Automated Solid Models From Serial Section Images

Chris M.W. Ho, Michael W. Vannier, and Steven J. Bresina

A new method for creating unambiguous and complete boundary representation solid models with a hybrid polygonal/nonuniform rational B spline representation was developed and tested using computed tomography scans of the wrist. Polygon surface approximation was applied to a sequence of parallel planar outlines of individual bone elements in the wrist. An automated technique for the transformation of edge contours into solid models was implemented. This was performed using a custom batch file command sequence generator coupled to a commercially available mechanical computer-aided design and engineering software system known as I-DEAS (Structural Dynamics Research Corporation, Milford, OH). This transformation software allows the use of biomedical scan slice data with a solid modeler.

COMPUTER-AIDED DESIGN and computer-aided manufacturing (CAD/CAM) technology has long been used in industry by engineers to more effectively design and test products before fabrication. This is usually accomplished by iterative improvements, evaluation of alternatives, and elimination of tedium. The use of solid modelers in this process has recently emerged as dominant in the field, particularly in especially demanding or complex applications.

A solid modeler is an analytical tool with the hardware and software necessary to define, manipulate, and display physical objects as solid models.[1] A solid model is a mathematical and graphic description of some geometric object that is both complete and unambiguous. Any such object, once defined as a solid model, may then be rigorously analyzed and subjected to mathematical manipulations that are analogous to geometric actions taken with real space-filling entities. Object properties such as center of gravity, volume, mass, and moments of inertia may be computed directly from the internal computer representation or description of the object.[2] Solid models can also be graphically manipulated using various Boolean functions. For example, objects can be joined into composites or assemblies, dissected or separated into subobjects, compared for unique and/or common volumes by subjecting the individual objects to Boolean operations, or simply reoriented to improve visualization by producing more informative views. Physical perturbations may be simulated: eg, forces can be applied to a system of objects and the distribution of stresses may be monitored. In essence, complete geometric information concerning an object may be entered into the model, and this information can be used to simulate and evaluate the responses of the object to actual conditions.

Although the use of solid modeling is quite common in mechanical engineering, few applications have been tested in biomedical research. Primarily this stems from restrictions and difficulties imposed by the methods used to construct solid models. One such method is the use of constructive solid geometry (CSG).[1] In this method, solid models are built from fundamental shapes called primitives. Primitives include blocks, cylinders, tubes, cones, and other derivatives of these shapes including solid polyhedra of variously numbered sides. An engineer uses these primitives along with Boolean operations (eg, union, intersection, etc) to create a working model of the object under investigation. While this task may be fairly complex, the object assemblies one often encounters in industry are composed of component parts that resemble the primitive objects. However, biological specimens are almost never composed of regular standard geometric primitives. Instead, irregular outlines and shapes dominate the structure. To view such objects both wireframe and surface modelers have been the mainstay of three-dimensional image generation in the biomedical sciences[3] (Fig 1). A wireframe modeler manipulates coordinate data describing only the surfaces and vertices of an object. Three-dimensional representations of physical objects are

From the Mallinckrodt Institute of Radiology, Washington University School of Medicine, St Louis, MO.

Supported in part by the American Society for Surgery of the Hand.

Address reprint requests to Michael W. Vannier, MD, Mallinckrodt Institute of Radiology, Washington University School of Medicine, 510 S Kingshighway Blvd, St Louis, MO 63110.

0897-1889/92/0502-0004$03.00/0

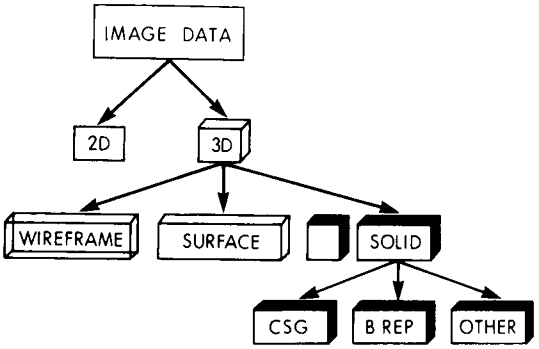

Fig 1. Different methods of representing image data.

produced by triangulating these coordinates to form representative surfaces.[4-6] Although very complex models may be displayed, such representations are ambiguous in that internal and external surfaces are not clearly delineated. Furthermore, hidden surfaces remain in view, requiring viewer interpretation to determine object orientation. With surface modelers, hidden surfaces can be removed. In addition, using ray-tracing algorithms, realistic illuminated displays of the object may be generated; however, these representations are merely graphical pictures. No other properties may be calculated, as surface modelers do not retain any data regarding the internal structure of the object.

Solid Model Creation—Irregular Objects

Special methods are required to create solid models of irregularly shaped objects. One such procedure entails the use of a mesh of curves (Fig 2). This consists of two orthogonal sets of planar contours that define both transverse and longitudinal cross sections. A solid modeler can then reconstruct the original object by joining the individual contours to form a single smooth outer surface using cubic spline functions. This process is called "skinning."

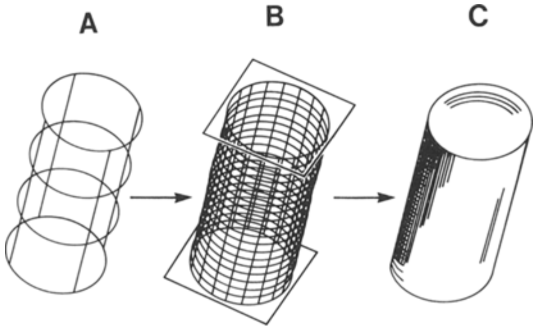

Fig 2. The creation of solid models by skinning a mesh of curves. A mesh of curves (A) consists of two orthogonal sets of planar contours that define the space an entity exists in. A solid modeler can use this mesh to produce a skin group (B), which is then converted to a valid solid model (C).

In solid modeling, planar contours are often created with cubic splines. An investigator manually selects consecutive control points along the cross section of a desired object (Fig 3). This is often accomplished with a mouse and pad by selecting loci from the border of an object in a photograph or drawing. After all points have been defined, the solid modeler produces an interpolative curve, thereby defining the contour. Using this method, virtually any complex shape can be approximated.[7] Ideally, all points lining the border should be used as control points to ensure an exact reproduction. Unfortunately, there are limits imposed by each solid modeler as to the maximum number allowed. Therefore, the task is to determine those border points that would produce the most accurate approximation.

Manually selecting and entering each control point for a number of cross sections can be quite tedious. Inconsistency and error can also occur. Figure 4 shows one common mistake. It is imperative that regions of high curvature be thoroughly defined with appropriately placed control points. Otherwise, ambiguities exist leading to the generation of inaccurate contours. From this example, one should also realize that it is insufficient to simply select every Nth point along the border. Regions of high curvature require more control points. To improve this process of data entry, we developed an algorithm to automate the designation of control points that results in a consistent, accurate, and computationally efficient contour definition. This algorithm allows the automated creation of simple solid models from serial slice data.

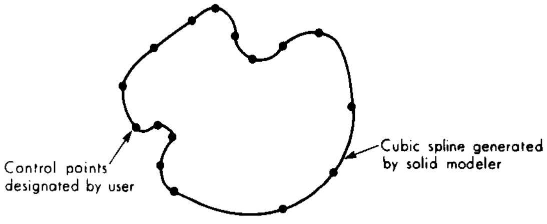
Control points designated by user
Cubic spline generated by solid modeler

Fig 3. The creation of planar contours using cubic splines.

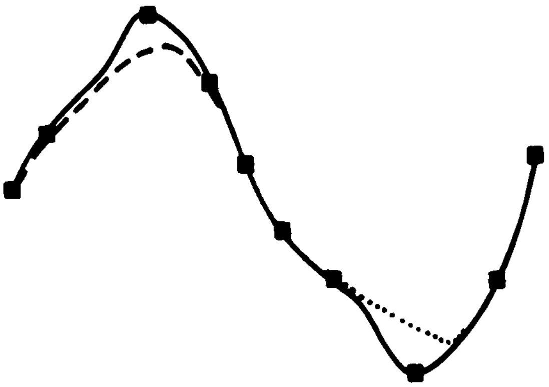

Fig 4. Error produced by ambiguous control point placement. The solid black curve depicts a cubic spline defined by the ten control points drawn as squares. The dotted and dashed lines show the resulting curves when a single control points along the regions of highest curvature are left out.

Polygon Approximation Methods

Similar issues have been addressed in the field of computer vision and pattern recognition.[8-10] Their objective is to extract useful visual elements from complex digital images or scenes. Of interest to us are methods that have been developed to transform two-dimensional complex contours into simpler shapes that are identified and labeled by visualization algorithms. One means of representing complex structures is by polygon approximation. A polygon approximation is a simplification technique that partitions the pixels making up a nonlinear contour into consecutive line segments (Fig 5). The task is thus reduced to selection of appropriate points to adequately represent the boundary curve. Psychological experiments described in the 1950s showed that a complex figure could be approximated by line segments connecting boundary curve points selected at regions of high local curvature.[11] Since then, numerous algorithms have been developed to perform polygon approximations.

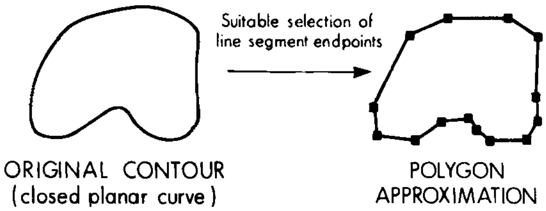

Fig 5. Polygon approximation of a curve. With the suitable selection of line segment endpoints, any contour may be represented as a polygon.

One particular approach shared by several methods involves the analysis and merging of consecutive contour pixels into segments whose deviation from linearity does not exceed a predetermined threshold. This threshold is made appropriately small to allow the approximation of each segment by a straight line. Investigators have quantitated this deviation from linearity by dividing contours into numerous small segments and monitoring the magnitude of the angle formed between consecutive segments.[12] In this way, segments whose intersecting angles are smaller than a given threshold are merged together. Others have monitored the distance that consecutive pixels deviate locally from an ideal linear course. Those pixels that fall within a predetermined distance are merged into a common segment.[13,14]

Another polygon approximation technique that several methods share involves the splitting of contours into segments of near-linearity. The iterative end-point fit algorithms of Duda and Hart,[15] Ramer,[16] and Lowe[17] exemplify this procedure. A contour is initially divided into two segments, each of which is bounded by two endpoints. Each segment is then analyzed to find the pixel within it that is the farthest from a line joining its endpoints. If that pixel is within a threshold distance, the segment is retained intact and is no longer processed. Otherwise, the segment is split into two others at that location, each of which is again analyzed. At every level a decision is made as to whether a single line or two smaller segments better represents the contour. This procedure continues iteratively until no new endpoints are found. Han et al[9] decreased the computational time necessary to implement this method by adding a line search procedure. A similar splitting algorithm was proposed by Lowe.[17] Rosin and West[18] enhanced Lowe's method by including the fitting of arcs as well as line segments.

These methods result in the generation of approximating segments as a function of the intrinsic equation of the curve. This function pertains to the curvature of a specified region in relation to its arc length.[15] In essence, as the nonlinearity of a contour region increases, more endpoints and line segments are required to represent it. In this regard, the concept of polygon approximation is ideally suited for the generation of control points. Each method has its advantages. The merging schemes are computationally faster as they basically are one-pass operations. However, according to Ballard and Brown[8] they are less accurate at selecting appropriate endpoints. Han et al[9] have also shown mistakes in the determination of vertices. Although the segment-splitting methods are more effective in selecting appropriate points, they are computationally more intensive. The needs of our particular application warrant special consideration. Contours are defined through the selection of consecutive control points through which cubic splines approximate the contour shape. A modified single-pass merging algorithm would be more efficient in this regard as approximating segment endpoints are determined in a sequential manner.

Regardless of the method used, problems arise in the definition of edges and corners. As shown in Fig 6, the definition of linear regions must not contain extraneous control points. Doing so results in a curved structure and the loss of corners as all control points influence the spline. To circumvent this problem, the linear stretches of a contour must be isolated and either defined with a separate spline or reinforced with numerous control points to ensure linearity. Limitations in our solid modeling software allowed us only one continuous spline per contour. Therefore, we used control point reinforcement. To perform the control point selection, we used a modified one-pass merging scheme in conjunction with a line-searching algorithm.

DISCUSSION OF THE ALGORITHM

An overview of the solid model creation process is shown in Fig 7. The three-dimensional entity to be transformed into a solid model consists of a serial set of N × N raster gray scale images. Thresholding and edge-detection methods are used to extract object borders.[15] Each border is subsequently processed using a tracking routine to sequentially store the *x-y* coordinates of successive edge pixels.

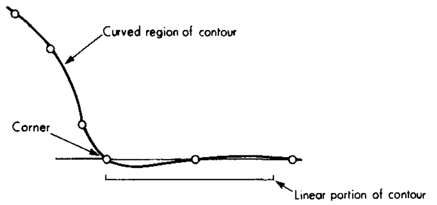

Fig 6. Error in linear segments produced by inclusion of extraneous control points. Due to the interpolative nature of cubic splines, both the straight-edge and the corner point are lost when extraneous, nonlinear control points are included in the definition of a linear region.

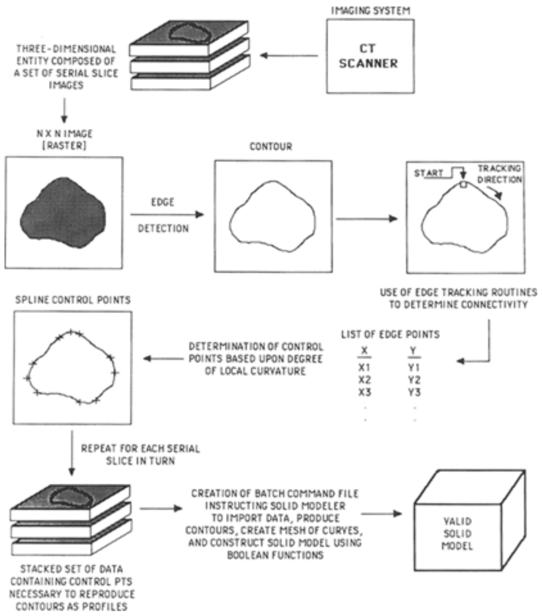
IMAGING SYSTEM
CT SCANNER
THREE-DIMENSIONAL ENTITY COMPOSED OF A SET OF SERIAL SLICE IMAGES
N X N IMAGE [RASTER]
EDGE DETECTION
CONTOUR
START
TRACKING DIRECTION
USE OF EDGE TRACKING ROUTINES TO DETERMINE CONNECTIVITY
LIST OF EDGE POINTS
X Y
X1 Y1
X2 Y2
X3 Y3
DETERMINATION OF CONTROL POINTS BASED UPON DEGREE OF LOCAL CURVATURE
SPLINE CONTROL POINTS
REPEAT FOR EACH SERIAL SLICE IN TURN
STACKED SET OF DATA CONTAINING CONTROL PTS NECESSARY TO REPRODUCE CONTOURS AS PROFILES
CREATION OF BATCH COMMAND FILE INSTRUCTING SOLID MODELER TO IMPORT DATA, PRODUCE CONTOURS, CREATE MESH OF CURVES, AND CONSTRUCT SOLID MODEL USING BOOLEAN FUNCTIONS
VALID SOLID MODEL

Fig 7. Overview of solid model creation from serial slice data.

As described above, polygon approximation methods assign representative segments based upon the intrinsic equation of the curve. An intrinsic function was defined by Duda and Hart[15] that expressed the tangent angle of a contour as a function of its arc length. We used this function to monitor the degree of nonlinearity (or deviation from linearity) throughout the border. This function is approximated at each pixel by calculating the slope of a line segment connecting two pixels a certain distance preceding and following the pixel of interest. By moving from one pixel to the next around the contour, progressive changes in the local slope are used to measure the degree of curvature change over a specified arc length.

The algorithm proceeds as shown in Fig 8. When a local slope is calculated at a new pixel location, it is first compared with the previous slope to see if a linear region exists. If the region is not linear, the current slope is then compared

```
Define Functions:
THETA(x)              Local slope of contour at pixel x
                      (where x = 1 -> number of pixels in contour)

Define Variables:
DELTA_PREV_THETA      = ABS [THETA(previous pixel) - THETA(present pixel)]
DELTA_LAST_THETA      = ABS [THETA(locus where last control pt found) - THETA(present pixel)]
PIXEL#                = Current pixel location within contour being analyzed
THRESHOLD             = Minimum contour slope change needed to designate a control point

Pseudo code:
WHILE (PIXEL# < Number of pixels in contour) DO
  Calculate THETA(PIXEL#)
  Calculate DELTA_LAST_THETA and DELTA_PREV_THETA

  IF (DELTA_PREV_THETA not equal to 0.00) THEN
    ----- Currently within a non-linear section ---------------
    IF (DELTA_LAST_THETA > THRESHOLD) THEN
      Drop a spline control point at the current PIXEL#
      Record current THETA for use in DELTA_PREV_THETA calculations
    ENDIF

  ELSE
    ----- Currently processing a linear section ---------------
    Drop spline control point at the start of the straight segment
    WHILE (THETA remains the same) DO
      Advance PIXEL#
      Calculate THETA(PIXEL#)
    ENDWHILE
    Drop spline control point at the end of the straight segment
    Intersperse several control pts between start and end of segment to
      enforce linearity
  ENDELSE

  PIXEL# = PIXEL# + 1
ENDWHILE
```

Fig 8. PSEUDOCODE: Spline control point placement.

with the slope at which the last control point was assigned. If the change in slope is greater than an established threshold, a new control point is assigned at that pixel location and its slope is recorded. The next pixel is then processed.

To reiterate, it is crucial that linear border segments be isolated. If the slopes calculated at two consecutive pixel loci are identical, a linear region is present. A line search is then carried out by simply advancing the segment until the measured slope changes. The starting and ending pixel loci of the straight segment are stored. At this point, either a new linear spline is created between the designated pixels, or a multitude of control points are defined to force linearity. The slope of the segment is recorded and the next pixel location is then processed. The routine continues as before until the entire border has been traversed.

Scattered throughout the border are nonlinear regions where minimal curvature change exists. By polygon approximation, few control points would be defined. To ensure adequate representation, a back-up routine was implemented to select additional control points if the distance between successive control points exceeded a predetermined value.

To produce a mesh of curves, an orthogonal set of contours is required. Figure 9 depicts how these contours are defined. Each border is first surrounded by a box. Border points intersecting with planes bisecting the box are used as control points for the orthogonal contours. Because serial slice data is usually anisotropic, having far greater resolution within each slice than between successive slices, control points are selected from every border to ensure an accurate reproduction.

Once all control points needed to produce the mesh of curves are defined, an instruction set is created as a sequence of batch commands to interface with the solid modeling software and produce the solid model.

COMPUTATIONAL METHODS AND RESULTS

Although any image could be used, our interests were in the creation of solid model representations of medical image data. A series of computed tomography (CT) slices of the wrist was used as a test image for solid model importation (Fig 10). Thresholding and edge detection were applied to extract outlines of the carpal bones that were then subjected to the polygon approximation routine. Control points were defined along the contour wherever the local slope changed by greater than 20°. Numerous control points were also used to reinforce linear edges as well as cornerpoints. Figure 11 shows an example of the borders and control points produced from various carpal bones in a single slice. The software necessary to perform these tasks was written in interactive data language (IDL)[19,20] and implemented on a DEC microVAX 3600 (Digital Equipment Corporation, Boston, MA). The entire process was automated, allowing the user to merely select which components to form as solid models. Approximately 15 minutes of central processing unit (CPU) time were required to process all the carpal bones contained in the dataset.

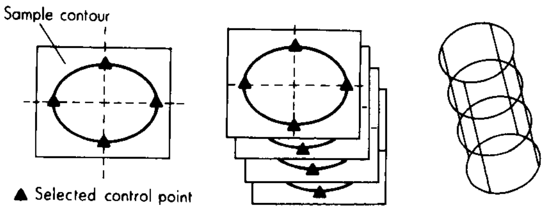

Fig 9. Selection of the orthogonal set of control points. Each contour is first placed within a box. Control points are selected where bisecting planes intersect the contour border. The four points from each successive slice are then used to form four vertical splines. These become the orthogonal set of splines necessary to produce the mesh of curves.

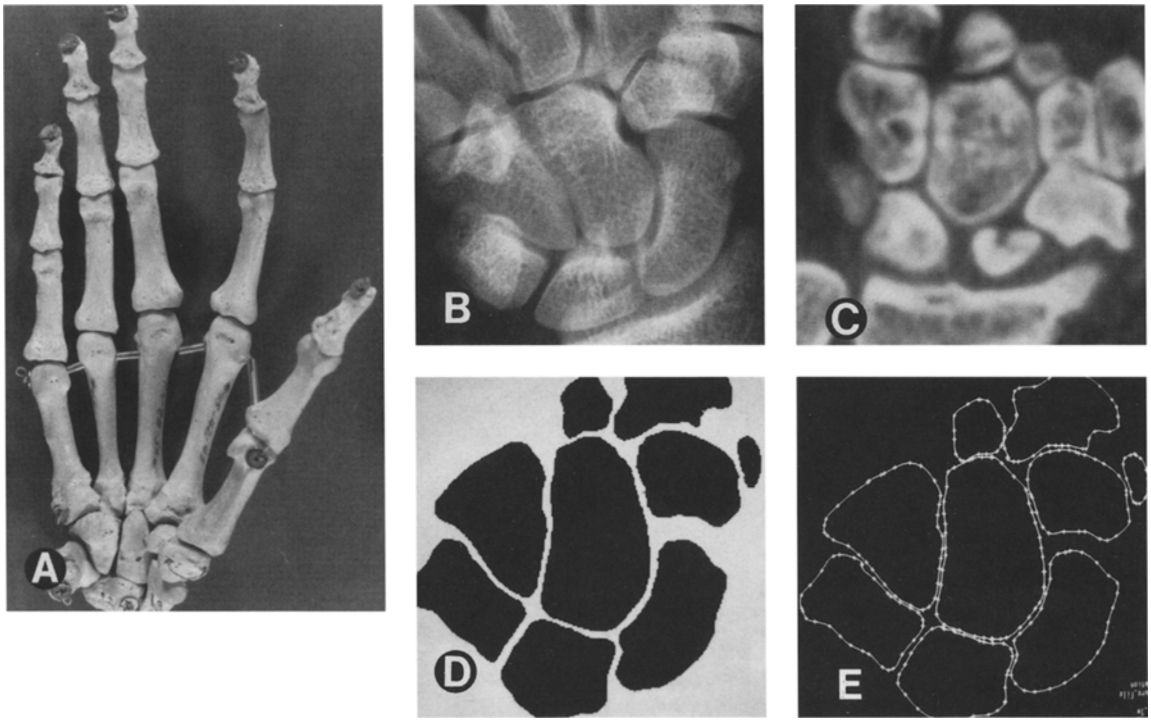

Fig 10. Extraction and cubic spline reconstruction of carpal bone contours. (A) The entire bony hand and wrist from a skeleton. A magnified plain wrist X-ray (B) reveals the individual carpal bones. A high resolution wrist CT scan slice (C) gives a detailed cross-sectional view of the carpal bones from which contours are extracted by thresholding and edge tracking (D). From these contours, suitable control points are selected and imported into the solid modeler to re-create each contour as a cubic spline approximation (E).

The I-DEAS (Integrated Design Engineering Analysis Software [Structural Dynamics Research Corporation, Milford, OH])[21] solid modeling package was used to create the solid models in this study. A batch file that contained the necessary I-DEAS instructions instructions was produced to reconstruct the mesh of curves, skin groups, and solid models. The selected control points were used by I-DEAS to form the cubic spline reconstructions shown in Fig 10E. Figures 11 and 12 display the result of the entire process as the carpal bones contained within the CT data have been imported and defined as solids.

By representing the wrist bones as an assembly of solid components, their separation to form an exploded view (Fig 12B) or to calculate mass properties from individual bones (Fig 12C) is possible.

DISCUSSION

The use of control points and splines to reconstruct contour shapes in itself reduces resolution. However, short of entering the complete set of border points into the solid modeler, our methodology provides an accurate reproduction. Polygon approximation methods ensure the proper selection of control points to fully define regions of high curvature. Likewise, the implementation of a line-searching algorithm prevents the loss of edges and corners.

The software we have developed provides an interface between image processing and solid modeling. Several structures can be interactively edited to correct imperfections and then selected for solid modeling. This interface allows the simultaneous creation of multiple objects while maintaining precise spatial relationships. This is especially important with biomedical data. In industry, component parts are oriented with respect to one another to produce an assembly. Because these objects are created from primitives, the orientation process is fairly straightforward. A piston rod is inserted into the shaft of a cylinder in the assembly of an engine. Two planar surfaces become juxtaposed in the creation of a joint. In contrast, the irregular shapes of biomedical objects are much

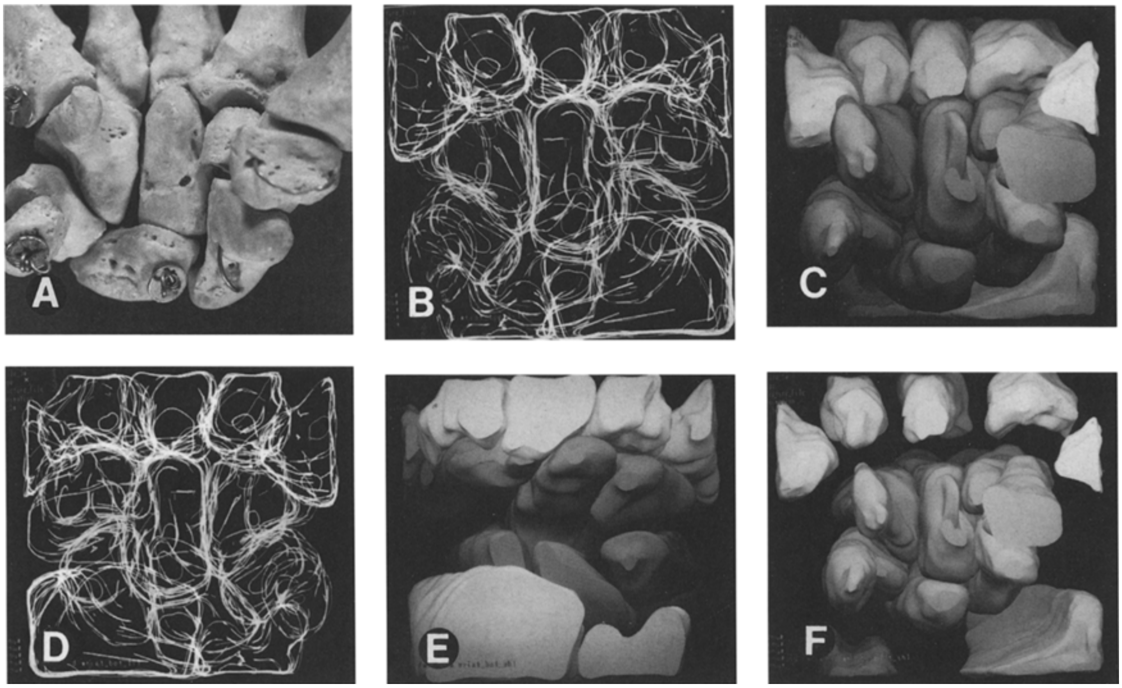

Fig 11. Solid model reconstruction or carpal bone assembly. (A) A magnified palmar view of the carpal bones (Fig 10E) Individual cubic splines are reconstructed into a solid model assembly as shown in B (wireframe, palmar view); C (shaded, palmar view); D (wireframe, dorsal view); and E (shaded, dorsal view). Because the solid model is fully defined and unambiguous, it can be manipulated to produce more informative views as shown in F (exploded view with retracted adjacent bones, eg, metacarpals, ulna, and radius).

more difficult to maneuver, and their relative positions are not intuitively obvious. Thus the preservation of spatial relationships is paramount, especially in the creation of complex biomedical assemblies.

With our solid model, we can begin to investigate numerous physical attributes. Immediately accessible are calculations to determine surface area, volume, density, mass, center of gravity, moments of inertia, and principal axes. Distance and angular measurements between selected object features can be performed. In addition, Boolean operations and interference checking allow us to study relationships between different objects. Finally, finite-element analysis enables us to study other processes

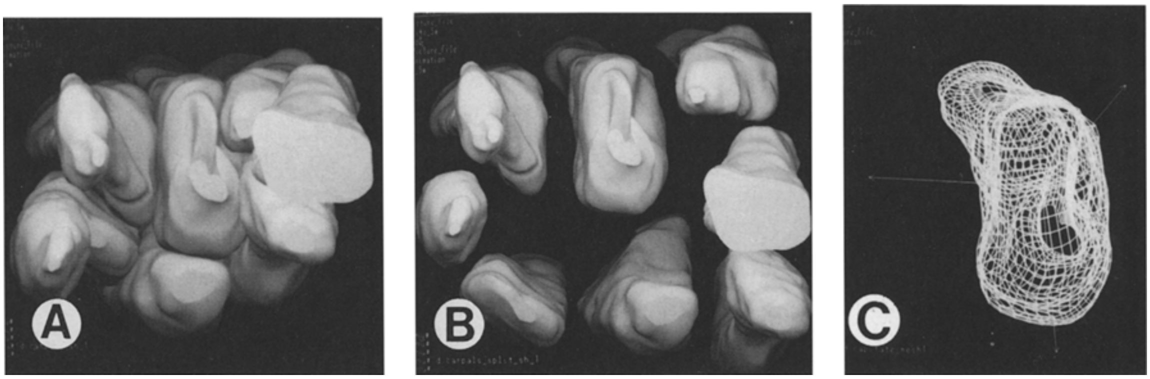

Fig 12. Critical evaluation of individual components. (A) The isolated assembly of carpal bones. (B) An exploded view separating each component. Individual bones can be evaluated and their internal properties displayed. (C) The principal axes (origin is at the centroid) superimposed upon a wireframe model of the capitate bone.

including the measurement of stresses from applied forces.

Serial slices of body regions are available as CT and magnetic resonance imaging scans. The slices often include well defined organs whose boundaries or edges may be readily detected visually in each slice. The synthesis of complete and unambiguous mathematical representations known as solid models from these serial slices offers advantages in the analysis and manipulation of complex organ geometries. These solid models may be used in morphometry, therapy planning and evaluation, visualization, and analysis.[22]

CONCLUSION

An automated technique that transforms complex image data in the form of cross-sectional slices into valid boundary representation (B-REP) solid models was developed. Raster gray scale CT scan image data from each slice was processed to extract edges, principally by thresholding. The edges were linked into closed planar contours, approximated by cubic splines in two dimensions. The surface of each gray scale object in the scene was defined by triangulation between adjacent spline curves. The surface was closed at the top and bottom, and a valid solid model generated. This routine was intended as an interface between image processing and solid modeling software, and can construct simple B-REP solid models from serial slice data.

ACKNOWLEDGMENT

The authors wish to thank Dr Louis A. Gilula for his invaluable input in computed tomography of the wrist.

REFERENCES

1. Mantyla M: An Introduction to Solid Modeling. Rockville, MD, Computer Science Press, 1987, pp 1-11

2. Foley JD, van Dam A, Feiner SK, Hughes JF: Computer Graphics Principles and Practice (ed 2). Reading, MA, Addison-Wesley, 1990, pp 533-562

3. Duncan JP, Mair SG: Sculptured Surfaces in Engineering and Medicine, New York, NY, Cambridge University Press, 1983, pp 20-41

4. Kim Y, Luo RC: Surface Reconstruction Based on Descriptions of Cross-Sectional Contours. SPIE (Sensing and Reconstruction of Three-Dimensional Objects and Scenes) 1260:191-198, 1990

5. Samet H: The Design and Analysis of Spatial Data Structures. New York, NY, Addison-Wesley, 1990, pp 316-318

6. Mortenson ME: Geometric Modeling, New York, NY, Wiley, 1985, pp 372-430

7. de Boor C: A Practical Guide to Splines. New York, NY, Springer-Verlag, 1987, pp 63-72

8. Ballard DH, Brown CM: Computer Vision. Englewood Cliffs, NJ, Prentice-Hall, 1982, pp 119-146

9. Han MH, Jang D, Foster J: Identification of Cornerpoints of Two-Dimensional Images Using a Line Search Method. J Pattern Recognition 22:13-20, 1989

10. Han MH, Jang D, Foster J: Inspection of 2-D Objects Using Pattern Matching Method. J Pattern Recognition 22:567-575, 1989

11. Attneave F: Some Information Aspects of Visual Perception. Psychol Rev 61:183-193, 1954

12. Shirai Y: Edge Finding Segmentation of Edges and Recognition of Complex Objects. Los Altos, CA, Kaufmann, 1975, pp 674-681

13. Tomek I: Two algorithms for piecewise linear continuous approximation of functions of one variable. IEEE Trans Comput 23:445-448, 1974

14. Roberts LG: Machine Perception of Three-Dimensional Solids, in Tippett JP (ed): Optical and Electro-optical Information Processing. Cambridge, MA, MIT Press, 1965

15. Duda RO, Hart PE: Pattern Recognition and Scene Analysis. New York, NY, Wiley, 1973, pp 338-377

16. Ramer U: An Interactive Procedure for the Polygonal Approximation of Plane Curves. Computer Graph Image Proc 1:244-256, 1972

17. Lowe DG: Three Dimensional Object Recognition from Single Two Dimensional Images. Artif Intell 31:355-395, 1987

18. Rosin PL, West GAW: Segmentation of Edges Into Lines and Arcs. Image Vision Comput 7:109-114, 1989

19. IDL (Interactive Data Language): Denver, CO, Research Systems, 1990

20. Also available as PV-WAVE Boulder, CO, Precision Visuals, 1990

21. I-DEAS (Integrated Design Engineering Analysis Software). Milford, OH, Structural Dynamics Research Corporation (SDRC), 1990

22. Sutherland CJ, Bresina SJ, Gayou DE, Blair VP III: Application of Computer-Assisted Engineering (CAE) Software to Orthopedic Surgical Planning. SOMA Engineering for the Human Body 1990