Interpolating Scattered Data Points by Catmull-Clark Subdivision Surfaces

Abdulwahed Abbas[1] and Ahmad Nasri[2] *

[1]The University of Balamand, abbas@balamand.edu.lb
[2]American University of Beirut, anasri@aub.edu.lb

ABSTRACT

This paper outlines an approach for the interpolation of scattered data points by Catmull-Clark subdivision surfaces. This approach relies in a fundamental way on an algorithm that groups the initial data points into a sequence of control polygons. Each one of these polygons is transformed in such a way that its corresponding limit curve interpolates its original control points. A polygonal complex is then constructed to embody each one of these polygons, which is then transformed to correspond to the original polygon it was embodying. Finally, automatic connection of each complex to its immediate neighbors is established. This process results in a control polyhedron whose limit surface interpolates the initial control points. This process is straightforward and easy to implement. While not utilizing too many auxiliary data points, these additional points can be used to tailor the interpolating surface to suit additional user requirements, without affecting the initial interpolation constraints.

Keywords: Catmull-Clark Subdivision Surfaces, Polygonal Complexes, Skinning, Active Contours, Interpolation.

1. INTRODUCTION

Numerical data in the form of 3D points usually arises in continuous domains both in science and in technology; e.g. engineering, medicine, earth science and meteorology. This data is usually collected in a fashion that cannot predictably be in any specific order. In this context, constructing a surface that interpolates these data points is motivated by the need to organize such data in a manner more meaningful to the goals it is collected for.

When constructing a continuous surface interpolating a random collection of points in the 3D space, one has the following two options: interpolation or approximation. While the first option produces an exact result (in the sense that it does not except any of the input data), the latter could be more appropriate in the sense that it will leave out noisy input, and thus produces a better result.

Whether to opt for one of the above options or the other largely depends on the expected input and on the expected use the result is going to be put for. In this paper, we choose to address the interpolation problem. Thus, although not known in advance, the sets of data points treated as input are expected to belong to a coherent surface admitting not too complex a representation. We follow this line even though an initial idea of what this surface is like should provide pointers for improving the quality of the result.

The research that has been done on the interpolation of scattered data points is already extensive [11, 12]. Some of the reported methods can be classified as global methods, because the construction of the interpolating surface requires the evaluation of a function over all the given data points. Understandably, global methods tend to be expensive when applied to larger sets of input data. Therefore more localized methods tend to fare better in this respect. The reader is referred to [7] for an in-depth review of the state of the art of the subject up to late 90's.

* Correspondence should be addressed to: Ahmad Nasri, American University of Beirut, Department of Computer Science, P.O.Box 11-236, Beirut , Lebanon, . Phone: ++961 1 350 000 ext. 4223/4224, Fax: ++961 1 744461.

This paper outlines an approach for automatically constructing a subdivision surface interpolating an initially unorganized set of data points in the 3D space. The only assumption required by this approach is that the data points come (or intended to be) from an (initially unknown) surface of reasonable smoothness. Other than that, the approach is quite straightforward, easy to understand and implement. While not utilizing too many auxiliary data points, these additional points can be used to tailor the interpolating surface to suit additional user requirements, without affecting the initial interpolation constraints. Furthermore, the amount of computation required by this method is reasonable when measured in terms of time as well as space metrics.

The major challenge that this task faces is that the set of data points is not initially given in any meaningful order. To address this difficulty, we follow the main direction taken in [5]. We also adopt the basic terminology used there, although the finer details and the ultimate goals of the approach reported in this paper are radically different.

2. SUBDIVISION AND POLYGONAL COMPLEXES

Before going into the details of the approach, we start by presenting some background information necessary to make this paper somewhat self-contained.

2.1. The Catmull-Clark Subdivision Scheme

We use Catmull-Clark subdivision [4] to illustrate the interpolation process. Catmull-Clark surfaces are a generalization of tensor product cubic B-splines. According to this scheme, an initial control polyhedron is subdivided into another control polyhedron as depicted in figures 1(a) and 1(b).

At the end of this process, each F-vertex is connected to the adjacent E-vertices and each E-vertex is connected to the adjacent V-vertices. The resulting faces will form the new subdivided polyhedron. In this context, note that repeated application of this subdivision process will in general lead to more faces and smaller edges. At the limit, this will converge to a smooth surface.

Initial polyhedron	Subdivided polyhedron	Subdivision Rules
face f	F-vertex v_f	average of vertices of f
inner edge *e*	E-vertex v_e	average of vertices of *e* together with the F-vertices of adjacent faces of *e*
inner vertex v	V-vertex	((n-2)*v + (R + S)/n)/n where n is the number of faces adjacent to v R = Sum ({v_i; i = 1 .. n}), where vv_i is an edge S = Sum ({v_{fi}; i = 1 .. n}), where v_{fi} is an F-vertex of a face fi containing v

Fig. 1(a). The Catmull-Clark Subdivision Scheme:
The rules for generating the various type of faces in a refined polyhedron

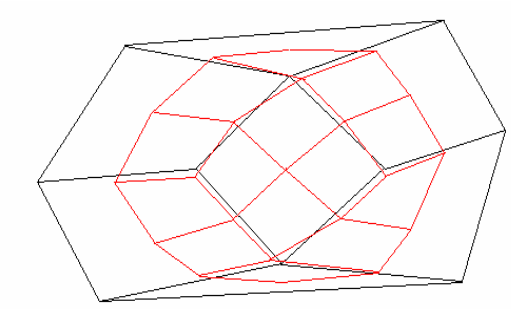

Fig. 1(b). The Camull-Clark Subdivision Scheme:
A red polyhedron resulting from a black polyhedron in one subdivision step

According to this formulation, the border edges and vertices do not contribute any new vertices. Therefore, these vertices and edges are kept away from the limit surface. However, these vertices and edges are sometimes incorporated into the main subdivision routine as special cases.

2.2. Polygonal Complexes and Their Limit Curves

In the context of any given subdivision scheme (S), a polygonal complex is thought of as a polygonal structure whose limit under (S) is just a curve (see figure 2).

In the context of Catmull-Clark subdivision, a general polygonal complex is simply a polygon (M) set as the intersection of a pair of sequences of faces (T) and (B) both of the same length as (M). It is important to note here that each inner vertex of this complex is regular in the sense that it connects exactly four edges of the surrounding faces. However, these faces do not have to be rectangular at their outer edges.

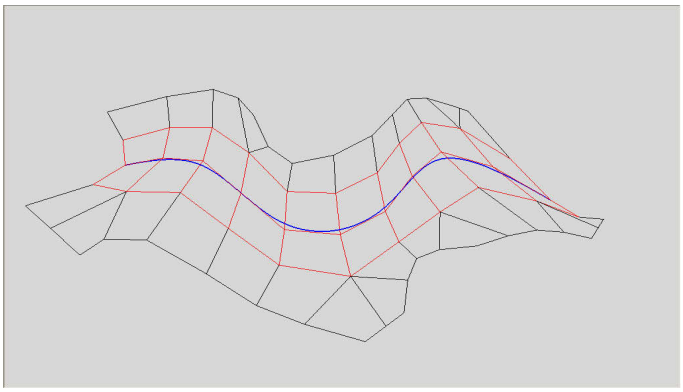

Fig. 2. A Subdivided General Polygonal Complex and its Limit Curve

Note here that a general complex can be transformed into a simple one after be performing a single subdivision step. Here, a polygonal complex is simple when all its constituent faces are rectangular. This way, the complex admits a representation of three rows of vertices (ti), (mi) and (bi), all of the same length. This makes the complex fits very nicely into a 3×n matrix M of vertices.

In this context, the limit of a simple complex M is a B-spline curve whose control polygon P is given by the following formula (see [8] and [9]):

$$(1/6)*[1\ 4\ 1]*M \tag{1}$$

Now, if a complex M' is obtained from a complex M by substituting the mid-polygon m of M by the polygon:

$$m'=(1/4)*[-1\ 6\ -1]*M \tag{2}$$

then the limit of M' is a B-spline curve identical to that of m.

3. SCATTERED DATA POINTS INTERPOLATION

The interpolation task can modularly be divided into three major steps:

1- Slicing: this is the process of partitioning the initial set of data points into separate subsets.

2- Fitting: this is the process of ordering the elements of each subset so as to form a specific polygon.

3- Linking: this is the process of linking (and maybe modifying) the polygons of step (2) above so as to form a control polyhedron. This is constructed in such a way that its subdivision will (at the limit) result in a surface interpolating the initial data points.

We note here that the polygons generated in step (2) need not be all of the same length. Moreover, along with other known interpolation methods, this process introduces more auxiliary points, especially in step (3) above. These auxiliary points are necessary in order to achieve interpolation. In our approach, an upper bound on the number of these auxiliary points can be estimated in advance and is modest in comparison to that required by other interpolation methods.

Note that these extra points give the overall process an added degree of freedom, which can be exploited in modifying the quality of the interpolating surface. This can be used to satisfy additional user requirements such as local normal and cross curvature values in specified regions of the generated surface.

3.1. Slicing the 3D Data Points

The first step of the process consists of partitioning the 3D data points into separate subsets. The point elements of every subset can be ordered then linked according to this order so as to form an open polygon. The result of this process will be a set of "parallel" (i.e. non-intersecting) polygons.

In the absence of any guiding information as to how the slicing should be conducted, the basic guiding intuition that remains is the relative distances between various points and the likelihood that this slicing process will lead to those parallel polygons. We might add here that the coordinate system with reference to which these points are represented will have a role in the success of the algorithm presented below. Moreover, a relative coordinate system [5] that improves the performance of the algorithm presented below is worth investigating together with the other techniques [13].

3.1.1. *The Slicing Algorithm*

Input: a set of scattered 3D data points S

Output: a set of slices (P_i). Each slice P_i becomes a polygon toward the end of this process

1. Calculate the minimum (n_x, n_y and n_z) and the maximum (m_x, m_y and m_z) of the corresponding coordinate value (x, y and z) of all points <x, y, z> elements of S.
2. Calculate the differences $d_x = m_x - n_x$ and $d_y = m_y - n_y$ and $d_z = m_z - n_z$.
3. Arrange d_x, d_y and d_z in decreasing order then rearrange the coordinates of each point of S in such a way that $d_x \leq d_y \leq d_z$.
4. Ignore d_x for the moment.
5. Subdivide the interval [n_y .. m_y] into a sequence of k subintervals each of which has a length equal to no more than d, where d is the minimum non-zero difference between the z-coordinates of various points of S.
6. Divide S into a sequence P of k sets s_z, where each s_z contains the elements of S whose z coordinates fall into the corresponding subinterval of step 5.
7. Perform the union procedure on P (see below) repeatedly until each element of P contains the minimum required number of points.
8. Undo the arrangement performed in step 3 over all the points in P.
9. Perform an ordering over each element of P (for example, lexicographical ordering) with respect to x and z and in that order). The aim of this ordering is to make a polygon out of each of these element sets. At the end of the process, these will be transformed into a sequence of "parallel" (i.e. non-intersecting) polygons (see section 3 below for more elaboration on this particular step).

It is important to note here that the polygons resulting from the above algorithm are not necessarily of the same size (see figure 3). Furthermore, it will be instructive to compare this algorithm to the one presented in [5] which is intended to do a similar task. In comparison with that algorithm, for example, the data points in each layer do not have to be coplanar; that is, the layer can have some thickness.

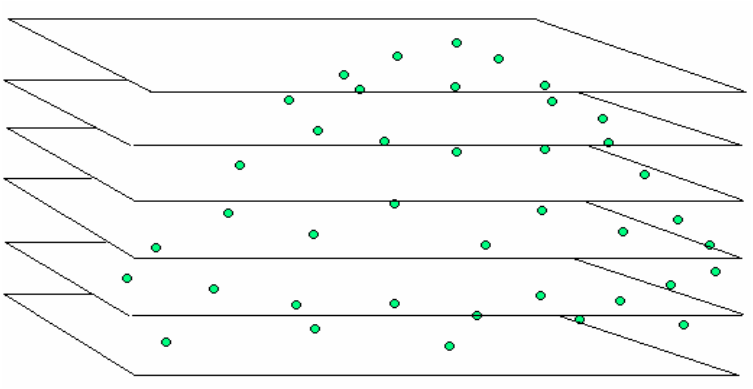

Fig. 3. The Slicing Procedure

3.1.2. *The Union Procedure over P*

1. Find the element s_z of P, which contains the minimum number of elements.
2. Unite s_z with the previous element p_z (if any) (resulting in u_z) and the next element q_z (if any) (resulting in v_z) in the sequence.
3. Choose from u_z and v_z the set that contains the minimum number of elements.
4. Replace by this set inside P, the pair of elements of P this set is the union of.

3.2. Fitting 3D Data Points into a Polygon

The task here is to order the data points of a given set so as to form a polygon with some desirable "smoothness" characteristics (see figure 4). The quality of the polygon resulting from this process will affect the quality of the resulting surface in a fundamental way. The main difficulty here resides in the fact that the "nice" property of the resulting polygon is inherently vague and, therefore, hard to capture.

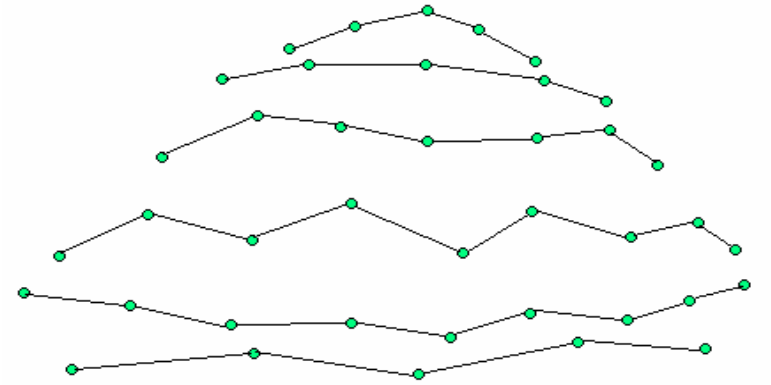

Fig. 4. The Fitting Procedure

A first attempt at doing that is to select some point in the set as an initial starting point and to pick the next point as the closest (in terms of Euclidean distance) and so on. However, experimental results have shown that reliance on the closeness property alone can quickly lead to anomalies (such as self intersection) in the resulting polygon.

This ordering problem may alternatively be looked at as an optimization problem. For example, in [11], this is seen as a contour deformation exercise. That is, the process starts from a known contour (a curve, such as the arc of a circle). This curve is then deformed step by step through minimizing a certain energy function. More details about this process can be found in [6]. This process terminates with the curve becoming a snake passing through all the initial data points. This way, the points are ordered as they are found on the curve, thus giving the desired polygon.

3.3. Generating the Control Polyhedron

3.3.1. *Interpolating a Set of Points by a B-spline Curve*
The task now is to link the generated polygons together so as to form a control polyhedron. When subdivided, this control polyhedron will converge on a smooth surface interpolating the initial data points.

This task requires the introduction of auxiliary points at two different stages of the process. These are:

1. Interpolating a set of points by a B-spline curve.

2. Interpolating a set of "parallel" (i.e. non-intersecting) curves by a subdivision surface.

This first stage is very well covered in the CAGD literature, and the second stage is often called lofting (or skinning [10]). However, it will be somewhat insightful to note here that the approach pursued in this paper regards the first stage as a particular case of the second. This can clearly be seen in the situation where every curve in the second stage collapses to a point. In this situation, the interpolating surface just collapses into the interpolating curve of these points!

The task of interpolating the data points specified by a polygon P = (mi)i is traditionally addressed through the construction of a system of linear equations. However, in our approach, all we have to do is to insert some more auxiliary points in between the points mi's to obtain a new polygon P'. This is done in such a way that there is at least one new point between any to consecutive initial data points. The knot refinement algorithm [3] may be employed here to produce these auxiliary data points. The minimum number of auxiliary points required for a polygon of length n is 2n+1.

Now, for every triplet M = [t m b] of consecutive points of the polygon P', were m is an initial data point and t and b are auxiliaries, we replace m by m' obtained by applying transformation (2) above. This way, we obtain another polygon P'' which, when subdivided, will converge on a curve interpolating the initial data points.

It is evident that the quality of the interpolating curve is very much affected by the way the intermediate auxiliary points are selected. In turn, the quality of the curve will affect the quality of the interpolating surface. So, we might as well exploit this additional degree of freedom to satisfy additional user requirements, such as normal and curvature constraints (see figure 5).

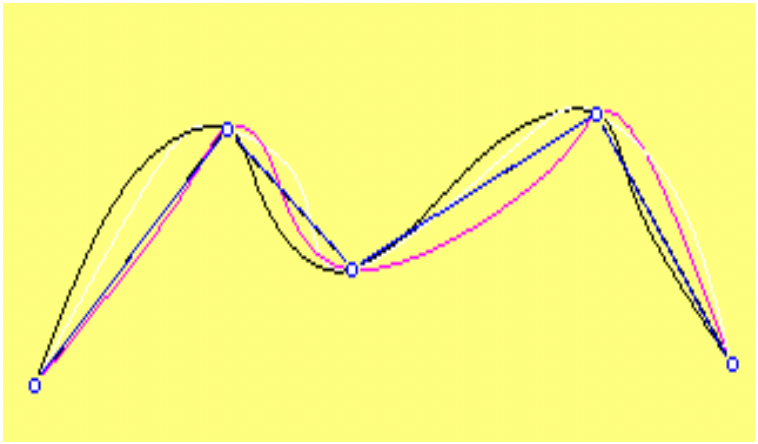

Fig. 5. Multiple Curves Interpolating a Given Set of Points

Here, we refer the reader to [1], where we discuss in more details how a surface can be modified to satisfy additional user constraints, while maintaining the original interpolation constraints intact.

3.3.2. *Interpolating the Resulting Curves by a CC Subdivision Surface*
The task here is to link the consecutive polygons obtained from the previous stage so as to form a control polyhedron. This polyhedron will then be subdivided to obtain the final interpolating surface.

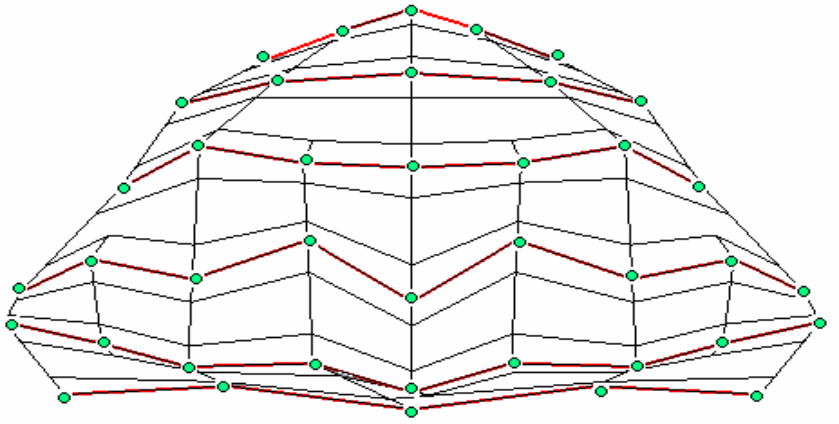

Fig. 6. The linking Procedure

We start the process by generating, for each polygon (mi), two more polygons (ti) and (bi) of lengths identical to that of (mi). This way we obtain a sequence of polygonal complexes Mj = (<t m b>i)j. Thus, two consecutive polygons (mi) and (mi+1) will be separated by two new polygons (bi) and (ti+1) not necessarily of the same length. Note here that these pairs of polygons are linked together using an algorithm that relies on information based on the relative Euclidean distances between pairs of points from corresponding consecutive polygons. The specific details about how this linking process is conducted may be found in [9]. Note here that if (bi) and (ti+1) both have the same length then both can be replaced by a single polygon: (<bi + ti+1>/2), for example.

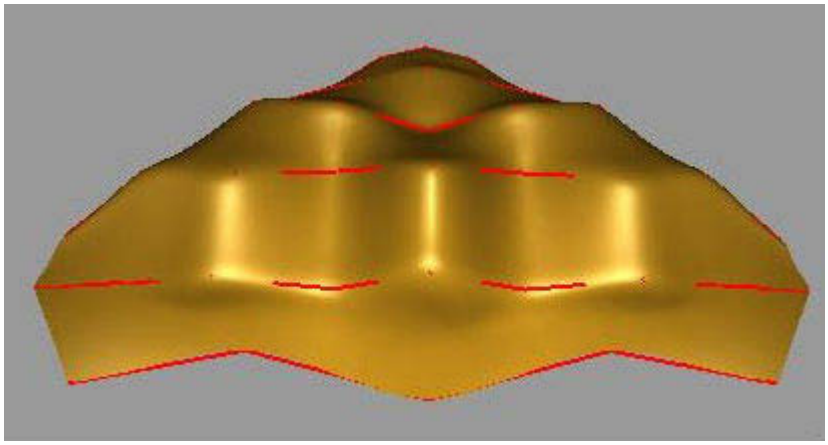

Fig. 7. The Interpolating Surface

Finally, we replace each polygon (mi) in the polygonal complex Mj = (<t m b>i)j by (m'i) obtained through transformation (2) above. When subdivided, the final polyhedron obtained as such will result in a surface that interpolates the curves corresponding to (mi). Therefore, this surface will interpolate the original data points (see figures 6 and 7).

Again here, the extra points obtained in this part of the process can be manipulated to satisfy user requirements as to the quality of the resulting surface. This is again related to the normal and curvature values at the specific points being interpolated. More details on this particular point can be found in [1].

4. SUMMARY AND FURTHER WORK

The approach outlined in this paper follows from a combination of several existing techniques: (1) slicing: to divide the main group of points into separate subgroups, (2) active contours (or snakes) to order each subgroup into a polygon, and (3) lofting (or skinning): to link these polygons into a single polyhedron whose subdivision will result in a surface interpolating the original control points. It is obvious that the quality of the resulting surface depends a lot on the relative position of the extra points added during the process. That is why plenty of care should be paid when selecting those points. Our approach comes equipped with useful tools for doing that.

Finally, we point out another distinct possibility that is worth following. In fact, the slicing phase may perhaps be substituted by a process (similar to triangulation) so that the sequence of non-intersecting polygons obtained at the end of this phase directly forms a network. This will eliminate the need for active contours and for skinning. The remaining steps of the process will then follow what is prescribed in [2].

5. REFERENCE

[1] Abbas, A. and Nasri, A. Deforming Subdivision Surfaces with Various Interpolation Constraints, AICCSA' 03, Tunisia, July 2003.

[2] Abbas, A. and Nasri, A Generalized Scheme for the Interpolation of Arbitrarily Intersecting Curves by Subdivision Surfaces, DEWS'05, February, Tokyo, Japan.

[3] Boehm, W. and Prautzsch, H. *The Insertion Algorithm*, Computer Aided Design, 17 (2): 58-59, 1985.

[4] Catmull-Clark, E. and Clark, J., *Recursively Generated B-Spline Surfaces on Arbitrary TMeshes*, in Seminal Graphics, Rosalee Wolfe Eds., pp 183 – 188, ACM Press, 1998.

[5] Hemayed, E. and Farag, A., *Slicing, fitting, and linking (SFL): a modular triangulation approach,* Three-Dimensional Image Capture and Applications II, Proceedings of SPIE, Volume 3640, San Jose, CA 1999, pp. 157-167

[6] Hofer, M., Pottmann, H., *Energy-Minimizing Splines in Manifolds*. To appear in: Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004).

[7] Mencl, R. & Muller, H. *Interpolation and Approximation of Surfaces from Three-Dimensional Scattered Data Points*, Scientific Visualization Conference June 9-13, 1997, Dagstuhl, Germany.

[8] Nasri, A. and Abbas, A. *Designing Catmull-Clark Subdivision Surfaces With Curve Interpolation Constraints*, The Journal of Computers & Graphics (special issue), Vol 26, No. 3, 2002.

[9] Nasri, A. and Abbas, A. Polygonal Complexes For Interpolating Curves by Catmull-Clark Subdivision Surfaces, CAD/Graphics 2001 in August 22-24 Kunming, China.

[10] Nasri A., Abbas, A. and Hasbini, I. *Skinning Catmull-Clark Subdivision Surfaces With Incompatible Cross-sectional Curves*, Proceedings of Pacific Graphics 2003**,** Canmore, Canada, ISBN 0-7695-2028-6, pp. 102-111, IEEE Press, 2003.

[11] Pottmann, H., Leopoldseder, S., Hofer, M., *Approximation with active B-spline curves and surfaces*. Proc. of Pacific Graphics 02, IEEE Press, 2002, pp. 8-25.

[12] Ramos, G. A. and Enright, W. H. *Interpolation of surfaces over scattered data*. Proceedings of the IASTED International Conference Visualization, Imaging, and Image Processing, 219—224, 2001.

[13] Wu, Y. F. Wong, Y. S. Loh H. T. and Zhang Y. F. *Modelling cloud data using an adaptive slicing approach*, Computer-Aided Design, 36 (2004) 231-240