EXPLORING MULTIPLE VIEWSHED ANALYSIS USING TERRAIN FEATURES AND OPTIMISATION TECHNIQUES

Young-Hoon Kim[1], Sanjay Rana[2], and Steve Wise[3]

[1] and [3] Sheffield Centre for Geographical Information and Spatial Analysis,

University of Sheffield, Sheffield,

E-mail: {y.h.kim, s.wise}@shef.ac.uk

[2] Centre for Advanced Spatial Analysis, University College London, London,

E-mail: s.rana@ucl.ac.uk

Abstract

The calculation of viewsheds is a routine operation in GIS and is used in a wide range of applications. Many of these involve the siting of features, such as radio masts, which are part of a network and yet the selection of sites is normally done separately for each feature. The selection of a series of locations which collectively maximise the visual coverage of an area is a combinatorial problem and as such cannot be directly solved except for trivial cases. In this paper, two strategies for tackling this problem are explored. The first is to restrict the search to key topographic points in the landscape such as peaks, pits and passes. The second is to use heuristics which have been applied to other maximal coverage spatial problems such as location-allocation. The results show that the use of these two strategies results in a reduction of the computing time necessary by two orders of magnitude, but at the cost of a loss of ten percent in the area viewed. Three different heuristics were used, of which Simulated

Annealing produced the best results. However the improvement over a much simpler fast-descent swap heuristic was very slight, but at the cost of greatly increased running times.

Key words: Viewshed, surface specific features, topography, optimisation, multiple viewpoints

1. Introduction

Predicting whether one point is visible from another (intervisibility analysis) and predicting the total area which is visible from a single point (viewshed analysis) are standard tools in Geographic Information Systems (GIS) software. Viewshed analysis has been used in a wide range of applications, including locating telecommunication relay towers (De Floriani et al., 1994), locating wind turbines (Kidner et al., 1999), protecting endangered species (Camp et al., 1997), analysing archaeological locations (Lake, M., et al., 1998), evaluating urban environment planning (Lake, I., et al., 1998) and optimal path route planning (Lee and Stucky, 1998).

The basic algorithm for generating a viewshed from raster elevation data is based on the estimation of the elevation difference of intermediate pixels between the viewpoint and target pixels. The determination as to whether the target pixels can be seen from the viewpoint is accomplished by examining each of the intermediate pixels between the two cells, to determine the 'line-of-sight'. If the land surface rises above the line-of-sight, the target is invisible. Otherwise, it is visible from the viewpoint. The line-of-sight computation is repeated for all target pixels from a set of viewpoints,

and the set of targets which are visible from the viewpoints form the viewshed. (Burrough and McDonnel, 1998)

Viewshed calculations are potentially time consuming, not least because of the large number of pixels which need to be considered when using a gridded DEM as the terrain model. Therefore a good deal of work has been done to develop efficient viewshed algorithms (Sorensen and Lanter, 1993; Wang et al., 1996; Fisher, 1991, 1993, 1996). Wang et al., (1996) developed a new fast viewshed calculation algorithm for DEM using neighbourhood grid cells. Recent developments have made use of visibility graph theory (O'Sullivan and Turner, 2001), statistical sampling (Franklin, 2000), reverse viewshed analysis (Kidner et al., 1999; Rallings, et al., 1999) and least-cost computation methods for determining optimal paths (Lee and Stucky, 1998).

In much viewshed work, there are a relatively small number of points of interest for which the calculations must be done. In applications which are assessing the visual intrusion of a development for example, this may be a single point (e.g. for a wind turbine), points along a line (e.g for a new road) or for points on the perimeter of an area (e.g.for a new housing development). In such cases the calculations can normally be done in real time.

Occasionally there is no single fixed point of interest and what is needed is a map of the variation in visibility across an entire area. This involves the calculation of the size of the viewshed for every pixel in the region to produce a visibility index. Each pixel is considered in turn as a target. Every other pixel in the image is considered in turn as

an observer, and an intervisibility analysis carried out to see whether the target is visible or not. If it is, 1 is added to the visibility score for the observer pixel. Once this has been repeated with every pixel as a potential target, the observer pixels will contain a count of how many targets they could see i.e. their overall visibility. If the DTM has n pixels and an intervisibility calculation is performed between every pixel and every other pixel, this is an $O(n^2)$ operation.

One approach to speeding up these calculations is to reduce the number of observers or targets or both (Rana, 2003). One way to do this is to perform the analysis on a TIN rather than a gridded DEM, because there will be far fewer points in a TIN than in a gridded DEM (De Floriani et al., 1994). The technique can also be applied to hierarchical TIN models to provide an efficient means of calculating viewsheds at varying resolutions (De Floriani and Magillo, 1997). This will only produce satisfactory results if the TIN is a good model of the terrain, since if key features such as prominent mountain peaks or ridges are not represented by TIN nodes, then the viewshed will contain errors. An alternative is to use only a subset of the pixels as observers for each target. Miller and Law (1996) used this approach in calculating a visibility map for the whole of Scotland. Visibility was calculated at 50m resolution, but points at 200m resolution were used as the observers. Rana (2003) took a similar approach, but instead of using a regularly spaced subset of points, used only points located on significant features of the landscape such as peaks, pits and passes and points on ridges and along valleys,. The computation times were reduced by 3 orders of magnitude, from nearly 8000 seconds down to 10 seconds. The absolute visibility index values will be underestimated when a reduced set of targets is used, but as Rana (2003) points out the absolute values are normally much less important than the

overall pattern, and particularly the identification of high visibility areas. The degree of match between the visibility index estimated using only surface specific features and the true values calculated using all the targets was generally good, with correlation coefficients ranging between 0.67 and 0.98.

The focus of the current paper is on a visibility problem in which the computational complexity is an order of magnitude higher even than the visibility index calculation. Visibility analysis is used in siting features which are part of a network, such as radio telephone masts (Goodchild and Lee, 1989; Lee, 1991; De Floriani et al., 1994). It is clearly important to locate each mast in a location which has high visibility. However, it is equally important to ensure that the masts in the network cover the terrain as efficiently as possible. This problem can be stated in two ways. Given n masts, how can they be placed to achieve the maximum coverage of the terrain. Alternatively, how many masts are needed to achieve at least n percentage coverage of the terrain. Both problems introduce a combinatorial element into the problem, because as well as the visibility of each potential location in the area, there is a need to consider the combined visibilities of all possible combinations of target locations. In this paper we will only be considering the first of these problems, which we will refer to as the optimal multiple viewpoints problem.

To analyse the time complexity of this problem let us consider the simple brute force algorithm. The two main steps are:

1. Calculate the intervisibility between all pairs of cells in the DEM i.e. undertake the first half of the visibility index calculation which as we saw

above is $O(n^2)$.

2. Select all possible combinations of v viewpoints from the n pixels, and for each combination, sum the number of visible pixels. The number of ways to select v items from n is $\dfrac{n!}{v!(n-v)!}$. When n is large and n >> v, this approximates to n^v. Summing the number of visible pixels is $O(n)$.

Of the three terms, n^v is by far the largest and hence the overall complexity is $O(n^v)$. So the multiple viewpoint problem has exponential complexity, compared with the quadratic complexity of the visibility index calculation (Wise, 2002). With a problem of even moderate size, this suggests that the simple brute force algorithm will never be computationally tractable and therefore strategies are needed to reduce the computation.

The first of these is to try and reduce the number of candidates to be considered as possible locations. Common sense suggests that some parts of the landscape are more likely to make good locations than others. Franklin and Ray (1994) showed that ridges and peaks tend to have higher visibility than other locations on average. In addition Lee (1992) demonstrated that the viewshed from many points in the landscape was often largely contained within the viewshed from a nearby ridge which means that only the ridge need be considered in the analysis. Rana (2003) has also shown that the visibility from a given point can be reasonably estimated by considering only points on key topographic features in the intervisibilty calculations.

By using a smaller number of candidate viewpoints, we will clearly reduce the number of computations we have to perform. For the example DEM used in our work, there were 1600 pixels, but only 136 of these are identified as peaks. If we are trying to position 10 viewpoints, we will only have to test 136^{10} (approximately 10^{21}) combinations if we use the peaks, compared with 1600^{10} (approximately 10^{32}) if we use all the pixels. This is clearly a saving in effort which is worth having. However the algorithm still has exponential complexity - every time we add another viewpoint, the size of the task goes up by an order of magnitude. This means it in addition to reducing the problem size, is worth exploring alternative algorithms which might offer better performance.

The optimal multiple viewpoint problem has many analogies with combinatorial problems which arise in location-allocation modelling. A classic location-allocation problem is the maximal coverage problem, in which a set of supply centres must be located such that they can satisfy demand from a number of points as efficiently as possible. This means that the network of supply centres must maximise the area which can be covered given the constraints of travels costs. There are some difference between this and viewshed analysis, but the underlying problem is very analogous, suggesting that some of the heuristics which have been developed in this field might be applicable.

The next two sections of the paper describe each of the two halves of the proposed methodology in more detail. This is then tested using a DEM for the same area in the Cairngorm mountains in Scotland as used by Rana (2003). The area, shown in Figure 1, is one 20x20 km tile from the Ordnance Survey PANORAMA DEM, which has

50m pixels. It is a relatively mountainous area, with a minimum elevation of 395m and a maximum of 1054. The original DEM had 401 pixels in both X and Y, which was too large for some of the programs used in the analysis. It was decided not to use a smaller area, since this gives too little variation in the terrain. Therefore the DEM was resampled to 500 metre resolution. This simplifies some of the detail of the terrain but the main pattern of ridges and valleys is maintained.

2 Use of terrain features

In seeking to locate features such as radio masts in an area, it clearly makes sense to consider points with high visibility as target locations. It is surprisingly difficult to identify which points in the landscape will have the highest visibility without calculating the visibility index for all pixels. Intuitively one might expect the highest points to have the best visibility, but this is often not the case. Franklin and Ray (1994) for example found a correlation coefficient of only 0.12 between visibility and elevation. In the Cairngorm test area the correlation was a little higher at 0.293 but it is clear that elevation is still a poor predictor of visibility. However a number of authors have observed that topographic position, rather than absolute elevation, might be a key factor. Franklin and Ray (1994), Lee (1994) and Rana (2003) all observed that ridges and peaks tend to have higher visibility on average than other parts of the landscape because of their relative elevation compared with other nearby points. This is by no means a simple pattern however. Some points on ridges can be obscured by land at higher elevation along the ridge, and a peak may well be surrounded by higher

peaks. As Rana (2003) observes, visibility is not simply a property of the point in question, but of its relationship with the rest of the landscape, and so it will always be impossible to predict visibility with a high degree of accuracy simply by considering local properties of the surface, such as elevation, or topographic feature type. However, if high visibility locations always tend to be found in certain topographic locations, such as on ridges, then identifying these locations could be a useful strategy.

In order to explore this the Landserf program [1] was used to classify every point on the surface into one of six morphometric types:

1. Pit
2. Valley
3. Pass
4. Ridge
5. Peak
6. Planar slope

The classification is based on modelling the form of the surface surrounding the central pixel using a polynomial trend surface. The coefficients of the polynomial equation are used to determine the slope and curvature of the surface. If the slope falls below a threshold, which can be set by the user, the pixel is deemed to be 'horizontal' and inspection of the curvature values decides whether it is a peak (convex curvature all round), a pit (concave curvature all round) or a pass (mix of convex and concave curvature). In the case of non-horizontal slopes, if the curvature is greater than a

second user-defined threshold, the point is classified as either a ridge (convex slope) or a valley (concave slope), but if it less then the point is classified as planar. Full details of the method and the theoretical background behind it can be found on Wood's website [1] from where a copy of Landserf may be downloaded.

This method of classifying the surface is much less sensitive to small errors in the DEM than other techniques (Wood, 1998). The least squares curve fitting procedure means that the polynomial surface is not forced to pass through every data point. In addition the surface can be fitted to a window of any size around the central pixel and a window larger than the 3x3 window often used by other techniques, means that the effects of small scale errors are reduced.

The 50m DEM of the Cairngorms was classified using Landserf. The slope and curvature parameters were left to their default values, but after some experimentation it was found that a window size of 9x9 produced a better looking classification than the default 3x3. The visibility index was calculated for every pixel and the result is shown in Figure 2. Visually it would appear that the higher ground of the ridges tends to have the higher visibility values. However, as Figure 3 shows, while the highest mean and absolute values are found on the peaks and ridges, the minimum values for all categories except peaks are very similar.

While not all peak and ridge pixels have high visibility, all points with high visibility do tend to be located on peaks and ridges as illustrated in Figure 4 which shows the location of all points with a visibility index which is two standard deviations above the mean.

However, we are not interested simply in the point or points with the highest individual visibility. We need to identify combinations of points which between them cover as much of the area as possible and there are two reasons why ridge points alone may not be good for this.

1. Neighbouring pixels along a ridge will be likely to have similar viewsheds. This will not always be true of course because small changes in position can produce large changes in the area which is visible. However, as Figure 4 shows, there is a tendency for visibility values to show positive spatial autocorrelation, with neighbouring pixels have similarly high or low values. To illustrate the degree of overlap, Figure 5 shows the viewshed for two neighbouring pixels shown by the white dots. Areas shaded light grey are only visible from one of these two, while the areas shaded in dark grey are visible from both. The white areas are visible from neither. In total there is an 89% agreement between the viewsheds of the two points.

2. There will be parts of the landscape which will be consistently invisible from ridge tops (Franklin and Ray 1994). Most erosive processes which sculpt mountain areas tend to produce hill slopes which have a convex profile near the ridge, and a concave profile lower down (Carson and Kirkby, 1972). This is particularly true of landscapes in which fluvial and mass movement processes are dominant. As a result, it is very often difficult to see the slope immediately below the ridge. This is illustrated in Figure 6, which shows the combined viewsheds of the ten most visible pixels. The areas which are not visible are all on the hillsides immediately below the viewpoints, or down in the valley.

The second point suggests that a combination of points on ridges and in valleys might be more likely to produce an optimal result than points from ridges alone. However in the Cairngorm DTM, taking all the points along ridges and valleys still resulted in a large number of potential candidates (1079). It was therefore decided to use the two extreme points of the ridge and valley network (peaks and pits respectively) and the points at which the ridge and valley networks intersect, the passes giving a total of 442 points altogether (Figure 7). It may seem counterintuitive to use pits as candidate viewpoints, but it must be remembered that these are not pits in the hydrological sense of pixels which are lower than all their neighbours. These pits have been identified by Landserf using a 9x9 pixel window, and represent the end points of the valley network. None of them are coincident with hydrological pits in the DEM, and the combined viewshed of all 53 actually covers 70.9% of the area.

3 Heuristics for the multiple viewpoint problem

One approach to designing a new algorithm for a problem is to look for ideas from an analogous problem. Since the multiple viewpoint problem aims to maximise visible areas, its objective function can be considered to be equivalent to the aim of obtaining maximum coverage in facility location planning. In facility location planning, there are a set of sources which supply material to a set of target locations. The aim is to allocate each target location to a source in an optimal way. A good example is the allocation of children to schools. If the aim is simply to minimise the travel time of the children, each child is allocated to the nearest school and this can be solved

algorithmically by generating isochrones from each school. However, if we wish to decide where to locate a new school in an area such that has the most benefit in terms of average travel times this simple algorithmic approach will not work, because we don't know the location from which to calculate the isochrones. In fact it now becomes a combinatorial problem for which heuristic solutions must be used.

In the optimal multiple viewpoint problem the function is to identify n positions (the viewpoints) in such a way that the percentage of the m pixels in the terrain model which are visible is maximised. This is equivalent to saying that we wish to position n facilities in such a way that the number of target locations connected to those facilities is maximised. Thus the viewpoints can be equated to the facilities in facility location planning which potentially means that heuristics developed for facility location planning may be applicable to the multiple viewpoint problem. There are some differences in the formulations of the problems however. The multiple viewpoint problem can be formulated as follows:

Objective function:

$$\text{Maximise } F(V) = \sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}$$

Subject to:

$$0 \leq \sum_{i=1}^{n} v_{ij} \leq n \quad \text{for } j = 1, 2, 3, ..., m$$

$$0 \le \frac{\sum_{j=1}^{m} v_{ij}}{} \le m \quad \text{for } i = 1, 2, 3, \dots n$$

$$0 \le \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}}{} \le (n \times m)$$

The objective function represents F ($v_{1,1}$; $v_{2,2}$; $v_{3,3}$; ... $v_{n,m}$) in which v_{ij} is 1 if viewpoint i can see the cell j and 0 if not, and it is the sum of this function over all combinations of viewpoints which must be maximised. The first constraint states that there may be grid cells which all viewpoints can see. In contrast, in the general procedure of a location-allocation model, each demand point is assigned to only one facility. The second constraint states that a viewpoint can potentially see all of the grid cells. As a result, the total number of visible grid cells may be equal to the number of the grid cell. The second and third constraints assume that in theory, all viewpoints or some of them can see the whole surface area with maximised visibility quality if located at the best site. These differences between the viewpoint site problem and the general facility location-allocation problem have an effect on the search procedures of the visibility optimisation algorithms because some grids can be visible and then assigned to all of the viewpoints in the 'allocation' sequence. Otherwise, like the general facility location-allocation procedure, some cells that are not visible are not assigned to the viewpoints in the 'location' sequence.

Although the visibility function is analogous with the form of facility location models, the problem nature has a different constraint relationship. This is illustrated in Figure 8 in which the viewshed problem is defined for 3 viewpoints on 5 grid cells, and that assumes if assigned to any viewpoint, 1 or 0 if not assigned on each grid pixel. As shown in Figure 8, there are some differences between the constraints. In the facility location problem, each demand has to be allocated to a facility but is only allocated to one facility. However, there may be DEM cells which are not visible from any viewpoint, as in the case of pixel 3. In addition it is possible for a DEM pixel to be visible from more than one viewpoint, as shown in the case of pixel 2. However these differences do not invalidate the use of spatial optimisation heuristics, because despite the differences in the constraints, the two problems are analogous because of the need to maximise a similar objective function - e.g. maximised customers or coverage, or minimised travel distances in location-allocation modelling.

Heuristic approaches are often employed to tackle combinatorial problems such as location allocation. They have numerous advantages for solving such spatial optimisation problems when compared with exact programming techniques including inexpensive computing cost, the flexibility of the objective function, and the identification of marginal sub-optimal solutions in large problems. These flexible characteristics allow for the extensive investigation of a wide range of alternative decision criteria to multiple viewshed analyses. However, the major drawback of heuristic techniques is that they are not guaranteed to generate optimal solutions compared to exact programming techniques. To overcome this weakness, many efforts have been made to improve their solution performance using more efficient search techniques or developing robust solution algorithms. For the former, Teitz and

Bart Vertex Substitution Heuristic (Teitz and Bart, 1968) has been proposed because of its ability to find optimal solutions with a high degree of regularity (Rosing, et al., 1979), and for the latter, recently, genetic algorithms and simulated annealing algorithms have demonstrated their superior search capability for converging on the best solutions with extensive facility location applications (e.g. Liu, et al., 1994; Houch et al., 1996; Murray and Church, 1995; Krzanowski, and Raper 2001).

Three different heuristics were chosen for these initial experiments. In each case the goal is to select n viewpoints from among the v candidates, such that their combined viewsheds cover the maximum percentage of the m pixels in the DEM.

Swap Algorithm. Given the objective function of the visibility problem, the substitution process of the Teitz and Bart algorithm can be used straightforwardly. For solving the viewshed optimisation, a starting solution is generated by selecting the first p candidates at random. The total visible areas are calculated for each candidate viewpoint. At each step, each candidate that is not in the current solution is substituted for each of the current solution viewpoints, and the combined viewshed recalculated. If any substitution shows improvement, the current viewpoint location is replaced with the candidate that achieves the best solution. If any improvement occurred in this step, new visible areas are defined. This iteration is continued until the last viewpoint is evaluated.

Spatial Genetic Algorithm. Genetic Algorithms apply principles derived from evolutionary biology in order to solve complex optimisation problems. Potential new solutions are generated by processes analogous to biological genetic operators. 'Good'

new solutions survive because they are better able to breed and replace each member of the old population by a newly bred individual. For solving the multiple viewshed problem, potential solutions to the problem are represented as 'genes' - strings of characters in which each character is a code representing a potential viewpoint. For this purpose the DEM cells are numbered from 1 to m. The gene is then a string of 4 byte integers containing the cell identifiers for the viewpoints and the length of the string is simply the number of viewpoints which need to be located. The GA randomly generates an initial population of genes. The fitness of each gene is then evaluated by measuring the total area covered by the viewsheds of viewpoints identified in the gene. In order to create a second generation of genes, a number of processes are applied to the current set

1. Asexual reproduction. The probability of a gene reproducing itself (i.e. surviving unchanged to the next generation) is linked to its fitness - this ensures that better solutions are likely to survive.

2. Crossover. This is name for the production of a new gene by combining material from two parent genes. Again the likelihood of a gene being involved in this type of sexual reproduction is linked to its fitness.

3. Mutation. This is the production of new genes by random modification of existing ones. It ensures that the genetic process does not get too locked in to a small number of combinations. Without it for example, any candidate viewpoint which was not part of one of the original pool of genes would never be considered.

Unlike the switch algorithm that uses a current solution, the GA uses a candidate pool among which good solutions are exploited by using the genetic operators. The search manner of the GA requires fewer search attempts than randomly trying variations, and

its evolutionary rules take fewer iterations than the brute force approach that inspects every solution rather than exploiting outstanding solutions.

Spatial Simulated Annealing Algorithm. The Simulated Annealing (SA) algorithm works in way which is analogous to the annealing process in material physics which is used to obtain a rigid metal structure. If hot metal is cooled too quickly, the crystals in the material do not have time to settle into optimal structures, and the metal will be brittle. Annealing is the process of letting the metal cool slowly, so that the crystals have time to lock themselves into rigid structures. The key element of SA heuristics is that in the initial stages, alternative solutions will be kept even if they are slightly worse than previous solutions, which avoids the danger of getting stuck in a local suboptimum. The decision on whether to accept a new solution or not is taken probabilistically using the Metropolis' criterion

$$p = e^{-\frac{\cos t}{temp}}$$

The probability of accepting a move, denoted by p, is related to the cost of the move (i.e. how much the objective function is worsened) and the temperature. By analogy with the annealing process, the system is initially considered to be at a high temperature, in which the probability of accepting a change will be high, even if the cost is high. As the temperature is slowly decreased, the changes become slowly less random until the system settles into a final, and hopefully optimal, configuration. Various facility location and spatial optimisation problems have been successfully tacked using SA heuristics (e.g. Liu et al., 1994; Murray and Church, 1995). For the

multiple viewpoint problem, a set of the potential viewpoints is selected at random to form the starting point. For each new solution, new viewpoint locations are chosen randomly from the neighbourhood of the viewpoints accepted in the previous iteration. For the annealing process, the SA heuristic reduces the temperature by fraction of acceptance moves and acceptance frequency of the Metropolis criterion at each step. The advantage of this method is that the run length of each temperature and cooling schedule (temperature decrease) are easily determined and well planned for the optimisation solution (see Liu et al. (1994) for more details).

The time complexity of these heuristics varies, and can also depend upon the exact nature of the problem. Rosing et al. (1979) reported that if the size of location-allocation problem is doubled, the computation of the swap algorithm would take over twice as many steps. He and Yao (2001, 2003) found that GA heuristics can display both polynomial and exponential time complexity behaviour in solving classical combination optimisation problems. According to their experiment, the number of generations, crossover and mutation, and algorithm parameters are important factors in determining the order of GA's computation time. They considered in their paper that crossover and mutation is O(n) and selection is O($n$log$n$). Cheung et al. (1998) reported that an adapted SA algorithm for solving the optimal placement of objects had complexity of O($n$log^{2n}), which was significantly improved from general SA approaches. Openshaw and Openshaw (1997) measured the practical computing times of SA to solve census zone design problem, which takes 100 times longer than other spatial heuristics and would become worse as additional constraints are introduced in the algorithm. However, more work is needed to understand the computational time complexity on different classes of problems, different problem

sizes, and different parameter sets of these heuristics.

4. Computational results

4.1. Use of surface specific features

For the first experiments the three surface specific features - peaks, pits and passes - were considered separately. For these experiments, the swap algorithm was used and the number of observers to be placed ranged from 2 to 10.

The numerical results are summarised in table 1 with running times given for a 866 GHz PC with 256 Mbytes of RAM. An exhaustive search was also run in which all 1600 pixels are considered as candidates, but only for the two observer case. The running time was 1.14 hours (4104 seconds) and 57.6 percent of the surface was visible.

<Table 1 about here>

None of the features can match the coverage of the exhaustive search. The best are the peaks, with 46.88 percent followed by the passes at 42.31 percent. The pits perform quite poorly with only 26.94 percent coverage. These results are consistent with the earlier findings regarding the higher visibility values from ridges.

Figure 9 compares the viewsheds produced from the exhaustive search (Figure 9a)

and when only peaks, passes and pits are considered (Figures 9b, 9c, and 9d) for the two observer case. It is interesting to note that the exhaustive search results in both observers being placed on the tops of ridges, although neither is on a point identified by Landserf as a peak or a pass. The patterns for the individual surface specific features show some similarities in that all include a considerable proportion of the north facing side of the large valley which runs east-west across the lower part of the study region. This is at least partly an edge effect since the majority of the selected observer positions are north of this point. Other than that they show the patterns one would expect, with peaks generating a viewshed which covers the hill tops but not many valleys, pits producing the opposite and passes a mixture of the two.

Taking only the peaks as candidate points reduces the CPU time by two orders of magnitude, but at the cost of a combined viewshed which is 10% smaller. As Figure 10 shows, this is not simply a matter of the peaks 'missing' 10% of the area. In fact 13% of the area is visible from the two optimal peaks but not from the two positions selected by the exhaustive search. In fact 24% of the area which is visible in the full set case is not visible from the two optimal peaks, largely because one of the two points can see the large lowland area in the north west corner of the study area.

There is clearly going to be a trade off between the time taken to run a multiple viewpoint query and the quality of the result. The CPU times clearly show the benefits to be gained from reducing the number of candidate points to be considered. The results for the individual features suggest that exactly which points are selected as candidates can make a large difference to the result. Despite having nearly twice as many candidates to choose from, passes always produce a poorer result than peaks.

Pits always perform poorly as candidates, although this may be partly due to the fact that there are far fewer of them. One conclusion from these first tests is that using surface specific features seems to be a promising strategy, especially in the case of peaks.

The next stage in the work was to see whether the inclusion of pits and passes increases the visible area compared with simply using the peaks as the candidate points. A number of other tests were run at the same time. The exhaustive search for two observers identified points which were on high ground, but not on a peak or a pass. Taken together with the earlier observations that points with high visibility tend to be on ridges, it was decided to run a series of tests using the highest ten percent of pixels as the candidates and a second series using the highest ten percent of points in terms of their visibility.

Table 2 shows that in all cases the best results are achieved by considering all pixels as candidates. However, it also illustrates that the loss in total coverage by taking only selected pixels as candidates is not very large. With the two observer case, we saw that the coverage achieved by using just peaks was some 10% below that produced by an exhaustive search. However, as Table 2 shows, this difference reduces considerably as the number of observers to be placed increases. This cannot simply be due to the fact that with more observers placed you are likely to cover more of the area by chance alone. This is clearly a factor, but with the 10 percent highest elevations as candidates, the result for the ten observer case is some 11 percent below the result of an exhaustive search.

The selection of the candidates does not make an enormous amount of difference in terms of the percentage coverage achieved. The 10 percent most visible points give marginally the best results for the cases from two to six observers inclusive, while at higher observer numbers peaks become better. Interestingly the addition of pits and passes to peaks makes very little difference to the coverage obtained, which seems strange given the earlier discussion about the complementarity between the viewsheds for these features individually. Inspection showed that in almost all cases the observers were placed on peaks rather than on pits or passes. This may simply be due to chance, and it would require many more runs to discover whether this is a consistent result.

The ten percent highest elevation points are interesting because at low observer numbers, they produce reasonable results. However above four observers they consistently produce the worst solution which supports the idea that the optimal location search should include some consideration of either the visibility or the terrain position of the candidates and not simply their elevation.

<Table 2 about here>

Since there are a small number of pixels with very high visibility (as stated by Franklin and Ray, 1994), this suggests that these might well figure in a large number of solutions. It can also be supported by looking at the visibility values for the locations which are picked and seeing if they are high - the fact that the 10% most visible pixels worked so well (almost as well as peaks) suggest that highly visible

pixels will be selected in the multiple viewpoint problem, rather than combinations of pixels which have lower visibility values but whose viewsheds do not overlap. As long as maximising total area visible is the objective function this may well be the case. Lee (1994) made the same observation stating that 'those pixels that are on ridge lines and that are peaks tend to have larger numbers of visible pixels' (p. 453). This suggests that some pixel positions may be almost always optimal, regardless of the number of viewpoints to be placed. To investigate this the optimal locations for the problems with between 2 and 10 viewpoints are plotted together in Figure 11. The figure supports the hypothesis that some of the optimal sites share same locations on the DEM. For example, position A in the top left of the map is an optimal location for all of the viewpoint problems, and other places are selected for a number of different location problems (points B and C). These sites would be ideal candidates for the solution heuristics of course, but the difficulty is identifying them in advance, without running a large number of tests as has been done here.

4.2. Comparing heuristics

So far a single heuristic has been used for all the tests, so that the focus has been on the selection of good candidate points. In the final section we compare the three heuristics described earlier to see if they differ in their speed. The previous section showed that there was little difference between the coverage obtained using peaks alone, combinations of surface specific features and highly visible pixels as the candidates. Because the original intention of the work was to explore the use of

surface specific features, it was decided to use all three types of feature as the input to these final tests giving 442 candidate points. All three heuristics require an initial solution from which to start. The same solution was used for all the runs reported here, so that differences in the final solutions are not a product of differences in the initial solution. As before both the total area visible from the selected points and the CPU time are given for each run. For comparison, we estimate that same algorithms using all 1600 pixels as potential candidates, would take 56 hours to solve the 10 observer case.

As the results show (Table 3) the SA heuristic produces the best solution in almost all cases, although the differences between the visible areas produced by each heuristic are not large. However, there is a marked difference in the computing time required by the methods. This becomes more marked as the problem size increases in that the computational performance of the Swap heuristic does not increase as much as the genetic and SA algorithms. For example, for the ten viewpoint problem, the Swap heuristic is approximately 10 times and 20 times faster than the GA and SA heuristics, respectively. This also emphasises the computationally tractable benefit of Swap algorithm on the terrain features because the algorithm completes ten optimal locations problem within 20 minutes compared to the estimated 56 hours for the run on the entire surface area search case (total 1600 candidates).

<Table 3 about here>

The heuristics produce similar results in terms of overall coverage, but it is also interesting to see whether they produce viewsheds which cover the same parts of the

study area. A series of pairwise comparisons was undertaken to measure the degree of agreement between the viewshed areas produced by the three methods, and the results are summarised in Table 4. The proportion of agreement has been adjusted to take account of the fact that because the viewsheds cover a large proportion of the study area, some degree of overlap is inevitable on purely geometric grounds. For example, assume method 1 gives a viewshed of 600 pixels, leaving 1000 pixels which fall outside the viewshed. If the viewshed produced by method 2 is as different as it can be, it will cover those 1000 pixels - if it is larger in extent than this, then the additional pixels will overlap with those in the viewshed of method 1 and from this it is possible to calculate the minimum overlap which will occur purely by chance. Similarly, the maximum possible overlap is simply the smaller of the two viewsheds. The proportional overlap figures in Table 4 are therefore scaled between these minimum and maximum figures. The degree of agreement takes into account the fact that some overlap is inevitable simply because of the size of the viewsheds, which cover over 50% of the study area. Hence 0 would mean there was no agreement other than what was dictated by the size of the viewsheds, it would mean perfect pixel by pixel agreement.

<Table 4 about here>

A fairly consistent pattern emerges from the results, in that there is a relatively high level of agreement between the Simulated Annealing and Genetic Algorithm results (with a perfect match in the two observer case) but a poor level of agreement between these two and the Swap algorithm. Since both the SA and GA methods assess a far larger number of potential solutions than the Swap algorithm, this would suggest that

the solutions they reach are more likely to be optimal. However, the total area of the viewshed produced by these heuristics is not very much larger than that produced by the much quicker Swap algorithm and these results would suggest that a simple and rapid heuristic may well produce results which are nearly as good as more exhaustive methods.

6. Conclusions

One of the shortcomings of the viewshed functions in current GIS is that they are not able to solve optimisation problems, such as identifying an optimal set of viewpoints. This paper has investigated the usefulness of two strategies for tackling this problem. Firstly, the use of fundamental terrain features as candidate viewpoints, and secondly, the modification of a range of spatial optimisation heuristics developed for use in location-allocation problems to the multiple viewpoint problem. The initial results are encouraging in that using these techniques it has been possible to solve the multiple viewpoint problem for a small DEM on a desk top computer in a reasonable time.

However, there is considerable scope for further work in this area. First, there is clearly scope for more work on predicting which parts of the terrain are likely to make good candidate points. The use of peaks, pits and passes in these initial tests was based as much on pragmatic considerations as anything else. The fact that the exhaustive search for the two observers case selected points which were on ridges but not on either a peak or a pass suggests that a more sophisticated selection process is required. The fact that one or two points were repeatedly selected in runs placing

different numbers of observers suggests that there might a few key points in the landscape and the identification of these could considerably speed up the multiple viewpoint problem. Secondly, other types of spatial optimisation algorithms could be explored to improve the optimal search capability for larger problem for which hybrid algorithm and parallel computing technologies can be developed (e.g. Magillo and puppo, 1998).

These developments open up the prospect of developing a system which is capable of producing solutions to the multiple viewpoint problem in a relatively short time, possibly even in real time. At this point the development of a graphical user interface for such a system would also become a topic for further work.

References

Burrough, P. A., McDonnel, R. A., 1998, Principles of Geographical Information Systems, Oxford University Press, London (346 pp.).

Camp, R. J., Sinton, D. T., Knight, R. L., 1997, Viewsheds: A complementary management approach to buffer zones, Wildlife Society Bulletin, 25, 612-615.

Carson M.A., Kirkby M.J., 1972, Hillslope Form and Process. Cambridge University Press, London (475 pp.).

Cheung, S. K., Leung, A., Albrecht, A., Wong, C. K., 1998, Optimal placements of flexible objects: An adaptive simulated annealing approach, Lecture Note in Computer Science 1498, Springer-Verlag, Berlin, 968-977.

Fisher, P. F., 1991. First experiments in viewshed uncertainty: the accuracy of the viewshed area, Photogrammetric Engineering and Remote Sensing 57, 1321-1327.

Fisher, P. F., 1993. Algorithm and implementation uncertainty in viewshed analysis, International Journal of Geographical Information Systems 7 (4), 331-347.

Fisher, P. F., 1996. Extending the Applicability of Viewsheds in Landscape Planning. Photogrammetric Engineering & Remote Sensing 62 (11), 1297-1302.

De Floriani, L., Marzano, L., Puppo, P. E., 1994. Line-of-sight communication on terrain models, International Journal of Geographical Information Systems, 8 (4), 329-342.

De Floriani, L., Magillo, P., 1997. Visibility computations on hierarchical triangulated terrain models. GeoInformatica 1, 219-250.

Franklin, W. R., 2000. Applications of Analytical Cartography, Cartography and

Geographic Information System, 27 (3), 225-237.

Franklin, W. R., Ray, C. K., 1994. Higher isn't Necessarily Better: Visibility Algorithms and Experiments. In: Waugh, T. C., Healey, R. G., (Eds.), Advances in GIS Research: 6th International Symposium on Spatial Data Handling, Edinburgh, Scotland. Taylor and Francis, London, pp. 751-770.

Goodchild, M., Lee, J., 1989. Coverage problem s and visibility regions on topographic surface, Annals of Operations Research, 18, 175-186.

He, J., Yao, X., 2001. Drift analysis and average time complexity of evolutionary algorithms, Artificial Intelligence. 127 (1), 57-85.

He, J., Yao, X., 2003. Towards an analytic framework for analysing the computation time of evolutionary algorithms, Artificial Intelligence, 145 (1-2), 59-97.

Houch, C. R., Joines, J. A., Kay, M. G., 1996. Comparison of genetic algorithms, random restart, and two-opt switching for solving large location-allocation problems. Computers and Operations Research. 23 (6), 587-596.

Kidner, D., Sparkes, A., Dorey, M., 1999. GIS and Wind Farm Planning. In Stillwell, J., Geertman, S., and Openshaw, S. (Eds.), Geographical Information and Planning, Springer, London, pp. 203-223.

Krzanowski, R. M., Raper., J., 2001. Spatial Evolutionary Modeling, Oxford

University Press, London (264 pp.).

Lake, I. R., Lovett, A. A. Bateman, I. J., Langford, I. H., 1998, Modelling environmental influences on property prices in an urban environment, Computers, Environment and Urban Systems, 22 (2), 121-136.

Lake, M. W., Woodman, P. E., Mithen, S. J., 1998, Tailoring GIS software for Archaeological Applications: An Example Concerning Viewshed Analysis, Journal of Archaeological Science, 25, 27-38.

Lee, J., 1991. Analyses of visibility sites on topographic surfaces. International Journal of Geographical Information Systems, 5(4), 413-429.

Lee, J., 1992. Visibility dominance and topographical features on digital elevation models, Proceedings of the fifth international symposium on spatial data handling, Charleston, South Carolina, 2: 622-631.

Lee, J., 1994. Digital Analysis of Viewshed Inclusion and Topographic Features on Digital Elevation Models. Photogrammetric Engineering and Remote Sensing 60 (4), 451-456.

Lee, J., Stucky, D., 1998. On applying viewshed analysis for determining least-cost paths on Digital Elevation Models, International Journal of Geographical Information Science, 12 (8), 891-905.

Liu, C. M., Kao, R. L., Wang, A. S., 1994. Solving Location-allocation problems with rectilinear distances by simulated annealing. Journal of Operational Research Society 45 (11), 1304-1315.

Magillo, P., Puppo, E., 1998. Algorithms for parellel terrain modelling and visualisation, In Parallel Processing and Algorithms for GIS, Healey, R., Dowers, S., Gittings, B., Mineter, M., (eds.) pp. 351-386.

Miller D.R., Law A.N.R., 1996. The mapping of terrain visibility. Cartographic Journal, 34 (2), 87-91.

Murray, A. T., Church, R. L., 1995. Heuristic solution approaches to operational forest planning problems. OR Spektrum 17, 193-203.

O'Sullivan, D., Turner, A., 2001. Visibility graphs and landscape visibility analysis. International Journal of Geographical Information Science 15 (3), 221-237.

Openshaw, S., Openshaw, C., 1997. Artificial Intelligence in Geography, Wiley. Chichester (pp.329.).

Rana, S., 2003. Fast approximation of visibility dominance using topographic features as targets and the associated uncertainty, Photogrammetric Engineering and Remote Sensing, 69 (8), 881-888.

Rallings, P., Kidner, D., Ware, A., 1999. Distributed viewshed analysis for planning

application. In Innovations in GIS 6. Getting, B. (ed.), Taylor and Francis, London, pp. 185 - 199.

Rosing, K. E., Hillsman, E. L., Rosing-Vogelaar, H., 1979. A note comparing optimal and heuristic solutions to the p-median problem. Geographical Analysis, 11, 86-89.

Sorensen, P., Lanter, D. P., 1993. Two Algorithms for Determing Partial Visibility and Reducing Data Structure Induced Error in Viewshed Analysis, Photogrammetric Engineering and Remote Sensing, 59 (7), 1149-1160.

Teitz, M. B., Bart, P., 1968. Heuristic methods for estimating the generalized vertex median of a weighted graph. Operations Research 16. 955 - 961.

Wang, J., Robinson, G. J., White, K., 1996. A Fast Solution to Local Viewshed Computation Using Grid-Based Digital Elevation Models, Photogrammetric Engineering & Remote Sensing, 62 (10), 1157-1164.

Wood, J. 1998. Modelling the continuity of Surface form using Digital Elevation Models. Proceedings 8th International Symposium on Spatial Data Handling. Vancouver, Canada, pp. 725-736.

Wise, 2002, GIS basics, Taylor and Francis, London (p.240.).

[1] Landserf web page. http://www.landserf.org

Hillsha

2   0   2   4   Kilometers

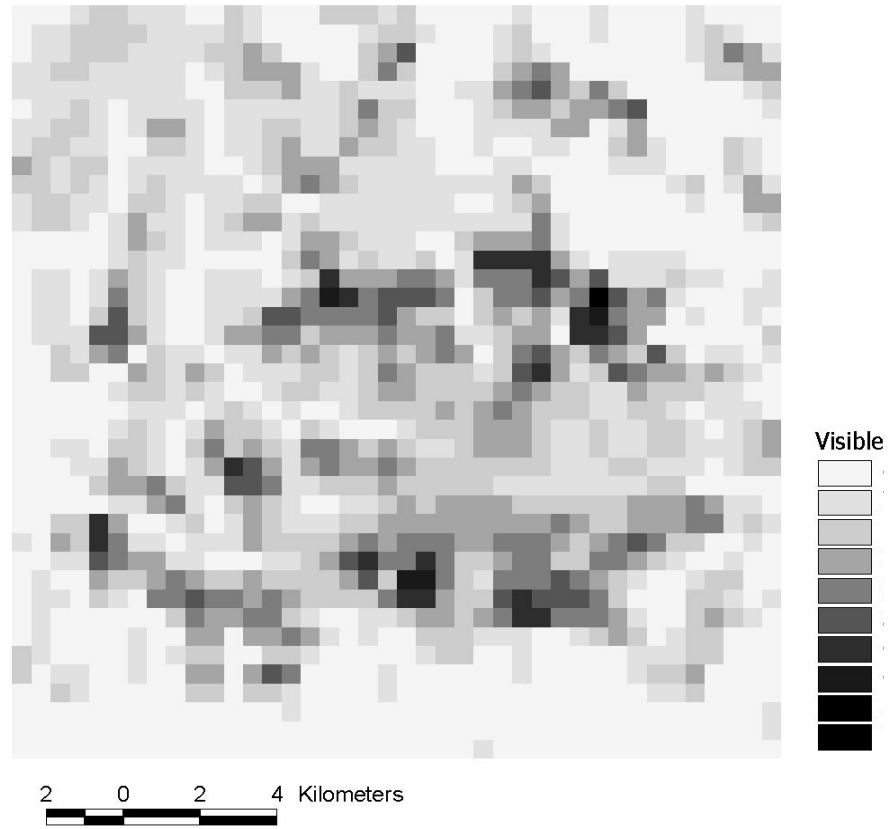Figure 1. Hillshade image and elevation map of Cairngorm test area
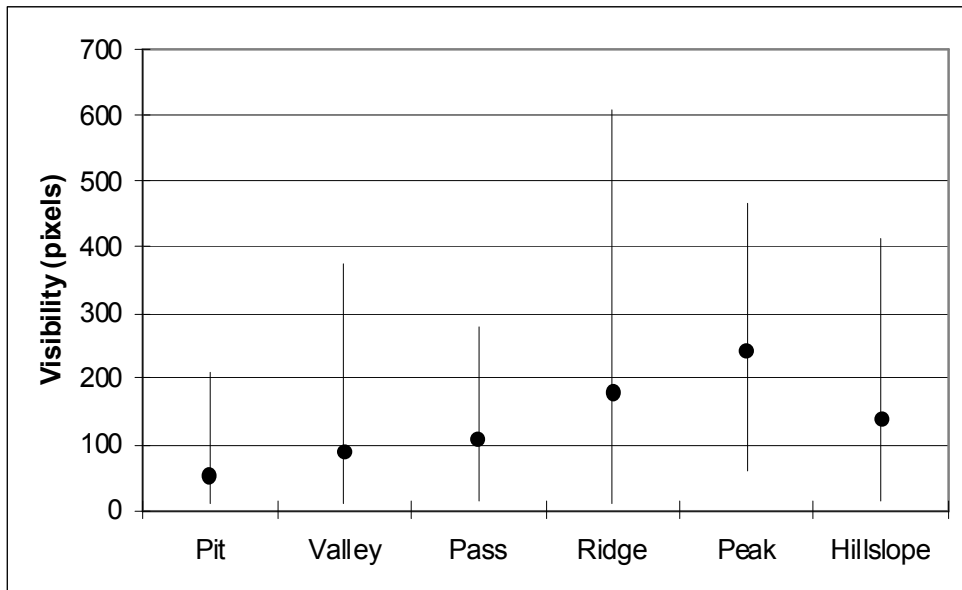
Figure 2. Visibility index map

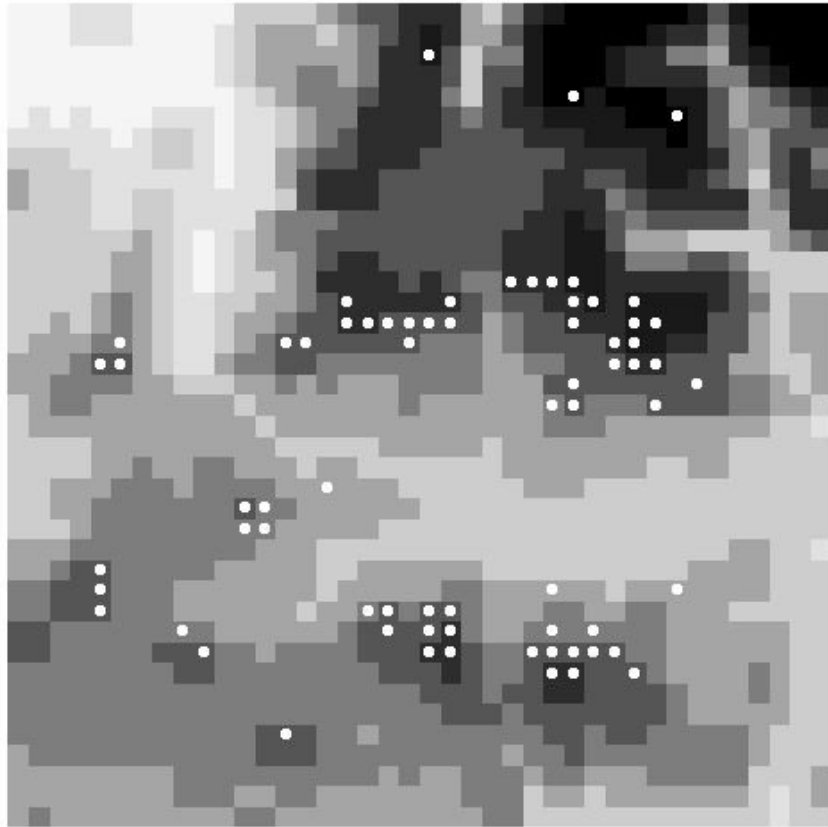Figure 3 Visibility index mean and range for each surface type.

Figure 4. Location of highly visible points

Light grey - visible from only one of the two points
Dark grey - visible from both points
White - visible from neither point

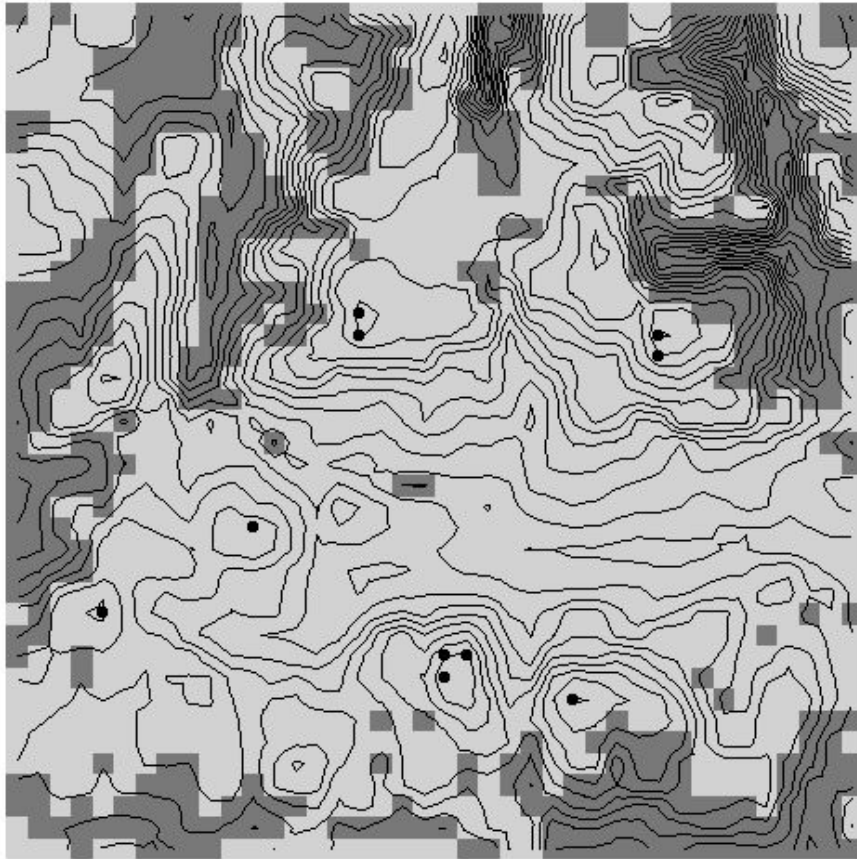Figure 5. Overlap visibility between viewsheds
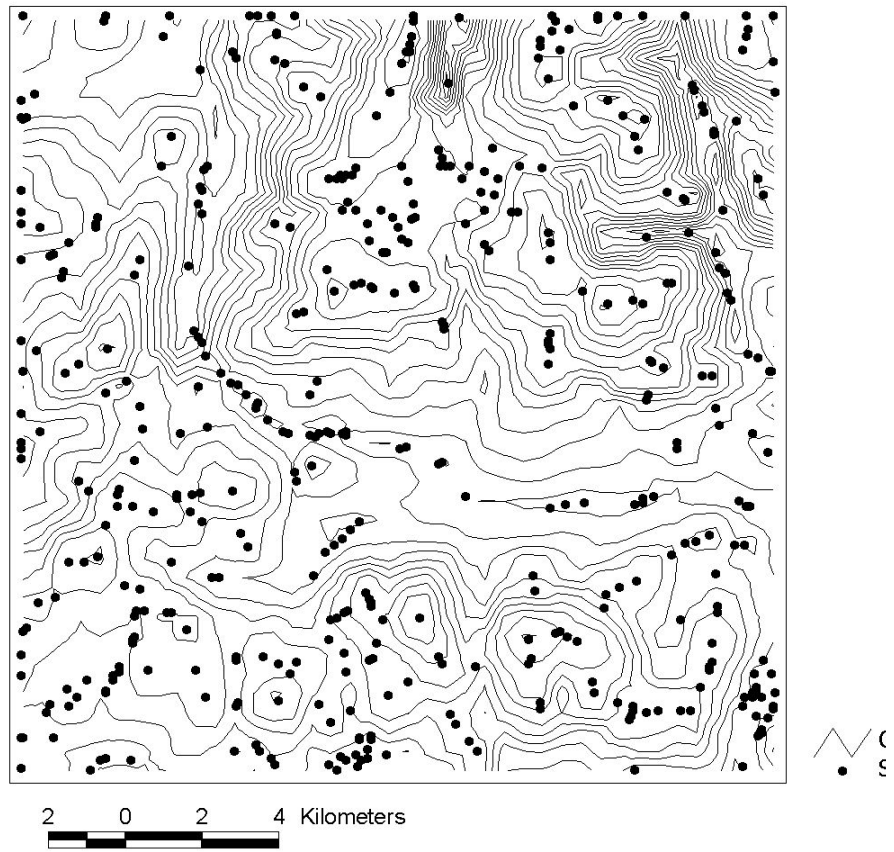
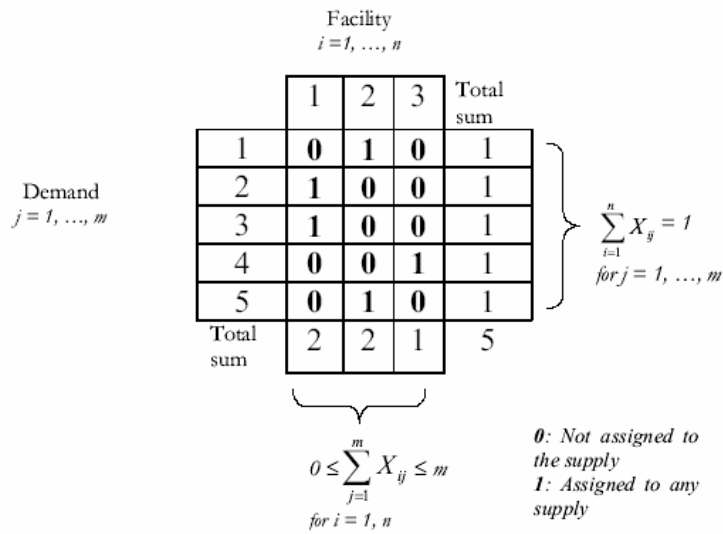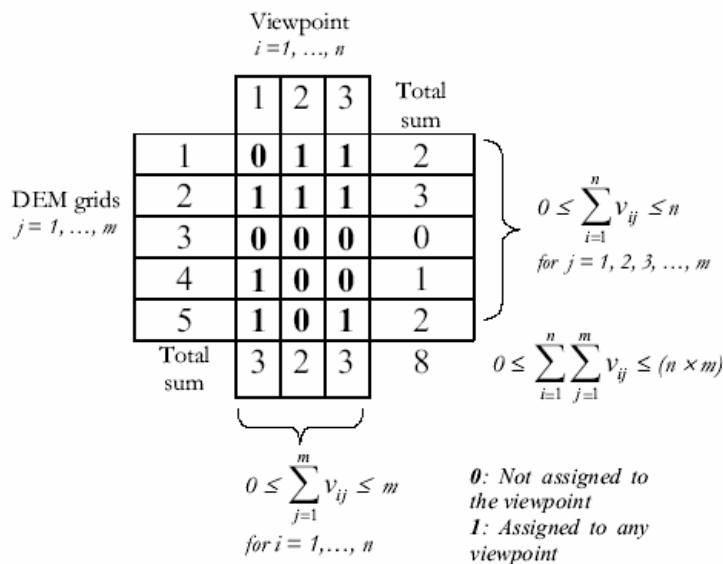Figure 6. The combined viewsheds of ten most visible pixels

Figure 7. The sample DEM contours and surface specific points distribution

Facility
$i = 1, ..., n$

Demand $j = 1, ..., m$	1	2	3	Total sum
1	0	1	0	1
2	1	0	0	1
3	1	0	0	1
4	0	0	1	1
5	0	1	0	1
Total sum	2	2	1	5

$$\sum_{i=1}^{n} X_{ij} = 1 \quad \text{for } j = 1, ..., m$$

$$0 \le \sum_{j=1}^{m} X_{ij} \le m \quad \text{for } i = 1, n$$

0: Not assigned to the supply
1: Assigned to any supply

(a) Constraint relationships in facility location-allocation problems

Viewpoint
$i = 1, ..., n$

DEM grids $j = 1, ..., m$	1	2	3	Total sum
1	0	1	1	2
2	1	1	1	3
3	0	0	0	0
4	1	0	0	1
5	1	0	1	2
Total sum	3	2	3	8

$$0 \le \sum_{i=1}^{n} v_{ij} \le n \quad \text{for } j = 1, 2, 3, ..., m$$

$$0 \le \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} \le (n \times m)$$

$$0 \le \sum_{j=1}^{m} v_{ij} \le m \quad \text{for } i = 1, ..., n$$

0: Not assigned to the viewpoint
1: Assigned to any viewpoint

(b) Constraint relationships in multiple viewshed problems

where X_{ij} and v_{ij} denote assignment of supply or viewpoint

Figure 8. The comparison of the constrains of facility location and multiple viewshed problems
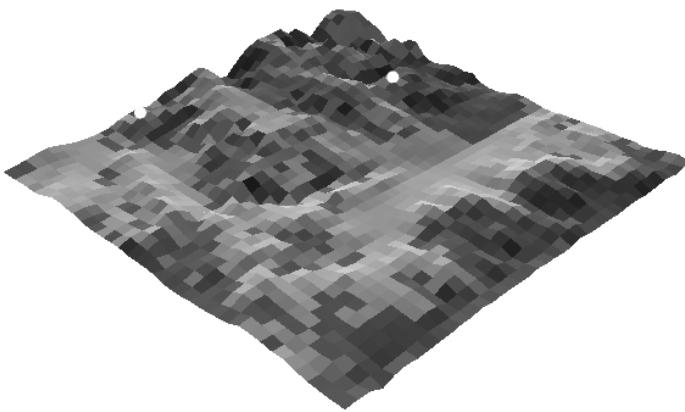
Figure 9. Example of different viewshed results: two best viewpoint location problems case
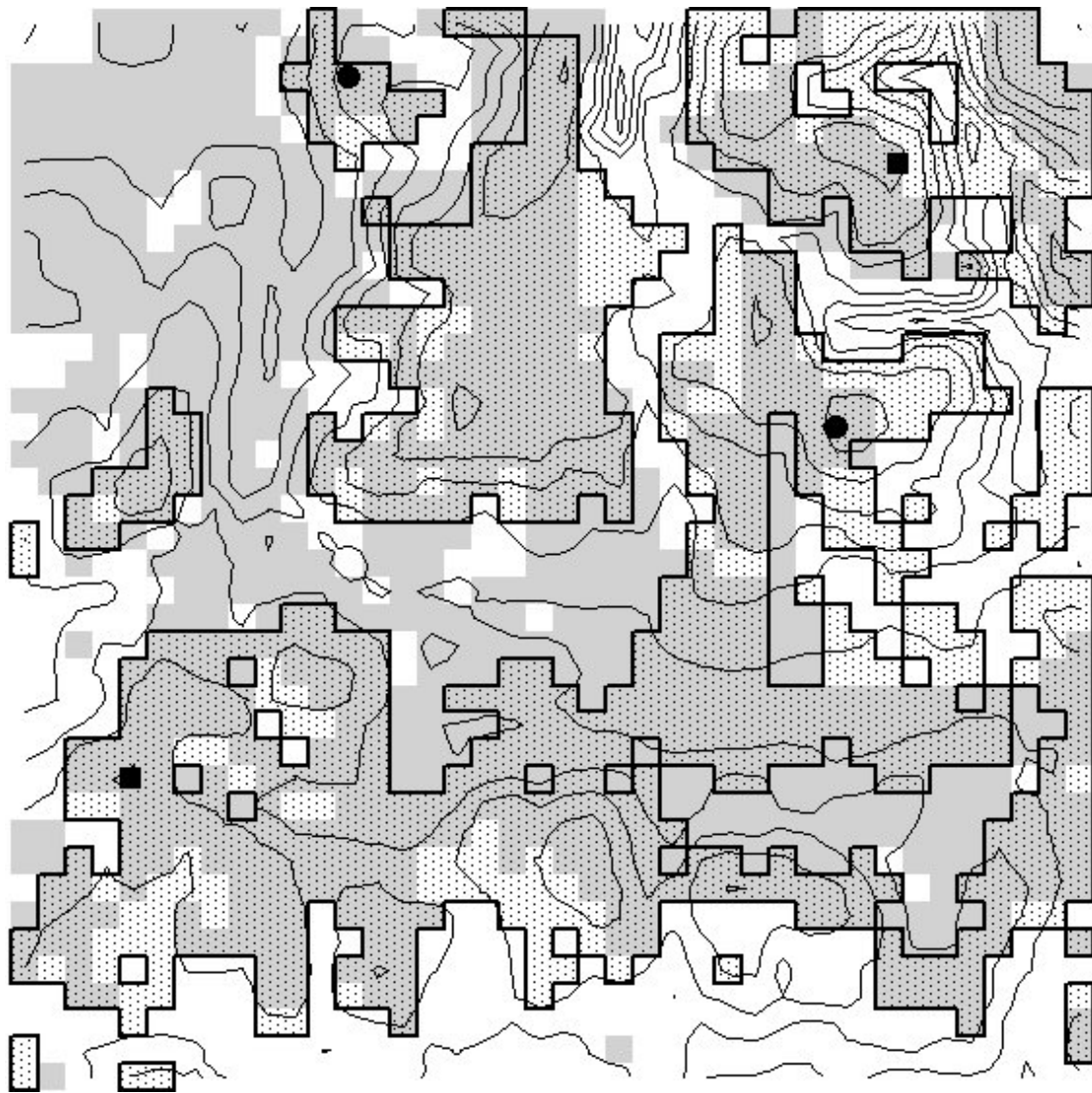
Figure 10. Comparison between results for two observers case when all pixels are considered as potential candidates (optimal locations as black square, viewshed as grey shading) and when only peaks are considered as candidates (optimal locations as black circles, viewshed as stippled areas).
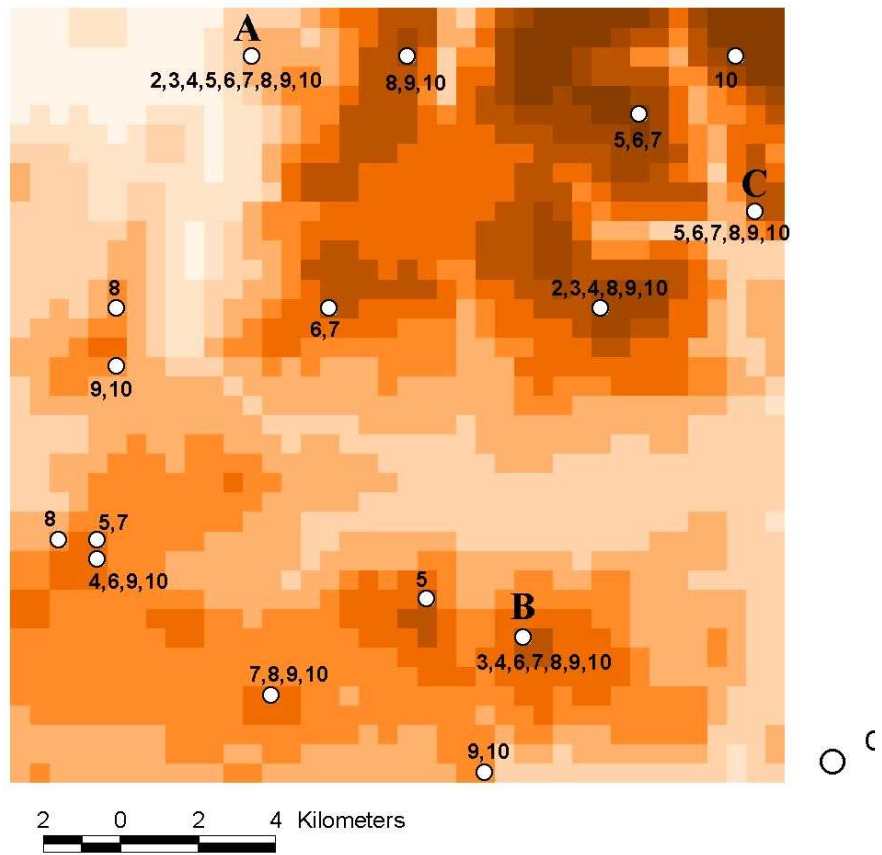
Figure 11 Mapping comparisons of optimal viewpoint locations

Observers (h)			Peak feature (136)		Pit feature (53)	Pass feature (253)
	Percentage of area visible		CPU time (sec)	Percentage of area visible		CPU time (sec)
		Percentage of area visible		CPU time (sec)		
2 3 4 5 6 7 8 9 10		46.88 57.19 64.81 71.75 75.50 79.63	81.81 83.13 85.00	34.88 78.98 153.57 209.82 294.62		
357.56 448.58 542.99 668.06	26.94 31.81 36.31 40.38 43.63 46.69 49.94 52.25 54.50		19.00 23.57 47.84 99.86 149.62 99.09			
120.83 148.08 180.81		42.31 52.94 58.88 63.50 67.31 70.56 74.50 76.50 78.69		51.73 94.91 168.07 235.14 310.66		
		419.46 519.59 647.74 812.29				

Table 1. Performance results of three terrain features

Viewpoints	Peak	Peak+Pass	Peak+Pit	10%_cov[1]	10%_elv[2]	All pixels[3]
2	46.9	46.9	46.9	51.1	50.2	52.2
3	57.2	57.2	57.2	62.3	59.9	63.3
4	64.8	66.3	64.8	66.3	64.3	69.1
5	71.8	71.9	72.3	73.2	67.8	73.6
6	75.5	76.6	75.5	76.3	70.5	77.7
7	79.6	79.6	79.6	79.4	73.1	80.6
8	81.8	80.9	81.8	81.1	74.4	83.1
9	83.1	83.1	83.7	83.0	75.5	85.3

10	85.0	84.9	84.9	84.5	76.4	87.0

1 10%_cov = the pixels of ten percent best visibility
2 10%_elv = the pixels of ten percent highest elevation
3 All pixels = all pixels of the study area' DEM data

Table 2. The viewshed results of peak type versus combined feature types

Obs	Swap algorithm		Genetic Algorithm		Simulated Annealing Algorithm	
	Percentage of area visible	CPU time (sec)	Percentage of area visible	CPU time (sec)	Percentage of area visible	CPU time (sec)
2 3 4 5 6 7 8 9 10	46.88 57.19 66.13 72.25 76.56 79.31 81.63 83.06 84.94	73.66 179.94 252.88 364.37 520.57 657.9 794.82 965.2 1157.45	47.06 56.94 64.19 66.44 72.69 72.38 78.19 82.00 78.19	208.77 355.48 780.22 1283.88 3694.51 1953.64 5898.34 9911.8 18627.2	47.06 58.56 66.31 72.00 77.13 79.63 81.81 83.81 85.56	1075.66 3492.16 4483.9 5475.64 7172.45 11548.7 18640.25 25731.8 29740.4

Table 3. The solution and computing time results of the visibility heuristics for all the three terrain features

Number of observers	Swap and GA	Swap and SA	GA and SA
2	0.66	0.66	1.00
3	0.74	0.59	0.72
4	0.56	0.57	0.94
5	0.61	0.54	0.63
6	0.66	0.52	0.84
7	0.56	0.53	0.75
8	0.73	0.55	0.90
9	0.54	0.57	0.79
10	0.66	0.57	0.76

Table 4. Agreement between heuristics in terms of predicted viewsheds.