

Eric B. Lum
James Shearer
Kwan-Liu Ma

Interactive multi-scale exploration for volume classification

Published online: 24 August 2006
© Springer-Verlag 2006

Abstract Filter banks are a class of signal processing techniques that can be used to reveal the local energy of a signal at multiple scales. Utilizing such filtering allows us to consider local texture and other data characteristics, and permits volume classification and visualization that cannot be accomplished easily using conventional, transfer function-based methods. Our filter bank approach increases the dimensionality, and thus, the complexity of the classification task. We have therefore developed an interactive user interface for specifying and visualizing these higher dimensional classifiers, which enables volume data exploration and visualization in a filter-bank space. We demonstrate that this technique

is particularly effective for the classification of noisy data, and for classifying regions that are difficult to approach using conventional methods.

Keywords Classification · Filter banks · Transfer functions · Scale-space filtering · Volume rendering · Hardware acceleration · Interface design · Visualization

E.B. Lum (✉) · J. Shearer · K.-L. Ma
University of California at Davis, One
Shields Avenue, Davis, CA 95616, USA
ericlum101@yahoo.com, {jjshearer,
klma}@ucdavis.edu

1 Introduction

One critical aspect of direct volume rendering is creation of transfer functions, which map voxel values to optical properties such as color and opacity. Because it is an interactive, explorative process, the very act of specifying a transfer function promotes data understanding that transcends what can be accomplished through changes in view. As a user modifies the transfer function, she explores the volume both spatially and in terms of the data space, thus providing her with insights into the inherent spatial/data relationships of the volume. Although transfer functions are a powerful framework for user interaction, they are typically primitive in their ability to classify different materials in a volume. This can be contrasted with

volume segmentation methods, such as level set [23, 27] and region growing [14, 15] techniques, which can better separate materials. However, these methods are often less interactive and, therefore, less suited for data space exploration.

It is possible to achieve more sophisticated classification of volume data without using segmentation if we take into account the local texture and neighborhood characteristics of the volume data during the classification process. To accomplish this, we introduce a filter bank [7, 32] approach to volume visualization. Our technique uses a series of filters to decompose volume data into multiple signals, each of which provides estimates of the characteristics of the neighboring values at a different scale. Volume classification thus becomes exploration of transfer function spaces with an input domain that takes into

account the multi-scale structure of each region. We also introduce an interface and accompanying methods of user interaction that allow the user to explore the filter-bank data space with the same expressive control as that found in traditional transfer function methods. We have been particularly careful to keep the classification task simple by using lower dimensional classifiers whenever possible.

We have built a hardware-accelerated, volume visualization system using this filter-bank approach, enabling highly interactive volume classification and visualization of noisy volume and volumetric regions that are difficult with traditional methods using 1D or 2D (gradient-based) transfer functions. Our main contributions include the identification of a set of filters that map well to the capabilities (and limitations) of current graphics hardware and, most especially, the design of an intuitive user interface for effectively exploring and utilizing the multi-scale volume data generated by the filter bank to refine volume classification.

2 Related work

The generation of transfer functions for volume visualization is an area of research that has received a great deal of attention [28]. Information about the volume data is often utilized directly by the system to aid transfer function definition, or is presented to the user to guide data exploration. For example, Fujishiro et al. [9] use topological information from a hyper-Reed graph of the volume data to define transfer functions, leading to visualization displaying the intrinsic structure of the scalar field. Bajaj et al. [1] show how to make informative visualization from volume data by examining a set of quantitative signature functions of contour surfaces in the data. Jankun-Kelly and Ma [19] present methods based on coherency measures to automatically generate transfer functions that best characterize a time-varying volume data set.

Levoy [24] discusses utilizing gradient magnitude to enhance material boundaries in volume data. Kindlmann and Durkin [20] suggest that by examining a two-dimensional scatter plot of data values and gradient magnitudes, opacity transfer functions can be easily defined to effectively capture boundary features between materials of relatively constant data value. Kniss et al. [21] extend this work by adding a set of direct manipulation widgets as an interface for defining multi-dimensional transfer functions. Lum and Ma [25] present a transfer function method that uses pairs of gradient aligned samples to enhance material boundaries using changes in illumination.

While the aforementioned methods are mainly data-centric, others are largely image-centric or employ machine learning. He et al. [13] utilize genetic algorithms to breed trial transfer functions. The user can either select functions from a set of rendered images or allow the system to be fully automated. Marks et al. [26] address

the parameter selection problem in general by rendering a multi-dimensional space of those parameters, resulting in a large number of images. In the context of volume visualization, the user navigates this space by browsing the images to choose appropriate transfer functions. König and Gröller [22] cast transfer functions specification as navigation in the data, color, and opacity domains, and introduce an interface design to support this paradigm. Finally, Tzeng et al. [30] introduce an intelligent interface for classifying volume data through interactive brushing and supervised learning. They demonstrate volume classification that cannot be achieved with conventional methods.

Our data-centric work is unique because it permits interactive user exploration and utilization of the multi-scale aspects of the volume data to assist transfer function specification. The multi-scale data is derived with a filter bank. Our design unifies 2D slice probing, filter-bank space and transfer function space exploration, and interactive volume visualization into one system, as shown in Fig. 1. We demonstrate volume classifications on a variety of data sets, which would be difficult using conventional methods.

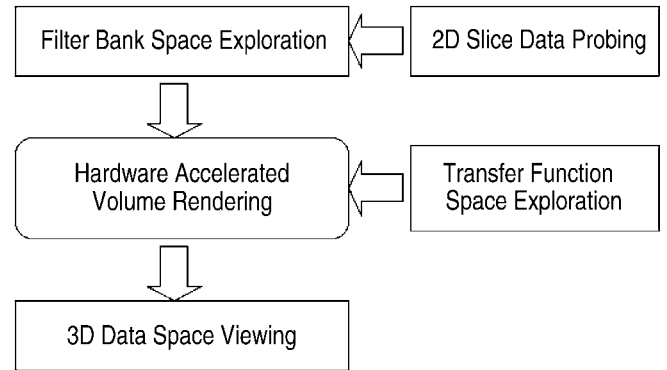


Fig. 1. Integration of interactive, filter-bank exploration into the classification process. Navigation in four distinct spaces enhances the user's expressive power

3 Filter banks for classification

Filter bank methods are popular in signal processing for analyzing data. A filter bank consists of a series of filters that splits a signal into a number of sub-bands for subsequent analysis. Figure 2 illustrates a filter bank that decomposes a signal $x(n)$ into N band limited signals y_1 through y_N using filters h_1 through h_N , respectively.

In the field of image processing, filter banks have been applied to the problem of texture classification [3, 6, 11, 17, 18, 29, 31]. The basic approach is to apply a set of filters to produce a set of images, each of which then provides a different estimate of the local energy distribution.

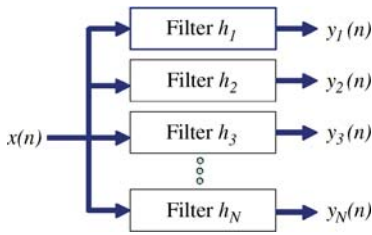


Fig. 2. A filter bank aids analysis by separating a signal into various sub-bands via the application of a set of filters

These energy estimates form a feature vector that can be used for classification, using methods such as Bayesian classifiers [31], neural networks [18], and clustering [11]. The significance of our work is the mapping of the filter bank approach to volume visualization and exploration. Rather than using statistical or machine learning based classifiers, we combine an interactive, hardware-accelerated filter bank volume renderer with a novel user interface. This interface allows both the exploration of the filtered data space, and specification of filter bank classifiers within this space.

To facilitate this interaction, it is important to select a set of filters that map well to the capabilities and limitations of modern graphics hardware. In particular, one must be conscious of the fact there is a limited amount of texture memory to store volumes and that rendering performance is often fill-rate limited. Applying an N -sized filter bank in a pre-processing step and using the N filtered copies would use a great deal of texture memory and thus limit the size of the volume that can be rendered interactively. On the other hand, applying the filter bank on-the-fly during rendering would negatively impact interactive performance.

Mindful of this concept, our implementation uses filter banks consisting of Gaussian low-pass filters that are applied in a pre-processing step. The resulting filtered volumes can be efficiently stored in an oct-tree data structure. Specifically, as the width of the Gaussian filter is increased and higher frequencies are attenuated, a lower sampling rate will accurately encode the filtered volume. Therefore,

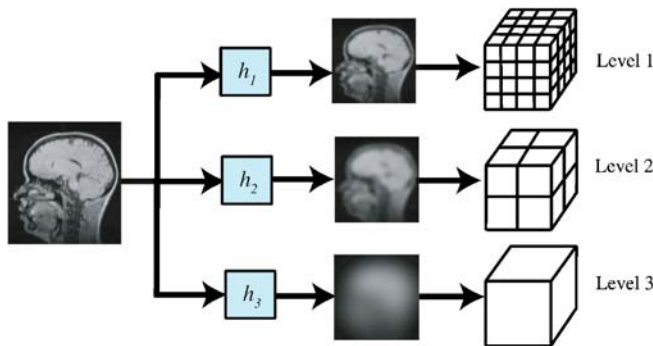


Fig. 3. In our implementation, the results of applying the Gaussian filter bank to the volume data are stored in the different levels of a mip-map texture

we employ an oct-tree data structure where each level approaching the coarsest resolution has had an increased amount of Gaussian smoothing, as depicted in Fig. 3. This representation has the added advantage of being natively supported in current graphics hardware in the form of mip-mapped textures. The ability to quadrilinearly filter textures (interpolate between mip-map levels) provides the capability of accessing the volume with filter levels that fall between those that were pre-computed. This multi-scale data structure is known as a Gaussian pyramid in the field of image processing [4].

The Gaussian filters utilized in our work have sigmas of two, four, eight, sixteen, etc. To reduce the number of mip-map levels that must be stored, larger sigma increments could be used, but at the expense of reducing the filtered feature vectors available for classification.

4 User interface

Given this hierarchical filtered representation of the volume, the task of specifying a transfer function with our filter-bank approach becomes one of specifying a mapping from the N -dimensional feature vector to a color and opacity. The overall design philosophy used in developing our user interface is to keep the dimensionality of the classifier as low as possible and only use additional feature vectors for classification where necessary. In order to accomplish this goal, it is not only necessary to have interactive volume rendering, but also to provide visual representations that will help guide the user by specifying these transfer functions and illustrating the relationship between spatial regions in a volume and their filter bank response.

Our interactive interface utilizes parallel coordinates [16] to visualize the multi-dimensional response of the filter bank. Parallel coordinates are a popular technique for visualizing multi-dimensional data and have been widely used by the information visualization community [2, 8, 10, 12]. As illustrated in Fig. 4, the top row (level zero) represents the distribution of data values for the original, unfiltered volume. As the lines move downward through the subsequent levels, they reveal a particular data value's response to progressively stronger filtering.

To specify the transfer function, we use material percentage classifiers as described by Drebin et al. [5]. In our application, the user assigns colors and opacities to a set of materials and then proceeds to specify classification functions that map data values to material percentages. The material percentage is then modulated to the material color, resulting in the transfer function color used for rendering.

User interaction in our work consists of the specification of classifiers for each material that users wish to classify. This task is accomplished by first specifying a 1D

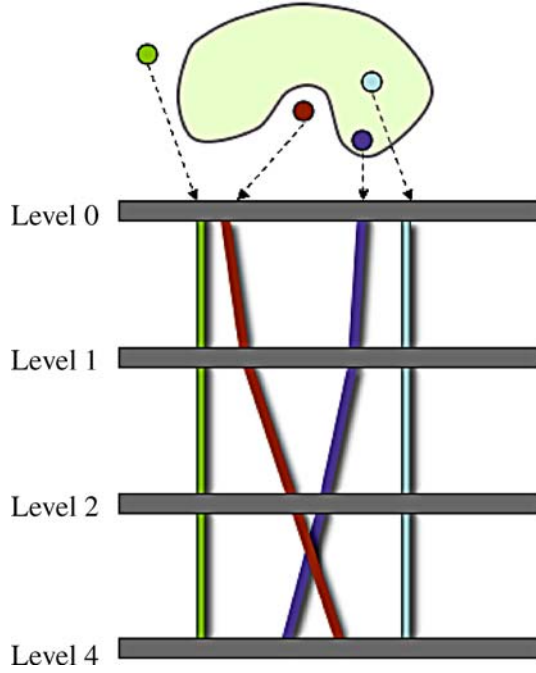


Fig. 4. An illustration of our multi-scale classification interface, which utilizes parallel coordinates to allow the user to operate in the filter-bank space. The thick horizontal lines are coordinate axes corresponding to a particular filtered data level. Each colored line corresponds to the data value of a sample point in each filter level. How the line changes with the application of progressively stronger Gaussian filtering can help to identify different types of data. For example, data points in the center of a large homogeneous area will change little, whereas small areas surrounded by unlike data values will be progressively blended with the dominant local value. Therefore, the lines for those points will move sharply to the right or left

transfer function for the top, unfiltered level and then refining the resulting classification by inserting additional classification widgets, as necessary, at varying filter levels. Figure 5 demonstrates our multi-scale interface before any refinement is done while operating on an MRI data set of a head. Probing the regions marked with the red and green arrows results in the parallel coordinate visualization of the filter bank response shown in Fig. 5c.

Isolating the brain from the surrounding material using only unfiltered data is not possible, since these areas are of similar data value. The mixing and crossing of data lines at level zero in the parallel coordinates representation highlight this trait. However, as the data values are plotted through the successive filter bank levels, we notice a clear distinction between the responses of the probed areas. This difference suggests that adding classifiers at higher dimensions will help to isolate our target regions. Figure 6 illustrates this refinement process, which adds additional 1D transfer function widgets at filter levels 2

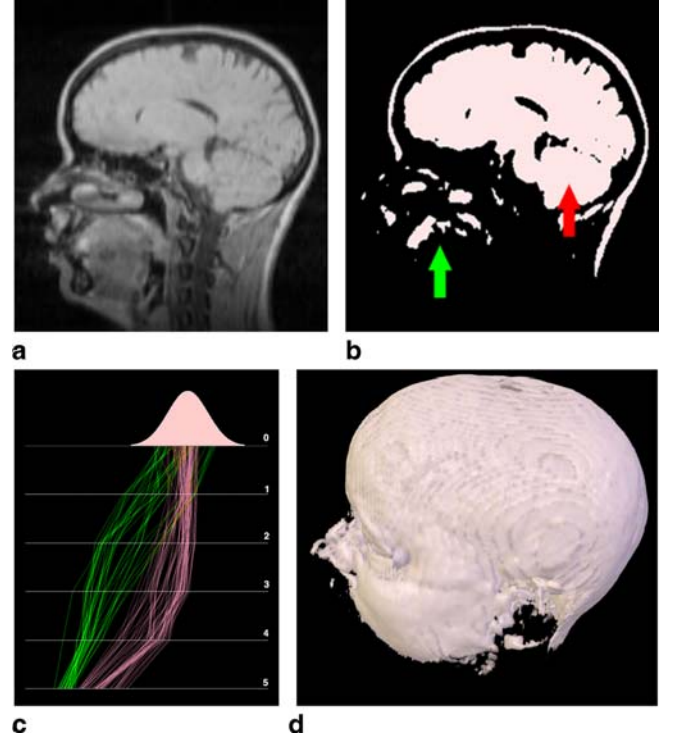


Fig. 5a–d. Results of applying a 1D transfer function to an MRI head data set. **a** shows a 2D slice from this data set, while **b** shows the corresponding classified 2D slice after the application of the transfer function. In the classified slice, the desired area is designated by the red arrow, while the green arrow indicates the unwanted region. After probing these areas in the 2D slice, the user can see the filter bank response, as shown in **c**. The resulting rendering is shown in **d**. As expected, and as shown by the mixing of wanted and unwanted data values at level zero, the brain is occluded by the surrounding material

and 3. The resulting higher dimensional classifier is the multiplication of the filtered response at each of the different N levels. This can be expressed as

$$p_i = \prod_{j=1}^N P_{ij}(s_j), \quad (1)$$

where p_i is the material percentage for material i , $P_{ij}()$ is a 1D classification function for filtered level j (if no classification is defined for that dimension), and s_j is the scalar value from level j . Each material classifier is thus linearly separable, while the overall transfer function is not separable and consists of the sum of M material percentage modulated material colors:

$$C = \sum_{i=1}^M p_i C_i. \quad (2)$$

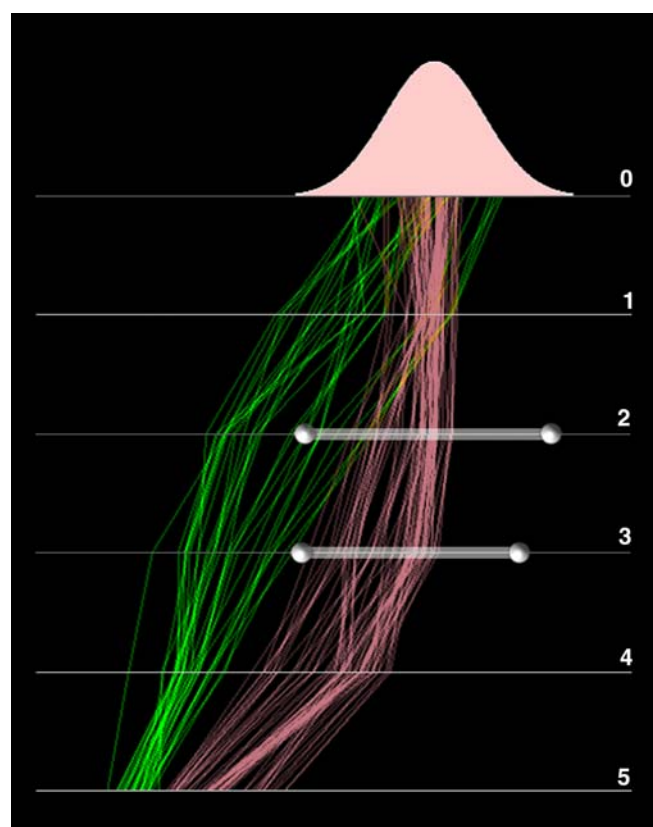
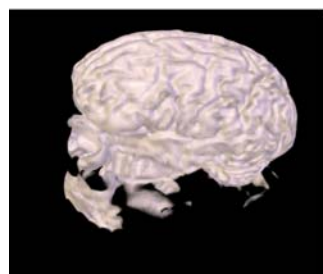
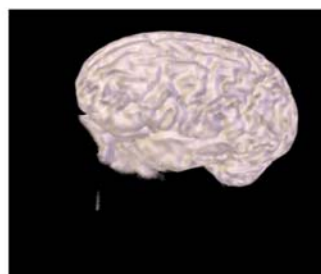
**a****b****c****d****e**

Fig. 6a–e. Refining the classification of the MRI data described in Fig. 5 with the goal of isolating the brain from the surrounding material. The bundles of pink and green lines in **a** are the result of user probes of the wanted and unwanted areas in a 2D slice. As before, the mixing of values at level zero indicates overlapping intensity values. However, the clear separation into pink and green bundles at subsequent filter levels suggest that a higher dimensional classification can help. Adding a classification widget at level two produced the improved results shown in **b** and **d**, yet was insufficient to fully isolate the brain. Adding another classifier at level three further refined the results, as shown in **c** and **e**

4.1 User interaction

As previously mentioned, the goal of this work is to use our filter bank approach with the lowest dimensional classifier possible. Therefore, the parallel coordinate representation is used as a tool to guide users in refining the specific material that they wish to classify. This requires providing a means for understanding how the various regions being classified map to the filter bank response, a means that we accomplished using a process of interactive probing.

We illustrate this process step-by-step with the MRI head data set example shown in Fig. 5. The user begins by classifying the volume using a traditional 1D transfer function (top portion of Fig. 5c). She can then evaluate the quality of the classification by looking at the volume rendered result (Fig. 5d), as well as the classified slices (Fig. 5b). If the 1D transfer function is not sufficient to classify the material of interest, the user can then interact with the segmented slice and probe on regions that are part of the desired material class (marked with the red arrow in Fig. 5b) and also regions that are misclassified (marked with the green arrow). As the user probes the 2D slice, the parallel coordinates representation displays the filter bank response. Notice that the regions that are not part of the brain have lines that move left as additional filtering is applied, indicating that these regions are small and surrounded by lower intensity materials. Based on this parallel coordinate visualization, an additional 1D transfer function widget is first added at level two, resulting in the classified result shown in Figs. 6b and 6d. An additional classifier is then added at level three (Fig. 6a), which further improves the classification as shown in Figs. 6c and 6e.

5 Hardware implementation

To maximize system interactivity, we implemented our volume renderer using graphics hardware. The volume is rendered using view aligned textured polygons [33] with the volume stored as a 3D mip-mapped texture. Our renderer makes use of post-data-interpolated classification, with the transfer function implemented in an OpenGL shading language fragment program. The transfer function itself is passed to the fragment program as a set of 1D textures – one set for each classified material class.

During rendering, on a per fragment basis, the filtered scalar values used for classification are read, using level-of-detail biased, quadrilinear interpolation. They are then used as indices for subsequent reads into the transfer function textures. The results from these transfer functions are then multiplied as expressed in Eq. 1 and used to determine the final color as expressed in Eq. 2. The vectorized nature of GPU architectures makes it possible to perform this classification process on four material classes

simultaneously by using the RGBA channels. With this approach, we can render a $256 \times 256 \times 256$ volume into a 512×512 window with approximately one sample per voxel in depth at approximately 2.5 frames per second using an NVIDIA GeForce 6800 Ultra. This performance is obtained with four material classes, classifiers with a three-dimensional input domain, and approximately one rendered sample per voxel in depth. The frame rate is slower than traditional 1D transfer function methods, but is sufficient for the type of interaction required for our work.

6 Results

In addition to the previously discussed MRI head data set, we have applied our multi-scale technique to three additional data sets: CT scans of a block of concrete, an aneurysm, and a Christmas tree. These data sets all contain material regions or features that cannot be clearly classified by exclusively using scalar values. The transfer function widget applied at each dimension of classification for the example data sets was either triangular, Gaussian, or a constant step.

We first apply our technique to a data set that was acquired for the purpose of non-destructive testing. This data set is a CT scan of a long glass rod embedded in a solid block of concrete. The rod is visible in the 2D slice in Fig. 7c as a dark gray vertical band approximately one-third from the left edge. The high-frequency, speckled nature of the concrete, as well as noise from the imaging process, makes the classification of this bar difficult for 1D transfer functions. Using the 1D transfer function at the top of Fig. 7e results in the volume rendered image and classified slice shown in Figs. 7a and 7d. Probing the two types of regions shown in Fig. 7d results in the filter bank distribution shown in red and green in Fig. 7e. Applying a 3D transfer function (i.e., three filters) that consider levels zero, two, and three results in the volume rendered result in Fig. 7b. The majority of the smaller, “noisy” material has disappeared, producing a much clearer classification of the glass rod.

Figure 8 shows a visualization of the aneurysm data set that was rendered with our method. The vessels shown can be classified with a traditional 1D transfer function. With our method, however, a transfer function that takes into account the *structure* of the surrounding material can be defined. In this case, different colors are used to show the relative sizes of the different vessels. The aneurysm is shown in light blue, the smaller vessel structures are shown in orange, while the intermediately sized vessels are shown in purple. A 2D transfer function was used for the blue region, while 3D transfer functions were used for the brown and purple regions.

Finally, we applied our method to the Christmas tree data set. Figure 9a depicts the result of using a 1D trans-

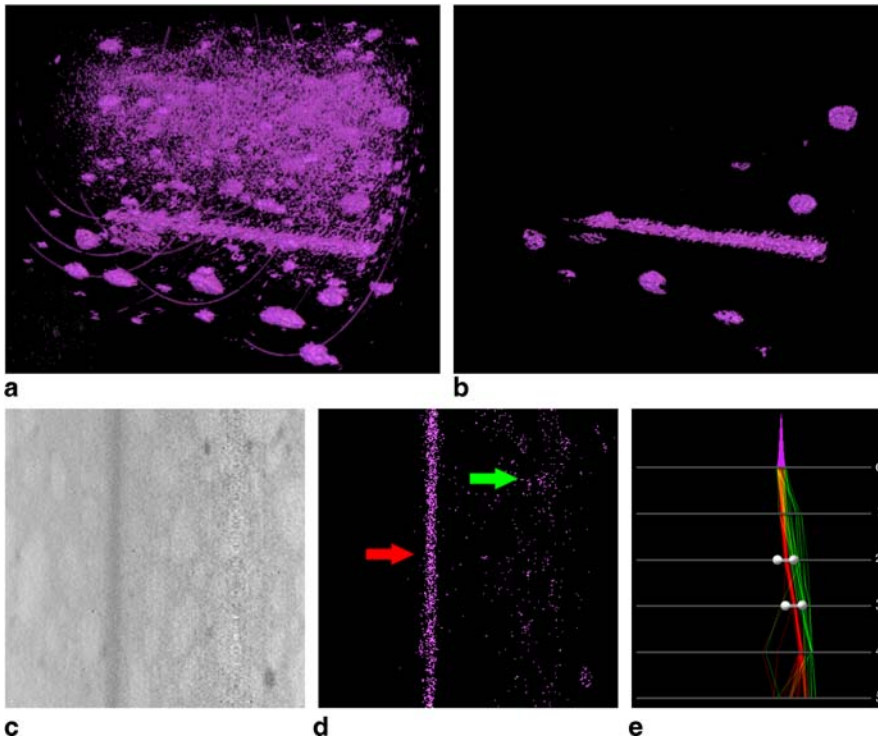


Fig. 7a–e. Results of refining the classification of a glass rod embedded in concrete using our technique. The embedded rod has a similar data value to many regions in the concrete, resulting in the poor classification shown in **a**. In **c**, the rod is visible as a dark gray band, while similar regions are visible as blotches. Once the user probes the areas shown in the classified slice, **d**, the resulting filter bank response is shown. Adding additional classifiers at filter bank levels containing a clear separation of probed regions, as in **e**, yields the improved volume rendering shown in **b**

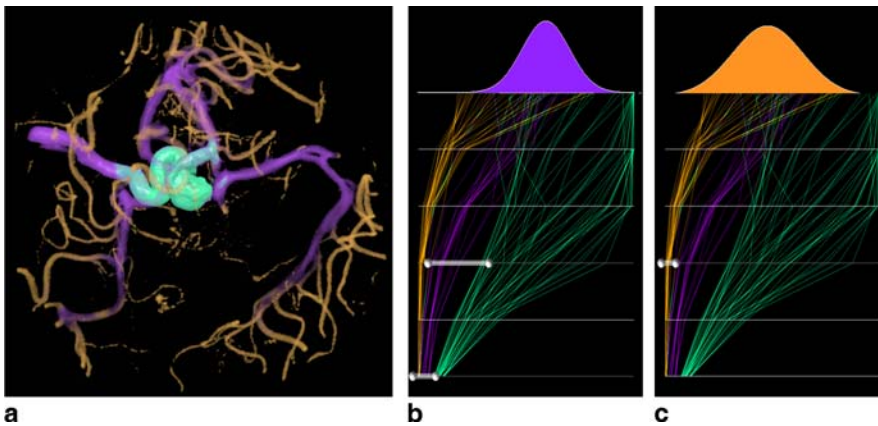


Fig. 8a–c. An aneurysm data set classified and rendered using our technique. Because it considers local data characteristics at multiple scales, it is possible to define a transfer function that assigns colors based on vessel size. **b** and **c** depict two of the three classification widgets used to assign the colors in **a**. This example demonstrates the ability to classify volumetric features based on size, a task not easily accomplished with conventional 1D or 2D transfer functions

fer function. Although the green needles can be classified with a 1D transfer function, the ornaments and branches cannot be separated. Figure 9b shows the result that can be obtained with our approach. The red ornaments and branches were classified with a 2D transfer functions, while the white candles were classified using a 3D transfer function.

7 Discussion

It is worth discussing the types of information provided by a Gaussian filter bank and how they differ from the most popular filter used in volume visualization, a non-linear

filter that computes gradient magnitude. Using Gaussian filters of varying widths provides an indication of both the local and global structure of a point at varying scales. This can be particularly useful for classifying materials based on neighboring attributes. As a low-pass filter, a Gaussian filter is also very tolerant of noise. Gradient magnitude, on the other hand, provides an indication of the local change in intensity and is thus very effective for use in classifying and visualizing those regions in a volume with the most change, specifically material boundaries. The local nature of gradient magnitude, however, makes it susceptible to noise and limits its ability to classify based on texture, or perform classification of homogeneous regions. The Gaussian filter bank approach that we utilize has its strength

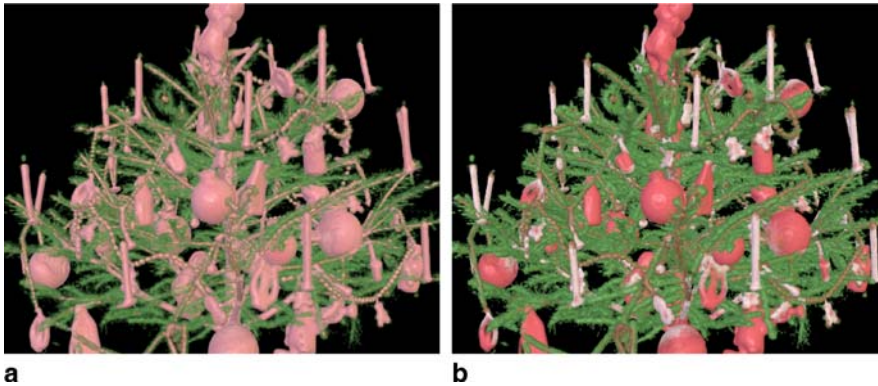


Fig. 9. **a** shows the Christmas tree data set rendered with a 1D transfer function. The similar material values found on the branches and ornaments prohibit classification based exclusively on scalar value. The image in **b** has been classified and rendered using our method. Notice that the ornaments and branches are classified as separate materials and have been assigned different colors

in classifying *regions* of a volume and is well-equipped to deal with noise and texture, while the gradient magnitude approach to volume visualization has its strength in classifying important *boundaries* in a volume.

We have demonstrated with concrete examples how scale-based filtering can be used to classify volumetric regions and noisy volumes. Further study of the limitations of this approach is needed. How do we relate structures across scales? Would directional Gabor filters or wavelet transforms with orthogonal bases be able to handle more complex classifications while maintaining the simplicity of the existing user interface? Our work shows a very promising direction for further research in volume visualization.

8 Conclusion

We have presented a new approach to interactive volume classification that allows the user to explore a filter bank space of inputs for specifying transfer functions. The method considers not only the material value, but also the

neighboring properties of a value at multiple scales. Careful attention has been paid to keeping the classification task simple, using small sets of 1D transfer function widgets (in all examples, three or fewer) to accomplish these classifications. We believe that noise, texture, and neighboring structure are important aspects of volume classification and our approach can help scientists overcome these aspects so that they can more effectively visualize and explore their data. In particular, the ability to classify volumetric features according to their sizes, as shown with the aneurysm data set, is very powerful and has wide applications in science and engineering, such as the study of 3D turbulent flow.

Acknowledgement This work was sponsored in part by the US National Science Foundation under contracts ACI 9983641 (PECASE award), ACI 0325934 (ITR), and ACI 0222991; the US Department of Energy under Memorandum Agreements No. DE-FC02-01ER41202 (SciDAC program), and No. B523578 (ASCI VIEWS); the LANL/UC CARE program; 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.

References

1. Bajaj, C.L., Pascucci, V., Schikore, D.R.: The contour spectrum. In: Proceedings of IEEE Visualization '97 Conference, pp. 167–173 (1997)
2. Barlow, N., Stuart, L.J.: Animator: A tool for the animation of parallel coordinates. In: Proceedings of the 8th International Conference on Information Visualisation, pp. 725–730 (2004)
3. Bovik, A.C., Clark, M., Geisler, W.S.: Multichannel texture analysis using localized spatial filters. IEEE Trans. Pattern Anal. Mach. Intell. **12**(1), 55–73 (1990). DOI <http://dx.doi.org/10.1109/34.41384>
4. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Trans. Commun. **31**(4), 532–540 (1983)
5. Drebin, R., Carpenter, L., Hanrahan, P.: Volume rendering. In: SIGGRAPH '88 Conference Proceedings, pp. 65–74 (1988)
6. Dunn, D., Higgins, W.E., Wakeley, J.: Texture segmentation using 2D Gabor elementary functions. IEEE Trans. Pattern Anal. Mach. Intell. **16**(2), 130–149 (1994). DOI <http://dx.doi.org/10.1109/34.273736>
7. Fliege, N.J.: Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Wavelets. John Wiley, New York (1994)
8. Fua, Y.H., Ward, M.O., Rundensteiner, E.A.: Hierarchical parallel coordinates for exploration of large datasets. In: VIS '99: Proceedings of the conference on Visualization '99, pp. 43–50. IEEE Computer Society Press, Los Alamitos, CA (1999)
9. Fujishiro, I., Azuma, T., Takeshima, Y.: Automating transfer function design for comprehensible volume rendering based on 3D field topology analysis. In: Proceedings of IEEE Visualization '99 Conference, pp. 467–470 (1999)
10. Gennady Andrienko, N.A.: Parallel coordinates for exploring properties of subsets. In: Proceedings of the 2nd International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV '04), pp. 93–104 (2004)
11. Georgescu, B., Shimshoni, I., Meer, P.: Mean shift based clustering in high dimensions: A texture classification example. In: ICCV '03: Proceedings of the 9th IEEE International Conference on Computer Vision, p. 456. IEEE Computer Society, Washington, DC (2003)
12. Hauser, H., Ledermann, F., Doleisch, H.: Angular brushing of extended parallel coordinates. In: INFOVIS '02: Proceedings of the IEEE Symposium on Information

- Visualization (InfoVis'02), p. 127. IEEE Computer Society, Washington, DC (2002)
13. He, T., Hong, L., Kaufman, A., Pfister, H.: Generation of transfer functions with stochastic search techniques. In: Proceedings of IEEE Visualization '96 Conference, pp. 227–234 (1996)
 14. Huang, R., Ma, K.-L.: RGVis: Region growing based techniques for volume visualization. In: Proceedings of Pacific Graphics 2003 Conference, pp. 355–363. IEEE (2003)
 15. Huang, R., Ma, K.-L., McCormick, P., Ward, W.: Visualizing industrial CT volume data for nondestructive testing applications. In: Proceedings of Visualization 2003 Conference, pp. 547–554 (2003)
 16. Inselberg, A.: The plane with parallel coordinates. *Visual Comput.* **1**(2), 69–91 (1985)
 17. Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using Gabor filters. *Pattern Recogn.* **24**(12), 1167–1186 (1991). DOI [http://dx.doi.org/10.1016/0031-3203\(91\)90143-S](http://dx.doi.org/10.1016/0031-3203(91)90143-S)
 18. Jain, A.K., Karu, K.: Learning texture discrimination masks. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(2), 195–205 (1996). DOI <http://dx.doi.org/10.1109/34.481543>
 19. Jankun-Kelly, T.J., Ma, K.-L.: A study of transfer function generation for time-varying volume data. In: Proceedings of Volume Graphics 2001, pp. 51–65 (2001)
 20. Kindlmann, G., Durkin, J.W.: Semi-automatic generation of transfer functions for direct volume rendering. In: IEEE Symposium on Volume Visualization, pp. 79–86 (1998)
 21. Kniss, J., Kindlmann, G., Hansen, C.: Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In: Proceedings of IEEE Visualization 2001 Conference, pp. 255–262 (2001)
 22. König, A., Grollier, E.: Mastering transfer function specification by using VolumePro technology. In: Tosiya L. Kunii, ed., *Spring Conference on Computer Graphics 2001* **17**, pp. 279–286 (2001)
 23. LeFohn, A., Cates, J., Whitaker, R.: Interactive, GPU-based level sets for 3D brain tumor segmentation. In: Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI), pp. 564–572 (2003)
 24. Levoy, M.: Display of surfaces from volume data. *IEEE Comput. Graph. Appl.* **8**(3), 29–37 (1988)
 25. Lum, E., Ma, K.-L.: Lighting transfer functions for direct volume rendering. In: Proceedings of the IEEE Visualization 2004 Conference, pp. 289–296 (2004)
 26. Marks, J., Andalman, B., Beardsley, P., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., Shieber, S.: “Design galleries”: A general approach to setting parameters for computer graphics and animation. In: *SIGGRAPH '97 Conference Proceedings*, pp. 389–400 (1997)
 27. Osher, S., Fedkiw, R.: *Level Set Methods and Dynamic Implicit Surfaces*. Springer, Berlin Heidelberg New York (2002)
 28. Pfister, H., Lorensen, B., Bajaj, C., Kindlmann, G., Schroeder, W., Avila, L.S., Martin, K., Machiraju, R., Lee, J.: The transfer function bake-off. *IEEE Comput. Graph. Appl.* **21**(3), 16–22 (2001)
 29. Randen, T., Husøy, J.H.: Filtering for texture classification: A comparative study. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(4), 291–310 (1999). DOI <http://dx.doi.org/10.1109/34.761261>
 30. Tzeng, F.Y., Ma, K.-L., Lum, E.: A novel interface for higher-dimensional classification of volume data. In: Proceedings of IEEE Visualization 2003 Conference, pp. 505–512 (2003)
 31. Unser, M.: Local linear transforms for texture measurements. *Signal Process.* **11**(1), 61–79 (1986). DOI [http://dx.doi.org/10.1016/0165-1684\(86\)90095-2](http://dx.doi.org/10.1016/0165-1684(86)90095-2)
 32. Vaidyanathan, P.P.: *Multirate systems and filter banks*. Prentice-Hall, Upper Saddle River, NJ (1993)
 33. Van Gelder, A., Kim, K.: Direct volume rendering with shading via three-dimension textures. In: *ACM Symposium on Volume Visualization '96 Conference Proceedings*, pp. 23–30 (1996)



ERIC B. LUM received his PhD degree in computer science from the University of California at Davis in 2004 and a BS and an MS degree in electrical engineering from UCLA in 1997 and 1999, respectively. From 2004 through 2005 he was a postdoctoral researcher in computer science at UC Davis where he investigated rendering and user interaction methods for expressive scientific visualization. He is currently at NVIDIA Corporation.



JAMES SHEARER is currently pursuing his PhD at the University of California at Davis with a focus on interfaces for scientific and information visualization. He is working under the guidance of Professor Kwan-Liu Ma. He enjoys raising his children and exploring the many wonders of Northern California.



KWAN-LIU MA is a professor of computer science at the University of California at Davis. He has a PhD in computer science from the University of Utah. His research spans the fields of visualization, user interfaces, and high-performance computing. He and his students are presently working on the problems of visualizing terascale scientific simulations, cybersecurity, social networks and nondestructive testing, as well as developing new visualization methodologies, infrastructures, and interfaces.