Pertanika J. Sci. & Technol. 28 (3): 937 - 950 (2020)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Construction of 3D-Terrain Model from Contour Lines using Parameterized B-Spline ruled Surface

Muhammad Ammad and Ahmad Ramli*

School of Mathematical Sciences, Universiti Sains Malaysia, 11800, Penang, Malaysia

ABSTRACT

In this paper work, three-dimensional terrain models were reconstructed from two-dimensional contour lines. Firstly, spatial curves were generated from contour lines using parameterized cubic B-spline curve interpolation. Then surfaces were constructed by using B-spline ruled surface. In the reconstruction process, some issues such as keyholes and branching may arise. Therefore, we propose a method that handles the branching object to construct a bilinear patch by following the proposed data point's extraction algorithm. We also solved keyholes issues by retaining the same knot vector condition on B-spline ruled surface. Results are also demonstrated for models with branching and without branching.

Keywords: 3D-terrain model, B-spline, contour lines, parameterization method, surface reconstruction

INTRODUCTION

Construction of 3D surface from 2D contour is very important for rapid prototyping and NC Machining in Computer Aided Design (CAD), medical imaging such as Computed Tomography (CT) and Geographical Information System (GIS) in order to be visualized for further analysis and planning. Numerous methods have been proposed in reconstructing of surfaces because it involves lots of area such as manufacturing, engineering, medical and geoinformatic. In surface reconstruction, type of data will determine the approach that is going to be used. In topographic map, shape of earth surface is represented by contour lines which is a distinguishing feature of a topographic map. Contour lines are imaginary lines connecting point of equal elevation to show the shape of terrain and topography of the landscape.

Different techniques on surface reconstruction from 2D contour have been addressed in the past but some complexities

ARTICLE INFO

Article history:
Received: 06 January 2020
Accepted: 30 March 2020
Published: 16 July 2020

E-mail addresses:
mammad343@yahoo.com (Muhammad Ammad)
alaramli@usm.my (Ahmad Ramli)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

are common like continuity, planarity, rapid changing, multifurcation, and keyholes (Meyers et al., 1992). Keyholes actually cause a problem with triangulation process which is due to the irregular placement of point on contour (Sunderland et al., 2015) and surface continuity can be achieved by applying contour interpolation method such as performing Hermite interpolation along the gradient paths (Hormann et al., 2003).

For branching object, fast reconstruction method by Shin and Jung (2004) is implemented to generate original geometry by connecting the vertices with edges between two consecutive slices. Straight skeleton method is used to create faces on key contour by linking the key contour lines in GIS map for generating 3D terrain (Salvatore & Guitton, 2004; Sugihara & Murase, 2017).

The complexity of many geometric operations greatly depends on the method of representation. Some information related to most commonly method of representing surfaces such as implicit (Guo et al., 2010; De Araújo et al., 2015; Akenine et al., 2018), parameterized B-spline and NURBS (Lim & Haron, 2014; Zhang 2016; Bhattarai et al., 2017) are also taken into account. The parameterized form is more natural for designing and representing shape in the computer as compared to implicit (Brigger et al., 2000). The construction of surface using parameterized B-spline and radial basis function from scattered data points was discussed by Jie et al. (2016). Tensor product of B-spline interpolation can also be used as a modern acquisition technique for reconstruction of surfaces (Vaitkus & Várady, 2018). As a parametric function B-spline possesses considerable geometric significance for constructing a ruled surface, such constructions are fundamental to many CAD systems (Jha, 2008). One basic problem in the study of the parametric curve is to approximate a curve with lower degree curve segment (Amirfakhrian, 2012). However, some geometric feature such as singular point cannot be preserved. In this study our aim was to reconstruct smooth surface. So we used parameterized B-spline curve interpolation to recreate contour lines. Then by using B-spline ruled surface, we constructed surfaces on each contour. The issues related to branching object was resolved by creating a bilinear patch. In order to construct bilinear patch to fill the space between branching objects, we also proposed a data extraction algorithm. The results and comparison of our method are also discussed in detail.

The organization of the paper is as follows. Section 2 discussed previous methods related to terrain model, followed by Section 3 where definition of B-spline are presented. In Section 4 methodology and computation of proposed method is stated. In Section 5 and 6 continuity constraints and the results are discussed respectively. Finally, conclusion and future work are presented in Section 7.

LITERATURE REVIEW

Straight skeleton method by Aichholzer et al. (1995) is defined as an appropriate shrinking process for the polygon. In contour map, altitude is an important element especially for the hilly area. As altitude increases, the map of contour line is redefined. After retreating the map, if contour lines cross itself at a certain height, then one possible way to manage contour lines or polygon is by the process of division of polygon. Contour polygon is divided into two or more polygons. By this approach the polygon will change topologically during the retreating process and gradually it will reduce the size of the polygon. In this case, this method is quite useful for splitting the edge or shrinking edges to zero, making it neighboring edges adjacent. This shrinking process gives a hierarchy of nested polygons and from such polygons B-spline curves are formed, which further examine the processing of 3D terrain model as a last step. This method takes some time to reconstruct the object.

In key contour polygon (Sugihara & Murase, 2017), to create a 3D terrain model, a physical simulation is performed which uses a more realistic environment such as town model based on the 3D terrain model. Some modifications such as land formation and site preparation require construction of 3D model on these sites. This is mostly done manually by labor and it takes enormous time but in the case of key contour polygon with given elevation, in order to automate laborious step the faces connecting contour are automatically formed by using the geographic information system and CG integrated system followed by straight skeleton method and thus automatically 3D terrain model is created.

Fast reconstruction (Shin & Jung, 2004) is also an automatic reconstruction method for constructing a 3D terrain model from a 2D contour line, which does not take into account matching region and clefts due to the degraded performance and long processing time. This method is totally based on triangle strips that are generated from a single tiling operation for a simple region that does not contain branches, but it also be useful for branching. If there are branches, then contour lines are converted into several sub-contour as a partition. By considering a number of vertices and their spatial distribution the geometry of the contour line will be reconstructed by connecting the vertices with edges on adjacent contour slices or lines. In this way, in less time a most realistic model can be formed by using this fast reconstruction method. However this method may be favorable for branching surfaces and is highly rated over other methods due to its quick execution. The algorithm used in this study has restriction that the center of gravity of a contour line should be located on its interior region. Therefore unnatural structure can be observed for some models.

DEFINITIONS

The B-spline curve of degree d (order d+ 1) with control points $P_0, P_1, \ldots, P_n$ and knot values $t_0, t_1, \ldots, t_m$ is defined on the interval [a, b] = [t_d, t_{m-d}] by Equation 1

$$B(t) = \sum_{i=0}^{n} P_i N_{i,d}(t) \qquad [1]$$

where $N_{i,d}(t)$ is the basis function of degree *d*. The underlying core of B-spline is its basis function.

B-spline basis function of degree *d*, defined by knot vector $t_0, t_1, \ldots, t_m$ are defined recursively as Equation 2 and 3

$$N_{i,0}(t) = \begin{cases} 1, & if\ t \in [t_i, t_{i+1}) \\ 0, & otherwise \end{cases} \qquad [2]$$

$$N_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} N_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+1,d-1}(t) \qquad [3]$$

for $i=0, \ldots, n$ and $d \geq 1$.

If the list of data points $Q_i, i \in [0, \mathrm{n}]$ are given, then the B-Spline curve interpolation of order *k* is to find the knot vector $T = (T_0, T_1, \ldots, T_{n+k-1}, T_{n+k})$, the parametric values t_i, for each $Q_i, i \in [0, \mathrm{n}]$ and the control points P_i such that the resulting B-spline curve $B(t)$ has the property as in Equation 4

$$B(t_i) = Q_i, i \in [0, n] \qquad [4]$$

B-spline surface is an extension of B-spline curve. The most common kind of B-spline surface is the tensor product surface. Let $N_{i,d}(s)$ be the B-spline basis function of degree *d* with knot vector $s_0, s_1, \ldots, s_m$ and let $N_{j,e}(t)$ be the B-spline basis function of degree *e* with knot vector $t_0, t_1, \ldots, t_q$. A mathematical description of tensor product of B-spline surface with control points $P_{i,j}$ $(0 \leq i \leq n = m - d - 1, 0 \leq j \leq p = q - e - 1)$ is defined by Equation 5

$$S(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{p} P_{i,j} N_{i,d}(s) N_{j,e}(t) \qquad [5]$$

for $(s,t) \in [s_d, s_{m-d}] \times [t_e, t_{q-e}]$.

A B-spline ruled surface is formed from two spatial curves *B(s)* and *C(s)* when point on each curve corresponding to the parameters are joined by a line. Consider two B-spline curves $B(s) = \sum_{i=0}^{n} b_i N_{i,d}(s)$ and $C(s) = \sum_{j=0}^{n} c_j N_{j,d}(s)$. The curves are assumed to have the same degree and to be defined on the same knot vector $s_0, s_1, \ldots, s_m$. The constructed B-spline ruled surface, linear in *t-direction*, is given by Equation 6

$$S(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{1} P_{i,j} N_{i,d}(s) N_{j,1}(t) \qquad [6]$$

the surface has knot vector $s_0, s_1, \ldots, s_m$ in *s-direction* and *0,0,1,1* in *t-direction.*

A digital terrain model (DTM) is a digital representation of ground surface landform using a set of height over 2D points residing on a reference surface to produce topographic map. A common mathematical representation for DTMs includes regularly spaced grid (2D raster or Matrix form), irregularly distributed 2D points (variable point distance), contour and Fourier series.

METHODS

Computation of B-Spline Ruled Surface

A 3D terrain model was constructed from 2D contour by using B-spline ruled surface. Firstly, the contour line needed to be extracted as a set of points. The data points on contour lines were interpolated by using parameterized method on cubic B-spline curve (Ammad & Ramli, 2019).

The next step was to create a surface from one slice of the contour line which was represented by a B-spline curve, to its adjacent slice using B-spline ruled surface by fulfilling all necessary and required condition. In this way, all required segment of surfaces were created between adjacent slices. The problem arising here was the closure of surface, because surface constructed by B-spline ruled surface remained open in *t-direction* at the top in fact. In *s-direction* periodic B-spline was used to reconstruct closed curves but it was much difficult to use periodic curves in *t-direction*. The idea used was to extend the surface to a single vertex (summit vertex) on top of the last contour. Another solution could be to cap the open surface with another surface but some joining problem may arise (Jaillet et al., 2001). When we chose that vertex, then the B-spline curve focused to a single point was created and the control points were summit vertex with required multiplicity. So, we generated an extra segment of surface which produced a more realistic look of 3D-terrain model. In the final step, all portions were connected into a single form which was the 3D terrain model from 2D contour line.

The computation of the B-spline ruled surface is not trivial. Special attention is required related to parameters, knots and degree of B-spline curve while constructing a ruled surface. Assume two spatial B-spline curves (Equation 7 & 8]

$$B(s) = \sum_{i=0}^{n} b_i N_{i,d}(s) \qquad [7]$$

and

$$C(s) = \sum_{j=0}^{n} c_j N_{j,d}(s) \qquad [8]$$

are defined over the same knots $s_0, s_1, \ldots, s_m$ and have the same degree d with control points b_i and c_j. B-spline surfaces are the tensor product in nature, so it is required that the two boundary curves to have the same degree and defined over the same knot values. We want to construct a ruled surface by using both curves, which is ruled in *t-direction* and there is a linear combination between these two curves. Furthermore in B-spline ruled surface, the interpolation is between the points of equal parameter values, not point of equal arc length. Because ruling according to parameter value is achieved by B-spline but not achieved by arc length due to the geometrically different surface, the constructed B-spline surface from same degree curves, defined on same knot vector, linear in *t-direction* is given by Equation 9

$$S(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{1} P_{i,j} N_{i,d}(s) N_{j,1}(t) \qquad [9]$$

Before implementing Equation [9] the following conditions must be fulfilled for both curves with degree d_1 and d_2 respectively.

1. Parameter range should be the same for both curves.
2. If $d_1 = d_2$ then set $d = d_1$, If $d_1 \neq d_2$ then set $d = \max\{d_1, d_2\}$, and raise the lower degree curve to d by using degree rising algorithm.
3. If knot vector of both curves are different then use knot insertion algorithm or knot refinement algorithm to obtain identical knot vector.

Branching Object

It is always a difficult task to deal with branching surfaces. We cannot reconstruct branching surfaces with general B-spline ruled surface. However, we present a method to handle this case. In our solution, we splitted the body contour into two parts (Left and Right) at common points, because for simplifying the branching process we needed to use patches to obtain the satisfying result of the branching process. By splitting the body contour into left and the right part we got two patches, one from the left part of body contour connected to the curve b1, that interpolated left branching contour and second from the right part of body contour connected to the curve b2, that interpolated right branching contour as shown in Figure 1. These two patches make the joint between the body and the branches. The center patch fills the space between the left and right patch to provide a smooth surface that lies on existing contours.

In order to generalize our work, we proposed an Algorithm 1 to numerically find the required data points in constructing the bilinear patch. Given a list of data points p_i, q_j, and r_k of left branch, right branch and body contour respectively, then the algorithm extract the required points *a, b*, R_k and R_l, where *a* is the point of left branching contour, *b* on right branching contour, while points R_k and R_l are on body contour. The data points are shown in Figure 2 (a). The algorithm work in three steps. In first step, the user get the

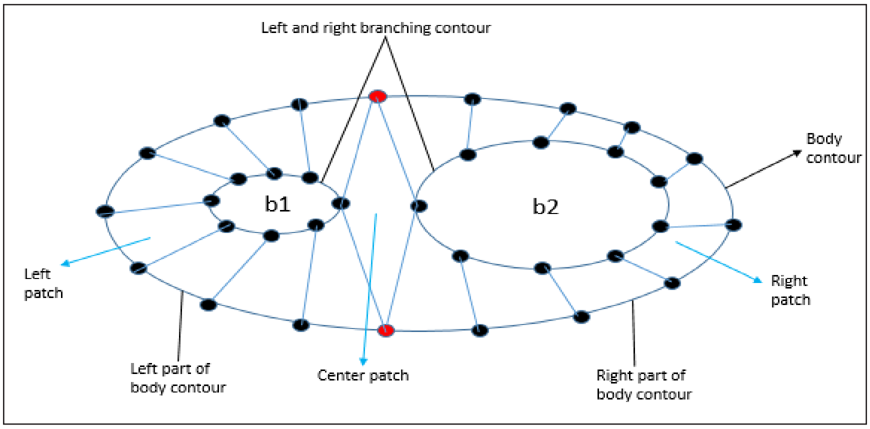


Figure 1. Skeleton of joining branches with trunk

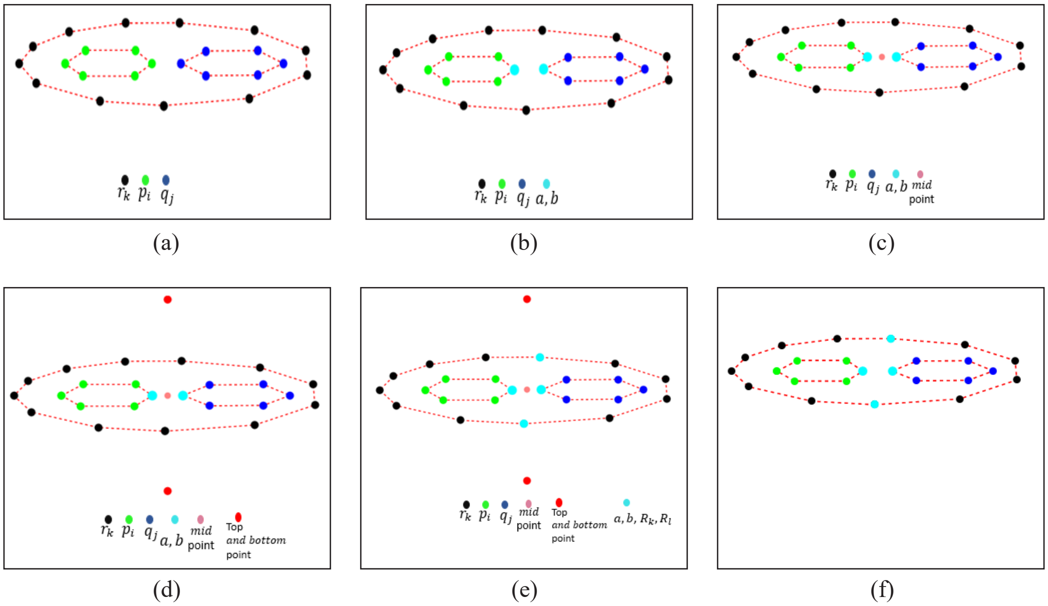


Figure 2. Step wise extraction of points for bilinear patch using Algorithm 1 (a) representation of data points. (b) Implementation of Step-1: branching points with minimum distance is marked by a and b, shown as cyan points. (c) Midpoint of a and b in marked by a pink point. (d) Step-2: top and bottom points shown as red points. (e) Step-3: Cyan points on body contour have minimum distance from top and bottom point, represented as R_k, R_l. (f) Extracted data points are represented by cyan points.

pair of points that have minimum distance, resulted *a* and *b* points as shown in Figure 2 (b). Then the midpoint of these two points is shown in Figure 2 (c). In second step the algorithm find the maximum distance of two points in body contour and compute the top (and bottom) point by adding (subtracting) maximum distance, $\boldsymbol{\alpha}$ to midpoint. The top and bottom points are shown in Figure 2 (d). In the final step, the algorithm search the points that have minimum distance from top and bottom points to all points on body contour and gives the points R_k and R_l as shown in Figure 2 (e). So the final extracted points are given in cyan color shown in Figure 2 (f).

Algorithm 1: Algorithm for finding data points for bilinear patch

```
Input:
p_i =(x_i, y_i, z_i), i= 1,2,...,length(p)
q_j =(x_j, y_j, z_j), j= 1,2,...,length(q)
r_k =(x_k, y_k, z_k), k= 1,2,...,length(r)
Result: {a, b, R_k, R_l}
Step 1- Initialization;
    While i=1 to length(p), j=1 to length(q) do
        If dist(p_i, q_j) ≤ minDist(p_i, q_j) then
            closestpair : (a=p_i, b=q_j);
            Return: a, b;
        end
    end
    Midpoint= mid(a, b)
Step 2- Initialization;
    Let R_1=r_k , R_2=r_l
    While k=1 to length(r)-1, l=k+1 to length(r) do
        If dist(R_1 ,R_2) ≥ maxDist(R_1, R_2) then
            α= maxDist (R_1, R_2);
            Return: α ;
        end
    end
    top=Midpoint+ α
    bottom= Midpoint- α
Step 3- Initialization;
    Let R_k=r_k , R_l=r_l
    While k=1 to length(r), l=1 to length(r) do
        If dist(top, R_k) ≤ minDist(top, R_k), dist(bottom, R_l) ≤ minDist(bottom, R_l) then
            topclosestpoint: R_k;
            bottomclosestpoint: R_l;
            Return: R_k, R_l ;
        end
    end
```

Creating a Bilinear Patch

We used B-spline ruled surface to create the left surface patch and right surface patch shown in Figure 3. Note that for this case with 4 points, the only possibility to fill the space (center patch) between these patches is to fill with a bilinear patch.

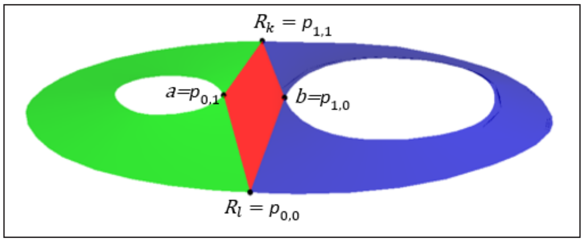


Figure 3. A closed view joining patches with bilinear patch

Given four points, $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, $P_{1,1}$ then the ruled surface defined by the line segment joining $P_{0,0}$ and $P_{0,1}$ and the line segment joining $P_{1,0}$ and $P_{1,1}$ is a bilinear surface shown in Figure 1. In B-spline form the surface is as in Equation 10

$$S(s,t) = \sum_{i=0}^{1} \sum_{j=0}^{1} P_{i,j} N_{i,1}(s) N_{j,1}(t) \qquad [10]$$

with knot vector *0,0,1,1* in both directions. We can also obtain the bilinear surface in Bezier form by replacing the B-spline basis functions by the linear Bernstein basis functions.

So the center patch which was a bilinear surface patch is created by using Equation [10] and the remaining space is filled by this patch. By joining all the three patches we got a smooth branching surface and the final result is shown in Figure 3.

Continuity

In the construction of 3D-terrain model by using B-spline ruled surface, one should also check the continuity constraints in order to join all patches seamlessly. In achieving G^0 continuity between all patches, most scheme set the same degree and knot vector for all patches of B-spline (Milroy et al., 1995). By doing this, all adjacent patches share their control points along the boundaries. So, G^0 continuity is achieved trivially. In our study, in order to check G^0 continuity we also followed the theorem stated as:

Let $B(u,t)$ and $C(v,t)$ be two B-spline surfaces defined as:

$$B(u,t) = \sum_{i=0}^{m} \sum_{j=0}^{n} b_{i,j} N_{j,p}(u) N_{i,q}(t)$$

$$C(v,t) = \sum_{i=0}^{m}\sum_{j=0}^{g} c_{i,j} N_{j,r}(v) N_{i,q}(t)$$

where $b_{i,j}$, $c_{i,j}$ are the control points in E^3, and the B-spline basis functions $N_{j,p}\ (u)$ with degree p, $N_{i,q}\ (t)$ with degree q, $N_{j,r}\ (v)$ with degree r are defined on the non-periodic knot vector U, V and T.

The sufficient and necessary conditions of G^0 continuity between $B(u,t)$ and $C(v,t)$ is:

$$b_{i,0} = c_{i,0}$$

then $B(0,t) = C(0,t)$, namely $B(u,t)$ and $C(v,t)$ are G^0 continuous with common boundary curve $C(t)$.

However in branching model, when we attached the bilinear patch to fill the space, it also shared the boundary control points between neighboring patches which guaranteed the G^0 continuity.

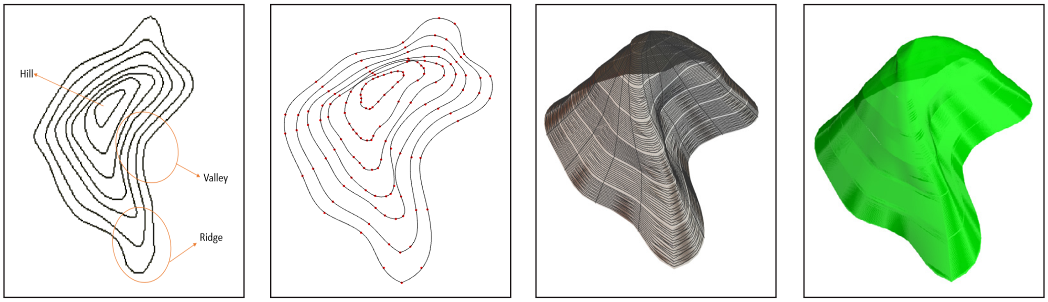
RESULTS AND DISCUSSIONS

The results of three different reconstructed 3D-terrain model from 2D-contour lines have been discussed in this section. These models are directly reconstructed from interpolated contour line shown in Figure 4 (b) using B-spline ruled surface and the work is done with programming software Mathematica V.10.2. For the quality of our automatic reconstruction method, we also produced results of the complex object (e.g branching model) along with a simple matching region. In branching object, each branch was reconstructed independently and then the bilinear patch was used to make joint and fill the space between trunk and branches.

Model-1

This is one type of non-branching model and the contour lines of this model include valley, ridges and hill. The boundaries are also sprawling at some points which indicate the ridges position as shown in Figure 4 (a).

The reconstruction process of such type of model having valley and ridges is not an easy task. The user can face discontinuity between adjacent patches of B-spline ruled surface if the selected position of interpolated points is more in inside position or more in outside position. The wrong selected position of interpolated points especially at valley position can also create an edge at joining position of adjacent patches. The reconstructed model with and without meshes is shown in Figure 4 (c) and (d) respectively.



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

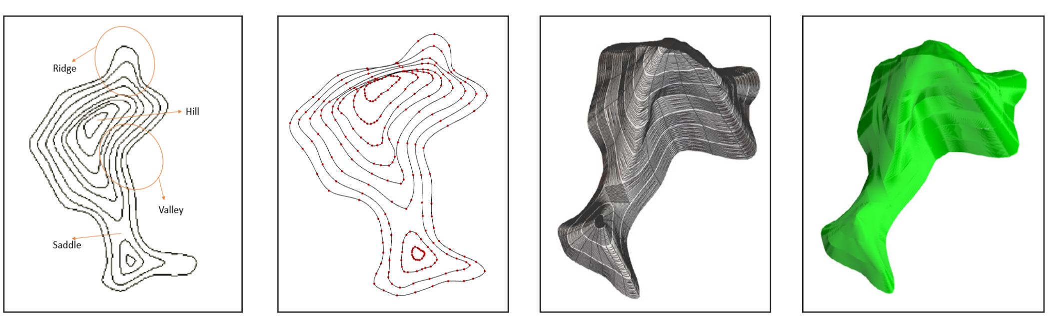
Figure 4. Reconstructed Model-1 by B-spline ruled surface; (a) contour lines; (b) Recreated contour lines using B-spline curve interpolation; (c) Aligned model with meshes; and (d) shaded image without meshes

Model-2

This is one type of a branching model and contour lines of this model include valley, ridges, hill and saddle. Most of the researchers call branching model a complex model, because the reconstruction process of branching model is much difficult and it produces many error such as key holes and continuity problem at the region of trunk joining with branches. The result obtained by applying B-spline ruled surface with necessary and sufficient condition and after creating a bilinear patch to join branching and trunk with G^0 continuity between adjacent patches is shown in Figure 5 (c and d) with and without mesh respectively. In Figure 5 the branching part joining with trunk indicates the saddle and we can clearly see through the shaded image that our technique produces good result also for saddle region.

Model-3

This is also a non-branching model of knife edged hill that consists of 7 contour lines and the top contour line indicates a knife edge hill. The elevation change is not quick between contour lines that shows a shallow slope. The reconstruction process of these type of models

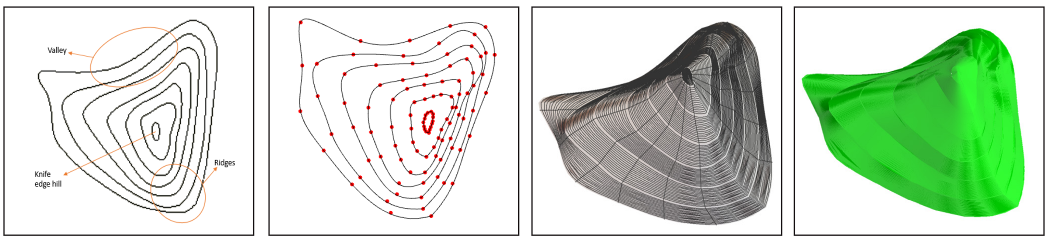


(a) (b) (c) (d)

Figure 5. Reconstructed Model-2 by B-spline ruled surface: (a) contour lines; (b) Recreated contour lines using B-spline curve interpolation; (c) Aligned model with meshes; and (d) shaded image without meshes

are very easy and possibilities of any type of errors are very rare. The resulting graph of model-3 is shown in Figure 6.

However, our method performs as well as Shin and Jung (2004) while using a different approach and a different process of dealing with branching object along with simple region. Moreover, Shin and Jung (2004) used the concept that the center of gravity of contour line should be in its interior region which might produce unnatural results for some models. But in our case there is no restriction in computation of B-spline ruled surfaces for reconstructing the surfaces.



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

Figure 6. Reconstructed Model-3 by B-spline ruled surface: (a) contour lines; (b) Recreated contour lines using B-spline curve interpolation; (c) Aligned model with meshes; and (d) shaded image without meshes

CONCLUSION

In this paper we used B-spline ruled surface to reconstruct the three-dimensional terrain model from two-dimensional contour lines. Our method started with the recreation of contour lines by using parameterized B-spline curve interpolation and ended up with the construction of ruled surface patches on each contour with G^0 continuity between all patches. We applied our method to both simple and complex region (branching object) and our method produced similar results to previous methods while using a different technique without the restriction that the location of the core of gravity of the contour line should be in its interior region, which could produce unnatural structure for some designs. The one drawback of the proposed method is that in branching object, the maximum achievable continuity between bilinear patch with its neighboring patches is G^0 but in the simple region, it can be higher than G^0. However, in future work, this method may be improved to reconstruct more complex objects such as saddle and ridges with higher continuity. Also, we can look into the connectivity between patches to preserve G^1 continuity.

ACKNOWLEDGEMENT

This work was supported by Universiti Sains Malaysia under RUI grant no. 1001/PMATHS/8011014.

REFERENCES

Aichholzer, O., Aurenhammer, F., Alberts, D., & Gärtner, B. (1995). A novel type of skeleton for polygons. *Journal of Universal Computer Science, 1*(12), 752-761.

Akenine-Mo, T., Haines, E., & Hoffman, N. (2018). *Real-time rendering*. New York, NY: AK Peters/CRC Press.

Amirfakhrian, M. (2012). Approximation of 3D-parametric functions by bicubic B-spline functions. *International Journal of Mathematical Modelling and Computations, 2*(3), 211-220.

Ammad, M., & Ramli, A. (2019, July 16-19). Cubic B-Spline curve interpolation with arbitrary derivatives on its data points. In *2019 23rd International Conference in Information Visualization–Part II* (pp. 156-159). Adelaide, Australia.

Bhattarai, S., Vichare, P., & Dahal, K. (2017, October 25-27). Performance comparison of adapted Delaunay triangulation method over NURBS for surface optimization problems. In *31st Annual European Simulation and Modelling Conference, ESM 2017* (pp. 76-80). Lisbon, Portugal.

Brigger, P., Hoeg, J., & Unser, M. (2000). B-spline snakes: A flexible tool for parametric contour detection. *IEEE Transactions on Image Processing, 9*(9), 1484-1496.

De Araújo, B. R., Lopes, D. S., Jepp, P., Jorge, J. A., & Wyvill, B. (2015). A survey on implicit surface polygonization. *ACM Computing Surveys (CSUR), 47*(4), 1-39.

Guo, G., Wu, X., Wang, M. Y., & Wu, J. (2010, August 4-7). Fast implicit surface reconstruction method based on normal constraints. In *2010 IEEE International Conference on Mechatronics and Automation* (pp. 1783-1788). Xi'an, China.

Hormann, K., Spinello, S., & Schröder, P. (2003, November 19-21). C1-continuous terrain reconstruction from sparse contours. In *Proceedings of the Vision, Modeling, and Visualization Conference 2003 (VMV 2003)* (Vol. 3, pp. 289-297). München, Germany.

Jaillet, F., Shariat, B., & Vandorpe, D. (2001). Smooth reconstruction of branching objects with B-splines. *Revue internationale de CFAO et d'in-formatique graphique, 16*(2), 183-201.

Jha, K. (2008). Construction of branching surface from 2-D contours. *International Journal of CAD/CAD, 8*, 1-15.

Jie, L. K., Ramli, A., & Majid, A. A. (2016). A comparison study between b-spline surface fitting and radial basis function surface fitting on scattered points. *Jurnal Teknologi, 78*(6-5), 103-107.

Lim, S. P., & Haron, H. (2014). Surface reconstruction techniques: a review. *Artificial Intelligence Review, 42*(1), 59-78.

Meyers, D., Skinner, S., & Sloan, K. (1992). Surfaces from contours. *ACM Transactions on Graphics (TOG), 11*(3), 228-258.

Milroy, M. J., Bradley, C., Vickers, G. W., & Weir, D. J. (1995). G1 continuity of B-spline surface patches in reverse engineering. *Computer-Aided Design, 27*(6), 471-478.

Salvatore, S., & Guitton, P. (2004). Contour line recognition from scanned topographic maps. *Journal of Winter School of Computer Graphics, 12*(1-3), 419-426.

Shin, B. S., & Jung, H. S. (2004). Fast reconstruction of 3d terrain model from contour lines on 2d maps. In *Asian Simulation Conference* (pp. 230-239). Heidelberg, Germany: Springer.

Sugihara, K., & Murase, T. (2017, September 20-22). Automatic generation of a 3D terrain model from key contours. In *2017 International Conference on Cyberworlds (CW)* (pp. 111-117). Chester, UK.

Sunderland, K., Woo, B., Pinter, C., & Fichtinger, G. (2015, March 18). Reconstruction of surfaces from planar contours through contour interpolation. In *Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling* (Vol. 9415, p. 94151R). Florida, United States.

Vaitkus, M., & Várady, T. (2018). Parameterizing and extending trimmed regions for tensor-product surface fitting. *Computer-Aided Design, 104*, 125-140.

Zhang, Y. J. (2016). *Geometric modeling and mesh generation from scanned images* (Vol. 6). Boca Raton, Florida: CRC Press.