IDEA

Innovations Deserving
Exploratory Analysis Programs

Highway IDEA Program

Quantitative Characterization of Asphalt Concretes Using High-Resolution X-ray Computed Tomography (CT)

Final Report for Highway IDEA Project 64

Prepared by:
Richard A. Ketcham, and William D. Carlson, University of Texas at Austin

December 2000

TRANSPORTATION RESEARCH BOARD
OF THE NATIONAL ACADEMIES

INNOVATIONS DESERVING EXPLORATORY ANALYSIS (IDEA) PROGRAMS
MANAGED BY THE TRANSPORTATION RESEARCH BOARD (TRB)

This NCHRP-IDEA investigation was completed as part of the National Cooperative Highway Research Program (NCHRP). The NCHRP-IDEA program is one of the four IDEA programs managed by the Transportation Research Board (TRB) to foster innovations in highway and intermodal surface transportation systems. The other three IDEA program areas are Transit-IDEA, which focuses on products and results for transit practice, in support of the Transit Cooperative Research Program (TCRP), Safety-IDEA, which focuses on motor carrier safety practice, in support of the Federal Motor Carrier Safety Administration and Federal Railroad Administration, and High Speed Rail-IDEA (HSR), which focuses on products and results for high speed rail practice, in support of the Federal Railroad Administration. The four IDEA program areas are integrated to promote the development and testing of nontraditional and innovative concepts, methods, and technologies for surface transportation systems.

For information on the IDEA Program contact IDEA Program, Transportation Research Board, 500 5th Street, N.W., Washington, D.C. 20001 (phone: 202/334-1461, fax: 202/334-3471, http://www.nationalacademies.org/trb/idea)

ACKNOWLEDGMENTS

We wish to acknowledge Mr. David Hirsch for providing important programming assistance in the early stages of this project, which was crucial in getting it off the ground. Ms. Charna Meth helped greatly with software testing, documentation, and image processing. We thank Dr. Naga Shashidhar for creating the phantoms that were used for verification. We also thank Dr. Shashidhar and the other members of our review panel, Mr. Maghsoud Tamoresi, Mr. Mansour Solaimanian, and Dr. Thomas Kennedy, for their time and input in helping make this project a success.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This project produced software tools and methodologies to permit the use of high-resolution X-ray computed tomography (CT) for quantitative, nondestructive evaluation of asphalt concrete pavements. These tools make it possible to analyze a core of asphalt concrete and determine its complete material makeup: the location, size, shape of each aggregate, contact relationships among aggregates, and distribution of void space. In other words, using our technology it is now possible to determine the complete internal structure of a pavement after it has been mixed and poured. In the immediate term, this analysis provides a wealth of new information that can be used to directly assess mechanical characteristics of the material. In the future, such a description promises to serve as the basis for a new generation of techniques for analysis and improvement of pavement design.

Traditional methods of characterizing and evaluating asphalt concrete pavements and design techniques involve mechanical tests, the results of which are then empirically compared to various aspects of pavement performance. Such tests contain an inherent ambiguity, as test results inevitably reflect a variety of features within a sample, each of which may have different effects on pavement performance under different conditions. As a result, the relation between test results and pavement performance is at best indirect.

Since its invention for medical diagnostic applications, X-ray CT has been adapted to address a wide array of industrial problems. With the rapidly advancing technology of industrial high-resolution, high-energy X-ray CT, it has become possible to image asphalt concretes in great detail. Rather than imaging a whole object in projection as is done with standard X-ray techniques, CT collects data along a single plane, and each resulting image is termed a ***slice*** as it corresponds to what would be seen if the scanned object were sliced along the plane. By stacking a contiguous series of slices, a complete three-dimensional data set is obtained. Because each slice actually has a thickness corresponding to the X-ray source spot size and the detector aperture, the pixels in CT images are termed *voxels* when considered as part of the volumetric whole.

CT images have already proven useful for 3D visualization of aggregate structure and void concentrations in 6-inch core samples, providing valuable insight into the internal structure of a concrete. Features that almost certainly affect pavement performance and can be observed in these visualizations include whether the aggregate is poorly mixed, with large and small aggregates separated from each other, and whether there are any unusual or aligned concentrations of void spaces. However, such visualizations are still one step removed from obtaining quantitative, reproducible data that can be used to evaluate and compare asphalt concretes and eventually improve design.

The principal thrust of this project has been to develop a software application called "Blob3D", which allows this final step to be taken. To our knowledge, Blob3D is unique in that it is designed to define and separate tens of thousands of distinct and irregular particles from a volume of CT data comprising tens to hundreds of megabytes. A rich and unique set of tools and techniques to process and interact with the data in 3D has been developed in support of this effort.

The data analysis process is divided into three steps: segmentation, separation, and extraction.

Segmentation is the process by which each voxel in the data set is classified as belonging to a particular component, such as a certain type of aggregate, or void space. Graphical analysis tools to support segmentation include filters for noise reduction and edge enhancement. Actual segmentation is done by thresholding, or selecting a range of voxel grayscales to be classified in a particular way. Specialized thresholding tools that utilize additional data (e.g., local connectivity of voxels within certain grayscale ranges) to improve segmentation were also developed and implemented. An advanced graphical interface allows real-time, interactive experimentation with the various filters to quickly achieve an optimal result.

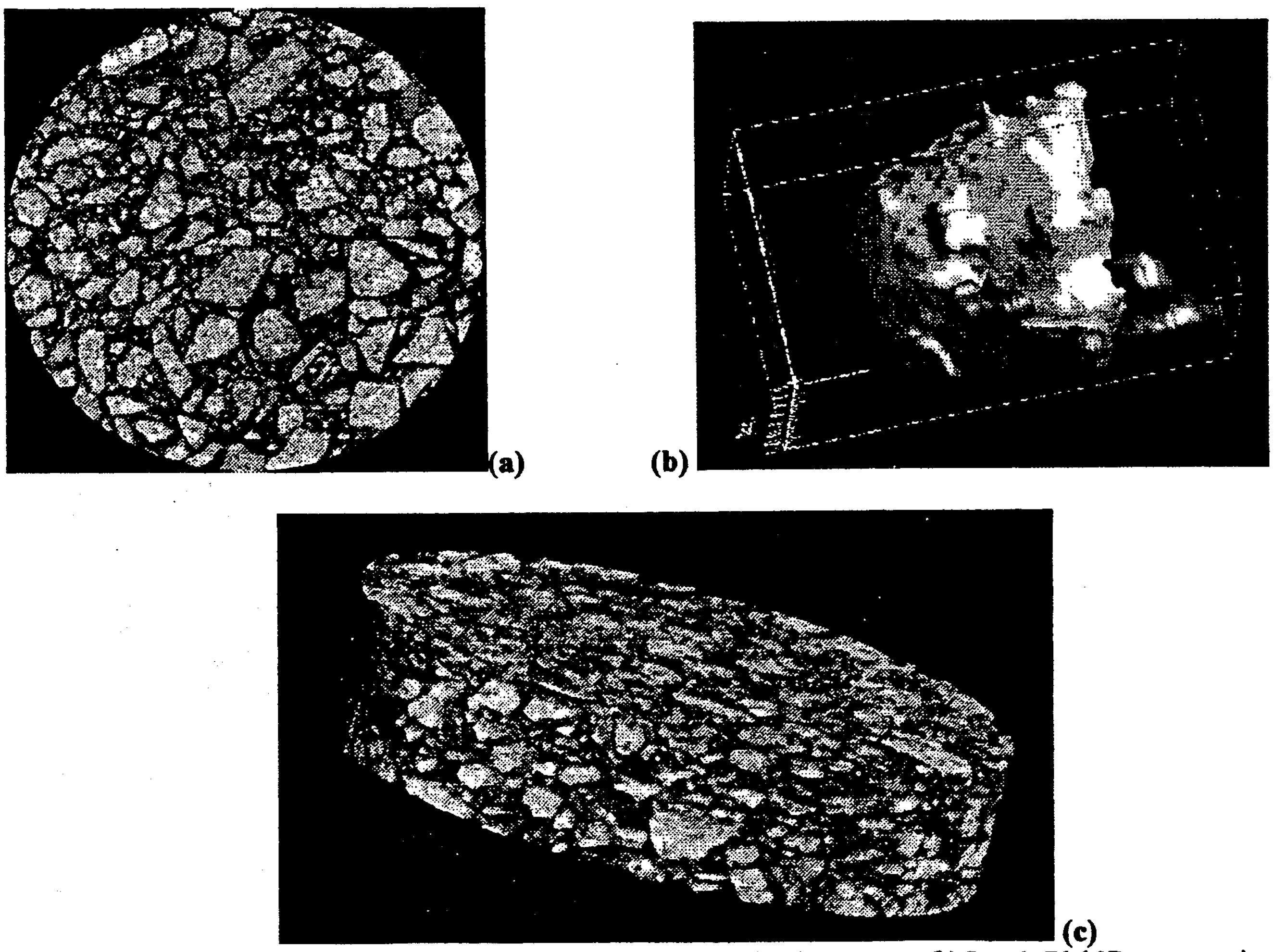
Separation is the process by which contiguous voxels classified as a particular component are divided into a series of distinct objects. A series of contiguous (touching) voxels of a single material is characteristically called a *blob*, from which the name of this software is derived. Because in an optimal asphalt concrete all aggregates should be in contact, the set of voxels corresponding to a certain material may comprise a continuous structure spanning the entire volume. Separating a blob that incorporates a potentially unlimited number of individual objects into its component parts is a challenging and problematical computational task, and solving this problem is the centerpiece of the technical challenge of this project. Associated challenges include being able to process the potentially large amounts of data involved in real time. These obstacles have been surmounted by a series of algorithms enabling efficient searching of the data volume and graphically based, user-controlled manual and semi-automatic separation of objects.

Once a CT data volume has been segmented and separated, it then becomes possible to mine it for data. Thus, a third module of Blob3D is used to extract information of interest in various investigations. The following data can be extracted: particle (or void) volume, center of mass, surface area, aspect ratio, long axis orientation; and location, direction, and surface area of all particle-particle contacts.

Blob3D was designed to be modular and extensible, so it would be straightforward to incorporate improvements and additions into all aspects of the data analysis process. For example, new segmentation filters or techniques, enhancement and automation of the separation process, and additional measurements that can be extracted from the data, could all be incorporated smoothly and easily into the existing program.

A series of tests were performed to verify that the information produced by the analysis is correct. Two phantoms consisting of glass beads in cubic and hexagonal packing were scanned and processed to test the detection and quantification of particle-particle contacts. A third phantom with glass beads of various sizes was scanned at higher resolution to test the software's ability to deal with varied data, and reproduce the grading of the bead sizes used to create the phantom. In all cases test results met expectations. In addition, insights were gained concerning the level of resolution necessary in CT data for successful discrimination of objects of various sizes.

This project has set the stage for the systematic inspection of asphalt concretes using high-resolution CT technology. The Federal Highway Administration has acquired a CT scanner capable of producing data for this analysis, and engineers there will use the tools developed in this project to process their data and evaluate test cores.



(a) (b) (c)

Figure 1: (a) Example CT scan of an asphalt concrete core. Field of view is 145 mm. (b) Sample Blob3D program view showing 3D processing to extract aggregates from scan data. (c) Three-dimensional visualization of complete data set.

IDEA PRODUCT

This project has created a new method for testing and evaluating asphalt concretes using industrial high-resolution X-ray computed tomography (CT) to quantify important material characteristics that directly impact pavement performance. Execution of this new type of analysis requires two things: access to a high-resolution industrial CT scanner, and the software that we have created (Blob3D) for analyzing the imagery obtained of asphalt concrete samples. This analysis will result in a complete material description of an already-mixed asphalt concrete, including the size, shape, composition and location of each aggregate particle (within scanner resolution), the distribution of void space, and the location and orientation of all particle-particle contacts.

This technique opens the door to a new generation of methodologies for evaluation of asphalt concretes, which promise to be more rigorous than current techniques, permitting more quantitative approaches to asphalt concrete design. These improvements will lead to better-designed pavements, resulting in substantial cost savings through reduction of premature failures.

The impact of this new technique on transportation practice is still difficult to predict at this stage, but is potentially profound. By providing a more robust set of measures of asphalt concrete properties and predicted performance, the procedures by which pavements are designed in terms of such parameters as material selection and grading and mixing parameters should be substantially improved, resulting in better-performing pavements.

In addition, the data that result from this technique provide the potential for rigorous computer modeling of pavement performance. Because this technique determines the complete material architecture of a sample, with the material properties of all of its constituents known, it will be possible to use standard computer modeling techniques to realistically simulate the effect of real-use conditions on the pavement. As a result, rather than relying exclusively upon expensive and time-consuming direct pavement testing as was done at Westrack, it will be possible to execute such testing at great savings, with no limitations on the number of pavements tested, environmental conditions, use conditions, and other important variables.

CONCEPT AND INNOVATION

Two technologies are at the core of this innovation: industrial high-resolution X-ray CT and the software developed to process the imagery.

Industrial high-resolution X-ray CT is based on the same underlying principles as conventional CAT scanners used for medical diagnosis. In standard CT, the object being scanned is imaged along a plane using an X-ray fan beam from multiple angles. The angle can be changed either by rotating the X-ray source and detectors, as is done in medical CAT scanning (in which the "A" stands for "axial"), or by rotating the object, as is more commonly done in industrial CT. The decrease in X-ray intensity due to attenuation as the beam passes through the object is measured with a linear array of detectors, resulting in a single line of data for every orientation from which the sample is imaged. These data are then processed using an algorithm known as filtered backprojection to produce the final image.

Industrial CT differs from medical scanning primarily in that it takes advantage of the lack of necessity to limit the radiation dosage given to the object being scanned. Thus, industrial scanners can utilize higher-energy X-rays, and there is no practical upper limit on exposure time. The former allow the imaging of objects that have a higher density than organic tissue. The latter allows the use of X-ray sources with smaller spot sizes and also smaller detectors than are typically used for medical purposes. Each of these refinements results in images with exceptional detail, with two or more orders of magnitude better resolution than can be achieved using medical CT: whereas a typical image slice thickness using a medical scanner is 1-2 mm, at the University of Texas we have achieved slice thicknesses of under 0.005 mm on sufficiently small objects.

Testing done at the Turner-Fairbank Highway Research Center of the Federal Highway Administration (FHWA) and the University of Texas (UT) before the project began verified that very high-quality images could be obtained from 6-inch cylinders of asphalt concretes, both on sample cores from pavements and cylinders produced by experimental mixing apparatuses. These images were readily imported into a three-dimensional visualization software package, where images of the grading of the cores and the distribution of pore spaces demonstrated the potential of the technique to distinguish specific physical characteristics of the concretes that influence performance. However, although the visualizations were very informative for qualitative characterization of concretes, they did not produce quantitative data. It was recognized that high-resolution CT data clearly has the potential to provide a complete description of the material and geometrical makeup of an asphalt concrete core. However, extracting such data form the imagery required specialized, high-performance computational tools.

A series of computer programs was written by the PI's in the Department of Geological Sciences at UT in the early 1990's to address the similar problem of extracting data on the size and locations of crystals in high-resolution CT scans of rocks. These programs required processing each 2D CT image individually to determine crystal locations and sizes, and subsequently stacking the interpreted sections to derive 3D information. Application was restricted to equant (near-spherical) crystals. This technique produced high-quality data, but was very time-consuming: a typical rock containing 2,000-10,000 crystals would take several weeks to process fully. Insofar as asphalt concretes have a much higher proportion of objects of interest, and because the shapes of aggregate particles are not well represented by a sphere, a new approach was clearly necessary. In particular, advances in the processing speed and memory capacity of affordable computers bought about the possibility of processing the data set with the aid of 3D visualization, rather than being limited to working with one image at a time.

A number of commercially available software packages are able to extract objects from three-dimensional data sets (such as VoxBlast and Materialise). However, these products are typically geared to extracting a fairly small number of relatively large objects from a data volume. Because of this, the tools for separating objects in contact with each other tend to be cumbersome. Furthermore, none of them calculated or recorded the full suite of data of interest, such as contact relationships.

Thus, the central thrust of this project was to create original computer software that would enable the desired analysis. The program is called Blob3D, as it implements a 3D version of what is traditionally called a "blob analysis" for extracting objects from 2D images. For each type of object that is to be extracted from the data set (a particular aggregate, or void), the first program module is used identify which voxels correspond to that component, a process known as *segmentation*. Then all contiguous sets of voxels of that type are identified; these sets are called ***blobs***. A single blob can contain many distinct objects that are in contact, so another program module is used to separate all touching objects. This module currently relies on the user to make all decisions concerning how and when to do separation, to ensure data quality. Once this task is complete, a final module is used to extract that data of interest (particle and void sizes, shapes, contacts, etc.).

The Blob3D program was written in IDL, a high-level scientific data processing and visualization programming environment created by Research Systems, Inc. of Boulder, Colorado. The advantages of using IDL are that many image processing and visualization algorithms are provided as a part of the package, eliminating the need to reinvent the wheel for many of the graphical and calculation components of the program. Furthermore, IDL implements a very efficient array-oriented calculation scheme that greatly enhances processing speed for volumetric data over what is possible in most computer languages. Programs developed in IDL also have the potential to run on any platform (Windows, Macintosh, Linux, Unix), although minor incompatibilities may need to be resolved. The single drawback of using IDL is that potential users of the program must purchase a copy of IDL; a run-time license that permits programs to be run but does not allow development of new software costs $300.

Innovations in Blob3D include:

- An enhanced volume-searching algorithm for locating and determining the extent of regions of contiguous voxels, which is significantly faster even than the algorithm provided with IDL. The efficiency of this algorithm is such that it makes Blob3D able to run very quickly on relatively inexpensive computers (<$2000).
- Original implementations of noise reduction and edge enhancement filters that process three dimensional data, rather than two dimensional as is available in most popular image processing packages (Adobe Photoshop, NIH Image). These filters also automatically take into account voxel aspect ratio (where the distance between pixels in each CT slice is not the same as the distance between slices).
- Original thresholding algorithms that use localized data features to improve the program's ability to correctly delineate objects.
- Semi-automated separation of touching objects using an "erode/dilate" scheme, in which a composite object is shrunk uniformly along all of its surface until it splits into separate objects along thin necks, and then is regrown while keeping the separate objects distinct.
- Manual separation of touching objects with a user-entered cutting plane.

- Efficient and automatic cut-off of growing blob regions so they don't swamp computer memory or processing capacity, ensuring continuous real-time performance.
- Efficient methods for extracting a number of unique three-dimensional parameters, such as the location, surface area, and normal direction of aggregate-aggregate contacts, and measures of three-dimensional aggregate shape and orientation.

Blob3D was also written with careful attention to make it easily extensible, so new procedures and techniques can easily be added, both from the standpoint of programming and of maintaining a consistent and easy-to-understand user interface.

INVESTIGATION

OVERVIEW OF PROJECT PROGRESS

Previous investigations at UT with the FHWA during 1997 and 1998, in which 33 asphalt concrete cores were scanned, established that high-resolution X-ray CT had the potential to generate detailed data on asphalt concretes. The success of these pilot studies led to the proposal for this project, and also to the acquisition by the FHWA of an industrial CT scanner with the same design as the UT instrument.

Following acceptance of the proposal for this project in the spring of 1999, a three-phase plan was created for its implementation. The initial stage of the project was dedicated to establishing the underlying software architecture, with the goal of being able to handle simplified test cases. The second phase was dedicated to expansion of program capabilities to allow it to deal with more real-world situations, taking into account the variability that will be encountered when asphalt concretes with varying aggregate composition and shape are encountered. The third phase was to then use the program in a small pilot project on a real set of asphalt concrete cores.

During the first stage of the project, the basic scheme of dividing the computational task into three parts was conceived and implemented. The first part, segmentation, consisted of assigning each voxel in the data volume to a component (aggregate type, void, matrix). Once the entire volume was segmented, the second part of the task was separation, taking all sets of contiguous voxels of each component and dividing them into distinct objects. After this task is complete, the third part of the program would extract the information of interest from the processed data. Each of these three divisions is discussed in further detail below. During the first stage of the project, we were able to establish the overall program architecture, and implemented functional versions of the modules for segmentation and extraction. The separation module identified and labeled all regions of contiguous voxels, but was not yet capable of further processing. The graphical program interface for viewing and interacting with volumetric data was created, with functionality for viewing along orthogonal cut surfaces.

The second phase of the project saw further progress for the segmentation module, with a number of new noise reduction and edge enhancement filters implemented, and a new thresholding method. The separation module progressed, and implemented manual separation using a surface erosion scheme. The graphical interface was enhanced to provide 3D viewing of objects using isosurfaces. During the second panel meeting, the panel members recommended a number of tests for verification of program results, consisting of creating phantoms with known properties, scanning them at the UT facility, and processing them to ensure the expected results would be achieved.

Soon after the second panel meeting, it became apparent that the program flow chosen in the early stages was not optimal. Whereas we had originally intended to segment the entire volume of data before separation began (i.e. assign each voxel to a certain type of material), it became apparent that during the separation process some voxels might be recognized as having been allocated incorrectly, and that these should be reassigned to some other component. In other words, it became clear that separation itself could be the final stage of a segmentation process. As a result, the program architecture was changed so that once a component was segmented, instead of proceeding directly to segmentation of other components, the component would have to go through separation first. This change involved substantial reorganization many parts of the program.

In addition to the reorganization, great strides were achieved in all aspects of the program during the third phase. The speed of many aspects of the program was increased by deriving and implementing an improved algorithm for searching and mapping blobs in the 3D volume. This improvement in program performance proved to be the key to making it practical to use the software on lower-cost systems. More new segmentation filters were implemented, and the level of quality control on that module of the program rose significantly. The separation module achieved full functionality, with semi-automated erosion/dilation implemented, as well as manual separation by defining a cutting plane. Location and measurement of contact points, contact normals, and contact surface areas was implemented in the extraction module.

With help from Naga Shashidhar of the FHWA, who made the phantoms, three verification tests were done during the third phase, as documented in more detail below. The first two tests verified that the calculation of contact points and normals worked correctly, using phantoms consisting of glass spheres of identical size glued together with cubic and hexagonal packing. The phantom for the third test was made from spheres with a range of sizes and was used to verify that the program could successfully reproduce the grading of the spheres. Results were excellent for all tests.

One important insight that was gained from the latter test, and from subsequent attempts to process the high-resolution CT imagery of asphalt cores obtained before the project began, was the importance of the voxel aspect ratio: the ratio between the pixel spacing (mm/pixel) in the slice images, and the spacing between slices. For almost all of the CT data on cores, the pixel spacing was 0.283 mm (a 145 mm field of view for each 512x512 pixel image), and the slice spacing was 0.8 mm, resulting in an aspect ratio of almost 3:1. These parameters were chosen at the time because they resulted in clear slice images and good 3D visualizations of the entire data sets, while allowing more cores to be scanned on the available budget by keeping per-core costs down. However, the high resolution observed in the 2D images led to unrealistic expectations for what could be extracted from the 3D data sets, because of the great disparity in resolution (see Fig. 2). As a result, data for previously scanned cores will need to be re-acquired in order to successfully resolve aggregates less than about 2 mm in size.

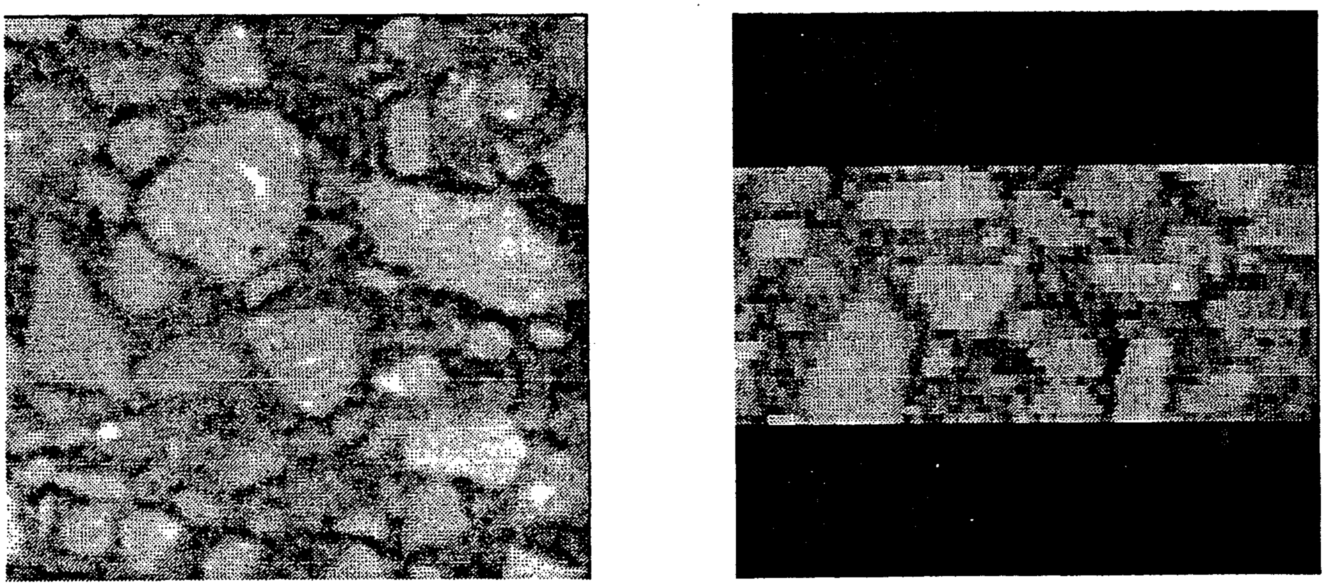


Figure 2. Example data from high-resolution X-ray CT scan of asphalt concrete core. The left figure shows a subsection of a typical image in the scan plane, with good resolution. The right figure shows a view of the same data set sliced along an orthogonal plane. The loss of resolution in the z dimension (vertical in the right-hand image) is caused by the inter-slice spacing being almost three times as high as the inter-pixel spacing in the scan plane. Higher-resolution data can be obtained with the scanner, but takes longer. The high aspect ratio can negatively impact the degree of detail obtainable from analysis of this data volume.

PROGRAM DESIGN

In this section we briefly describe the different components of the Blob3D software. In addition to this section, a more complete set of program documentation for people interested in using the program is supplied in an appendix.

Navigator

The Navigator (Fig. 3) is the central organizing unit of the program, ensuring the proper flow of data through its various stages of processing. It is always present, regardless of what stage of processing is being performed. It begins by allowing the user to read in the CT data, as a set of individual TIFF-format image files. It the queries the user for the pixel and slice spacing of the data, followed by the names of the components the user wants to extract from the data volume. Once the data has been read in, the Navigator allows the user access to the various processing modules of the program: Segment, Separate, and Extract. The Navigator also organizes various other aspects of the program. It permits the user to save and re-load the current state of the program from almost any point, allowing work in progress to be preserved. It also keeps track of various user preferences, such as the default size for data viewing windows.

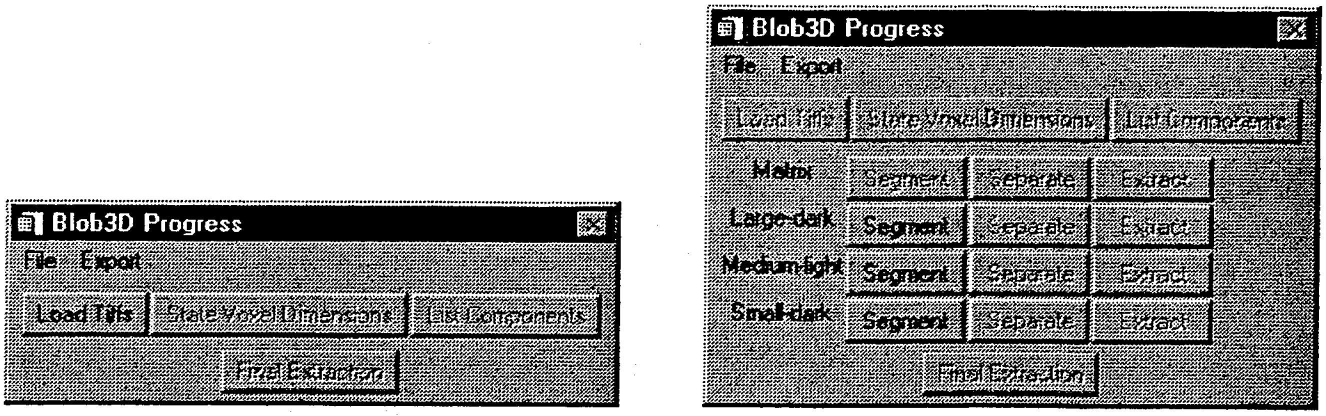


Figure 3. Navigator windows from two stages of program execution. Left window is shown at program startup. Right window shows state after data has been loaded.

The Navigator window shown in the left side of Figure 3 is what the user sees immediately after the program is started up. The principal available option, shown as the non-grayed button, is for the user to "Load Tiffs", specifying where the data files that are to be processed reside. Once this is complete, the program directs that the user enter the voxel dimensions, if necessary, followed by the names of the components to be analyzed.

After the data entry process is complete, the Navigator takes on an appearance similar to that shown in the right side of Figure 3. The original data entry buttons have been grayed, and the user is given the option to Segment any of the components that have been specified. Once segmentation is complete for a component, the Separate button for that component is un-grayed, and the other segmentation buttons are grayed, until the separation is completed.

Segment

The Segment module provides the user with a series of tools to aid in the selection of which voxels in the data set belong to a particular component. It implements a number of filters for smoothing the data, enhancing edges, adjusting brightness and contrast, defining both volume-wide and local mechanisms for mapping grayscale ranges to the object of interest, and removing holes and islands from the thresholded data. It implements these filters in a scheme that allows the user to see how the results of the processing map back to the original data, to help ensure a correct result. It maintains a standard object-oriented graphics pipeline that ensures that all display windows are up to date, even when the parameters of a filter used early in the process impact the input and output for later filters.

An example Segment module interface window is shown on left side of Figure 4. The "Segmentation steps" section shows the list of filters that have been used thus far. The "Add filter" drop list allows the user to add filters at any stage of the pipeline; the "Edit", "Remove", "Deselect" and arrow buttons allow the user to change, delete, and rearrange the filters in the list. At all stages, the list and buttons are maintained in the correct state so the user can only do actions that are reasonable given the current program state. The "Display" button allows the user to bring up a window that shows the result of processing with whichever filter in the steps list is currently selected. If the "Interactive display" box is checked when a viewer is brought up, the viewer updates while filter parameters are being entered, so that the user can interactively see the effects of varying the parameters. Each filter has its own set of parameters that must be entered to specify how it is to execute its calculations. An example data entry window for the Threshold filter is shown in the right-hand window in Figure 4.

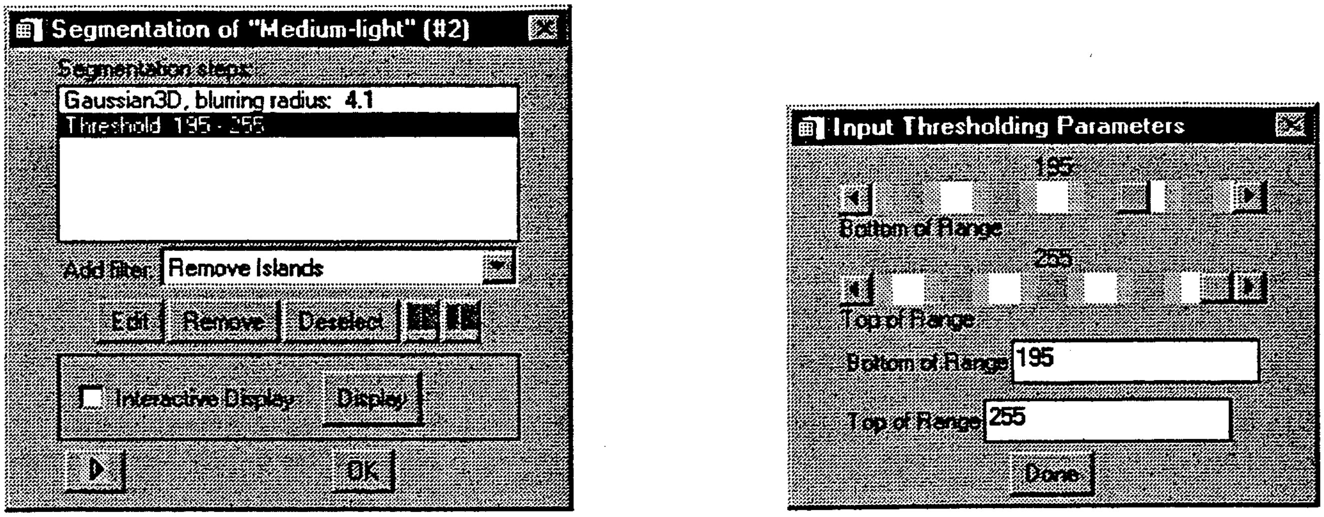
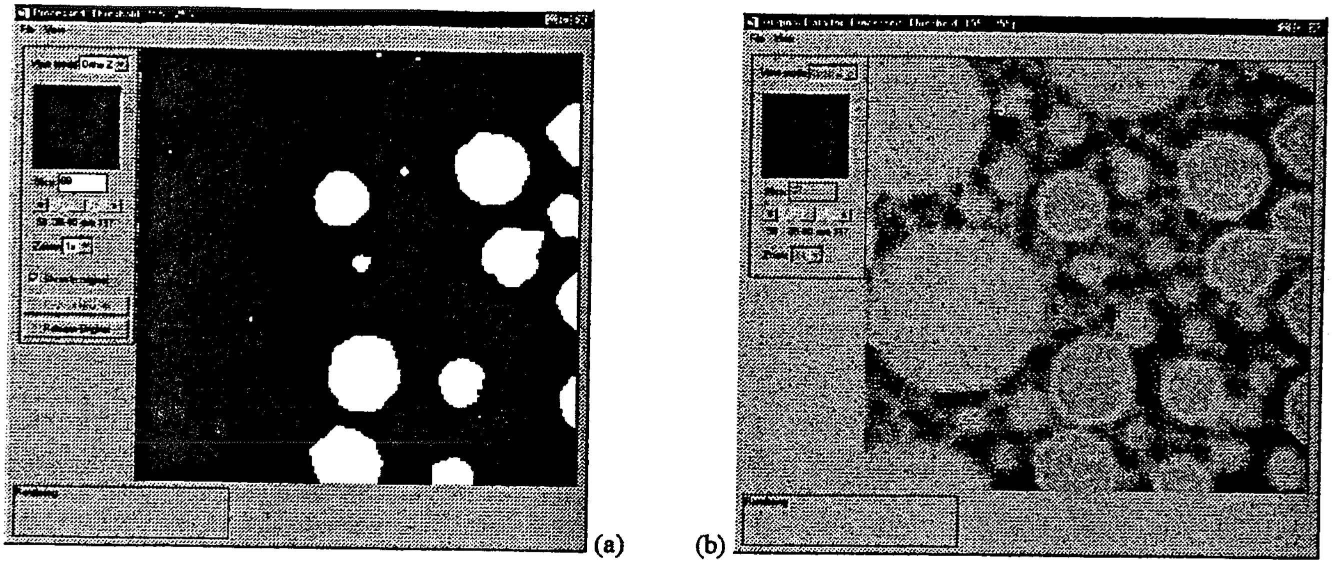


Figure 4. Segmentation module windows. The left window manages the segmentation process, while the right window is an example of a parameter entry window for a segmentation filter.

In order to be able to quickly show the user the results of processing without having to perform calculations on all of the 10's to 100's of megabytes that comprise a typical data set, a sub-volume is extracted from the data for processing while the user decides upon the optimal set of filters. The extent of this sub-volume can be accessed and changed using the triangle button on the bottom of the segmentation window.

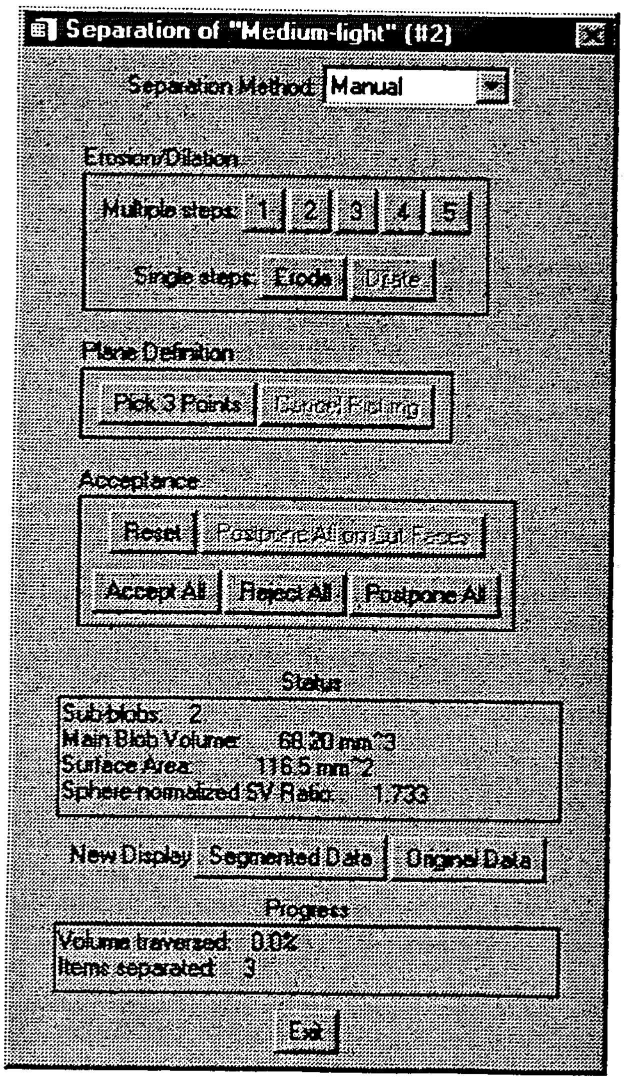
There are three basic categories of filters currently implemented. The first set takes grayscale CT data as input and provides grayscale data as output. These filters include "Adjust Levels" for enhancing data contrast, "Gaussian 2D Smoothing", "Gaussian 3D Smoothing" and "Median Smoothing" for noise reduction, and "Edge Enhancement" to sharpen the boundaries of objects in the images. All except "Gaussian 2D" are implemented so they process data in 3D, and correctly account for the voxel aspect ratio. The second category of filters takes grayscale data as input and provides binary, or black-and-white, data as output. Once the data have passed through one of these filters, voxels appearing as white are treated as belonging to the component being segmented, while black voxels are excluded from it. The basic segmentation scheme is "Grayscale Range," which maps a range of grayscale values to white and leaves the rest as black, a process also known as thresholding. Two additional filters expand on this functionality to incorporate additional information into the segmentation. "Seeded GS Range" maps to any white any voxel within a certain grayscale range provided that it is directly or indirectly in contact with voxels with grayscale in a special seed range. "Expanding GS Range" was custom-designed to take advantage of an idiosyncrasy of CT data: rather than having distinct boundaries, all objects blur into each other to some extent because of the finite resolution of the data, and because of "partial-volume effects" which come about because the volume encompassed by a voxel may contain more than one component, and in such cases the grayscale of that voxel will be some average of the grayscales for those two components. As a result, voxels near the proper boundary for an object will tend to have lower grayscales than voxels at the center. The Expanding GS Range filter allows this situation to be accounted for by specifying one "central" range of grayscales to map to white, followed by an additional "expand" range in which voxels may be selected if they are within a certain distance of "central" voxels. This has proven to be one of our most successful filters. The final category of filter takes binary data as input and produces binary data as output. These include "Remove Holes" and "Remove Islands", to change the classification of small clusters of voxels.

A segmentation may be considered complete when the data that result from processing are binary rather than grayscale (Fig. 5a). Once the processing has reached this stage, the user can compare this result to the original grayscale CT data in a special viewing window (Fig. 5b). Once the user is satisfied that the match is acceptable, the Segment window is closed, and the entire data set is processed using the selected set of filters and their parameters.



(a) (b)

Figure 5. A component segmentation can be considered complete once the data is reduced from grayscale to binary. The left window shows a binary image, and the right window shows the corresponding pixels superimposed over the original grayscale data in pink.



Separate

The Separate module finds all blobs (sets of contiguous voxels belonging to a single component), and enables the user to specify whether each blob consists of one or more distinct objects. A set of tools allows the user to allocate each voxel within a blob to a particular component, including manual and semi-automated erosion/dilation, and manual entry of a dividing plane.

The Separate module commences by systematically searching the data volume until it finds a voxel belonging to the component in question that has not been processed yet. Starting with this voxel, it then searches the volume to find all contiguous voxels. This set of voxels, called a blob, is then shown to the user for evaluation. The user can either accept it as a single object, reject it (in which case the voxels become unclassified and may be re-segmented as a different component at a future step), or begin to subdivide it using the available tools. Once the blob is completely processed, the Segment module continues its traverse of the data volume until the next unprocessed voxel is encountered, until finally the process is complete.

Figure 6 to the left shows the Separate module control window, which gives the user access to all currently implemented separation techniques. At the top of the window is a selector allowing the user to choose which mode of separation to employ. Aside from the "Manual" mode that is depicted in the figure and described here, there is also an option that allows the all blobs to be treated automatically as single objects; such an option may be chosen when one is processing voids, for example, as there is probably no reason for them to go through a separation step. Below is the series of controls that implement the erosion/dilate scheme of separation, both manually and semi-automatically. These are followed by the controls that allow manual cutting plane definition. Next is a series of controls for accepting, rejecting, and postponing processing on the blobs. These are followed by controls to open display windows of the segmented data being separated, along with the corresponding volume from the original unprocessed grayscale data.

Erosion/dilation works by progressively removing voxels on the outer margins of a blob. This has the effect of causing objects to break apart at necks where the thickness of the object is at a minimum (Fig 7). Once a certain number of erosion steps has been executed and the original blob is broken into a number of sub-blobs, a dilation operation is performed, in which all of the voxels that were eroded away are reapportioned to one of the sub-blobs. The result is often a natural, well-placed separation. Erosion/dilate has been implemented to either allow the user to erode one step at a time and inspect the result before dilating, or to simply do a set number of erosions and then re-dilating and displaying the end result. The sub-blobs that result can be accepted, rejected, or postponed individually or as a group.

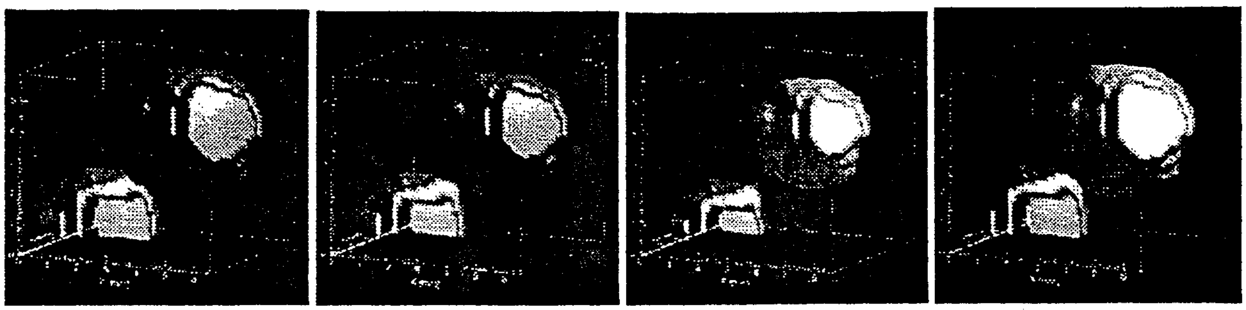


Figure 7. Four steps showing separation by erosion/dilation. The first image shows a blob consisting of a pair of objects that need to be separated. The second shows the same blob after a single erosion step. The third image shows the result of a second erosion step. The blob has pinched off along the narrow neck, and the two objects have been separated. In the fourth image the dilation function has been used, re-apportioning all voxels in the original blob to the two objects.

Plane definition works by allowing the user to click three points on any object in the display window and using them to define a plane, which then separates the blob into two and perhaps more sub-blobs, depending on the shape of the blob and the orientation of the plane.

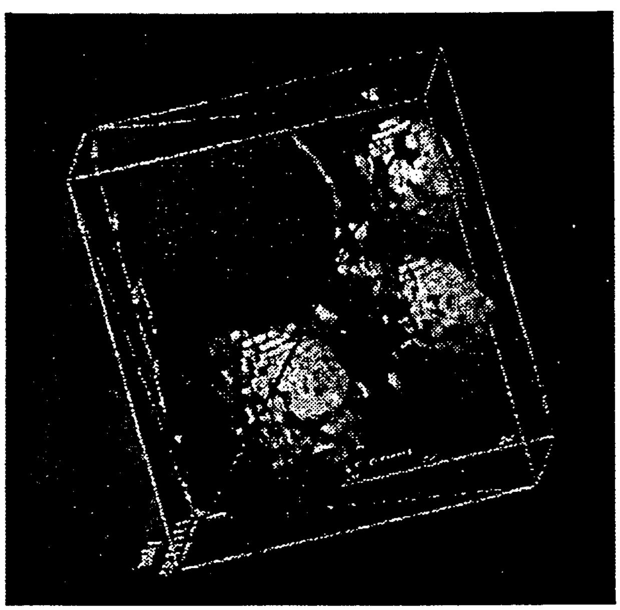


Figure 8. Separation module display indicating that the blob was cut off before growing too large for computer resources to handle effectively.

One special feature in the Segment module accounts for the possibility that a single blob can span much or all of the data volume. In such a case, the single blob can grow to be up to 10's of Mb in size. Such blobs would take a long time to fully map out, and require extensive computation time and graphics resources to display. Rather than slowing the program to a crawl when such blobs are encountered, Separate implements a very fast scheme that causes a blob to stop growing once it reaches a user-defined size. This allows the cut-off blob to be processed while the computer's processing speed and graphics and memory resources still permit reasonably efficient processing. The user is warned when this function is triggered by a series of red lines in the display window marking which faces of the blob volume were cut off (Fig. 8). After some separation has been done, program functions are then provided to automatically postpone the processing of any sub-blobs that intersect cut boundaries, allowing the applicable voxels to be processed at a later time after voxels on the far side of the cut face can be included with them.

The Separate module also calculates data concerning blobs and sub-blobs that may be of use to the user in making decisions about the data. These include the volume, surface area, surface-to-volume ratio, and sphere-normalized surface-to-volume ratio. The last is a calculation original to this program (to the best of our knowledge) that has the advantage of scaling to any object size, unlike the surface to volume ratio, which can vary with radius (or even radius units, giving different answers if inches and millimeters are used).

Extract

The Extract module is used to take a partially or fully segmented and separated data set and derive data of interest. It can be run once any component has been separated, although measurements can only be made on components that have gone through the Separate module.

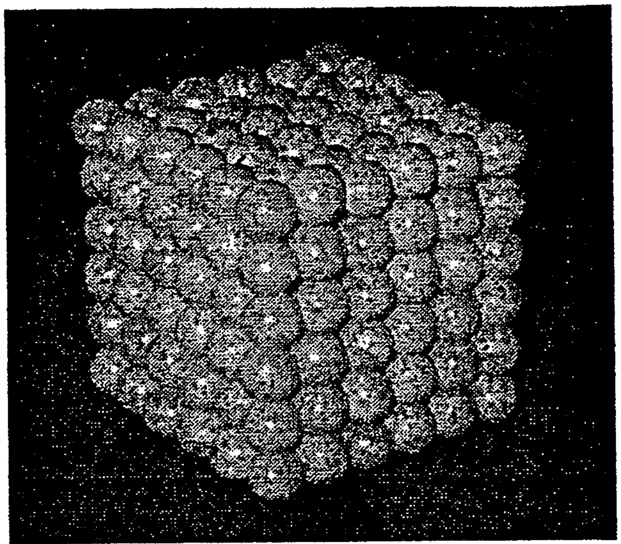
Two basic types of information can be extracted: data intrinsic to each object (volume, surface area, shape, center of mass, orientation, etc.) and data about the relationships between objects (contact points, contact normals, contact surface areas). For each type of data, the user simply selects which components to derive which types of data from, and the program proceeds. The data are written to a tab-delimited text file that is readable in any standard spreadsheet or word-processing program.

VERIFICATION

A series of tests were performed to verify that the program procedure and calculations produce reliable results. The first pair of tests verified that the calculations of contact properties are correct, and that the correct number of objects was detected. The third test verified that the program could reproduce the grading characteristics of a sample, and to successfully process a sample with subtly different components.

Hexagonal and Cubic Packing

The phantoms used for these tests were made by Dr. Naga Shashidhar of the FHWA, and are shown in Figure 9. Each consists of a series of uniformly sized marbles glued together with different modes of packing: cubic, where each marble is in contact with up to six others in the orthogonal directions, and hexagonal, in which each marble may be in contact with up to twelve others.



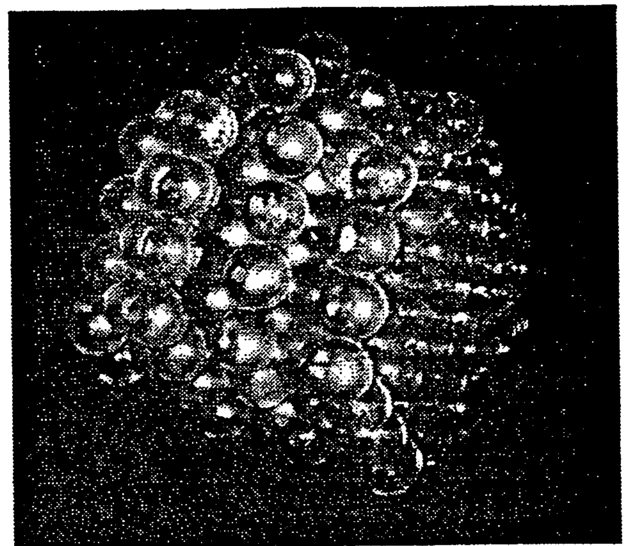


Figure 9: Phantoms for testing Blob3D measurement of contact information. The left phantom has cubic packing, and the right has hexagonal packing.

Both phantoms were scanned at the University of Texas, and processed using then-current versions of the Blob3D software. Scanning was performed by Dr. Richard Ketcham, and processing was done by Charna Meth and Dr. Richard Ketcham. Segmentation and separation of each sample was straightforward, due to the simplicity of the material.

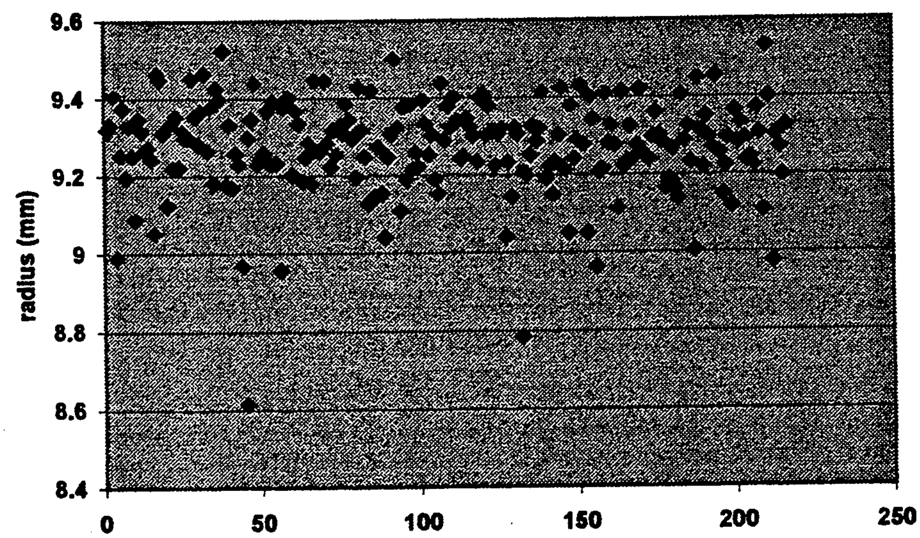


Figure 10: Distribution of marble sizes in cubic packing phantom

Figure 10 shows the distribution of marble sizes detected. The mean radius is 9.27 mm. This matches well with caliper measurements of the marbles themselves. The variation observed is small, and at least partly caused by true variation in the marbles themselves, as they are neither strictly round nor uniform.

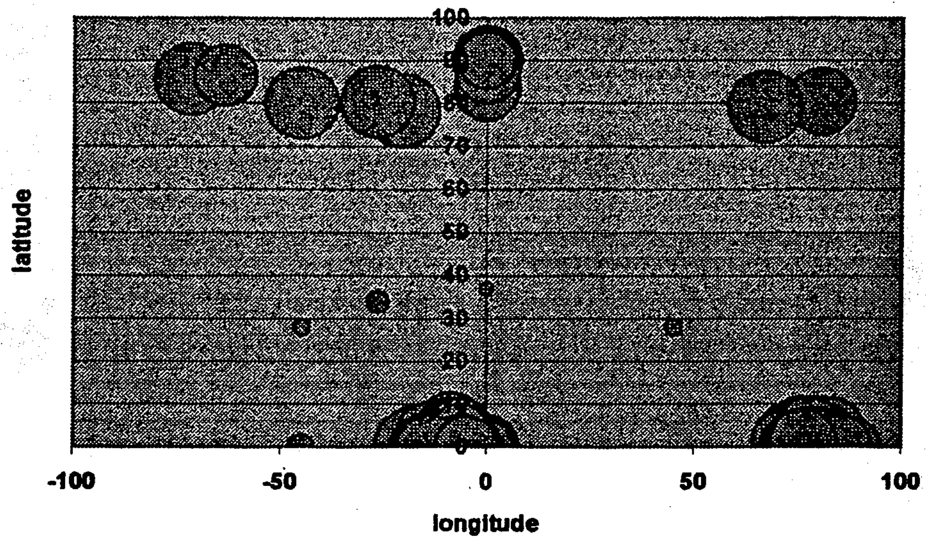


Figure 11: Bubble plot of contact normal information for cubic phantom. Bubble locations reflect direction (latitude and longitude), and bubble sizes reflect surface area of contact.

Figure 11 shows the distribution of contact normals. The result is as expected, with almost all contact normals either at zero latitude and offset 90 degrees from each other, or at nearly 90 degrees latitude. A small number of spurious measurements are apparent, and easily detectable from their small surface area relative to the others. The spurious measurements may be due to operator error, or may be due to a minor program bug that has since been fixed. The variation in normal locations is partly due to the phantom itself being slightly irregular.

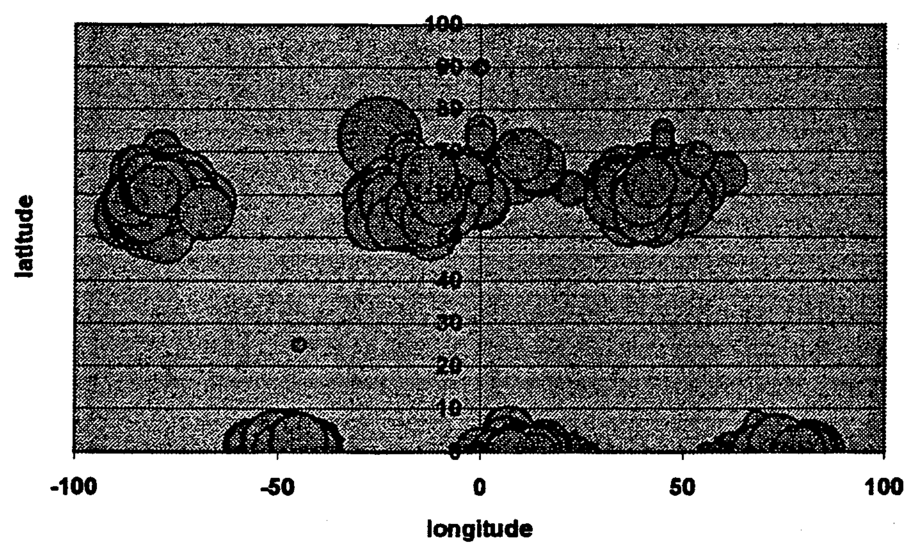


Figure 12: Bubble plot of contact normals and surface areas for hexagonal packing phantom. Bubble locations reflect direction (latitude and longitude), and bubble sizes reflect surface area of contact.

Figure 12 shows the results for the analysis of the hexagonal packing phantom. Once again the results are as expected, with three different populations each at zero and 60 degrees latitude, spaced at 60-degree longitude increments. Once again, the irregularity is in part due to slight imperfections in the phantom.

Grading

Another sample produced by Dr. Naga Shashidhar at FHWA was used to test how well the program can reproduce sample grading. The phantom consisted of a 6-inch diameter cylinder, with thousands of glass spheres of various sizes ranging from 1 mm to 12 mm in diameter, held together by an asphalt binder. The sample was scanned at very high resolution, with a 1024x1024 reconstruction of a 140 mm field of view and an inter-slice spacing of 0.4 mm. An example from this data is shown in Figure 13. A complete review of the processing of this sample is given in the next section of this report.

Processing results are given in Table 1. The size measurements of the spheres vary slightly from the manufacturer specifications. This is due in part to some variation in the spheres. There also may have been some degree of operator error in selecting the correct grayscale ranges for segmentation. It is possible that there are subtle biasing effects induced during the segmentation process. When a component is processed, it is advantageous to minimize the number of voxels from other components that are included in the segmentation, so grayscale ranges are selected that are less likely to include unwanted voxels. This would tend to bias results towards making components include fewer voxels than would be ideal. In this study, all components had measured mean radii slightly smaller than anticipated, although in all cases the errors were small (<5%). The extent of variation in the 6.25 mm component probably reflects true variation in the materials.

Table 1. Results of analysis of phantom G3.

Manufacturer specified radius (mm)	Mean measured radius (mm)	Original grading (wt. %)	Measured grading (wt. %) (side contacts not included)	Measured grading (wt. %) (side contacts included)
1.15-1.30	1.11 ± 0.12	18.2	25.0	19.0
2.5	2.39 ± 0.03	36.4	41.8	37.4
5.0	4.79 ± 0.03	25.4	15.3	43.6
6.25	6.15 ± 0.20	20.0	17.8	

In order to be able to complete the analysis of the phantom in a reasonable amount of time, only a sub-volume of the complete CT data set was processed. Although a large statistical sample was taken (over 8,300 spheres were measured), the confined volume leads to potential biases in the result, as larger spheres are more likely to be truncated by the edges of the volume. As a result, the grading estimated from using only spheres entirely enclosed within the sub-volume (Table 1, Column 4) is more heavily weighted towards the smaller fractions. However, because the three different component types were distinguished during segmentation, it is possible to include truncated spheres in the calculation, although the largest two size fractions have to be combined. The result, given in Table 1 Column 5, shows excellent agreement with the expected values in Column 3, with all three results within 2%.

EXAMPLE DATA PROCESSING

To demonstrate a complete program run to process a sample, we document here the steps taken to do the third verification test, of the phantom with graded spheres. The processing involved segmentation and separation of three different components: glass spheres of various sizes and grayscales, the latter probably reflecting subtly different glass compositions.

It was found to be very advantageous to be able to segregate the three components from each other and process them in isolation. With the completion of each component, processing the remaining ones became progressively simpler. It is clear that a "divide and conquer" strategy will often be preferable to attempting to process all types of aggregate in the volume at the same time.

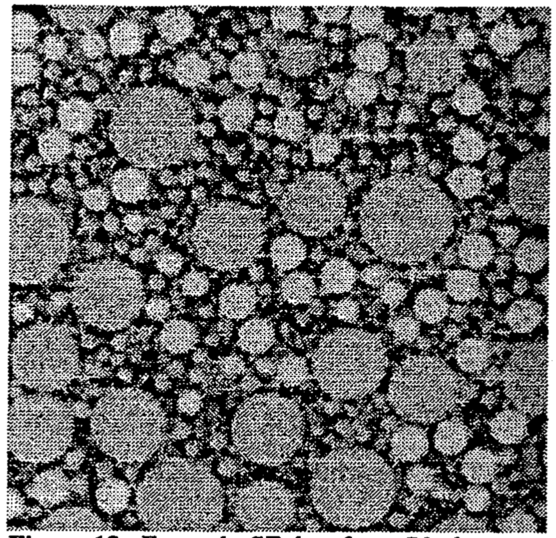


Figure 13. Example CT data from G3 phantom.

An example of the original data is shown in Figure 13. The spheres have two distinct grayscales, with brighter ones at about 5 mm in diameter, and darker ones at larger and smaller sizes. There are 5 size classes in the sample, with radii of 0.6 mm, 1.2 mm, 2.5 mm, 5 mm, and 6.25 mm. All size classes can be distinguished in the figure. However, because the scan slice spacing was large compared to the pixel spacing, the smallest size class cannot be distinguished in the z dimension, and thus could not be reliably analyzed. The two largest size classes are indistinguishable on the basis of grayscale, so they were treated as a single class. The second smallest size class is subtly different from the largest ones, so we were able to treat it separately. The bright spheres are clearly distinguishable from the others. Thus, there were a total three components to be processed. Because it was the most distinct, the first component to be processed was the bright spheres. The segmentation steps were:

1) Gaussian3D, blurring radius: 5.0
2) Threshold, 201-255, Expand by 4 into 162-255

The first step was used to decrease the level of noise in the data. The second step uses an expanding threshold, in which the first grayscale range captures the centers of the desired spheres, and the second allows the selected area to expand into the surrounding regions to capture the boundary voxels. The resulting segmentation is shown in Figure 14.

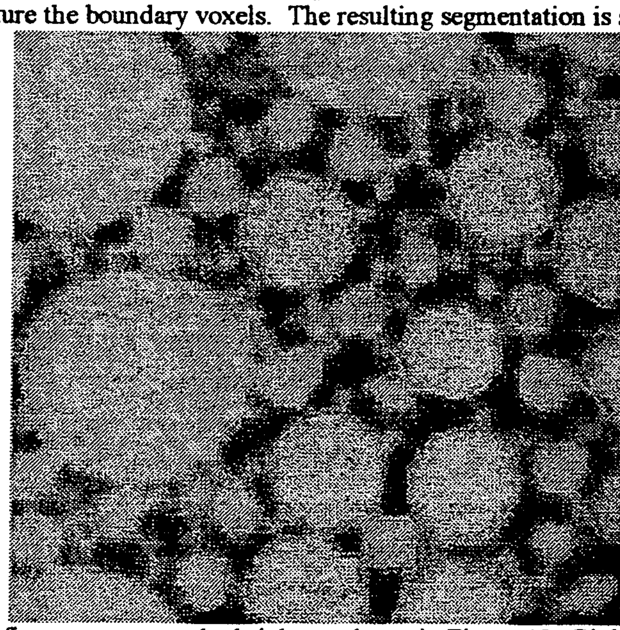


Figure 14. Segmentation of first component, the brighter spheres in Figure 13. Pink areas denote selected voxels.

As can be seen, this segmentation was quite robust, and little foreign material was included. The separation was thus straightforward. An example is shown in Figure 15.

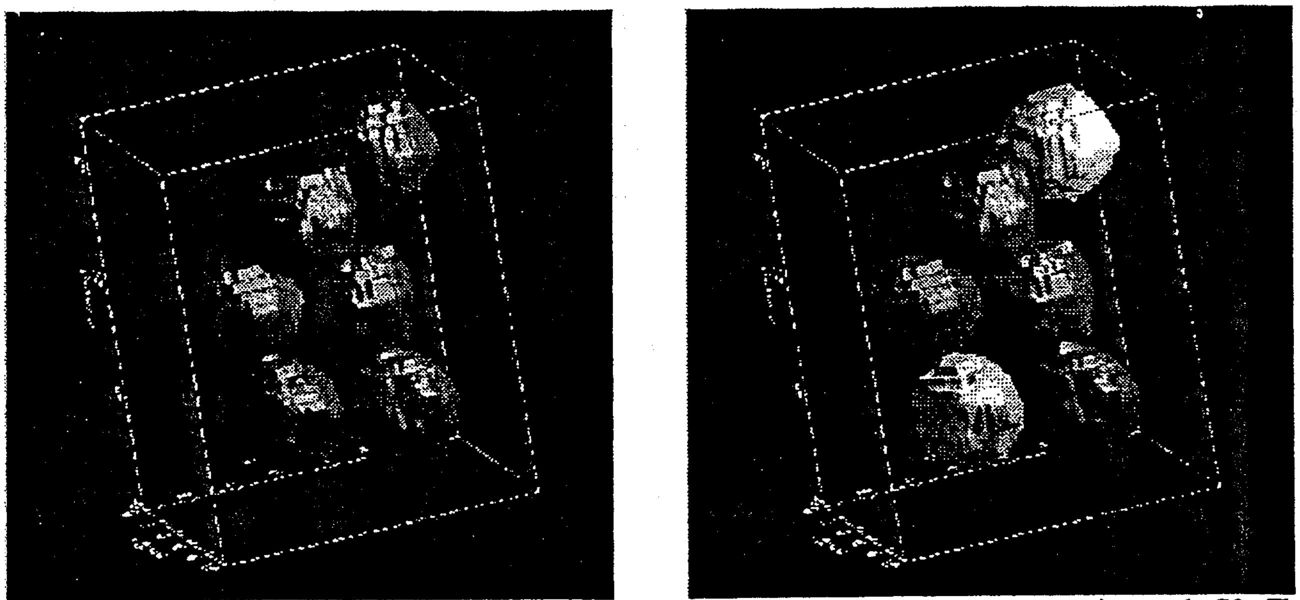


Figure 15. Example separation of a single blob comprised of six glass spheres of first component in sample G3. The separation was done by semi-automated erosion/dilation with 3 erosion steps.

The second component to be processed was the larger spheres. Because they had grayscales very similar to the smaller size fractions, a segmentation that excluded the smaller fraction was impossible. However, because of subtle differences in mean grayscale the two components could also not be accurately segmented with identical parameters. Thus, filter parameters were optimized to process the larger fraction, and the smaller fraction was removed during separation. The segmentation steps were:

1) Remove separated components, infill with 0
2) Adjust Levels, Input 125, 1.00, 200
3) Gaussian3D, blurring radius: 3.0
4) Threshold, 196-255, Expand by 4 in to 147-255

The first filter removes all previously separated components, overwriting their voxels with a constant grayscale value. The second adjusts the grayscale levels of the volume, in order to increase contrasts. The third filter removes noise, and the fourth defines an expanding threshold, as defined above. Note that the grayscale values selected for the thresholding are not directly comparable to those used for the previous segmentation, because the underlying grayscales were changed. An intermediate stage and the final segmentation are shown in Figure 16.

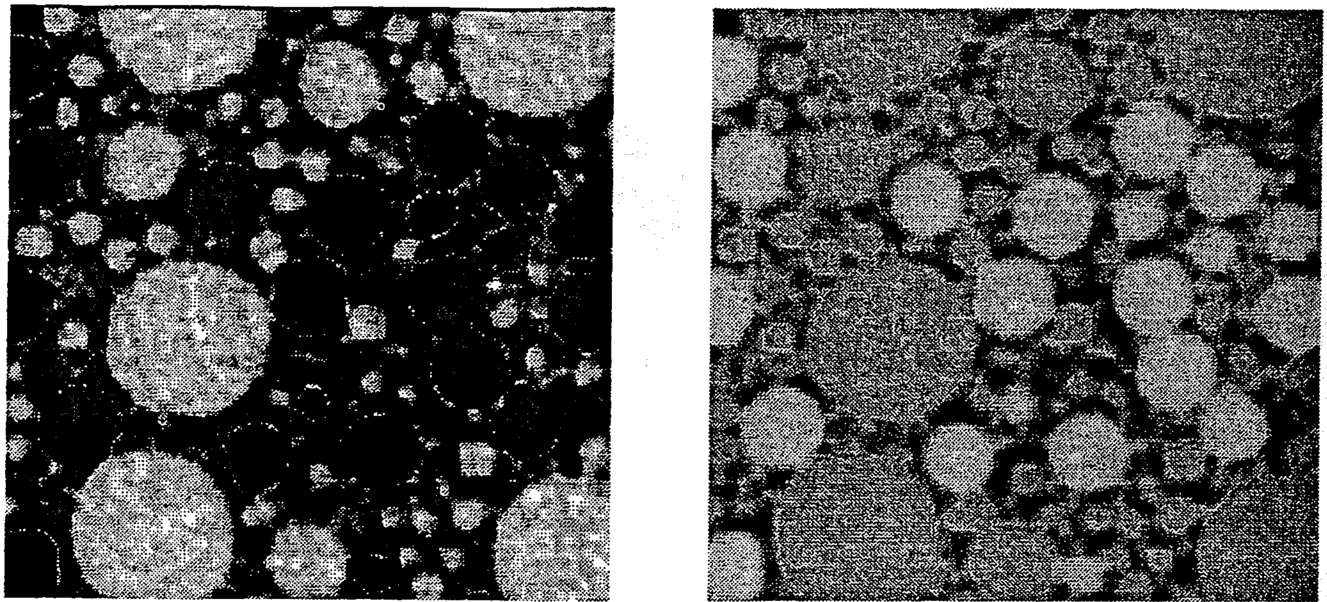


Figure 16. Segmentation of the large spheres in sample G3. The left figure shows the state after step 2 above, and the right figure shows the final segmentation superimposed on the original data, with selected voxels in pink. Small spheres are also included in the segmentation, but were easily removed during separation because of the size difference.

The separation process was somewhat more labor-intensive than for the first component, because of the additional work required to exclude the smaller spheres that were included in the segmentation. A typical series of steps is shown in Figure 17.

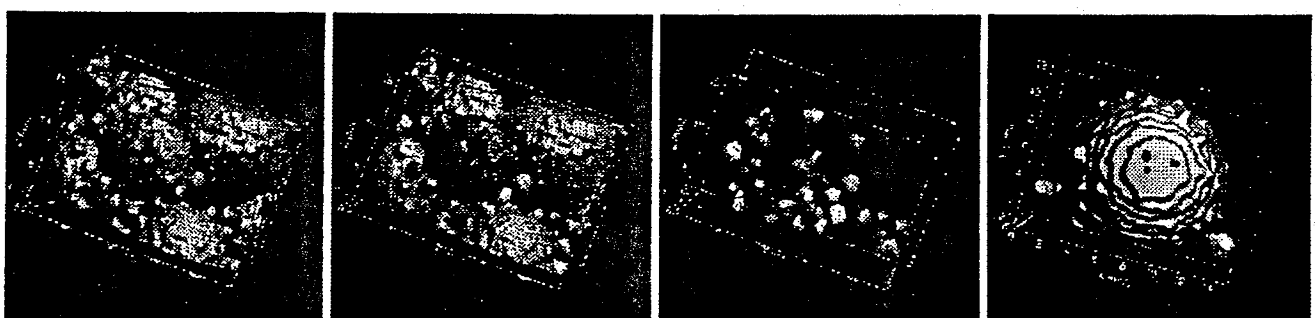


Figure 17. A series of segmentation steps for the large sphere component of G3. A typical starting situation is shown in the first picture, in which a large volume contains many large and small spheres. The red frame indicates that the sides of the blob have been cut off. The second figure shows the result of an erode/dilate operation, resulting in a high degree of separation. The third figure shows the result after all sub-blobs intersecting cut faces have been postponed, as well as all large spheres. These remaining small spheres could then be rejected en masse. As a result, later processing of one of the large spheres, as shown in the fourth figure, is greatly simplified.

The third component was relatively straightforward to segment, as most objects with similar grayscales had already been processed. The steps taken were very similar to those for the large sphere component:

1) Remove separate component, infill with 0
2) Adjust Levels, Input 100, 1.00, 200
3) Gaussian3D, blurring radius: 4.5
4) Threshold, 178-255, expand by 2 into 131-255

Note that the threshold levels used can be directly compared to those used for the second component, as the grayscales were shifted using the same parameters. The included grayscales are slightly lower, owing to the spheres being slightly darker. Intermediate and final steps are shown in Figure 18.

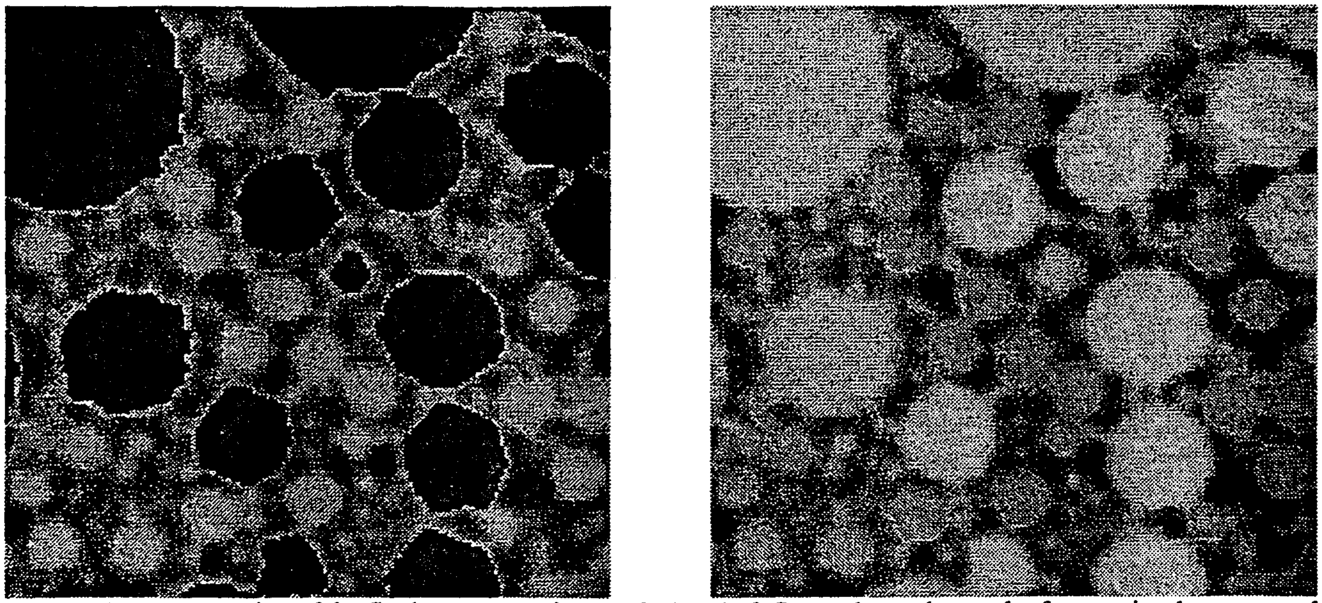


Figure 18. Segmentation of the final component in sample G3. Left figure shows the result after previously segmented components have been removed. Right figure shows final segmentation superimposed over original data.

The separation of the final component was straightforward, but also quite time-consuming, as over 6,000 spheres were present in the data volume, and each had to be inspected and approved by the operator. A small number of the smaller spheres were recognized and removed.

In all, the processing identified and measured 8343 spheres, and required approximately 21 hours to complete. Thus the time required was approximately 9 seconds per sphere, although different sphere populations required different amounts of processing. This compares very favorably with the PIs' previous experience using 2D-based tools to process CT imagery of crystalline rocks, for which a similar number of crystals would have taken weeks to process fully.

ASSESSMENT OF NEED FOR FUTURE WORK

While it has been demonstrated that the Blob3D software is capable of producing high-quality data, there remains room for improvement in allowing it to better handle the variability that will be present in asphalt concretes, and for making its use easier, faster, and more efficient. A net processing time of approximately 9 seconds per object in a data volume represents a vast improvement over anything previously possible, but is still rather slow if a large number of cores are to be processed. While increased user familiarity with the program should lead to more efficient processing, this may be offset by the increased difficulty of processing cores with irregular and inhomogeneous rocks fragments. Desirable program improvements would include implementing an Undo capability in the Segment module, and enabling automatic acceptance and/or rejection of blobs or sub-blobs according to user-selected measurements. In addition, more sophisticated methods for data processing and thresholding would likely result in substantial time savings. In the longer term, it would be desirable to progressively automate the processing so as to minimize the need for human intervention. This would require considerable work, and much experience dealing with natural materials, to be able to establish the kinds of criteria automated processing would require.

PLANS FOR IMPLEMENTATION

The next stage of development of this technology will take place at the Turner-Fairbank Highway Research Center of the FHWA, where an industrial CT scanner has been procured and installed in order to begin systematic investigation of core samples from experimental mixers and from field tests, such as WesTrack.

CONCLUSIONS

This project has created the technical capability to revolutionize the way in which asphalt concrete pavements are evaluated and designed. What is now required is a systematic study to establish a knowledge base for the optimal ways in which this technology can be used. A series of cores from field tests and experimental mixers should be scanned and processed to begin to develop this knowledge base. Important questions to be answered include:

- How can parameters obtained from this analysis best be related to pavement performance?
- Are there additional parameters that are obtainable from these data that would be of use?
- What characteristics of asphalt concretes can increase or decrease the quality of this analysis?
- What size classes of aggregate particles are necessary for obtaining useful predictive results?
- What is the resolution of scan data required to achieve the detail necessary?

Personnel at the Turner-Fairbank Highway Research Center of the FHWA have had the foresight to put into place the infrastructure necessary to commence such a study, having obtained an industrial CT scanner, a set of useful samples, and researchers with expertise in pavement design and mechanics.

APPENDIX: BLOB3D DOCUMENTATION

For version as of 7/27/00.

BLOB3D PROGRESS

The Blob3D Progress window guides the user through the Blob3D process in the correct order. A button will only be highlighted when that step can be taken. The first three steps instruct the program which files to use (Load Tiffs), how to reconstruct the images (State Voxel Dimensions), and of the components in the images (List Components). After listing the components they will appear listed in the Blob3D Progress window followed by the steps needed to process each component: Segment, Separate, and Extract.

Segment divides the image into two regions: those corresponding to the component and those not corresponding to the component. Only one component can be segmented at a time and it must be evaluated in Separate before segmenting the next component.

Separate will identify each three-dimensional continuous area of voxels (a blob) as allocated in Segment. Each blob must be evaluated to determine if it is a single object or if it is multiple objects. Multiple objects can be separated into individual objects. Sometimes the blobs do not represent the component, these voxels are removed from the component and sent back to be segmented with the next component.

Extract calculates the desired information from the objects identified. The information can be either about individual objects (e.g. volume distribution) or about relationships between objects (e.g. grain contacts). Extract can take one or all of the components separated into consideration.

File Menu

Save State

Saves the current stage of progress of Blob3D.

Load State

Loads a previously saved stage of Blob3D.

Preferences

The preferences window allows alteration of the size of the display windows for separation and segmentation. An example value is 400 pixels. The maximum volume of voxels to process in separate can also be changed. An example value is 500,000 voxels.

Load Tiffs

Select the folder containing the TIFF files to be processed. They must be in alphabetical order with the suffix ".tif".

List Components

Type the name of each component (e.g., matrix, garnet, pebbles, voids) that you will segment in a separate field box leaving unused fields blank. There is a maximum of 7 components and Component 0 is defaulted as the Matrix. After naming the components they will appear listed in the Blob3D Progress window followed by the three steps needed to process each component: Segment, Separate, and Extract.

State Voxel Dimensions

The inter-pixel spacing and inter-slice spacing must be entered for reconstruction of the CT slices. Inter-pixel spacing is calculated by dividing the field of reconstruction by the number of pixels in the field. The inter-slice spacing is the slice increment during scanning. This information is saved in the file containing the tiff files, so it only needs to be entered the first time a set of tiffs is loaded.

SEGMENT

Segment divides the voxels of an image into two regions (those corresponding to the component and those not corresponding to the component) based on grayscale value. This is accomplished by applying a filter to the image that results in a binary image. Filters can also be applied to reduce the amount of noise in the images. It is often easiest to segment the most distinct component first, such as the one with the highest grayscale value.

Segmentation Steps

The Segmentation Steps text box will list the filters in the order they are applied to the images.

Adding filters

Filters (discussed in more detail below) are added to the list of segmentation steps by using the drop list box and then selecting the desired filter. A window will then appear to input the parameters for that filter. The new filter will be added in the sequence of steps following the step that is highlighted at the time the new filter is added. To add a filter to the beginning of the sequence of segmentation steps, click on the "Deselect" button before adding the new filter. Remove Previous Components and smoothing filters (Adjust Levels, Median Smoothing, 2D Gaussian Smoothing, 3D Gaussian Smoothing, Edge Enhancement), which reduce noise in the image, can only be added before threshold filters (Grayscale Range, Expanding GS Range, Seeded GS Range) which produce binary images. The Removing Islands and Removing Holes filters can only be applied after threshold filters.

Removing filters

Filters can be removed from the list of segmentation steps by selecting the filter to remove and then clicking the "Remove" button.

Editing filter parameters

The parameters of a particular filter can be edited either by double clicking the filter in the list of segmentation steps, or selecting the filter in the list of segmentation steps and then clicking the "Edit" button. The order of the filters in the list of steps can be changed by selecting the filter to be moved and then clicking on the up or down arrow button to move it to the desired location.

Display

The display button will display a sub-volume of the data set. If there aren't any segmentation steps or none are selected, the display button will open a window showing the original data. If a segmentation step is selected, the display button will open a window showing the data after being processed by the first step through the step that is selected. A window showing the original data will also be opened if it has not already been displayed. More then one processed window can be displayed on the screen at the same time. If the parameters of a filter are changed, the filter will reflect the changes after the changes have been accepted. Subsequent segmentation steps will also be automatically updated.

Processed windows will automatically reflect changes to the parameters of the filter as they are being made if the interactive display button is checked.

Changing the Sub-Volume

To view the slices of the sub-volume displayed, click on the triangle button located in the bottom left of the segmentation window. To adjust the sub-volume displayed, click the "Change" button. The "Full", "Half", "1/3", and "1/4" buttons will automatically change the slice numbers to display that fraction of the data set. The slice numbers displayed can be changed manually by typing the desired numbers into the text field boxes.

Filters

Adjust Levels

The Adjust Levels filter increases brightness and contrast of the image by revising the range of grayscale values. The parameters window displays a histogram of the original grayscale values and three field boxes for the Input Levels entered by the user. The first value is the grayscale value in the original image that will be set to 0 in the processed image. The second value determines the midpoint of the revised histogram. The third value is the grayscale value in the original image that will be set to 255 in the processed image. After each value is entered a red line will appear on the histogram showing the position of the value entered relative to the original grayscale histogram. The Adjust Levels filter will appear in the segmentation steps as "Adjust Levels, Input: lowervalue#, midpoint#, uppervalue#."

Median Smoothing

The Median Smoothing filter is used to reduce noise in the images with minimal alteration of the geometry of the objects. The Median Smoothing filter replaces a voxel with a grayscale value equal to the median of the grayscale values of the voxels around it. The median is determined from a roughly square box centered on the original voxel with an edge length of ($2r+1$), where r is the radius chosen by the user. The Median Smoothing filter be repetitively applied to the image by change the "Times to repeat" scroll bar. The larger the edge length and times repeated, the more the noise will be reduced and the more the image will appear smoothed. The Medain Smoothing filter can be applied in three dimensions by checking the "3D" box in the parameters window. The filter can only be applied in three dimensions if the radius of the examination box is larger then the aspect ratio. The Median Smoothing filter will appear in the segmentation steps as "Median, radius: *r*, reps: #, 3D", where *r* is the chosen radius value.

Gaussian filters

Gaussian filters can be applied in either 2 dimensions or 3 dimensions. They reduce noise in the image much quicker than the Median Smoothing filter, but may be more likely to alter the geometry of the object. Gaussian filters have weights specified by the probability density function of a Gaussian, or Normal, distribution. The filter is applied over an area determined from the pixel length entered by the user. The larger the blurring radius, the more the noise will be reduced. The Gaussian filters appear in the segmentation steps as "Gaussian2D, blurring radius: #", where # is the pixel length chosen.

Edge enhancement

The Edge enhancement filter enhances the edges of the objects by lightening the edges while darkening the rest of the image. The parameters window allows the user to adjust the degree of the edge enhancement as well as the examination area over which the enhancement is applied through two scroll bars. The higher the Degree entered, the more enhanced the edges will appear. The higher the radius, the greater the area examined to apply the enhancement. The Edge enhancement filter will appear in the segmentation steps as "Sharpen, deg: #; rad: *r*", where # is the degree of enhancement and *r* is the radius of the examination box.

Remove Separated Components

Remove Separated Components replaces voxels allocated to previously separated components with a grayscale value chosen by the user. It is often beneficial to choose a value similar to the matrix value or component not being segmented.

Grayscale Range

This filter allocates voxels to the component if they fall within a specific threshold grayscale range and excludes them if they do not. The user inputs the bottom and top of the range either by entering them in the field boxes or by moving the scroll bars to the desired value.

Because CT is inherently blurry, the boundary between an object and its surroundings is always gradational. Thus there will be a drop-off in grayscales at the edge of an object. The grayscale that will result in the most accurate depiction of an object is midway between the average grayscale of an object and the average grayscale of what surrounds it. For example if an object has a grayscale of 200 and the surrounding material has a grayscale of 120, then the correct value is 160.

The Grayscale Range filter will appear in the segmentation steps as "Threshold, bottom# - top#", where top# is the value for the top of the range and bottom# is the value for the bottom of the range. The Grayscale Range filter will produce a binary image where white corresponds to voxels that will be designated to the component, and black corresponds to voxels that will be segmented in the component.

Expanding GS Range

The Expanding Grayscale (GS) Range begins by allocating voxels within a threshold grayscale range. It then will expand to continuous areas of voxels by an amount specified by the user as long as the expanded voxels are within additional bounds set by the user. The purpose of this filter is to allow easy segmentation of objects that have average grayscale ranges that are distinct from their neighbors but the values of their margins overlap those of their neighbors.

The filter will appear in the segmentation steps as "Threshold, bottom# - top#, Expand by # into bottom# - top#." The Expanding GS Range filter will produce a binary image where white corresponds to voxels that are designated as the component while black corresponds to voxels that are not part of the component.

Seeded GS Range

The Seeded Grayscale (GS) Range is similar to the Grayscale Range except it takes into account a "seed" grayscale value. A voxel to be allocated to the component must be within the threshold range and be connected to a voxel with a value within the "seed's" grayscale range. The seed cannot have a lower value than the threshold range. The user must input the top and bottom of the threshold range, as well as the top and bottom of the "seed" range. The filter will appear in the segmentation steps as "Seeded Threshold, bottom# - top#, seed:bottom# - top#." The Seeded Grayscale Range filter will produce a binary image where white corresponds to voxels that are designated as the component while black corresponds to voxels that are not part of the component.

Remove Islands

The Remove Islands filter will reduce or remove small areas of voxels designated to the component (white) that are surrounded by areas of non-component voxels (black). To find the islands, an area within a specific box size is examined. The user inputs the length of the edge of the examination box. A larger edge length will remove larger islands then a smaller edge length, but a large edge length might also smooth the edges of the component that is not removed. The Remove Islands filter will appear in the segmentation steps as "Remove Holes, edge: #", where # is the edge length.

Remove Holes

The Remove Holes filter does the opposite of the Remove Islands. It removes or reduces small areas of non-component voxels (black) from areas surrounded by component voxels (white). To find the holes, an area within a specific box size is examined. The user inputs the length of the edge of the box. A larger edge length will remove larger islands then a smaller edge length will, but a large edge length might also smooth the edges of the component that is not removed. The Remove Islands filter will appear in the segmentation steps as "Remove Islands, edge: #", where # is the edge length.

Segmentation completion

When the all of the appropriate filters and parameters have been chosen, click the "OK" button at the bottom of the Segmentation window to segment the entire data set. Alterations to the segmentation steps can be made by clicking the "Re-Segment" button in the Blob3D Progress window.

DISPLAY WINDOWS

The images can be displayed in four different modes: ortho Z, ortho Y, ortho X, or 3D. These are selected from the view mode drop box menu.

Orthogonal views

The orthogonal views show the data normal to the selected axis, either X, Y, or Z. Different slices can be viewed by inputting the selected slice number in the box or using the scroll bar. The scroll bar also gives the amount of millimeters the current slice is from the bottom of the data set. The image can be magnified by selecting the desired magnification from the Zoom drop text box and then clicking on the image where the center of the magnification is desired. To determine the location and grayscale value of a voxel, click on the desired voxel in the display image. The location and grayscale value will then display in the dialog box in the bottom left corner of the window.

3D view

Isosurface

The three-dimensional surface of the blobs is constructed by connecting voxels of the same grayscale value, producing an isosurface. This grayscale value can be selected by clicking on the histogram of grayscale values, or typing the desired value in the "Grayscale" box. The isosurface has four display options: None, Wire, Smooth, and Points. The isosurfaced image can be rotated by holding down the left mouse button on the isosurfaced image and moving the mouse in the direction desired to rotate the image. The frame and axes can be removed from the image by unselecting the appropriate box in the bottom left of the window.

3D Slicing

Two-dimensional slices of the original image can be superimposed on the isosurfaced image. To do so, select the box next to orthogonal view desired. To change the slice number, move the scroll bar located to the right of the view direction.

Capture Original

When a processed data window is opened it automatically "captures" the original data window. That is, navigating, changing views, and zooming in the processed window will also be done in the original data window. If more than one processed window is opened, the original data window will only mimic the processed window which currently has it captured. Clicking on the "Capture Original" button in the bottom left of the processed window will cause that processed window to capture the original. To release the original data window, click the "Release Original" button. The original data window will then return to its settings (view, slice number, magnification) before being captured.

Show in original

The "Show in original" check box will display the pixels allocated to the component superimposed in pink on the original image. This option is only available with binary outputs.

SEPARATE

Separate looks for three-dimensional continuous areas of voxels (a blob) to be evaluated by the user. The blob might represent a single object or multiple objects. Blobs representing multiple objects can be separated into the actual number of objects (sub-blobs) by either an erosion/dilation method or by cutting the object with a plane. Due to noise, some blobs do not represent the component being analyzed (such as a few voxels of matrix), these voxels can be removed from the current component to be analyzed again in the next segmentation.

Status and Display

The volume, surface area, and sphere-normalized surface to volume (SV) ratio of the blob are given in the Status text box. This information will always reflect the original blob no matter how many objects it is separated into (sub-blobs). The Sub-blob line in the status box will always reflect the current number of sub-blobs.

Separation of blobs into individual sub-blobs is an interactive process with the display windows, so the segmented data needs to be displayed before beginning to separate the blobs. To display the image of the current blob being evaluated, click the "Segmented Data" button. The display window is set up with the same options available during segment. Sometimes a large number of objects are connected resulting in a volume of continuous voxels too large for Separate to process at one time. In such a case, the axis and faces of the display area will be marked with red lines. The red lines represent areas where the blob continues beyond the boundaries seen on the display.

The "Original Data" button displays the unprocessed grayscale image corresponding to the segmented data. These windows will automatically refresh after all the sub-blobs have been accepted/rejected to show the next blob to be separated.

Separation Method

Choose the method used to separate a single blob into separate sub-blobs from the Separation Method drop list box. There are three methods to choose from: Manual, Primitive Fitting, and No Separation. Manual separation allows each blob to be evaluated by the user and then separated as required. Primitive Fitting separation has not yet been implemented. No Separation will process all the blobs as single objects.

Manual Separation

Erosion/Dilation

Erosion is a process that incrementally decreases the volume of the blob from the outside inward. After one or more erosions, a blob representing multiple objects might no long be connected. Each resulting separated object (sub-blob) will be represented by a different color. At this point the voxels removed during erosion can be added back on with out any distortion in the data through dilation. Erosion/Dilation can be implemented as single steps or as multiple steps. With the single step method, press the "Erode" button to erode one layer of voxels. The "Erode" button may be pressed as many times in as needed. After the blobs have eroded sufficiently to separate the blob, press the "Dilate" button once to add all the removed voxels back. The Multiple Steps method is quicker to use. The numbered buttons correspond to the number of erosions that the program will automatically execute on the blob. It will also automatically dilate after the final erosion.

While the shape and area of the blob will not be altered during Dilation, the program does have some discretion in assigning which sub-blob the eroded voxels belong. To help assure an accurate separation, it is best to erode in the least amount of steps as possible.

Plane Definition

The Plane Definition uses a plane to separate the blob into two sub-blobs. The plane is based on three points picked by the user. To choose the points, the "Segmented Data" display must be opened. Then click the "Pick 3 Points" button and click on the blob image at the desired location of the three points. The points must be either on the blob or on the axis. It is advantageous to pick three points that are not close together. So if two points are chosen in a small area, it is helpful to choose the third point on the axis. The blob can also be rotated during picking by clicking on the black background. If a mistake is made, click the "Cancel Picking" button. The plane can be erased by pressing the reset button or by picking three new points.

Acceptance

Every blob must be accepted or rejected as part of the component. If a blob does not need to be separated or if all of the sub-blobs are part of the component, click the "Accept All" button. If the blob is not part of the component being analyzed, click the "Reject All" button to allow these voxels to be available to segment with another component.

Sub-blobs can be individually accepted or rejected by double clicking on the sub-blob in the display window. A "Process Sub-Blobs" window will appear in which the sub-blob can be individually accepted or rejected by pressing the appropriate button. To close the window, click "Cancel". This window also displays the volume, surface area, and sphere-normalized surface to volume ratio of the sub-blob. "Postpone All" saves the blob to be evaluated at a later time. If a blob is cut because its volume is too large to be displayed, the axis and faces of the display will be marked with red lines. After separating such a blob into sub-blobs, the blobs touching the edges should be postponed since it might continue in a part cut off by the size limitation. Later the sub-blob can be more accurately evaluated. To quickly postpone all sub-blobs intersecting a cut face, press the "Postpone All on Cut Face" button.

Example of processing a large blob: First use the erosion/dilation method to separate the maximum number of objects correctly. Next, use the "Postpone All on Cut Face" button to remove those sub-blobs that intersect the cut faces. A number of different paths can be taken from here. If all the objects have been separated, click "Accept All." A sub-blob that needs to be separated further can be postponed so the remaining can be quickly accepted. Individual sub-blobs can also be accepted or rejected by double-clicking on the sub-blob and choosing the appropriate button. Erode/dilate will erase a previously defined plane definition and vise versa. So, if a sub-blob needs to be further separated it either needs to be postponed or the other sub-blobs in the display need to be dealt with first.

Progress

The Progress text box keeps a record of the blobs processed. The "Volume traversed: # %" shows the percentage of the volume that has been examined in separate. The volume sample volume is examined systematically from one end to the other until it finds a blob. Since blobs occur across a number of slices and many are often connected, this number does not represent the percentage of blobs separated. "Items separated: #" shows the number of sub-blobs accepted so far.

Separation Completion

When all of the blobs have been analyzed in separate, a dialog box will appear saying "Finished Separating *'Name'* Component". To continue, click the "Done" button. You will then be returned to the Blob3D Progress window where the component can be extracted or a different component can be segmented.

EXTRACT

Extract calculates the desired information from the objects defined in separate. The information can be calculated for a single component or for multiple components. There are two types of calculations, those done taking only a single object into consideration and those taking the relationship of objects into consideration. Once all the options have been chosen, click the OK button. To postpone extract till a later time, click the Cancel button.

Components

Check the components to be considered in the calculations. Only the separated components can be considered. Components used with the individual object parameters can be different then those used with the contact information.

Individual Object Parameters

Parameters to extract

Simply check the parameters you wish to be calculated:

- Center Position finds the center of each object.
- Volume calculates the volume of each object.
- Surface Area calculates the surface area of each object. The surface area calculated here will be slightly higher than the actual surface area of the object due pixelation.
- Sphere-Normalized Surface to Volume Ratio calculates the surface area to volume ratio normalized to a sphere. So an object that is a perfect sphere will have a sphere-normalized surface to volume ratio of 1. The more an object deviates from a sphere, the more the ratio will deviate from 1.
- Aspect Ratio calculates the inter-slice spacing divided by the inter-pixel spacing.
- Long Axis Orientation gives the direction of the long axis for each object.
- Long Axis Length gives the length of the long axis of each object.
- Roughness
- Side Contact will show if the object is in contact with side of the sample volume. The Contact column in the spreadsheet will show a 1 if the object intersects the edge of the volume and 0 if it does not.

Contact Information

Contact Information calculates the position of the center of the area where two objects touch. It also gives the magnitude of the contact. Each contact is calculated twice since the program evaluates where object A touches object B and then where object B touches object A.

Saving Extract Results

Individual and contact information are saved in separate files. Type the base of the file name in the field box. "-Dat" will be appended to the end of the individual data file name, and "-Cont" will be appended to the end of the contact file name. To change the folder where the files are saved, click the change path button.