



HAL
open science

An Integrated Multi-purpose Ray Tracing Framework For The Visualization Of Medical Images

J.L. Dillenseger, C. Hamitouche, J.L. Coatrieux

► **To cite this version:**

J.L. Dillenseger, C. Hamitouche, J.L. Coatrieux. An Integrated Multi-purpose Ray Tracing Framework For The Visualization Of Medical Images. Annual International Conference of the IEEE Engineering in Medicine and Biology Society Volume 13: 1991, Oct 1991, Orlando, United States. pp.1125-1126, 10.1109/IEMBS.1991.684374 . hal-04677000

HAL Id: hal-04677000

<https://hal.science/hal-04677000v1>

Submitted on 24 Aug 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AN INTEGRATED MULTI-PURPOSE RAY TRACING FRAMEWORK FOR THE VISUALIZATION OF MEDICAL IMAGES

J.-L. Dillenseger, C. Hamitouche and J.-L. Coatrieux

Laboratoire Traitement du Signal et de l'Image, Inserm U335,
Université de Rennes 1, Campus de Beaulieu, Rennes, France

ABSTRACT

A generalization of ray tracing techniques to render medical volume data is described. The proposed framework is based on the combination of primitive algorithms operating along the rays. This combination can be reduced to sequential tasks or can lead to more complex processes. Two particular applications are considered to demonstrate the improvements that can be obtained.

1. INTRODUCTION

In the medical field, most of the imaging modalities (CT, MRI, PET, ...) provide a sequence of 2D cross-sections. They allow to reconstruct 3D sampled functions known as volume data. Different visualization techniques are available to display 2D projections of these data. They can be classified into surface (polygon- or voxel-based) and volume rendering (ray tracing, octree encoding) methods [1]. A discussion of these methods, as well as the main issues in 3D medical imaging such as special purpose hardware, image analysis and therapeutic applications, as been recently reported in [2]. However, significant advances within each methodological approach remain to be done to better handle the computer resources and to improve the overall performance (e.g. segmentation, tissue characterization, etc...). This paper is focused on ray tracing techniques applied to a 3D grid of voxels. The first section describes a new conceptual framework. The second section is aimed at demonstrating the validity of these ideas. Two applications are presented, involving detection and interpolation algorithms, and the results are discussed.

2. GENERALIZED FRAMEWORK FOR RAY TRACING

The integrated ray tracing framework is aimed at providing basic functionalities involved in medical image rendering (such as dissection, manipulation, quantification) without **any prior processing**. No restrictions are assumed in terms of the sampled functions representing the objects, the observer viewpoint, the surface or volume display.

This framework is completely defined by :

- the nature of the primitive operators: they can be devoted to filtering, enhancement, surface detection and shading computation as well as compositing (for example colors and opacities [3]), and they can be linear or non linear, recursive or not.
- the spatial dimension in which they operate (one dimensional operators up to three dimensional).
- the logical masks where the operators are applied (they can specify screen areas from which the rays are casted but also the 3D bounding volume of a specific region of interest).
- the traversal mode along the ray : front to back, back to front, mixed and even iterative modes can be used.
item the combination rules allowing to specify the "on the fly" processing sequence along the rays.

The first and the last two points are here the key issues. Let us take two situations from which a number of schemes can be derived:

- sequential operations : the overall processing can be stated as : $F_n(F_{n-1} \cdots F_2(F_1(I)))$ where I denotes the data, F_1 the innermost function which is applied first to define the input to F_2 and so on. Surface rendering can be easily represented this way (refer to 3.1).
- composition of functions : it can be expressed, omitting the arguments of the operators, by: $C(F_1, \dots, F_n)$. The composition can stand for weighting multiple analysis results, mask controlled operations, combination of parameters in sampled multiparameter images. To illustrate this concept, a complex process is reported in 3.2 allowing to hypothesize a surface location and to refine it after 3D interpolation.

3. METHODS AND RESULTS

Two original algorithms are considered to demonstrate the benefits of this unified approach. The first one deals with the 3D boundary detection in an isotropic volume. The second one merges local surface detection with 3D interpolation

to render **anisotropic** data within the ray tracing frame. These methods have been implemented on a HP 9000 835 computer.

3.1. Statistical detection-estimation scheme

Sequential detection algorithms, within the class of the so-called jump detection approaches [4], are based on the design of a stochastic marker indicating a change (i.e. arr edge here) and statistical rules. These rules are aimed at deciding between the two hypotheses "change (H_1) / no change (H_0)" and also at locating the spatial coordinates of the change. Without loss of generality (more sophisticated markers such as autoregressive models could be used). surface detection is performed by looking for a change in the mean. The problem can be stated as follows.

Let ε_n be a white noise sequence with variance σ_a^2 and v_n be a sequence of observations (i.e. v_n represent the values of the voxels intersected by a given ray) such that:

$$v_n = \mu_n + \varepsilon_n$$

with

$$\mu_n = \begin{cases} \mu_0 & \text{if } n < r \\ \mu_1 & \text{if } n \geq r \end{cases}$$

where μ_0 and μ_1 are the mean values before and after the jump, and r the estimated edge position along the ray. When the means are a priori known, the decision (H_0 against H_1) can be formulated in a maximum likelihood framework and leads to the Page-Hinkley stopping rule or the Cumulative Sum (Cusum) test. In the more realistic case where μ_1 and δ (the jump magnitude) are unknown, a solution can be obtained by setting δ to the minimum value to be detected (δ_m) and to run the following test:

$$\begin{cases} U_0 = 0 \\ U_n = \sum_{k=1}^n (v_k - \mu_0 - \frac{\delta_m}{2}) \\ M_n = \min_{0 \leq k \leq n} U_k \\ \text{detection when } M_n - U_n \geq \lambda \text{ (}\lambda: \text{decision threshold)} \end{cases}$$

This test handles a decrease in the mean. A similar formulation can be derived to look for an increase. The edge position corresponds to the last maximum M_n observed before the detection.

This method has been applied to MRI isotropic data and the results are depicted in Figure 1.

The automatic detection of the gyri is known as a difficult problem because of the noisy nature of the data and the tissue contrast variability. The computation of the mean has been achieved on a $3 \times 3 \times 3$ local window centered on the ray.

The parameters were set to $\delta = 150$, $\lambda = 750$. This example points out that 2D windowing, surface detection and localization, normal estimation and shading (gradient based method) can be concatenated together and that they provide fairly good results.



Fig. 1. 3D rendering of a head reconstructed from 109 MRI slices (256×256) (courtesy of the University of North Carolina). The automatic detection of the gyri based on ray tracing with jump detection operator is performed in a 2D screen window, interactively defined.

3.2. Interpolating ray tracing

When dealing with data acquired in clinical situations, the spatial image sequence is completely defined by the slice number (coordinate along the axial direction), the slice thickness and the inter-slice spacing and leads to an anisotropic data volume definition. This situation is classically handled through a preprocessing step (e.g. interpolation) to recover a cubic grid description. Consequently, a larger number of voxels have to be stored. The key improvement offered by the present technique is that it allows the image display directly from the original, anisotropic 3D set. This capability alleviates the required memory resources at the cost of a small increase in computation time .

The sequence of operations involved in the interpolating ray tracing is :

- (1) delimitation of the image area by projecting the bounding volume onto the screen and definition, for each ray, of the entry point and the maximal path depth (these operations are equivalent to 2D and 3D masks respectively),
 - (2) casting of a ray throughout the anisotropic volume and search for a rough intersection (approximate detection),
 - (3) local resampling of the original volume in a neighborhood centered at this point (an interpolation operator is mn),
 - (4) search for the refined intersection point along the viewing ray within the resampled volume (new detection step),
 - (5) shading computation from surface normal estimation.
- This method has been applied to complex and simple shaped structures. Its effectiveness is illustrated in Figure 2 where the operations (2) and (4) make use of thresholding when (3) is carried out by linear interpolation. Quantitative evaluations have been performed on different databases, with several de-

tection and interpolation (spline based) operators. They show that this simple scheme represents a well balanced solution with respect to the quality of the rendering and the computational cost (in average the increase is about 20%). The coefficient of memory saving is at least 2 (one interpolated slice).

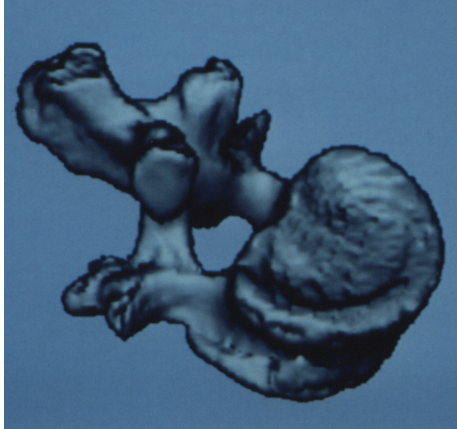


Fig. 2. 3D rendering with interpolating ray tracing of a vertebra reconstructed from 51 CT slices (256×256) (courtesy of the South Hospital of Rennes). Pixel size: 1 mm, slice thickness and inter-slice spacing : 2 mm (anisotropic ratio is 2). The coefficient of memory saving is 2 (one interpolated slice).

4. DISCUSSION

The framework we have described was designed to provide more flexibility and capability to the ray tracing technique. It can be applied to preprocessed or raw data, scalar or vectorial images. Sophisticated processing can be described by sequencing multiple operators or by merging them into a single one. The former has been illustrated in 3.2 where a simple thresholding is used for surface detection. The latter has been depicted in 3.1 where the detection estimation algorithm performs also a smoothing operation. The performances depend both on the quality of the primitive functions and on the correct use of the combination rules. The operations are done locally which could lead to unsatisfactory results when contextual informations are required. In that case, multiple rays (e.g. beams) have to be designed. The only inconvenient of the approach comes from the additional computations to carry out, however, the main operating modes can be parallelized and implemented on multi-processor units.

5. CONCLUSION

An integrated approach has been proposed to display and to analyze 3D medical data. It relies on local transformations applied along the rays, further combined in a sequential or

non sequential way. More emphasis has been put on the binary classification problem (e.g. surface detection) and the interpolation issue. Validation has been worked out on real data.

6. REFERENCES

- [1] COATRIEUX J.L., BARILLOT C., A survey of 3D display techniques to render medical data, in 3D Imaging in Medicine, K.H. Höhne Ed., Springer Verlag, 1990, pp 175–195.
- [2] 3D Computer Medical Imaging, Special Issue, IEEE Eng. Med. & Biol., 9, 4, 1990.
- [3] LEVOY M., Display of surfaces from volume data, IEEE Comp. Graph. & Appl., 8, 4, 1988, pp 29–37.
- [4] BASSEVILLE M., BENVENISTE A., Design and comparative study of some sequential jump detection algorithms for digital signals, IEEE ASSP, 31, 3, 1983, pp 521–535.