

Institut für Computergraphik und
Algorithmen

Technische Universität Wien

Karlsplatz 13/186/2

A-1040 Wien

AUSTRIA

Tel: +43 (1) 58801-18601

Fax: +43 (1) 58801-18698

Institute of Computer Graphics and
Algorithms

Vienna University of Technology

email:

technical-report@cg.tuwien.ac.at

other services:

<http://www.cg.tuwien.ac.at/>

<ftp://ftp.cg.tuwien.ac.at/>

TECHNICAL REPORT

Visual Exploration for Parameter Studies of Industrial CT Dataset Series

M. M. Malik ¹ and C. Heinzl ² and M. E. Gröller ¹

¹Institute of Computer Graphics and Algorithms, TU Wien, Austria

²Upper Austrian University of Applied Sciences, Wels Campus, Austria

TR-186-2-08-12

December 2008

Visual Exploration for Parameter Studies of Industrial CT Dataset Series

M. M. Malik¹ and C. Heinzl² and M. E. Gröller¹

¹Institute of Computer Graphics and Algorithms, TU Wien, Austria

²Upper Austrian University of Applied Sciences, Wels Campus, Austria

December 17, 2008

Abstract

This paper proposes comparison and visualization techniques to carry out parameter studies for the special application area of dimensional measurement using 3D X-ray computed tomography (ICT). A dataset series is generated by scanning a specimen multiple times by varying parameters of an industrial CT device. A high resolution series is explored using our multi-planar reformatting (MPR) based exploration system. We present an evaluation system (multi-image view) and an edge explorer for comparison and visualization of grey values and edges of several datasets simultaneously. Visualization results and quantitative data are displayed side by side. For fast data retrieval and convenient usability we use bricking of the datasets, and efficient data structures. Graphics hardware is used for interactive visualization. We studied the applicability of our proposed visualization techniques in collaboration with our company partners.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1 Introduction

Industrial 3D X-ray Computed Tomography (ICT) is widely used in industry for nondestructive testing (NDT) and first part inspection, especially in the automotive, aeronautic, electronic, and leisure industry, as well as many other industrial fields. Quality-control engineers compare the volumetric datasets scanned using an ICT with the design specifications of the industrial components. This helps them to evaluate the precision and accuracy of the production pipeline. The resolution of the volumetric dataset directly effects the precision of the first part inspection.

A wide variety of Computed Tomography (CT) scanners have been introduced since the first CT machine was produced in 1972. Microtomography (MicroCT) [ED82] and Nanotomography (NanoCT) were produced in 1982 and 2005 respectively. The basic principle of data acquisition in MicroCT and NanoCT is the same as in early CT machines. The main difference is the size of the focal spot in the CT source. The size of the focal spot is in micrometers and nanometers in MicroCT and NanoCT respectively. A micro or nano focal spot results in higher resolution volumetric datasets. On the other hand a high energy macro-focus X-ray source better captures the structure of the specimen.

Lately, dual energy CT devices have been introduced to combine the advantages of high energy and low energy X-ray sources. A high energy and a low energy dataset are produced by a dual energy CT machine. The two datasets are later fused together in software. The voxels are typically stored with either 16 bit or 32 bit precision in an ICT dataset. Contemporary volumetric datasets have a high resolution and are quite large in size (table 1).

ICT machines provide various parameters that can be manipulated and set by the user before performing a CT scan. Some parameters severely influence the quality of a volumetric dataset depending on the material and geometry of the specimen. Some of the parameters that can be manipulated are voltage, current, number of projections, width and alloy of the X-ray filtering plates etc.

Unfortunately the parameters cannot be changed in real-time with live feedback from the ICT. A typical 3D scan takes approximately 30 to 45 minutes. Also the variations in the volumetric datasets, induced by changes in the parameter settings are hard to locate in 3D datasets. It is highly interesting for the ICT users to study the change in the volumetric datasets due to the parameter variations. The study helps to predict the parame-

Table 1: Dataset series. Each parameter produces a series.

parameter	DS ¹ resolution	no. of DS	size	voxel size
voltage	1000x1000x882	5	8.21GB	122 μm
no. of projections	1000x1000x882	6	9.86GB	136 μm
current	856x856x882	6	7.22GB	122 μm
integration time	800x800x882	5	5.26GB	136 μm
filter plates	848x848x882	15	17.72GB	122 μm
mean value dataset	1000x1000x882	3	4.93GB	136 μm

ters that will introduce the least amount of artifacts in the volumetric dataset of a given specimen. We present an innovative visualization system in our paper for the study of ICT parameters.

We visualize the ICT scan series in collaboration with our company partners. A dual energy source HWM Rayscan 250XE (voxel size: 5 microns, 225keV and 450keV micro focus tube) and a Phoenix: x-ray nanotom (voxel size: 0.5 microns, micro focus tube: 180keV) devices are used to generate CT datasets. A dataset series is produced by varying a single parameter over the entire parameter range. Table 1 gives information on the various dataset series that were used to test and evaluate our techniques. All the datasets in a series were generated in a row without removing the specimen from the ICT device.

Some of the parameters in table 1 have a known behavior. For example voltage can induce cupping at high penetration lengths due to beam hardening. A small number of projections in a single 360 degree turn of the rotary plate can result in visible streaking artifacts. The number of projections is directly proportional to the scanning time. A mean value dataset is produced by measuring each projection image multiple times and evaluating a mean image. Mean value measurement is supposed to better estimate the grey values but it increases the time and cost of each scan.

There are many public domain tools available like VGStudio MAX [VG04] for the exploration of a CT dataset. The design of such tools is mostly based on multi-planar reformatting (MPR) rather than 3D visualization techniques. The choice of MPR over 3D visualization is based on some valid reasons described below. We are not aware of any software that allows a simultaneous exploration and comparison of dataset series.

The huge amount of data produced from an ICT makes it hard to provide interactive comparison and 3D visualization of the volumetric datasets. The amount of main memory in the latest computer systems has increased considerably but the memory in the graphics hardware

remains the bottleneck. Graphics hardware is crucial for interactive rendering of volumetric datasets. We have a varying number of 3D datasets for every parameter and the amount of data that needs to be processed is rather large (table 1). Efficient management of the data and interactive rendering is possible through MPR (section 4). 3D visualization also has occlusion and shading issues. Shading typically alters the visibility of grey values according to the light in the scene. This is not the ideal case given the fact that we are interested in detecting variations of grey values with changes in the parameter settings. Also, 3D visualization does not provide as much detail as we can extract using MPR. The use of 3D visualization therefore negates the advantage gained through the use of latest MicroCT and NanoCT machines. MPR on the other hand can provide interactive exploration, comparison, and visualization of very large dataset series.

Our system is therefore based on MPR. We provide a *multi-image view* for simultaneous viewing and comparison of dataset series. The *edge explorer* provides comparative visualization of edges in a dataset series. We use efficient algorithms and data structures for fast retrieval of data from the main storage device.

2 Related Work

The simultaneous visualization of multiple datasets is mostly used in medical imaging. Datasets from different modalities are first registered and then visualized by the radiologists for diagnosis. Wilson et al. [WBO97] propose to use a different rendering style for each modality. The aim of multi-modal visualization is to render different parts of a volume using the modality which best captures the respective object part.

Some general techniques, which are not specific to medical imaging, have been proposed for rendering multi dimensional datasets [SJH08]. Ebert et al. [ERS*00] use innovative blobby objects for simultaneous rendering of multiple parameters. Each blobby object has as many bulges on its surface as the number of parameters or dimensions to visualize. The bulges point in different directions and this might impact comparison of parameters for the viewer. Also, the blobby objects are 3D and therefore the technique is prone to occlusion issues. Taylor [Tay02] uses color coding and orientation of simple shapes to visualize multiple datasets on a single surface.

Interactive rendering of multiple volumes is challenging. The amount of memory available in graphics hardware is limited. Time-space partitioning (TSP)

¹ Dataset

trees [SCM99] is an efficient data structure used for storage and rendering of time varying datasets. Fout et al. [FMA05] propose compression of the multi variate datasets. The technique assumes a certain correlation between variables. Grimm et al. [GBKG04] propose bricking and an efficient addressing scheme for large datasets that do not fit into the main memory. Kohlmann et al. [KBKG07] describe a bricked volume layout for MPR based exploration of medical datasets. The work is based on the results achieved by Grimm et al. and produces interactive MPR for large datasets.

As our work is based on MPR, research on image comparison is interesting for us. Gesú and Starovoitov propose a distance function for comparing images [GS99]. The difference between images is computed as a single number. Yang et al. [YLC07] use the Hausdorff distance and the normalized gradient to evaluate the similarity between images. A survey of image comparison metrics is provided by Zhou et al. [ZCW02]. Image comparison research has mainly focused on producing a similarity metric for images without focusing on visualization techniques. We propose a method to compare and visualize the differences between multiple slices or images. We show our experimental results using ICT datasets but our techniques are general enough to be used in other image-comparison scenarios.

3 Comparative Visualization of Multiple Datasets

We introduce two novel visualization techniques, namely *multi-image view* and *edge explorer*. Furthermore, a variety of tools are proposed for user interaction. Quantitative data about slices and their comparison is displayed along with the rendered images.

The natural choice for comparing two slices or images will be to either display them side by side or to display them in a single checkerboard display. The checkerboard display has been previously used for multi-modal datasets. For example Stokking et al. [SZV03] display single photon emission computed tomography (SPECT) and magnetic resonance (MR) images in a checkerboard arrangement. The checkerboard is used to simulate the transparency effect and also to verify registration (figure 1).

We have an unknown number of datasets to compare and visualize. The checkerboard display handles just two datasets and is therefore unsuitable for our purposes. For the evaluation of a dataset series, a representation is needed which is able to handle multiple datasets. In the *multi-image view* we divide our viewport into circles

rather than squares. A circle can be easily divided into multiple, equally sized sectors with each sector representing a dataset. The circles however are not space filling. A good approximation to a circle, which also tiles the viewport is a hexagon. So we divide our screen into tightly packed hexagons, where the user can control the size and location of the hexagons.

Each hexagon has a circle in the middle and is divided into multiple sectors as shown in figure 2. Datasets from a series are displayed in each sector of the hexagon and in the central circle. The dataset which is displayed in the circle is called *central dataset*. All other datasets are compared and visualized with respect to the *central dataset*. Every hexagon provides a concurrent view into multiple datasets. We name our custom designed hexagon as *base tile*. The tiling of the viewport is called *multi-image view*.

3.1 Multi-image View

Multi-image view is divided into a grid of *base tiles*. The size of a *base tile* is controlled by the user. The first dataset in the series is used as the *central dataset*. The rest of the datasets are initially assigned sectors of the hexagon in a clockwise direction. The *central dataset* always displays the grey values. The user can choose the display mode for the datasets in the sectors of a *base tile*. The grey values or the difference values to the central dataset are the available options. Difference is evaluated by computing the difference between the grey values of the *central dataset* and the datasets in each sector. The result is color coded according to a user provided color scale. A honeycomb frame for visual distinction can be rendered over the result. The transparency and width of the honeycomb is user controlled.

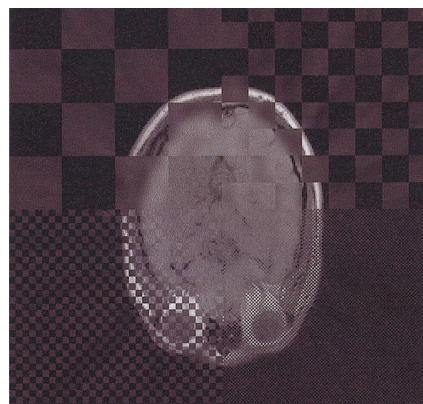


Figure 1: SPECT and MR slices displayed in a checkerboard pattern [SZV03].

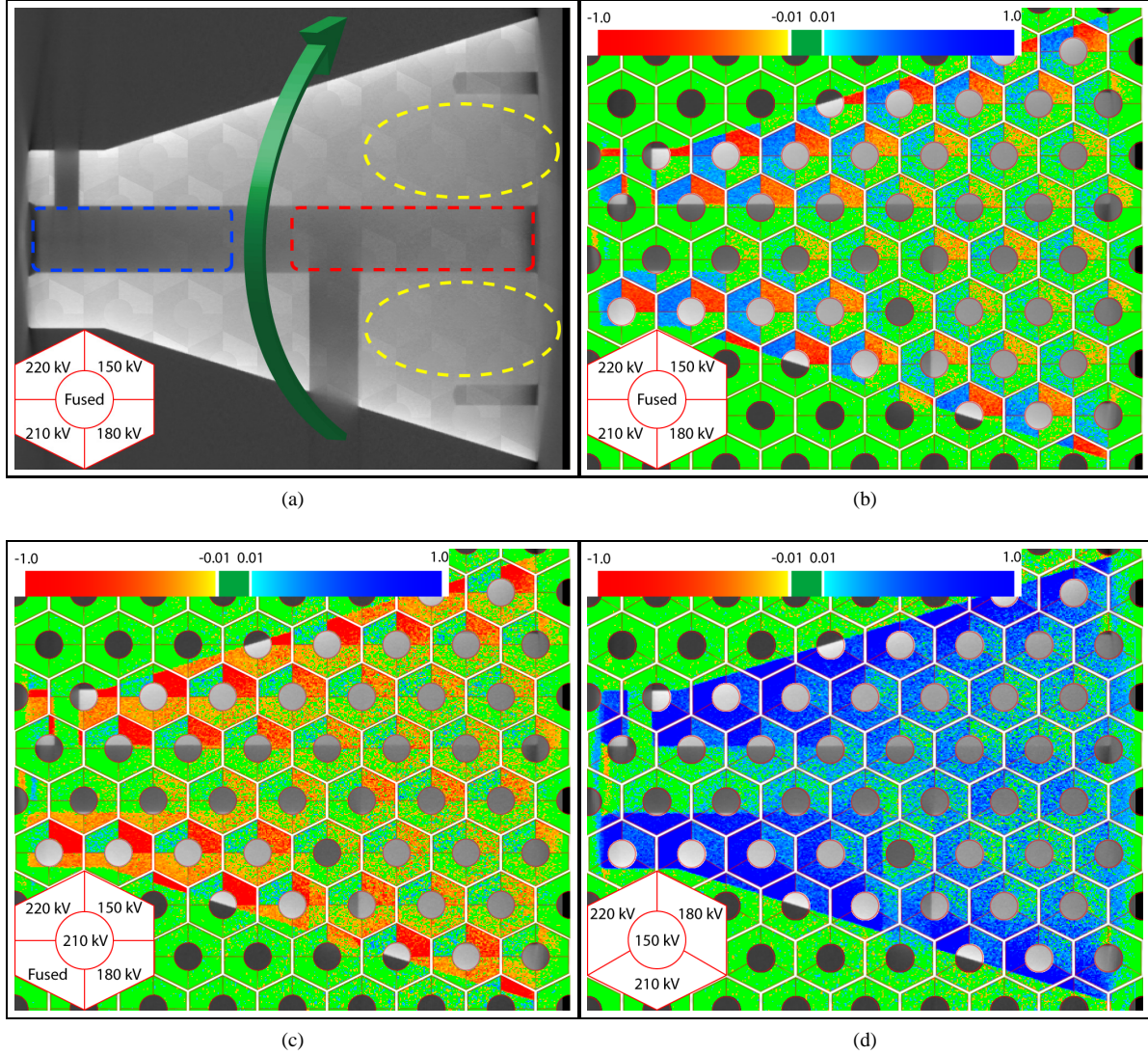


Figure 3: (a-b) multi-image view of the voltage data-series. Our system uses normalized grey values. The color scale is in the range of -1 to 1.(c) central dataset changed. (d) central dataset changed and fused dataset removed from the evaluation process.

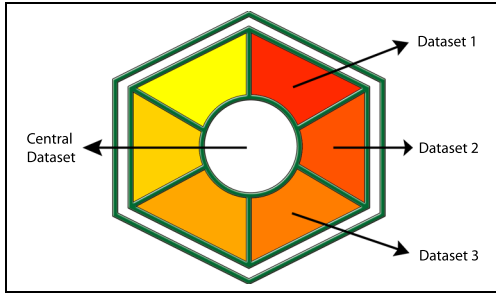


Figure 2: Base tile for multi-image view.

Several interaction tools are provided for effective comparison and live feedback. For example the placement of the datasets in a *base tile* is crucial for the visualization. The user can rearrange the sectors in which datasets are displayed by a simple drag and drop operation. The *central dataset* can also be interactively chosen and changed by the user. Any dataset, which is not interesting for analysis or has been shown as an outlier can be removed from the process. The number of sectors in all the *base tiles* will be reduced and rearranged by the system accordingly. The *base file* can be moved freely over the viewport with live update of the images.

We also use low-level fusion-techniques like a multi-scan fusion algorithm [HKMG08] to generate a fused dataset from a dataset series. The fused dataset can then be used as a *central dataset*. Figures 3(a-d) show a voltage dataset series. There are four datasets in the voltage series (150 kV, 180 kV, 210 kV and 220 kV), each recorded with 16 bits per voxel. We also generate a fused dataset from the given series. Figure 3(a) shows a rendering of the grey values in a *multi-image view*. The fused dataset is the *central dataset*. We see a clear pattern emerging in the left half of the image. The response of the X-rays decreases with an increase in voltage. The low-voltage dataset is better suited to capture edges in the test part.

The comparison of the slices is easier if we display difference data (figure 3(b)). For visual distinction the *base file* border is also rendered and the color scale is shown in the image. We observe that the difference between datasets is well pronounced in *base tiles* which are in the left half of the image and at the outer edges of the dataset. The dataset with 150 kV performs better (bright red sectors of the *base tiles*) while the result is almost the same for datasets measured using 210 kV and 220 kV (blue sectors). The fused dataset looks similar to the dataset recorded with 180 kV (green sectors). Our system reported the mean and the maximum difference between the fused and the 180 kV dataset as 238

and 1948. The grey values range from 0 to 65535 in the voltage series.

The difference between datasets diminishes in the right half of the image in figure 3(b). A low voltage X-ray source produces almost the same measurement as a high voltage source. We examine the *base tiles* in figure 3(b) which are in the area highlighted by yellow ovals in figure 3(a). The difference between the datasets has reduced considerably and we observe a lot of green color coding. This is due to beam hardening artifacts. Beam hardening occurs along high penetration lengths along the X-ray direction. The X-ray source was rotated around the specimen in the direction shown by a green curve in figure 3(a). So the X-rays were subjected to greater beam hardening in the right half of the shown images.

Beam hardening also modified the grey values of the air in the sections of the drill holes of our test part. The grey value of air is almost constant in all of the datasets in the region highlighted by a blue rectangle in figure 3(a). The red rectangle shows an area occupied by air but affected by beam hardening. The difference between metal and air is not recognizable anymore (figure 3(a) and (b)).

The datasets measured with 210 kV and 220 kV seem to produce almost the same result. We can compare these datasets by making one of these datasets the *central dataset*. We drag the 210 kV dataset into the *central dataset* location (figure 3(c)). Although the differences between these two datasets are very small, still the 220 kV dataset produces slightly lower grey values (sector with blue and green color). The mean and maximum difference between their grey values is 522 and 2891 respectively. The standard deviation of the difference is 1.09.

In figure 3(d) we choose the 150 kV dataset as the *central dataset* and remove the fused dataset from the evaluation process. The number of sectors in the *base tile* are changed from four to three. We can see areas where the 150 kV dataset has greater contrast between metal and air (blue region) and regions where there is not much difference (green).

The grid of *base tiles* can be interactively switched to a single *base tile*. This corresponds to a single multifaceted magic lens. The user can control the size and location of the *base tile* and thus can probe the specific regions of the dataset. This is particularly helpful for comparative visualization of areas severely affected by artifacts. We can determine the parameter setting which is most robust against the artifacts. Also the quantitative data is only displayed for the user selected area (*base tile*). Our system not only displays quantitative data of

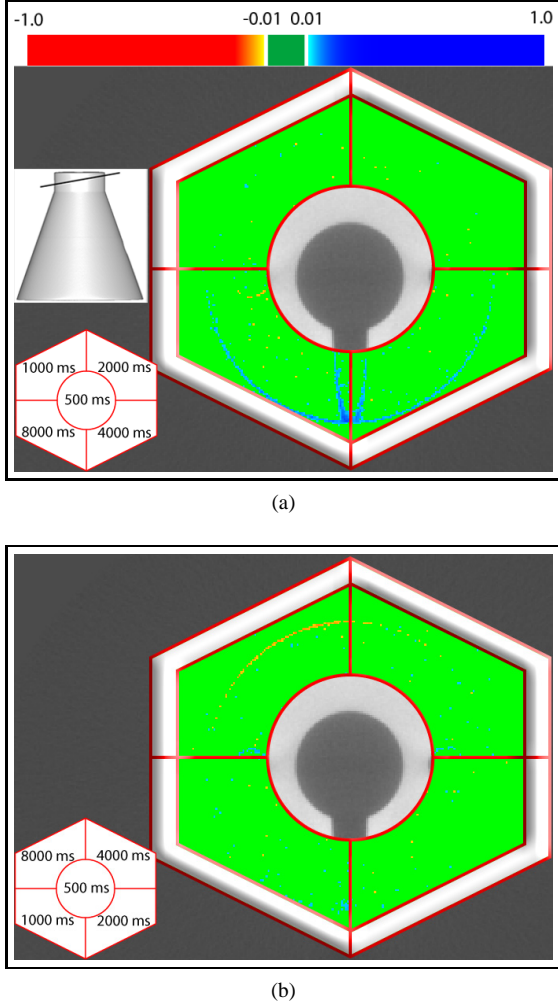


Figure 4: (a) slice position shown as a black line on the DVR of the test part (left) and multi-image view (right). (b) multi-image view with base tile sectors exchanged.

the current *base tile* but also of the last two user selected *base tiles*. Again, the display of history is useful for evaluation of artifacts.

Figure 4 shows the integration-time dataset series rendered with a single *base tile*. There are five datasets in the series with integration times of 500 ms, 1000 ms, 2000 ms, 4000 ms, and 8000 ms per projection image. The slice location is shown as a black line on the DVR of the test part in figure 4(a) left. We see grey value differences at the edges in the sectors containing the 4000 ms and 8000 ms datasets.

In figure 4(b) the sectors for the 1000 ms and 8000 ms datasets, and the 2000 ms and 4000 ms datasets are exchanged. The grey value differences at the edges are not present in the datasets with 1000 ms and 2000 ms

integration times. Only edges show a change in grey values with an increase of integration time, particularly between 2000 ms and 4000 ms integration time. There is no considerable visual difference when the integration time is increased from 500 ms to 2000 ms or from 4000 ms to 8000 ms.

Comparative visualization can be difficult to interpret when the *central dataset* has a high variation in grey values or has a lot of salt and pepper noise. This is apparent when evaluating the dataset series with a varying number of projections. If the number of projections is small, the volumetric dataset computed from these projections has streaking artifacts (figure 5(a)). The slice position is shown in the upper left corner of the image. Figure 5(b) shows a *multi-image view* where the dataset with 90 projections is set as the *central dataset*. We observe high variations and patterns. Blue and red colors build patterns in all the sectors of the *base tiles*. It is important to determine if it is because of salt and pepper noise in the datasets or due to a wrong selection of the *central dataset*.

We use a simple method to quickly determine the homogeneity of the slices and visualize it using a *multi-image view*. For homogeneity rendering we render a pixel based on the following conditions: If the grey value of a pixel is the same as in the 8-neighborhood, the pixel is rendered green. If a pixel and its neighbors in the 4-neighborhood have the same grey value, it is colored yellow. Otherwise the pixel is colored red. This provides a quick overview of the amount of deviation in the grey values in a slice. Homogeneity rendering (figure 5(c)) shows the *central dataset* (90 projections) to be most in-homogeneous. Therefore the dataset with 1440 projections is dragged to the central circle (figure 6). We conclude that the streaking artifacts are greatly reduced when the number of projections is increased from 90 to 270. A further increase in the number of projections does not have a big impact on the dataset quality.

3.2 Edge Explorer

Edge detection is an important subject in 3D ICT. Edge detection is useful for segmentation, feature detection, feature selection, etc. Edges are blurred in a 3D scan due to various artifacts like beam hardening, streaking, etc. In figure 3(b) the edge between metal and air is hardly perceivable in artifact affected regions. State-of-the-art edge detection algorithms fail to detect the edge either. We observed in figure 4(a) and (b) that various datasets of a series have different grey values at the edges.

Our company partners consider edges as a strong indication of scan quality. The *edge explorer* is designed to

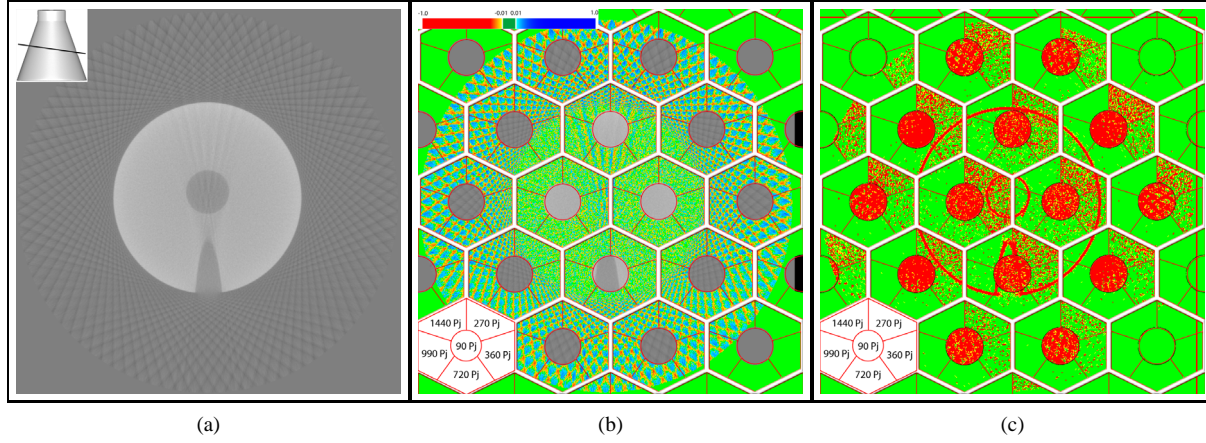


Figure 5: The no of projections dataset series has six datasets with 90, 270, 360, 720, 990 and 1440 projections for a single 360 degree turn of the rotary plate. (a) slice from the dataset with 90 projections. (b) multi-image view. (c) homogeneity rendering.

provide an overview about the edges in a set of slices or images. It also guides the user to problematic areas where edges are detected in one dataset but not in another one.

The *edge explorer* uses either the Sobel edge detection algorithm or the Canny edge detection algorithm [Can86]. The choice is based on user preference. The Sobel edge detector is faster than the Canny edge detector. It also requires the manipulation of a single threshold for edge detection whereas the Canny edge detector has two thresholds. Nevertheless the Canny edge detector provided better results in all our experiments. The result of an edge detection algorithm is a binary image.

Edge detection is performed on a slice of the *central dataset* and the result is displayed on the screen (figure 7(a)). Edge detection is also performed on the other datasets in a series and the result is stored for further processing. As we have edge information from all the datasets we can easily test if an edge is present or not in all the datasets at a particular position in an image.

We also have two graphs with blue background in figure 7(a). A vertical graph has yellow bars and is displayed on the right of the image. The horizontal graph is plotted above the image and has green bars. We scan the rows of the slices between the two vertical yellow lines on the image and count the number of locations where an edge is present in one of the datasets but is not present in any one of the rest of the datasets. The number of locations found in a row are shown on the vertical graph as a bar. Similarly we scan the columns between

the horizontal green lines and plot the result on the horizontal graph. Both graphs can be scaled separately.

The vertical and horizontal lines can be moved by the user with live update of the graphs. The combined information of the two graphs helps the user to find areas where edges are missing. In figure 7(a), two long bars on the vertical graph (highlighted by a dark blue oval) show an area where edges are missing in some of the datasets. The information from the horizontal graph (red oval) can be used to further localize the problematic area on the slice.

We also have a zoom-in tool in our *edge explorer*. The user selects the location on the image for zoom-in. The normal of an edge at that location (if an edge is present) or the gradient of the grey values is used to automatically select the dimensions of the zoom-in rectangle. If the gradient direction is close to the horizontal axis, the zoom-in rectangle has a bigger height (long rectangle). If the direction is close to the vertical axis, the rectangle has a bigger width (flat rectangle). Otherwise a square is chosen as a zoom-in region. More shapes can easily be added to the zoom-in tool based on gradient direction. Figure 7(b) shows a zoom-in using a long rectangle. The zoom-in shows each dataset of a series in a separate window. The edge strength in the zoom-in region is color coded according to the scale shown in the image. The edge in the 150 kV dataset is the most well pronounced.

4 Implementation and Results

We implemented a prototype on a Pentium 4 3.4 GHz CPU and an NVidia GeForce 8800 graphics board. We have very large datasets to visualize (table 1). In a pre-processing step bricks are created from the dataset series. Each element of a brick is a list containing a grey value from each dataset of the series. Brick fetching is fastest when the brick size is about 64 KBytes [GBKG04]. The pre-processing step dynamically chooses such brick dimensions that the brick size is close to 64 KBytes. We only fetch those bricks into memory that are required for rendering.

To speed up the overall performance of the application we also implement pre-fetching of the bricks. We have three threads in our application. The first thread runs in the background and loads those bricks into main memory that may be needed in the near future. Based on the orientation of the slice and the direction of navigation, we can calculate the next set of bricks probably needed. The second thread which has the highest priority is responsible for fetching those bricks that are needed for rendering the current *multi-image view*. The second thread will only be executed when the currently required set of bricks has not already been fetched into memory. The third thread performs rendering of images and handles user interaction with the system.

Based on the available memory in the system we keep four sets of bricks in memory. These sets of bricks in the order of importance are: first; the set required for rendering the current *multi-image view*, second; the next set of bricks required in the future, third; the set of bricks most recently used, and fourth; the set after the next set of bricks. The performance of our system is given in table 2. The time shown includes fetching the data from main storage and rendering. We observe approximately a 26% performance drop for the axial case because of bricking. We gained about 61% and 21% rendering speed-up by bricking in the sagittal and the coronal cases respectively. The loss in performance in the axial case is due to the alignment of the monolithic volume with the slicer.

Table 2: Performance: Average time (in milliseconds) required for rendering a grey value slice using a monolithic volume, bricked volume, rendering a multi-image view and rendering an edge explorer view.

	monolithic	bricked	multi-image view	edge explorer
Axial	19	24	26	29
Sagittal	90	35	38	44
Coronal	35	28	31	36

Our tool was evaluated by two domain specialists. The first professional is involved in the simulation of X-rays and the second works in post-processing of dual energy CT datasets. The second specialist is also a co-author of the paper. Both of them found our techniques very helpful for their work. The first professional found the *multi-image view* particularly helpful. He performed detailed low level analysis with our methods to evaluate his simulation results. They earlier used VGStudio MAX, Matlab, etc for comparison and it took them about three hours to analyze a set of ICT datasets. They predict the need of a more intensified comparison of multiple datasets in their future work. They plan to exploit the proposed techniques to test fusion algorithms, to evaluate ICT simulations, to visualize the influence of different X-ray energies in multi-energy ICT, etc.

We are not aware of any software that allows comparative visualization of multiple datasets. Multiple tools are needed for a single comparison. We used VGStudio MAX, Matlab and Adobe Photoshop to compare datasets and recorded the necessary time (table 3). Most contemporary MPR tools only allow exploration of a single 3D dataset at a time. We have to load each dataset of a series separately in VGStudio MAX and generate a slice image. Once we have all the slices from the dataset series, we compare them using Adobe Photoshop and by writing code in Matlab.

Table 3: Time required to setup an environment to compare a single set of slices.

	Time spent on ICT scan	Time spent on slice extraction and comparison	Our techniques
Voltage-series	120 min	60 min	interactive
Projections-series	240 min	72 min	interactive

5 Conclusion

We have presented a visualization system for comparative studies of multiple images. We use bricking for efficient memory management and threading for pre-fetching of the bricks. We take advantage of the graphics hardware to achieve maximum rendering speed up. We present *multi-image view* for comparative visualization of multiple datasets. It helps the user track artifacts and also provide interaction tools to find datasets that have better contrast or stronger edges, etc. The *edge explorer* is specifically designed for visualizing edges in multiple datasets. The system is evaluated by domain specialists for ICT dataset series. Our techniques are

generic and applicable for sets of images produced by various other imaging devices.

References

- [Can86] CANNY J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 6 (1986), 679–698.
- [ED82] ELLIOTT J. C., DOVER S. D.: X-ray microtomography. *Journal of Microscopy* 126 (1982), 211–213.
- [ERS*00] EBERT D. S., ROHRER R. M., SHAW C. D., PANDA P., KUKLA J. M., ROBERTS D. A.: Procedural shape generation for multi-dimensional data visualization. *Computers & Graphics* 24, 3 (2000), 375–384.
- [FMA05] FOUT N., MA K., AHRENS J.: Time-varying, multivariate volume data reduction. In *Proceedings of the 2005 ACM symposium on Applied computing* (2005), pp. 1224–1230.
- [GBKG04] GRIMM S., BRUCKNER S., KANITSAR A., GRÖLLER M. E.: A refined data addressing and processing scheme to accelerate volume raycasting. *Computers & Graphics* 28, 5 (Oct. 2004), 719–729.
- [GS99] GESÚ V. D., STAROVOITOV V.: Distance-based functions for image comparison. *Pattern Recognition Letters* 20, 2 (1999), 207–214.
- [HKMG08] HEINZL C., KASTNER J., MÖLLER T., GRÖLLER M. E.: Statistical analysis of multi-material components using dual energy ct. In *VMV 2008, Vision, Modeling and Visualization* (2008), pp. 179–188.
- [KBKG07] KOHLMANN P., BRUCKNER S., KANITSAR A., GRÖLLER M. E.: Evaluation of a bricked volume layout for a medical workstation based on java. *Journal of WSCG* 15, 1-3 (Jan. 2007), 83–90.
- [SCM99] SHEN H., CHIANG L., MA K.: A fast volume rendering algorithm for time-varying fields using a time-space partitioning (tsp) tree. In *Proceedings of the IEEE Visualization '99* (1999), pp. 371–377.
- [SJH08] SISNEROS R., JOHNSON C. R., HUANG J.: Concurrent viewing of multiple attribute-specific subspaces. *Computer Graphics Forum* 27, 3 (2008).
- [SZV03] STOKING R., ZUBAL I. G., VIERGEVER M. A.: Display of fused images: Methods, interpretation, and diagnostic improvements. *Seminars in Nuclear Medicine* 33, 3 (2003), 219–227.
- [Tay02] TAYLOR R.: Visualizing multiple fields on the same surface. *IEEE Computer Graphics and Applications* 22, 3 (2002), 6–10.
- [VG04] VOLUME-GRAPHICS: *VGStudio Max 1.2-User's Manual*. 2004. www.volumegraphics.com.
- [WBO97] WILSON D. L., BADDELEY A. J., OWENS R. A.: A new metric for grey-scale image comparison. *International Journal of Computer Vision* 24, 1 (1997), 5–7.
- [YLC07] YANG C. T., LAI S., CHANG L.: Hybrid image matching combining Hausdorff distance with normalized gradient matching. *Pattern Recognition* 40, 4 (2007), 1173–1181.
- [ZCW02] ZHOU H., CHEN M., WEBSTER M. F.: Comparative evaluation of visualization and experimental results using image comparison metrics. In *Proceedings of the IEEE Visualization 2002* (2002), pp. 315–322.

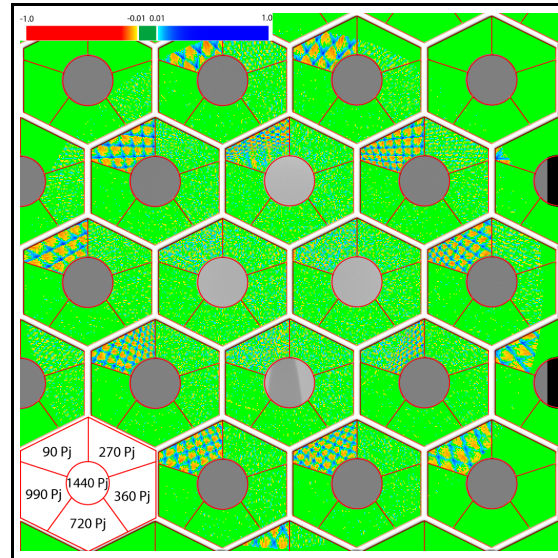
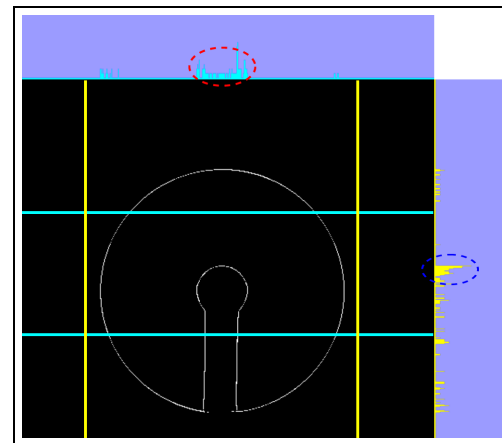
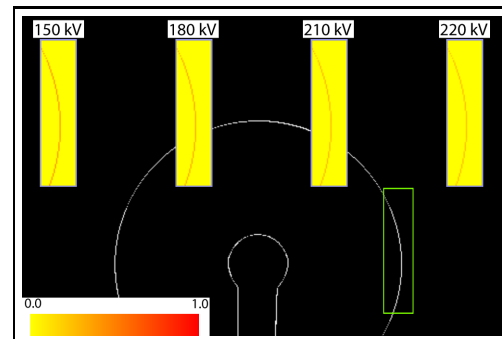


Figure 6: Multi-image view for number of projections series.



(a)



(b)

Figure 7: (a) edge explorer. (b) zoom-in tool.