Non-Invasive Automatic Breast Volume Estimation for Post-Mastectomy Breast Reconstructive Surgery

G. Passalis I.A. Kakadiaris T. Theoharis M. Miller

Visual Computing Lab, Dept. of Computer Science, University of Houston, Houston, TX 77204
Dept. of Plastic Surgery, MD Anderson Cancer Center, Houston, TX 77030

Abstract

Automatic computation of breast volume data is required in post-mastectomy breast reconstructive surgery, where it is very useful to have an estimate of the volume of the tissue to be extracted in advance of the operation. Such an automated process is essential when conducting studies on large patient databases where manual volume estimation is be both tedious and subjective. In this paper, we present a non-invasive automatic breast volume estimation method. It employs 3D scanned data of the torso. First, a surface underneath the breast that resembles a 'breast-less' torso is automatically constructed, and then the volume of the breast is estimated as a function of the difference between the new surface and the actual breast. We have performed a number of experiments on both synthetic and real data with very encouraging results.

1 Introduction

Pre-operative knowledge of the volume of a breast would be particularly useful in breast reconstructive surgery, as it could allow clinicians to better prepare replacement tissue for women who will undergo a mastectomy operation. Currently, such volume is measured after the breast is removed by simply putting the extracted tissue in a vessel filled with liquid and measuring the displaced liquid volume. At the same time, breast studies are often conducted that involve breast volume estimation on large patient data bases. In both cases manual breast volume estimation is both tedious and subjective.

Recent improvements in 3D scanning methods and equipment make 3D data of the human body widely accessible to clinicians. Such data potentially allows the automatic execution of tasks which could only be executed manually until now. Breast volume estimation is a good example. Until now it is measured using an invasive method and there is great need for a method that can automatically and non-invasively estimate it.

In this paper, we present such a fully automated, non-invasive, method that measures the volume of the breast from 3D scanned data thus allowing the surgeon to know the volume before the actual mastectomy.

1.1 Related work

Riggoti et al [5] presented a method for scanning the surface of the torso and used it for various measurements. They presented two procedures for measuring the volume of the breast but they were both not automated and only returned an approximation of the breast volume. In the first instance they manually select a contour on the scanned surface that defines the boundary of the breast. They then connect the contour points with its center of gravity thus creating a closed volume. In the second instance, they manually place a regression plane gives a closed volume when intersected with the torso surface. In both cases the closed volume was considered equivalent with the volume of the breast. In reality the breast is not divided from the rest of the torso by a plane, since the tissues of the breast lie upon the rib cage which is curved.

Concerning commercial software available for this function, we experimented with the 3DMDPatient product from 3DMD. 3DMPatient is the software that we used to scan the subjects and except from scanning, it offers a number of measurements one of which is the volume of the breast. Its method has some similarities with the one proposed in this paper since it too uses four landmarks and connects them with a coon's patch. The placement of the landmarks is manual and they are connected with 'straight' lines on the torso surface. Because of the manual landmark placement and the not optimum path between them the result can be erroneous as shown in Figure 1. Additionally, this procedure takes several seconds to complete instead of the milliseconds our method takes. We measured the left breast of a subject before and after she had an augmenting surgery. In 3DMpatient we measured it using many different sets of landmarks and then computed the average (the cases where the result was obviously wrong

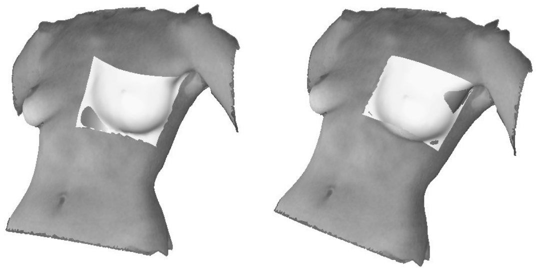

Figure 1. Illustration of the problems associated with the manual selection of landmarks.

did not contribute in the average). These results were: 184ml,169ml,238ml,279ml,and 226ml (mean 220ml) and 338ml,342ml,339ml (mean 340ml) for before and after the surgery respectively. For the same data our method returned 250ml and 357ml respectively.

2 Breast Volume Estimation Method

An overview of our method and a separate detailed description of each step is presented in this section. But first let us briefly discuss the required input data. A 3D triangular mesh is retrieved using a number of range cameras. The software that comes with the cameras is responsible for co-registering the different views in 3D. A second pass through standard graphics software is required in order to smooth the mesh, fill any holes and undo any irregularities that are present in the scanned data. Our algorithms expect that the above pre-processing steps have already been performed.

During the scanning process, the human subject should ideally place her hands on top of her head.

2.1 Overview

The main idea behind this method is to create a surface underneath the breast that resembles the torso so that we can compute the volume between this surface and the breast. Since our goal is a fully automated procedure, every individual step operates without any user intervention. There are six steps:

- *Alignment of the data:* The data are automatically aligned so that the Z and Y axes correspond to the ’front-back’ and ’top-down’ axes of the body.

- *Landmark placement:* Four landmark points are automatically placed on the torso which will guide the rest of the process.

- *Optimum path between landmarks:* The four landmarks are connected with the optimum path along the surface. The breast lies inside these four contours.

- *Surface construction:* An interpolating surface is built using the four contours. This surface represents the torso that is beneath the breast.

- *Volume estimation:* The volume between the constructed surface and the actual breast is computed.

- *Skin reduction:* (Optional). If information about skin thickness is available then the corresponding skin volume is subtracted from the total breast volume.

2.2 Alignment of the data

The data resulting from the scanning stage do not have a specific alignment, especially if they were pre-processed by another 3D program. Additionally, our algorithm should work with input data from any scanning method and therefore can not assume any a priori alignment. However, alignment is crucial to our method because we need to know where the front of the torso is, what is the orientation of the breasts, which way is the head etc. Our only assumption is that we are given 3D scanned torso data that do not include the backside.

We first compute the mean value of the data set vertices and translate our torso data so that the mean value is the center of the torso coordinate system. Next, we compute the average face normal by summing all the unit polygon normals and renormalizing. Since our data may be irregularly sampled, and in order to make our method resilient to sampling density, we weigh the polygon normals by the area of their polygons. The average normal computation then becomes:

$$\overrightarrow{N}_{avg} = \frac{\sum A_i \overrightarrow{N}_i}{\sum A_i} \qquad (1)$$

The idea is that since only the front side of the torso was scanned the average normal vector will have a ”forward” direction. Due to symmetry, the contributions of the normals from the left and right side will cancel each other. After this vector is computed it is used as the Z axis of the torso.

$$\begin{aligned} \mathbf{M}' =& \mathbf{R}_{\overrightarrow{AXIS}}(\varphi) * \mathbf{M} \\ & \text{where} \\ & \overrightarrow{AXIS} = \overrightarrow{N}_{avg} \times \overrightarrow{Z} \\ & \varphi = \arcsin(\overrightarrow{N}_{avg} \cdot \overrightarrow{Z}) \\ & \mathbf{R} \text{ is the matrix for rotation around an axis} \end{aligned} \qquad (2)$$

To complete the alignment process we need to identify a second axis. Since the torso data are obtained with the

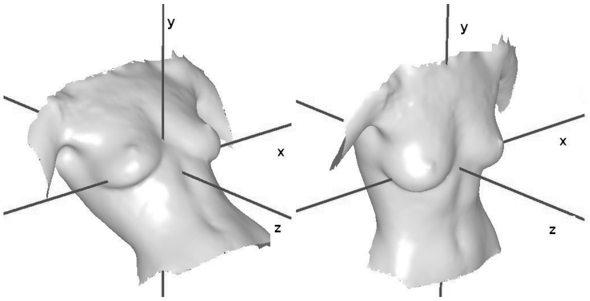

Figure 2. The three axes before and after alignment.

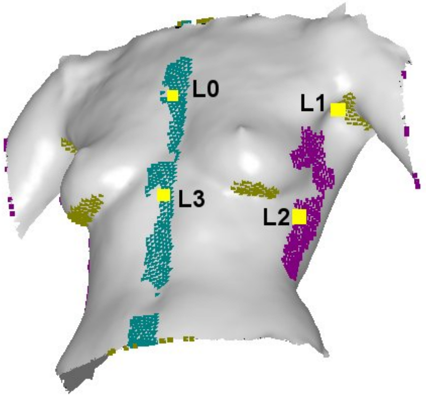

Figure 3. The 4 landmarks (bright yellow dots). Green, Purple and Yellow faces have +Z, +X and -Y orientation respectively

hands of the subject placed either on the waist or behind the head, we assume that maximum variance in the data will be found along the X axis. Thus, by performing principal component analysis[7] to our data (using only the x and y coordinates) we can find the eigenvectors. The first eigenvector will be along the direction with the highest variance and this can be considered as the X axis for our data. Since we have the X axis we can also compute the Y axis.

In the end, we will have the X,Y and Z axes of the data pointing to the 'right', 'up' and 'front' of the torso respectively as shown in Figure 2.

2.3 Landmark placement

In this step four landmarks need to be defined which will serve the purpose of defining four contours (when connected, see Section 2.5) which will in turn define a surface underneath the breast that follows the curvature of the torso (Section 2.6).

After experimenting with positioning the landmarks manually we found that the best positions are those shown in Figure 3 (bright yellow dots). These are not anatomical landmarks but where selected because:

- when connected they enclose the breast completely and tightly
- the connecting contours lie on areas of the torso not affected by the breast

As long as the positions of the landmarks fulfil these two conditions the optimum path algorithm presented in the next section will be able to connect them properly. Thus, the computed volume will not be heavily dependent on these positions, and will be the same even if their positions are slightly varied.

The landmark points are identified based on selecting faces from the mesh according to their normals and clustering them properly. We will describe this method for the left breast (for the right breast we simply mirror it). Three groups of faces are defined, one for the vertices whose normals have a front (+Z) orientation, one for those with left (+X) and one for those with down orientation (-Y). The selection is done using the dot product of the normal with each axis. The face will belong to the group of that axis if the angle is smaller than a ceratin threshold (20 degrees).

As seen in Figure 3 there are two areas with normals in the -Y direction. One of them is due the bottom part of the breast and the other due to the armpit. These two are easily separated by clustering and the mean Y coordinate is computed for each of them.

Picking out the landmarks is then straightforward: L_0 and L_3 belong to the cluster with the +Z orientation and they are the vertices whose Y coordinate is closer to the two mean Y coordinates we computed above. For L_1 and L_2 we do the same only now L_1 must be selected from the area with the +X orientation and L_2 from the area with -Y orientation.

2.4 Optimum path between landmarks

Having selected the four landmark vertices, we need to connect them in a way that they will surround the whole breast. Due to the somewhat arbitrary selection of these four landmarks and marked differences in the shape of the breasts of different subjects, we can not simply connect pairs of landmark points based on a straight line, as this could result in paths that 'climb' on the breast. We need an algorithm that determines the optimum path between each pair of landmarks; our optimality criteria are geodesic distance between landmarks as well as the degree to which the path 'climbs' on the breast; the latter is simply the amount of displacement on the Z axis.

We construct a graph out of the mesh data where each vertex is a node and each triangle edge is a graph edge.

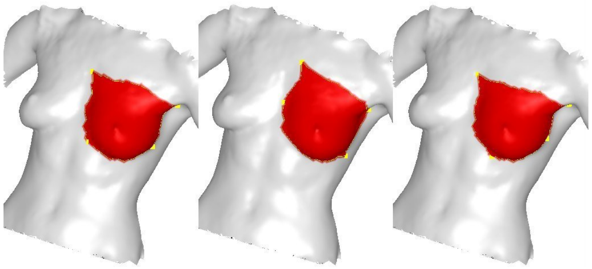

Figure 4. Different sets of landmarks and the resulting path.

Then our problem is equivalent to finding the shortest path between node A and B in a weighted graph. Dijkstra's[1] classic shortest path algorithm has been improved upon by others including the fast marching algorithm[6][4]. In our implementation we use Dijkstra's algorithm modified in two aspects.

The first modification uses the heap structure introduced by the fast marching algorithm to reduce data retrieval/update to $O(logn)$ from $O(n)$. Thus the overall computational cost is now $O(nlogn)$ instead of $O(n^2)$. The impact in computational time is significant since the scanned data usually consist of several thousand vertices.

The second modification is in the way we compute the cost between two nodes of the graph. To avoid 'breast climbing', we penalize movement in the Z direction by adding a weight parameter to the Euclidean distance equation between two connected nodes. The final equation is given by:

$$\begin{aligned} cost(i,j) = &\sqrt{(i_x - j_x)^2 + (i_y - j_y)^2 + a(i_z - j_z)^2} \\ &a \text{ is the Z weight parameter} \\ &\text{if } a\text{=1 this is equivalent to Euclidean distance} \end{aligned} \tag{3}$$

Figure 4 depicts the path created by the algorithm for different sets of landmarks. It is evident that the optimal path never climbs on the breast, even if the shortest geodesic distance is along a climbing path.

2.5 Surface construction

Having constructed the four border paths, we form a Coons patch [2] between them. Since it is an interpolating surface the Coons patch will pass through all the points of every path and will be smooth. This surface is shown in Figure 5 where we artificially removed the breast by running a modified flood fill algorithm in the mesh vertex graph and selecting the nodes that are inside the path. As can be seen, the surface follows the curvature of the torso and can be used for volume estimation.

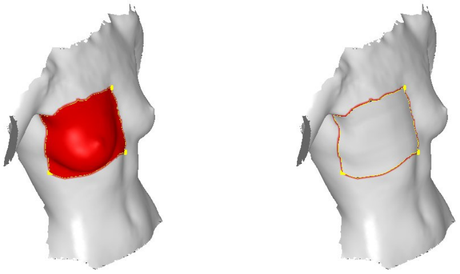

Figure 5. (Left) The optimum path and the selected breast area. (Right) The constructed surface.

The Coons patch is constructed using the formula

$$\begin{aligned} S(u,v) = \quad &(1-u)*PATH_1(v) + u*PATH_2(v) \\ &+ (1-v)*PATH_3(u) + v*PATH_4(u) \\ &- (1-u)*(1-v)*PATH_1(0) \\ &- u*(1-v)*PATH_2(0) \\ &- (1-u)*v*PATH_1(1) \\ &- u*v*PATH_2(1) \end{aligned} \tag{4}$$

where $PATH_1$ connects L_0 and L_1, $PATH_2$ connects L_1 and L_2 etc. In order to use the paths obtained from the previous step in this equation we first re-parameterize them in the interval [0,1] using arc-length parameterization. The above formula gives us a continuous surface but in order to render it we need to sample it. In our implementation a grid of 16x16 points is sufficient for rendering.

2.6 Volume estimation

For computing the volume between the two surfaces we use a z-buffer[3] method. Due to modern hardware the z-buffer is proven to be one of the fastest methods for measuring volume or acquiring a voxelization of a triangular mesh. In our implementation we use OpenGL as the rendering API of our application so retrieving the z-buffer of a scene is simple.

In order to be able to use the z-buffer we must first position the camera of our scene in a way so that the breast is visible. We set the viewing center at (0,0,0) (which is the center of the mesh's axis). To compute the volume of the left or right breast we use as the viewing vector the left or right mean normal vector. These vectors are found as the mean normal vector of the vertices that lie on the left and right (negative and positive X coordinate). Of course, we again weight them according to the area or the face they belong. The reason this works is that if we compute the mean

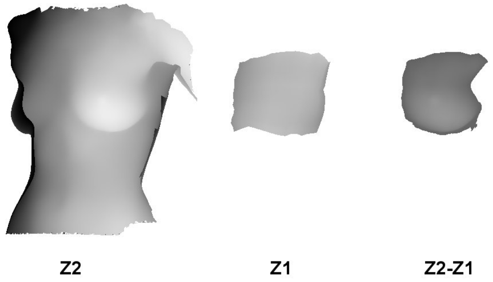

Figure 6. The two z-buffers of the scene after orienting the camera.

normal of only the right vertices then this normal will have an orientation which will approximating the orientation of the left breast. It is possible for the user to select a different orientation by rotating the mesh, but we have found that this method gives a correct orientation. Moreover the differences in the computed volume by rotating the mesh around the auto-selected vector up to 10-15 degrees are less than 1%.

After the virtual camera is set we render the scene twice. First we get a z-buffer called Z_1 by rendering the constructed surface and another called Z_2 by rendering the mesh as shown in Figure 6. Every pixel of the z-buffer has a unit volume that is calculated by the settings and the measurement units of the data. Thus, the result is given by the sum of the values of $Z_2 - Z_1$ multiplied by the unit volume. If in a pixel of $Z_2 - Z_1$ we have a negative value we disregard it. In Z_1 the pixels that were not rendered have a default 'infinite' value so they produce a negative value in the sum, thus not contributing in the volume. This way only the areas of the mesh that are above the surface contribute to the volume. Of course a high resolution is selected in order to have high accuracy and the mesh is rendered in a way that it will use all available bits of the z-buffer (typically 24bit).

2.7 Skin reduction

In this optional step, we take into consideration the thickness of the breast in the computation of the volume. The reason for it is simple: the surgeons want to know the volume of the breast tissues, but when we compute it using scanned data we unavoidably compute a slightly larger volume due to skin thickness. Note that this step is not necessary because not always the thickness of the skin on the breast is known and the difference in the volume is not very important. A typical reduction in the computed volume for a skin thickness set to 2mm was 3-5%.

The initial idea was to measure the area of the breast and then multiply it with a thickness value in order to take the volume we must subtract. Unfortunately our method does not compute the area of the breast. The reason is that areas where the constructed surface are just beneath the breast will not contribute much in the computation of the volume but would affect the computation of the area. A different approach was implemented: we 'shrink' the whole triangular mesh by simply moving each vertex along its negative normal direction by a value equal to the skin thickness. Since the triangular mesh represents the skin surface if we want to 'remove' the skin it is logical to move this surface inward according to the thickness value we have. This step is actually done first so the rest method will not change, it is just computed for the shrunk version of the mesh.

3 Results

We have performed numerous experiments to assess the accuracy, robustness and reproducibility of our method.

First, we emmployed synthetic data in order to estimate the accuracy of the z-buffer method for measuring volume. We used a triangular mesh created by a modelling software which simulated the breasts with two half spheres. The spheres where of known radius so we knew the Euclidean volume. Using the z-buffer the volume estimated was about 2% smaller than the Euclidean which was expected since a triangular representation of a sphere has always smaller volume than the Euclidean sphere.

Upon visual inspection of the results on real data, the path created from the algorithm and the reconstructed surface seem correct and the computed volume is considered within bounds. Also after running this algorithm on a specific mesh and on a re-triangulated version of this mesh the results were close which shows that the method is not heavily dependent on the triangulation of the mesh.

Even though speed is not a crucial factor in this kind of applications we have implemented the algorithms in the most efficient way. The slowest step would have been the search for the optimal path, but this was successfully reduced to $O(nlogn)$ time. The automatic orientation and the retrieval of the four landmarks is only linearly dependent with the number of vertices. Finally, the rendering of the mesh and the retrieval of the z-buffers is done extremely fast with modern graphics hardware. The method takes well under half a second, which means that the user sees the result almost immediately after he requests the breast volume.

4 Conclusion

We have presented a method that can non-invasively compute the volume of the breast using 3D scanned data. This can be used by surgeons for planning the reconstructive surgery before post-mastectomy breast reconstructive surgery. We are in process of validating our volume measurements extensively using clinical data of the excised volume of the breasts after surgery.

References

[1] E. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematic*, 1:269–271, 1959.

[2] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, 1990.

[3] R. W. Jr and M. Sweet. *OpenGL SuperBible*. Waite Group Press, 1999.

[4] R. Kimmel and J. Sethian. Computing geodesic paths on manifolds. *Proc. National Academy of Science*, 1997.

[5] C. Rigotti, G. Ferrigno, A. Aliverti, and A. Pedotti. Surface scanning: An application to mammary surgery. *Journal of Biomedical Optics*, 3(2):161–170, 1998.

[6] J. Sethian. A fast marching level set method for monotonically advancing fronts. *Applied Mathematics*, 93:1591–1595, 1996.

[7] G. Strang. *Linear Algebra and its Applications*. Harcourt Brace Jovanovich, 1988.