GRAPHIC ENVIRONMENT FOR INDUSTRIAL COMPUTERIZED TOMOGRAPHY

Authors: M.Iovea, A.Marinescu, P.Chitescu, T.Sava, C.Rizescu, Gh. Georgescu,
ICPE - BUCHAREST, ROMANIA.

Extended abstract

The Industrial Computerized Tomography was a field of concern in our institute since 1989, starting with high resolution non-destructive control investigation using computerized control means. In 1990 we achieved the first experimental model of a tomograph for small size objects, in 1992 we developed an industrial tomograph.

Using this equipment we carried out hundreds of tomograms, in our attempts to improve reconstruction algorithms for 2D and 3D Non-Destructive Testing analyzing.

During the first experimental stage we thought to develop same software facilities, for promoting some industrial applications [1,2].

In 1994 we started to use our equipment in archaeological objects analysis where we have some interesting results. In this field of research is much important to analyze an object without damaging any part of it, so the computerized tomography is one of the best methods to obtain complete information about the tested object.

The problem of the tomography is to determine the attenuation coefficients (μ) for every pixel in the scanned zone. If we build a histogram with the $\mu_{x,y,z}$ for the entire 3D matrix we can see that every material is in correspondence with a peak in that histogram.

There are some particularities in tomographic data: the objects are not clearly delimited. So if we have two objects, build from different materials, there are some intermediate values between those attenuation coefficients (scattering effect). For this reason, it is not possible to make a surface generation algorithm with large level of generality that get best results for all kind of objects and materials. So, we built a program that works directly in the 3D matrix without any surface generation and using the scattering effect to compute the normal of the surfaces at rendering time. The 3D matrix is obtained from successive 2D slices scanned with our industrial computerized tomograph. The step between slices is usually constant for all tomographed objects.

The software was designed using the new OOP concept and written in C++ for Windows target. It has a user-friendly graphic interface and many 2D and 3D processing facilities.

We use for rendering an algorithm that takes the following steps:

1. Compute the ray corresponding with each pixel from the screen image
2. Compute the intersection of this ray with the volume of voxels
3. If the ray does not intersect the volume then go to step 8
4. Find the first visible voxel (some materials could be declared invisible) intersected by that ray
5. If there is no visible voxel found then go to step 8
6. Compute the normal of surface. This computation is based on an original fitting of a cube centered in that voxel in a 4 - dimension space, where the last dimension is represented by the attenuation coefficient. So we take a point **P**(x,z,y) and a kernel centered in P with the size S (usual it has value 1). Every point has an additional dimension represented by the attenuation coefficient (scaled in a range 0-255). With all points contained in that kernel we make a fit with a plane by equation:

$$c = a_1x + a_2y + a_3z + a_4$$

We try to minimise the function:

$$\chi^2 = \sum_{i=1}^{N}(c_i - (a_1x_i + a_2y_i + a_3z_i + a_4))^2$$

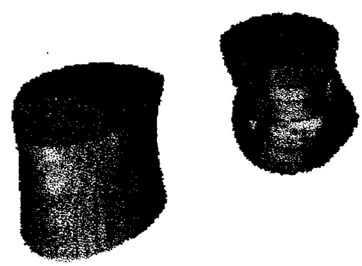
From this condition we compute the a_i coefficients from which we obtain the direction cosines of the normal of surface in point **P**:

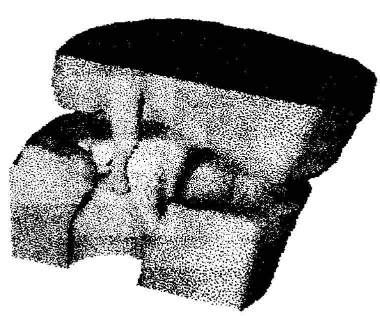
$$C_1 = \frac{a_1}{\sqrt{a_1^2 + a_2^2 + a_3^2}} \qquad C_2 = \frac{a_2}{\sqrt{a_1^2 + a_2^2 + a_3^2}} \qquad C_3 = \frac{a_3}{\sqrt{a_1^2 + a_2^2 + a_3^2}}$$

7. Compute the lighting using that normal and the point of the lighting source
8. Go to next ray in the scan

The computation time depends on the image size on the screen, the 3D array size, the number of colors declared invisible. For example, it is about 15 s for usual sizes (200 x 200 pixels on screen image and 150 x 150 x 150 voxels in 3D matrix) for an IBM PC - 486DX4 / 100MHz.

We take for example an ancient roman statue, whose gamma ray 3D tomography, obtained using above algorithm, is shown in figure. The matrix dimensions are 182 x 100 x 328 voxels and the file size is about 6 Mb. It is remarkable that our 3D tomographic representation is very realistic, looking alike with the visual image. This statue has a fixed broken leg that was restored some years ago and apparently the leg looks perfect. But in gamma ray tomography the leg looks with a pregnant flaw zone.





The advantages for using directly 3D matrix in representation are: very realistic image even in the presence of the scattering effect, much flexibility in allocation in color table and cursors modification, opening many possibilities in directly visualizing the effects of some 3D data processing (Spline and linear interpolation, cutting any part of object, filtering with different kernels to obtain some effects), extracting some materials, computing the volume for every material, low time for rendering, interactive modifying of the matrix, automatic material recognition by processing the histogram, detecting the peaks and computing the optimal domain for each color, etc. The drawback is the large matrix needed in the memory during the rendering. This tomographic program is an open system that allows all functions needed to be added eventually later by using dynamic libraries.