Fast Opacity Control in Rendering of Volumetric CT Data *

Balázs Csébfalvi, László Szirmay-Kalos, Gábor Márton
Department of Process Control, Technical University of Budapest,
Budapest, Műegyetem rkp. 11, H-1111, HUNGARY
cseb@fsz.bme.hu

Abstract:

In 3D rendering of volumetric CT data the alpha-blending technique can be used for modeling semi-transparent tissues. This method assigns different opacity values to different tissue types. In practice, only four materials can be separated according to their CT numbers (Hounsfield density): air, fat, soft tissue and bone. Application of alpha-blending rendering enhances the more important tissues, assigning high opacity value for important tissues and low opacity for less important ones. For each pixel of the image, a ray is traced and the colour of the appropriate pixel is computed as a sum of intersected voxel colours weighted by opacity values. However, this computation is very slow if the size of 3D array is big. Furthermore, if the opacity value of one of the tissues needs to be changed, then the complete time-consuming rendering must be repeated in order to recompute the new weighted sums. In order to allow fast control of opacity values, a new algorithm is presented that provides fast opacity and colour modification after preprocessing the volume with fixed point of view and light source locations.

Keywords: Volume Rendering, Alpha blending, Ray Tracing, Medical Imaging, CT.

Introduction

The input data of the volume-rendering process is a 3D array of acquired $f_0(x_i)$ values at voxel locations x_i. In Computer Tomograms (CT), the data describe the Hounsfield density (CT number) distribution [NFML88].

The traditional pipeline of processing a volume array is the following [Lev88]: The first step is *data preparation* which may include correction for patient's motion in CT data, contrast enhancement and interpolation of additional samples. The output of this step is an array of prepared values $f_1(x_i)$.

This array is used as an input to the *shading computation* which results in an array of voxel colours $c_\lambda(x_i)$, $(\lambda = r, g, b)$. In a separate step, the array of prepared values is used as an input to the *classification procedure*, yielding an array of voxel opacities $\alpha(x_i)$.

Rays are then cast into these two arrays from the viewpoint. For each ray, a vector of colour samples $c_\lambda(x_i)$ and a vector of opacities $\alpha(x_i)$ are computed by resampling the voxel database at K evenly spaced locations along the ray and trilinearly interpolating from colours and opacities in the eight voxels closest to each sample location. The colour of the pixel of the associated ray is computed by a sum of colours weighted by opacities according to the alpha blending technique [Lev88].

Having the image generated by this pipeline, there is no way to effectively change the colours and opacities that are assigned to different tissues since it would require the repetition of the complete ray tracing phase.

*This work has been supported by the National Scientific Research Fund (OTKA), ref. no.: F 015884

In practical situations, however, the physician has usually some preliminary idea from where he intends to look at the volume. Having looked at the first image, he usually starts searching for possible defects eliminating unimportant parts by making them transparent or applying segmentation or cutting. Thus in practice, opacity and colour values have to be changed more frequently then viewpoint or lightsource locations.

This paper presents a method that optimises the rendering pipeline for fast opacity and colour control. To speed up opacity and colour control, the proposed method exchanges the processing phases. The data preparation is the same as in the traditional pipeline, but the next step is ray tracing or resampling that uses the $f_1(x_i)$ density values as input. The densities in sample points can be computed by trilinear interpolation from density in the eight voxel closest to each sample location. The gradient vector required in shading step is also computed by trilinear interpolation. The output of ray tracing is a vector of density samples for each ray. These values are then used as an input to shading and classification steps.

There are several classification models that define opacity map as a function of density and absolute value of approximated gradient vectors. For the sake of simplicity, our algorithm uses threshold classification. According to the density, a colour and an opacity value are assigned to sample vector elements. The shaded colours are computed from colours assigned to sample points according to the shading model. The colour of a pixel that is associated with a ray is computed by alpha-blending as in the traditional pipeline.

Our method preprocesses the prepared density volume during resampling and builds a structure to each ray, providing fast opacity and colour modification, which does not require the recomputing of weighted sums from the original volume. The preprocessing depends on the viewpoint, so when this parameter is changed, it must be repeated.

Classification model

The traditional classification assigns to each voxel the percentages of different materials present in the voxel [NFML88]. There are four major materials that can be separated by CT numbers. In general, each voxel represents a mixture of these materials. In the human body, however, voxels that are mixtures of more than two materials are rare. This classification model assumes that only the following mixtures are possible: air/fat, fat/soft tissue, soft tissue/bone. Assigning an opacity and a colour to each material, the opacity and colour values of different voxels are sums weighted by the percentage of materials present in the voxel.

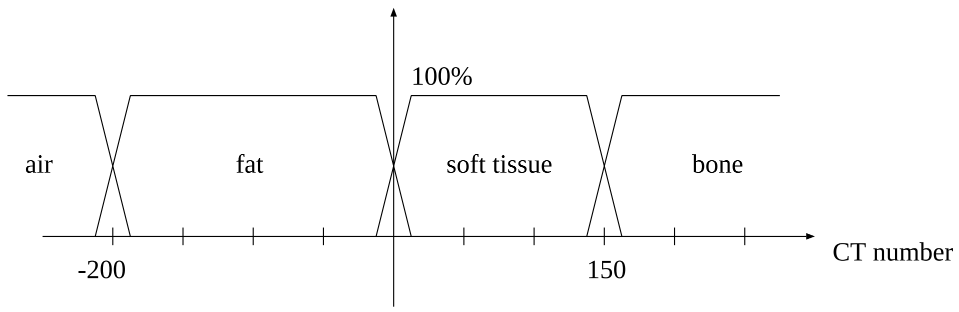

Figure 1: Classification of tissues

In the classification model the percentage of different materials is a function of the CT number, that can be approximated by a trapezoidal function (Figure 1) [DCH88]. These functions can be used easily because they are defined by only four parameters. These parameters are defined by the tissue type and are the same for different CT volumes.

After determining the trapezoidal classification parameters, a colour and an opacity value are assigned to each of these tissue types. The assigned values represent the colour and opacity of a voxel containing 100 percent of that material. When less than 100 percent of a material is present in the voxel, the colour and opacity are scaled by the percentage of occupancy. When the voxel contains a mixture of different materials, then the opacity value is the sum of the weighted opacity values of individual materials. This classification scheme is implemented with a lookup table, which allows for rapid classification.

Since physicians are interested mostly in the boundaries between tissues, from which the sizes and spatial relationships of anatomical features can be inferred, opacity values are usually multiplied by the absolute value of approximated gradient vectors. This absolute value is high at the boundaries between tissues, and low in a homogeneous environment (Figure 2).

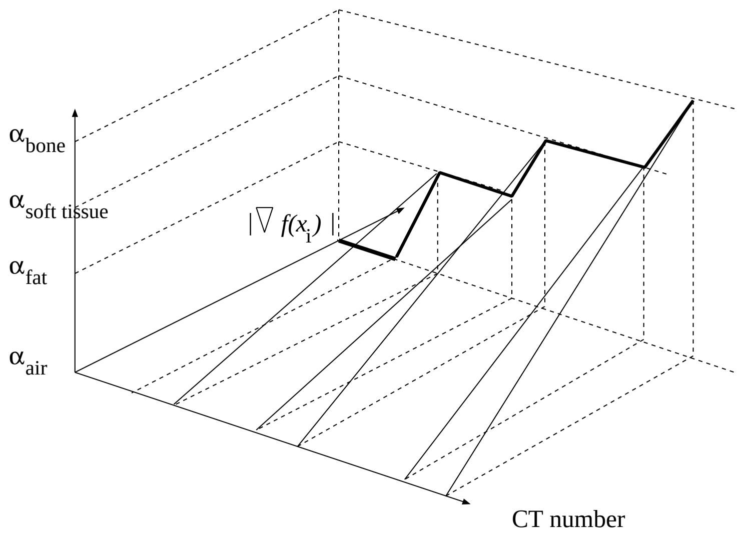

Figure 2: The Opacity Function

Another way of defining the opacity function is the application of linear interpolation in the following way [Lev88]:

$$\alpha(i) = |\nabla \hat{f(x_i)}| \begin{cases} \alpha_{v_{n+1}} \cdot [\frac{f(x_i)-f_{v_n}}{f_{v_{n+1}}-f_{v_n}}] + \alpha_{v_n} \cdot [\frac{f_{v_{n+1}}-f(x_i)}{f_{v_{n+1}}-f_{v_n}}] & \text{if } f_{v_n} \leq f(x_i) < f_{v_{n+1}}, \\ 0 \text{ otherwise,} & \end{cases} \quad (1)$$

where $f(x_i)$ is the density of ith sample point, f_a,f_f,f_s and f_b are the typical density values of air, fat, soft tissue and bone respectively, and $|\nabla \hat{f(x_i)}|$ is the approximated gradient.

For the sake of simplicity, our algorithm uses the following opacity function, where l_f,l_s,l_b are the lower density thresholds of appropriate materials:

$$\alpha(i) = \begin{cases} \alpha_a \text{ if } f(x_i) < l_f, \\ \alpha_f \text{ if } l_f \leq f(x_i) < l_s, \\ \alpha_s \text{ if } l_s \leq f(x_i) < l_b, \\ \alpha_b \text{ if } l_b \leq f(x_i). \end{cases} \quad (2)$$

Shading

In the shading step, the Lambert's model [SKe95] can be used, that requires the surface normal vectors in each voxel location. In the traditional volume rendering pipeline the normal vector is the approximated gradient [Lev88]:

$$\begin{aligned}|\nabla \hat{f(x_i)}| = [&\frac{1}{2}\cdot(f(x_{i+1},y_i,z_i) - f(x_{i-1},y_i,z_i)),\\ &\frac{1}{2}\cdot(f(x_i,y_{i+1},z_i) - f(x_i,y_{i-1},z_i)),\\ &\frac{1}{2}\cdot(f(x_i,y_i,z_{i+1}) - f(x_i,y_i,z_{i-1}))].\end{aligned} \tag{3}$$

Since in our algorithm the input of the ray tracing is the prepared volume of CT data, the approximated gradient is computed by trilinear interpolation from the density in the eight voxels closest to the sample location. Shaded colours of sample points are then computed according to the Lambert's model:

$$c_{\text{shaded}}(x_i) = c(x_i)\cdot\frac{\nabla \hat{f(x_i)}}{|\nabla \hat{f(x_i)}|}\cdot L, \tag{4}$$

where L is the direction of the lightsource.

Preprocessing

Having prepared the volume of CT data, rays are cast into this array from the viewpoint via pixels of the image plane. In the shading and classification steps an opacity and a colour value are assigned to each sample point. Starting with the sample point farthest from the viewpoint, let $c(i)$ be the shaded colour and $\alpha(i)$ the opacity of ith sample location. The colour of a pixel associated with the viewing ray is computed recursively according to the following formula:

$$C_{\text{out}}(i) = C_{\text{in}}(i)\cdot(1-\alpha(i)) + c(i)\cdot\alpha(i) \tag{5}$$

where

- $C_{\text{out}}(i)$ is the weighted sum after the ray has left the ith sample point,
- $C_{\text{in}}(i)$ is the weighted sum before the ray enters the ith sample point,
- $\alpha(i)$ is the ith opacity sample,
- $c(i)$ is the ith colour sample.

If $c(0)$ is the background colour and $\alpha(0)$ is 1 (fully opaque), then the resulting colour of n samples is

$$C_{\text{out}}(n) = \sum_{i=0}^{n} c(i)\cdot\alpha(i)\cdot\prod_{j=i+1}^{n}(1-\alpha(j)), \tag{6}$$

where $C_{\text{out}}(n)$ is the colour of the appropriate pixel.

For each opacity assigned to different tissue types, the colour of a pixel is a polynomial function of degree at most n. Suppose that the opacity of air is 0 and its colour is black, thus the $c(i)$ and $\alpha(i)$ coefficients of air samples can be neglected. In this case there are only three possible opacity values in the weighted sum. The formula can be further simplified by separating it according to the homogeneous segments intersected by the viewing ray.

Let us take an example. Consider a typical ray intersection sequence that contains the following homogeneous segments:

$$A_1(6) - F_1(2) - S_1(3) - B_1(6) - S_2(2) - F_2(1) - A_2(5),$$

where $A_i(j)$, $F_i(j)$, $S_i(j)$ and $B_i(j)$ denote the ith air, fat, soft tissue and bone sample segments respectively, containing j number of sample points. It means that the ray first passes through 6 air samples, then 2 fat samples, then 3 soft tissue samples, etc.

Ignoring the air segments, the weighted sum can be separated into terms associated with the different homogeneous segments:

$$C_{\text{out}}(n) = \sum_{i \in F_1} c(i) \cdot \alpha(i) \prod_{j=i+1}^{n} (1 - \alpha(j)) +$$

$$\sum_{i \in S_1} c(i) \cdot \alpha(i) \prod_{j=i+1}^{n} (1 - \alpha(j)) + \sum_{i \in B_1} c(i) \cdot \alpha(i) \prod_{j=i+1}^{n} (1 - \alpha(j)) +$$

$$\sum_{i \in S_2} c(i) \cdot \alpha(i) \prod_{j=i+1}^{n} (1 - \alpha(j)) + \sum_{i \in F_2} c(i) \cdot \alpha(i) \prod_{j=i+1}^{n} (1 - \alpha(j)), \quad (7)$$

where F_i, S_i and B_i denote sets of those j indexes, for which the jth sample is in the ith fat, soft tissue or bone segment, respectively.

Let $g_{T_k}(\alpha_t)$ be the piecewise linear approximation of

$$\sum_{i \in T_k} c(i) \cdot \alpha_t \cdot \prod_{j > i, j \in T_k} (1 - \alpha_t), \quad (8)$$

where T_k is the set of i indexes, where ith sample point is in the kth homogeneous segment of type T. Using these piecewise linear approximation functions, the $\hat{C}_{\text{out}}(n)$ approximation of $C_{\text{out}}(n)$ can be written as

$$\hat{C}_{\text{out}}(n) = \prod_{F_2}(1-\alpha_f) \prod_{S_1 \cup S_2} (1-\alpha_s) \prod_{B_1}(1-\alpha_b) \cdot g_{F_1}(\alpha_f) + \prod_{F_2}(1-\alpha_f) \prod_{S_2}(1-\alpha_s) \prod_{B_1}(1-\alpha_b) \cdot g_{S_1}(\alpha_s) +$$

$$\prod_{F_2}(1 - \alpha_f) \prod_{S_2}(1 - \alpha_s) \cdot g_{B_1}(\alpha_b) + \prod_{F_2}(1 - \alpha_f) \cdot g_{S_2}(\alpha_s) + g_{F_2}(\alpha_f) \quad (9)$$

Note that calculation of the colour in this example requires the computation of the weighted sum of equation 9. The structure of this equation is similar to this, thus the key to effective colour computation is the efficient evaluation of such weighted sums.

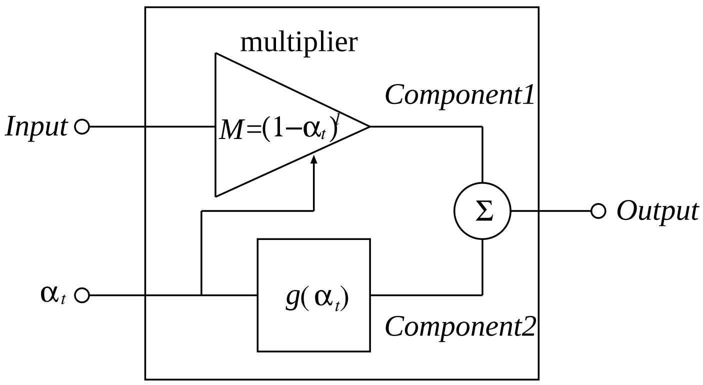

Figure 3: The processing element (PE)

This weighted sum can be computed with a linked list of Processing Elements (PE) shown in figure 3. A processing element consists of a multiplier unit that multiplies the *Input* with $M = (1 - \alpha_t)^k$, a box that is responsible for approximating $g(\alpha_t)$ from α_t and an addition unit that adds up the two calculated components.

The notations applied by figure 3 are as follows:

- *Input* is the previous weighted sum before ray enters this segment,
- *Output* is the current weighted sum after ray has left this segment,
- t is the type of the tissue of the segment,
- k is the number of sample points in the segment.

Let us denote the multiplier term $(1 - \alpha_t)^k$ (k is the length of the segment) by M. This function can be implemented with lookup tables, where the size of a lookup table is the maximum length of homogeneous segments of type t. These lookup tables must be refreshed whenever the opacity of a tissue is changed.

The application of the proposed processing elements and lookup-tables are presented in the following algorithm providing fast opacity and colour control:

- In a preprocessing phase, a linked list of the proposed processing elements is built for each pixel of the image, using the ray passing through this pixel. In this structure there is a reference to the first processing element for each tissue type. The type of processing element is the same as the type of the represented tissue.
- The fast opacity control loop, which generates the colour C_{ray} for each ray, is the following:

```
for each ray,
    Find the first PE of current tissue.
    Let it be the current PE.
    Let end-of-list = false
    while (not end-of-list)
        if type of current PE is the current tissue then
            Refresh multiplier M and Component2 by new opacity value.
        endif
        Let Output = M · Input + Component2.
        if next PE exists then
            Let the Input of next PE = Output.
            Let the current PE be the next PE.
        else
            Let end-of-list = true
        endif
    endwhile
    Let C_ray = Output. (This is the colour of the current pixel).
endfor
```

The presented algorithm gives approximated weighted sums because $g(\alpha)$ functions are implemented as piecewise linear functions. Increasing the number of segments in this approximation the output will be more exact, but memory requirements will also be higher.

The discussed structure supports also fast changing of the colours assigned to the different tissues because only the $g(\alpha)$ functions depend on the colours. The $g(\alpha)$ values associated with the division points have to be multiplied by the ratio of the new and the old colour values of current tissue. Then only processing elements of the current tissue have to be refreshed.

Implementation

The discussed algorithm has been implemented in C++ language. The program has been tested with a $255 \times 255 \times 128$ CT volume of a human head (Figure 4). The test machine was a 486 DX2/66 IBM PC. The results of the tests are the following:

Tissue type	previous opacity value	next opacity value	running time (sec)
fat	0.000	0.235	1.59
fat	0.235	0.541	1.59
soft tissue	0.047	0.352	2.20
bone	1.000	0.784	1.71
bone	0.784	0.509	1.70
soft tissue	0.352	0.501	2.25
bone	0.509	0.431	1.76
fat	0.541	0.823	1.59

Table 1: Running time measurements

Figure 4: Human head used in the tests

Conclusion

The preprocessing algorithm presented in this paper provides fast changing of opacity and colour values assigned to different tissues. The preprocessing assumes fixed point of view and light location parameters. After preprocessing the generated structure can be saved in a file that represents a 3D image of the CT volume. An appropriate viewer for this file can provide the rapid changing of the opacity and colour values. Since during 3D analysis of CT data the radiologist modifies the opacity and colour values more frequently than point of view or light parameters, such an image viewer could be useful in practice. On the other hand, in medical information systems the main problem is to record CT data of patients because of the enormous size of volumes. The proposed structure requires much less disc capacity than the direct storage of CT volumes, but still has much higher information depth than simple bitmap images.

References

[DCH88] A.D. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *Computer Graphics*, 22(4), 1988.

[Fru91] M. Fruhauf. Combining volume rendering with line and surface rendering. In *Eurographics '91*, pages 21–32. North Holland, 1991.

[GEJ96] R. Grosso, T. Ertl, and Aschoff J. Efficient data structures for volume rendering of wavelet-compressed data. In *Winter School of Computer Graphics '96*, pages 103–112, Plzen, Czech Republic, 12–16 February 1996. University of West Bohemia.

[Gro95] E. Groeller. *Scientific Visualisation.* Technical University of Vienna, Institute of Computer Graphics, Vienna, 1995. in German.

[Lev88] Marc Levoy. Display of surfaces from ct data. *IEEE Computer Graphics and Application*, 8, 1988.

[NFML88] Derek R. Ney, Elliot K. Fishman, Donna Magid, and Marc Levoy. Computed tomography data: Principles and techniques. *IEEE Computer Graphics and Application*, 8, 1988.

[SKe95] L. Szirmay-Kalos (editor). *Theory of Three Dimensional Computer Graphics.* Akadémia Kiadó, Budapest, 1995.