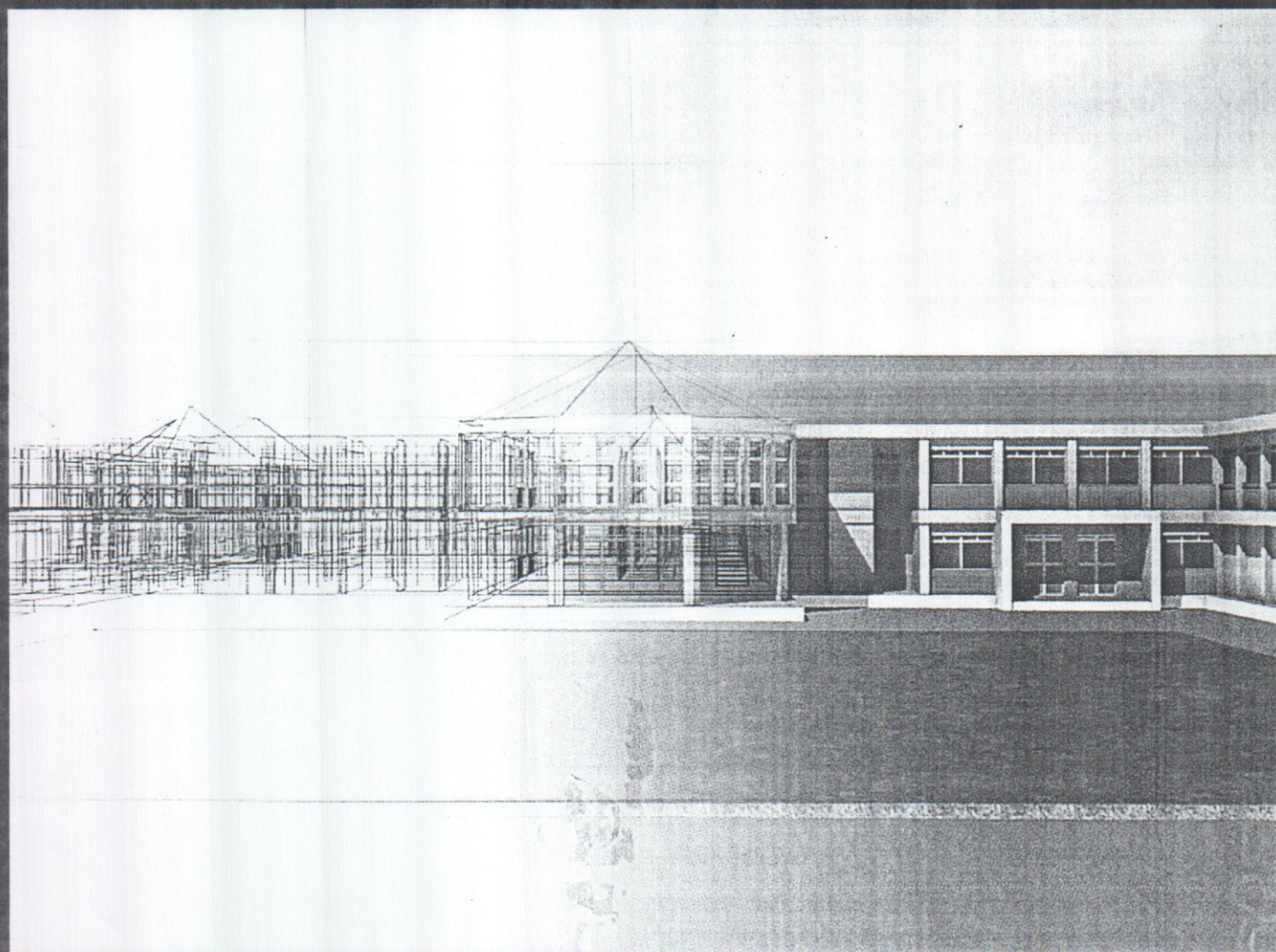


SVR 2007

IX SYMPOSIUM ON VIRTUAL AND AUGMENTED REALITY
MAY, 28-31, 2007

PETRÓPOLIS - RJ - BRAZIL

PROCEEDINGS



Proceedings

SVR 2007 IX Symposium on Virtual and Augmented Reality

Petrópolis, RJ. May 28-31 , 2007

Organizing Committee

General Chair
Jauvane C. de Oliveira
Laboratório Nacional de Computação Científica – LNCC

General Co-Chair
Alexandre Cardoso
Universidade Federal de Uberlândia – UFU

Program Committee

Chair
Romero Tori
SENAC-SP / Universidade de São Paulo – USP

Co-Chair
Alberto Raposo
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Edited By:

Romero Tori – Universidade de São Paulo – USP / Centro Universitário Senac – SENAC-SP
Alberto Raposo – Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio
Jauvane C. de Oliveira, Laboratório Nacional de Computação Científica – LNCC/MCT
Alexandre Cardoso, Universidade Federal de Uberlândia – UFU

Three-Dimensional Soil Physics Image Analysis Based on a Virtual Reality Environment

Leonardo C. Botega

Computer Department

Federal University of São Carlos

Rodovia Washington Luís, km 235 - CEP 13565-905

São Carlos - SP - Brazil

leonardo_botega@dc.ufscar.br

Paulo E. Cruvinel

Embrapa Agricultural Instrumentation

Rua XV de Novembro, 1452 - CEP 13560-970

São Carlos - SP - Brazil

cruvinel@cnpdia.embrapa.br

ABSTRACT

This paper presents the development of an environment to shelter the integration of three-dimensional reconstructed tomographic images with a synthetic ambient that provides the use of Virtual Reality tools, through a user interface. Such fusion occurs by the implementation of *Loader* classes, capable to import the resulting models of the reconstruction process and to prepare them to the VR resources treatment, such as visualization and manipulation techniques.

Keywords

Three-dimensional Tomographic Reconstruction, Virtual Reality Environment, Computerized Tomography, Soil Physics.

1. INTRODUCTION

Until the present day, just a few research units have developed projects with Virtual Reality (VR) applications for the teaching and training in specific areas, as the digital image processing and analysis, due the technical difficulties and high cost. However, some proposals are coming up to minimize the development difficulties, systems maintenance and necessary devices [4].

Under other perspective, the use of two-dimensional (2-D) images generally is not enough for a precise diagnostic, in studies that use the tomography as the main fountain of information, being necessary the use of three-dimensional (3-D) reconstruction techniques. Hence, developed algorithms to the three-dimensional reconstruction propose the composition of surfaces and volume of an object, starting from one-dimensional (1-D) projections and two-dimensional tomographic images.

To reconstruct an image starting from projections and provide a precise analysis platform, in a way to inspect the interior of a body or object in a non-destructive form, became a vast practical in the scientific visualization area. This practice becomes extremely important in many applications as medical diagnosis, radio-astronomy studies, microscopy and in soil studies.

2. OBJECTIVES

The main objective is to prepare a virtual desktop environment to give shelter to the fusion between 3-D reconstructed tomographic images, using digital image processing, and VR interactive techniques, needed by a virtual qualitative environment. Hence, a user interface was built, which, by 1-D tomographic projections as input, allows the user to visualize and manipulate 3-D images rebuilt, dynamically illuminated and under several rendering

manners. Tomographic rehearsals of images from the Science of the Soil are considered to the system's validation, since the obtaining of the samples until the three-dimensional analysis.

3. METHODOLOGY

The present work is divided into phases in the following order: (1) 2-D and 3-D Reconstruction, (2) Development of the VR Environment e (3) First and second phases fusion.

In the first phase of the work, an x and γ rays minitomographic scanner from Embrapa Agricultural Instrumentation was used for the tomographic images obtaining, making possible the generation of coefficient maps of x and γ rays attenuation, with spatial resolution equals or above 1 mm [1] [2].

To the 2-D reconstruction, a filtered backprojection algorithm was used, together with frequency domain filtering, using *Hamming* windows; implemented under 1-D *Fast Fourier Transform*, taking use of the C++ programming language [5]. Later on, with the two-dimensional image already rebuilt, an adaptative 2-D filtering technique was used, with *Daubechies Wavelet Transform* [3], filtering determined areas, preserving edges and the details of the objects represented in the image, through a 76 coefficients window.

In this paper, it was adopted the most diffused 3-D reconstruction technique for volumes and surfaces: the superposition of 2-D slices, taking use of the interpolation algorithm by *B-Wavelets* (or *B-Splines*), which determines the intermediate value on a sequence of known points, making the curve get through the estimated points [5].

To the Virtual Reality phase, a main user interface was developed as a system main control, which provides an intuitive access to all the resources above described. This main interface, through independent frames, also allows access to a 3-D visualization of the reconstructed model as a whole, as much as the analysis of its components, individually, by the user selection. All the resources and classes described in this phase were developed with the Java programming language and the Java3D API [6].

Physical transformations such as rotation, translation and scale, by keyboard input entries, were chosen to be initially implemented, either in the complete model frame, or in all other frames. These Geometric operations were performed through the *Transform3D* class, by a 4x4 transformation matrix of real numbers (*float*). Hence, objects from class *TransformGroup* specify a transformation through a *Transform3D* object, which is inherited

to all children. It must be considered that the effects in a Scene Graph are cumulative [6].

Later, *Behavior* objects were instantiated, responsible to interpret the user actions by input events and translate them into position transformations, also available in all system frames. The *Behavior* class makes the connection between impulses and reactions, providing node changes on the Scene Graph. In the *Behavior* class, there are several subclasses for the interaction, for example the *MouseBehavior* and the *Keynavigatorbehavior*, which controls mouse and keyboard events, respectively.

Taking use of the *MouseListener* subclasses and some graphical methods, responsible to capture the voxels data, the user is able to recover informations about the 3-D representation by mouse event. Among the available data, there are the distances between the model and the viewer, the model coordinates, the voxel color (in RGB format) and the 4x4 transformation matrix, used to recount new positions. The process to obtain such effect is described by the Source Code 1.

```
PickCanvas pickCanvas = null;
public static Color color = null;
Point3d p = new
Point3d(e.getX(), e.getY(), z_vrml_initial_position)
color=new Color(pickCanvas.pickClosest().
getClosestIntersection(p).getPointColor().get().
getRGB());
```

Source Code 1. Representation of the methods for information recovery

Also inserted in the VR system, a dynamic light control let the user choose between four sources of lighting, such as *Ambient*, *Directional*, *Point* and *Spot* (*PointLight* subclass). Each lighting choice loads a new frame, fill of several parameters, which have direct influence about the light direction, position, attenuation, angle and colors. The effects of the dynamic light control can be applied into all 3-D visualization frames. The Source Code 2 represents the main method description of the lighting system and its calls.

```
protected void createLight( LightObject light,
TransformGroup transform_g )
createLight(new AmbientLight(), transform_g);
createLight(new DirectionalLight(), transform_g);
createLight(new PointLight(), transform_g);
```

Source Code 2. Methods' description of the lighting system

Beyond the lighting process, the user is able to choose the 3-D rendering model, such as *Full Polygon Surfaces*, *Polygon Lines* (Wireframe) and *Polygon Points* (Point Cloud). These rendering options can be applied in the 3-D model as a whole or over its internal components, obtained by the isosurfaces demarcation. The rendering options are provided by the *PolygonAttributes* class, described by Java3D.

Later, a dynamic color system was implemented, aiming the interactivity, one of the original directives of VR systems. Thus, either the complete model, with all components, or a single part of the 3-D tomographic image, can be colored at runtime. The new colors of the models can be chosen by picking them on a 2-D color palette, built under a *BufferedImage* object and provided as an event of the interface to speed up the process, rather than a operational system palette event. The Source Code 3 basically

describes the process of obtaining a specified color by picking in a 2-D palette.

Finally, on the phase of technologies fusion, a new class was implemented, called *Loader*, derived from the package *Xj3D* [7], which provides subsidies to bring all the model characteristics to the Java environment, taking use of its previously implemented methods, portable to reuse.

```
BufferedImage buffered_image =
ImageIO.read(getClass().getResource("palette.png"));
Color=new Color(buffered_image.getRGB(event.getX(),
event.getY()));
```

Source Code 3. Representation of the main set of methods for the coloring process

The *Loader* class is capable to import the reconstructed model, obtained on the first phase, to the VR environment, keeping all its original characteristics, such as Materials, Appearance and Geometry. Also, the imported model can be treated under all the resources previously described. Among the file formats that can be imported are: *3D Studio* (.3ds), *Wavefront* (.obj), *VRML* (.wrl), *AutoCAD* (.dxf), *Visualization Toolkit* (.vtk) and others.

The file formats chosen to this study development were *Visualization Toolkit*, used for the reconstruction phase, *VRML*, for the model visualization as a whole, and *Wavefront*, for the components analysis.

4. RESULTS AND VALIDATION

The evaluation of the present study has been occurred under tomographic rehearsals, initially with samples of Soil Science and later on with agricultural samples.

Thus, the validation of the system occurs by the precise 3-D reconstruction, visualization and analysis of the tomographic images, obtained from the scanned samples. The manipulation of the reconstructed images in a Virtual Reality environment must allow a full interaction, providing the participation of the user with the scene.

So far, concerning the reconstruction phase, the new two-dimensional filtering shown good results about the noise reduction, when compared to the *Haar Wavelet Transform*, used in other implementations, as described by Pereira [5].

Concerning the phase that involves techniques of Virtual Reality, to implement the interface and all the necessary resources to the VR system, the Java programming language was used, under the Java3D API. Thus, all the structure of a Scene Graph was implemented, according to the previously described classes, supported by Java3D. Figure 1 shows the main interface and its controls.

Aiming the complete manipulation of the objects, in a totally interactive manner, the physical transformations and manipulations, applied on the 3-D tomographic models, filled their roles, turning input choices into satisfactory visual feedbacks. By providing the new positions and scale dimensions, the user can stretch, move and spin the models, through a canvas update option. Also, all representations can be rotated, translated and scaled, having precise visual feedbacks by mouse event. As the user picks the object in the scene, the transformation matrix is

updated and shown at runtime by the main interface, beyond other model data.

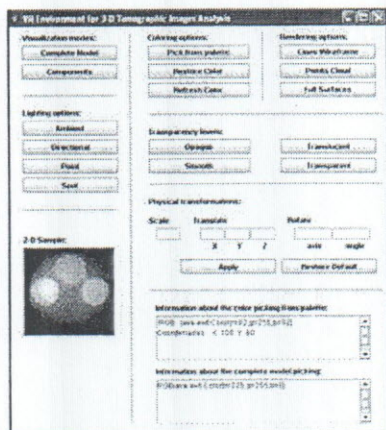


Figure 1. Main user interface with VR controls

By the lighting system, every lights of the present displayed view can be changed at runtime, translating the frames fields into visual feedbacks, depending on the source and the reflectance of the objects surfaces, which can be simply visualized with a full faced representation or analytically studied by their point and lines. Depending on the type of lighting, some parameters are changed, but always preserving the color, bounds and direction. Hence, the chosen light source is inserted in the scene, striking the normal vectors. Thus, by a refresh option, the scene is updated and the models are illuminated. The Figure 2 shows two examples of lighting, with different sources and color, applied on a complete model.

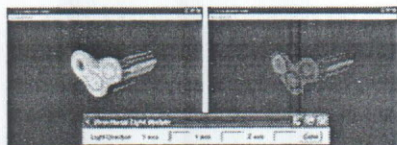


Figure 2. Representation of lighting system on a complete reconstructed model

The color system also contributes for the system interactivity, once it brings to the user the possibility to change the objects appearance at runtime, returning convincing visual and information responses. Provided by the main interface, the color palette allows the user to easily pick the desired color, as described by Source Code 3, and update the current model.

By the new options of the models rendering, an analytical view of the objects in the scene is provided, allowing the user to examine the points and lines generated by the reconstruction phase, specially the estimated ones. The Figure 3 shows examples of the color system usage, applied on different kinds of components and rendering.

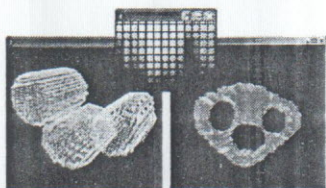


Figure 3. Examples of different kinds of components and rendering with the interactive color system

Considering the integration phase, aimed to bring the implementation of the 3-D reconstruction to the Virtual Reality environment, the reconstructed model could be treated by all the last cited concepts. Thus, the reconstructed model, saved under the *Visualization Toolkit* (.vtk) format, could be brought to the system by the *Loader* classes, preserving its original characteristics.

5. CONCLUSIONS

Considering the reconstruction phase, there was a filtering improvement with the 2-D *Daubechies Wavelet Transform* summed to the 1-D filtering with *Hamming* windows, compared with the unique 1-D filtering on the frequency domain, taking use of *Fast Fourier Transform* (FFT).

For the matter of the Virtual Reality environment development, the results presented satisfactory feedbacks, considering the mouse and keyboard manipulation, as well all the illumination, rendering and coloring process, crucial for a synthetic ambient destined to real model analysis. The next stage of this phase falls in with the model manipulation by a digital glove and the visualization by Head Mounted Devices, with LCD displays, aiming to reach immersion and a bigger interaction.

The phase of the models load shown great results, given the computational complexity involved by the joint of different technologies, even when used for convergent use. All the resources present in the *VRML* and *Wavefront* files could be accessed by the Java/Java3D ambient, such as Geometry, Appearance and Material, however, a few adjusts were needed for the complete adoption, mainly referred to the right positions of the models, to guarantee that the objects appear in the scene origin. By convention, the manipulation of the models position in the scene was made in the Java ambient, turning easy the integration with the resources presented in the main interface.

6. REFERENCES

- [1]Crúvinel, P.E., Cesario R., Crestana S., Mascarenhas S., x and y-ray computadorized minitomograph scanner for soil science. IEEE Transactions on Instrumentation and Measurement, v.39, N.5, p.745-750, (1990).
- [2]Crúvinel, P.E. Minitomógrafo computadorizado de raios x e y para aplicações multidisciplinares. Ph.D thesis. State University of Campinas – UNICAMP. Campinas, (1987).
- [3]Daubechies, I. Tem Lectures on Wavelets, Philadelphia. SIAM, (1992).
- [4]Kimer, C. Apostila do ciclo de palestras de realidade virtual, Atividade do Projeto AVVIC- CNPq (Protem - CC - fase III) – Federal University of São Carlos - UFSCar, pp. 1-10, (1996).
- [5]Pereira, M. F. L., Crúvinel P. E., Parallel DSP Architecture for Reconstruction of Tomographic Images Using Wavelets Techniques. Proceedings of XIV SIBGRAPI, p. 384-384. Florianopolis. (2001).
- [6]Sun Microsystems, Inc. Java 3D 1.3 API Documentation. Available at <<http://java.sun.com/products/javamedia/3D>>(2006).
- [7]Web3D Consortium. Xj3D Developer Documentation. Available at <<http://www.xj3d.org/arch.html>> (2006).