AN AUTOMATED METHOD TO ESTIMATE FEMORAL SHAPE AND MINERAL MASS

Danilo P. Pau,* Daniele Masala,** Xinfeng Bao,*** Alberto Gnemmi,*** Rachel C. Entwistle,****
and Dan Dragomir-Daescu****

Abstract

Medical assessment of bone health often uses quantitative computed tomography (QCT) scans and requires a reliable segmentation of bone geometry from surrounding tissues for a quick determination of bone mineral mass. Because of its shape and position in the body, the femur is one of the most challenging bones to investigate. In the current study, we developed a new automated way to accurately evaluate both the shape and the mineral mass of cadaveric femur. The results were achieved through a series of steps including the segmentation of bone tissue from sets of QCT images, the estimation of the bone's outer surface, the calculation of the volume enclosed, and finally the evaluation of bone mineral mass in a user-defined region. We compared our algorithms outputs to results obtained by expert manual segmentation and those obtained using other published methods.

Key Words

Image based meshing, deformable models, femur shape QCT, mineral mass, finite element analysis, osteoporosis

1. Introduction

Three papers by the Mayo Clinic's Division of Engineering [1]–[3] describe a technique to make three-dimensional (3D) models of the proximal femur geometry and mineral distribution using mostly manual and semi-automated methods. Their models were used for accurate finite element estimation of stiffness and load of the bone. This paper is based on our contribution and presentation to the Biomedical Engineering (BioMed 2012, Feb. 15–17, 2012) conference [4]. We used an automated process that could be performed with minimal or no human interaction. Therefore, it has the potential to dramatically reduce the processing time that is up to 2 or 3 h to segment an osteoporotic bone, needed in the clinic to evaluate its quality. We started from the same high quality quantitative computed tomography (QCT) image sets provided by the Mayo Clinic, with a resolution of 0.4 mm, and calculated bone shape and mineral densities in areas of interest. As an option, we allowed the users to modify the parameters used in each step of the process. The accuracy of the estimated shape and mineral mass could therefore be improved through the iterative use of our algorithms. Our method consisted of three steps: (i) a femur isolation tool (named FEMTool in this paper) which was used to segment the bone tissue, (ii) a graphic renderer based on a state of art implementation of the marching cubes algorithm [5]–[9], and (iii) a bone mineral mass computation algorithm used to evaluate the mineral mass in areas of interest. We will refer to the complete method as the femur shape and mineral (FSM) tool in the remainder of the manuscript.

2. Methods

A detailed description of all the processes in FSM is presented in the following sub-sections.

2.1 FEMTool for Femur Segmentation

The objective of the first step of the FSM method was to segment a 3D model of the femur for subsequent visualization and mass computation steps. The main steps of FEMTool were derived from an existing method named brain extraction tool (BET) [10]–[13] and re-designed to accommodate the characteristics of our femoral QCT dataset.

Prior to segmentation, a pre-processing was performed for the femoral QCT as follows: (i) First, the minimum and maximum intensity values of the tested femoral dataset were found and a threshold, which would be used to roughly separate femur region from other tissues, was calculated based on the two intensities; (ii) second, a binary image was computed by using the threshold value; (iii) a 3D dilation filter, using a structuring element of size 3, was applied on the binary image. As a result, the sparse holes inside the

* Advanced System Technology, STMicroelectronics, Agrate Brianza, Italy; e-mail: danilo.pau@st.com
** Politecnico di Torino, Italy; e-mail: dany.masala@gmail.com
*** Politecnico di Milano, Italy; e-mail: {xfkeats, gnemmi.alberto}@gmail.com
**** Division of Engineering, Mayo Clinic, 200 First Street SW, Rochester, MN 55905; e-mail: {Entwistle.Rachel, DragomirDaescu.Dan}@mayo.edu

Recommended by Dr. Z.-H. Duan
(DOI: 10.2316/J.2012.210-1044)

femur region were largely eliminated while a clear contour of the femur surface was preserved.

Next, the centre of the geometry of the 3D femur model was calculated as the centre of the femur bounding box, together with an initial estimate of the radius and height of an equivalent cylinder representing the femur. We created a deformable surface mesh, which was composed by a cylindrical body connected to two hemisphere heads placed at the opposite sides of it. This mesh was allowed to slowly deform, similar to the BET method, but with some modifications to the deformation rules. These rules were defined and applied on per iteration basis to force a small distant deformation of each vertex, accordingly to the deformation vector $\boldsymbol{u}$ defined as:

$$u = u_1 + u_2 + u_3 \tag{1}$$

The first update component $\boldsymbol{u}_1$ took the movement within the local surface, along the tangential direction of the local surface. It was used to keep the deformable surface vertices equally spaced during the updates. $\boldsymbol{u}_2$ moved the vertex along the local normal. The role of this component was to increase the smoothness of the surface. Considering the variation of curvature for the femur surface, $\boldsymbol{u}_2$ was defined by a nonlinear function. It depends on the local curvature of the surface to enforce a weak smoothing to high curvature part while a strong smoothing to low curvature part.

$\boldsymbol{u}_3$ also acts along the local normal and aimed at forcing the vertex to track the femur surface. When the deformable surface was strictly inside the femur region, the update component 3 enforced the surface to expand. When the vertex moved out of the femur region, it enforced the surface to shrink. When the surface just found the boundaries of the femur, expansion or shrink was stopped. For example, after about 300 iterations, the deformable model was in the static state and the deformation stopped.

During vertices deformation, the average voxel greyscale values were calculated for a region extending 3 mm outwards. The average voxel intensity was compared to the binary dataset to determine both the direction of the deformation, *i.e.*, expanding or shrinking, and the value of the movement. The deformation stopped when all vertices reached the contour of the reconstructed femur. At this point, a temporary mesh was created representing the surface of the femur. For the next iteration, the mesh was then shrunk to half of the original size and re-located to the centre of the geometry. The modified mesh was re-used as more detailed initial mesh of the deformable model and the algorithm was repeated until convergence was reached (*i.e.*, the differences between two meshes as measured by the Hausdorff distance (HD) [14] was below a threshold value). Finally, the converged mesh was used to create a 3D mask and the mask was applied to the original QCT dataset with a simple selection/rejection rule to obtain the segmented femur. Since the deformable model was by construction water-tight, the output of the rule was still as such, therefore removing the need of time-consuming manual verification on per slice basis to make sure such requirement was met. This effort was increasing when processing normal, osteopenic and osteoporotic bones.

2.2 Marching Cubes for Triangle Mesh

The marching cubes (MC) algorithm was first proposed by Lorensen and Cline in 1987 [5] to reconstruct geometry from greyscale datasets. We used the algorithm to build a set of triangles all having the same value specified by the user. It was therefore possible to visualize multiple surfaces for the same dataset by using different iso-values. This algorithm gave the user the choice to visualize an entire femur reconstruction but also to visualize different surfaces of the same object, each of them having a specific greyscale value associated with material density.

We implemented a slightly different version of the MC in order to solve the known ambiguities present in the original definition of the algorithm. In particular, we adopted the solution proposed by Lewiner *et al.* [8] based on the computation made by Chernyaev [7] and Nielson and Hamann [15], which led to a minimal increment in the number of triangles of about 6%.

In addition, we adopted a solution proposed by Glanznig *et al.* [9] for the modification of the iso-value along a direction of the dataset. In this way, it was possible to visualize a mixed surface that had different density values with respect to the position in the model. This feature was very useful when dealing with femur datasets that had very high values of density in the diaphysis and lower density values at the femoral head and neck. MC was used to create a layered structure which was composed of a mesh inside of another mesh. As further improvement, we added a feature for mesh selection. The user had the ability to view a selected layer within the mesh.

The goal of the algorithm was to build a watertight outer contour of the bone for each QCT image slice, so it would be possible to distinguish between external and internal voxels involved in the MC computation. Tests showed that when applying this procedure the number of triangles was reduced by about 30%.

2.3 Mass Computation

In order to compute it, we adopted the same strategy used by MC because the dataset consisted of cube-like voxels. After choosing an iso-surface that bounded the volume, the total mass value related to that volume was the summation of the contributions given by all the cubes crossed by the surface and by the cubes completely inside that surface as per (2):

$$Mass = \Sigma_1^n V_{bc} d_a + \Sigma_1^m V_p d_{ati} \tag{2}$$

The first sum included n cubes completely inside the surface and the second sum included m cubes crossed by the surface. V_{bc} was the volume of the cube, d_a was the density average of a cube which was estimated by averaging the eight voxels at the corners of the cube. V_p was the partial volume contribution of a cube intersected by the surface. For a more accurate estimation of volume,

Table 1
HD Accuracy and Execution Time of Femur Segmentation

	Number of Samples	Minimum Value (mm)	Mean Value (mm)	Maximum Value (mm)	Execution Time (s) ×86 2.66 GHz 4 GB RAM	Manual Segmentation Time
Dataset 1 Osteoporotic	45803	0.000013	0.308288	4.885335	59/135	2–3 h
Dataset 2 Normal	26185	0.000041	0.429033	4.602976	75/248	15 min

Table 2
HD Comparison between Mayo Ground Truth versus Marching Cubes

<table>
<tr><th colspan="5">Dataset 1</th></tr>
<tr><th>Iso-Values Used</th><th>Minimum Value (mm)</th><th>Mean Value (mm)</th><th>Maximum Value (mm)</th><th>RMS (mm)</th></tr>
<tr><td>−t 0.1</td><td>0.000000</td><td>0.983030</td><td>6.273026</td><td>1.278504</td></tr>
<tr><td>−t 0.2</td><td>0.000001</td><td>0.670587</td><td>3.689515</td><td>0.805856</td></tr>
<tr><td>−a 0.2 0.1 0</td><td>0.000001</td><td>0.864244</td><td>4.798577</td><td>1.094258</td></tr>
<tr><td>−a 0.3 0.1 0</td><td>0.000004</td><td>0.754805</td><td>4.553466</td><td>0.937613</td></tr>
<tr><td>−a 0.4 0.1 0</td><td>0.000001</td><td>0.664291</td><td>3.721582</td><td>0.815852</td></tr>
<tr><td>−a 0.5 0.1 0</td><td>0.000001</td><td>0.675783</td><td>4.274777</td><td>0.837875</td></tr>
<tr><td>−a 0.6 0.1 0</td><td>0.000001</td><td>0.791206</td><td>5.716393</td><td>1.008160</td></tr>
<tr><th colspan="5">Dataset 2</th></tr>
<tr><th>Iso-Values Used</th><th>Minimum Value (mm)</th><th>Mean Value (mm)</th><th>Maximum Value (mm)</th><th>RMS (mm)</th></tr>
<tr><td>−t 0.1</td><td>0.000000</td><td>0.796746</td><td>6.382667</td><td>1.044301</td></tr>
<tr><td>−t 0.2</td><td>0.000001</td><td>0.394278</td><td>5.718165</td><td>0.638745</td></tr>
<tr><td>−a 0.2 0.1 0</td><td>0.000000</td><td>0.628749</td><td>6.230423</td><td>0.910276</td></tr>
<tr><td>−a 0.3 0.1 0</td><td>0.000000</td><td>0.531296</td><td>6.041454</td><td>0.818764</td></tr>
<tr><td>−a 0.4 0.1 0</td><td>0.000000</td><td>0.475577</td><td>5.839408</td><td>0.748784</td></tr>
<tr><td>−a 0.5 0.1 0</td><td>0.000000</td><td>0.461140</td><td>5.637133</td><td>0.699850</td></tr>
<tr><td>−a 0.6 0.1 0</td><td>0.000001</td><td>0.492587</td><td>5.408714</td><td>0.682123</td></tr>
</table>

the partial cube was further divided into a high number of smaller cubes, t. Therefore, $V_p = \frac{p}{t} V_{bc}$, where p was the number of smaller cubes, that lay inside the surface. To compute p, a density value was assigned to each smaller cube and trilinear interpolation of the eight voxels at the corners of the bigger cube was used. If this value was above the threshold value, the smaller cube was added to the volume of the bone. The last variable d_{ati} was the mean of the interpolated densities above the threshold. The value obtained represented the mineral mass of the bone region delimited by the triangle surface created by MC with the same iso-value threshold.

3. Results

The validation of results quality was performed at each step. First, we compared the numerical results obtained with our FSM method to results obtained by expert users who employed manual segmentation methods on Mayo's QCT datasets (ground truth). We compared: (i) the final iteration meshes of FEMTool step, (ii) the meshes produced by just the MC and (iii) the meshes that resulted from the whole FSM process. In order to compare the meshes, the HD between any two meshes was calculated using MeshLab v1.3.0b [16].

Table 3
HD Comparisons between Mayo Ground Truth and FSM

Dataset 1				
Iso-Values Used	Minimum Value (mm)	Mean Value (mm)	Maximum Value (mm)	RMS (mm)
−t 0.1	0.000002	0.992021	6.746704	1.361168
−t 0.2	0.000001	0.662665	3.448801	0.846973
−a 0.2 0.1 0	0.000005	0.889290	6.635574	1.215395
−a 0.3 0.1 0	0.000001	0.803438	6.602804	1.104818
−a 0.4 0.1 0	0.000000	0.755474	6.529173	1.027861
−a 0.5 0.1 0	0.000000	0.786331	6.416336	1.042601
−a 0.6 0.1 0	0.000002	0.926759	6.341863	1.187317
Dataset 2				
Iso-Values Used	Minimum Value (mm)	Mean Value (mm)	Maximum Value (mm)	RMS (mm)
−t 0.1	0.000000	0.756087	6.375305	1.013704
−t 0.2	0.000001	0.398594	5.718165	0.645803
−a 0.2 0.1 0	0.000001	0.611377	6.228402	0.894477
−a 0.3 0.1 0	0.000000	0.520899	6.041454	0.809416
−a 0.4 0.1 0	0.000000	0.469100	5.839408	0.744414
−a 0.5 0.1 0	0.000000	0.457635	5.637133	0.700392
−a 0.6 0.1 0	0.000001	0.490232	5.408714	0.685197

In order to test the numerical performance of the FEMTool algorithm, the contour vertex of the output image was projected onto the reference image. We measured the error as the HD between mesh vertices. Also, we recorded the execution time of the FEMTool using some datasets. This included pre-processing and initialization of a deformable model on a 2.66 GHz ×86 computer with 4 GB of RAM. All datasets had same voxel sizes: $0.4 \times 0.4 \times 0.4$ mm. Table 1 shows accuracy and computation time for two of the datasets.

The maximum HD values were much greater than the QCT voxel size value of 0.4 mm, while the mean distances were comparable to the voxel size suggesting reasonable segmentation results obtained by the FEMTool. There were two main causes that explained these differences: (i) the FEMTool was inaccurate when the contour of the femur had low density values of the original QCT dataset and (ii) the outputs of the FEMTool were defined by voxels but were compared to the referential meshes from Mayo which were defined by re-meshed surface triangles. Although the processing time depends on the resolution of the test femur, our femur segmentation method performs much faster than manual segmentation which could take few hours.

For the last two validations (MC and whole FSM method), the HD results varied according to the different iso-values used. A smaller HD value was reached for a certain iso-value, while the HD increased for other values. This happened because (for the characteristics of the datasets and of MC) the volume bounded by the surface decreased monotonically using higher iso-values. Thus, it was possible to obtain the optimal size of the produced mesh, as close as possible to the reference mesh, using an intermediate iso-value. Table 2 shows the computed HD between the ground truth and the meshes produced using just the MC method. Datasets 1 and 2 were the same as in Table 1. It is important to note that datasets 2 was obtained from normal bones, while dataset 1 was obtained for an osteoporotic bone which had lower density values. In Tables 2 and 3, “$-t$” indicates a single iso-value, while “$-a$” indicates a locally adaptive iso-value that varies along the vertical direction (bottom-up). Iso-values range from 0 to 1 because the density values of the datasets were normalized during the MC calculations. Values are expressed in mm.

Most of both mean and RMS values in the MC solution were less than 1 mm, and they were the most important parameters to consider since directly comparable to voxel size of the dataset being considered. Moreover, the maximum difference of about 6.4 mm was attributed to small local artifacts produced by the MC, located where blood vessels enter into the femur, and it is not representative of its global quality. These values can be considered reasonable because the method used to build the two different meshes were very different, and because the time durations needed

Table 4
Triangle Reduction using Mesh Selection

Iso-Values Used	Dataset 1	Mesh Selection	%Red	Dataset 2	Mesh Selection	%Red	Dataset 3	Mesh Selection	%Red
−t 0.1	164198	148706	10.42	250320	220920	13.31	1094220	921220	18.78
−t 0.2	243122	167949	44.76	249379	200779	24.21	2595991	1571845	65.16
−a 0.2 0.1 0	181183	148067	22.37	248691	214209	16.10	1095346	845797	29.50
−a 0.3 0.1 0	226111	150982	49.76	263979	202030	30.66	2282915	878348	159.91
−a 0.4 0.1 0	250084	210384	18.87	276016	240234	14.89	2815834	1878035	49.94
−a 0.5 0.1 0	260713	215045	21.24	307111	243092	26.34	3537815	2978481	18.78
−a 0.6 0.1 0	239064	214500	11.45	356286	312860	13.88	4105568	3400650	20.73

Table 5
Mass Values with Different Iso-Values

Iso-Values	Dataset 1 Mass Values	Dataset 2 Mass Values	Dataset 3 Mass Values
−t 0.1	55272.415385	72176.525440	99040.706697
−t 0.2	51568.280064	70331.753017	97435.210781
−a 0.2 0.1 0	54362.526165	71581.467663	98582.653369
−a 0.3 0.1 0	50228.722872	70022.751972	95784.467794
−a 0.4 0.1 0	44899.016505	67112.193533	91544.094269
−a 0.5 0.1 0	38000.993753	64473.968513	87280.599403
−a 0.6 0.1 0	31539.452710	59450.705482	82084.475701

Table 6
Mineral Mass Convergence

Smaller Cubes per Side	Dataset 1 Mass Values	Dataset 2 Mass Values	Dataset 3 Mass Values
2	51542.819559	70305.225212	97489.704521
4	51563.331653	70326.835245	97444.250922
6	51568.280064	70331.753017	97435.210781
8	51570.281314	70333.230100	97432.096701
10	51570.852564	70334.104485	97430.647747
12	51571.531224	70334.416547	97429.816306

to build these meshes were considerably lower than the durations needed to make the manual meshes. For example we achieved a parallel implementation of MC on Intel Mobile Core 2 Duo T7200@2.00 GHz in 15 s compared to 2–3 h of manual segmentation needed on the osteoporotic dataset.

Table 3 presents the HD between the ground truth and the whole FSM method.

Results obtained with the whole FSM were slightly better than the single MC results, on dataset 2, while less evident on dataset 1. Additionally, there was a dramatic reduction in the number of triangles using the mesh selection option to triangulate and visualize the external surface of the femur, as shown in Table 4 in the column %red (minimum reduction was 10.42% while maximum was 159.91%). Dataset 3 had voxels size equal to 0.4 × 0.4 × 0.31 mm and was obtained from normal bone.

Once algorithm validation was completed through the building of the triangular mesh, the quality of the bone mineral mass computation algorithm was evaluated. The values obtained from the algorithm were compared by varying two parameters: the iso-values and the precision with which the computation is performed. Table 5 shows how the computed mass varied using different iso-values. When using higher iso-values, the volume bounded by the built surface decreased, so it was expected that the mass value would also decrease. The values in the table were normalized density values (mm^3) as it has not been used a calibration method between greyscale scalar values in the QCT dataset and bone mineral density in g/cm^3. Table 5 shows a comparison of values obtained from Dataset 1,2 and Dataset 3. The first was an osteoporotic femur while the second and third were normal. For the same iso-values, the mass values estimated for the normal femurs were consistently higher than those for the osteoporotic femur which was consistent with bone mineral loss in the osteoporotic femur. Table 6 shows how the mass mineral estimate converged with increased number of smaller cubes used to divide a basic cube for an iso-value of 0.2.

One of the possible applications of a triangular mesh was for finite element analysis (FEA), which computes and evaluates the mechanical properties of the femur [1]. The goal was to determine if the mesh built, using MC was also suitable for FEA. Application of the mesh selection algorithm was a good option for achieving this goal of producing a good bone surface mesh. Shewchuk [17] in his analysis underlined the importance to deal with good-shaped triangles in order to perform correctly the aforementioned finite element methods. In particular, the constraints were stricter *i.e.*, even the worst-shaped triangle had to have a good quality. The best possible triangle shape for FEA is equilateral.

Table 7
Quality of the Mesh Produced by FSM

Deciles	Dataset 1	Dataset 2	Dataset 3
<0.1	7.05%	6.31%	4.79%
<0.2	5.22%	4.72%	4.71%
<0.3	5.51%	8.14%	4.99%
<0.4	12.24%	8.24%	5.55%
<0.5	8.85%	7.56%	6.27%
<0.6	8.81%	8.58%	6.43%
<0.7	10.23%	10.23%	7.61%
<0.8	13.48%	13.95%	12.12%
<0.9	18.25%	21.49%	31.26%
<1	10.38%	10.79%	16.28%

Three methods [17] are primarily used to measure the triangle shape quality: (i) maximum angle in the triangle (smaller is better); (ii) minimum angle (larger is better); (iii) ratio of the radius of the circle inscribed in the triangle and the radius of the circle circumscribed by the triangle (larger is better); this latter technique was used in our study.

For each triangle the radii ratio was computed; these values were normalized such that equilateral triangles have a ratio equal to 1. Then all the values have been grouped in 10 deciles. The first contained all the triangles having a radio ratii less than 0.1; the second contained all the triangles having a radii ratio less between 0.1 and 0.2 and so on. Table 7 shows the quality of the mesh obtained executing FSM on the three different datasets using an iso-value equal to 0.1. Left most column reports the deciles while each column for each dataset reports the percentage of triangles grouped in deciles.

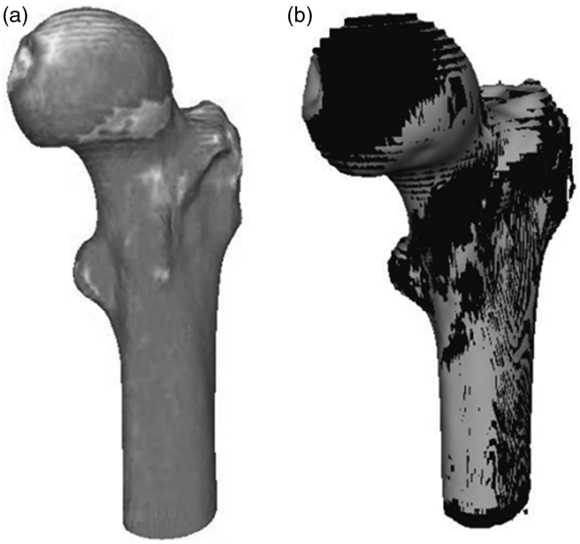

Figure 2. (a) Relative errors between two meshes; (b) gray mesh was the ground truth; black mesh was produced by FEMTool.

The majority of the triangles were concentrated in the better quality deciles (>0.5). Nevertheless, the quantity of the worst triangles was not negligible; therefore, the mesh produced by FSM was not suitable for use in FEA. Future work on FSM will consider adapting the MACET method to MC, described by Dietrich *et al.* [18], which guaranteed higher shape quality.

Figure 1 shows an example of a 3D mask and the corresponding surface model obtained during the deformation

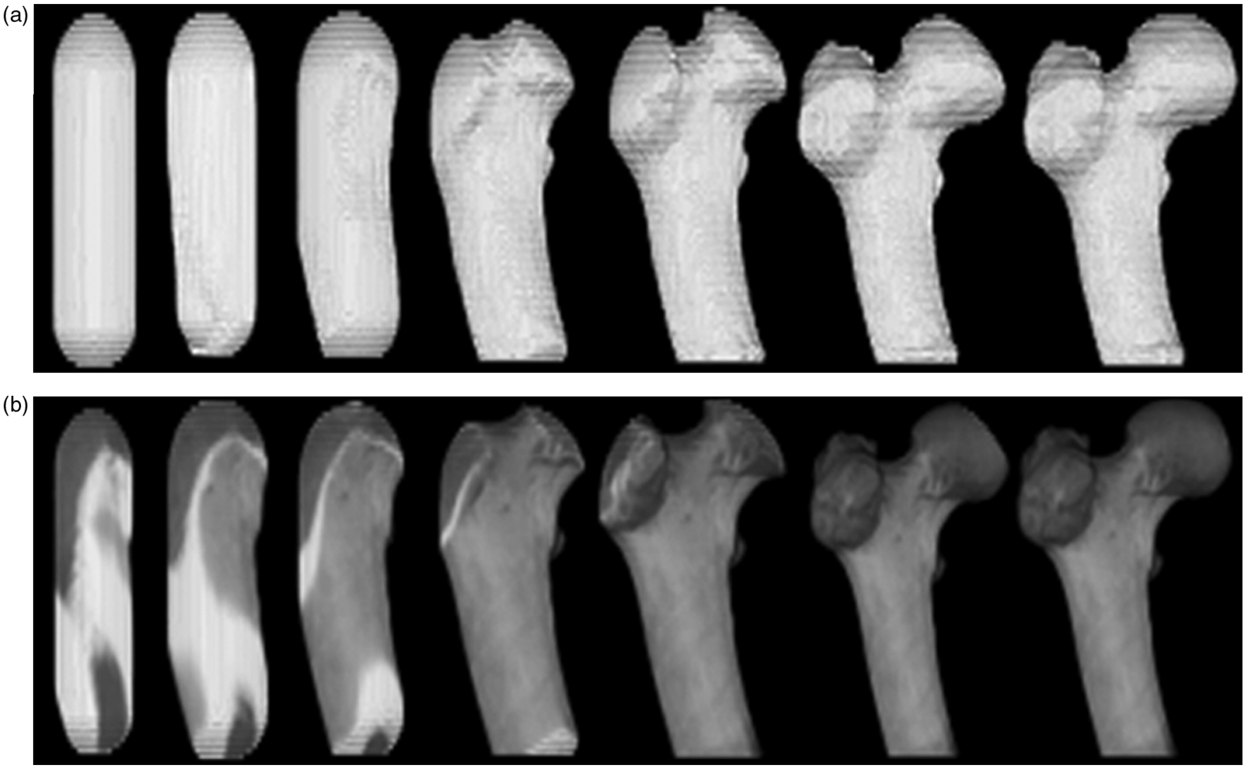

Figure 1. Temporal evolution of the 3D mask (a); corresponding surface model (b).

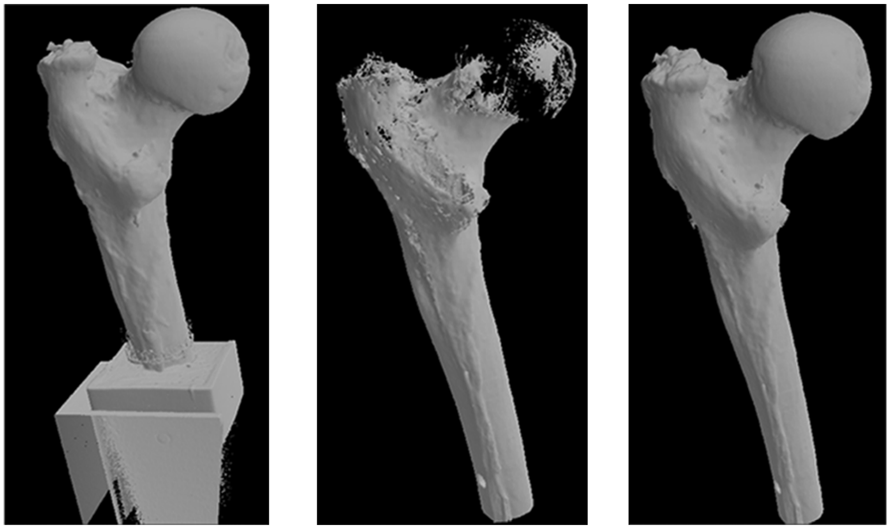

Figure 3. From left to right, Marching Cube set with different thresholds, respectively 0.1; 0.4; 0.1 to 0.4 (local adaptive behavior).

achieved by the FEMTool. Figure 2 shows one example of the mesh produced by FEMTool and the ground truth model after alignment. In the right image, the grey mesh was the ground truth, while the black mesh was the one produced by FEMTool. The differences between the two meshes are shown on the left side. For a better visualization of the errors, these values were converted into gray scale. With a function of MeshLab the errors have been mapped onto a white-gray-black color map where minimum error corresponds from white to gray.

Some of the methods used in FSM were evaluated by showing the meshes produced and rasterized using OpenGL APIs. Locally adaptive procedures proved to be very effective with the Mayo datasets. These procedures were used to eliminate a distal end bone-cement block from the images which was used to prepare the femora for mechanical testing. The cement had low radiographic density values and required the use of locally adaptive iso-values. This allowed us to correctly resolve the femoral head, which had comparably low radiographic densities. Figure 3 shows three meshes of the same dataset obtained with different iso-values. The left image had an iso-value of 0.1, the middle image had an iso-value of 0.4, while the right image, had a locally adaptive range from 0.4 (bottom) to 0.1 (top).

The tool has been optimized and tested on a notebook with Intel Centrino 1.73 GHz processor, 1 GB RAM, nVidia GeForce6600 [19] 256 MB GPU supporting OpenGL 1.5 [20]–[22]. The execution time for un-optimized serial execution on a ×86 single processor of Mesh Creation was typically 90 s and output a number of triangles. Fifty per cent of the time was spent on normal computations from local gradients computed with a Sobel filter. The rendering speed was 15–20 fps. However, if the mesh was too detailed in the range between 1 and 4 M triangles per mesh, then the performances were between 7 and 2 fps, respectively.

4. Conclusions

We presented in this study a new way to use femur QCT datasets in order to evaluate their shape and total mineral mass. The proposed method consists of a first step called FEMTool (FEMur isolation tool) which segments the bone tissues from the original QCT dataset, generating at its output a water-tight model. A graphic renderer, using a state of art implementation of the MC algorithm, integrating a number of algorithm evolutions, was used to estimate bone shape. At the end of the process, a mass estimation algorithm calculated the mass of a user-specified volume, defined by a set of triangle meshes created using the same iso-value(s) obtained during the graphic stage.

Results showed a very good quality model built with the FEMTool, MC, and complete FSM tool compared to the results obtained by expert manual segmentation. A mesh selection algorithm helped reduce the number of triangles produced and with mesh selection. The mass computation algorithm was validated by comparing different values, estimated with different input parameters. However, the resulted surface meshes were not suitable for application to FEA models but will be improved in future work.

References

[1] D. Dragomir-Daescu, J. Op Den Buijs, S. McEligot, Y. Dai, R.C. Entwistle, C. Salas, L.J. 3rd Melton, K.E. Bennet, S. Khosla, and S. Amin, Robust QCT/FEA models of proximal femur stiffness and fracture load during a sideways fall on the hip, *Annals Biomedical Engineering*, *39*(2), 2011, 742–755, epub 2010 Oct. 29.

[2] J. Op Den Buijs and D. Dragomir-Daescu, Validated finite element models of the proximal femur using two-dimensional projected geometry and bone density, *Computational Methods Programs Biomedicine*, *104*(2), 2011, 168–174.

[3] A. Cong, J.O. Buijs, and D. Dragomir-Daescu, In situ parameter identification of optimal density-elastic modulus relationships in subject-specific finite element models of the proximal femur, *Medical Engineering and Physics*, *33*(2), 2011, 164–173, epub 2010 Oct. 27.

[4] D.P. Pau, X. Bao, D. Masala, A. Gnemmi, R.C. Entwistle, and D. Dragomir-Daescu, An automated method to estimate femoral shape and mineral mass, Proceeding (764) Biomedical Engineering/765: Telehealth/766: Assistive Technologies, 2012. doi: 10.2316/P.2012.764-004

[5] W. Lorensen and H. Cline, Marching cubes: A high resolution 3D surface construction algorithm, *Computers and Graphics*, *21*(4), 1987, 163–169.

[6] T.S. Newman and H. Yi, A survey of the marching cubes algorithm, *Computers and Graphics*, *30*(5), 2006, 854–879.

[7] E. Chernyaev, *Marching cubes 33: construction of topologically correct isosurfaces* (Cern Libraries, Geneva: CERN Report, 1995).

[8] T. Lewiner, H. Lopes, A.W. Vieira, and G. Tavares, Efficient implementation of marching cubes cases with topological guarantees, *Journal of Graphics Tools*, *8*(2), 2003, 1–15.

[9] M. Glanznig, M.M. Malik, and M.E. Groller, Locally adaptive marching cubes through iso-value variation, *Proc. Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, Plzen, Tschechien, 2009, 33–40.

[10] T. McInerney and D. Terzopoulos, Deformable models in medical image analysis: A survey, *Medical Image Analysis*, *1*(2), 1996, 91–108.

[11] S.M. Smith, Fast robust automated brain extraction, *Human Brain Mapping*, *17*(3), 2002, 143–155.

[12] S.M. Smith, *BET: Brain Extraction Tool*, FMRIB technical Report TR00SMS2b, Oxford Centre for Functional Magnetic Resonance Imaging of the Brain, Department of Clinical Neurology, Oxford University, John Radcliffe Hospital, Headington, Oxford.

[13] K. Boesen, K. Rehm, K. Schaper, S. Stoltzner, R. Woods, E. Lüders, and D. Rottenberg, Quantitative comparison of four brainextraction algorithms, *NeuroImage*, *22*(3), 2004, 1255–1261.

[14] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, Mesh: measuring G errors between surfaces using the hausdorff distance, *Proc. IEEE International Conference in Multimedia and Expo (ICME)*, *1*, 2002, 705–708.

[15] G. Nielson and B. Hamann, The asymptotic decider: resolving the ambiguity in marching cubes, *Proc. Visualization '91*, San Diego, 1991, 83–91.

[16] MeshLab an open source, portable, and extensible system for the processing and editing of unstructured 3D triangular meshes. Software download at http://meshlab.sourceforge.net/

[17] J.R. Shewchuk, What is a good linear element? Interpolation, conditioning, and quality measures, *Proc, 11th Int Meshing Roundtable*, IMR, 2002, 115–126.

[18] C.A. Dietrich, C.E. Scheidegger, J. Schreiner, J.L.D. Comba, L.P. Nedel, and C.T. Silva, Edge transformations for improving mesh quality of marching cubes, *IEEE Transactions on Visualization and Computer Graphics*, *15*(1), 2009, 150–159. doi: 10.1109/TVCG.2008.60

[19] G. Johannson and H. Carr, Accelerating marching cubes with graphics hardware, *Proc. 2006 Conf. of the Center for Advanced Studies on Collaborative Research, ACM*, IBM Corp. Riverton, NJ, USA, 2006.

[20] S. Richard, W.B. Lipchak, N. Haemel, G. Sellers, *OpenGL SuperBible: comprehensive tutorial and reference, 5th edn.* (Addison-Wesley, July 23, 2010). ISBN-10: 0321712617, ISBN-13: 978-0321712615.

[21] R. Fosner, *OpenGL Programming for Windows 95 and Windows NT* (Addison-Wesley), 1996.

[22] OpenGL Architecture Review Board, *et al.*, *OpenGL Programming guide: the official guide to learning OpenGL, version 2, 5th edn.* (Addison-Wesley, 2005).

Biographies

Danilo P. Pau received an electronic engineering degree from Politecnico di Milano in 1992. In 1991, he joined STMicroelectronics in Advanced System Architectures as a contractor on high definition decoding HD-MAC hardware design. Since 1994, he has been within Advanced System Technology working on MPEG2 video-processing algorithms, memory reduction, and related system architectures for Set Top Box and mobile application processors. He moved to STMicroelectronics Ltd, Bristol (UK) in 2004, where he worked mainly on algorithms and architectural studies in the field of 3D graphics for mobile processors compliant with the worldwide standard OpenGL-ES. He then moved back to Agrate, where he continued to apply himself to 3D graphics development and started work on 2D scalable graphics algorithms, as well as architectural studies for Set Top Box and thin clients. He is currently interested in computer vision, in particular structure from motion, optical flow and visual search. On the latter he acts as co-chair of MPEG CDVS ad-hoc group. During his career, he has authored or co-authored many European, American and Japanese granted patents, as well as international publications in conferences and technical journals. Currently he is a senior principal engineer, senior member of technical staff in STMicroelectronics, and an IEEE senior member.

Daniele Masala received his Master's degree in computer engineering, embedded system specialization, in 2011 from Politecnico di Torino with 110/110 *cum laude* top score. At the same university, he received his Bachelor's degree in 2008. From April to July 2008, he was a software developer at LogiTagTech, where he implemented a bike-rent procedure for a bike-sharing project installed in a town close to Torino, Italy. From March 2010 to February 2011, he did his internship at STMicroelectronics, where he created an application to analyze and visualize 3D models from medical images (TC, MRI), using OpenGL libraries, in cooperation with the Mayo Clinic. He is currently a software engineer at GFI Informatique working on the implementation, maintenance and upgrading of the Open Back End software in charge of handling transactions for the flight insurance and cruise market, worldwide.

Xinfeng Bao received his B.Sc. degree in electronics and information engineering from Huazhong University of Science and Technology, China, in 2008. In the same year, he started his Master's study in computer engineering at Politecnico di Milano, supported by an ICE-UNIONCAMERE scholarship. In 2010, he joined the Graphics 4 Embedded Systems research group at STMicroelectronics in Italy as an intern researcher. During his one year of research at STMicroelectronics, he was engaged in a joint project with the Mayo Clinic, USA, in developing a 3D visualization and analysis tool for magnetic resonance imaging (MRI) and quantitative computed tomography (QCT). In early 2011, he received his M.Sc. degree from Politecnico di Milano. Since 2011, he has been a Ph.D. student in the video coding and architectures group at the Eindhoven University of Technology, the Netherlands. His research interests are image and video processing, pattern recognition and machine learning.

Alberto Gnemmi received his Master's of Science in computer science from University of Illinois, Chicago, USA in 2011, his Master's of Science in computer engineering from Politecnico di Milano, Italy in 2011 and a Bachelor's degree in computer engineering from Politecnico di Milano in 2009. He spent an internship at STMicroelectronics, between March 2010 and June 2010, working on 3D reconstruction of medical images in cooperation with the Mayo Clinic. Then, he was a research assistant at the University of Illinois in Chicago between January 2011 and May 2012, working on robotic rehabilitation of stroke survivors at the Rehabilitation Institute of Chicago. His interests are motor learning, motor adaptation and human–computer interaction. He is currently a software engineer at NAVTEQ.

Rachel C. Entwistle received her Bachelor's degree in engineering science, with a mechanical concentration, from Smith College in 2005. She also spent a year abroad studying at the University College London while pursing her Bachelor's degree. At Smith, she participated in two research projects, one looking at the variation of collagen fibril diameter in murine Achilles tendon and the other exploring murine tail tendon functional changes associated with a certain growth factor. In 2008, she received a Master's of Science degree in biomedical engineering, with a biomechanics concentration, from the University of California at Davis. Her graduate work explored bone functional and microstructural changes associated with exercise intensity and stress fracture in equine bone. After graduate school, she worked at the Mayo Clinic for 2.5 years as a mechanical engineer. Her primary focus there was on a project that aimed to evaluate bone strength in the hip. This project used patient-specific CT scans to build computer models used for finite element models, which in turn could provide an evaluation of bone strength in a fall-on-the-hip orientation. She has authored or been a co-author on more than eight peer-reviewed publications, in journals such as the Journal of Orthopaedic Research, Annals of Biomedical Engineering, Equine Veterinary Journal and Veterinary Surgery. She has given podium presentations and presented posters at conferences such as the Annual Meeting of the Orthopaedic Research Society, International Sports Engineering Association and the American Association of Equine Practitioners. Currently, she is employed as a biomedical test engineer at Heartflow, Inc.

Dan Dragomir-Daescu is an associate professor of biomedical engineering at the College of Medicine, Mayo Clinic, Rochester, Minnesota. His research interests are in tissue engineering, bone mechanics, hemodynamics and cardiovascular devices. Since his appointment as a faculty at Mayo Graduate School's Biomedical Engineering track, he has advised more than 30 postdoctoral fellows, interns and Mayo Clinic graduate students. He has served as a member on Ph.D. committees and has taught graduate level courses in biomedical engineering and mathematics. Before joining Mayo, he was an assistant professor at North Dakota State University, where he developed computer simulations to estimate material properties of carbon nanostructures. He also taught graduate classes on advanced numerical methods and undergraduate classes on engineering mechanics. He also worked at the General Electric Global Research Center, where he led engineering teams that developed new product performance and reliability testing methods, as well as innovative methods and algorithms to predict product lifetime. He received three awards for technical excellence for his work in signal processing and reliability. Dan currently holds six patents, with four more pending. He has co-authored more than 60 technical papers and one book chapter, and has presented on numerous occasions at national and international engineering conferences. He is member of several research and engineering organizations and is an active reviewer of publications for engineering journals and conference peer-reviewed proceedings. He holds a Ph.D. in mechanical engineering from University "Politehnica" in Timisoara (UPT), Romania, and a M.S. degree in mechanical engineering from Arizona State University (ASU) in Tempe, Arizona, US. He also holds a M.S. degree in Applied Mathematics from the West University in Timisoara (WUT), Romania. He also has two B.S. degrees, one in mechanical engineering from UPT and the other in mathematics from WUT. During his graduate studies he acquired expertise in materials, smart composite structures, structural dynamics, signal processing and numerical methods including finite element analysis (FEA) and computational fluid dynamics (CFD).