TERRAIN RECONSTRUCTION FROM GROUND-BASED LASER DATA

Bruce King[1], Elzbieta Matuk[1], Krzysztof Matuk[1], Christopher M. Gold[2]
[1]Address: The Department of Land Surveying & Geo-Informatics
The Hong Kong Polytechnic University, Hong Kong.
Tel: (852) 2766 5968
Fax: (852) 2330 2994
E-mail: {bruce.king | ela.matuk | krzysiek.matuk}@polyu.edu.hk
[2]Address: School of Computing, University of Glamorgan, Wales, UK
E-mail: christophergold@voronoi.com

ABSTRACT:

Terrestrial laser scanners are capable of collecting large data sets from an individual location in a few minutes. This permits the remote construction and monitoring of terrain, such as road cuts susceptible to landslides, in a very short time. Software is under development to merge these individual scans into a large global terrain model capable of being updated as required. The method is based on the pair wise matching of surface patches using the method of spin images. Once the pair wise transformation matrices between pairs of scans have been estimated, a global terrain model is constructed. This model may be edited by replacing individual scans with newer versions.

1. INTRODUCTION

In the Hong Kong environment, with steep slopes, tropical weathering, heavy rainfall and high erosion, any road construction introduces a significant risk of landslides. The Hong Kong Government maintains a large slope monitoring program, which includes rapid response to any slope failures. One aspect of this response is a preliminary survey of the landslide surface – a procedure which is both approximate and dangerous using traditional techniques.

We have undertaken a project to build and maintain models of road cuts based on ground-based laser scanning. This project has several parts. Firstly, we need to be able to construct a terrain surface from the range data. This involves straightforward triangulation techniques for simple surfaces, but becomes more complex when scans cover several discontinuous portions of the terrain. Secondly, overlapping scans need to be merged. This can be done by precise surveying of the scanner location and orientation, but this is difficult in an emergency-response situation. Recent techniques allow matching of pairs of terrain patches of unknown relative orientation, using "spin images". This gives an approximation of the transformation matrix that gives a best-fit of the overlapping portions of the two patches. Thirdly, multiple pair-wise matches need to be adjusted to give the best overall fit. Fourthly, the system needs

to be capable of removing individual scans and replacing them with more recent ones as required, e.g. during response to slope failures. The result will be a system that can maintain large-scale terrain models for emergency planning and slope stabilization. While the main algorithmic processes have been defined, and in some cases implemented, much work remains to be done on the development of a system capable of handling the large quantities of data involved.

In this paper, the basic concepts of spin images are explained and the results of testing presented. Two data sets - Bunny, a data set used in the original development of the spin image concept, and Tiles, one generated from scanning a tile dump with a CYRAX 2500 model scanner and used to represent the intended application –are used for testing purposes.

2. CONSTRUCTING A TERRAIN MODEL FROM A SINGLE SCAN

A laser scan is collected as a grid of depth observations, and this grid may be considered to be in the "plane" of the scanner. The resulting set of height (or depth) values may then be triangulated by any triangulation (TIN) procedure available, to produce a Delaunay triangulation (DT) with the (X, Y) coordinates defined in this plane. This DT is valid in this orientation, but will not be valid when compared with another DT collected with the scanner in another orientation. Nevertheless, this model serves to provide the basic TIN structure of a single scan.

3. MATCHING PAIRS OF TERRAIN MODELS COLLECTED FROM DIFFERENT SCANS

The approach to surface matching is based on matching individual surface points in order to match complete surfaces. Two surfaces are said to be similar when many points from the surfaces are similar. An effective measure of point similarity is based on the concept of "Spin images" developed in [12,13] by Johnson et al.. The objective is to estimate the transformation matrix, composed of rotations and translations, that best moves one surface model onto the other, assuming some reasonable overlap area. The scales are assumed to be the same in both cases. To match two points on different surfaces we compare the surfaces around the normals to the surface at each point, as we do not yet know the relative orientations of the surfaces. A spin image is a summary of the neighbouring points, spun around the normal axis. If two points are from the same shaped surface these images would be the same. The matching of several pairs of spin images from point pairs on the two surfaces, gives the control points necessary for surface matching.

3.1 Spin Images

Spin images (Johnson, 1997) are simply transformations of the surface data; they are created by projecting 3-D points into 2-D images. A fundamental component of that surface matching representation is an ***oriented point***- a three-dimensional point with an associated direction.

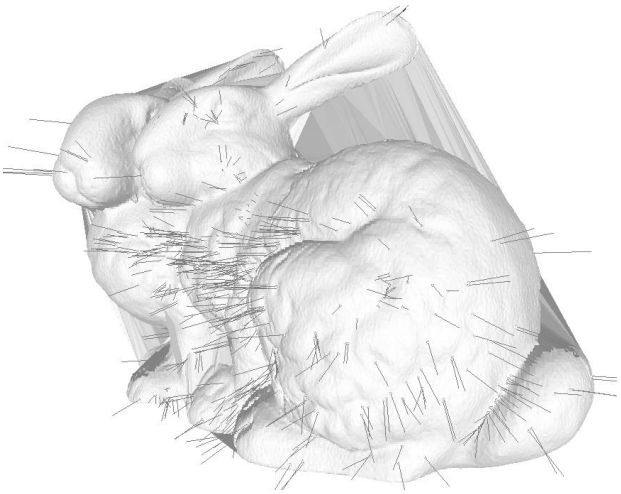


Figure 1: Oriented points and their normal vectors

Oriented points are used to create spin images. An oriented point *O* at a surface mesh vertex is defined using the 3-D position of the vertex ***p*** and surface normal at the vertex ***n***. The surface normal at a vertex is computed by fitting a plane to the points connected to the vertex by the surface mesh (Fig. 1).

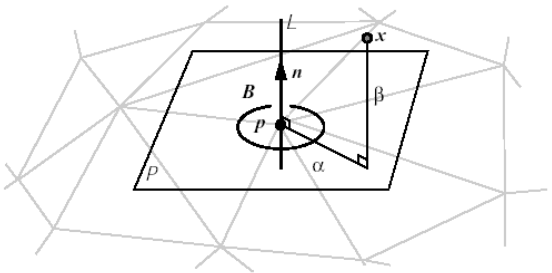


Figure 2: An oriented point basis created at a vertex in a surface mesh. From [12].

An oriented point defines a 5 degree of freedom (DOF) basis ***(p,n)*** (i.e., local coordinate system) using the tangent plane *P* through ***p*** oriented perpendicularly to ***n*** and the line *L* through ***p*** parallel to ***n***. The two coordinates of the basis are α, the perpendicular distance to the line *L*, and β, the signed perpendicular distance to the plane *P* (Fig. 2). An oriented point basis is a cylindrical coordinate system that is missing the polar angle coordinate (because this coordinate cannot be determined using just surface position and normal). Using an oriented point basis *O*, we can define a *spin-map* S_O as the function that projects 3-D points ***x*** to the 2-D coordinates of a particular basis ***(p,n)*** corresponding to oriented point *O*.

$$s_0 : R^3 \to R^2$$
$$S_0(x) \to (\alpha, \beta) = \left(\sqrt{\|x-p\|^2 - (n.(x-p))^2}, (n.(x-p)) \right) \quad (1)$$

Although α cannot be negative, β can be both positive and negative.

A spin image is created by:

a. first an oriented point *O* is created from a vertex of the surface mesh
b. for each vertex ***x*** on the surface of the object, the spin-map coordinates with respect to *O* are computed (1).
c. if ***x*** meets some criteria based on distance from *O* and angle between *O* and the surface normal of ***x*** , the bin that its spin-map coordinates index is determined and the 2-D array is updated by incrementing the surrounding bins in the array. A simple way to update the array for each point ***x*** would be to increment the bin to which the point is spin-mapped by one.

Laser scan point clouds typically contain a large number of points. In order to reduce the computation time only a subset of all available points are used as possible oriented points. In our case for both Bunny and Tiles – each scan has approximately 40,000 points - we selected 600 points for further processing (we also run process up to 1500 oriented points). Thus it is important to make sure that selected points are meaningful (e.g. are located on a feature).

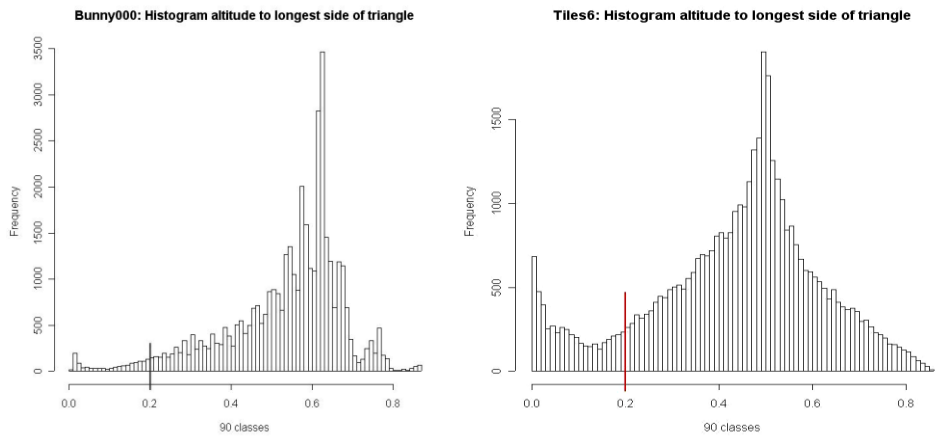


Figure 3: Sample histograms from the Bunny (left) and Tiles (right) data sets showing the cut-off points.

Some of the triangles created by the triangulation process must be removed because they do not accurately represent the surface. These include points on the boundary of the object as well as noise along the line sight of the scanner. Usually those triangles are long and skinny (Fig. 3). To reject those undesirable triangles we checked the ratio of altitude to longest side and built a histogram of this statistic. The histogram was used to decide a cut-off value above which the triangles were considered to be valid. The remaining points were assigned an importance value based upon minimum cosine value of the angle between two normal vectors: one at the point (created as a sum of the normal

vectors of the adjacent triangles) and second for an each adjacent triangle to the point. The bigger angle → the lower cosine value → the more important is the point.

When all of the points on the surface of the object have been accumulated, a 2-D array representation of the spin image is generated.

There are three parameters that control spin image generation:

- bin size (based on mesh resolution) - it determines the storage size of the spin image
- image width - define the number of rows or columns in a square spin image. (Image width times the bin size is called the spin image *support distance Ds*; support distance determines the amount of space swept out by a spin image.)
- support angle *As* - the maximum angle between the direction of the oriented point basis of a spin image and the surface normal of points that are allowed to contribute to the spin image. Support angle is used to limit the effect of self occlusion and clutter during spin image matching. In general A_S is set between 90° and 60°.

$$\mathbf{a\cos(n_A.n_B) < A_S} \qquad (2)$$

where:

A is an oriented point with position and normal $\mathbf{(p_A, n_A)}$

B is an oriented point with position and normal $\mathbf{(p_B, n_B)}$

Equation (2) means that B will be accumulated in the spin image of A (Fig. 4).

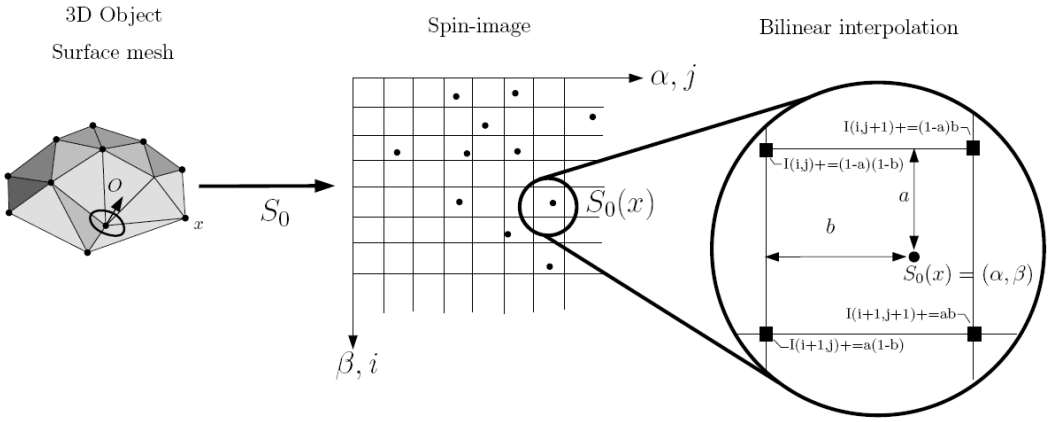


Figure 4: The addition of a point to the 2-D array representation of a spin image. From [12].

3.2 Comparing spin images

Because spin images are pose independent, where the pose represents the position and orientation of the scanner, the two objects will have similar spin images if they are generated with the same spin image generation parameters. By directly comparing spin images from one object with spin images from the other object, point correspondences can be established between points from two different objects if they have the same spin image. The point correspondences can then be used to localize one object with respect to the other. Implicit in this method of recognition is a method for comparison of spin images (Fig. 5).

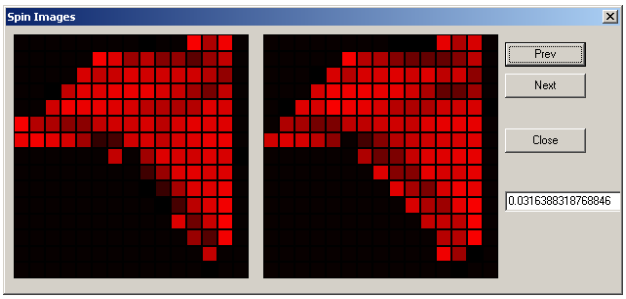
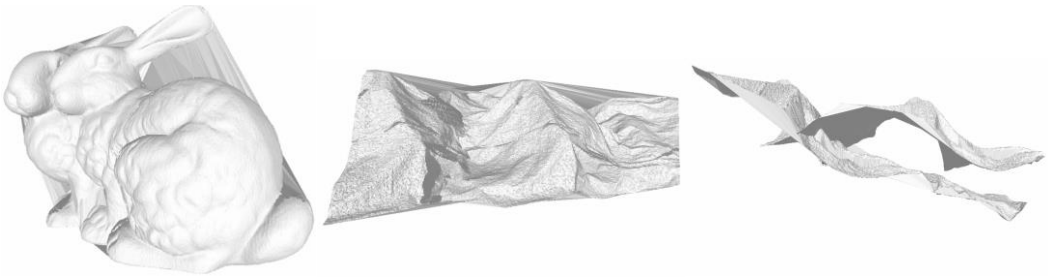


Figure 5: Spin images of two scans of Bunny

Two spin images from proximal points on the surface of two different instances of an object can be expected to be linearly related because the number of points that fall in corresponding bins will be similar. Since the bin values are directly related to the number of points falling into the bins, the bin values will be similar. Fig. 6 shows two examples pairs of meshes that must be matched by finding a suitable transformation matrix.



a) **b)** **c)**

Figure 6: Pairs of meshes to be merged by transformation: (a) Bunny, (b) and (c) Tiles

The standard way of comparing linearly related images is the normalized *linear correlation coefficient R*. Given two spin images *P* and *Q* with *N* bins each, the linear correlation coefficient *R(P,Q)* is:

$$\mathbf{R(P,Q)} = \frac{N\sum p_i q_i - \sum p_i \sum q_i}{\sqrt{(N\sum p_i^2 - (\sum p_i)^2)(N\sum q_i^2 - (\sum q_i)^2)}} \qquad (3)$$

where **R** is between -1 (anti-correlated) and 1 (completely correlated).

Since the linear correlation coefficient is a function of the number of pixels used to compute it, the amount of overlap between spin images will have an effect on the correlation coefficients obtained.

The confidence in the correlation coefficient can be measured by its variance. Both R and its variance are combined into a single function, the *similarity measure C*:

$$C(P,Q) = (a\tanh(R(P,Q)))^2 - \lambda \frac{1}{(N-3)} \qquad (4)$$

This similarity measure is a heuristic loss function that will return a high value for two images that are highly correlated and have a large number of overlapping bins. λ weights the variance against the expected value of the correlation coefficient. It controls the point at which overlap between spin images becomes important for comparison of spin images.

3.3 Establishing point correspondences

The method used for establishing a set of point correspondences between oriented points in a model and in a scene is:

1. create spin images m_j for all points on a model surface mesh M
2. select a random vertex from a scene mesh S, and create spin image s_i
3. compute the similarity measure between s_i and all m_j in M
4. build a similarity measure histogram
5. reject those correspondence pairs for which length difference between fixed point on mesh (point with maximal similarity value) and corresponding points on both matching meshes are larger than an assumed value (based on length difference histogram)
6. eliminate multiple pairs (one point from the model mesh is related with only one on the scene mesh, and vice versa) – by choosing the pair with larger similarity measure value

Originally only steps 1 to 4 were used which resulted in redundant matching of orientation points. Once this problem was recognised the process was improved by adding steps 5 and 6. For the Bunny data set, using only steps 1 to 4 resulted in 60 sets of matching points. Applying step 5 the set of correspondence points reduced to 22. Upon inspection it was found that 3 were duplicated, so finally step 6 gave an unambiguous set of 19 matched points.

3.4 Transformation of scan

Single correspondences cannot be used to compute a transformation from model to scene because an oriented point basis encodes only five of the six necessary degrees of freedom. At least two oriented point correspondences are needed to calculate a transformation if position and normal are used. In our method at least three correspondence points in both systems are required. To avoid combinatory explosion, geometric consistency is used to determine a groups of correspondences from which plausible transformations can be computed.

A plausible 3D rigid transformation *T* from model to scene is calculated from each group {[***mi,si***]} of correspondences by minimizing:

$$\mathbf{E_T = \sum \left\| s_i - T(m_i) \right\|^2} \qquad (5)$$

Initial approximation of the transformation parameters (three rotations – omega, phi, kappa and three translations – in x, y, z) has to be estimated. To do so, the method proposed by Dewitt [7] has been used. When more than three correspondence points are available, the altitude from the longest side of the all possible triangles created from selected points is computed and the triangle with the largest altitude is chosen. It appeared to be not good enough, for choosing the best triangle for initial approximation. Process was modified by adding one more condition related to similarity between triangles on both matching meshes. Together with altitude, sum of length difference of corresponding sides of the triangles from two merged meshes should be not higher then median values of measures calculated for all possible triangles. Figure 7 shows points which create the triangles for Bunny and Tiles.

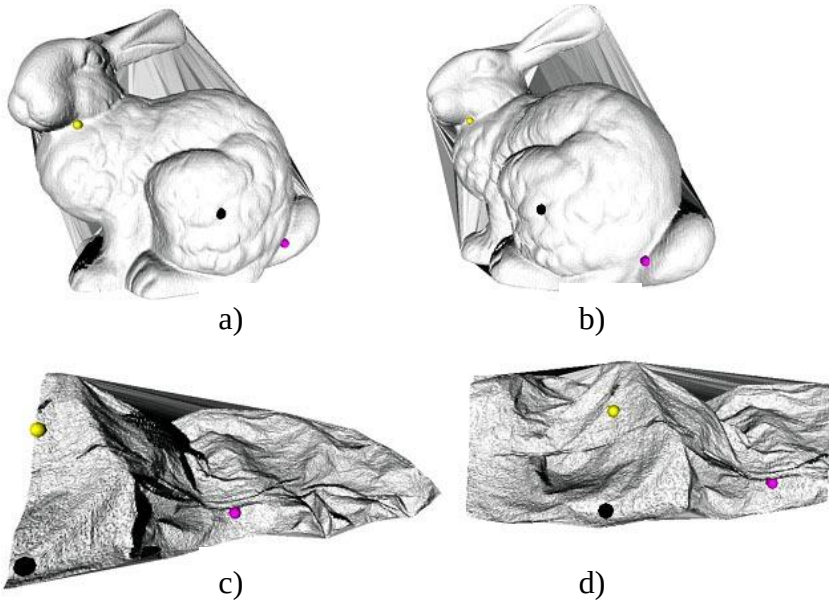


Figure 7: Correspondence points used in initial approximation, (a) and (b) – two merged Bunny patches, (c) and (d) – two merged Tiles patches

During least squares iteration the RMS error between the two data sets can be computed and used for detection of points which should be excluded from the adjustment.

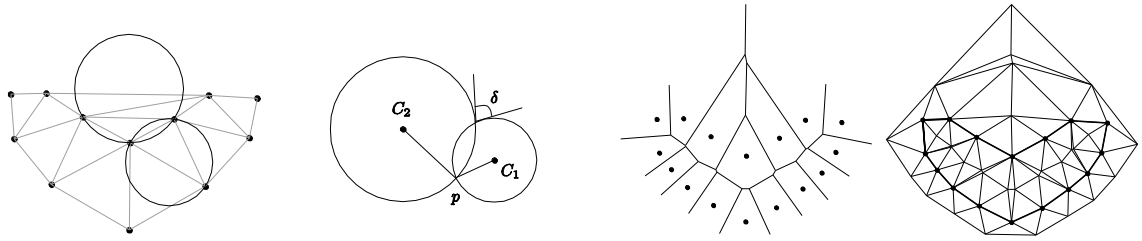
4. BUILDING AND EDITING THE GLOBAL MODEL

After model registration the global model must be built. This involves the selection of an appropriate subset of points from the various overlapping scans. This can be done within the original coordinate systems of the individual scans, by eliminating points in one scan that too closely approach the surface

of the second, and vice versa, for various scan pairs. Then a global triangulated surface must be constructed. This is not usually feasible for a complete scan of a solid "in the round" when using a 2D Delaunay triangulation, as there is no appropriate projection plane, but it is feasible using the 3D Voronoi diagram or Delaunay triangulation. Finally, in order to permit the insertion and removal of individual scans on an operational basis for surveying applications, all points used must be labelled with their source scan number, for updating purposes.

5. SURFACE RECONSTRUCTION

A visualization of the model requires the surface of the object to be extracted from aligned point clouds. The reconstruction of a curve in 2D from samples, taken from a smooth curve, can be done quite easy by utilizing properties of a medial axis of the curve (see [6], [1] and [9] for details). The algorithms by Attali in [6]**,** Amenta [1] and Gold [9,10] use fact that in two dimensions when sampling density increases, Voronoi diagram of the sample points converges to the medial axis of the object (curve). The algorithm by Attali checks the angle δ formed by two Delaunay circles (Figure 1.a)) and 1.b)). The circles are circumscribed on two triangles sharing a common edge. If the angle is smaller than some δ_0 the edge has chance to be a boundary edge. As shown in [6], for $\delta_0=\pi/2$ a Delaunay edge is the boundary edge.



a) **b)** **c)** **d)**

Figure 8: Planar curve reconstruction algorithms by Amenta a) and b), and Attali c) and d).

Another approach for boundary extraction has been proposed by Amenta et al. in [1]. Their method is composed of two stages. In the first stage, it computes the Voronoi diagram of an input dataset. In the stage two, the Delaunay triangulation of the input plus the Voronoi vertices computed in the first stage is computed (Figure 2.c and 2.d). The extraction of the boundary of the shape (crust) is based on search for Delaunay edges connecting two input points and skipping these joining sample points with Voronoi vertices from stage one.

A very simple algorithm for extraction of the skeleton from samples placed along two dimensional curves has been described by Gold and Snoeyink in [9] and [10].

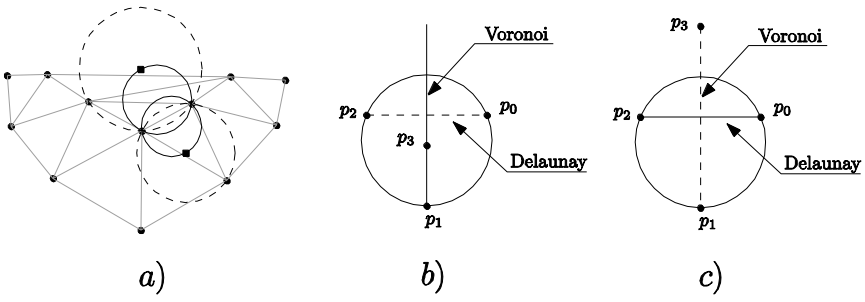
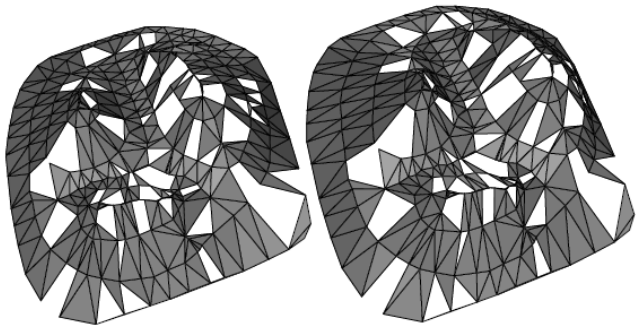


Figure 9: One step crust and skeleton extraction algorithm

The crust and the skeleton extraction relies on the fact that a Delaunay edge is a part of the crust if a circle build on both samples forming a Delaunay edge (p_0p_2 in Figure 9) and one of the vertices of the dual to the Voronoi edge p_1p_3 does not contain other Voronoi vertices.

As can be found in [10] the algorithms by Gold and Attali are in fact the same. Bearing in mind this fact and results presented in [6], extension of the one step crust and skeleton extraction algorithm by Gold et al. to 3D should give the same results. This was confirmed by experiment and results can be seen in Figure 10.



a) **b)**

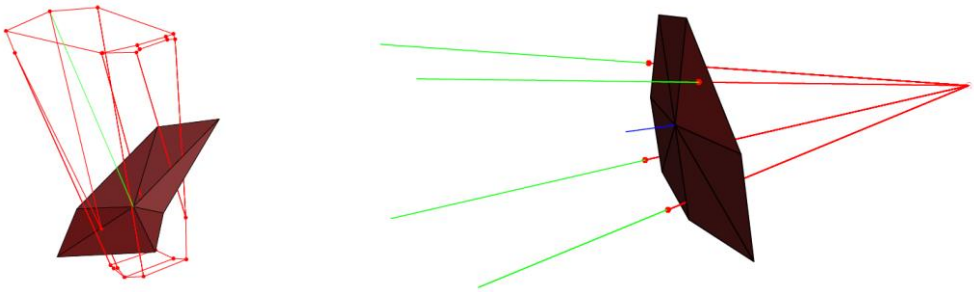
Figure 10: Extension to 3D algorithms by a) Gold b) Attali.

The algorithms fail to robustly extract surface from 3D samples because the Voronoi diagram of samples from the surface of a three dimensional objects does not converge to the medial axis. The amount of holes in surfaces obtained from those two algorithms still makes possible to observe approximate shape of the surface but is not comfortable for a viewer. However both algorithms are very simple and do not use additional memory except for Delaunay triangulation.

The algorithm by Amenta et al. has been extended to 3D [3] and gives reasonable results. However, in the three dimensional case the medial axis is approximate by so called pole vectors which approximate normal to the surface in a given sample point.

The poles are subset of the Voronoi diagram computed by taking the Voronoi cell of each sample and marking the farthest Voronoi vertex of the Voronoi cell as a positive pole (Figure 11a)) in case of

finite Voronoi cell. In the case of an infinite cell, an average direction off all infinite Voronoi edges is computed and considered as the positive pole vector (Figure 11b)). Each sample vertex has also negative pole, which is the farthest Voronoi vertex from the sample, and is placed on the opposite side of the surface than positive pole (see [2,3,4,5,8] for details).



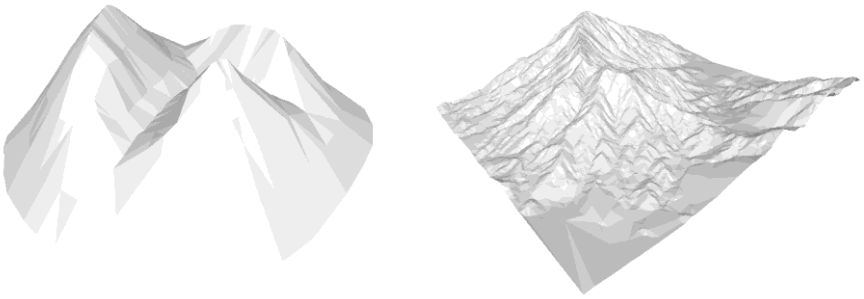
a) b)

Figure 11: Computation of positive pole in case of finite a) and infinite b) Voronoi cell.

Except this simple algorithm many others can be found in literature e.g. [5,8], however we decided to implement algorithm presented by Amenta et al in [3] due to its simplicity and acceptable results reported by authors. The algorithm besides steps taken from the two dimensional case, perform also some post processing. Our implementation does not perform manifold extraction since at this stage we are more interested in the visualization than in a correct surface mesh.

6. RESULTS WE HAVE ACHIEVED SO FAR

Although the final goals have not been achieved yet we have achieved some satisfactory results. So far we worked mostly with simulated point clouds data such as Bunny or simple terrain such as Three Peaks (Figure 12a) and Sha Tin (Figure 12b) terrain model and finally we performed tests on real data from the laser scanner (Tiles, Gate, Bunny).



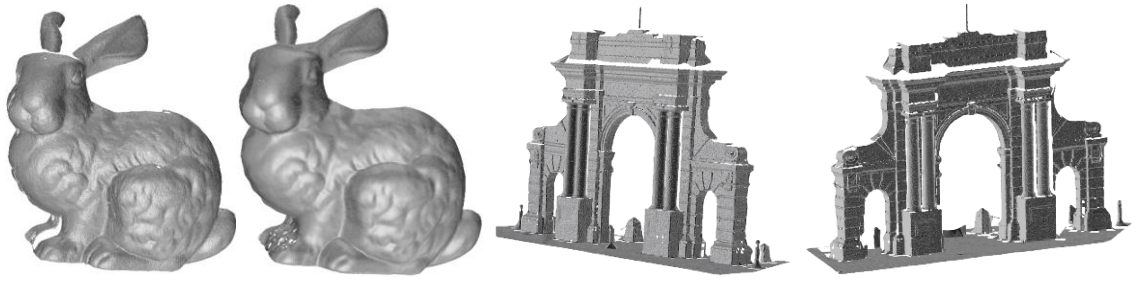
a) b)

Figure 12: Terrain models: (a) Three Peaks, (b) Sha Tin

In the cases of Three Peaks and Sha Tin the data sets were transformed by rotating and/or translating to create the second for merging. For these two, good merging results were easily achieved because

of 100% overlap in the case of Three Peaks and at least 50% in the case of Sha Tin and the fact that each point on one point cloud corresponded exactly to a point on the second one - there was no need to filter correspondence points.

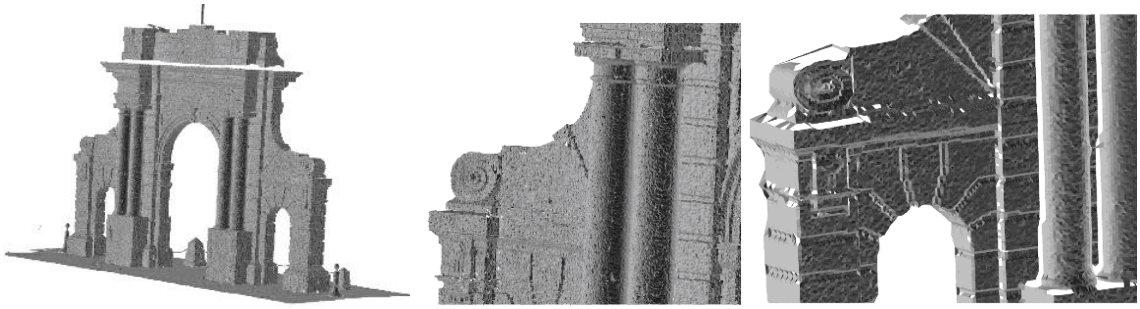
The surface reconstruction algorithm was tested on few datasets. The algorithm gives good results in case of Bunny 000 (Figure 13 a) and b)) and the same model smoothed using natural neighbors interpolation.



a) b) c) d)

Figure 13: Bunny 000 a), Bunny 000 smoothed before reconstruction b). Two scans of Tsing Hua Gate, datasets: c) Gate1, d) Gate2.

Test performed for two single scans of Tsing Hua Gate as well as full model of Tsing Hua Gate aligned by using manually selected control points give also good looking (from some distance) surfaces (Figures 13 c) and d) as well as Figure 14a)), however closer look at the surface shows some amount of noise present in data (Figure 15 b) and c)).

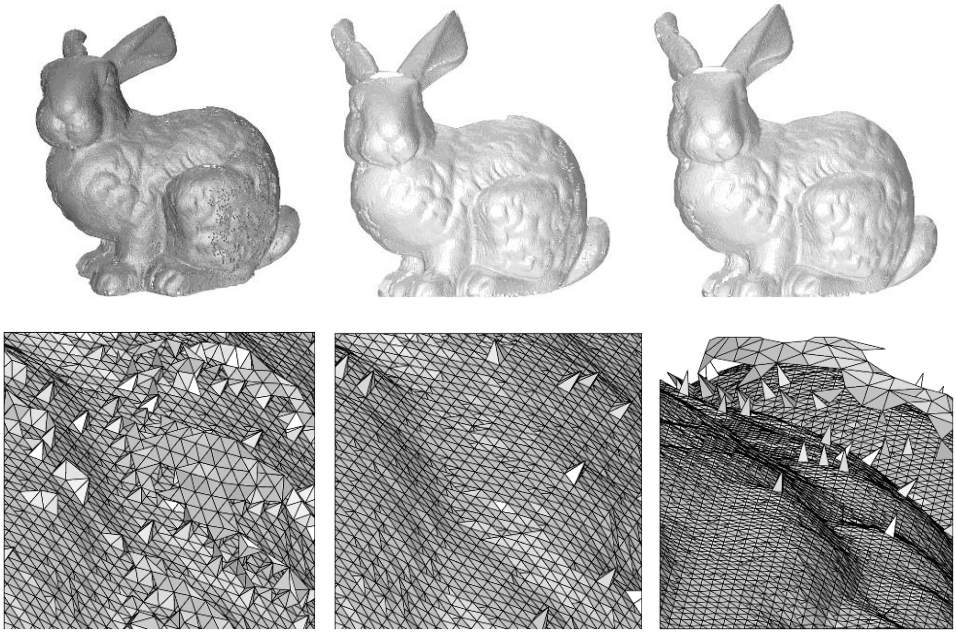


a) b) c)

Figure 14: Reconstruction of the surface of the dataset Gate (a), zoomed reconstruction of the surface of the dataset Gate (b) and Gate2 (c).

Automatically aligned Bunny 000, Bunny 045 and Bunny 315 and merged to single file results in a bit noisy dataset and show weakness of the surface reconstruction algorithm in the presence of noise (Figure 15). To avoid this problem, test using incremental construction of the model was performed. The incremental construction was done by first creating 3D triangulation of Bunny 000 and than inserting Bunny 045 and Bunny 315, but only those samples which are further, from samples already present in the triangulation, than γ, 1.5γ and 2.0γ, where γ is 2D mesh resolution of Bunny 000 (Figure 16).

Figure 15: Surface reconstructed from Bunny 000, Bunny 045, and Bunny 315 brought to common coordinate system and merged together.



a) **b)** **c)**

Figure 16: Bunny 000, Bunny 045 and Bunny 315 brought to a common coordinate system by using control points selected automatically. Model was constructed incrementally using disk size equal to a) γ, b) 1.5γ and c) 2.0γ, where γ is the mesh resolution of Bunny 000.

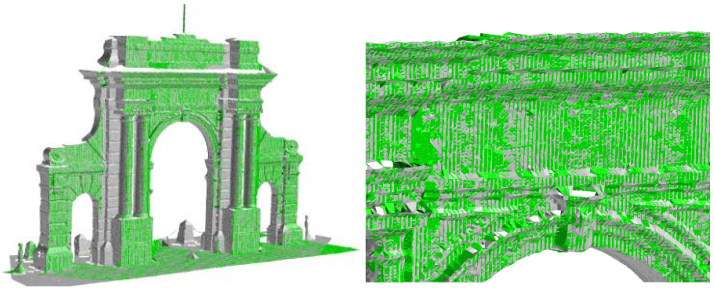


Figure 17: Gate1 (gray) and Gate2 (green) transformed to coordinates system of Gate1 using manually chosen control points.

Dataset	Size (points)	Alignment	Smoothed	Time of [s]		Number of surface triangles
				triangulation	reconstruction	
Gate1	68457	None	No	249	480	207352
Gate2	55745	None	No	207	438	171898
Gate	815693	Manual	No	2154	6360	2527867
Gate1+Gate2	124202	Manual	No	465	1070	387532
Bunny 000	40256	None	No	99	306	109211
Bunny 000	40256	None	Yes	93	319	101188
Bunny 000+045+315	115689	Automatic	No	371	931	345488
Tiles6	42464	None	No	137	291	130195
Tiles6	42464	None	Yes	94	306	128140

Table 1. Triangulation and surface reconstruction times for various models.

7. CONCLUSIONS

We have outlined our approach to large-scale terrain surface modelling and management. The need is clear, and the data capture instrumentation is available. The modelling software, however, remains a problem. Based on our work so far, and incorporating the work of various computing science specialists, pair-wise matching of surfaces is achievable, and overall adjustment of multiple views can be managed – especially as our road-cut application can benefit from some preliminary knowledge of the area. We are still working on the construction of the 3D surface model and the editing of individual views in the database. The surface reconstructed from automatically aligned scans does not give yet rewarding results due to slight misalignment of order 1-2 times of scan resolution (Figure 16). However, as can be seen in Table 1 timings of construction of 3D triangulation and surface reconstruction seem to be reasonable enough to be used in tool we want to create. We believe that the application of modelling techniques derived from computer science can be of great benefit to our terrain management system.

ACKNOWLEDGEMETNS

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 5161/03E) and The Hong Kong Polytechnic University.

REFERENCES

1. Nina Amenta, Marshall Bern, and D. Eppstein (1998). *The crust and the b-skeleton: Combinatorial curve reconstruction*. Graphical models and image processing: GMIP, 60(2):125–.
2. Nina Amenta, Marshall Bern, and Manolis Kamvysselis (1998). *A new Voronoi-based surface reconstruction algorithm.* Computer Graphics, 32(Annual Conference Series):415–421.
3. Nina Amenta and Marshall W. Bern (1998). *Surface reconstruction by Voronoi filtering.* In Symposium on Computational Geometry, pages 39–48.
4. Nina Amenta, Sunghee Choi, Tamal K. Dey, and N. Leekha (2002). *A simple algorithm for homeomorphic surface reconstruction.* International Journal of Computational Geometry and Applications, 12(1-2):125–141.
5. Nina Amenta, Sunghee Choi, and Ravi Kolluri (2001). *The power crust.* Sixth ACM Symposium on Solid Modeling and Applications, pages 249–260.
6. Dominique Attali (1997). *r-regular shape reconstruction from unorganized points* . In Proc. of the 13th ACM Symposium on Computational Geometry, pages 248–253.
7. B.A Dewitt (1996). *Initial approximations for the three dimensional conformal coordinate transformation.* Photogrammetric Engineering and Remote Sensing, 62(1):79–83, 1996.
8. Tamal K. Dey and S. Goswami (2003). *Tight cocone: A water tight surface reconstructor.* Proc. 8th ACM Sympos. Solid Modeling Appl., pages 127–134.
9. Christopher M. Gold (1999). *Crust and anti-crust: A one-step boundary and skeleton extraction algorithm.* In Symposium on Computational Geometry, pages 189–196.
10. Christopher M. Gold and Jack Snoeyink (2001). *A one-step crust and skeleton extraction algorithm.* Algorithmica, 30(2):144–163.
11. Daniel Huber (2002). *Automatic Three-dimensional Modeling from Reality.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
12. Andrew Johnson (1997). *Spin-Images: A Representation for 3-D Surface Matching.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
13. Andrew Johnson and Martial Hebert (1999). *Using spin images for efficient object recognition in cluttered 3d scenes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5):433 – 449.