Dynamic Display and Quantitative Analysis of Three-Dimensional Left Ventricular Pathology

William A. Barrett
william_barrett@byu.edu

Jayaram K. Udupa

Follow this and additional works at: https://scholarsarchive.byu.edu/facpub

Part of the Computer Sciences Commons

Original Publication Citation

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/facpub/1200
DYNAMIC DISPLAY AND QUANTITATIVE ANALYSIS OF THREE-DIMENSIONAL LEFT VENTRICULAR PATHOLOGY

William A. Barrett* and Jayaram K. Udupa**

* Department of Computer Science, Brigham Young University, Provo, Utah 84602
** Medical Image Processing Group, Department of Radiology, University of Pennsylvania, Philadelphia, PA, 19104

Summary

Techniques have been developed for automated extraction and dynamic interactive display of three-dimensional (3D) left ventricular (LV) surface anatomy from Cine CT images using a PC-based image display architecture. Images of both endocardial and myocardial surface anatomy are generated from multiple views at multiple time instances to demonstrate various LV pathologies including apical akinesis, apical and posterior aneurysms, LV Failure, IHSS, and a left atrial myxoma. Surface generation requires interpolation between scans, surface tracking, and rendering. Generation of 60 views corresponding to a single time instance requires approximately 15 minutes. LV dimensions are measured between two or more surface points using one or more views. Dynamic interactive display and measurement of surface anatomy provides a valuable tool for improved understanding and quantitative analysis of LV shape, size, and dimensions in the presence of these pathologies.

Methods and Materials

Scanning geometries for CT and Cine CT are diagrammed in Figure 1. The conventional CT geometry (left) consists of a fixed ring of detectors, a mechanically rotating tube gantry, and requires between 1.5 and 8 seconds to complete a scan. The Cine CT apparatus consists of scanning electron beam and a magnetic deflection coil used to steer and focus the beam through a 210 degree arc across one of four semicircular target rings, producing X-rays which are recorded by one of two semicircular detector rings (Figure 1, right). The Cine CT geometry produces two adjacent scans simultaneously, 50 msec/scan, allowing an 8-level study (10 images/level) to be completed over four sequential heart cycles.

Figure 1. Scanning Geometries for CT and Cine CT.

Pathologies were selected which demonstrated abnormal LV shape or size at a given time instance or over the course of the cardiac cycle. These included apical akinesis, apical and posterior aneurysms, LV Failure, IHSS, and a left atrial myxoma. All studies were performed through intravenous injection of contrast material. The six software modules listed above were implemented on a Virtual Imaging VIEW 2000 system which is a PC-based machine with a special purpose tri-bus architecture diagrammed in Figure 2.

The View System contains 16 Mbytes of video memory (2048x4096x16 RAM) and two 1 Mbyte display buffers. The 64-bit image (video) bus and the 32-bit display bus provide internal image read/write capability at 112 Mbytes/sec and 32 Mbytes/sec respectively. The high-speed display architecture allows interactive selection and frame-rate display of individual views, producing the effect of arbitrary tilt and rotation of the surface anatomy. The tri-bus architecture also allows high speed pipeline processing (zoom or shrink in factors of 2 using bilinear interpolation, and panning) from image memory across the 64-bit video bus to the display buffers via the 32-bit display bus, with image output through multiple windows or multiple independent buffered displays. The CPU is the 80386 processor with a 387 coprocessor.

Algorithms for creation and measurement of three- and four-dimensional LV surface anatomy are described below.
Figure 2. View System 2000 tri-bus architecture.

Generation of 3D and 4D LV Surfaces

Generation of LV surfaces involves five sequential tasks described below. These tasks include scaling and formatting of the original scan data into a binary cubic voxel array, surface tracking and rendering, and creation of an interactive display matrix of views.

Scaling

Spatial or dimensional scaling is optional but may be desirable to optimize the resolution of the resulting 3D surfaces. The maximum resolution of the 3D surfaces is 256x256x256. Thus, if the maximum resolution of the region of interest is less than 256, the region can be magnified up to 256 in any given dimension using bilinear interpolation described in Figure 3, below.

\[ I(x,y) = \frac{(1-A)a + Ad}{1-B} \]

Figure 3. Bilinear Interpolation.

Magnification using bilinear interpolation is performed by subsampling (imposing a finer screen on) the grid. Let a, b, c, and d represent the densities of four adjacent pixels in the original image, and j, i the discrete coordinate position of pixel a in the x and y dimension respectively. The density I(x,y) at a new real coordinate position (x,y) can be expressed as a function of a, b, c, and d by defining A as the distance between the real coordinate value x and its nearest discrete counterpart, j (equation 1) and V1, V2 as the linear interpolants across a, b and c, d (equations 3 and 4). I(x,y) (equation 5) is then the result of applying the distance B (equation 2) to V1 and V2. Minification is handled in a similar manner.

Intensity scaling is generally necessary because the brightness produced by the blood-contrast mixture varies over time. In Figure 4, each pair of vertical bars represents the time instance at which two adjacent slices are acquired. Since adjacent slices are acquired simultaneously, the brightness in the left ventricle will be approximately the same for each pair but will vary between successive pairs due to the time-dependent brightness of the underlying contrast curve. Since the ventricular surface for each pair is defined by the brightness thresholds b1, b2, b3, and b4, brightness levels must be equivalenced to obtain a single threshold for the entire volume. This is done by selecting a given level (say b1) and multiplying densities in the remaining image pairs by b1/b2, b1/b3, and b1/b4 respectively.

Creation of a Binary Cubic Voxel Array

Since the x,y (in-plane) resolution is generally much finer than the z-axis (scan-increment) resolution, image planes are interpolated between the original scan data at data planes (resulting in a cubic voxel array) using the x,y resolution to produce a cubic voxel array filling the imaged volume. A brightness threshold is then applied to each element in the array to separate object (1's) from background (0's). Adjacent object and background elements are then packed (16 per word) for economy of storage and processing.

Because of the computationally intensive nature of these tasks, a new algorithm was produced which performs the interpolation, thresholding, and packing in a single step. The main idea of the algorithm is to generate a bit pattern which represents the interpolated, thresholded, and packed binary voxels between two adjacent scan positions.

Consider the voxel planes corresponding to two adjacent scan positions i and i+1 in Figure 5 and let A and B represent the voxel densities at corresponding locations in each scan. Suppose, for example, the voxel densities represented by A and B are 96 and 102 respectively, and that nine additional slices between i and i+1 must be interpolated. The linearly interpolated densities, v(j), are shown along the diagonal in the middle of Figure 5. Suppose the object of interest is identified by voxel densities ranging between two threshold values t1 and t2, where t1=100 and \( t_2 = 255 \). Application of these thresholds to the interpolated line would result in voxels having densities less than 100 being set to 0 (shaded) and those with densities greater than or equal to 100 being set to 1. Rather than thresholding and then packing individual values, interpolation, thresholding, and packing can be done directly using bit arithmetic to create a bit pattern whose length is that of the line, with 0's corresponding to background and 1's corresponding to the object, as illustrated at the bottom of Figure 5.

Surface Tracking

The purposes of surface tracking is to isolate one "connected" surface of the object for subsequent display as a shaded-surface rendition. The binary cube array obtained as described previously contains a representation of the object: the 1-cubes constitute an elemental volume of the object of interest (the myocardium for example) and the 0-cubes that of the space outside the object. Due to a variety of reasons, mostly related to imper-
spatially adjacent values \( A \) and \( B \) from original scans at positions \( i \) and \( i+1 \) linearly interpolated values, \( v(j) \), between original scan values \( A = 96 \) and \( B = 102 \).

Figure 5. Computation of a bit pattern representing interpolated, thresholded, and packed values between original density values \( A \) and \( B \).

Effect segmentation, not all 1-cubes always constitute the object of interest, which implies that a subset of the set of 1-cubes has to be somehow identified to isolate the object of interest.

A paradigm that is found very useful in this regard\(^2\) uses the notion of a connected component of 1-cubes to define the object. A connected component is a maximal subset of 1-cubes such that every pair of cubes in the component are connected by 1-cubes. Maximality implies that there are no 1-cubes outside the component that are connected to some 1-cube in the component. A surface of interest is then simply a boundary of the object. A boundary of the object is the set of all faces of 1-cubes in the object that separate a connected component from the complement of the object (0-cubes). Figure 6 illustrates the notion of an object and its boundary.

Of course, for visualization, only a specified surface is needed and not the entire object (set of connected 1-cubes) bounded by the surface. Given a binary cube array, the algorithm outlined below extracts a surface, starting with a single face belonging to the surface, without extracting the entire object from the binary array.

Imagine that every 1-cube in the object has two circuits wrapped around it as in Figure 7a. This mechanism assigns vectors to the faces in the surface as in Figure 7b which results in a (directed) graph laid out on the surface. Each face in the surface represents a node in the graph and the vectors represent (directed) arcs. The surface tracking procedure simply consists of, starting from a specified face, following the arcs in search of new faces. The idea is to visit all faces in the surface in an efficient way (a typical myocardial surface may be composed of several hundreds of thousands of faces). Note that some faces (usually a small fraction of the total number of faces in the surface) have two vectors assigned to them. Such faces (called type-z faces) not only lead to two new faces but also can be reached from two other faces. A majority of the faces have one vector assigned to them implying that they lead to exactly one new face and that they can be reached from exactly one face.

The algorithm marks a type-z face the first time it is visited so that when it is visited the second time, the faces to which it leads are not discovered again. Of course, the algorithm also keeps track of open ends (faces from which new faces are to be discovered). When no more open ends remain, the algorithm terminates and the faces output exactly represent the surface that contains the starting face.

Figure 7. (a) A cube with two circuits wrapped around. (b) an object made up of four cubes with the vectors assigned to the boundary faces (by the mechanism of wrapping circuits) shown. Note that only faces perpendicular to the z-axis have two vectors, and that all the rest have only one vector. Surface tracking consists essentially of spanning the graph formed by the vectors on the surface.
Surface Rendering

Given a surface (such as that of the myocardium), the purpose of surface rendering is to create a depiction of the surface on a two-dimensional display screen. There are two major aspects to this depiction: removal of hidden parts of the surface and shading of visible parts. Both of these operations are made computationally efficient because of certain properties of the geometry of discrete surfaces.

Hidden Part Removal. Suppose each face in a surface is grouped based on the direction of the face (the vector joining the center of the 1-cube and the 0-cube). Then we have six groups, each corresponding to one of the six directions x, y, z, and -x, -y, -z. For any fixed viewing angle, the faces in at least three of these groups are not visible.

Another important property of the faces in a discrete surface is that just the distance of the center of the faces from the viewpoint is sufficient to determine visibility using a z-buffer technique and the faces do not need to be sorted. If the scale factor that determines the size (XxY) of the rendered image is chosen such that the largest possible projection of a face falls completely inside a pixel in the screen (image), then the expensive scan conversion problem is totally eliminated. Combining these two properties, the z-buffer technique consists simply of first determining the normal at the center of the face corresponding to the desired view of the surface, determining the pixel into which the face projects, and updating the pixel value in the image buffer to the shading assigned (see below) to the face, provided the distance of the face from the viewpoint is smaller than the value of the corresponding pixel in the z-buffer. (The image- and z-buffer are XxY arrays which are initialized, respectively, to zero and a large value. Both of these operations are done in constant time.) In the latter case, the value of the pixel in the z-buffer is updated to the distance of the face.

Shading. The shading or intensity associated with a face is determined by the lighting conditions under which the surface is viewed and the optical properties assumed for the surface. The light reflected in the direction of the viewpoint is determined by the normal to the surface estimated at the center of the face. Proper estimation of face normals is crucial to the accurate depiction of the original organ surface. One approach is to use the discrete surface itself to determine the normal. Observing that each face in the surface has four "neighbors" that share one of the edges of the face, the orientation of the face and that of its neighbors can be used to estimate the normal. The images shown in this paper are computed using such an approach. An alternative method is to use the original grey-level image to estimate a normal at the center of the face by fitting an interpolation function to the grey-level distribution and using the normalized gradient of that function. This approach generates exquisite images which carry very subtle details unobservable in images generated using normals based on the geometry of the surface.

Dynamic Interactive Display of 3D LV Surface Anatomy

Dynamic interactive display of 3D LV surface anatomy is done through the use of a display matrix of precomputed views. Each precomputed view, v, is a function of the tilt and rotation angles (\(\alpha\) and \(\beta\)) which specify its orientation:

\[
v = f(\alpha, \beta)
\]  

(6)

The display matrix, consists of \(m \times n\) (m tilts, n rotations) precomputed views \(v_{ij}\) where the index \(i\) corresponds to a given tilt, \(\alpha_i\), and the index \(j\) to a given rotation, \(\beta_j\), at that tilt. Thus, each row is a rotational sequence of views at a given tilt. In the case of 4D anatomy, the display matrix takes on an added dimension of time such that

\[
v_{ijt} = f(\alpha_i, \beta_j, t)
\]  

(7)

However, rather than compounding the number of views that must be generated with the time element, \(t\), it is convenient to use only one tilt and replace \(m\) with the time increment \(t\). The result is still a two-dimensional display matrix (t x m) of precomputed views consisting of rotational increments (columns) of which varies over time increments (rows).

The architecture described previously permits dynamic display of LV surfaces through multiple windows simultaneously. This means that all three parameters (tilt, rotation, and time) can be incorporated into the display by using a separate window for the additional parameter. For each anatomy can be examined from multiple projections (different tilts, rotations, or time instances) simultaneously.

Measurement of Surface Anatomy

Once a surface is available and a method of rendering the surface is devised, a variety of measurements can be carried out on the surface. It is most natural to directly interact with the 3D rendition of the surface to probe the measure. Volume, three-dimensional distance between points on the surface, angles formed by three or more points specified on the surface, measures with respect to user-defined anatomic planes, and local curvature are all easily computed at interactive speeds (even on modest computers). Of course, in order to use such an "electronic scale" effectively, any desired view of the surface should be computable at interactive speeds (not more than a couple of seconds). Display generation, however, requires far more time than the measurement operation, and to achieve true interactive surface display, the rendering algorithm should be implemented on a workstation with at least 3-4 MB of memory and sufficiently fast (5-10 Mflops) floating point operation.

Volume enclosed by the surface is easily computed (with just one arithmetic addition for each face perpendicular to a fixed axis) as a byproduct of the surface tracking algorithm. Suppose the faces perpendicular to the x-axis are numbered, then a face pointing in the -x direction, its x-coordinate is subtracted from the current value of volume (which is initially set to zero) and the x-coordinate is added for the +x direction. The volume so computed gives the number of cubes enclosed by the surface.

The basic capability required for the other types of measurements is the ability to represent the 3D distance between any two points on the surface which is specified via the (x,y,z) coordinates of each point. Suppose a point (x, y, z) is indicated in an image depicting one view. If that point is assumed to be on the surface, the z-buffer value, say \(z_1\), at \((x_1, y_1)\) gives the z-coordinate of the point on the surface. Suppose that a second point \((x_2, y_2, z_2)\) is specified similarly in a subsequent view. Then the distance between the two points is computed by (forward) transforming \((x_1, y_1, z_1)\) to \((x_1', y_1', z_1')\) so that it is represented in the same (image) space in which \((x_2, y_2, z_2)\) is specified. It is now trivial to compute the distance between \((x_1', y_1', z_1')\) and \((x_2, y_2, z_2)\). It is also clear how to compute the angle between any two line segments or to measure distance, angle, etc. with respect to an anatomic plane specified by three (non-coplanar) landmarks indicated on the image.

To measure curvilinear distance between any two points, a sequence of points (not necessarily all in the same view) between these points have to be deposited on the surface. The curvilinear distance is then simply the cumulative distance between successive points in the sequence. The procedure is essentially equivalent to laying a thread around the surface between the two points and measuring the length of the thread.
Figure 8. LV dimensions, RAO, end-diastole.

Figure 9. (a) end-diastole (b) end-systole.

Figure 10. Apical and posterior aneurysms, Lateral view, end-systole.

Figure 11. (a) end-diastole (b) end-systole.

Figure 12. (a) end-diastole (b) end-systole.

Figure 13. IHSS - LV myocardial surface (left and right half). (top) end-diastole (middle) end-systole (bottom) normal myocardium.
Results and Discussion

Cardiac surfaces for the pathologies mentioned previously are shown in Figures 8 - 13. Endocardial surfaces appear as a blood-contrast "cast" and are analogous to a 3D angiographicogram. Myocardial surfaces are created by geometrically dissecting the object (myocardium) into a right and left half.

Figure 8 (apical akinesia) shows the LV endocardial surface in the RAO projection at end-diastole. Dimensional measurements were obtained through interactive selection of surface points as described above. The long axis was determined using two surface points (both from this view) at the position of the mitral valve plane (upper left) and the apex (lower right). The short axis was determined from an additional surface point in the LAO projection. Long and short axes at end-systole were 7.04 cm and 4.58 cm respectively. The end-diastolic volume enclosed by the surface (calculated from the surface description) was 211 cc, but includes a portion of the left atrium, shown in the upper left, and the aorta, shown near the middle with two of the aortic valve cusps in front. The end-systolic volume of the surface in this same projection was 133 cc, but also included the aorta and a substantial portion of the left atrium. As a result, volumetric measurements for this surface are not directly useful for quantitation of LV function.

Both right and left ventricular surfaces appear in Figure 9. An apical aneurysm of the left ventricle is indicated in Figure 9a where the chambers appear in the anterior septal view at end-diastole. The aorta is seen extending from the left ventricle near the middle of the image with the left atrium in the background. The aneurysm is more apparent in Figure 9b where both chambers are shown in the right anterior view at end-systole. An aneurysm of both the apex and posterior wall is shown in the lateral view of the left ventricle at end-systole in Figure 10. The imaged volume intersects the mitral valve plane at the top where a small portion of the left atrium is seen above the large bulging posterior aneurysm on the right. However, the imaged volume does not include the entire left ventricle, due primarily to the apical aneurysm which is prematurely truncated by the bottom imaging plane (lower left). The indentation in the middle of the chamber is an impression in the "cast" made by the papillary muscle. When viewed dynamically, the anterior wall contracts vigorously, expanding the apical aneurysm, while there is very little change in the shape or size of the posterior aneurysm. Dimensional measurements for the chamber or the aneurysms can be obtained in a manner similar to those illustrated in Figure 8.

Figure 11 shows the left ventricle in the lateral projection at end-diastole (a) and end-systole (b), and supports the diagnosis of left heart failure. The imaged volume again intersects the mitral valve plane at the top with a small portion of the left atrium seen posteriorly at the upper right. The height of the imaged volume (6.8 cm) is fixed by the geometry of the Cine CT scanning apparatus and again fails to encompass the entire chamber, this time due to the enlarged size of the ventricle. This is evidenced by the "clipping" effect of the bottom imaging plane. When displayed dynamically, the left ventricle demonstrates a greatly enlarged volume but uniformly depressed function and volume changes over the cardiac cycle. Wall motion is similarly depressed except for a more strongly contracting posterior region which causes the akinetic anterior wall to exhibit paradoxical motion.

IHSS is demonstrated in Figure 12 which shows both the right and left ventricles in the anterior view at end-diastole (a) and end-systole (b). The reduced chamber size and large septal gap are indicative of left ventricular hypertrophy. This is illustrated more dramatically through examination of the left and right LV myocardial surface anatomy in Figure 13. The top and middle pair show the myocardial geometry at end-diastole and end-systole, and demonstrate a dramatic thickening of the myocardium during contraction when compared with a normal myocardium (bottom). Accordingly, LV volume is greatly reduced and the septal region greatly enlarged at end-systole.

The atrial myxoma was attached posteriorly and appears as a cavity in the endocardial surface which extends through the mitral valve region into the ventricle during contraction. In all cases, the shape, size, and spatial interrelationships of cardiac structures are better understood through dynamic interactive display of endocardial and myocardial surfaces. The ability to interactively rotate the surface anatomy (as though it were held in the hand) provides motion parallax depth cues and is a natural, valuable aid in shape perception. Surfaces which are changing shape over time demonstrate inherent motion parallax, and when combined with interactive rotation, provide a powerful environment for analysis of cardiac anatomy.

The speed with which surfaces can be generated greatly influences the clinical utility of the techniques described above. Use of a 32-bit processor/coprocessor has reduced processing time by an order of magnitude. The new algorithm for creation of the binary cube array and the faster surface tracking algorithm have reduced processing time by an additional factor of four and two for their respective modules. As a result, 60 views of a typical surface can be generated in approximately 15 minutes, making clinical use of 3D and 4D surface analysis practical.

The greatest opportunities for improvement lie in quantitative analysis and general interaction with the 3D surface anatomy. Dimensional measurements can be obtained, but not at interactive speeds. Local storage of z-buffer parameters may help to overcome this limitation.

Object volumes are also easily obtained directly from the surface description, but are not always meaningful. When the left ventricle can be isolated through optimal positioning in the imaging apparatus or segmentation via threshold selection, then the object volume equals the chamber volume. However, the object volume frequently includes other attached structures (eg. atrium, aorta, right ventricle) and is therefore not always useful in quantifying LV function. As a result, techniques for interactive segmentation of the 3D object are being explored.

Conclusions

The methodologies described for definition, extraction, representation and display of 3D and 4D cardiac surface anatomy have been implemented on a low-cost, PC-based system. Use of faster processors and improved algorithms provide an efficient and economical tool for visualization and analysis of surface geometry. Measurement of anatomical dimensions, volumes, etc., through interaction with the 3D image is also possible but could benefit from techniques which would allow these measurements to be obtained with interactive speed. Measurement capability coupled with dynamic interactive display of surface anatomy provides improved understanding of 3D and 4D endocardial and myocardial shape, volume, and dimensional abnormalities in the presence of specific pathologies.

Bibliography