VISBONE: 3D Visualization of Bone Mineral Density

Yi-King Choi[†] J.C.Y. Leong[‡] W.W. Lu[‡] Wenping Wang[†]

[†] Department of Computer Science and Information Systems
[ykchoi, wenping]@csis.hku.hk
[‡] Department of Orthopaedic Surgery

The University of Hong Kong
Pokfulam Road, Hong Kong

Abstract

Volume visualization is an essential part of many computer-assisted medical applications. While general-purpose commercial visualization software can provide common functions for visualizing three-dimensional (3D) medical data, there are many occasions where specific requirements cannot be fulfilled by such software. In orthopaedic surgery involving screw insertion, surgeons must consider not only the surface topology but also the bone mineral density (BMD) distribution to ensure a high success rate of the operation. In this paper we present a new system, called *VISBONE*, for visualizing the 3D BMD distribution within a bone. Design issues and specific considerations required by this application are discussed.

1 Introduction

During an implant operation, screws are inserted into bones to maintain skeletal structures. The success of this surgery depends highly on whether the screw is inserted to an appropriate region whose densities are sufficient to “secure” the screw. A misplaced screw will lead to early loosening and breakdown of the structure, as well as other neurological complications [14]. In common practice, surgeons insert the screws based on anatomic landmarks, nerve positions and also the BMD distribution within the bone. Since all these properties vary among different individuals, surgeons must examine the 2D CT sectional images of a patient carefully before an operation to determine screw insertion positions. The failure rate of screw insertion without computer assistance varies between 28.1% and 39.9% [14]. With knowledge on how the BMD are distributed over the bone volume, screws can be inserted more safely and accurately.

Three-dimensional (3D) visualization has become the core part of many systems for surgical planning and medical education. Two-dimensional (2D) images acquired by different imaging techniques, such as Magnetic Resonance (MR) or Computed Tomography (CT), are the rich sources for composing 3D medical data. Although specialists, such as radiologists, are trained to interpret 3D information from a series of 2D CT or MR images, surgeons who perform operations often find it difficult to relate cross-sectional slices to actual 3D organs, because 2D slices do not provide an intuitive understanding of the presented structures as readily as direct 3D volume visualization [15].

Currently available commercial medical software provides certain degree of 3D visualization by displaying a 3D iso-surface model extracted from 2D medical images. One limitation of the iso-surface visualization is that it can only show surface details but not the internal structures of an object.

The visualization component of current experimental systems for computer-assisted screw insertion is mainly concerned with the length and location measurement of screw [14]. However, in many cases of osteoporosis, bone density is also an important factor to consider for screw insertion. This entails the capability to visualize clearly the 3D BMD distribution in a bone.

This paper is based on a joint project between the Computer Graphics Group and the Department of Orthopaedic Surgery at The University of Hong Kong. Our primary goal is to build a customized visualization system, VISBONE, to be used in surgical planning for screw insertion. The main functions provided by the system include:

- support interactive bone display,
- show the BMD distribution, not only on the bone surface but also across the entire volume,

- indicate screw insertion path feasibility.

The remainder of this paper is organized as follows. In section 2 we shall give a brief overview of volume visualization. The implementation of VISBONE will be described in section 3. We shall then discuss design issues and some interesting findings in section 4. Finally, the paper will be concluded in section 5 with some possible future developments.

2 Background

Volume visualization is the display of a 2D projected image of a 3D volume from a given viewpoint. In the seminal paper on volume rendering [10], Levoy derived the well-known compositing functions which accumulate the color and opacity of each voxel to form a 2D image. To determine the intensity of a pixel, a ray is cast from the viewpoint through the pixel into the volume. The intensities and the opacities of the sample points along the ray are accumulated to obtain the final intensity of the pixel. When a sample point does not coincide with a voxel, interpolation technique, such as the tri-linear interpolation, is applied to the voxels surrounding the sample point to obtain an estimation of the intensity and opacity of the sample point.

Volume graphics is attracting increasing attentions from the researchers over the last decade. While some conventional surface-based algorithms are adapted to work on volume data, new algorithms which address intrinsic requirements of volume graphics have been developed. The problems of slow rendering and huge data sets can now be overcome by the availability of increasing memory capacity and the graphics chipsets [9] which are designed specifically for volume graphics. These together have made possible the migration of the computation-intensive components in the volume rendering pipeline down to the hardware level.

The ray-casting approach requires intensive computation for rendering a volume. The view dependent nature implies that each time the viewpoint is changed, the computation will have to be carried out again. Interactive rendering is therefore difficult to achieve. Various methods have been proposed to reduce the rendering time. In [7], Laur and Hanrahan describe a progressive refinement algorithm which is based on hierarchical splatting. Gouraud-shaded polygons are used to approximate the splats. Since graphics workstations are optimized to draw these polygons, the rendering time is improved.

Rendering a subvolume instead of the entire volume will also speed up the frame rate. In [6], Shin et. al. suggest that only a thin slab within the volume is drawn. In this way, not only the rendering can be made faster, occluding organs and structures can also be removed.

Accelerated volume rendering algorithms using 3D texture mapping hardware are proposed in [3, 4, 5]. A volume is treated as a 3D texture image which is loaded into texture memory. Planes are then drawn orthogonal to the view direction. The 3D texture is mapped to the planes by supplying appropriate texture coordinates for the vertices of the polygons. Direct volume rendering is done by blending the planes back-to-front into the frame buffer using a specifically chosen alpha blending function.

Besides hardware enhancement, graphics library may also provide effective operations for volume data. Recently, Westermann and Ertl [3] propose several efficient techniques to edit volume data with OpenGL and its extensions.

3 System Overview

VISBONE is a 3D visualization system developed at The University of Hong Kong for orthopaedic surgical planning of screw insertion. It uses direct volume rendering accelerated by 3D hardware texture mapping to achieve interactive frame rates. In this section, we will describe the steps in preparing the CT data sets and also the features of the current system.

3.1 Data Set Preprocessing

Radiologists use a *window* to define a range of the scanned pixel values, called CT values, to highlight the tissue of interest contained in CT images (Fig. 1). Within the window range, density values will be assigned more intensity levels so that minute density variations can be magnified and be shown clearly. A bone window is defined accordingly to highlight the bone structure as shown in Figure 1(c). The series of CT slices are then piled together to form a volume which is loaded as a 3D texture.

3.2 Main Features

VISBONE runs on an SGI Onyx2 workstation equipped with an Infinity Reality2 graphics board, MIPS R10000 CPU, 512 Mbytes of main memory and 16 Mbytes of hardware texture memory. Accelerated volume rendering with 3D hardware texture mapping is used to achieve interactive bone display. Apart from this, several features are supported by the system which are mainly for 3D visualization of the BMD. These features are described in the following subsections.

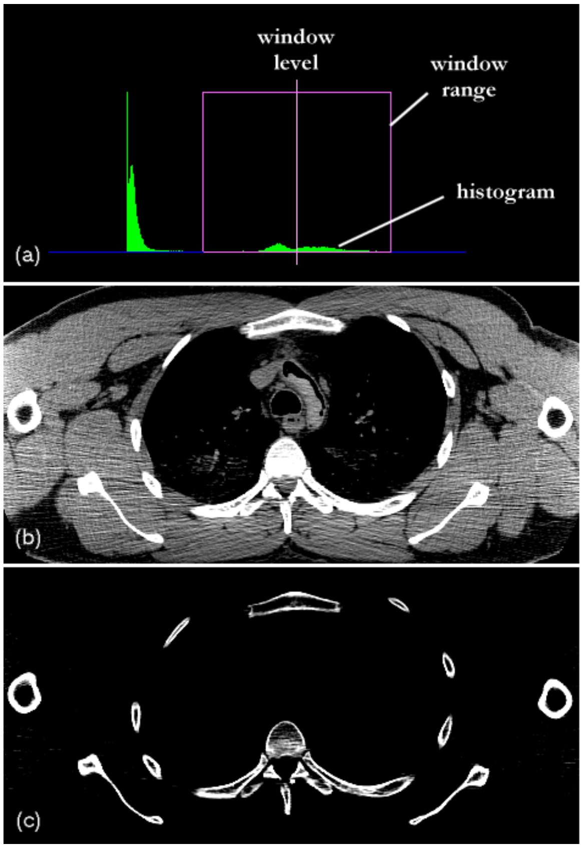

Figure 1: Preprocessing: (a) The histogram: intensity levels will only be assigned to density range within the window range centered at the window level; (b) The effect with soft tissue window; (c) The effect with bone window. The slice shown is from the Visible Human CT data set.

3.2.1 Clipping

Clipping is provided as a basic feature in many visualization toolkits [12]. The main function of clipping in our application is removing part of the bone to reveal the BMD distribution over the cutting plane (Fig. 2(b)). Although other geometries, such as cones and spheres, can be treated as clipping primitives as described in [3], planes are used in all clipping modes. The reason is that surgeons are most confident when dealing with cross-sections and interpreting density changes on a planar surface.

A new clipping modes has been incorporated into VISBONE. Parallel slices within the volume are drawn as shown in Fig. 2(c). The slices are separated by a certain adjustable distance. The orientation of the slices can easily be changed by handling a normal vector to the parallel slices. Different regions can be visualized when the slices slide along the plane normal. The rest of the volume can be made highly transparent to provide a 3D spatial relationship between the series of 2D slices. With this setting, surgeons no longer need to rely on mental visualization to stack up the 2D CT images for a 3D picture.

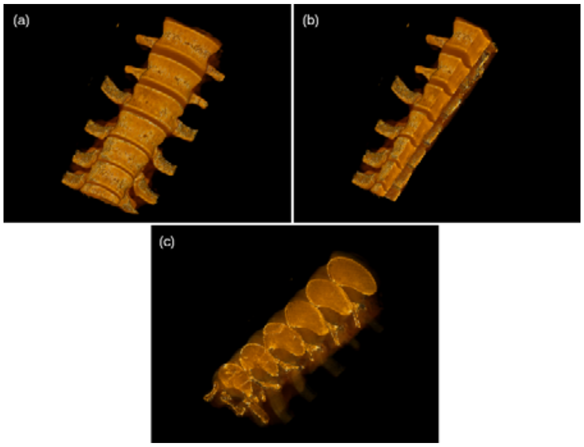

Figure 2: (a) The spine; (b) Clipping planes in arbitrary orientations; (c) Parallel slices.

3.2.2 Partitioning

Partitioning is used to visualize the BMD distribution in different bands. The entire range of density values of the CT data is divided into several sub-ranges. Voxels with their densities falling into different sub-ranges are partitioned into different bands or groups, i.e. voxels within the same group have their densities in the same sub-range. Each group of voxels can be turned on or off so that they are visible or invisible (Fig. 3). This visualization mode can show how a particular band of BMD is distributed inside the volume. Also, since most of the voxels can be excluded by choosing only a band to be viewed, serious occlusion among the voxels is reduced.

Initially, boundaries resulting from partitioning were expected to look as regular as mechanical parts, as by the surgeons we consulted about this feature. This turned out not to be the case because there are low density voxels scattered all over the volume. Therefore there is no obvious boundary between different bands of BMD. The reason for such discrepancy is that in their understanding surgeons have in mind a rough boundary between the bands without considering the existence of tiny pores, which are shown as low density voxels in the visualization. See Fig. 3(a).

3.2.3 Color Coding

Color plays an important role in the human visual system. Voxels are assigned colors according to their

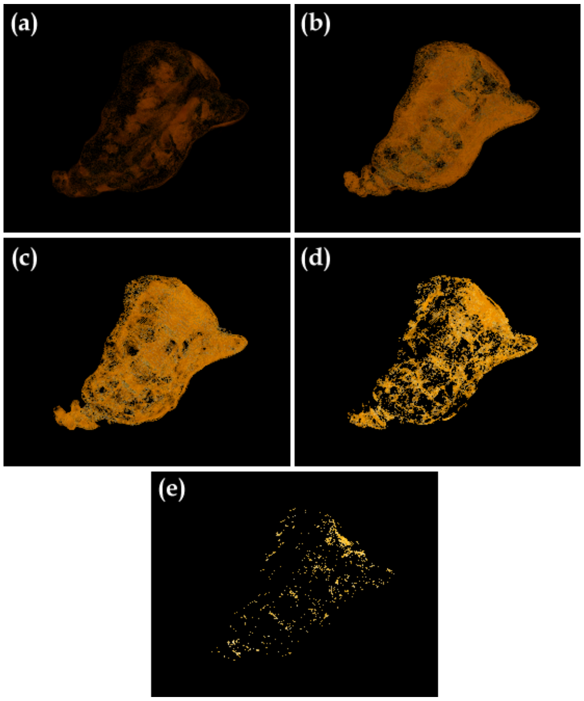

Figure 3: Partitions. The density range is divided into 5 partitions. (a)-(e) show these bands from low density region to high density region.

BMD. Different coding schemes have different visual effects. The selection of a color coding scheme depends highly on the application. In VISBONE, several coloring schemes are available for the users to choose from. Various visual effects can be achieved by combining color coding with different opacity settings for the voxels. For instance, an X-ray like visualization can be obtained by using grey-scaled encoding with all voxels having a very low opacity.

3.2.4 Animation

A feature has been implemented to provide animation effects. This feature can show the evolution of the volume from low density voxels to high density voxels. Two modes are supported: accumulative and non-accumulative. In the accumulative mode, subsequent voxels with increasing density values are added to the scene until the whole volume is visualized. In the non-accumulative mode, only voxels of a particular band are shown at any instance.

Animation is a useful visualization technique in situations involving dynamic data transitions, such as flow visualization. Although VISBONE deals only with a static 3D scalar field, animation is still applicable, as transitions between frames can be interpreted as the change of focus from one density range to another. This animation feature proves to give the user a better understanding of BMD variations within the volume than still images.

3.2.5 Transfer Functions Editing

VISBONE provides means to edit the opacity or intensity transfer functions interactively. The transfer functions are defined as piecewise functions instead of continuous ones (Fig. 4). This offers the user more flexibility to adjust attributes, such as intensity and opacity, in different density ranges. For each density range, the transfer function can be linear or constant. For instance, the opacity transfer function can be made directly proportional to the density values for the low density range. However, it can also be inversely proportional to the density values for the high density range. The result of the editing will be visualized immediately. Combining the piecewise transfer functions and the ability to freely divide the density range, various visual effects can be achieved. We shall discuss in section 4.2 how the transfer function editing feature can be applied to screw positioning in orthopaedic surgical planning.

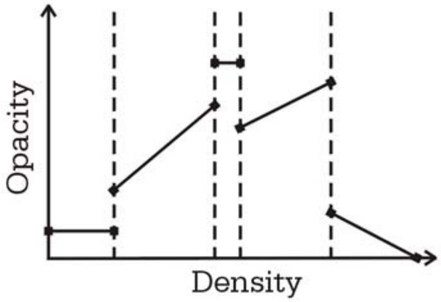

Figure 4: Piecewise transfer function. The function is defined independently in each density range.

With special treatment on the data volume, the transfer functions may be defined according to the locations rather than the density values of the voxels. An example of this is surface identification, which will be discussed in section 4.1.

3.2.6 Measurements

Besides visualizing the BMD of the bone volume, quantitative measurements are also important for planning screw insertion. For instance, a slab is cut from the volume by two parallel planes and the thickness of this section needs to be determined (Fig. 5(a)). Also, when selecting a screw of appropriate length, the distance between two locations within the bone volume needs to be found (Fig. 5(b)). However, specifying a certain location in the volume is not a trivial task since exact 3D pin-pointing is difficult. In VISBONE two points can be picked on an appropriately positioned cutting plane. The plane is one that passes through the two points whose distance is to be measured.

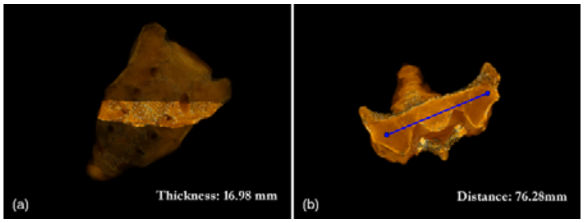

Figure 5: Measurements. (a) Thickness of a slice; (b) Distance between two points on a clipping plane.

3.2.7 Feasible Insertion Path

One of surgeons' main conern is whether a screw will be inserted into bone regions with sufficient bone mineral densities. Once the user has specified a screw insertion point on the bone surface, VISBONE will generate a map indicating the feasibility of inserting the screw into the volume from the entry point in all directions. The minimum BMD that will be encountered along each direction is shown. In the bone display, a spherical subvolume centered at the screw insertion point will be clipped away from the bone [3]. The radius of the sphere is equal to the length of the screw as selected by the user. A pointer representing the insertion direction has its tail fixed at the screw insertion point and the head being able to move freely on the spherical clipping boundary (Fig. 6(a)). At the same time, the region corresponding to where the pointer points to on the feasibility map will be highlighted (Fig. 6(b)).

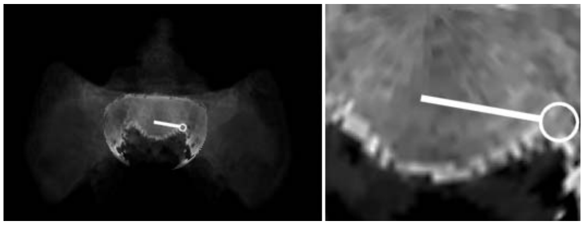

Figure 6: Feasible Path Indication. (a) Bone display; (b) Feasibility map. The circled region in (b) corresponds to the pointer head in (a).

Section 4.3 describes in details the techniques used in generating the above mentioned map.

4 Discussions

With the special requirements posed by our application, i.e. visualization of 3D BMD distribution, instead of just resorting to individual feature described above, it is sometimes necessary to apply special processing to the data in order to produce satisfactory visualization. In this section we shall discuss some of these cases arising in our test, as well as our solutions.

4.1 Surface Removal

If the opacity of a voxel is made proportional to its density value, the visualization of the bone will look like a surface model. This is due to the existence of a thin layer of high density voxels lying on the bone surface, and this layer makes the bone surface highly opaque, thus blocking the interior bone structures. To reveal the internal bone structures, elimination of the surface layer of the bone is mandatory. A preprocessing step is therefore needed to identify the bone surface voxels, which will then be treated differently from other internal voxels.

Given the density range of bone structures, a voxel will be classified as bone surface voxel if it satisfies the following criteria:

1. the voxel itself is of bone density value, and
2. at least one of the 26 neighboring voxels does NOT have bone density value.

						✓
					✓	✓
✓	✓			✓	✓	▨
▨	✓	✓	✓	✓	▨	▨
▨	▨	▨	▨	▨	▨	▨
▨	▨	▨	▨	▨	▨	▨
▨	▨	▨	▨	▨	▨	▨

Figure 7: The 2D analogy of bone surface finding. White and grey pixels represent soft tissue and bone, respectively. The checked pixels are classified as boundary pixels. All of them are bone pixels and at least one of the 8 neighboring pixels is a soft tissue pixel.

A 2D analogy is shown in Figure 7. The above criteria is simple and easy to test. Also, the thickness of the resulting surface can readily be adjusted by altering the number of neighboring voxels being tested. After identifying the surface voxels, they can be treated differently from the other voxels. The color can be changed so as to make the surface distinguishable from the others. The surface can also be made almost transparent to provide a spatial hint for the relative positions of the interior voxels within the volume.

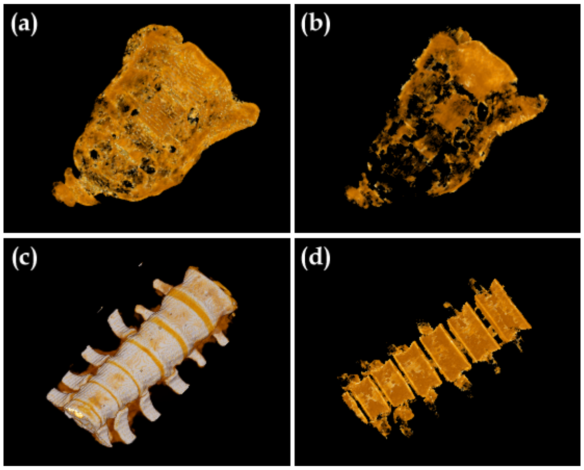

Figure 8: Surface removal. (a)&(c) show the sacrum and the spine without removing the surface. (b)&(d) show the two bones with a thin surface layer being removed, using the same opacity and intensity transfer function as in (a)&(c).

It is clearly shown in Figure 8(a) and 8(c) that without surface removal, the interior bone structures cannot be seen because these regions are blocked by the high density bone surface layer. With the opacity and intensity transfer function unchanged, removing the surface layer assists in visualizing the interior tissues to some extent. In Figure 8(b) and 8(d), each vertebrae can be identified clearly and also the interior BMD distribution is clearly visualized.

Surgeons sometimes would also like to have a rough idea on how thick the high density surface layer is. This surface layer in general does not have uniform thickness over the bone. The difficulty in extracting the surface layer is that the inner boundary of the layer cannot be simply identified by a threshold value which differentiates the hard and soft bone tissues. The inner boundary of the surface layer depends not only on the threshold value but also on the topology and location of the surface region. Another interpretation of this is to consider the threshold to vary among different surface regions. A convenient way to tackle this problem is to peel off a layer of 1 voxel thickness, layer by layer. Surgeons can control the number of thin layers to be peeled off until a certain portion of the volume has been removed or a targeted region can be seen.

4.2 Thresholding

In section 3.2.5, we discussed using the transfer function editing feature to facilitate the determination of screw positions. The idea is as follows: the density range is divided into two regions by a threshold value as shown in Figure 9. This threshold is the lowest density value which will be considered "safe" to insert the screw. By applying different transfer functions to the two regions, the visualization can be much enhanced.

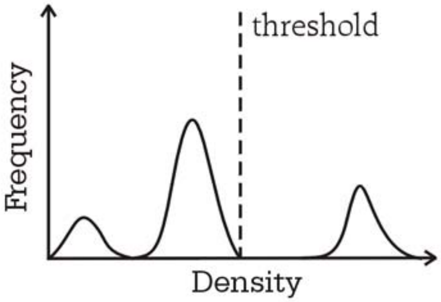

Figure 9: Setting up a threshold value on the histogram.

For instance, a low opacity value can be assigned uniformly to regions with density values lower than the threshold (Fig. 10). This will make such regions almost transparent. The *qualified region* (i.e. region where it is "safe" to insert the screw into) can be highlighted by raising the opacity of voxels with their density values greater than the threshold.

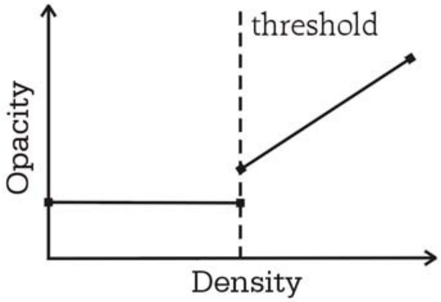

Figure 10: Applying a low opacity to density values below the threshold and high opacities to density values above the threshold.

4.3 Insertion Path Feasibility Map

4.3.1 Map Generation

An insertion path can be considered as feasible if all the BMD along the path are above a preset threshold. To qualify each path originated from an insertion point, we perform a perspective minimum intensity projection (MIP) with the insertion point acting as the ray converging point.

To achieve fast and responsive perspective MIP, we make use of the OpenGL `GL_MIN_EXT` blending equation to render the bone. We also rely on the stencil buffer to limit the effective rendering region to be the spherical subvolume centered at the insertion point. The radius of the sphere is defined to be the length of the selected screw. This is mandatory for filtering out voxels which the screw cannot reach to prevent these voxels from affecting the final result.

Before rendering each slicing polygon parallel to the image plane, a clipping plane is set up at the same position as the slicing polygon which clips against the sphere centered at the insertion point. By setting up the stencil buffer appropriately, only voxels within the intersection of the sphere and the clipping plane are rendered. The blending effect will thus be along paths orginated from the insertion point.

Figure 11 shows the result of hardware accelerated perspective MIP. The final images highly resemble that of the relatively slow perspective MIP by ray-casting while attaining real-time rendering.

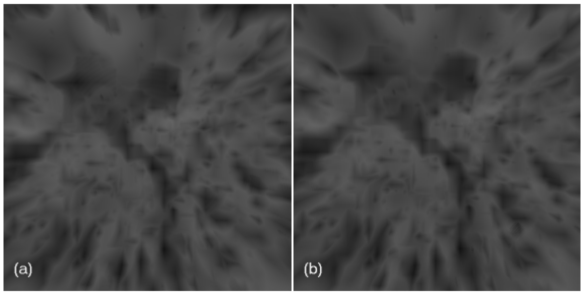

Figure 11: (a) Ray-casted perspective MIP; (b) Hardware accelerated perspective MIP,

4.3.2 Pores Removal

Although perpective MIP provides means to obtain information on insertion path feasibility, it is vulnerable in situations where an unqualified voxel of low density comes very close to the insertion point. The voxel will affect more paths than when it is farther away (Fig 12). All paths passing through it will be claimed unqualified. On the other hand, the definition of a feasible path may as well be relieved to allow for the existence of small pores along the path. Unqualified regions consisting of voxels with density lower than the threshold may still contain feasible paths if the regions are not significantly large.

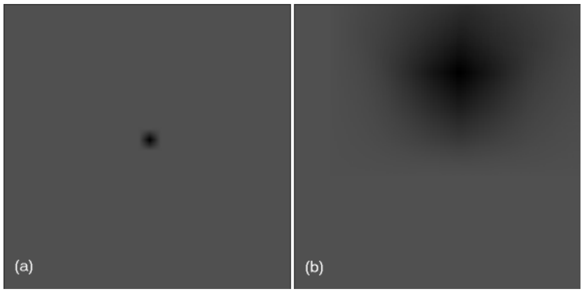

Figure 12: Perspective MIP for a voxel (a) far away from; (b) close to, the screw insertion point.

To address the above issues, the bone volume is preprocessed to remove small pores with densities lower than the preset threshold. A common image processing technique called *closing* is applied to the volume but in a 3D manner. A 3D filter is set up whose dimension is half of the maximum allowed dimension of an unqualified region. After the closing operation, low density regions smaller than the maximum dimension allowed will be filled up.

The closing operation can be decomposed into two stages: dilation followed by erosion. The dilation filter is applied to qualifed voxels and fill up unqualifed voxels within the filter. In order to ensure the same result regardless of the sequence in applying the filter, an unqualified voxel will take the maximum density of its surrounding qualified voxels. This can also eliminate the influence of the voxel in the MIP process.

The algorithm for applying the dilation filter is as follows:

```
Let $D$ be the dilation-filtered volume and $V$ be the original volume.
Let $V(i)$ denote the BMD of voxel $i$ in $V$.
$D \leftarrow V$
    For each voxel $i$ with $V(i) > threshold$
        center a $d \times d \times d$ filter $O$ at $i$, where $d$ is
        half the maximum dimension of unqualified region
        For each voxel $j$ in $O$
            If $V(j) < threshold$ and $D(j) < V(i)$
            then $D(j) \leftarrow V(i)$
```

Figure 13 shows the perpective MIP of the orginal volume and the result after applying the dilation filter. Small unqualified regions are filled up. However, the size of large unqualifed regions is also reduced.

As part of the the closing operation, the second step is to apply the erosion filter to moderate the result. The erosion filter is applied to unqualifed voxels and

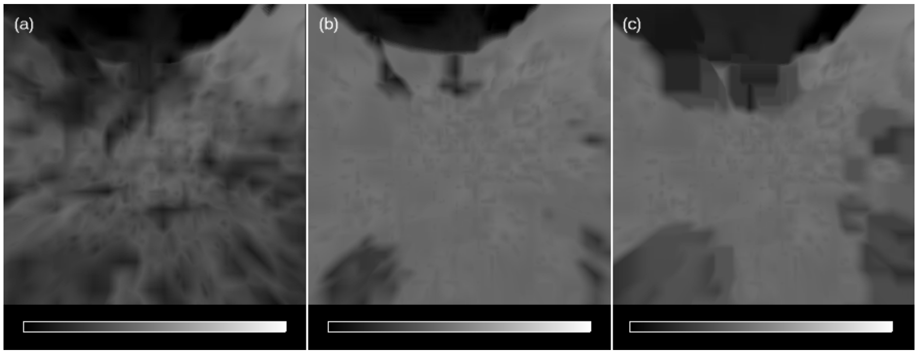

Figure 13: (a) Original perspective MIP; (b) After dilation; (c) After erosion. The legend under each image shows the color scale over the BMD range.

can enlarge unqualifed regions. Since small unqualifed regions are filled up by the dilation filter, erosion has no effect in these regions. The algorithm for applying the erosion filter is as follows:

```
Let $E$ be the erosion-filtered volume and $D$ be the dilated-filtered volume.
Let $D(i)$ denote the BMD of voxel $i$ in $D$.
$E \leftarrow D$
    For each voxel $i$ with $D(i) < threshold$
        center a $d \times d \times d$ filter $O$ at $D(i)$, where $d$ is
        half the maximum dimension of unqualified region
        For each voxel $j$ in $O$
            If $E(j) > threshold$ and $E(j) < D(i)$
            then $E(j) \leftarrow E(i))$
```

Figure 13(c) shows the result of applying the erosion filter. Note the recovery of the large unqualified region (in dark) at the top of the image.

4.4 Histogram

In the process of assigning intensity and opacity to voxels, the histogram plays an important role. Two issues about histogram operations, window setting and histogram equalization, will be discussed in this section.

4.4.1 Windowing

In section 3.1, we mentioned applying window setting to 2D CT images during the data preprocessing stage. This step can in fact be performed at a later stage, where the window setting will be applied directly to the 3D volume data. We choose not to do so because we are only concerned about bone density values in this application, so static window setting will be sufficient for our purpose. Dynamic window settings will be desirable when an application needs to highlight structures of different density ranges.

Another advantage of applying the window setting during the preprocessing stage is to reduce texture memory consumption. In each CT slice, each pixel is represented by 2 bytes. With preprocessed window setting, we can convert the CT values inside the window to 256 intensity levels, using only 1 byte to represent each voxel value. So the size of the resulting volume will be reduced by half and a larger volume can be fitted into the texture memory.

4.4.2 Histogram Equalization

Image enhancement is necessary in many applications where image quality is the primary goal. We tried to apply histogram equalization to the window defined on the 2D CT slices. Although the quality of the final rendering has been enhanced, it is then relatively hard to tell the differences in the densities based on color separation shown on the color scale. That is because histogram equalization does not result in a linear mapping from density values to voxel values, which are used to determine the voxel intensities.

Besides, a non-linear color scale may also give a false impression on how the densities are distributed over the volume [13]. Two colors located at different positions of the color scale can convey the difference between the values represented by the two colors. Ideally, if two pairs of colors have the same differences, the perceived differences provided by the two pairs of colors should be the same. If this cannot be achieved, the user will have difficulty in interpreting differences in the density values by directly comparing the color differences, and will need to frequently refer to the color scale. A color scale can be linearized such that the perceived differences between consecutive colors are as uniform as possible. For applications where density distribution visualization is of higher priority than image quality, a combination of linear mapping from density values to voxel values and a linearized color scale is recommended.

5 Conclusions

We have presented a visualization system to assist surgeons to analyze bone mineral density distribution for planning screw insertion. It has been observed that the special requirements raised by this application cannot be fulfilled by common features available in most general-purpose visualization toolkits. Hence, we need to develop a new volume-based visualization system enhanced with a number of tailor-made functions for our particular application.

On-going research includes simulations of screw insertion process which provide a pre-operation evaluation of bone structures after implant. This involves fast interaction of two volumes, each for the bone and the screw.

We also plan to test visual enhancements by applying stereoscopic display to visualize a CT volume. Hopefully, this can reveal more depth information and provide extra spatial hints on how the densities are distributed.

Haptics feedback is one of the planned development. When this function is incorporated, surgeons will have more opportunities to practise surgical techniques prior to actual operations.

References

[1] K. Akeley. Reality Engine Graphics. *ACM Computer Graphics, Proc. SIGGRAPH'93*, pp. 109-116, July 1993.

[2] P. Haeberli and M. Segal. Texture Mapping as a Fundamental Drawing Primitive. *Proc. Fourth Eurographics Workshop on Rendering 1993*, pp. 259-166.

[3] R. Westermann and T. Ertl. Efficiently Using Graphics Hardware in Volume Rendering Applications. *ACM Computer Graphics, Proc. SIGGRAPH'98*, pp.169-177.

[4] A. Van Gelder and K. Kwansik. Direct Volume Rendering with Shading via Three-Dimensional Textures. *ACM Symposium on Volume Visualization '96*, pp. 23-30.

[5] B. Cabral, N. Cam, and J. Foran. Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware. *ACM Symposium on Volume Visualization '94*, pp. 91-98.

[6] Y.Y. Shin, S. Napel, and G.D. Rubin. Fast Sliding Thin Slab Volume Visualization. *ACM Symposium on Volume Visualization '96*, pp. 79-86.

[7] D. Laur and P. Hanrahan. Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering. *ACM Computer Graphics, Proc. SIGGRAPH'91*, pp. 285-288.

[8] A. Kaufman. *Volume Visualization.* IEEE Computer Society Press, Los Alamitos, CA, 1991.

[9] H. Pfister, A. Kaufman, and F. Wessels. Towards a Scalable Architecture for Real-Time Volume Rendering. *Proceedings of the 1995 Eurographics Workshop on Graphics Hardware*, pp. 123-130, Maastricht, The Netherlands, August 1995.

[10] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29-37, March 1988.

[11] B. Lichtenbelt, R. Crane, and S. Naqvi. *Introduction to Volume Rendering.* Hewlett-Packard Company, 1998.

[12] W. Schroeder, K. Martin and, B. Lorensen. *The Visualization Toolkit*, 2^{nd} ed., Prentice Hall, 1998.

[13] H. Levkowitz. *Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications*, Kluwer Academic Publishers, 1997.

[14] T. Laine, D. Schlenzka, K. Mäkitalo, K. Tallroth, L. Nolte, and H. Visarius. Improved Accuracy of Pedicle Screw Insertion With Computer-Assisted Surgery. *SPINE*, Volume 22, Number 11, 1997, pp. 1254-1258.

[15] M. LaRouere, J. Niparko, S. Gebarski, and J. Kemink. Three-dimensional X-ray Computed Tomography of the Temporal Bone as an Aid to Surgical Planning. *Otalaryngology-Head and Neck Surgery*, Volume 103, Number 5, Part 1, November 1990, pp. 740-747.