Δ-Connection: A Solution for 3D Object Reconstruction

Guilherme Rossetti Anzollin, Marcelo da Silva Hounsell, Alexandre Gonçalves Silva

LARVA – Laboratório de Realidade Virtual Aplicada, DCC – Departamento de Ciência da Computação, UDESC – Universidade do Estado de Santa Catarina.

Campus Universitário "Avelino Marcante", Bom Retiro, Joinville, 89223-100, Brasil.

guianzollin@gmail.com; {marcelo, alexandre}@joinville.udesc.br

ABSTRACT

Given a set of consecutive slices resulting from a non-invasive examining device, there is an expectation to be able to reconstruct the 3D original object regardless if it is a human organ or the channeling of underground petroleum resources. The slices however, identify a set of curves, which need to be properly connected to give rise of a coherent representation of the object. This analysis is made by a correspondence algorithm within a 3D reconstruction software. This paper presents Δ-connection, a simple and flexible algorithm for the correspondence problem. Δ–connection relies on the well-known heuristic approach of proximal curves. Tests have shown that Δ-connection grows linearly with the size of the raw-data (the slices) and can be fine-tuned by a user-defined parameter to produce a 3D model. The heuristic, advantages and limitations of Δ-connection will also be shown in detail.

Keywords: 3D Object Reconstruction, Reconstruction from Slices, Correspondence Algorithm, Visualization.

(Received December 12, 2007 / Accepted March 26, 2008)

1. INTRODUCTION

The increasingly popular use of non-invasive measurement devices, such as Magnetic Resonance Imaging (MRI) and Computerized Tomography (CT), has made it possible to visualize a sequence of planar sections of three-dimensional objects. This fact has motivated the development of many (such as [1], [2], [3], [5] and [7] which will be presented later) three-dimensional object reconstruction techniques. Three-dimensional reconstruction became a very interesting and importante research technique since it builds a 3D model of the object that is being analyzed through the use of two-dimensional images.

3D reconstruction can be executed in order to obtain various information about the original model, basically for two reasons:

- To study the model´s structure, the relationship among parts, understanding of the whole and the existence or non-existence of certain formations (tumors, swellings, etc.), and;
- To obtain measurable characteristics of the model (volume, area, length, etc.).

Figure 1 illustrates a slice obtained from a CT scanner already treated in order to separate the contours for the reconstruction (a); the reconstruction process executed over two of these slices (b) and; the finished process (c) of a skull [2].

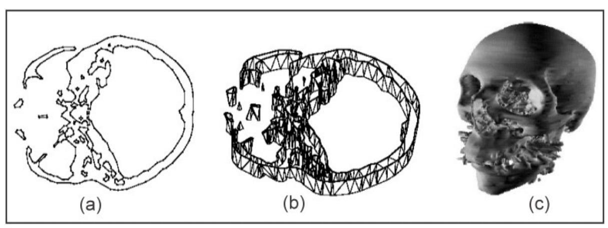


Figure 1: The 3D Reconstruction Process on a Skull: (a) one slice, (b) two connected slices and (d) the complete model [2].

Figure 2 shows the sequence of two-dimensional slices that represent blood veins (a) and the interpolation (b) of these slices generating all the corresponding channels/veins [3].

This is an example of reconstruction where the model visualization as a whole is more important than the intrincacies and details between slices.

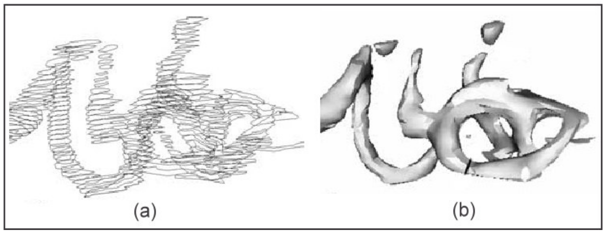


Figure 2: The 3D Reconstruction of Veins: (a) the set of slices and (b) the reconstructed model [3].

It can be observed that this division leads to two possible approaches regarding the application of the 3D reconstruction techniques: one aimed at the object visualization, and the other at accurately representing the object. For the former objective, the reconstruction process can be done in a simpler way, without the need to accurately represent the saddle point on the branching, for instance. On the other hand, if the reconstruction does not take into account the accuracy of the consequent branching and surfacing, any measurements taken from the model can be misleading.

Reconstruction techniques aiming the visualization of structures are useful to applications such as the identification of increasing density of capillaries (several grouped ramifications), a congenital bad formation (the appearance of a strange geometry), tumors (protrusions on the geometry) or even undesired/unexpected connections. These applications take advantage of the power such techniques have to make visible a formation whose visualization would be, in other ways, invasive. For these techniques the flexibility and the rules of correspondence identification are of great importance along with the fact that a fast response is needed.

This paper will present an algorithm to identify the correspondence between curves in consecutive slices aimed at object visualization and focused on flexibility and efficiency. The text will initially identify the problem and then, show related approaches. After that, heuristic solutions will be thoroughly discussed so that, in the following section, the Δ-connection algorithm will be presented. Implementation details, results, analysis and the conclusion will finish this text.

2. PROBLEM DEFINITION

The 3D reconstruction solution is usually performed considering three clearly separated steps (correspondence, branching and tiling) having a set of techniques that are specified to decide the geometry and topology of the final model.

The correspondence problem arises when there is more than one curve in each of parallel consecutive slices and some of them must be connected to generate the 3D object model. The correspondence problem can be stated as:

> Given a set of i curves C_{ij}, where i=1,…,n and n is the number of different curves in the plane Xj, identify all subset of curves in plane X_k that correspond (must be connected) to some in plane $X_{(k+1)}$.

The correspondence step has been considered the main problem of the 3D reconstruction from planar sections [1]. Figure 3 presents 3 of 11 possible correspondence solutions that can be generated from the two initial sections shown on the upper-leftmost corner of the figure.

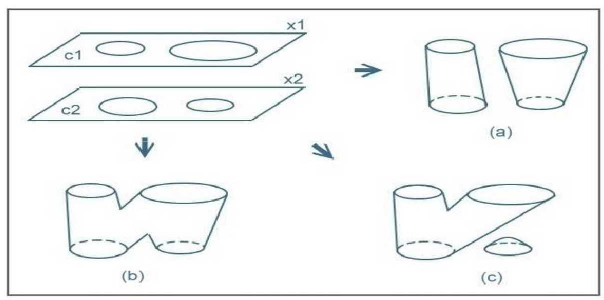


Figure 3: Three Examples of Correspondence Alternatives (a), (b) and (c) from Slices x1 and x2.

In one of the cases (a) each curve on a plane is connected with only one nearest curve on the next plane; in another, one of the curves from the inferior slice is connected with the two others on the superior slice, while the other is connected only with the nearest (b) and; on the last illustration only one of the curves form the inferior slice is connected with the curves of the superior plane (c).

3. RELATED WORK

Various techniques have been developed to deal with the correspondence between curves. The following can be highlighted:

The **deformable models** approach uses geometry, physics and the theory of approximation for reconstruction. The geometry is used to represent the shape of the object, the physics impose confinements on how the shape can vary in space and time, and the theory of approximation provides mechanisms and techniques to approximate the reconstructed models to the original measured data. On this method, deformations are made on an initial model, to reach the final object.

McInerney and Terzopoulos [4] presented a reconstruction work applied to medicine that uses deformable models, and proved to be efficient to start from a sphere and promote deformations and approximations until a desired model is achieved. One advantage of this technique is that the image segmentation process, where a polygonal representation of the curves from the original image is obtained, is part of the reconstruction process. The authors asserted that deformable models overcome many of the limitations of low-level techniques for image processing, providing compact and analytical representations of the object's shape. However, the reconstruction process is not an isolated process and it can be said that the reconstruction techniques through deformable methods use more image processing concepts than geometric modeling.

The **implicit approach** uses an implicit function to interpolate the curves and generate the object, in a way that the object surface (the edge of the object) is on the zero set of this function, that is, in f(x,y)=0. This function is determined from the interpolation of the functions of each parallel planar section (slice) that contain the curves to be connected.

Peixoto and Gattass [5] describe the implicit approaches through two steps: the definition of the functions that represent the curves' slices, called field functions, and; the interpolation of these functions to form the implicit function that will represent the final object's surface as a whole.

For this approach the matrix-based (also raster-based) representation of curves is more adequate, since there is a natural correspondence between the matrix representation and the implicit function, i.e., a curve represented as a matrix can be defined as the set of points (x, y) of the slice, such that f(x, y) represents each field function used to generate implicit function.

The correspondence definition step does not have much flexibility in the implicit approach because (i) they are automatically defined by the function; (ii) the result is a unique interpolation solution for a given initial set of curves [5], not all alternatives and; (iii) the connection determination for this type of approach is considered one of it's major problems [6: page 3]. Implicit approaches deals with the reconstruction problem in an automatic way but it does not generate all the possible models from a set of curves.

Approaches that use **heuristics** deal with the correspondence criteria with more flexibility. According to Peixoto and Gattas [5], the decision of the correspondence can be taken computing somehow the distances between curves.

The heuristics used in the work of Barequet and Sharir [7] decide on the correspondence of the curves based on a XY projection of two consecutive planes. The heuristic is the following: if there is an intersection on the projected area of the curves they are connected, otherwise they are not.

The work of Treece and colleagues [6] is based on the calculation of the distance between regions of the curves. To each plane containing the curves a set of discs is created. Internal discs are used to represent internal regions of each curve and are considered to loosely represent the shape of a curve.

Figure 4 shows an example of slices with many curves (one at the bottom slice and three at the top one, see left-most drawing); their XY projection (shown in the center), and; the resulting correspondence (at the right-most drawing one can see that the bottom curve was found to correspond to two of the top ones).

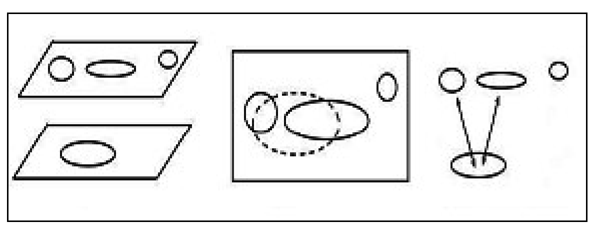


Figure 4: Curves on two slices are projected in one plane and the resulting correspondence.

To each disc, its center is calculated, called centroid, which will be used to calculate the distance between each pair of discs of two consecutive planes. The correspondence calculation is based on the distance of each pair of discs. The heuristics defined in this algorithm is the following: the regions on two consecutive planes will be connected if the distance between the related discs is smaller than the radius of both discs. Then, to each two consecutive planes, a comparison of distances between the centroids of each pair of discs is done. In this way, the necessary distance to connect to discs may vary without user control. Also, the correspondence does not take into account the whole area of the curve, but each region represented by a disc.

Another technique that uses heuristics is proposed by Cuadros-Vargas [1]. It is a volumetric reconstruction strategy called β-Connection that has the flexibility to produce a family of objects constructed from the same set of planar sections, making it possible to obtain multiple options of a final object.

To solve the correspondence problem the algorithm performs a calculation of the smaller distance between each two curves in terms of tetrahedrons. Afterwards, it takes a user defined integer parameter, called β, to perform the heuristics: if the distance between any two curves (measured by the number of inbetweening tetrahedrons) is less than the value of the β parameter defined by the user, then these curves are connected.

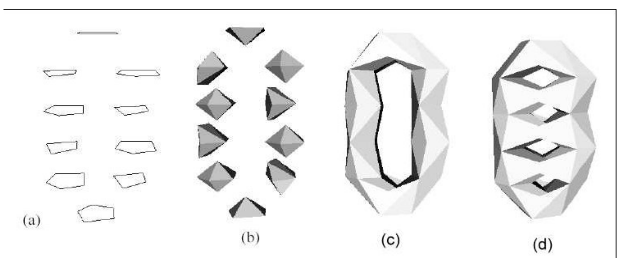


Figure 5: Reconstruction via β-connection [1].

Figure 5 shows different connection solutions resulting from the strategy proposed by [1]: Beginning with the curves situated in parallel sections (a); in the first solution (b) the value of the β parameter is less than all the distances between the curves of two consecutive planes, with no connection occurring; in the second solution (c) the value of the parameter β is 3, then all the curves of two consecutive planes with distances between each other less than or equal to 3 are connected, and; when the value of β is greater than the distance between any curves of two consecutive planes all of the curves of those consecutive planes are connected (d).

An important feature of the work of Treece and colleagues [6] is that the regions represented by the discs will only connect with the other closest regions of each consecutive plane if this relation is reciprocal; this allows regions to be left without connection. These authors also emphasize that traditional branching and correspondence problems are combined by determining “regions correspondence”. For Barequet and Sharir [7] it is not necessary for two curves to overlap to connect to each other.

According to Cuadros-Vargas [1], the β-Connection reconstruction technique offers more flexibility on the choice of the connected components, since from a same set of planar sections it is possible to obtain different shapes of objects, which is difficult through other algorithms in the literature.

Comparing the heuristic approaches presented, one notices that all existing algorithms employ some form of distance calculation between curves:

- This can be a very detailed and time consuming comparison of point by point of the curves in search for the smaller edges (as in [8], apud [1]);
- Indirectly, by enclosed or enclosing discs (as in [6]);
- Indirectly by the resulting overlapping of the projections that occur when the curves are next to each other (as in [7]).
- Ingenious calculations, that take into account the distance in units of volume (tetrahedron), has also been tried and by-producing tiling with great flexibility of results with the cost of greater computational demand (as in [1]).

This paper presents another solution to the calculation of the curves proximity with the same flexibility as in [1] but keeping the correspondence stage totally isolated from any others.

4. THE Δ CONNECTION SOLUTION

The Δ-connection algorithm has the following scope for its proper functioning:

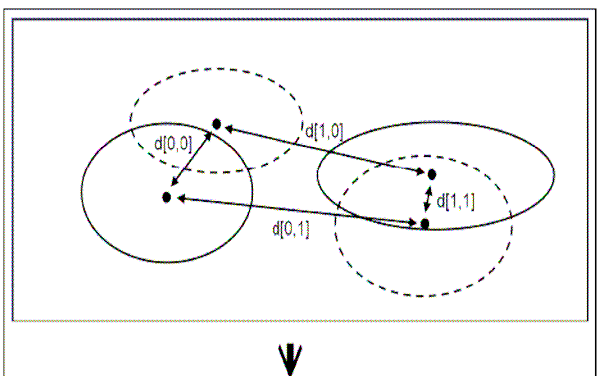
- Curves represented in a polygonal form (vectorized) with the same orientation;
- Convex and not self-intercepting concave curves;
- Closed non self-intercepting curves and that do not contain other curves in its interior;
- Resulting object represented in Boundaring-representation, using VRML;
- The saddle point of the branching calculation is not dealt with;
- The preferential application is structure visualization (such as channels).

The Δ-connection solution defines the correspondence of the curves in 3 steps:

1. The centroid of each curve is calculated regardless its slice (Z value).
2. A matrix is built with the Euclidian distances of the curve's centroid in consecutive slices altogether with the minimum and maximum distances (Δmin, Δmax);
3. Given a user-defined Δ, the heuristics is evaluated.

Figure 6 illustrates, from a projection of two planes, the distances between two centroids of the curves (labeled d[x,y]) and the matrix where they are stored. The curves drawn in dotted lines belong to the projected plane.

The minimum and maximum distances in the matrix have the purpose of informing the user which is the interval of distances among all the curves. The centroids are calculated as the center of the curve's bounding box. The distances are calculated with the curves projected in the same plane, that is, despising the height between planes, which simplifies the distance calculation.



	CentroCurvaP2[0]	CentroCurvaP2[1]
CentroCurvaP1[0]	d[0,0]	d[0,1]
CentroCurvaP1[1]	d[1,0]	d[1,1]

Figure 6: Curves´ Distances in the Δ-connection.

The detailed description of the heuristic is as follows:

- If Δ is less than the minimum distance in the matrix, no curve is connected;
- If Δ is greater than or equal to the maximum distance in the matrix, all curves on those two consecutive slices are connected;
- If Δ is in-between the minimum and maximum distances, the following rule is adopted:
  - If the distance in the matrix is less than or equal to Δ, then these curves are connected;
  - If the distance is greater than Δ, then no connection between these two curves will occur.

To each two curves that are connected from the defined heuristics, these curves are marked in the algorithm as "connected". After correspondence determination, the algorithm performs tiling. To the curves that had no connection (unmarked curves), a face from its vertices (to work as top or base to the visualization) will be generated.

The novelty of Δ-connection is three-folded:

- it considers the distances between the center of the curves (centroids) in consecutive planes for proximity reasoning;
- it is a conceptually simple solution and;
- the flexibility that can be found in (i) a parameter-controlled reconstruction process; (ii) the possibility to use several alternatives for distance calculation and; (iii) a variety of ways to define the curves´ centers.

4.1 Implementing Δ-Connection

Firstly, a curve editor application (Figure 7) was implemented that reads and writes the curves in a standard XML file format (that will be defined latter). The Δ-connection reconstruction algorithm will use these data and perform the heuristics, creating a VRML file with the geometrical information of the resulting object.

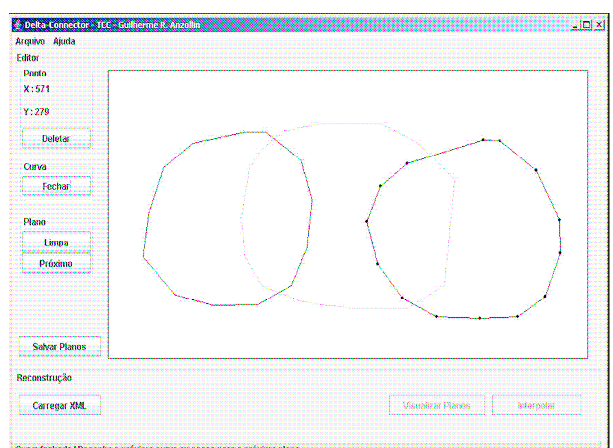


Figure 7: A Curve Editor for a Given Slice.

The VRML file format was chosen as an output format as it is an ISO standard for 3D data definition across the internet and can be visualized at any web browser with the appropriate, usually free of charge, plug in.

During the implementation, input files containing curves from real data [9], ISO standards [10] and vectorized data in VTK format [11] were analyzed. The former uses specific but not readily explicit data organization. The second is a too verbose solution and the latter divided the representation of an object in many files, each containing a plane, which contradicts the idea of storing all the set of planes pertaining to an object in a single file.

The XML file is used as storage format for the edited curves and as input to the reconstruction algorithm and it was preferred because it could be organized in a way that the representation of the curves was made clear and as close as possible to the raw data from [9].

5. RESULTS AND ANALYSIS

In order to demonstrate the capabilities of the algorithm, curves were created using the curve editor with the purpose of generating specific and controlled situations (all tests were carried out considering a constant distance between slices). A set of curves was confectioned aiming to create many possibilities of branching to exercise the influence of the Δ parameter.

The first set of curves generated (see Figure 8) produced many situations of connection and non-connection as well as top and bottom faces.

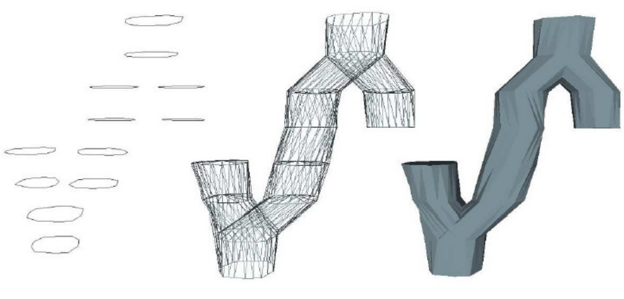


Figure 8: Reconstructed Test Case.

In Figure 9, another set of curves at various slices can be seen (a) which is also the result of performing Δ-connection with Δ equals to 0. As Δ increases, more (b) and more (c) connections are formed at consecutive slices. For a very large value of Δ, all curves are connected. These results are visually and functionally similar to those obtained with other approaches (see Figure 5) but instead; the distance is Euclidean, fast and simple to follow.

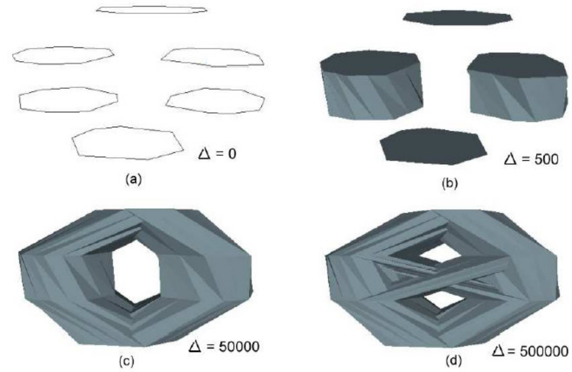


Figure 9: Proving the Concept.

Once the proof-of-concept was validated, and some visualization resources were refined in the application (the list of curves that originated the object, its wireframe model and the rendered object), the behavior of the Δ-connection with real data was then, evaluated.

To do this, data available in [9] were converted to XML and the Figures 10 to 13 were generated. These figures shows at the left-hand side the set of slices used for the reconstruction process and, at the right-hand side the complete reconstructed and rendered model.

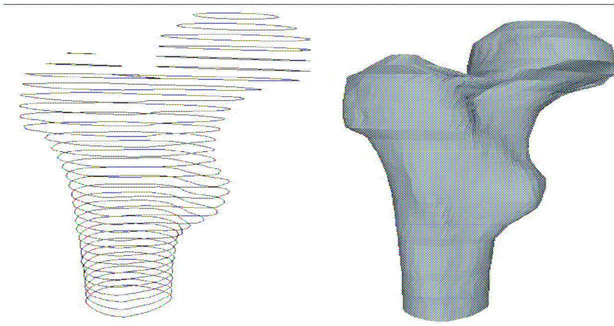


Figure 10: The reconstruction of a Femur.

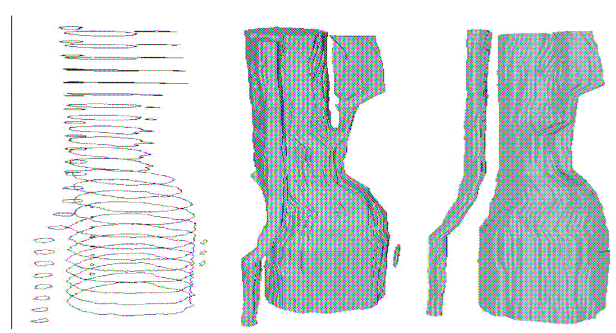


Figure 11: The Reconstruction of Veins.

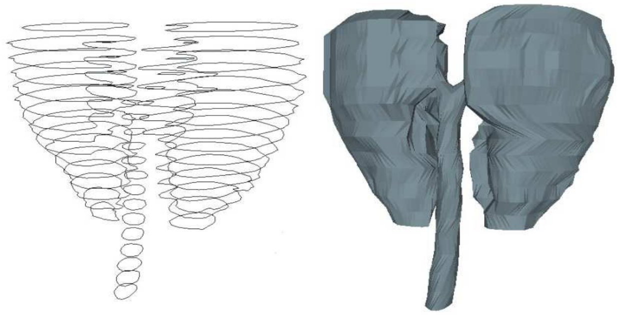


Figure 12: The Reconstruction of Lungs.

An interesting case to highlight was the reconstruction of a heart, presented in Figure 13, formed by a set of 1285 points distributed in 30 slices.

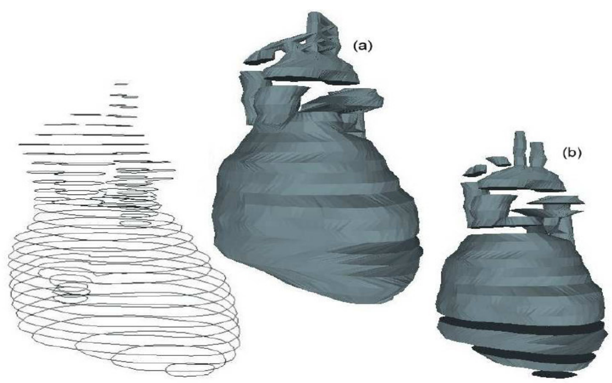


Figure 13: The 3D Reconstruction of a Heart.

In Figure 13, one notices that the top region of the model possess a fair amount of curves which are close to each other if compared to the rest of the model that possesses fewer and larger curves. A model with these features may have an undesirable result whatever the Δ value is because a given value can be good for large curves but generate undesirable connections in the region of much smaller curves, as in figure 13(a) where, practically all the curves are connected to each other (note the connection configuration in X in the superior part of the figure 13(a)).

On the other hand, defining a smaller value of Δ, in order to control the excess of connections between small curves (top of the model in Figure 13(b)), may result in the lack of connections in the region of larger curves (as in the bottom part of figure 13(b)).

Another analysis that was done was in relation to the correspondence parameter Δ used to generate each one of the real examples illustrated.

Figure 14 shows the minimum and maximum Δ values to each model. The dot in the scale refers to the value of Δ that was chosen as satisfactory to generate the desired connections. The lines refer to Δ values for the femur, lungs, hear and veins reconstructions, respectively. One notices that, to all cases, the value given to Δ is small (approximately 10%) in relation to the distance interval [Δmin , Δmax]. This occurs because the connections were considered necessary only for those curves that were really close to each other.

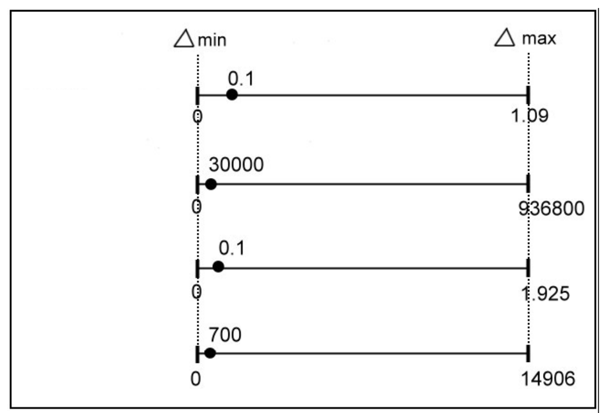


Figure 14: Chosen values for Δ.

The case in which Δ was farther from the minimum distance value was in reconstruction of a femur, where the branching only occurs once and most of the slices had only one curve that was always interpolated with the curve from the next slice. It can be concluded that, since it is desirable to interpolate only the curves that are next to each other, the value of Δ will have the tendency to be small when compared to the distance interval.

Δ-connection efficiency was measured and the graphic on Figure 15 was generated presenting the elapsed time (shown as the vertical axis) registered for several models with a different number of points (shown as the horizontal axis), up to 40000 generated points.

The curve in Figure 15 shows that the Δ-connection algorithm has a nearly linear growing pattern with the increase of the number of points. The graphic however, refers to the total time of the algorithm that, besides the correspondence analysis, also deals with the tiling between curves (which was not covered in this paper due to the lack of space).

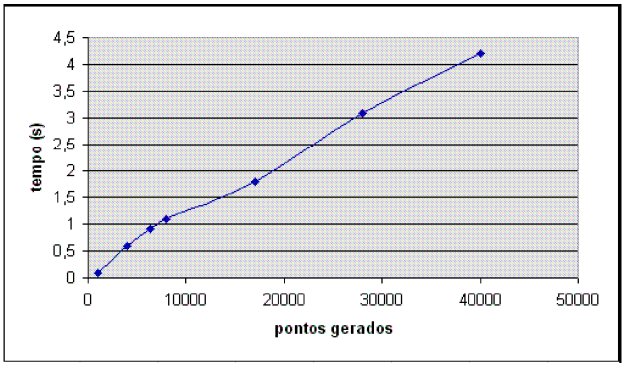


Figure 15: Δ-connection Time Performance.

The tiling algorithm used was a straight-forward one that aims to connect every single point in one curve to one in the corresponding curve of the other slice so that a triangular face is generated. The search starts at any point and an edge is created. Then, the reference point for the edge construction alternate from one curve to the other up to all points are connected and all possible triangles between the slices are created.

The experiments were carried out on an AMD Athlon 1.3 Ghz processor with 512 MB of video memory, without a 3D acceleration card. All the data relating to the results obtained, as well as the developed application and the algorithm' source code, are available at the web [12].

As a functional analysus, it can be said that the algorithm expects the input of a parameter that makes it flexible to decide on the result of the reconstructed 3D object, feature not commonly available in other solutions. The parameter Δ is directly related to the distances between the centers of the curves, that is, the heuristics infers the proximity between the curves (a well accepted heuristic for it happens in several solutions in the literature).

Both the distances calculation and the definition of the center of the curves can be done using several alternative forms (that weren't explored in this work). To this work decisions were taken in order to obtain performance. To this end, as stated before, the square of the curves distances projected in one of the planes (2D) were used on distances comparisons instead of the square-root (which is much more time demanding) of the distance in the 3D (where there is another axis do compute) and; the centroid used was, in fact, the center of the curve´s Bounding Box (and not an average of all curve's points). These decisions do not affect the accuracy of the results.

The Δ correspondence control value can be chosen at random by the user, but the user receives information about the limits for better guidance, avoiding the choice of the parameter by simple trial-and-error. The Δ-connection has also the advantages of being independent of the dimensions used to represent the curve; gives the same result if executed top-down or bottom-up, and; deals exclusively with the correspondence problem allowing fine-tuning without affecting other steps of the reconstruction process.

6. CONCLUSION

A novel solution to the correspondence problem based on the heuristic approach of three-dimensional reconstruction has been presented. The solution considers the Euclidean distance between the centers of the curves (centroids) in consecutive planes. This solution has speed, conceptual simplicity and, most importantly, the flexibility as efficiency criteria. Flexibility is a major advantage over other approaches and can be found in (i) the control of the reconstruction result; (ii) in the possibility to use several alternatives for distance calculation; as well as (iii) in a variety of ways to define the curves´ centers.

The tests were performed with data generated by an implemented curve editor and also with real data from medical images. The algorithm generated satisfactory visual results and presented linear performance with the increase on the number of points and slices.

A limitation was observed when the algorithm is applied to models with great variability on the size and distance between curves, which shows the need for future work, experimenting alternatives that allow the definition of different values to Δ for different regions of the model and/or a way of automatically changing the value of Δ depending of the size of the curve. For this, however, one must further study the concept of what can be understood as the size of the curve and how this adaptation can occur.

7. REFERENCES

[1] Cuadros-Vargas, A. J. *β-Connection: A family of Reconstructed 3D Objects from 2D Planar Sections*. Master Thesis Dissertation — Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, 2001. (in Portuguese).

[2] Bajaj, C. L., Coyle, E. J., and Lin, K.-N. Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models and Image Processing*, v. 58, n. 6, p. 524–543, 1996.

[3] Nonato, L. G., Cuadros-Vargas, L. J., Minguim, R., and De Oliveira, M. C. F. Beta-connection: Generating a family of models from planar cross sections. *ACM Transactions on Graphics*, v. 24, n. 4, p. 1–20, Oct. 2005.

[4] Mcinerney, T., and Terzopoulos, D. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, v. 1, n. 2, p. 91–108, 1996.

[5] Peixoto, A., and Gattass, M. *Surface Reconstruction from Two-dimensional Sections*. Technical Report MCC 28/00, PUC-Rio, July 2000. Also available at http://www.visgraf.impa.br\Data\RefBib\OS_PDF\peixoto-gattassjul00=SurfaceReconstruction.pdf. (in Portuguese). Accessed in October 13, 2006

[6] Treece, G. M., Prager, R. W., Gee, A. H., and Berman, L. *Surface interpolation from sparse cross-sections using region correspondence*. Technical Report CUED/F-INFENG/TR 342. Cambridge University. Engineering Department. 25 pages. March, 1999.

[7] Barequet, G., and Sharir, M. Piecewise-linear interpolation between polygonal slices. *Computer Methods in Applied Mechanics and Engineering*, v. 63, n. 2, p. 251–272, 1996.

[8] Ekoule, A.B., Peyrin, F.C., and Odet, C. L. "A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours", *ACM Transactions on Graphics*., 10(2), pp. 182-199. 1991.

[9] Barequet, G. *Publicly Available Medical Sliced Raw Data*. Available at ftp:\\ftp.cs.technion.ac.il\pub\barequet\psdb. Accessed in November 16, 2006.

[10] *International Organization For Standardization. Technical Specification* - ISO/TS 10303-28 - Industrial Automation Systems and Integration – Product Data Representation and Exchange - Part 28: Implementation Methods: XML Implementation of EXPRESS Schemas and Data. [S.l.], 2002.

[11] VTK. *Visualization Toolkit*. Available at: http://public.kitware.com/VTK. Accessed in November 16, 2006.

[12] LARVA. Site of the Applied Virtual Reality Laboratory (LAboratório de Realidade Virtual Aplicada, in Portuguese). Available at www.joinville.udesc.br/larva. Accessed in March 31, 2008. (in Portuguese).