Comparison of irregular data structures for data reduced CT reconstruction

Benjamin Betz[1], Steffen Kieß[1], Kaicong Sun[1], and Sven Simon *[1]

[1]Institute of Parallel and Distributed Systems, University of Stuttgart, Stuttgart, Germany

Keywords: tomographic reconstruction, irregular data structures, octree, nested grid

Summary: To reduce the size of volume data generated by CT, the resolution of the voxels is inhomogeneous and voxels outside the reconstructed object are processed in low resolution. The quality of the object voxels is thereby not decreased. Several irregular data structures are compared, which are suited to define voxel volumes with different resolutions. The partially reduced voxel resolution leads to reduction of the volume data which also results in reduced computation times.

INTRODUCTION

In the last years, the resolution of computed tomography increased from 1 giga-voxel to 64 giga-voxels or more. The resulting three-dimensional datasets can reach sizes of over one hundred gigabytes. Due to the requirements of the scanning process, irrelevant voxels outside the object are reconstructed the same way all voxels are reconstructed. Furthermore, in some applications only a part of the reconstructed object is of interest. Typically, all parts of the volume are saved with the same resolution. One way to reduce the amount of data is to process the irrelevant parts of the volume in a lower resolution. The reduction of voxel data in the volume should also lead to a faster computation time as less data has to be processed. In order to find a suited data representation of inhomogeneous voxel volumes, three irregular data structures are investigated in this paper.

METHOD

Typically, a voxel volume is subdivided into equally shaped volume elements which form a regular grid [1] [2]. In order to relax this regularity, a rectilinear grid of arbitrary size for each dimension of the voxels is used and analyzed concerning memory consumption and computation time. As the object generally lies in the center of the volume, only the corners of the reconstructed object can be saved in a low resolution. Because the size of the voxels cannot vary over all voxels, the ability to reduce the volume data is limited significantly.

The second data structure being investigated is the octree. In the work of S. Kim et al. [3] the octree was used in an iterative CT reconstruction algorithm to extract the shape of an object. They refined the octree iteratively according to the geometrical complexity of the scanned shape. They were able to reduce the volume data from previously 3.1 gigabytes to 26 megabytes. Their results are not directly comparable to this work, because the authors improved the octree iteratively during the reconstruction process and achieved a high resolution only for the edges of the object. In this work, the octree is constructed before the reconstruction and is used to adjust the resolution of the voxel volume for the basic steps of the CT reconstruction. The octree can adapt the voxel volume more freely than the rectilinear grid. Therefore, it can reduce the size of the data even better. However, one major downside of the octree is the slow interpolation, which is for instance necessary to compute the projection step of the Maximum Likelihood Estimation Method (MLEM) [4].

The third data structure being considered is the nested grid. It divides the volume in partial volumes with a low resolution analogous to a regular grid. The partial volumes are then subdivided again in a regular grid pattern of an arbitrary size. This allows to adapt the volume resolution similar to the octree, but has the advantage to perform a faster interpolation.

*e-mail: Sven.Simon@ipvs.uni-stuttgart.de

RESULTS

The data structures previously discussed are evaluated by performing basic steps of the CT reconstruction based on conebeam-CT. The qualities of the data structures are obtained with the dataset of a Raspberry Pi [5] which consists of 800 projections with 992 × 992 pixels.

For the classical filtered backprojection proposed by Feldkamp, Davis, and Kress [6] in Table (c), the computation time and the memory requirements are reduced with all three data structures for the considered example. The rectilinear grid was able to half the original size and computation time. The octree performed better and was able to reduce the size of the volume structure to about 200 megabytes without losing the high resolution for the object. Similarly, the nested grid was able to reduce the size to about 500 megabytes. Both structures reduce the computation time to 3 hours.

Being part of the MLEM reconstruction algorithm, the projection is performed to demonstrate the impact of the complexity of the interpolation. In all computations the nearest neighbor interpolation is used. Because of the more complex interpolation, the rectilinear grid and the octree could not speedup the projection step. Only the use of the nested grid was able to achieve an improvement.

References

[1] Foley J. D., van Dam A., Feiner S. K. & John H. F., Computer Graphics: Principles and Practice, Addison-Wesley Longman Publishing Co., Inc. Boston, USA, 1990

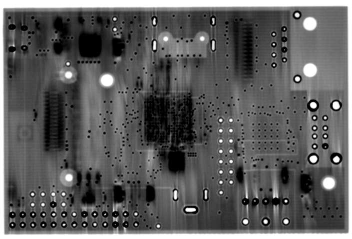
[2] Watt A., 3D Computer Graphics, Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1993

[3] Kim S., Sakane A., Ohtake Y., Suzuki H., Nagai Y., Sato K., Fujimoto H., Abe M., Sato O. & Takatsuji T., Efficient Iterative CT Reconstruction on Octree Guided by Geometric Errors, 6th Conference on Industrial Computed Tomography, Wels, Austria, 2016

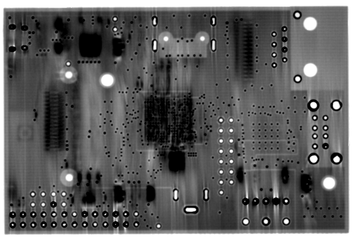
[4] Shepp L. A. & Vardi Y., Maximum likelihood reconstruction for emission tomography, IEEE Trans. Med. Imaging 1 113-22, 1982

[5] Raspberry Pi, Raspberry Pi Foundation, https://www.raspberrypi.org, Accessed: 2017-27-01

[6] L. A. Feldkamp, L. C. Davis & J. W. Kress, Practical cone-beam algorithm, 1984



(a)



(b)

Filtered Backprojection			
Regular	Rectilinear	Octree	Nested
38.24 h	16.87 h	2.98 h	2.89 h
3.9 GB	1.5 GB	207 MB	504 MB
Projection			
Regular	Rectilinear	Octree	Nested
61.38 h	72.9 h	86.20 h	54.71 h

(c)

Figure 1: A slice through the reconstructed volume of a Raspberry Pi is shown in (a) and (b). In Figure (a) the voxel volume is stored as a regular grid with volume data of 3.9 gigabytes whereas in Figure (b) with the use of an octree the size is reduced to 207 megabytes. In Table (c) the computation time and memory requirements of the filtered backprojection and projection is shown for all irregular data structures and compared to the regular grid.