A Novel Interface for Higher-Dimensional Classification of Volume Data

Fan-Yin Tzeng Eric B. Lum Kwan-Liu Ma

Department of Computer Science
University of California at Davis
{tzengf,lume,ma}@cs.ucdavis.edu

Abstract

In the traditional volume visualization paradigm, the user specifies a transfer function that assigns each scalar value to a color and opacity by defining an opacity and a color map function. The transfer function has two limitations. First, the user must define curves based on histogram and value rather than seeing and working with the volume itself. Second, the transfer function is inflexible in classifying regions of interest, where values at a voxel such as intensity and gradient are used to differentiate material, not taking into account additional properties such as texture and position. We describe an intuitive user interface for specifying the classification functions that consists of the users painting directly on sample slices of the volume. These painted regions are used to automatically define high-dimensional classification functions that can be implemented in hardware for interactive rendering. The classification of the volume is iteratively improved as the user paints samples, allowing intuitive and efficient viewing of materials of interest.

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces—; I.2.6 [Artificial Intelligence]: Learning—Knowledge acquisition; 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—partitioning

Keywords: classification, graphics hardware, interactive visualization, multidimensional transfer function, neural network, user interface design, volume visualization

1 Introduction

Direct volume rendering has become a popular technique in visualization, which allows scientists to gain insights into their data sets through the display of materials of varying opacities and colors. Volumetric data sets often have a single scalar value per voxel, so classification of these voxels to assign color and opacity is critical in obtaining useful visualizations that help to provide understanding into a data set. Without a proper classification function to show interesting features or remove obscuring data, it is impossible to correctly interpret the volumetric content.

The transfer function is typically used to perform this classification. The traditional one-dimensional transfer function for volume rendering only considers a voxel's scalar value; that is, there is a direct mapping of transparency to scalar value. Designing this type of transfer function is often not intuitive for the users since they must work in some derived transfer function space (show in the bottom left of Figure 1), where the users might specify a curve defining a mapping between scalar values and opacity. In addition, the traditional transfer function is of limited effectiveness in performing the actual classification. For example, for the MRI head data set (also shown in Figure 2), it is difficult to specify a one-dimensional transfer function that differentiates the brain and the region near the skull since the scalar values of the two regions are similar. As such, the user can only show both materials together as shown on the left of Figure 2, in which case the outer layer might obscure the brain material of interest. The user could reduce opacity of the outer layer to make the brain more visible, but the brain would simultaneously become more transparent and difficult to see as shown on the right of Figure 2. The user often needs to iteratively try to find a transfer function that is a compromise between the two, as shown on the left of Figure 1 which still has some material obscuring the brain, with the brain to some degree transparent as well.

Recently, there has been research into the use of 2D and 3D transfer functions which take more information into account such as first- and second-order gradients [Kindlmann and Durkin 1998; Kniss et al. 2001]. An example of a visualization of the brain using a 2D transfer function and its interface are shown in the center of Figure 1. The additional gradient information allows more refined classification, and works well when a user wants to visualize the boundaries between different materials. However, other properties, such as textures and position, are not taken into account. It is our belief that these additional properties should be used to perform better classification. The challenge of using these additional neighborhood properties is that the complexity of transfer function specification grows significantly beyond two dimensions.

In our work we describe a new method for specifying high-dimensional classification functions that consists of the user simply painting on a few slices from the volume data sets. The user is given full control of what materials they want to classify by applying one color paint to parts of the volume they want to see, and another color paint to regions they do not want to see. Abstracted from the user is the generation of a high-dimensional classification function using artificial neural networks. The network uses the painted regions as training data to 'learn' a classification function that maps voxels into uncertainty of whether the given voxel is part of the material of interest. This uncertainty can then be mapped to opacity during rendering. The high-dimensional function used in our work uses as inputs a voxel's scalar value, gradient magnitude, the values of its neighbors, and its x, y, z location. The scalar value is fundamental information directly from that voxel, its neighboring values provide information that can be incorporated for texture, while position can be used to take into account a material's structural properties. Two materials might have similar scalar value, but they are much less likely to also have similar gradient magnitude, texture and location. Thus by using a high-dimensional classification function, it is possible to better differentiate the materials for visualization.

All stages of the method described in this paper are fully in-

IEEE Visualization 2003,
October 19-24, 2003, Seattle, Washington, USA
0-7803-8120-3/03/$17.00 ©2003 IEEE

Proceedings of the 14th IEEE Visualization Conference (VIS'03)
0-7695-2030-8/03 $ 17.00 © 2003 IEEE

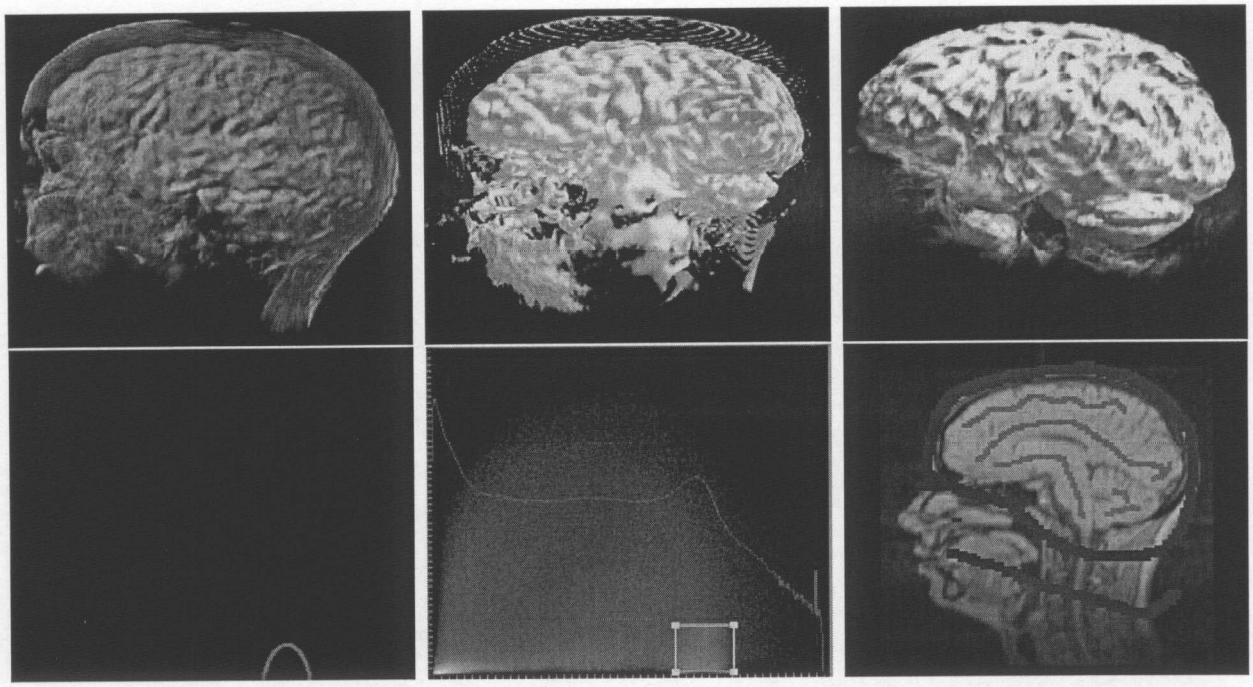

Figure 1: Left: Volume rendering of MRI head with a 1-D transfer function. Middle: 2-D transfer function. Right: 10-D classification function.

teractive including the implementation of a hardware-accelerated, neural-network-driven volume renderer. The user is able to paint on the volume, immediately see the results of the network as it trains, and continue to paint to provide additional training data to steer the network toward achieving a classification function meeting the user's visualization objective.

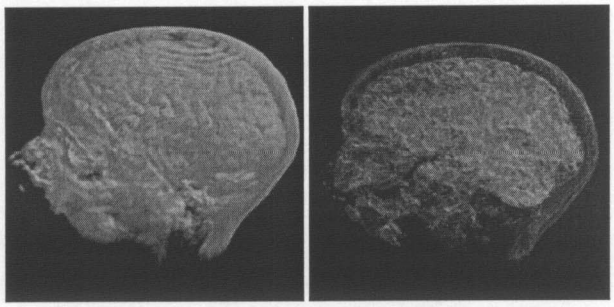

Figure 2: Visualization of an MRI head data set. The left image shows the result of a one-dimensional transfer function where the brain and the region near the skull are both shown. Since the scalar values of the two materials are very similar, the outer layer obscures the brain. The right image shows the result of reducing the opacity of the outer layer, reducing the opacity of brain at the same time.

2 Related Work

There has been a great deal of research devoted to the generation of transfer functions for volume visualization [Pfister et al. 2001]. Fujishiro et al. [Fujishiro et al. 1999] use topological information from a hyper-Reed graph to derive transfer functions. Bajaj et al. [Bajaj et al. 1997] present techniques for capturing isosurfaces of interest. He et al. [He et al. 1996] use generic algorithms to breed trial transfer functions. The user can either select functions from generated images or allow the system to be fully automated. Marks et al. [Marks et al. 1997] address parameter selection problem in general by rendering a multidimensional space of those parameters. The user then navigates this space, in the context of volume visualization, to choose appropriate transfer functions. Jankun-Kelly and Ma [Jankun-Kelly and Ma 2001] present automated methods for generating transfer functions to visualize time-varying volume data.

Levoy [Levoy 1988] shows how to use gradient magnitude to enhance material boundaries in volume data [Levoy 1988]. Konig and Groller [Konig and Groller 2001] introduce a user-interface paradigm with a set of specification tools assisted with realtime volume rendering to make it easier for the user to select transfer functions. Kindlmann and Durkin [Kindlmann and Durkin 1998] suggest that by looking at a two-dimensional scatterplot of data values and gradient magnitudes, which is 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. 2001] extend this 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 (i.e., the volume data space and the transfer function space) interaction they described allows the user to specify regions in volume space and immediately see the resulting regions in transfer function space. This interaction helps incorporate the user's spatial understanding of the volume data into the transfer function space. Huang and Ma [Huang and Ma 2003] present a technique that can suggest a 2D transfer function by using the results of partial region growing from a point selected in volume space. Our technique eliminates the transfer function space entirely, replacing it with a volume space painting interface.

Artificial neural networks have received a lot of attention in the

Proceedings of the 14th IEEE Visualization Conference (VIS'03)
0-7695-2030-8/03 $ 17.00 © 2003 IEEE

field of biomedical imaging, especially to assist in image segmentation tasks [Cloete and Zurada 2000; Duch and Jankowski 1997; Gelenbe et al. 1996a; Gelenbe et al. 1996b; Hall et al. 1992; Mitchell 1997; Perlovsky 2000]. Our work employs a neural network coupled with an interactive user interface and a hardware-accelerated volume renderer for classification and visualization of biomedical volume data.

3 Our Method

Our work provides a painting user interface which allows the user to easily specify the region of interest, and uses high-dimensional classification functions to classify the volume. Figure 3 shows the overall process of interactive classification and visualization. The portion most visible to the user is the painting-based interface used to collect sample points of the region he or she wants to see. The user starts by painting sample points on a slice of the volume. These painted sample points are fed to the neural network where training begins to produce the network.

Training is an iterative process, with the user able to interactively view the results from the current network on classified slices and volume is in real-time. The user can use this feedback to further revise the painting and add additional training data so that the resulting network leads to their desired classification.

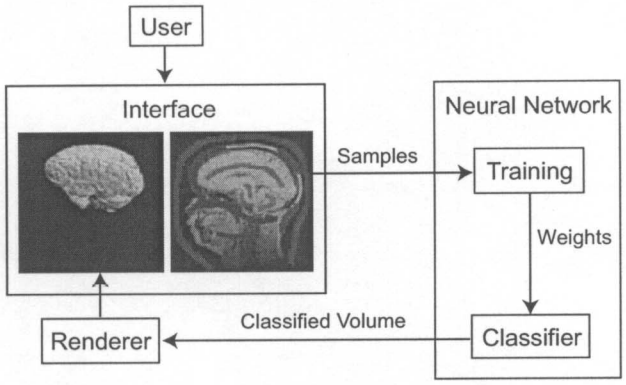

Figure 3: The visualization process. The painting user interface gives the user direct access to the volume data and allows the user to indicate the region of interest. "Samples" are the points in painted regions and are also the input to the neural network. The neural network trains using the samples to obtain a set of weights. By applying the trained network to a volume, the user can classify and then render the volume. The resulting visualization is shown to the user so that paint can be added or removed to refine the classification until a satisfying result is obtained.

3.1 User Interface

In traditional volume rendering interfaces, a user requires a certain level of understanding of the intensity distribution of a data set and familiarity with the user interface to make an acceptable transfer function. When a user tries to visualize a new data set, a great deal of time might be spent experimenting with different mapping functions for that data set.

The goal of our user interface is to provide a means for the user to partition a data set into different material classes. The user specifies membership into a material class by painting on slicing planes using two different colors. One color is applied to indicate example regions that are part of the material class, while the other color is used to indicate regions that are not part of the material class. For example, the user might apply red paint to the brain region of a slices of an MRI scan to indicate it is a material of interest, and blue in other regions to indicate otherwise.

The painting user interface is intuitive to the user since it allows the user to work directly with the volume data, rather than a derived transfer function space. The user is also provided a set of familiar painting user-interface tools which include brushes and erasers of varying sizes. There are x, y and z axis-aligned slicers for the user to view and paint on the volume.

The user does not need to paint on all slices of the volume, but instead must paint in some areas of a couple of slices to classify the entire volume. The required number of painted voxels varies depending on the data set. When using position information, the painted voxels need to be more spread out across different slices in order to provide general samples for training. Thus, real-time visual feedback must be provided such that the user can look at the resulting slices or volume to find the regions that are not well-classified, and quickly go to the corresponding slices to paint more sample points to improve the classification.

When painting on a slice, the corresponding painted regions are also shown in the other two slicers, with all painted sample points recorded in a table that is used to train the neural network. For scalar data, the sample point data includes the scalar value of the voxel, gradient magnitude, the scalar values of its neighbors, and the position of this voxel, and a value that indicates membership into a material class. For color data, the sample point data includes the R, G, B values of the voxel, values of its neighbors, and position. Therefore, when a user chooses to visualize a color data considering 12 neighbors, the dimensionality of its classification function is $3+12\times3+3 = 42$.

3.2 Artificial Neural Network

An artificial neural network is an intelligent system that acquires knowledge through a training process and applies the knowledge to solve similar problems. It is powerful because it can learn both linear and non-linear relationships between inputs. Figure 4 shows the structure of an artificial neural network.

Each connection between neurons has a weight, with the weights modulating the value across that connection. Training is the process of modifying the weights until the network implements a desired function. To train a network, a set of training inputs and desired outputs are required. At the beginning, the weights are set at random, and are iteratively modified to obtain a network which minimizes the error at the output for the training data. Once training has occurred, the network can be applied to data that was not part of the training set.

The neural network topology we use is three-layer perceptron, and it is trained with the Feed-Forward Back-Propagation Network (BPN) algorithm. The back-propagation algorithm, which is designed for supervised training, was introduced by Paul Werbos [Werbos 1974], and has been widely used since the work of Rumelhart and McClelland [Rumelhart and McClelland 1986].

A back-propagation neural network consists of at least three layers: an input layer, at least one hidden layer, and an output layer. The structure of a neural network is shown in Figure 4. In this example, there are m inputs in the input layer, n hidden nodes in the hidden layer, and one output in the output layer. Neurons are connected in a feed-forward fashion, and every neuron in the input layer is connected to all neurons in the hidden layer, similarly, hidden nodes are fully connected to the nodes in the output layer.

The input data to the neural network in our work is a set of vectors where each vector contains information such as the voxel's value, gradient magnitude, values of its neighbors and position. The output data is a number between zero and one which indicates if a

Proceedings of the 14th IEEE Visualization Conference (VIS'03)
0-7695-2030-8/03 $ 17.00 © 2003 IEEE

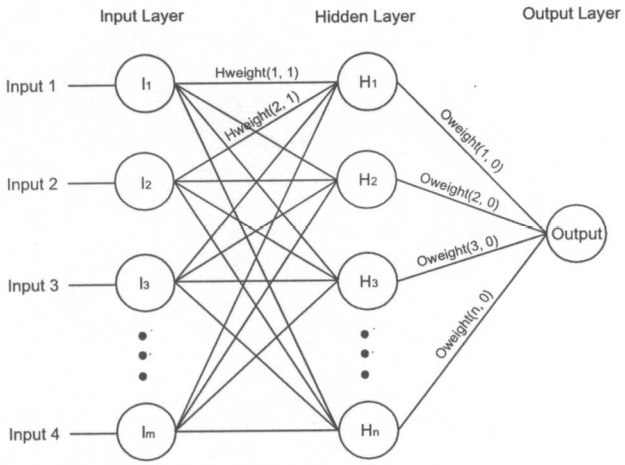

Figure 4: Structure of an artificial neural network with m inputs, n hidden nodes and one output.

voxel is a member of a material class. The input vector is propagated forward through the network, influenced by the weights of each connection between neurons from different layers. The output of the back-propagation network is compared with the desired output and an error value is calculated. The error is backpropagated to the inputs and used to adjust the weights to better match the desired output.

The implementation of a backpropagation neural network requires a non-linear transformation, and the sigmoid function $f(x) = 1/(1+e^{-x})$ is a typical one used in neural network for non-linear mapping.

In order to perform as much training as possible while maintaining interactivity, the incremental training process occurs in the system idle loop. After each training iteration, a classified slice is shown so the user can determine whether the network performs satisfactorily in that slice region or if more paint should be applied to supply additional training data.

In our work, the training of the network is in real-time. For example, Figure 5 shows the iterative refinement of neural network weights over a period of three second in one second increments when applied to the MRI Head data set. The image on the left shows the result of the network without any training, while the figure on the far right shows the solution after three seconds of training. If additional paint samples were added, the training algorithm would use this data to refine the previous network.

Classification of an entire volume can be performed in hardware during rendering as described in the next section, or in software. Applying the network in software consists of feeding each voxel and its neighboring properties into the network to get an uncertainty value between zero and one. This uncertainty value can then be stored in a new classified volume that can be rendered using traditional volume rendering methods in either software or hardware. The advantage of using software for classification is that it does not require any graphics hardware, the size of the network can be arbitrarily large, and classification is a preprocessing step that occurs once, rather than for each frame during rendering. The disadvantage of software is performance, where it can take up to 9 seconds to apply the network with 11 input nodes, 8 hidden nodes for classifying a 256×256×256 volume using a Pentium V 2.8 GHz computer, making it unsuitable for applications where the user must see the result of the trained network during training.

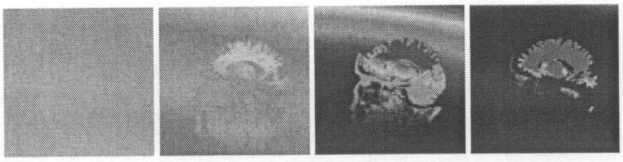

Figure 5: The user can immediately see the result of the iterative refinement of the neural network. This image sequence shows the progression of training results over a three second period.

3.3 Hardware-Assisted Neural Network and Renderer

It is very desirable that the user be able to quickly see the results of the neural networks training on the entire volume during the application of paint to steer the network to best classify the regions of interest. The application of the network to a volume in software limits this interactivity because of the sheer number of voxels typically involved. With this in mind, we have implemented the neural network in hardware and calculated the weights dynamically during rendering.

The neural-network volume renderer is implemented using a pixel shader, which permits the execution of assembly language instructions that are run on a per-pixel basis during the rasterization of a polygon. Rendering uses the standard technique of drawing a stack of view-aligned polygons that sample a 3-D texture [Van Gelder and Hoffman 1996], with the enhancement that a pixel shader is used that implements the neural network as these polygons are rasterized. We tested our implementation on a ATI Radeon 9700 Pro graphics card. Our pixel shader requires seven texture lookups for retrieving the center voxel values and its neighbors as inputs for the neural network. The position of pixel in 3-D volume space is simply derived from the texture coordinates of the pixel.

The same nearest neighbors used for neural network calculation are also used for calculating central differences for lighting. Thus, normal map textures are not required for lighting which permits the storage of significantly larger volumes on the graphics card.

While the hardware-assisted neural network is much faster than the software version, it is limited by the hardware capabilities. The size of the network implemented in hardware is restricted by the amount of data that can be passed to a pixel shader for the neural network weights, and the number of assembly language instructions that can be used in a pixel shader. Thus, our hardware implementation uses only up to eight hidden nodes, a limit that does not exist in our software implementation. It is desirable to use a network with a number of hidden nodes that is divisible by four since the highly-vectorized nature of hardware allows for the efficient execution of data in vectors of length four. The pixel shader itself can be implemented fairly efficiently with a series of vectorized MAD (multiply add) instructions, with additional instructions to compute the sigmoid excitation function. In our work, both software and hardware neural networks are implemented, and the users can choose one of the networks depends on the required network size.

Rendering performance is limited by the time required to rasterize the textured polygons to the screen, which is hampered by the necessary seven texture lookups and the length of the pixel shader. The amount of I/O sent to the renderer for each frame consists only of the newest neural network weights. The rendering of a 256×256×256 volume to a 512×512 window occurs at approximately 1.5 frames per second using the graphics card mentioned previously. Since the amount of data that must be sent to the renderer is low, rendering can be easily done asynchronously on a separate computer, where for each frame the rendering PC queries over a network the most recent set of weights (under one kilobyte

Proceedings of the 14th IEEE Visualization Conference (VIS'03)
0-7695-2030-8/03 $ 17.00 © 2003 IEEE

of data) from the PC where painting and neural network training occur.

3.4 Classifying Multiple Materials

The method described so far can be used to classify a single material in the volume. Often it is desirable to classify more than one object in a volume. For example, a user might want to show more than one material at a time, or a certain organ with high opacity value and other regions with low opacity to provide context. Our work is also able to handle multiple material classes as well.

To classify more than one material at a time, we use multiple networks. First, we classify a material by the method described in the previous sections. When adding another material class, a new neural network is created and used. For the second material, we also follow the same procedure as classifying the first material. Two sets of weights are generated separately by two different neural networks with two different groups of sample points. The results of all networks are used to volume render the classified volume.

Training is only performed on the newest material class. Therefore, when classifying a volume into multiple materials, the training time is the same as for single material classification since only one neural network is trained at a time. Rendering is accomplished using multiple passes, one for each of the material classes. The more expensive hardware neural network renderer pixel shader is used only when displaying the most recent material class, with the previously materials being rendered from precomputed volumes.

4 Results

The data sets we used to test our system are a $128\times128\times128$ MRI head data set, a $256\times256\times109$ MRI head data set with the skull partial removed, and a cryosection color brain data set resized to $256\times256\times256$.

We first applied our technique to the first MRI head data set which has brain material in the middle surrounded by materials of similar intensity near the skull.

In Figure 6 through Figure 8, the progression of a session of a user employing our technique is shown. The left image of Figure 6 shows a slice painted with pink representing the area the user wants to see and blue represents the area the user does not want to see. The middle image shows the result of the color-coded classification by the neural network. As indicated by the color bar, if a color of a pixel is closer to blue, the pixel is less likely to be part of the material of interest. If the color is closer to pink, the pixel is more likely to be part of the material of interest. When the user only paints on the empty region and the brain, the image shows the brain with pink, which indicates that the voxels in this region have very similar characteristics to the regions the user painted. The other regions of the head are shown in red to green since the user has not given input to classify them. The right image is the result of hardware-accelerated volume rendering for this classification.

When the user obtains a result containing unclassified areas, more painting is required. Figure 7 shows a painted slice, the classified slice, and the classified volume rendering after more paint has been applied. Most materials except the top of the head and brain have been removed. By continuing to paint on the top of the head, as shown in Figure 8, the brain is the only material remaining, with all other regions removed.

The left image in Figure 9 shows the results of the classification of multiple materials. The brain is rendered with high opacity value, and the skull and skin are rendered with a very low opacity so as not to obscure the brain. The right image shows the classification of the brain and the rest of the head without the skull and materials cover the brain.

Figure 10 shows another pair of examples of classifying multiple materials in the second MRI head data. The left image is the classified parietal lobe and cerebellum. It is difficult to classify those two materials using low-dimensional transfer functions because of the similarity of their scalar values. But when texture and position are taken into account, the two materials can be separated. The right image contains the parietal lobe and cerebellum rendered with high opacity, and the skull rendered with low opacity.

For color data sets, the classification function is only used to assign opacity since each voxel has an RGB color associate with it. Assigning opacity by specifying a three-dimensional transfer function for this type of data is made difficult by the traditional 2D interface and display. Our method, however, is able to handle color data sets in the exact same manner as scalar value data sets, where the user can paint on 2D slices that pass through the volume. The information used in classification includes the R, G, B color values of a voxel, the colors found in a voxel's six or twelve neighbors, and its position. That is, the dimensions of our classification function is either 24 or 42.

An example of classifying the cryosection color data set is shown in Figure 11. The left image is one of the photographic slices of the original data set. Each slice consists of the brain in the middle with surrounding white ice and a table the brain was placed on. Some regions in each slice show gaps in the brain which reveal deeper regions of the data set that are not part of the current slice. Thus, to visualize the brain properly it is necessary to remove the ice and surround regions, as well as those gaps in the brain, which often have a very similar color as the brain itself. Takanashi et al. [Takanashi et al. 2002] developed a method for specifying three-dimensional transfer functions that consists of transforming the RGB color values using independent component analysis into a derived ICA space that allows a transfer function to be specified as a series of 1D transfer functions aligned to each of the ICA axis. Their method simplifies the task of specifying RGB color transfer functions, but requires the user to work in a derived data space that is further from the original data. With our method the brain can be classified from its surrounding material but the user is able to work directly with the data, avoiding the requirement of specifying a three-dimensional transfer function.

The right image is the result of the classified brain generated by using the values of a voxel, six neighbors and the position with 20 hidden nodes when two cutting planes are applied to the volume so that the users can look at the inner structure.

5 Conclusions

An effective visualization is able to communicate information about those specific spatial structures that are of interest to the viewer, without the distraction of materials that are not of interest. Since the types of structures of interest vary widely depending on the user of system, user interfaces for classification must be powerful enough to provide high quality classification, yet intuitive enough to be accessible to a wide range of scientists.

For future work we would like to investigate the use of less computationally expensive excitation functions to compute the sigmoid excitation function which is used in hardware neural network for better performance during rasterization.

Other future work includes determining how our painting interface could be combined with the traditional 1-D or 2-D transfer function paradigm. One of the limitations of neural networks is that there is no intuitive relation between the numerical values of the weights used in a network and the resulting function the network implements. This makes it extremely difficult to make any direct changes to a network. The technique described in our paper could be adapted to generate traditional 1-D or 2-D transfer functions by using neural networks with one or two inputs, however

Proceedings of the 14th IEEE Visualization Conference (VIS'03)
0-7695-2030-8/03 $ 17.00 © 2003 IEEE

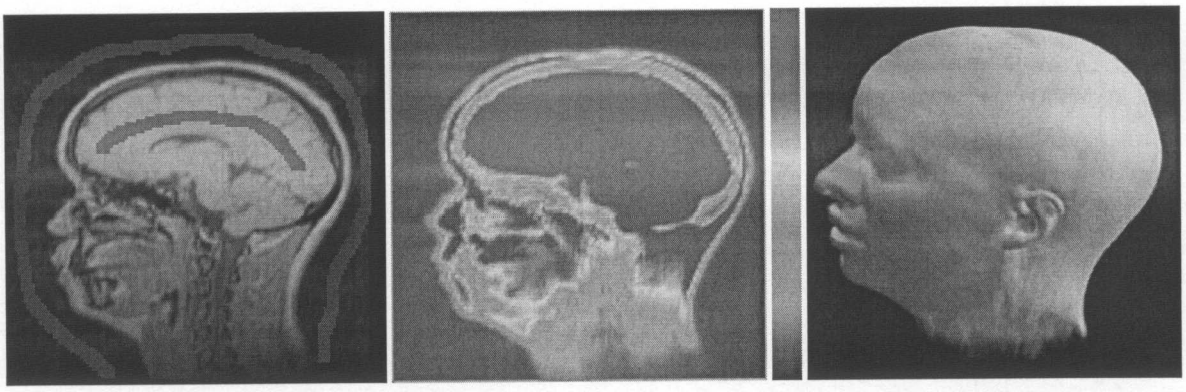

Figure 6: The left image shows a slice painted by a user where pink represents the material the user would like to see and blue represents the other materials. The middle image shows the result of classification with a color bar to its right. The right image is the rendered result of the classified volume.

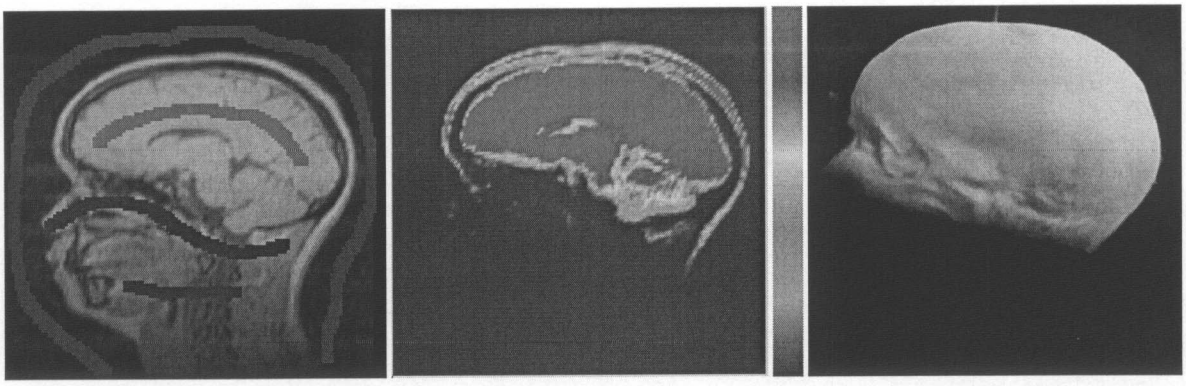

Figure 7: These images show the result after more painting information has been added. From left-to-right: A painted slice, the classified slice and the classified volume. Most materials except the top of the head and brain have been removed.

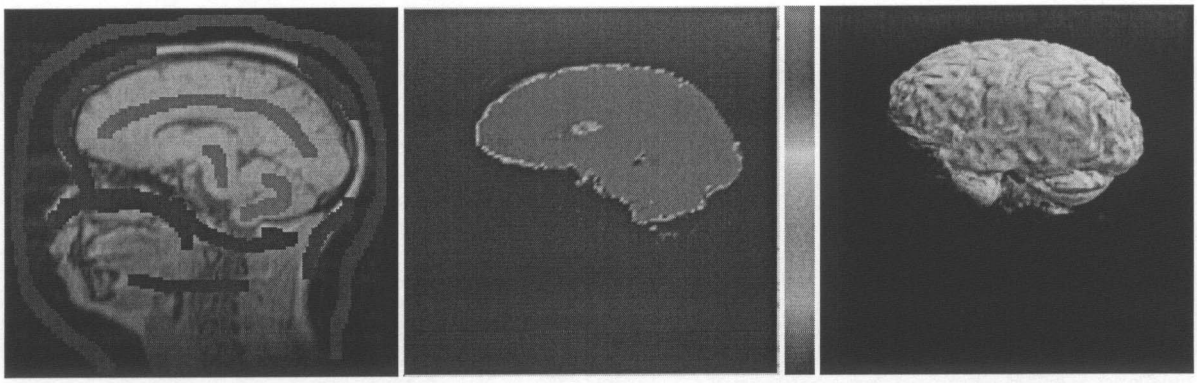

Figure 8: Additional paint is added to remove all regions except the brain. From left-to-right: A painted slice, the classified slice and the classified volume.

Proceedings of the 14th IEEE Visualization Conference (VIS'03)
0-7695-2030-8/03 $ 17.00 © 2003 IEEE

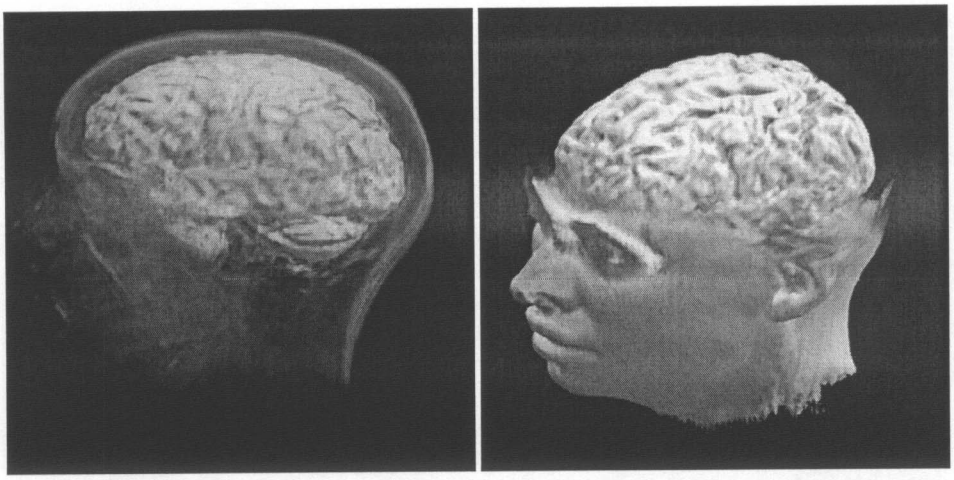

Figure 9: The results of the classification of multiple materials. Both images have two different classified materials. The left image shows the brain with a high opacity value and surrounded by the outer layers with a lower opacity. The right image is the brain and the bottom of the head.

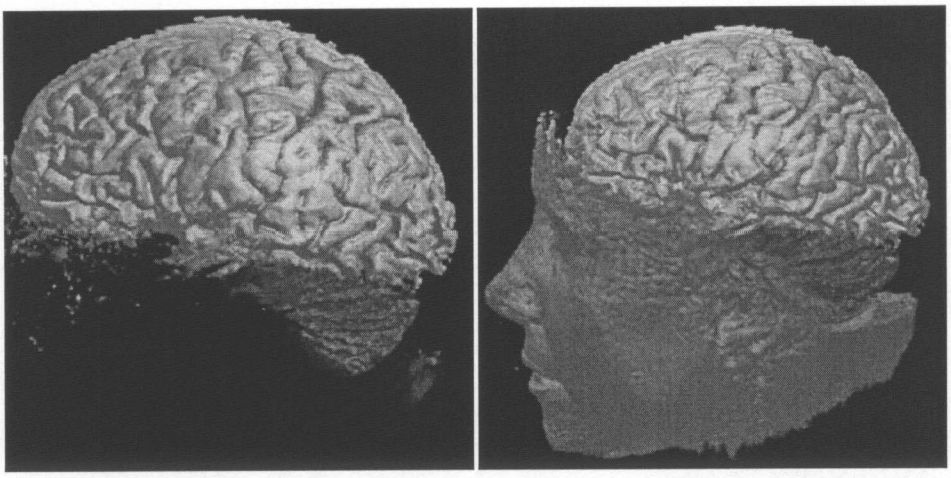

Figure 10: The left image is the classified parietal lobe and cerebellum of the MR head data set. The right image is the classified parietal lobe, cerebellum, and the bottom of the head with lower opacity.

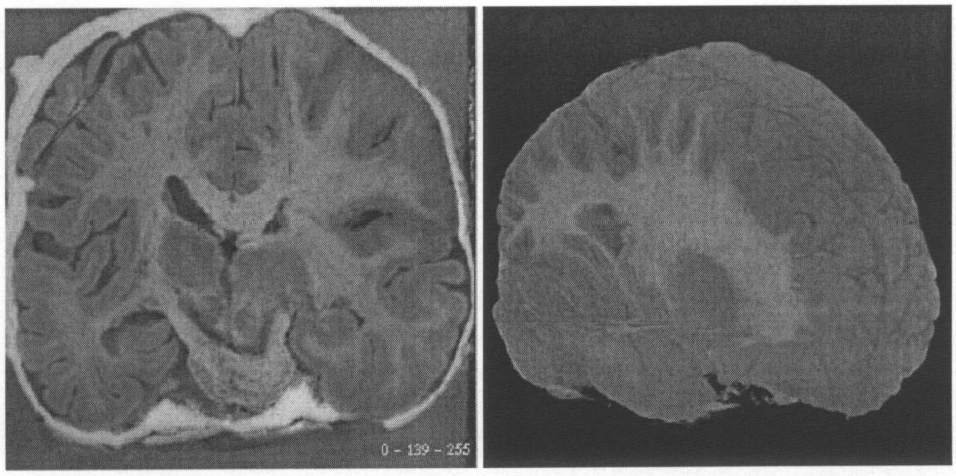

Figure 11: The left image is a slice of the original cryosection data set, and the right image is the classified result of the brain with a region cut away to reveal the inner structures.

Proceedings of the 14th IEEE Visualization Conference (VIS'03)
0-7695-2030-8/03 $ 17.00 © 2003 IEEE

much of the power of our method that comes from higher dimensional classification functions would be lost. Thus the challenge in this type of integration of our method and traditional transfer function techniques would be trying to maintain the high quality classification that comes from high dimensional classification with some of the interface characteristics of the traditional transfer function interface.

In this paper we have described a new type of volume visualization user interface that allows the users to specify which regions in a volume they would like to visualize by simply painting on a few slices from that volume. Abstracted from the user is a higher-dimensional classification function implemented using artificial neural networks that makes effective use of a voxel value, gradient magnitude, its individual nearest neighbors, as well as spatial position. Thus the new user interface is not only more intuitive than specifying a one-dimensional transfer function curve, it is also more powerful in that it uses far more information for classification to occur.

The rendering of the classified volume is implemented in graphics hardware, providing maximum interactivity, which is critical for giving users the ability to control which aspects of the volume they visualize. We believe more intuitive interfaces, like the one we describe, will make volume visualization more accessible to a wider range of scientists to meet their visualization needs.

Acknowledgments

This work has been sponsored in part by the U.S. National Science Foundation under contracts ACI 9983641 (PECASE award) and ACI 0222991; the U.S. Department of Energy under Memorandum Agreements No. DE-FC02-01ER41202 (SciDAC program) and No. B523578; the National Institute of Health through the Human Brain Project; and a United States Department of Education Government Assistance in Areas of National Need (DOE-GAANN) grant P200A980307. The larger MRI head data set was provided by Siemens Medical Systems, Inc. through the University of North Carolina at Chapel Hill. The cryosection data set was provided by Dr. Arthur Toga of UCLA Brain Research Institute. The authors would also like to thank members of the UCD visualization and graphics group for the valuable discussion and providing some of the test data sets.

References

BAJAJ, C. L., PASCUCCI, V., AND SHIKORE, D. R. 1997. The contour spectrum. In *Proceedings of IEEE Visualization '97 Conference*, 167–173.

CLOETE, I., AND ZURADA, J. M. 2000. *Knowledge-based neurocomputing*. The MIT Press.

DUCH, W., AND JANKOWSKI, N. 1997. New neural transfer functions. *Journal of Applied Mathematic and Computer Science*.

FANG, S., BIDDLECOME, T., AND TUCERYAN, M. 1998. Image-based transfer function design for data exploration in volume visualization. In *Proceedings of IEEE Visualization '98 Conference*, 319–326.

FRANKLIN, D. http://ieee.uow.edu.au/ daniel/software/libneural/.

FUJISHIRO, I., AZUMA, T., AND TAKESHIMA, Y. 1999. Automating transfer function design for comprehensible volume rendering based on 3D field topology analysis. In *Proceedings of IEEE Visualization '99 Conference*, 467–470.

GELENBE, E., FENG, Y., AND KRISHNAN, K. R. R. 1996. Neural network methods for volumetric magnetic resonance imaging of the human brain. In *Proceedings of IEEE*, vol. 84.

GELENBE, E., FENG, Y., AND KRISHNAN, K. R. R. 1996. Neural networks for volumetric MR imaging of the brain. *International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing* (August).

HALL, L. O., BENSAID, A. M., CLARKE, L. P., VELTHUIZEN, R. P., SILBIGER, M. S., AND BEZDEK, J. C. 1992. A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. In *IEEE Transactions on Neural Networks*, vol. 3.

HE, T., HONG, L., KAUFMAN, A., AND PFISTER, H. 1996. Generation of transfer functions with stochastic search techniques. In *Proceedings of IEEE Visualization '96 Conference*, 227–234.

HUANG, R., AND MA, K.-L. 2003. RGVis: Region growing based techniques for volume visualization. In *Proceedings of Pacific Graphics 2003 Conference*.

JANKUN-KELLY, T. J., AND MA, K.-L. 2001. A study of transfer function generation for time-varying volume data. In *Proceedings of Volume Graphics 2001 Workshop*, 51–65.

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

KNISS, J., KINDLMANN, G., AND HANSEN, C. 2001. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization '01 Conference*, 255–262.

KONIG, A., AND GROLLER, E. 2001. Mastering transfer function specification by using VolumePro technology. *In Tosiyasu L. Kunii, editor, Spring Conference on Computer Graphics 2001 17* (April), 279–286.

LEVOY, M. 1988. Display of surfaces from volume data. In *IEEE Computer Graphics and Applications*, vol. 8, 29–37.

MARKS, J., ANDALMAN, B., BEARDSLEY, P., FREEMAN, W., GIBSON, S., HODGINS, J., KANG, T., MIRTICH, B., PFISTER, H., RUML, W., RYALL, K., SEIMS, J., AND SHIEBER, S. 1997. Design Galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of SIGGRAPH '97*, 389–400.

MITCHELL, T. 1997. *Machine Learning*. McGrawHill.

PERLOVSKY, L. I. 2000. *Neural Networks and Intellect: Using Model-Based Concepts*. Oxford University Press.

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.

RUMELHART, D., AND MCCLELLAND, J. 1986. *Parallel and Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. The MIT Press.

SAMAD, T. 1988. Back-propagation is significantly faster if the expected value of the source unit is used for update. In *International Neural Network Society Conference Abstracts*.

TAKANASHI, I., LUM, E. B., MA, K.-L., AND MURAKI, S. 2002. ISpace: Interactive volume data classification techniques using independent component analysis. In *Proceedings of Pacific Graphics 2002 Conference*.

VAN GELDER, A., AND HOFFMAN, U. 1996. Direct volume rendering with shading via three-dimension textures. In *ACM Symposium on Volume Visualization '96 Conference Proceedings*.

WERBOS, P. 1974. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Department of Applied Mathematics, Harvard University.

Proceedings of the 14th IEEE Visualization Conference (VIS'03)
0-7695-2030-8/03 $ 17.00 © 2003 IEEE