July 2005)

Dr Michael J de Smith

Centre for Advanced Spatial Analysis

University College London (UCL)

mike@desmith.com

Correspondence address (postal):

Little Plat, German St, Winchelsea, East Sussex, TN36 4EN, UK

for submission to:

Computer-Aided Civil and Infrastructure Engineering

Abstract

This article provides an analysis of gradient and curvature constraints on path form and length, with particular reference to road, rail and pipeline route selection. Initially we examine the case of a single (global) gradient constraint and a planar surface, with or without boundaries and obstacles. This leads on to a consideration of surface representation using rectangular lattices and procedures for determining shortest gradient-constrained paths across such surfaces. Gradient-Constrained Distance Transforms (GCDTs) are introduced as a new procedure to enable such optimal paths to be computed, and examples are provided for a range of landform profiles and gradients. Horizontal and vertical curvature constraints are then analysed and incorporated into final solution paths as subsequent stages of the optimisation process. Such paths may then be used as pre-analysed input to detailed cost and engineering models in order to speed up, and where possible improve, the quality and cost-effectiveness of route selection.

1 Introduction

This paper analyses the impact of gradient and curvature constraints on optimal path form and length, with particular reference to selecting alignments for road, rail and pipeline routes. In this context an optimal path is assumed, initially, to be the shortest path (or paths) satisfying the constraints specified. As expected, the analysis confirms that the introduction of gradient constraints and additional engineering design standards often results in the need to consider other factors, notably those of direction change, boundaries, obstacles, horizontal and vertical curvature and cut-and-fill. Each of these factors may have implications for construction costs, but each also may impact usage and environmental costs.

In Section 2 we examine the simplest case, that of shortest paths across a tilted planar surface subject to a constraint on the maximum path gradient permitted. The inclusion of additional constraints involving boundaries, obstacles or no-go areas is also discussed. This leads on to a description of a 3-stage procedure for path selection: (i) initial route alignment subject to a pre-specified range of gradient constraints (Section 3); (ii) horizontal path smoothing to meet horizontal path curvature and smoothness objectives (Section 4); and (iii) vertical path smoothing to achieve similar objectives with a target of minimal cut/fill in the vertical plane (Section 5).

In Section 3 we describe the first stage, that of shortest route determination. Initially we describe the representation of physical landscapes using a regular lattice of points or cells for which height values are available. We show that when the acceptable path gradient in the directional of travel across a tilted plane is constrained to some prior maximum value shortest paths traverse the landscape lattice in a complex manner, zig-zagging to avoid steep uphill or downhill directions. We then demonstrate that the shortest gradient-constrained path(s) across general surfaces can be determined using a development of a class of lattice scanning

algorithms known as Distance Transforms (DTs). These transforms were originally designed for use in image processing applications. Within this field very fast DT algorithms have been developed - see for example: Borgefors, 1986; Leymarie & Levine, 1992; Breu et al, 1995; and most recently Lee et al, 2003, who provide an O(*ln*N) procedure, where N is the number of cells in the lattice. These sequential and parallel image processing algorithms may be enhanced in a similar manner to the extended sequential processing procedure we describe in this paper.

Section 4 then examines horizontal path curvature constraints in some detail and a model is proposed that incorporates the inclusion of curvature constraints as a second stage of the path alignment process using spline smoothing techniques. Finally, in Section 5, we provide an analysis of vertical path profiling using similar techniques to those we have applied for horizontal smoothing. Vertical smoothing is applied to the horizontally smoothed version of the shortest path. The resulting final path profile is designed to be as short as possible, smooth and continuous in both horizontal and vertical profiles, and satisfying pre-defined horizontal and vertical curvature constraints with a minimum of expected cut and fill. By amending the parameters of each of the three stages of path selection (distance minimisation under gradient constraints, horizontal curvature and smoothness, and vertical curvature and smoothness) a range of alternative paths and profiles may be obtained and compared in terms of construction and overall costs. Such procedures may be incorporated into modern Geographic Information Systems (GIS) and Civil Engineering route design packages, thereby providing a key input to the process of designing and costing new transportation routes and extensions and adjustments to existing networks.

The paper concludes with comments on the procedures and a number of Appendices that provide worked examples and details of the algorithms described. For a selection of references addressing some of the broader issues covered in this paper see: Goodchild, 1977;

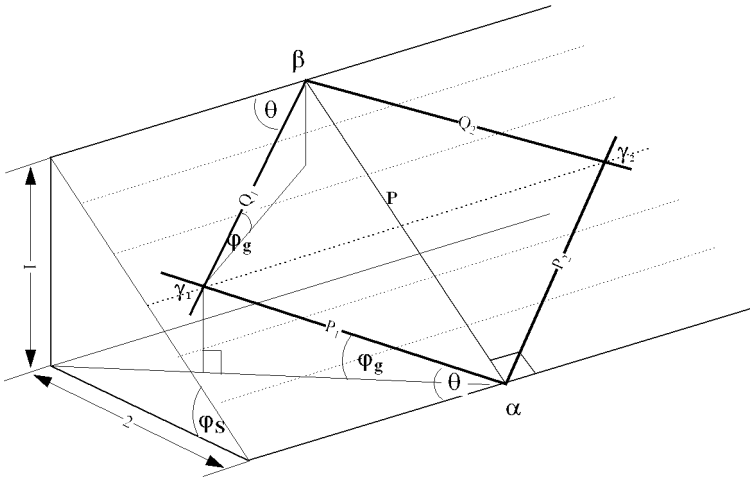
Trietsch, 1987; Rowe and Ross, 1990; Rowe, 1997; and de Smith, 2004. The latter paper describes how procedures based on the same core DT algorithm can readily incorporate obstacles and variable costs (e.g. land costs, environmental costs) across the study region.

2 Paths on planar surfaces

2.1 Shortest paths

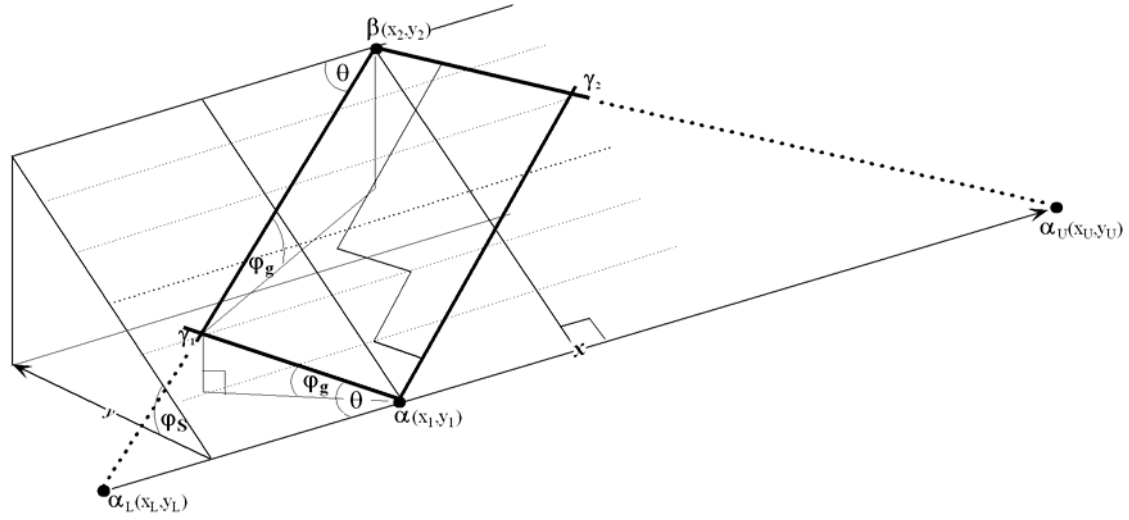
The shortest (unobstructed) path across a uniform horizontal or tilted plane is a Euclidean straight line. If the surface is tilted the path will have a non-zero path gradient with respect to the underlying horizontal plane (Figure 1, path P_1). Problems involving motion (e.g. walking, cycling, the use of powered vehicles) frequently require that paths traversed be subject to a maximum gradient constraint, φ_g. For example, road gradients rarely exceed 1:4 and design values of 1:10 (10%) to 1:25 (4%) are typical maxima used. Let us start by considering path crossing a tilted planar surface, S, with a constant slope, φ_S, of 1:2, i.e. just under 30 degrees (Figure 1). A path P between a sample point, α, and a target point, β, immediately above or below α on S will have a path gradient of 1:2. Every other straight line through α will have a lesser gradient, with the Normal to P at α having a zero gradient, i.e. defining the surface contours, in this case horizontal lines. Thus there is some path, P_1, making an angle θ with the Normal at α, to the 'left' of P, through α having a gradient of φ_g, such that $\varphi_g \leq \varphi_S$. If S is unbounded, a second path, P_2, will exist through α that also makes an angle of θ with the Normal, but to the 'right' of P. The same construction applies at β, and the two pairs of lines will intersect at points γ_1 and γ_2, as shown by the bold lines in Figure 1. The paths $[\alpha,\gamma_1,\beta]$ and $[\alpha,\gamma_2,\beta]$ across the surface determine the outer envelope of all shortest paths which satisfy the gradient constraint and *uniquely* determine the two paths which exhibit only one change in direction.

Figure 1 Gradient constrained paths on a sloping planar surface



Within this envelope there are an infinite number of paths of minimal length, d_g, but with a greater number of changes of direction, whose segments are parallel to those of the outer envelope. If the point α does not lie directly below or above β, but is offset, much of the preceding discussion still applies, but in this case the bounding envelope is a parallelogram in form (Figure 2). The darker line identifies the outer envelope of solution paths whilst the finer line within the outer envelope provides an example of a path of equivalent length but with additional changes of direction.

Figure 2 Solution path envelope



If we describe α and β in terms of their planar Cartesian coordinates, $\alpha(x_1,y_1)$ and $\beta(x_2,y_2)$, we see that the path length $d_g(\alpha,\beta)$ *is unaltered for all* α where $\alpha \in [\alpha_L,\alpha_U]$, and α_L and α_U are the upper and lower points defined by constant gradient lines with no direction changes through β to the contour at $y=y_1$. This analysis applies to more general surfaces if one regards these as approximated (locally) by planar segments. Insofar as solution algorithms utilise local neighbourhoods to construct solution paths, the assumption of local planar form is a reasonable first approximation.

2.2 Boundaries and obstacles

The preceding discussion assumes that the surface, S, is unbounded. If this is not the case, the points γ_1 and γ_2 may lie outside of the sample region. In this case the boundary will impose a corner condition on the path and a solution with a single change of direction will no longer be possible. Assuming that the boundaries are straight lines parallel to P, the solution paths with the minimum number of changes of direction will include path segments from the bounding envelope as far as the boundaries together with segments parallel to the bounding envelope.

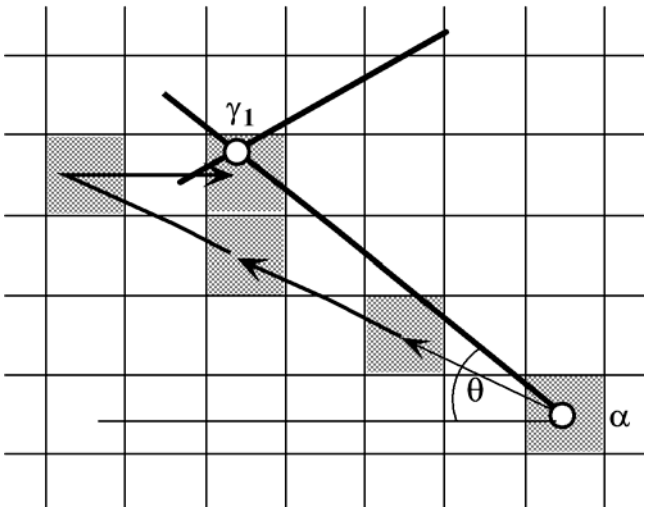
An outer boundary of the type described may be regarded as a form of obstacle. In a similar manner one or more obstacles may exist within the optimal bounding envelope. Unless an obstacle crosses the bounding envelope it will not affect the optimal path (least distance with fewest direction changes). However, if an obstacle does cross the outer envelope then solution paths will have to incorporate addition direction changes, and in some cases a set of obstacles will prevent any solution paths being found. Furthermore, in non-planar cases, obstacles or restricted regions (e.g. areas of water, housing, areas zoned for environmental protection, depths or heights not to be exceeded) within the solution space inevitably will result in optimal path realignment.

3 Lattice search and the use of Distance Transforms

3.1 Lattice representation

Real-world topography is often represented using a regular rectangular lattice of points or cells for which elevation values are provided. Datasets of this form are readily available from many national mapping agencies and are known as Digital Elevation Model or DEM data. If the tilted plane we have been examining is discretised and represented by a square lattice in the plane (i.e. as a DEM) then the representation of the surface may result in problems for path finding. Consider the gradient-constrained path $[\alpha,\gamma_1]$ in Figure 3. The complete set of lattice cells through which the optimum (bold) route passes would approximate the path with a step-like structure, clearly breaking the gradient constraint (locally), and their accumulated length will significantly overestimate the true path length. Acceptable solutions must omit intermediate cells in order to comply with the gradient constraint and to estimate path length more accurately (as in the shaded cells shown in Figure 3).

Figure 3 Lattice distortion of optimal route



Such solutions will always lie outside or on the optimal bounding envelope and thus may only reach the target point by initial, intermediate or final horizontal path adjustments, as illustrated. If path determination on a discrete lattice involves some form of localised search,

the size of the locality or neighbourhood will affect both the nature and feasibility of the solution. For example, with direct neighbour search there are only 8 available path directions on a square lattice. If we assume (as a worst case, and without loss of generality) that the lattice is aligned with the tilted plane there is effectively only one intermediate direction to try, $\pi/4$, and if the path gradient in this direction is too great, no solutions will be found. If the neighbourhood is extended to 5x5 cells (as in Figure 3) then 16 directions are available, or 3 distinct intermediate values for θ: $3\pi/8$, $\pi/4$, and $\pi/8$. With a 7x7 neighbourhood we have 32 directions and 5 distinct intermediate values. Such neighbourhoods are regularly used in lattice-processing algorithms and in graph-theoretic methods that remap the lattice as a fully connected graph. 3x3 and 5x5 neighbourhood sizes are the most commonly used values – in all such cases there will be a set of critical values for φ_g and φ_S for which solutions will fail. The general formula that relates φ_g and φ_S to θ, where θ is taken here as the angle shown in Figure 3, is:

$$\tan \varphi_g = \tan \varphi_S \cos \theta$$

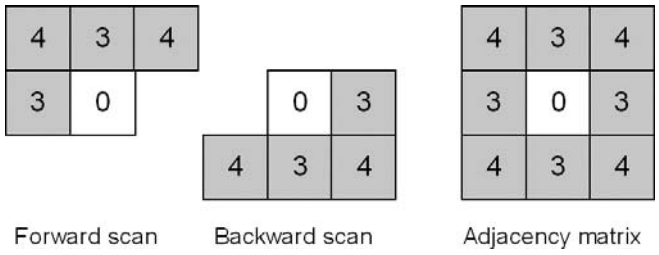
This expression allows us to analyse the size of search neighbourhood required for a variety of surfaces and gradient constraints. For example, with surfaces having gradient of up to 1:1 (100% or 45 degrees), a neighbourhood of 11x11 or greater is required in order to satisfy gradient constraints of 1:5 or 20%. With the earlier example cited where the tilted surface had a slope of 1:2, a 7x7 neighbourhood will suffice for a 1:5 constraint but a 13x13 neighbourhood is needed to satisfy a 1:10 constraint. Furthermore, when solutions are found by such methods they may generate solution matrices within which many alternate paths are feasible, and separate selection of paths that have the fewest changes of direction and/or that satisfy additional criteria will be necessary.

It also should be noted that increasing neighbourhood size: (i) hides (and smoothes) within-neighbourhood surface variation; (ii) will decrease estimated path lengths; and (iii) may increase the border region of the study area that remains excluded from the analysis. Increasing or decreasing the spatial (x,y) resolution of the lattice will not alter these results. However, resolution changes will affect the vertical (z) data as a result of interpolation or data collection procedures. Where such procedures involve non-linear changes they will impact path alignment.

3.2 Distance Transforms

Distance transform algorithms provide a very simple and extremely fast method for the approximation of Euclidean distances from every cell of a square lattice to the *nearest cell* in a target set. Distances over the lattice are calculated in an incremental manner based entirely on the distance to neighbouring cells. The standard algorithm involves a two-pass scan of a square or rectangular lattice dataset: a forward scan from top left to bottom right, and then a backwards scan from bottom right to top left. Each pass involves adding the values in a divided form of the adjacency matrix or *mask* to cell values in the underlying lattice - see the example masks in Figure 4, where 5 values are used in each mask section based on the (3,4) integer-valued mask.

Figure 4 3x3 integer masks for distance transformation



The value in mask position 0 of the transformed lattice is then set to the minimum of the sums calculated. The central function in the algorithm is of the form:

```
d0=min(d+LDM(k),d0)
```

where d0 is the current value at the central point (0) of the mask, LDM(k) is the local distance measure to the k^{th} element of the mask, and d0 is the current value at row r, column c of the lattice (this notation corresponds to the code sample provided in Appendix 3). The underlying lattice is normally a binary image, but could be a single target point (or set of points such as a line, e.g. a road) from which distances are automatically generated. In this case the target point(s) would be initialised to 0 and all other points as a large value (any value greater than the maximum possible distance to be computed, e.g. 9999). On completion of the two-pass scan each cell in the resulting lattice will contain the distance to the nearest point in the set of target points. In the example above, division of the values by 3 can be made on completion of the scanning process, giving an approximation that will be within 6.1% of the true Euclidean distance. With this integer-based model the distances to adjacent cells directly North, South, East and West (NSEW) of the central point are all exactly 1 unit away. With non-integer mask values the NSEW values are modified to be slightly less than 1 and the remaining directions to slightly less than $\sqrt{2}$. This alteration ensures that the maximum error in distance calculation across the entire lattice can be reduced to under 4% with a 3x3 mask and to under 1% with a 7x7 mask (see further, Appendix 2). Fast DT algorithms that compute Euclidean distance exactly now exist for both sequential and parallel processing environments.

In an earlier paper, de Smith, 2004, we have shown that Distance Transforms (DTs) can be extended by applying very minor modifications to the core algorithm to solve a wide range of spatial optimisation problems. For example, in order to incorporate continuously variable costs, e.g. land acquisition costs, environmental costs or some composite generalised cost field defined over the rows (r) and columns (c) of the lattice as COST(r,c), the central function can be modified as follows:

```
d0=min(d+LDM(k)*(COST(r,c)),d0)
```

These extended versions of the standard DT procedure have applications in many areas including location theory, traffic analysis, planning and decision-support. This paper takes this process further by incorporating gradient constraints into the standard DT algorithm, and then refining the initial solution paths that have been obtained. The refinement process uses horizontal and vertical smoothing techniques to ensure that path smoothness and minimum curvature objectives are also met.

3.3 Gradient Constrained Distance Transform solutions

Distance Transform (DT) procedures can be extended to incorporate gradient constraints by simply adding the constraint to the central function of the algorithm. We shall call such a procedure a Gradient Constrained DT or GCDT. The central function in this case is simply:

```
if ((d+LDM(k))<d0) & (|slope|< slopemax)
```

where *slopemax* is the constraint value and *slope* is taken as the magnitude of the path gradient.

Consider a uniform slope of 10%, similar to that shown in Figures 1 and 2 - data and trace vectors for this example are provided in Appendix 1, optimal parameters are provided in Appendix 2 and sample code is provided in Appendix 3. A 5x5 neighbourhood DT without gradient constraints will show a cumulative distance to a point 10 cells directly below or above the target point of 9.866 units in the horizontal plane, or 10 units if integer values are applied in the transform mask. However, with a 5% gradient constraint the planar distance is found to be at least 22.062 units using an optimal 5x5 mask. For the example path highlighted in Appendix 1, commencing slightly to the right of the target point, a total of 11 steps are required - these all consist of two steps across the slope and one step up or down the slope,

plus one initial horizontal adjustment. Since each of the longer steps has an optimal value of 2.2062 the total path length in this case is 10*2.2062+0.9866=23.0486 units. Note that such paths do not follow the direction of the maximum gradient of the distance transform. Furthermore, since the algorithm is executed in a specific sequence with the inequality shown above, vector tracking of paths will generate specific solutions reflecting this sequence, and therefore may not be entirely symmetric. Finally, if the gradient constraint is made more severe, e.g. 1%, then the only solution paths obtainable will be those that follow the surface contours. These comments and the distance values cited have ignored the additional surface length that applies due to the gradient of the test surfaces. This simplification is valid in this case because the error factor is a constant and close to 1 (being $1/\cos\varphi_g$). With DEM or equivalent models of real-world surfaces the use of gradient-corrected distances may be warranted, although the error in path length remains under 5% for relatively smooth landscapes with slopes of 1:3 or less and errors in the selection of local path direction will be rare for most surfaces. Surface distance adjustment may be applied after the planar procedure has been completed if required.

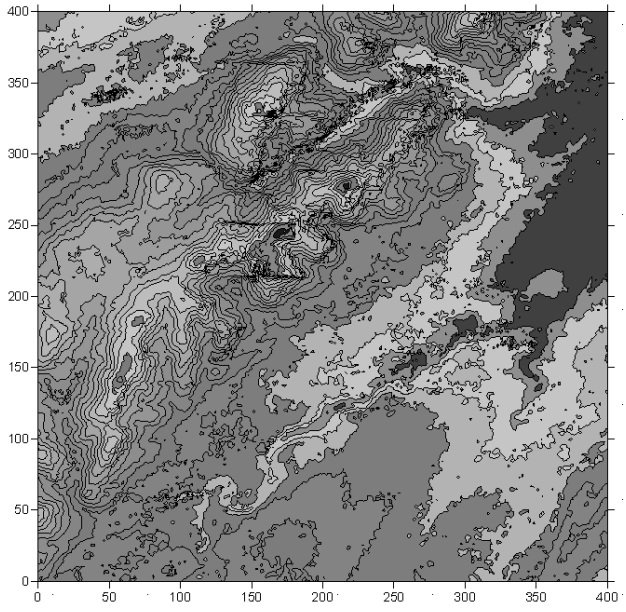
An algorithm that incorporates the information provided in Appendix 2, and/or systematic extension of the local neighbourhood by exploration of larger neighbourhoods until the gradient constraint is satisfied, will be more effective in finding solutions for all points in the sample region, but will still result in over-estimation of the path length and elements of path correction in order to reach the target cell. Finally, all such procedures assume (at least initially) that the intervening path is either smooth or must be graded by a cut-and-fill/levelling process if a road, rail or pipeline route is to be constructed.

For illustrative purposes, we now test the above GCDT procedures using a 100 sq km sample region covering part of the Pentland Hills area south of Edinburgh (Figure 5). Figure 5a shows a shaded contour map of the region derived from a 400x400 grid of UK Ordnance

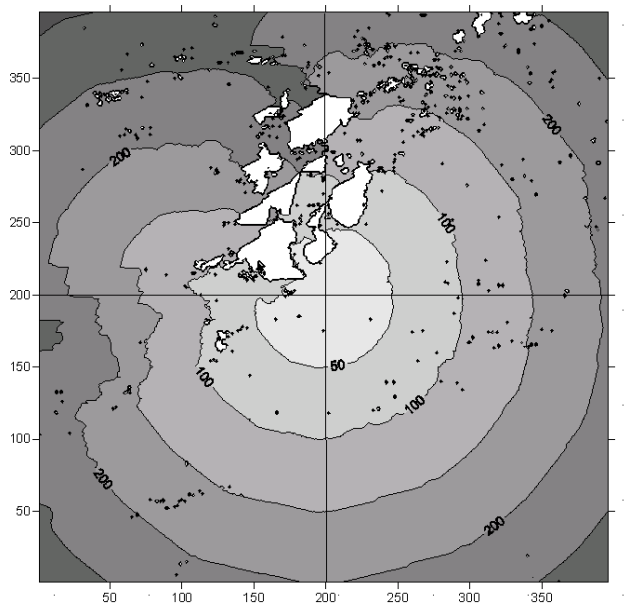
Survey Digital Elevation Model (DEM) data with each cell being 25m x 25m in size. Figure 5b shows a contour map of distances generated using the GCDT algorithm with the point (200,200) as the target (end point) and a constraint of 20% on the path slope. The contour lines show points that are equidistant from the target point (in grid units) along shortest paths that are subject to the gradient constraint applied. Areas shown in white (which includes the many small dot-like regions on the map) are not reachable using a simple 5x5 transform with this constraint – transforms with larger neighbourhoods will reduce the size and number of such regions.

Figure 5 Pentland Hills – Modelling gradient-constrained optimal paths

A. Contour map of study region



B. Distance transform, gradient<20%



The GCDT procedure may also be applied to the case where the target is one or more points, lines (e.g. existing roads) or areas, rather than a single point. Likewise, a sequence of GCDT procedures can be applied to cases where the route from A to B must incorporate predefined intermediate intersection points (IPs). However, where such IPs are relatively closely spaced,

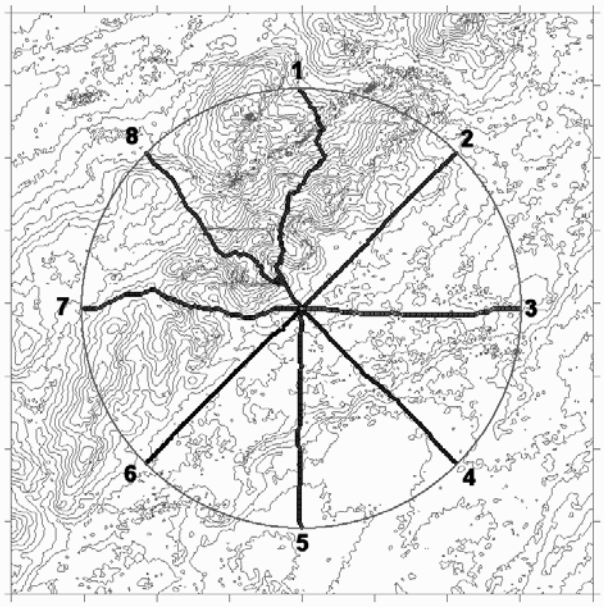
the large-scale path alignment may be predetermined and procedures which commence with curve fitting to such points followed by grading of the surface may be more effective.

Routes that are orthogonal to the distance contours shown determine optimal paths across the region. However, for a detailed picture of solution paths one must follow the individual trace vectors from selected points back to the target. This process is illustrated in Figure 6 using 8 test points (shown numbered in the diagram) on a circle of radius 3750m from the map centre. The average surface length of the 20% constrained paths is 4550m, compared with 5250m for the 10% constrained paths – the path highlighted in Figure 6c from Point 1 to the circle centre is roughly twice the crow flight surface distance.

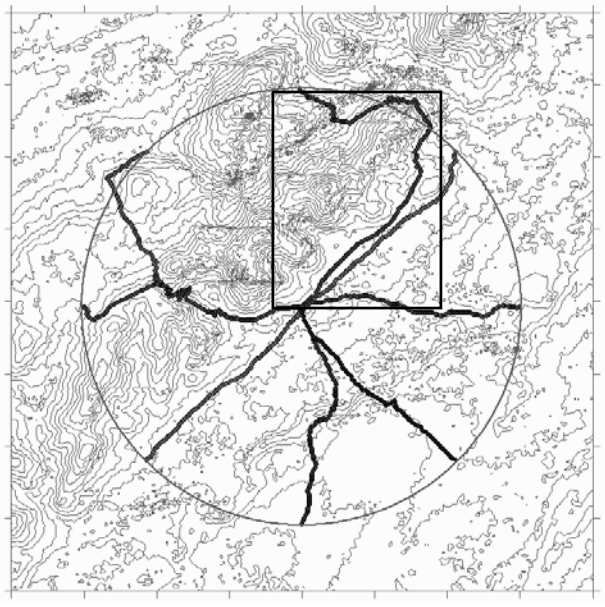
The maximum gradients for these paths are 19.8% and 8.9% respectively, these specific figures being a result of the use of a 5x5 transform and a DEM with 1m vertical resolution and 25m horizontal resolution. The solution path highlighted in Figure 6c exploits a narrow valley in the hills which contains a small reservoir, and in a real-world road or rail modelling exercise the areas of water and any other restricted areas would be excluded within the DT procedure, as described in de Smith, 2004. The route immediately to the right of the highlighted route follows, almost precisely, the current route of the A702 trunk road, which itself follows the line of a former Roman road to Edinburgh.

Least distance paths that meet gradient constraints provide a good first estimate of optimal (least cost) path alignment, since construction costs, maintenance costs and usage costs are closely linked to path length. However, environmental and other costs and factors may require consideration of a range of alternative ‘optimal’ paths.

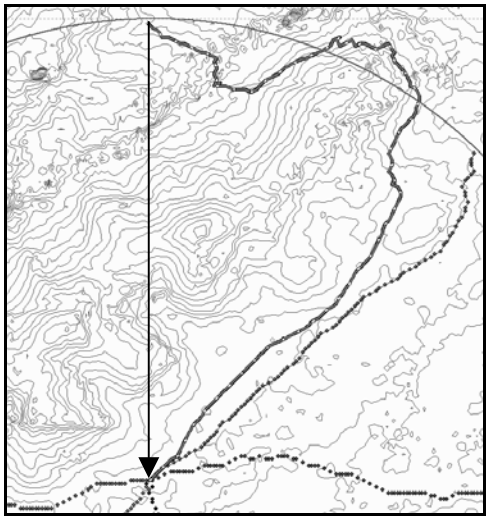
Figure 6 Optimal gradient constrained paths: 20% and 10% constraints



A. 20% max path gradient, 8 sample points



B. 10% max path gradient, 8 sample points



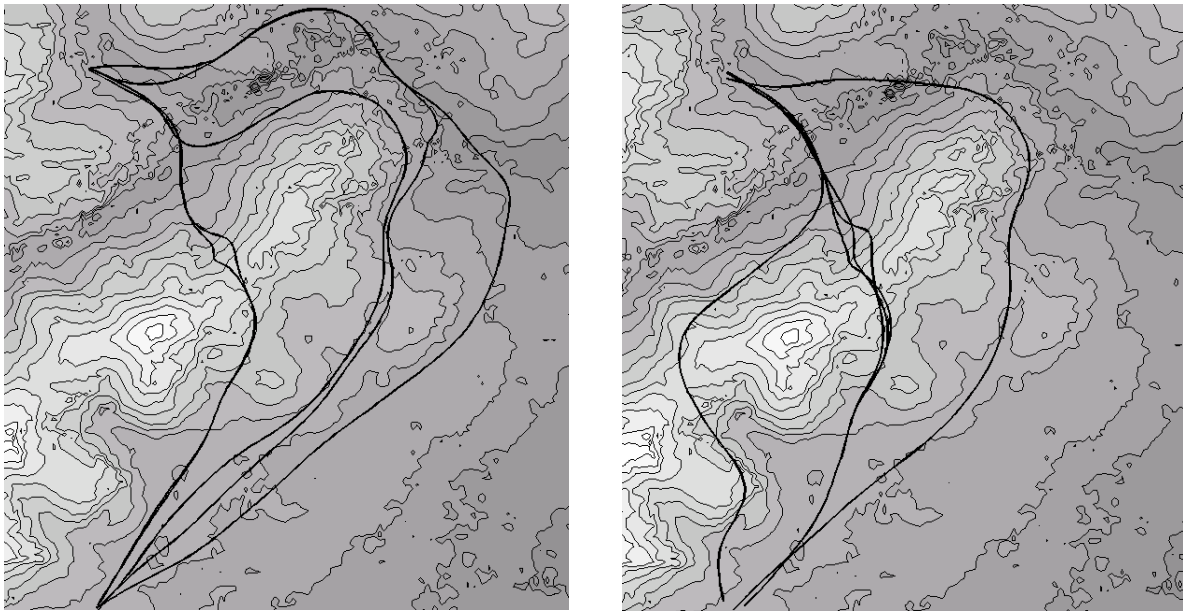
C. 10% max path gradient, linear point connection

(enlargement of area marked in diagram B)

By modifying the maximum path gradient constraint over a range of values above and below the target, a series of solution paths will be determined for every reachable point in the study region. This is illustrated in Figure 7 (for the same region as shown in Figure 6c), where the smoothed routes shown are optimised for path gradients of magnitudes 1:12, 1:11, 1:10, 1:9 and 1:8. The latter two paths in Figure 7a enable a much more direct route to be considered, illustrated here as the leftmost pair of routes. These are some 20% shorter than the originally

selected path (see Figure 6c), but if the gradients for these routes were to be set at 1:10 or 1:12 say, there would be a considerable amount of cut required in the central section of the route.

Figure 7 Optimal paths with varying target gradients



A. 5x5 GCDT, 1:8 to 1:12 gradients

B. 9x9 GCDT, 1:8 to 1:12 gradients

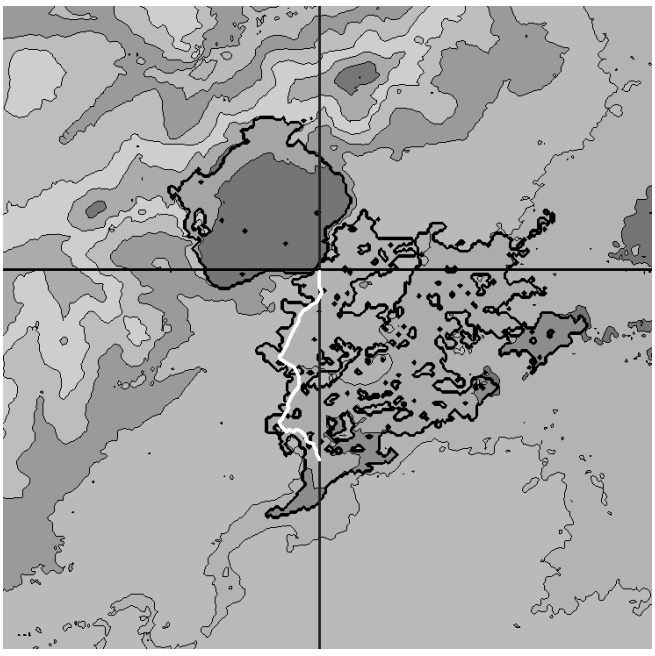
Figure 7b shows the result of the same operation but implemented using a 9x9 distance transform, which tends to smooth out local DEM variations owing to the longer step lengths permitted (up to 100m+). The 9x9 transform requires just over twice as much processing time as the 5x5 transform and generates paths that may require higher construction costs (greater cut/fill). However, it does pick out other potential routes (e.g. the leftmost of those shown in Figure 7b), paths tends to be smoother, and there are almost no regions in the study area which are unreachable (unlike Figure 5b).

Similar procedures to those described may be applied to terrestrial and subsea pipeline, powerline and cable route selection. In the case of subsea projects availability of fine resolution data may be problematic – much of the published data is at a spatial (x,y) resolution of 500m or greater, and thus of limited value for detailed route modelling purposes.

The importance of accurate data is frequently even greater for such problems owing to the limited scope for path profiling (cut/fill/smoothing) once a route has been selected. In such cases high resolution bathymetric and related subsea surveys will be required and routes selected that seek to minimise the risk of pipeline damage and ensure that any applicable curvature constraints are met, rather than meeting specific gradient constraints.

However, when considering fluid flow (e.g. drainage, sewerage, hydro-power, pumping) gradient is of great importance. In such cases the (signed value of) path gradient, φ_g, may be expressed as falling in a range of values, such as $\varphi_g \geq \delta$ or $\varphi_g \leq \delta$ (where often $\delta \approx 0$). If a single target point is taken (e.g. a water treatment plant, a reservoir) the GCDT procedure will identify the regions 'above' or 'below' the target, based on optimal pipeline flow routes. This process is illustrated in Figure 8, where natural drainage zones above and below the target point are shown bounded by a darker line; the landscape backdrop is shown as in previous figures.

Figure 8 Natural drainage zones to and from location (200,200)



The white line in Figure 8 shows a traced pipeline path flowing away from the target point. As before, adopting a larger neighbourhood for the GCDT will result in a reduction in 'unreachable' areas and a slight expansion of the solution regions due to smoothing. Increasing neighbourhood size in this instance is only appropriate if it can be justified in terms of the pipeline segment lengths and the pipe laying (grading) process.

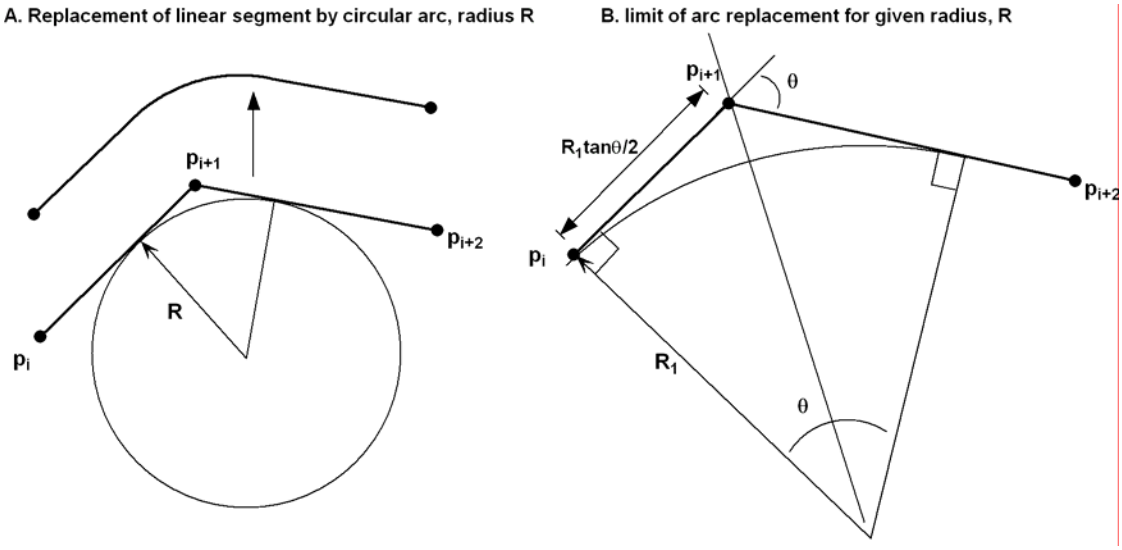
4 Horizontal path curvature

Sharp turns are not normally acceptable when designing paths for moving vehicles, robots or (in 3D) aircraft, spacecraft or missiles. Similar considerations may apply to some instances of terrestrial and subsea pipeline construction. In general there is a minimum radius of horizontal curvature that is specified for each vehicle type and speed, or pipeline type and diameter. For example, in Figure 6c we see that the optimal solution path includes relatively sudden changes of direction, which would be unacceptable for road or rail construction. Furthermore, in some instances paths are required to have non-zero curvature, for example in order to maintain driver attention on long stretches of highway. In other cases, such as solid pipeline construction, there may be little or no scope for curvature of the segments or joints, but angled connectors may be permitted.

When the surface crossed is well within the design gradient constraints the inclusion of horizontal curvature constraints does not provide special problems, unless additional factors such as obstacles or intersection points have to be included in the design. However, when surface gradients are greater, solution paths with a single direction change that do not permit sharp turns (discontinuities) will be found to lie outside of the envelope described earlier (Figure 2). With real-world problems, such as those illustrated in Figure 6, paths may be horizontally smoothed using piecewise analytical forms such as circle arcs, smoothing splines or Bezier functions. For example, Jong et al, 2000 utilise linear segments and planar circular

arcs in order to obtain smooth horizontal alignments, and their paper provides the geometric specifications for this process (Figure 9a, below). However, if the triples of solution points $[p_i,p_{i+1},p_{i+2}]$ are positioned such that the radius, R_1, of a circular curve with tangents at or within the limits imposed by $[p_i,p_{i+2}]$ (Figure 9b) is less than the minimum design radius, R, alternative smoothing (realignment) procedures will be required. This limit occurs where the exterior path angle θ exceeds the critical angle θ_1 for the minimum radius or where the length of the shorter of the two linear segments exceeds $R\tan(\theta/2)$.

Figure 9 Circular arc smoothing

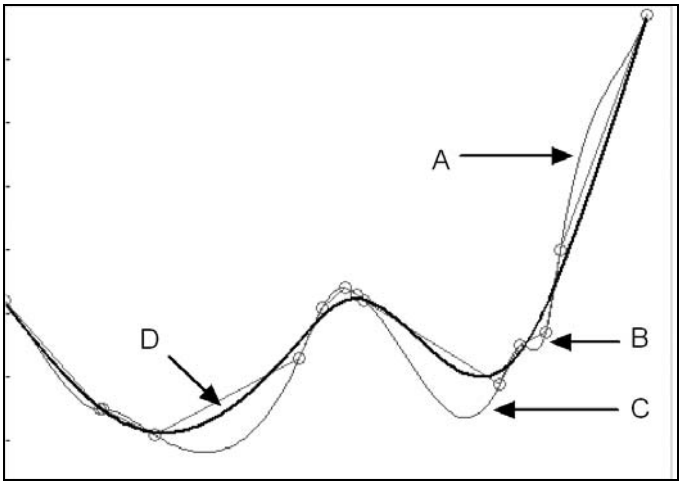


Using a 5x5 GCDT the maximum step length is 2.2062xscale, so with a 25m DEM this equates to 55m. With a maximum exterior angle, θ, in a sample path of say 30 degrees, this gives a figure of just over 200m as the maximum radius of curvature, which is below current UK guidelines for roads with a design speed of only 50kph (Department of Transport, 1993) unless the curves are superelevated (graded across their cross-sectional profile by up to 7 degrees).

Figure 10 illustrates the process of horizontal realignment by use of a variety of spline functions. The small circles in this case are the set of initial solution points, $\{p_i\}$, which are shown connected by linear segments. The line A is the result of fitting a piecewise cubic

spline to the set $\{p_i\}$, such that the directions at the initial and final points match the initial linear path directions. Despite this constraint the path exhibits curvature that remains unacceptable (e.g. at point B) and excessive deviation from the optimal path (e.g. at point C). One very effective solution to such problems is to apply a smoothing spline (or series of linked splines), which meets the curvature constraints, passes through or very close to the start and end points, but no longer passes exactly through the intermediate points of the initial solution set (Figure 10, line D, shown in bold).

Figure 10 Horizontal realignment using spline functions



The smoothing spline is similar to a regression curve through the point set, but amended so that the 'roughness' of the original points is reduced by a user-defined parameter, whose value facilitates curve fitting ranging from an exact spline fit to linear regression. Paths of this type must be analysed using parameterisation by path length, *t*, since paths in the plane will generally include repeated values for *x* (easting) and/or *y* (northing) components. Repeated values of both components (i.e. a point on the path being reached twice) should not occur. The roughness parameter (or smoothing factor) may be dynamically or interactively altered until the desired minimum radius of curvature is met for the path under consideration.

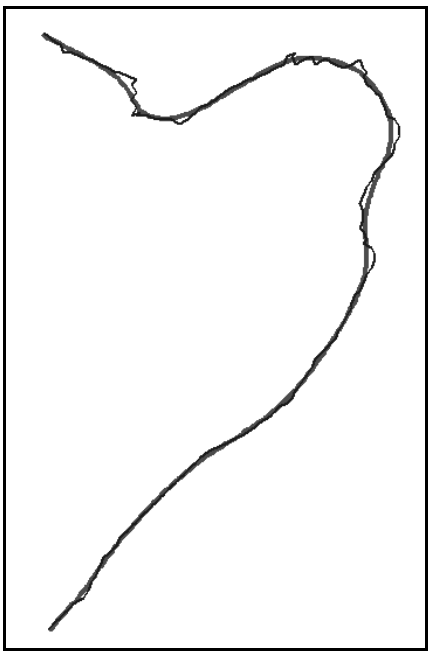
Path curvature, κ, at point *t*, is defined by the expression:

$$\kappa(t) = \left\| \gamma'(t) \times \gamma''(t) \right\| / \left\| \gamma'(t) \right\|^3$$

where $\gamma(t)$ is the parameterised space curve, $\gamma'(t)$ and $\gamma''(t)$ are the first and second differentials of the curve, × is the vector cross product and $||x||$ is the Euclidean Norm of x. The radius of curvature is simply $R=1/\kappa$. Sample computation of κ for use within MATLab is described in de Boor, 2003, p. 2-31.

For example, if we take the initial solution path shown in Figure 6c and seek a minimum radius of curvature of 160m, a solution path is found as shown in Figure 11. Here, the original solution path is shown as the finer black line and a smoothed spline fit to the set $\{p_i\}$ is shown in grey, where the smoothing factor equates to a minimum radius of curvature of 163m. The smoothed path is 10.4% shorter than the original path, and its proximity to the original horizontal alignment (mean deviation 0.7 grid units and a maximum of 3.2 grid units) makes it likely that the target gradient constraints will be retained over much of its route or may be obtained with limited additional path smoothing and grading. Since this new path is shorter than our original path, which we have already determined is the shortest path across the DEM surface, subject to the gradient constraint and neighbourhood size applied, the expectation is that this new path will break the gradient constraint.

Figure 11 Path smoothing using curvature constrained splines

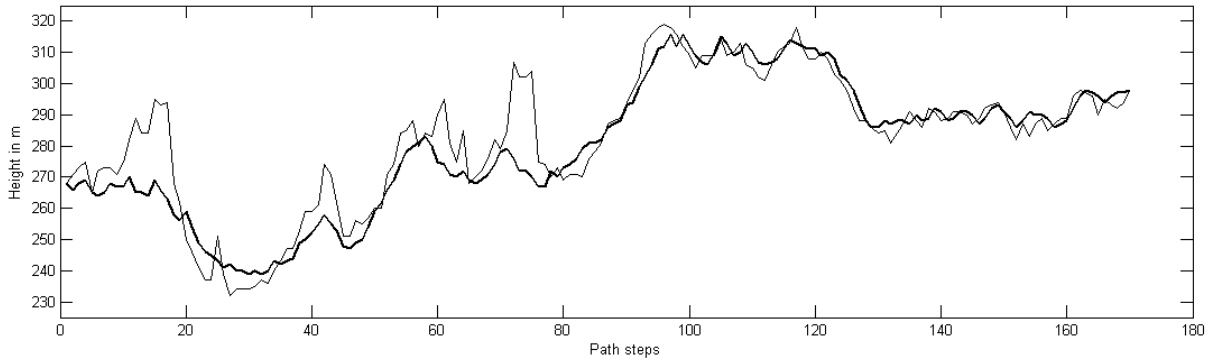


In the case of the planar model discussed earlier and in Appendix 1, fitting a smoothed spline to a sharply angular solution path may be problematic in some instances. The initial horizontal adjustment, which is a feature of the discretisation process in this example, can cause problems for the smoothing process – with appropriate smoothing parameter selection the resulting path avoids very small radius curves, although the selected path may not precisely pass through the start and finish points.

5 Vertical profiling and cut/fill operations

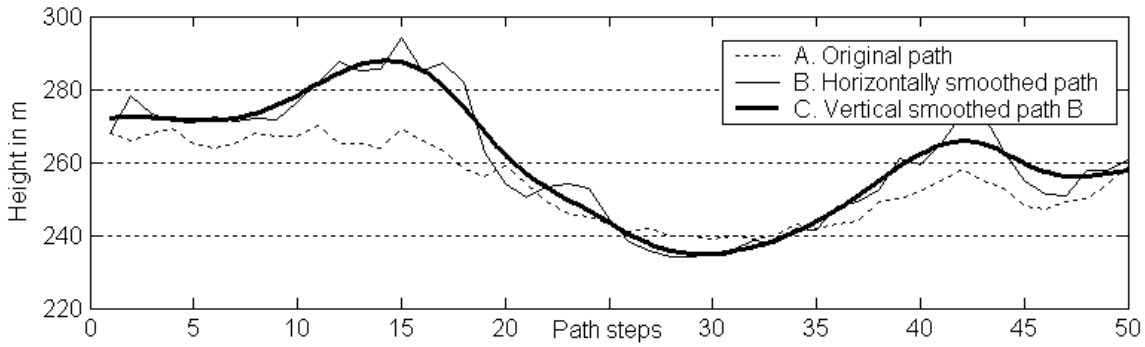
Despite the fact that the smoothed route in Figure 11 is clearly closely aligned to the original route, some parts of this revised route may diverge substantially from the design gradient constraint, especially where the original path is associated with steep side slopes. In such cases cutting, filling, grading or even bridging of the path route may be necessary in order to satisfy the gradient requirements. Figure 12 shows illustrative vertical profiles of the original and horizontally smoothed routes (horizontal scaling is much greater than that shown, so the diagram exaggerates heights substantially).

Figure 12 Vertical path profiles: Original path (heavy line); Horizontally smoothed path (fine line)



Both the original and new, horizontally smoothed paths exhibit vertical profiles which are not continuous or smooth. This is clearer if we examine the most disparate subsection of the solution path in Figure 12, covering the first 50 steps (see Figure 13 – each step is approximately 50m so the transect shown is approximately 2.5kms). The dotted line represents the vertical profile of the original, unsmoothed path, and the thinner solid line provides the vertical profile of the horizontally smoothed path, i.e. the path that seeks to minimise total path length whilst satisfying the horizontal smoothness and curvature constraints we have been given. This latter path breaks our path gradient constraint, as noted earlier, but as with the horizontal profile this path's vertical profile may also be spline smoothed (Figure 13, bold line).

Figure 13 Vertical path smoothing



Alternative vertical path curve fitting procedures may be applied in a similar manner – for example, using a combination of linear and symmetric parabolic elements as is commonly applied in road engineering.

This final process should be designed to ensure that the vertical profile: (a) is smooth and continuous; (b) satisfies the original gradient constraint (<10% in this example); and (c) satisfies further constraints that may apply on crest vertical curvature (e.g. >200m), sag curvature and line of sight in a similar manner to that applied to its horizontal profile. Note that this final solution path no longer remains entirely on the surface of the landscape, but involves a modest degree of cut and fill, with a maximum vertical cut or fill of a few metres in our example. An estimate of the total cut/fill required can be obtained by computing the cumulative positive and negative differences in height between the elevations of the horizontally smoothed path and its final smoothed vertical profile. This computation can be weighted by factors that vary along the selected path and affect construction costs, such as soil type, geology, water courses and other relevant elements. The resulting sum provides an initial indication (effectively an index or metric rather than an absolute measure) of additional construction costs by which alternative paths generated during the horizontal alignment stages may be compared.

However, detailed profiling and computation of cut-and-fill requirements (rather than just volume balancing) is not a simple optimisation problem for many reasons – amongst these are: the costs of cut and fill operations may be very different; there may be a preference for fill over cut, for example, where rock waste is readily available; the cost of picking up loads may be different for small and large loads; there may be insufficient materials or an over-supply over a single longitudinal and/or lateral stretch to achieve the target gradient; materials to be cut may be unsuitable for use in fill operations; the process of cutting vertically requires horizontal cutting and grading to ensure stability of banks and cross-sectional profile form;

and routing of transport between cut and fill locations may be very complex and difficult to optimise. For a recent discussion of such issues see, for example, Henderson et al, 2003.

The horizontally smoothed path is defined by non-integer planar coordinates, and hence surface heights along its length may be computed either by allocation to the DEM cell they are contained within or (preferably) by applying simple local surface interpolation with a neighbourhood matching the GCDT template size. For more complex surfaces and problems involving multiple paths and/or more sensitive criteria (e.g. 1:40 for gravity flow in waste water pipelines) it is advisable to generate a new, fine detail, grid using thin spline or Kriging techniques and assign height values from this generated surface (a fuller discussion of some of these issues may be found in Ehlschleger and Shortridge, 1996, and Zhang and Goodchild, 2002, pp.112 et seq). Generation of a complete, finer grid, using non-linear interpolation may also be desirable at the start of the analysis process, i.e. as an input to the GCDT procedure, in order to reduce the vertical step sizes inherent in DEMs with larger grid sizes. For example, in the Pentland Hills case a grid at 5m intervals could be generated which would result in a 2000x2000 matrix of values as input and a ‘generated’ vertical resolution of 0.2m. In all cases it is recommended that the DEM to be used is inspected together with any associated information (Metadata) that is available in order to determine its suitability for the task, and whether any input data errors, artifacts or interpolation procedures are likely to affect the procedures adopted.

Depending on the feasibility of constructing the original path based on the gradient and curvature criteria set, and on the results of analysing construction and other costs, it may become necessary to consider alternative (sub-optimal) path alignments, which have lower generalised costs. Thus optimisation of path length subject to gradient and curvature constraints does not, by itself, provide a complete solution to any specific alignment problem. Rather this procedure determines optimal or near optimal solutions to a subset of the overall

task, in a manner that meets the criteria set and provides full numerical information on the nature and quality of the solution.

6 Conclusions

The preceding analyses incorporate observations that facilitate the development of new algorithms, satisfying the requirements of least distance determination with gradient and curvature constraints, applicable to generalised surfaces. It has been shown that the family of algorithms known as Distance Transforms can readily incorporate gradient constraints, and are ideally suited for use in conjunction with modern GIS software. Horizontal and vertical curve fitting techniques may than be applied to ensure that additional design criteria are met whilst seeking to minimise construction and other costs. Although the analyses and procedures described in this paper are principally applied to terrestrial landscapes and transport routes, much of the discussion may be applied to terrestrial and submarine cable and pipeline routes. The approach proposed addresses the problem of identifying preliminary alignments that satisfy a number of broad requirements, in a very fast and efficient manner.

Initial tests have shown that iteration of the GCDT scanning algorithm results in very minor changes to the optimal paths selected, with small differences in the total surface path length. This suggests that some iteration may be desirable in order to obtain alternative solutions, which may have preferable profiles. Likewise, by modifying the maximum gradient constraint over a range of values above and below the target, a series of solution paths will be determined for every reachable point in the study region.

Discretisation of the surface does not appear to have adversely affected the procedures adopted, even using a simple 5x5 neighbourhood for analysis. DEM datasets with larger cell sizes and/or higher maximum errors in the height representation may provide less acceptable results. Subsequent analysis of the solution paths found may lead to the selection of

alternative paths whose distance-related costs are higher, but whose construction, operating or environmental costs may be lower. The incorporation of variable land costs (location costs), usage costs and environmental impact clearly increases the complexity of optimal path selection and techniques such as the use of GIS tools and genetic algorithms may be applied to incorporate these additional factors (see further, Jong et. al., 2000; de Smith, 2004).

It is also clear from our analysis is that assigning a simple ‘friction’ value or ‘cost’ to sloping regions and then applying the widely used procedure known as Accumulated Cost Surface (ACS) construction is not equivalent to direct solution of such problems. This is immediately apparent for a constant tilted surface, since in this case friction costs will be constant and thus will have no bearing on the solution. Conventional ACS methods that incorporate higher costs for regions with steeper slopes simply provide a method of partially accounting for expected increases in construction costs across such regions. Distance transform procedures that incorporate explicit path-related gradient constraints (GCDT) and which make allowances for variable generalised cost surfaces in order to determine least cost routes (LCDT) provide a more appropriate approach to such problems. To this end we propose the implementation of gradient constrained least cost distance transforms with subsequent curvature smoothing as one of the most effective and direct approaches to a wide range of physical path alignment problems.

References

Borgefors G. (1986) Distance transformations in digital images, *Computer Vision, Graphics, Image Processing*, **34,** 344-371

Breu H., Gil J., Kirkpatrick D. & Werman M. (1995) Linear time Euclidean distance transform algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 529-533

Department of Transport (1993) *Design manual for roads and bridges, TD 9/93*, *Volume 6 Road Geometry*, HMSO

de Boor C. (2003) *Spline toolbox user guide*, 6th Ed., The Mathworks, Natick, Mass., USA

de Smith M.J. (2004) Distance transforms as a new tool in spatial analysis, urban planning and GIS, *Environment & Planning B*, **31**, 85-104

Ehlschlaeger C.R. & Shortridge A. (1996) Modelling elevation uncertainty in geographical analyses, *Proceedings of the International Symposium on Spatial Data Handling*, Delft, Netherlands, 9B, 15-9B.25

Eungcheol K., Jha M.K., Lovell D.J. & Schonfeld P. (2004) Intersection modelling for highway alignment optimization, *Computer-Aided Civil and Infrastructure Engineering*, **19**, 119-129

Goodchild M.F. (1977) An evaluation of lattice solutions to the corridor location problem, *Environment and Planning A*, **9**, 727-738

Henderson D., Vaughan D., Jacobson S.H., Wakefield R.R. & Sewell E.C. (2003) Analyzing the cut and fill problem using local search algorithms, *European Journal of Operational Research*, **145**, 72-84

Jong J-C., Jha M.K. & Schonfield P. (2000) Preliminary highway design with genetic algorithms and geographic information systems, *Computer-Aided Civil and Infrastructure Engineering*, **15**, 261-271

Lee Y-H., Horng S-J. & Seitzer J. (2003) Parallel computation of the Euclidean distance transform on a three-dimensional image array, *IEEE Transactions on Parallel and Distributed Systems*, **14**, 203-212

Leymarie F. & Levine M.D. (1992) A note on fast raster scan distance propagation on the discrete rectangular lattice, *Computer Vision, Graphics, Image Processing*, **55**, 84-94

Rowe N.C. (1997) Obtaining optimal mobile-robot paths with non-smooth anisotropic cost functions using qualitative-state reasoning, *Internat. J. Robot. Res.*, **16**, 375-399

Rowe N.C. & Ross R.S. (1990) Optimal grid-free path planning across arbitrarily contoured terrain with anisotropic friction and gravity effects, *IEEE Transactions on Robotics and Automation*, **6**, 540-553

Trietsch D. (1987) A family of methods for preliminary highway alignment, *Transportation Science*, **21**, 17-25

Zhang J. & Goodchild M.F. (2002) *Uncertainty in geographic information*, Taylor and Francis, London

Appendix 1. 5x5 Distance Transform with Gradient Constraint

Gradient constraint φ_g = 5%, Slope φ_S =10%

A. Distance transform — Start | End | → Solution path

17.417	16.43	15.443	16.43	17.417	16.43	15.443	16.43	17.417	16.43	15.443	16.43	17.417	16.43
13.237	14.224	15.21	14.224	13.237	14.224	15.21	14.224	13.237	14.224	15.21	14.224	13.237	14.224
13.004	12.018	11.031	12.018	13.004	12.018	11.031	12.018	13.004	12.018	11.031	12.018	13.004	12.018
8.8248	9.8114	10.798	9.8114	8.8248	9.8114	10.798	9.8114	8.8248	9.8114	10.798	9.8114	8.8248	9.8114
8.5918	7.6052	6.6186	7.6052	8.5918	7.6052	6.6186	7.6052	8.5918	7.6052	6.6186	7.6052	8.5918	7.6052
8.3588	7.3722	6.3856	5.399	4.4124	5.399	6.3856	5.399	4.4124	5.399	6.3856	5.399	4.4124	5.399
8.1258	7.1392	6.1526	5.166	4.1794	3.1928	2.2062	3.1928	4.1794	3.1928	2.2062	3.1928	4.1794	5.166
7.8928	6.9062	5.9196	4.933	3.9464	2.9598	1.9732	0.9866	0	0.9866	1.9732	2.9598	3.9464	4.933
8.1258	7.1392	6.1526	5.166	4.1794	3.1928	2.2062	3.1928	4.1794	3.1928	2.2062	3.1928	4.1794	5.166
8.3588	7.3722	6.3856	5.399	4.4124	5.399	6.3856	5.399	4.4124	5.399	6.3856	5.399	4.4124	5.399
8.5918	7.6052	6.6186	7.6052	8.5918	7.6052	6.6186	7.6052	8.5918	7.6052	6.6186	7.6052	8.5918	7.6052
8.8248	9.8114	10.798	9.8114	8.8248	9.8114	10.798	9.8114	8.8248	9.8114	10.798	9.8114	8.8248	9.8114
13.004	12.018	11.031	12.018	13.004	12.018	11.031	12.018	13.004	12.018	11.031	12.018	13.004	12.018
13.237	14.224	15.21	14.224	13.237	14.224	15.21	14.224	13.237	14.224	15.21	14.224	13.237	14.224
17.417	16.43	15.443	16.43	17.417	16.43	15.443	16.43	17.417	16.43	15.443	16.43	17.417	16.43
17.65	18.636	19.623	18.636	17.65	18.636	19.623	18.636	17.65	18.636	19.623	18.636	17.65	18.636
21.829	20.842	19.856	20.842	21.829	20.842	19.856	20.842	21.829	20.842	19.856	20.842	21.829	20.842
22.062	23.049	24.035	23.049	22.062	23.049	24.035	23.049	*22.062*	23.049	24.035	23.049	22.062	23.049

B. Row vectors for tracking

0	0	1	1	0	0	1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0
1	0	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	0	0	0	1	1	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-1	-1	0	0	0	-1	-1	-1	0	-1	-1	-1	-1
-1	0	0	0	-1	-1	-1	0	-1	-1	-1	0	-1	-1
0	0	-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0
-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0	-1	-1
-1	0	-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0
-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0	-1	-1
-1	0	-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0
-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0	-1	-1
-1	0	-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0
-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0	-1	-1

C. Column vectors for tracking

1	1	-2	-2	1	1	-2	-2	1	1	-2	-2	1	1
-2	-2	1	1	-2	-2	1	1	-2	-2	1	1	-2	-2
1	1	-2	-2	1	1	-2	-2	1	1	-2	-2	1	1
2	2	1	1	-2	-2	1	1	-2	-2	1	1	-2	-2
1	1	2	2	1	1	-2	-2	1	1	-2	-2	1	1
2	1	1	1	2	2	1	1	-2	-2	1	1	-2	-2
1	1	2	1	1	1	2	2	1	1	-2	-2	-2	-2
1	1	1	1	1	1	1	1	0	-1	-1	-1	-1	-1
1	2	2	1	1	1	2	2	2	1	-2	-2	-2	-2
2	1	1	1	2	2	2	1	-2	-2	-2	1	-2	-2
1	1	2	2	2	1	-2	-2	-2	1	-2	-2	-2	1
2	2	2	1	-2	-2	-2	1	-2	-2	-2	1	-2	-2
2	1	-2	-2	-2	1	-2	-2	-2	1	-2	-2	-2	1
-2	-2	-2	1	-2	-2	-2	1	-2	-2	-2	1	-2	-2
-2	1	-2	-2	-2	1	-2	-2	-2	1	-2	-2	-2	1
-2	-2	-2	1	-2	-2	-2	1	-2	-2	-2	1	-2	-2
-2	1	-2	-2	-2	1	-2	-2	-2	1	-2	-2	-2	1
-2	-2	-2	1	-2	-2	-2	1	-2	-2	-2	1	-2	-2

Appendix 2. Distance Transforms: Optimal Local Distances

Neighbourhood	Max error %	Angular resolution °	Optimal local distances
3x3	3.96	45.00	0.96194, 1.36039
5x5	1.36	26.57	0.9866, √2, 2.2062
7x7	0.65	18.43	0.9935, √2, √5, 3.1419, √13
9x9	0.37	14.04	0.9963, Euclidean distances
11x11	0.24	11.31	0.9975, Euclidean distances
13x13	0.17	8.46	0.9983, Euclidean distances

Notes:

- For neighbourhoods of 9x9 and above we have suggested use of the planar Euclidean distance values except for the direct rook's case, since local Euclidean distances will approximate the optimal values to a sufficient degree of accuracy. If preferred, unit local distances may be used for rook's move neighbours with larger neighbourhoods, resulting in a small increase (under 0.5%) in max error% (distance estimation). Mean errors are well below these figures.
- The maximum error reduces very slowly as the neighbourhood size increases – for example, to only 0.12% for a 21x21 neighbourhood. The maximum error shown is the distance error when compared with the true Euclidean planar distance rather than angular error (or resolution) which is a feature of using a rectangular grid
- Angular resolution shows the minimum non-zero angle (in decimal degrees) in the plane resolvable using a grid with the neighbourhood size indicated

Appendix 3. Gradient Constrained Distance Transform: Sample MATLab code

In the simple code extract below, the forward scan of the DT algorithm is shown, based on a 5x5 Local Distance Metric, LDM(), and an input Digital Elevation Model (DEM) dataset stored in HT(,). The vectors XV(,) and YV(,) are used to store the incremental path movements of the optimised solution, whilst DT(,) stores the distance transform result set. DT(,) is initialised to 999 or 9999 for this problem with the target point, DT(200,200), initialised to 0. Scale is the DEM horizontal scale, which in this case is 25m. No vertical scale adjustment is required as HT(,) values are in metres.

```
% define mask values
a1=2.2062; a2=1.4141; a3=0.9866;
% forward scan
LDM=[a1 a1 a1 a2 a3 a2 a1 a3 0];
DX=[-2 -2 -1 -1 -1 -1 -1 0 0];
DY=[-1 1 -2 -1 0 1 2 -1 0];

for i = 3:xdim
for j = 3:ydim
   d0=DT(i,j);
   for k = 1:9
      r=i+DX(k); c=j+DY(k); d=DT(r,c);
      if LDM(k)>0
         slope=abs(HT(i,j)-HT(r,c))/(scale*LDM(k));
      else
         slope = 0;
      end %if
      if ((d+LDM(k))<d0) & (slope < slopemax)
         d0=d+LDM(k);
         XV(i,j)=DX(k);
         YV(i,j)=DY(k);
      end %if
   end % k
   DT(i,j)=d0;
end % j
end % i
```

this is followed by essentially identical code for the backwards scan, followed by data exporting for use within engineering modelling, mapping and/or related visualisation products.

For a 9x9 transform the LDM, DX and DY arrays will contain 25 elements. For a forward scan values can be used as follows:

```
% forward scan
% define mask values using local Euclidean metric
a1=1;a2=sqrt(2);a3=sqrt(5);a4=sqrt(10);a5=sqrt(17);a6=sqrt(13);a7=5;
LDM=[0 a1 a5 a4 a3 a2 a1 a2 a3 a4 a5 a6 a3 a3 a3 a6 a7 a6 a4 a4 a6 a7 a7 a5 a5 a7];
DX=[0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -2 -2 -2 -2 -3 -3 -3 -3 -3 -3 -4 -4 -4 -4];
DY=[0 -1 -4 -3 -2 -1 0 1 2 3 4 -3 -1 1 3 -4 -2 -1 1 2 4 -3 -1 1 3];
```

Smoothed spline curve fitting utilising the spline toolbox function csaps()

```
% apply smoothing spline (cv) to horizontal path profile, where t is the curve parameterisation,
% xy is a 2xN array of x,y coordinates and factor is a smoothing factor (e.g. 0.1 or 0.001)
% that may be varied dynamically to achieve a desired level of smoothing and minimum curvature
cv=csaps(t,xy,.factor);
```

List of figures in text: