UC Davis
IDAV Publications

Title
Visualizing Industrial CT Volume Data for Nondestructive Testing Applications

Permalink
https://escholarship.org/uc/item/2s08d924

Authors
Huang, Runzhen
Ma, Kwan-Liu
McCormick, Patrick
et al.

Publication Date
2003

Peer reviewed

Visualizing Industrial CT Volume Data for Nondestructive Testing Applications

Runzhen Huang   Kwan-Liu Ma
University of California at Davis
{huangru,ma}@cs.ucdavis.edu

Patrick McCormick   William Ward
Los Alamos National Laboratory
{pat,ww}@lanl.gov

Figure 1: Volume rendering of a CT of a mechanical toy ($512 \times 512 \times 2048$ voxels).

Abstract

This paper describes a set of techniques developed for the visualization of high-resolution volume data generated from industrial computed tomography for nondestructive testing (NDT) applications. Because the data are typically noisy and contain fine features, direct volume rendering methods do not always give us satisfactory results. We have coupled region growing techniques and a 2D histogram interface to facilitate volumetric feature extraction. The new interface allows the user to conveniently identify, separate or composite, and compare features in the data. To lower the cost of segmentation, we show how partial region growing results can suggest a reasonably good classification function for the rendering of the whole volume. The NDT applications that we work on demand visualization tasks including not only feature extraction and visual inspection, but also modeling and measurement of concealed structures in volumetric objects. An efficient filtering and modeling process for generating surface representation of extracted features is also introduced. Four CT data sets for preliminary NDT are used to demonstrate the effectiveness of the new visualization strategy that we have developed.

CR Categories:
    I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary representations' I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques; I.3.8 [Computer Graphics]: Applications—; I.4.6 [Image Processing and Computer Vision]: Segmentation—Region growing

Keywords: Computed tomography, feature extraction, hardware-acceleration rendering, image processing, interactive visualization, nondestructive testing and evaluation, scientific visualization, surface modeling, user interface, volume rendering

1 Introduction

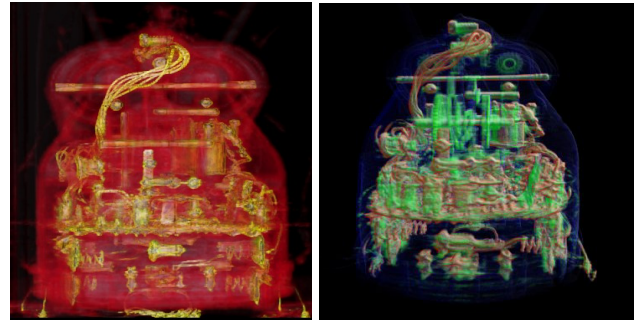Computed Tomography (CT) is a noninvasive imaging technique that has been used extensively in not only medicine for diagnosis and surgical planning, but also in nondestructive testing (NDT) for many industrial applications such as mechanical part manufacturing, production of composite materials, waste container inspection and even hardwood lumber processing. The current industrial CT technology is capable of generating very high-resolution images. According to the application requirements, the resolution of the data can be in the range of 512^3–2048^3 voxels. Such sizes present great challenges to the visualization process which involves both batch processing and interactive visual inspection. Figure 1 displays volume rendering of such a CT of a mechanical toy. Due to its size, interactive viewing was achieved with a hybrid rendering approach [Wilson et al. 2002].

This application paper focuses primarily on the problems of feature extraction and modeling for NDT applications. Both an interactive method and a semi-automatic method are presented. The interactive approach is based on the use of 2D transfer functions and a 2D histogram interface [Kindlmann and Durkin 1998; Kniss et al. 2002], hardware-accelerated volume rendering, and a user interface mechanism that supports incremental and comparative visualization. The semi-automatic approach is based on a region growing algorithm that captures features with complex geometry or small, scattered features in noisy data. A particular interesting and unique approach is the derivation of 2D transfer functions based on the partial features extracted using region growing to visualize the full, similar features in the volume data. This is particularly useful when several features of the same type are disconnected in the volume, and thus region growing cannot capture all of them at once. We also show how to produce a boundary surface representation for the volumetric feature extracted with region growing, and how the quality of such surfaces can be improved with an efficient filtering algorithm. A prototype system has been built which incorporates all these techniques with a coherent user interface. Our experimental study on this system

using four CT volume data sets from NDT shows that all relevant features can be quickly identified and displayed.

## 2 Driving Application: Nondestructive Testing and Evaluation

Nondestructive testing (NDT) is an industrial technology commonly used to detect surface and subsurface discontinuities in material without damaging or destroying the object being tested. For example, for aircraft inspection, NDT is used to aid in the prevention of premature or inadvertent failures in various types of structures and materials. In construction engineering, NDT plays a major role by providing a means of checking compliance to applicable codes, specifications, and standards without significantly interfering with physical construction of the project. Other important NDT applications include CAD part design and quality control, explosive ordinance examination, and nuclear waste inspection.

A variety of NDT techniques are available, which include ultrasonic, radiography, dye penetrant, eddy current, and magnetic particle methods. In our work, we are mainly concerned with the problem of identifying and visualizing features of interest in volume data generated from an industrial computed tomography system. One challenge that we face is the low-contrast and noisy image characteristics of CT data. The other is the size of a typical data set. Because radiation is not an issue when scanning non-organic objects, a very large number of high-resolution images can be made available for each study.

In our preliminary study, we have applied our feature extraction and modeling techniques to four CT volume data sets obtained from the Nondestructive Testing Group at the Los Alamos National Laboratory (LANL). Figure 2 displays two images for each data set. The first two data sets are rather noisy. The first one is a volume reconstruction of a block of concrete with a long glass rod off the center. The features that the scientists are after include the long glass rod and some of the surrounding small blobs. The multiple thin rings are artifacts due to volume reconstruction, which we would want to remove from the visualization. The second data set is a volume reconstruction of a small plastic part surrounded by an aluminum sleeve. The plastic was thermally shocked to produce small cracks which are the target of the feature extraction and visualization. In the images in Figure 2, the cracks are barely revealed. The third one is a Maglite flashlight. With direct volume rendering using 1D or 2D transfer functions, we can effectively depict its shape and overall internal structure, as shown on the left image. To capture a specific part, as shown on the right image, segmentation methods are often required. The last data set is a volume reconstruction of a container with a circuit board and some batteries inside. Our objective is to extract selected features in each data set and generate enhanced visualizations of the features.

## 3 Feature Extraction and Visualization

The features which the scientists are interested in are generally related to some local density variation and can be either clustered or scattered. The detection of such features is further complicated by the possibly very complex structure of the volume, the low-contrast of the features, and the noise inherently present in the data.
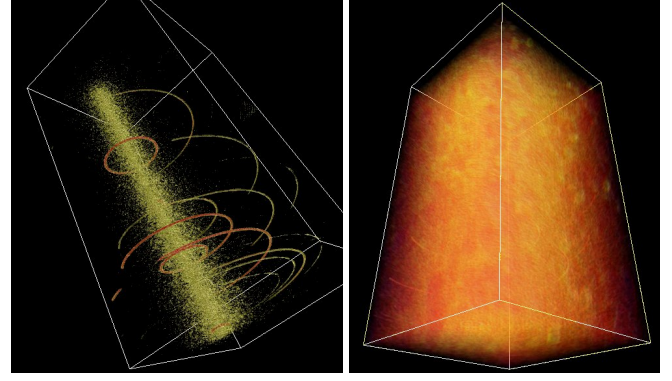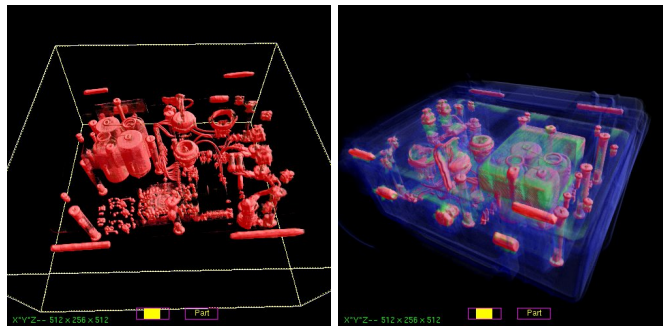

Concrete: 256×256×512 voxels.
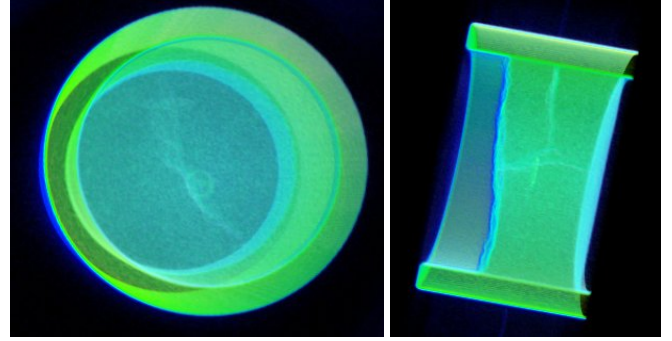
Plastic: 697×697×260 voxels.
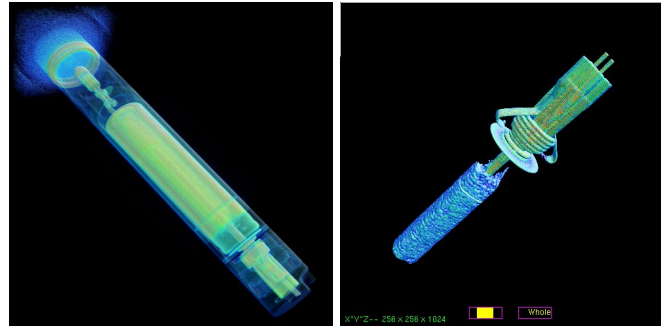
Flashlight: 256×256×1024 voxels.

Fire Set: 512×256×512 voxels.

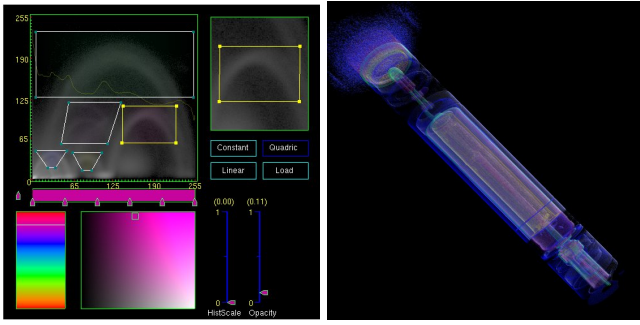Figure 2: Four test data sets obtained from the Nondestructive Testing Group at the Los Alamos National Laboratory.

Figure 3: Left: The interface for editing a two-dimensional transfer function. Right: The resulting image of the CT flashlight volume.
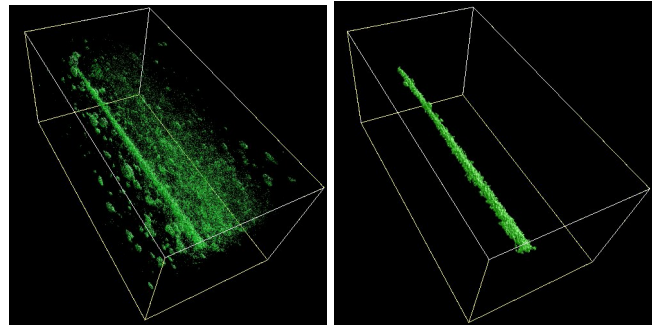

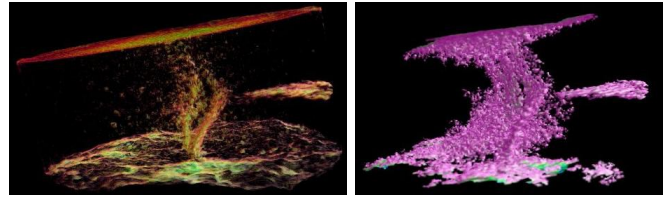Figure 4: Feature extraction results. Left: transfer function. Right: region growing.

Figure 5: Left: Direct visualization of the CT plastic data using a two-dimensional transfer function that was interactively defined. The cracks revealed in this image are rather incomplete. Right: Volume rendering of the cracks extracted using region growing. The images show the cracks actually extend from the top to the bottom of the data set.

3.1 Interactive Exploration

The advent of hardware-accelerated volume rendering enables interactive feature finding and visualization. Hardware accelerated volume rendering requires the loading of the volume data into the texture memory of the video card prior to rendering. The size of the volume that can be rendered interactively is thus limited by the amount of video memory present on the card and the speed at which the data can be transferred between system memory and the graphics card.

If interactive rendering is possible, data exploration is generally achieved by editing color and opacity transfer functions [Pfister et al. 2001]. Kindlmann and Durkin [Kindlmann and Durkin 1998] introduced an intuitive way to derive opacity transfer functions based on the fact that the derivatives of data values suggest material boundaries. They show that by looking at a two-dimensional scatterplot of data values and gradient magnitudes (i.e., a 2D histogram), opacity transfer functions can be easily defined to effectively capture features composed of boundaries between materials of relatively constant data value. Kniss, et al. [Kniss et al. 2002] extended their work by introducing a set of direct manipulation widgets as the interface for defining multidimensional transfer functions for volume visualization. The concept of dual-domain interaction is particularly powerful and demonstrated with several effective visualization examples.

Our system adopts the 2D transfer function interface introduced in [Kindlmann and Durkin 1998; Kniss et al. 2002] to assist interactive feature extraction. The user is presented with an interface displaying two-dimensional scatterplots of the volume data. The left image of Figure 3 shows such an interface in which the upper left is a scatterplot of data values (x-axis) and gradient magnitudes (y-axis). The user can explore the data by interacting with the scatterplot which is superimposed with the current color map and the histogram of the data to provide additional information to the user. The trapezoid boxes, which are more general than triangular and rectangular boxes used in [Kniss et al. 2002], on the scatterplot define the classification functions used to generate the CT flashlight image shown in the right image of Figure 3.

Most of the time a feature in the data can be revealed by interactively varying the two-dimensional transfer function but often other features (or noise) in the data are also brought out, as shown in the left image of Figure 4. While in this case our goal is to extract only the long glass rod in the CT concrete data, some noises surrounding the structure also appear, which is undesirable. When applying the region growing method to be discussed next, only the long rod is

captured, as shown in the right image of Figure 4.

Furthermore, interactive exploration involves an iterative trial-and-error process which can be very time consuming. This process is made even more difficult as the size of the data increases; thus making the manual search for several features an impossible or unpleasant task. In order to study large volumetric data sets, automatic feature extraction techniques are desirable since it is very likely that important features are less likely to be missed. For example, the plastic data set is difficult to visualize directly. Even after some careful editing of the two-dimensional transfer function, we can barely capture the crack in the middle, as shown in the left image of Figure 5. With region growing, a more complete crack structure can be captured. The right image in Figure 5 shows the region growing result in which we can see the cracks actually extend from the top to the bottom of the data set. In this particular case, the region growing calculations take under 3 seconds on a PC with a 1.8GHz Pentium 4 processor.

3.2 Region Growing Methods

In our study, we couple interactive browsing with local region growing methods [Ballard and Brown 1982]. Region growing is a fundamental segmentation technique in image processing [Rosenfeld and Kak 1982]. In the context of volume data segmentation, the goal of region growing is to map the input volume data into sets of connected voxels, called regions, according to a prescribed criterion which generally examines the properties of local groups of voxels [Lohmann 1998]. The growing starts from a voxel in the proximity of the the seed point selected by the user. The voxel can be chosen based on either its distance from the seed point or
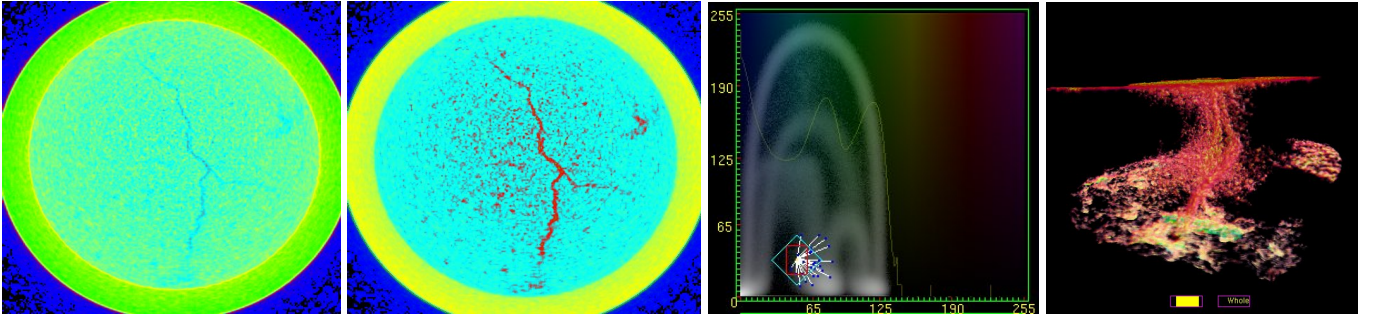
Figure 6: From left to right: A slice of the plastic volume data, the slice was enhanced to facilitate seed picking; additional information displayed on the histogram to assist seed selection; volume rendering of the corresponding region growing result.

the statistical properties of the neighborhood. Then each of the twenty-six immediate neighbors of that voxel are visited to determine if they belong to the region. This growing expands further by visiting the neighbors of each of these twenty-six voxels. This recursive process continues until either some termination criterion is met or all voxels in the volume are examined. The result is a set of connected voxels determined to be located within the region(s) of interest.

Feature finding thus becomes semiautomatic starting with an interactive seed point selection step, followed by region growing process. As a result, the user only needs to find a few representative features and lets the region growing locate all features of similar properties in the same volume data. This approach is similar to the volume seedling method introduced by Cohen et al. [Cohen et al. 1992]. In their work, region growing is used to help identify fine blood vessels in MRA volume data. More recently, Hahn, et al. [Hahn et al. 2001] develop a pipeline of 3D image processing steps to derive accurate models for visualization and exploration of vascular structures from radiological data. The resulting vessel models are used to study the branching patterns, for measurement, etc. Hu, et al. [Hu et al. 2001] apply region growing to the segmentation of lung data to derive airway tree for surgical planning and treatment.

3.2.1 Seed point selection

Region growing begins from a seed point. In our system the user picks a seed by interactively slicing through the volume data. As soon as a slice is picked the user can move the cursor into the desired region and click to complete the selection. For noisy data sets, seed point selection becomes difficult, especially when regions are small or thin. First, according to the histogram of the data values, our system can automatically map more colors to a particular value range of interest to increase the visibility of the features, as shown in Figure 6. Such enhancement can also be done by the user in an interactive manner while looking at the two-dimensional scatterplot of the data.

To further assist interactive seed point selection, some information about the seed voxel and its surrounding voxels is presented to the user. As shown in the third image from the left in Figure 6, when a point is selected, it is highlighted on the scatterplot together with its 26 neighbors. This resembles to the dual-domain interface but more information is provided to the viewer. The coordinates of each highlighted point are the corresponding voxel's scalar and gradient magnitude values. A solid line connects the current point to each of its 26 neighbors. Another very important piece of information is the red rectangle which shows the standard

deviation of the data values (x-axis) and the standard deviation of the gradient magnitudes (y-axis) of the neighbors. Essentially, a tighter rectangular box suggests a region of high homogeneity. The right most image in Figure 6 shows the corresponding region growing result.

3.2.2 Criteria selection

To do region growing, a set of criteria must be appropriately selected to effectively extract the regions. Possible criteria include region homogeneity and contrast with the background, strength of the region boundary, size, conformity to a desired texture or shape, and so on [Rosenfeld and Kak 1982]. We have derived three criteria mainly based on region homogeneity and region aggregation using either data values or gradient magnitudes of the voxels. To make our region growing method more robust, at each voxel we also take into account some first-order statistical information about its 26 neighboring voxels.

Each criterion is defined by a *cost function* which is designed to extract a particular type of feature. If the value returned by the cost function is less than 1, then the voxel under consideration is within the region. Three functions are defined:

A. $f_{ca} = \frac{|v - v_s|}{k\sigma_v}$ where v is the data value of the current voxel, v_s is the data value of the seed voxel, k is a constant specified by the user, and σ_v is the standard deviation of the values of the 26 neighboring voxels of the seed voxel. Note that $0 < k$. It is used to control the strictness of the criterion, and its default value is 1. This cost function exploits the data values. It works well for capturing homogeneous regions in which gradient magnitudes are nearly zero.

B. $f_{cb} = \frac{|g - g_s|}{k\sigma_g}$ where g is the gradient magnitude value of the current voxel, g_s is the gradient value of the seed voxel, and σ_g is the standard deviation of the gradient values of the 26 neighboring voxels of the seed voxel. This cost function exploits gradient magnitudes and is used to capture only the boundaries of a region.

C. $f_{cc}(p) = f_{ca}p + f_{cb}(1-p)$ where p is a weight specified by the user or by the system. By default, $p = \frac{\sigma_g}{(\sigma_v + \sigma_g)}$. As a result, this cost function exploits both the data values and gradient magnitudes. It is more flexible than f_{ca} and f_{cb}. The left image of Figure 4 was generated using f_{cc}.

Figure 7 shows the results of applying each cost function to the segmentation of the flashlight data set. In this case, among the three functions, f_{cc} gives us the desired result.

Modayur et al. [Modayur et al. 1997] design an adaptive cost function which takes into account the statistical information of the region extracted so far to handle the situation where small connections between two regions should be discarded. Basically, an adjacency criterion is employed in a postprocessing step. However, their cost function does not consider the gradients of voxels which are useful for growing boundaries.

3.2.3 Postprocessing

After region growing, a postprocessing step might be desirable to extend the results or to improve the results such as removing erroneous voxels or to fill gaps introduced by noisy data. Depending on which criteria is used, the results of feature extraction may vary. For example, due to noise it is possible for a feature to grow into another through a very thin connection one- or two-voxels wide. A postprocessing step can be performed to remove this type of erroneous connection. The three images in Figure 8 show the result of applying f_{ca}, the result after the cleanup, and the result of applying f_{cc}. In this case, f_{cc} works better because of the use of neighboring gradient values to verify connectivity. However, f_{cb} alone performs poorly in this case.

Since it is easier for the user to extract one feature at a time, our system allows the user to compose multiple extracted features into a single visualization, as depicted in Figure 9. The composition is done interactively through the user interface providing the user with multiple views. In this way, the user can extract and enhance each of the features completely independent of the other features in the same data set but present them in a single visualization for other specific purpose. The multiple views also allow the user to perform comparative visualization of similar features in the same volume or two different volumes.

3.2.4 Partial region growing and transfer functions

When the cost of region growing is high or there are multiple regions, it might be desirable to perform partial region growing instead. That is, feature extraction is only done for a subset of the volume, or only a small subset of all the interested regions in the data is extracted. It is then possible to derive a two-dimensional transfer function based on the partial results of feature extraction for the visualization of other regions/features in the data.

Such a transfer function can be derived by using the average data value and average gradient magnitude value in the regions already extracted, and standard deviations of the data values and gradient values for the whole extracted region. These four values define a rectangular region on the two-dimensional scatterplot of the data values and gradient values, which is the classification function we are looking for to capture the extended features. The two average values are the coordinates of the center of rectangular region, and the two σ values define the size of the region. The left most image in Figure 10 displays a single metal screw extracted using partial region growing. The second image shows the result of applying the derived transfer function to the whole volume to reveal other screws inside the container. The third image shows the derived transfer function (in red), and two other transfer functions (in green and blue) manually defined for adding context to the visualization, as shown in the right most image.
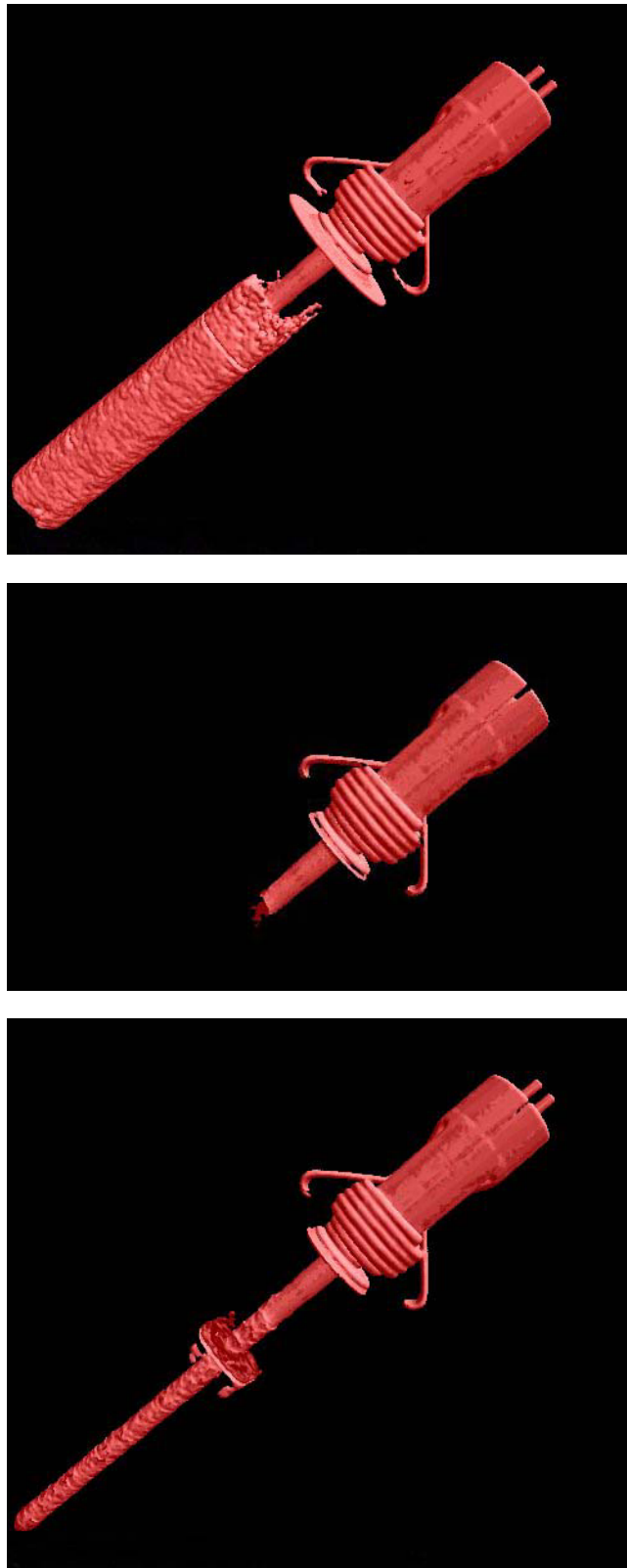

Figure 7: Segmenting the flashlight data set. Top: Using f_{ca} alone extracts unwanted parts. Middle: Using f_{cb} extracts an incomplete part. Bottom: Using f_{cc}. f_{cc} gives us the desired result.
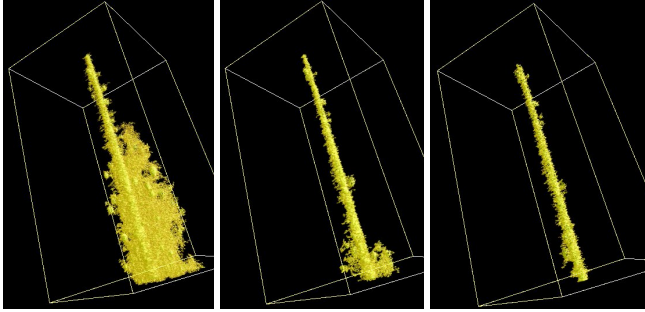
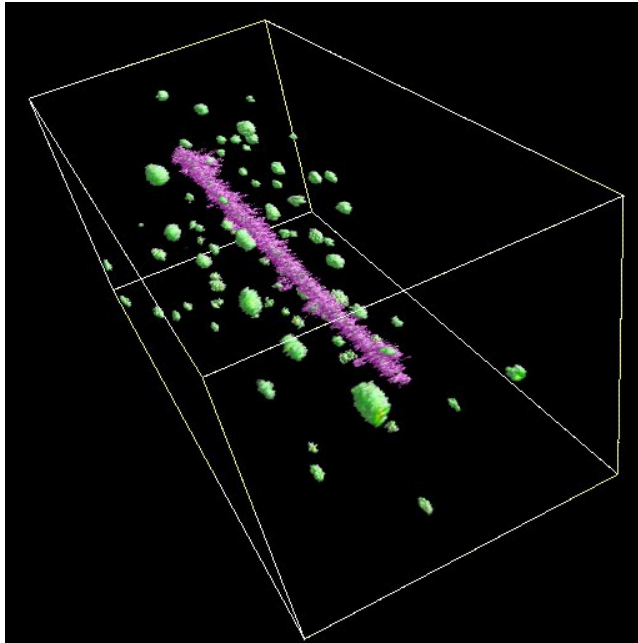Figure 8: Left: the result of using f_{ca}. Middle: after cleaning up. Right: the result of using f_{cc}.

Figure 9: Composition of two features.Each feature is independently extracted through an interactive user interface providing multiple views of features. This visualization allows the user to see both the glass rod and the surrounding blobs but not the noises.

3.3 Feature Modeling

Although volume rendering can generate images of the features of interest without first extracting any geometric information from the data, a geometric representation of the features allows for evaluation beyond visual inspection. The commonly used isosurface extraction method [Lorensen and Cline 1987] based on a binary decision would not work well here because the CT data are noisy and may contain small, fine features.

To construct boundary surfaces for the features, we make use of the segmented volume from region growing. A boundary tracking algorithm [Artzy et al. 1981] is used to identify boundary faces. This is followed by a procedure to locate the voxels internal to the boundary identified. The resulting volumetric feature representation allows us to construct compact and hole-free boundary surfaces. The next step is to filter out high-frequency artifacts in the boundary cells. At this point, we could rendering the boundary faces directly by interpolating the faces' normals. However, this would only give us a better picture of the boundary surfaces but not the actual smoother surfaces. The more desirable boundary surfaces can be extracted by using the marching cubes method [Lorensen and Cline 1987]. A reasonable isovalue used is the average values of those voxel on the boundary faces. The resulting surfaces are smooth while preserving the main topological information of the features, as shown in Figure 11 for the concrete data set. Figure 12 displays the extracted surface of the crack for the plastic data set.

Because of the size of the volume data, the filtering operation can become time-consuming. In this work a Box or Guassian filer is used according to the characteristics of features. To speed up the filtering operation (i.e., convolution), we have applied a set of high-performance computing techniques. Specifically, we have extended a blocked algorithm [Lam et al. 1991] for our 3D filtering problem to improve the effectiveness of memory hierarchies which include virtual memory, caches, vector registers and scalar registers, and the main memory. The basic idea is to explore the localities of both program and data. The best block size to use thus depends on the volume data size and the memory architecture.

We have also followed the PHiPAC coding guidelines [Bilmes et al. 1998] to achieve further performance improvement. Among the set of coding guidelines, we found that loop unrolling and register caching result in the biggest performance improvement. Our experimental results show that an overall 5-8 times speedup may be achieved depending on the size of the filter kernel used. Figure 13 displays test results using a $3\times3\times3$ filter kernel for the concrete data set with the block size $= 64$. The performance difference is apparent. Adding PHiPAC-based optimization is about three times faster than using blocking alone. The same PC with a 1.8GHz Pentium 4 processor was used for testing and the filtering time was cut from about 25 seconds to 5 seconds. PHiPAC coding changes the behavior of the program. According to our test results, PHiPAC coding allows us to use large block sizes leading to better performance; that is, PHiPAC coding improves the locality of the program.

4   Conclusions

In this paper we have presented our preliminary results on visualizing industrial CT volume data. A prototype system has been created which allows us to experimentally study both interactive and semi-automatic techniques. Two-
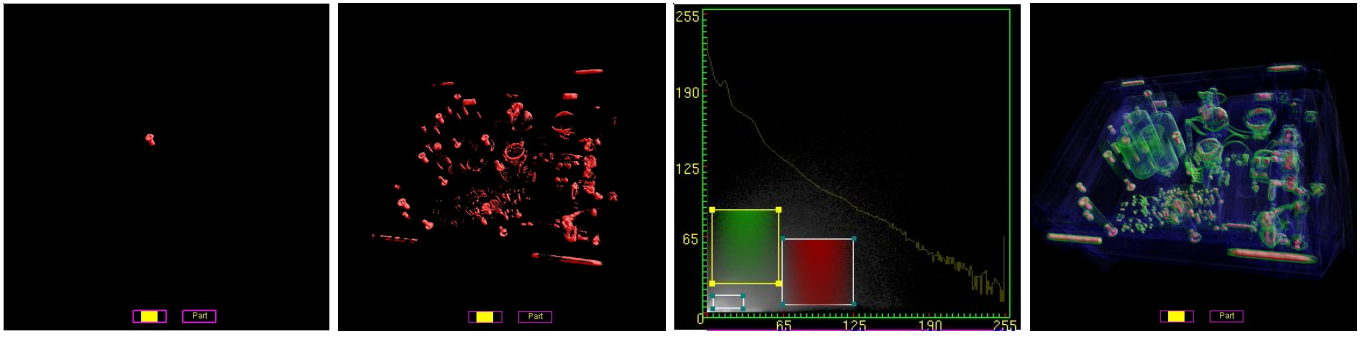
Figure 10: Left: The result of a partial region growing for capturing a single screw. Middel-left: Using the derived transfer function to capture other screws. Middle-right: The derived transfer function (in red) plus two manually defined transfer functions for adding context to the visualization. Right: Visualization using the three transfer functions.
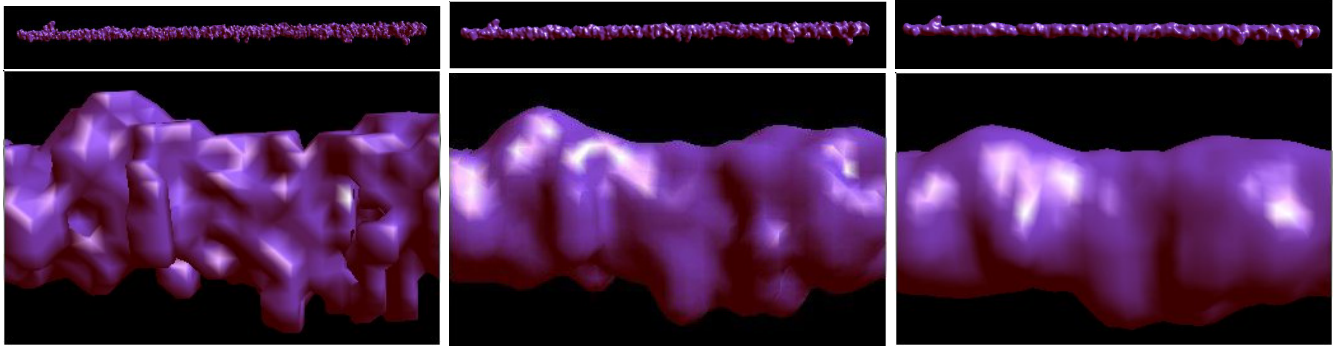

Figure 11: Left: Original boundary surface. Middle: After applying 3×3×3 filter. Right: After applying 5×5×5 filter.
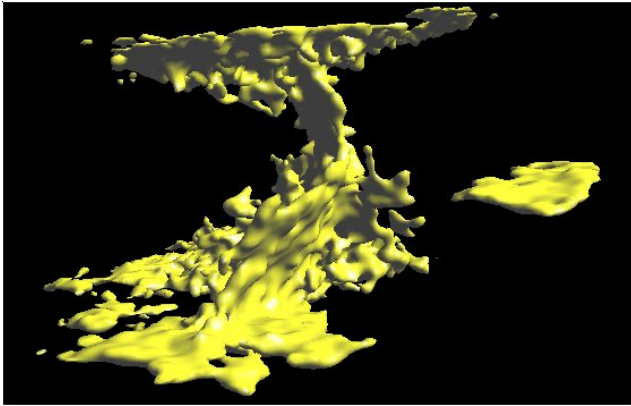

Figure 12: A surface representation of the extracted crack features in the plastic volume data.
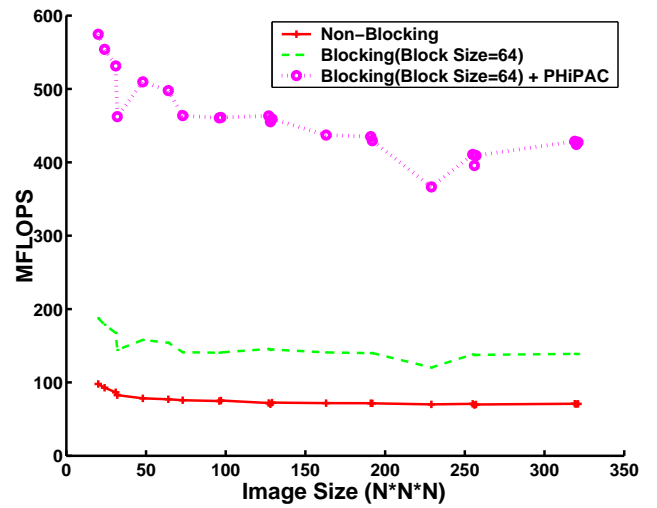

Figure 13: Timing with and without optimization for the filtering operation.

dimensional transfer functions are powerful for visualizing material boundaries but often fail to completely capture complex volumetric features in noisy data. The region growing methods we have developed supplement the transfer function methods. During our experimentation using several representative CT data sets, we have found that the seed selection interface and the cost functions we have developed are very helpful and effective in accomplishing the intended tasks.

We have also applied a refined region growing based approach to medical volume data, and obtained very promising results [Huang and Ma 2003]. We should point out that while these test data sets do present challenges to our study, the data sets we expect to work on for the actual nondestructive evaluation tasks will be larger and could contain multiple, more complex features.

A related project of ours has shown that by employing an intelligent system it is possible to derive a more intuitive user interface and also eliminate the typical trial-and-error process of visualization parameter specification required in complex classification and visualization tasks [Tzeng et al. 2003]. We are currently incorporating an intelligent system into our NDT data visualization system.

We will also continue our study in several other directions. We plan to investigate geometrically-sensitive segmentation such that even more robust criteria can be derived for region growing. Adaptive region growing such as incorporating the statistical information about the regions already identified into cost functions is also a promising approach. Morphological operations can help to further improve the feature extraction results and should be experimentally studied. We are also interested in deriving topological information about the extracted features for region merging and splitting and to create more compact models. We have the ability to construct and display boundary surfaces for the extracted volumetric features. As suggested by the application scientists, it is helpful to render these surfaces by coloring them according to selected statistical information about the features.

The large data problem must also be addressed. Presently, realtime rendering and manipulation of the volume data is limited to only a subset of the volume since the entire volume will not fit into the texture memory of a PC graphics card. We are in the process of porting the current system to a PC cluster for not only interactive rendering of large CT volume data, but also the feature extraction and modeling tasks.

Acknowledgments

References

ARTZY, E., FRIEDER, G., AND HERMAN, G. T. 1981. The theory design, implementation and evaluation of a three-dimensional surface detection algorithm. *Computer Graphics and Image Processing 15*, 1–24.

BALLARD, D., AND BROWN, C. 1982. *Computer Vision*. Prentice-Hall Inc.

BILMES, J., ASANOVIC, K., DEMMEL, J., LAM, D., AND CHIN, C. 1998. The PHiPAC v1.0 matrix-multiply distribution. Tech. Rep. UCB/CSD-98-1020, University of California at Berkeley. (www.icsi.berkeley.edu/~bilmes/phipac).

COHEN, M., PAINTER, J., MEHTA, M., AND MA, K.-L. 1992. Volume seedlings. In *Proceedings of the 1992 ACM Symposium on Interactive 3D Graphics*, 139–145.

HAHN, H. K., PREIM, B., SELLE, D., AND PEITGEN, H.-O. 2001. Visualization and interaction techniques for the exploration of vascular structures. In *Proceedings of the IEEE Visualization 2001 Conference*, 395–402.

HU, S., HOFFMAN, E. A., AND REINHARDT, J. A. 2001. Automatic lung segmentation for accurate quantitation of volumetric x-ray ct images. *IEEE Transactions on Medical Imaging 20*, 6 (June), 490–498.

HUANG, R., AND MA, K.-L. 2003. Rgvis: Region growing based techniques for volume visualization. In *Proceedings of the Pacific Graphics 2003 Conference (to appear)*.

KINDLMANN, G., AND DURKIN, J. 1998. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of 1998 Symposium on Volume Visualization*, 79–86.

KNISS, J., KINDLMANN, G., AND HANSEN, C. 2002. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics 8*, 3, 270–285.

LAM, M., ROTHBERG, E., AND WOLF, M. 1991. The cache performance and optimizations of blocked algorithms. In *Proceedings of the 4th International Conference on Architectural Support for Programming Languages and Operating Systems*.

LOHMANN, G. 1998. *Volumetric Image Analysis*. Wiley & Teubner Press.

LORENSEN, W. E., AND CLINE, H. E. 1987. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of SIGGRAPH '87*, 163–169.

MODAYUR, B., PROTHERO, J., OJEMANN, G., MARAVILLA, K., AND BRINKLEY, J. 1997. Visualization-based mapping of language function in the brain. *Neuroimage 6*, 4, 245–258.

PFISTER, H., LORENSEN, B., BAJAJ, C., KINDLMANN, G., SCHROEDER, W., SOBIERAJSKI AVILA, L., MARTIN, K., MACHIRAJU, R., AND LEE, J. 2001. The transfer function bake-off. *IEEE Computer Graphics and Applications 21*, 3 (May/June), 16–22.

ROSENFELD, A., AND KAK, A. 1982. *Digital Picture Processing, Volume 2*, second ed. Academic Press.

TZENG, F.-Y., LUM, E., AND MA, K.-L. 2003. A novel interface for higher-dimensional classification functions. In *Proceedings of the IEEE Visualization 20003 Conference (to appear)*.

WILSON, B., MA, K.-L., AND MCCORMICK, P. 2002. A hardware-assisted hybrid rendering technique for interactive volume visualization. In *Proceedings of 2002 Volume Visualization and Graphics Symposium*, 123–130.