

Texture-based Transfer Functions for Direct Volume Rendering

Jesus J. Caban, *Student Member, IEEE*, and Penny Rheingans, *Member, IEEE*

Abstract— Visualization of volumetric data faces the difficult task of finding effective parameters for the transfer functions. Those parameters can determine the effectiveness and accuracy of the visualization. Frequently, volumetric data includes multiple structures and features that need to be differentiated. However, if those features have the same intensity and gradient values, existing transfer functions are limited at effectively illustrating those similar features with different rendering properties. We introduce texture-based transfer functions for direct volume rendering. In our approach, the voxel's resulting opacity and color are based on local textural properties rather than individual intensity values. For example, if the intensity values of the vessels are similar to those on the boundary of the lungs, our texture-based transfer function will analyze the textural properties in those regions and color them differently even though they have the same intensity values in the volume. The use of texture-based transfer functions has several benefits. First, structures and features with the same intensity and gradient values can be automatically visualized with different rendering properties. Second, segmentation or prior knowledge of the specific features within the volume is not required for classifying these features differently. Third, textural metrics can be combined and/or maximized to capture and better differentiate similar structures. We demonstrate our texture-based transfer function for direct volume rendering with synthetic and real-world medical data to show the strength of our technique.

Index Terms—visualization, statistical analysis, volume rendering, data variability, medical imaging

1 INTRODUCTION

Direct volume rendering provides a visual representation of three-dimensional volumetric data based on functions that map densities to specific colors and opacities. These *transfer functions* are crucial in the understanding of the overall volumetric data and individual features contained within the volume space. Specification of transfer functions to disambiguate the various materials and structures is a tedious and time-consuming task. In addition, accurately identifying and visually distinguishing various objects in a volume is a challenging task.

Researchers have used first- and second-order derivative-based functions to more accurately identify boundaries between different materials [7, 8]. Kniss et al. [9] introduced dual domain interaction to facilitate identification of boundaries in the three-dimensional domain using a probe that facilitates manual segmentation of various materials. However, most transfer functions are still attached to the voxel's properties such as intensity, gradient, and curvature. This leads to problems when the volumetric data under consideration contains multiple structures with similar intensity values and geometrical properties.

Like images, volumes commonly contain characteristic patterns and textures that our visual system can clearly identify. In particular, volumes frequently present very small structures – textons – which combined create different structures and characteristic regions. Studies in human perception and psychological research have shown that textures and patterns play an important role in the overall interpretation and understanding of the underlying structure. Motivated by those studies and the need to better differentiate similar structures within volumetric data, we have designed a texture-based transfer function which uses local textural properties of the data in addition to the individual intensity values.

We introduce texture-based transfer functions (TbTF) to enhance the visualization of volumetric data by allowing similar structures to be rendered differently. In our technique, a volume is first decomposed into overlapping or non-overlapping regions or subvolumes of interest. For each subvolume, a multi-dimensional texture-based de-

scriptor is computed which captures the local textural statistical properties. All the statistical descriptors are pre-computed and stored in a vector image. During the raycasting integration process, the vector image is used to look up the voxel's textural properties. Based on those properties, different opacity and color are assigned to each voxel, thus enhancing the visualization and illustration of regions with similar density values.

Additionally, we evaluate our techniques by experimenting with synthetic data as well as real-world medical data. Based on our observations, we provide guidelines regarding the texture sizes that more effectively capture local structural details. We also demonstrate the benefits of weighting and combining individual textural properties to provide effective visualization of similar structures.

2 RELATED WORK

The design of effective transfer functions for direct volume rendering has been a widely researched topic. Early work introduced the use of gradients for improved classification in the raycasting process [11]. Accurately classifying materials has been a challenging task that was approached using gradient-based multi-dimensional transfer functions [7, 9]. These techniques were particularly useful in the medical domain where the boundaries of adjacent anatomical parts are not easily distinguishable using only the voxel intensity. Multi-dimensional transfer functions were further extended to accentuate and highlight regions of high curvature [8].

Transfer functions have also been used in the illustrative visualization domain. Hauser et al. [4] were the first to propose a novel framework to mix illustrative representations with standard volume rendering (direct, MIP, x-ray) to highlight features in the visualization while providing context to the viewer at the same time. Style transfer functions were introduced by Bruckner et al. [1] to give more control to the visualization designer. These techniques have been mostly used with segmented data.

There has been some work in the field of analyzing the spatial characteristics around the voxel under consideration to apply some smart classification techniques. Roettger et al. [18] used the voxel barycenter and the region variance to assist manual specification of colors for similar features in the process of volume rendering. Sato et al. [19] characterized tissues of interest by explicitly defining rules and filters such as edges, sheets, lines, and blobs for each local structure. The filter characteristics were analyzed using a Gaussian models. Two limitations of this technique were that all tissue classification was reduced to four simple structures and that the smoothing could cause loss of important local structures and patterns. Their classification was fur-

- Jesus J. Caban is with the Department of Computer Science, University of Maryland (UMBC), E-mail: caban1@cs.umbc.edu.
- Penny Rheingans is with the Department of Computer Science, University of Maryland (UMBC), E-mail: rheingan@cs.umbc.edu.

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

ther improved with a multichannel approach but it required complex user interaction for good classification results. Ljung et al. [12] developed a technique that analyzed the neighborhood using statistical techniques such as range weight.

In addition, recent work has suggested the use of local texture analysis to better classify volume data. Tzeng et al. proposed a technique to classify volume data through a supervised learning technique that considers neighbors density and gradient values [21]. The technique was limited by the amount of user iteration required to select regions of interest and by the few neighboring metrics used. Lum et al. [13] suggested the use of local textures, scale-based filtering, and parallel coordinates to better classify volume data interactively. The effectiveness of that approach was limited by how well different materials can be distinguished and identified from the changes of the parallel coordinates.

In our work, we use first-, second-, and high-order local statistical texture properties to effectively assign voxels to different opacities and colors. Based on local statistical texture properties and the set of statistical metrics we employ, our texture-based transfer functions can accurately capture differences and similar local properties to better differentiate individual structures. In contrast with most existing techniques which are primarily attached to the voxel's density values, our technique automatically assigns voxels with similar intensity values and different textural properties to individual transfer functions.

Some of the benefits of our texture-based transfer functions are that different structures which have the same intensity values can be rendered differently by the raycast function. Second, previous knowledge of the specific structures or segmentation of the different structures is not needed. In addition, textural properties and statistical metrics can be combined in a manner that maximizes the differences in structures that are similar in density and even some textural properties.

3 TEXTURE ANALYSIS

Texture analysis is a fundamental technique used in medical imaging and computer vision to identify, characterize, and compare regions with similar properties. Statistical textural analysis measures and captures local image properties which are not necessarily based on intensity properties, they are rather a combination of intensity, local patterns, and local image statistics. It has been demonstrated that the generation of textural properties can be used to enhance classification and characterization of individual image regions[15, 16].

Our TbTF uses local textural properties to map a given voxel to a specific opacity and color. Instead of only considering the intensity value of the voxel, the gradient information, or the interpolation between neighboring voxels, our TbTF maps textural properties to specific rendering attributes.

We use a combination of first-, second-, and high-order statistics to capture textural properties. In particular, we use histogram statistics in conjunction with co-occurrence and run-length matrices to capture structural and geometrical properties.

First-order statistics are calculated from the probability of observing a particular pixel value at a randomly chosen location in the image. First-order statistics are among the simplest textural measurements that can be extracted from 2D/3D images, specially given that they are obtained from a histogram. A histogram is a simple measurement that maps intensity values into various disjoint bins. Let $\pi(i)(i = 1, 2, \dots, n)$ be the number of voxels whose intensity is i in a given subvolume v_k of size m^3 . The probability of intensity i being within the subvolume v_k is computed by

$$h(i) = \frac{\pi(i)}{m^3} \quad (1)$$

Using such probabilities and the image histogram it is possible to extract several local statistical properties including mean, variance, kurtosis, skewness, and deviations.

A number of local statistical properties cannot be captured by first-order statistics, only with more advanced texture-analysis techniques such as second- or high-order statistics. Second-order statistics measure the likelihood of observing an intensity value i and j at an average

distance $\vec{\delta} = (dx, dy)$ apart[5]. Frequently, second-order statistics are computed using co-occurrence matrices as demonstrated by Haralick et al.[3] Let $\delta = (r, \phi)$ denote a vector in the polar coordinate of the subvolume. We can compute the joint probability of the pairs of gray levels that occur at pairs of points separated by δ . That joint probability can be stored on a matrix $P(i, j)$ which contains the probability of observing the pair of gray levels (i, j) occurring at separation δ . In our case, we estimate the co-occurrence matrices by computing different matrices for the angles $\phi = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ within each axis. To guarantee rotation-invariant properties, we compute the average co-occurrence matrix and then extract statistical properties including energy, contrast, correlation, inertia, entropy, and sum of entropies.

To better identify and characterize regions with similar properties, we use run-length matrices of higher-order statistics to estimate the number of graylevel runs within the volume[20]. The run-length matrix $P_\theta(i, k)(i = 1, \dots, m; k = 1, \dots, n)$ represents the frequency that k points with 32-bit quantized gray level i continue in the direction $\theta \in 0^\circ, 45^\circ, 90^\circ, 135^\circ$. From a run-length matrix textural statistics such as the amount of short (fine) runs, long (coarse) runs, and the uniformity of such runs can be estimated.

1 st Order Statistics	2 nd Order Statistics	Run Length Matrices
Mean	Energy	Short Run (SRE)
Variance	Inertia	Long Run (LRE)
Skewness	Inverse Diff	Gray Level Nonuniform (GLNE)
Kurtosis	Entropy	Low Gray Run (LGRE)
Absolute Deviation	Correlation	High Gray Run (HGRE)
Standard Deviation	Contrast	Short Run Low Gray (SRLGE)
	Sum Entropy	Long Run Low Gray (LRLGE)

Table 1. Texture metrics used to create a multi-dimensional descriptor for each voxel or region of interest.

In total, our TbTF uses 20 textural metrics within a multi-dimensional descriptor to differentiate local textural properties. Table 1 lists all the metrics used by our system. For a mathematical description and formulas, please see [3, 20]. In our system, all the textural metrics are pre-computed and stored in a vector image which the raycast function uses during the rendering process to look-up the local textural properties.

4 APPROACH

Our approach is based on our observation that volumetric data can be described as a collection and specific arrangement of 3D *textons*. By analyzing the specific textural pattern that each collection of 3D *textons* creates, we can automatically enhance the visualization and more effectively illustrate structures with similar intensity values.

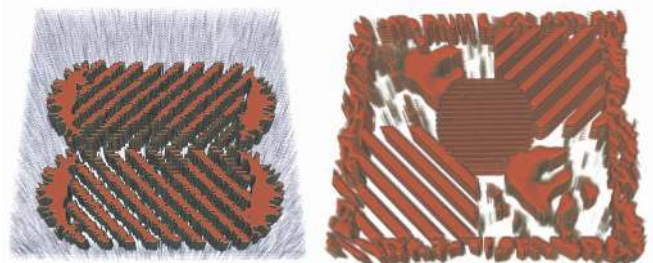


Fig. 1. Two synthetic datasets used to evaluate our system. Most of the structures within the volumes have the same intensity value limiting how effectively existing rendering techniques can differentiate individual structures. Note that the individual structures and different patterns can be visually distinguished.

Figure 1 shows two synthetic volumes used to test our system. Both 3D datasets have different patterns and structures with the same intensity values which existing transfer functions are unable to automatically differentiate. We can see different structures and patterns by

looking at the images in Figure 1, however existing volume rendering techniques have limitations and difficulties when it comes to effectively differentiating structures that have the same density values and similar gradients (as seen in Figure 2(left)).

Our texture-based transfer functions (TbTF) work as follows: Given an input volume V , we first partition the 3D image into $|P|$ overlapping or non-overlapping regions. For each region or subvolume $v_i \in P$, a set of statistical texture properties are computed and stored in a vector image V' . Once the vector image has been generated, the pre-computed statistical properties are used by the raycasting function to look up local structural properties and assign different color and opacity to voxels with similar density but with distinct textural properties.

4.1 Volume Partitioning

The first step is to partition an input volume into meaningful regions where individual statistical properties can be estimated more accurately. Given an input image V , we first partition the volume into $|P|$ regions or subvolumes of interest. Each region is described by two parameters: (ρ, σ) . The value $\rho \geq 1$ denotes the region size and $\sigma \geq 0$ denotes the overlap. For instance, to compute the per-voxel statistical properties of a given volume V using a neighboring window w , the volume partition will be created with $\rho = 1$ and $\sigma = w$. The two primary considerations for estimating the value of ρ and σ are that the region or subvolume has to be large enough to capture statistical textural properties, but it has to be small enough to capture only local properties relevant to surrounding voxels.

4.2 Pre-computing Textural Properties

After determining the values ρ and σ and generating a specific partition, a multi-dimensional vector containing the structural texture properties is estimated for each region $v_i \in P$. A combination of first-, second-, and high-order statistics are used to extract textural properties from each subvolume of interest. Once all the 20 metrics are computed and combined into a multi-dimensional vector they are normalized and stored in a vector image V' .

The volumetric partitioning and the statistical textural analysis are pre-computed and used during the visualization step. Alternatively, the partitioning and texture analysis can be done during the initialization step given that the process takes only a few seconds in cases with relatively large partitions or subvolumes of interest. The performance and timing results are further discussed in section 5.1.2.

4.3 Texture-based transfer function

After the vector image with all the textural properties has been generated, it is used to enhance volumetric visualization. In our system, during the rendering process, a voxel is represented by its local textural properties which enables the flexibility of enhancing and distinguishing regions with similar density and gradient values, but with different local textural properties.

We present four specific ways to use our TbTF to enhance visualization and improve the differentiation of individual structures. First, the user can specify a value k with the number of structures to discriminate, then by using the multi-dimensional descriptors, vectors can be automatically classified in k groups. Second, the user has the flexibility of enhancing the visualization by manually or automatically changing individual metrics. Third, the user can select a specific area and based on similarity measurements, all equivalent structures can be highlighted by the transfer function. Finally, we also present a technique which provides the flexibility of selecting two regions and computing individual weights for each metric, thus maximizing the differences between those regions.

4.3.1 Automatic Classification

After generating or loading a pre-computed vector image V' containing all the textural statistical properties, automatic classification of the data is possible. The user can specify a value k with the number of structures to highlight or display with different rendering properties.

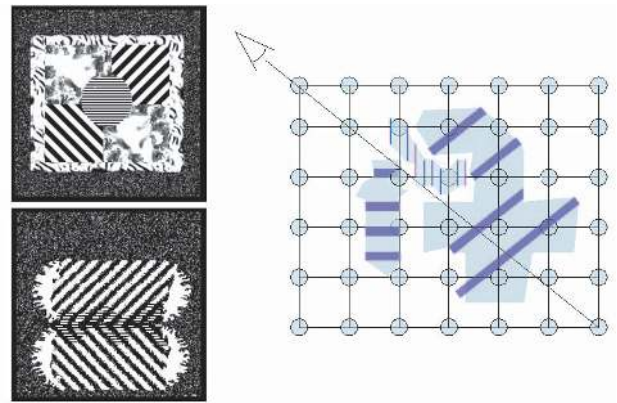


Fig. 2. (Left) 2D slices of our synthetic data. Since most of the structures have the same intensity value, existing transfer functions are limited in effectively differentiating them. (Right) Diagram of the modification to the raycasting process to incorporate our TbTFs. As the ray traverses through the volume, each voxel location is identified by the pre-computed local textural properties. Thus, enabling effective enhancement of structures with different textural properties.

Then, the entire collection of vectors or a set of non-neighboring vectors $v_i \in V'$ can be used to classify the data. By using an inequality-based fast k-means implementation[2], the data can be clustered within seconds and rendered differently. Note that during the classification process all selected metrics of each sub-volume v_i are uniformly weighted. In Section 4.3.4 we will show how to automatically assign non-uniform weights to each metric.

Our technique of using unsupervised k-means to automatically classify the data can be seen as a type of segmentation. However, given the flexibility the user has to change the value k during the visualization process and/or selecting specific metrics to use during the unsupervised classification step, we believe that automatic classification is an important benefit our TbTF provides.

4.3.2 Individual Weighted Metrics

A second technique our system provides is to enhance the visualization of structures with similar intensity values by selecting individual metrics. The user can automatically select and change any of the 20 individual metrics to better discriminate the data. Since every region or voxel is represented by a vector that captures the local textural properties, the user can select individual metrics or a combination of different metrics to highlight specific structures. During the rendering process, for every region or voxel under consideration, if the value of the specific metric(s) under consideration are above the user's threshold, the raycast function will assign the voxel a different opacity, color, and rendering properties.

Given the large amount of metrics the user can change and the possible complexity of understanding individual metrics, weighted metrics can be difficult for data exploration. However, a quick review of medical imaging and computer-aided diagnosis systems shows that researchers have provided specific textural metrics that are important to highlight particular diseases or anatomy[6, 10, 17]. TbTF provides the flexibility of using such information to individually weight different metrics and enable domain experts to better highlight important anatomical structures or patterns.

4.3.3 Similarity Measurements

Another technique used to enhance the visualization of structures with similar density and gradient values are similarity measurements. We have seen that in high-dimensional volume rendering [9, 13] 3D widgets have been accepted as techniques to select areas of interest. In our system, the user can employ a 3D widget to select a specific region or area of interest S . The combination and averaging of all the multi-dimensional descriptors ($v_i \in S$) contained within the selected region

result in a characteristic descriptor. That multi-dimensional descriptor can then be used to highlight similar structures during the rendering process.

During the raycasting and accumulation process, if the textural properties of a specific voxel is within a certain distance from the characteristic descriptor, then the region or voxel under consideration receives different opacity values and color properties. It is important to note that this approach does not add any additional complexity to the raycasting process given that it only adds a single lookup to determine if the local statistical properties are within a given threshold. Figure 2(right) shows a diagram with our modification to the raycast process.

The raycasting process results in the creation of a visualization that highlights different structures and displays features with similar density values and distinct textural patterns differently.

4.3.4 Non-uniform Weights

Another way to use our TbTF to enhance structures with similar intensity and gradient values is to non-uniformly assign weights to each textural metric. By using a 3D widget, the user can select two small regions of the structures that need to be differentiated. Then, since the vector image V' containing all the textural metrics is always kept in memory, two sets – S_1 and S_2 – can be generating by grouping all the multi-dimensional descriptors of the selected areas. The two sets – S_1 and S_2 – represent two particular classes. All the textural metrics for each class are analyzed and based on their textural metrics, differences are maximized. By using the mutual information between the two classes, the textural metrics that are redundant between the two classes and the features that are the most relevant for each independent class can be identified[14]. From such results, individual weights can be assigned to each group – thus maximizing their differences. Finally, during the raycast function by comparing the non-uniformly weighted descriptor, two similar structures can be automatically differentiated.

5 RESULTS

To test the strengths and limitation of our system, we have used synthetic and real-world medical volumes. In addition, we have compared different ρ and σ values to provide specific information regarding the parameters which work more effectively.

5.1 Synthetic Data

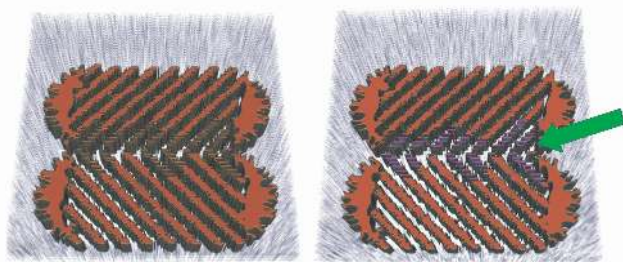


Fig. 3. (Left) Synthetic data being visualized with standard transfer functions. Since most of the areas have the same intensity value, existing visualization techniques are limited in how effectively individual structures are rendered. (Right) The same synthetic dataset visualized using our TbTF. The center of the volume (pointed by the arrow) can be automatically highlighted by just analyzing local textural properties.

Figure 3(left) shows one of the synthetic volumes used to test our technique. Given that most of the areas and structures have the same intensity value, existing volume rendering techniques are limited in how effectively different structures can be illustrated separately and with individual rendering properties. We compared the results of using 1D and our texture-based transfer functions. Figure 3(right) shows the results of using $\rho = 4$, $\sigma = 0$, and the automatic classification with $k = 3$. Note that in our automatic classification the value k has to be $n + 1$ where n is the number of structures to differentiate (ie.

n structures plus noise). From the images we can see that existing transfer functions are limited in automatically differentiating regions with distinct patterns and textural properties.

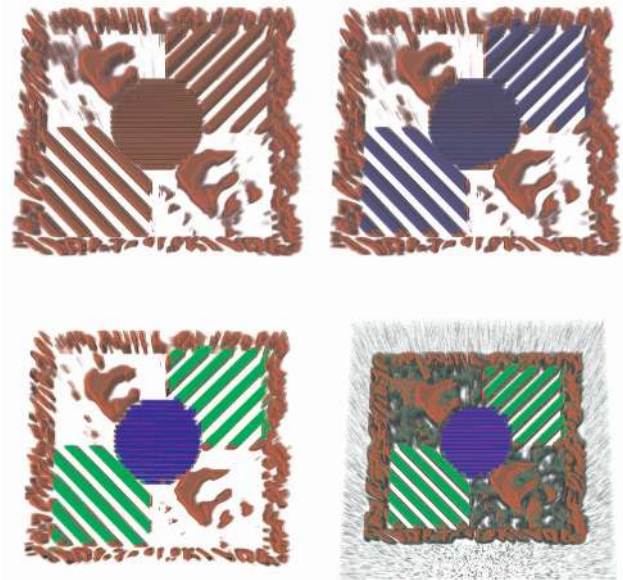


Fig. 4. (Top left) Results of a 1D transfer function. (Top right) Results of our TbTF with $\rho = 2$, $\sigma = 4$, and $k = 3$. (Bottom left) Results of our TbTF with $\rho = 2$, $\sigma = 4$, and $k = 4$. (Bottom right) Results of our TbTF with $\rho = 2$, $\sigma = 4$, $k = 4$, noise, and structures in different intensity ranges that do not have to be differentiated.

Figure 4 shows the results with our second synthetic volume which has both fine structures as well as coarse regions. From the result, we can see that a 1D transfer function is limited in separating structures. In addition, we can see that our TbTF was able to find the structures under different parameters, including the use of different values k to automatically classify the data.

From our results, we can see that structures with linear patterns at 0° , 45° , and 135° were classified and colored together. This is the result of using co-occurrence and run-length matrices which are invariant to rotation. Note that this property of rotationally invariance is not required. However, given that in most real-world 3D images textures or characteristic patterns can be found within the entire volume space and in very limited situations we can find patterns pointing in the same direction, rotationally invariant features help texture-based classification.

5.1.1 Texture Size

Using the same synthetic datasets, we tested our TbTFs with different texture sizes ρ and different overlaps σ to find the specific size which best captures structural patterns and textural properties. Figure 5 compares the results of generating a per-voxel multi-dimensional descriptor using an overlap σ in the range of [1 – 6]. From the results of both synthetic datasets that contain coarse and fine patterns, we found that when $\sigma = 4$ the individual structures were better differentiated and rendered more accurately.

Based on our experiments with synthetic data and different overlaps, we can conclude that a region of 5^3 ($\rho = 1$, $\sigma = 4$) voxels is large enough to capture local statistical textural properties, while at the same time was small enough to accurately differentiate surrounding structures. Note that our results are not a definite solution for the texture size, however we found that when the textural statistical properties are computed in regions between four (4) and six (6) voxels, different patterns and structures can be accurately distinguished and separated despite them containing fine or coarse textural patterns. In

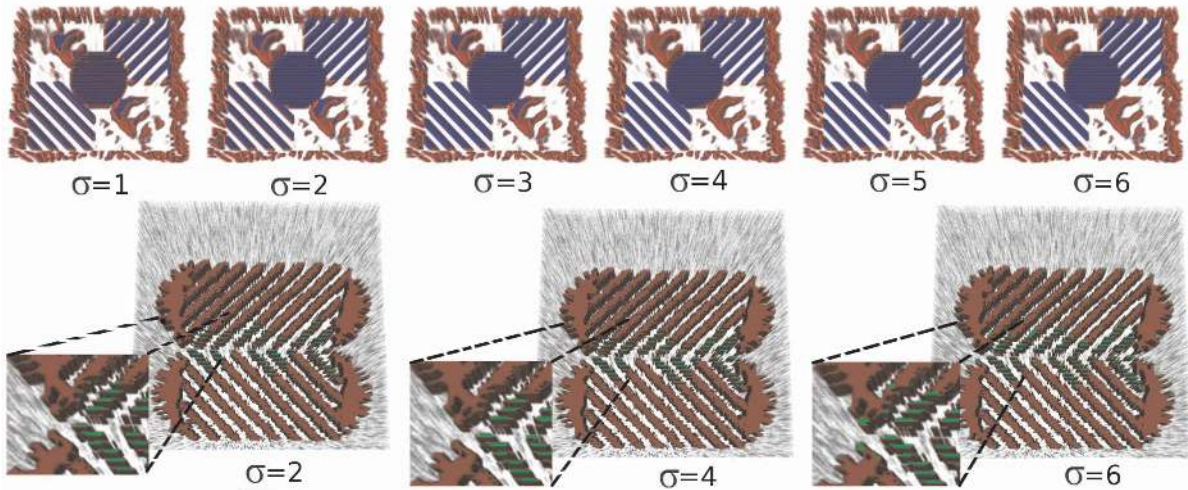


Fig. 5. (Top row) Results of using $\rho = 1$, $k = 3$, and changing $\sigma = 1, 2, 3, 4, 5, 6$. (Bottom row) Results of using $\rho = 1$, $k = 3$, and changing $\sigma = 2, 4, 6$. In both experiments we found that when $\sigma = 4$ (i.e area of 5^3), the data was classified more accurately. Note that with both small and large σ values, there is significant increase on the amount of error and misclassification.

Volume Size	$\rho = 4, \sigma = 2$	$\rho = 2, \sigma = 4$	$\rho = 1, \sigma = 2$
64x64x32	0.15s	0.61s	4.09s
$k = 3$	0.01s	0.10s	0.74s
$k = 4$	0.02s	0.12s	0.82s
64x64x64	0.30s	1.21s	8.33s
$k = 3$	0.04s	0.18s	1.32s
$k = 4$	0.05s	0.20s	1.50s
128x128x64	0.86s	4.57s	32.43s
$k = 3$	0.08s	0.76s	6.25s
$k = 4$	0.11s	0.94s	6.02s
128x128x128	1.33s	8.80s	61.47s
$k = 3$	0.14s	1.46s	12.16s
$k = 4$	0.20s	1.54s	12.93s
256x256x128	5.25s	32.75s	233.88s
$k = 3$	0.62s	6.36s	25.02s
$k = 4$	0.76s	6.50s	27.12s

Table 2. Performance of our texture-based transfer functions. The first row of each section shows the time required to pre-compute the textural properties for different volume sizes, ρ and σ values. The other two rows show the time required to automatically classify the pre-computed vectors as explained in Section 4.3.1.

addition, from our experiments we found that there are not significant and visible advantages of computing per-voxel textural metrics ($\rho = 1$) versus using $\rho = 2$. This is, $\rho = 2$ can be used without any significant reduction in accuracy.

5.1.2 Timing and Performance

The vector image containing all the local textural properties can be either pre-computed or generated during the initialization. To demonstrate the performance of our system and the time required to estimate the statistical textural properties, we have experimented with different volumes sizes and different ρ and σ values. In addition, to show the minimum impact that the optional and automatic classification presented in Section 4.3.1 has on the rendering pipeline, we have classified each experiment with $k = 3$ and $k = 4$.

Table 2 summarizes the results. From the table we can see that the processing time for computing the textural properties increases as ρ decreases and the volume size enlarges given the large number of multi-dimensional descriptors to compute. However, it is important to

note that for the suggested value of $\rho = 2$, $\sigma = 4$, and relatively large volumes, the pre-processing time is about thirty seconds. In addition, from Table 2 we can see that when automatic classification is used, the overhead and time introduced to the rendering pipeline is minimum. In particular, in the worst case scenario that all the multi-dimensional vectors are used, only about five seconds are required to automatically classify relatively large volumes.

5.2 Medical Volumetric Data

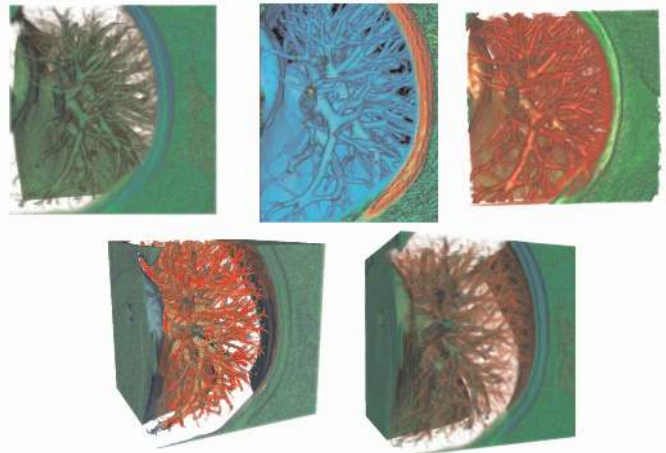


Fig. 6. Visualizations of the lung dataset using various transfer function-based techniques. (Top left) Standard 1D transfer function using VTK. (Top center) Volume rendered image with lighting and gradient-based transfer function to accentuates the boundaries of the vessels. (Top right) Image generated using Simian[9] and multi-dimensional transfer functions to emphasize boundaries between adjacent features. (Bottom left) Visualization generated using our TbTF with $\rho = 2$, $\sigma = 4$, and $k = 3$. (Bottom right) Visualization generated using our TbTF with a slightly modified opacity transfer function, $\rho = 4$, $\sigma = 0$, and $k = 3$.

In addition to synthetic datasets, we have tested our TbTFs with actual CT and MRI data. The experiments with real-world data helped us refine our techniques and strengthen them to deal with the challenges that synthetic data cannot provide.

We first applied our techniques to lung data. Figure 6 shows our

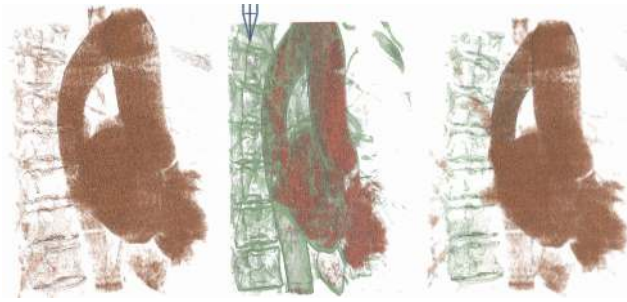


Fig. 7. Images showing a segment of the heart, aorta, and tissue around the spine. (Left) Visualization using a 1D opacity-based transfer function. (Center) Resulting image after applying a multi-dimensional transfer function and carefully picking the the specific materials and boundaries to enhance. (Right) Results of our TbTF with $\rho = 2$, $\sigma = 4$, and $k = 3$ which highlighted individual structures based on their textural properties.

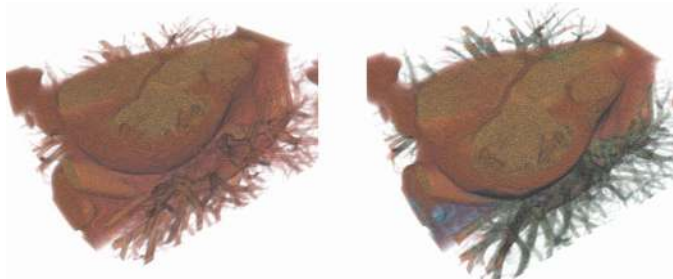


Fig. 8. Experiment of applying our TbTF to different intensity ranges simultaneously. (Left) Visualization obtained using standard transfer functions. (Right) Image obtained using our TbTF to automatically highlight structures with similar intensity values and different textural properties. In this example, the heart is colored in red, adjacent vessels are colored in green, and the tissues surrounding the spine are colored in bluish purple.

results in comparison with other transfer function techniques. The top row shows volume rendered images using a one-dimensional transfer function. The top left image shows a simple transfer function where all the voxels of the same intensity are colored in green. The density based transfer function was not able to disambiguate features in the data. The top center image highlights the boundaries by using lighting and gradient enhancement techniques. The top right image shows the image obtained by using multi-dimensional transfer functions that highlight surface boundaries and allow dual domain interaction. The bottom images show the results of using our TbTF that leads to automatic identification and assignment of colors to the vessels and surrounding anatomy in the lung. Our results were obtained automatically in contrast to the other three techniques which required significant transfer function manipulation. Also it is important to notice that even with relatively large ρ our technique was able to automatically separate different structures.

We also tried our TbTFs on heart data as can be seen in Figures 7 and 8. The heart data was more challenging as visualizing specific anatomical features such as the aorta, the heart, and surrounding structures was extremely hard using standard transfer function or dual-domain interaction.

Figure 7 shows a region of the heart and the aorta along with the tissue around the vertebrae in the spine. The left image was generated using a one-dimensional transfer function. Opacity-based transfer functions were unable to clearly separate the heart's anatomy from the tissue around the vertebrae due to their similar intensities. Using a multi-dimensional transfer function we were able to separate and color the tissue around the vertebrae separately. However, as can be seen the intensity and gradient-based techniques end up picking some features

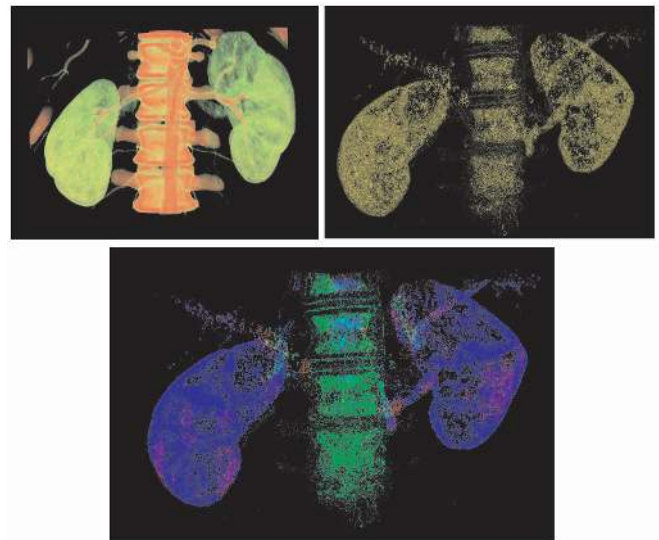


Fig. 9. (Top left) Image obtained by using a combination of opacity- and gradient-based transfer functions. (Top right) By only selecting the voxels with density 118 we can see that existing transfer functions are limited in effectively differentiating individual structures. (Bottom) Local textural properties can be used to improved the accuracy of existing transfer functions. This image was obtained with our TbTF and parameters $\rho = 2$, $\sigma = 4$, and $k = 3$.

co-located with the heart causing a green boundary around the heart. In addition, generating this particular visualization took a considerable amount of time even using the advanced feature of dual-domain interaction, primarily because identifying the exact boundary of the heart anatomy and the tissue was extremely hard. The right image was obtained using our TbTF technique which automatically identified the two anatomical structures separately and colored them differently.

We have also applied our TbTF techniques to more complex scenarios. By using our heart dataset, we experimented with automatically highlighting individual structures from multiple intensity value ranges. Previously, we saw that the tissues surrounding the spine have the same intensity values as the aorta. In addition, the cardiac vessels surrounding the heart have the same intensity values as the heart anatomy. By applying a TbTF to individual intensity ranges we can highlight and differentiate multiple structures at a time. Figure 8 shows our results. From the images we can see that the heart was colored in red, adjacent vessels in green, and the tissue surrounding the spine was automatically colored in blue.

Opacity- and gradient-based transfer functions are limited in differentiating structures when all the voxels have the same intensity and similar gradient values. Figure 9(top left) shows a visualization obtained by using intensity- and gradient-based transfer functions. If we just focus on a particular intensity value of that visualization, we can see the large amount of uncertainty and misclassification introduced by standard transfer functions. Figure 9(top right) shows the set of voxels with intensity value 118. We can see that always within a specific classification and visualization, there are voxels along a particular density range misclassified. By using our TbTF with $\rho = 2$, $\sigma = 4$, and $k = 3$ we were able to automatically highlight and render differently each structures as shown in Figure 9(bottom).

5.3 Weighted Textures

Statistical textural properties have been proven to be strong approaches to characterize similar 2D/3D structures[15]. However, there are situations where structures have the same intensity values and share multiple textural properties, thus reducing the benefits of local statistical textural properties. We found such a limitation when applying our TbTF to brain MRI images, particularly when trying to highlight the part of the cerebellum.

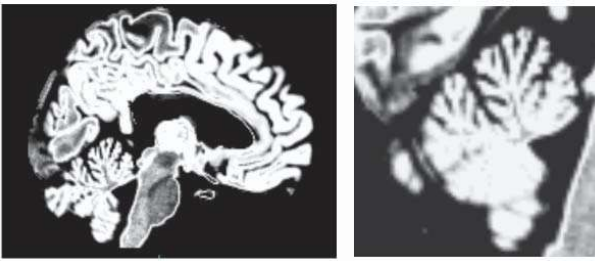


Fig. 10. Images of the gray matter of the brain. It is possible to see that the cerebellum has specific textural properties that can be used to differentiate the gray matter from the cerebellum.

Figure 10 shows a 3D slice of the brain MRI dataset. From the image it is possible to see that there are textural differences between the gray matter and the cerebellum. However, by using a uniformly weighted multi-dimensional descriptor such as the one found in our vector image V' , the structures cannot be highlighted correctly.

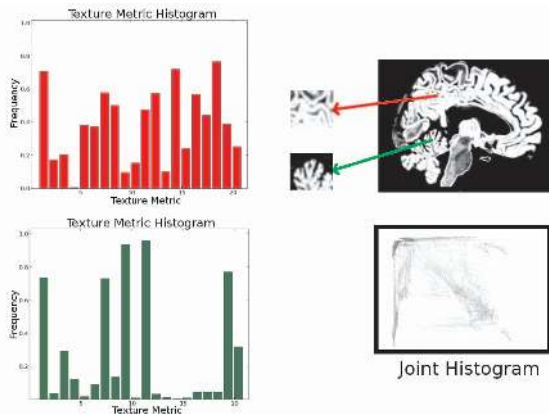


Fig. 11. Example of the process followed to maximize differences between two regions. The user selects two regions, a joint-histogram is generated with the metrics of all the voxels or regions of interest, and individual weights for each textural metric are estimated to better highlight different structures.

In such situations, it is possible to maximize the differences between the structures by assigning individual weights to each textural metric. We have extended our texture-based transfer functions to incorporate weighted textural properties. Our approach works as follows: Given a set of structures that need to be highlighted by our transfer function, small areas within each of them are selected. Then, by combining the statistical textural properties of each of the structures in a joint histogram, joint entropy and mutual information can be estimated. The mutual information measures the amount of information that the structure X conveys about the structure Y .

Such information can then be used to find the textural metrics which are redundant between the structures and most importantly, find the specific metrics which characterize each individual area under consideration. That information is used to estimate a weighting value for each individual textural metric, thus allowing better differentiation of similar structures. In our system, the technique used to generate weights for each metric is based on mRMR[14]. Figure 11 shows the general idea of our weighted texture-based transfer function. Different areas of interest are selected, per-voxel metrics combined, the mutual information computed, and individual weighting values estimated for each textural metric.

In our experiments with MRI data we found that only second- and high-order statistical properties of the data were among the primary metrics to differentiate the gray matter from the cerebellum. In particular, we found that long-run low-gray, gray-level nonuniform, sum of

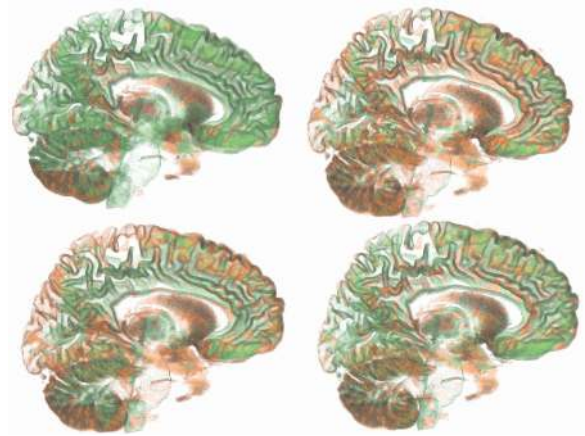


Fig. 12. Results of highlighting individual textural metrics. None of individual textural metrics was able to accurately highlight different structures. (Top left) Energy, (Top right) Sum Entropy, (Bottom left) Absolute Deviation, (Bottom right) Standard Deviation.

entropy, and inertia were the most characteristic features with resulting weights of 0.31, 0.22, 0.21, and 0.17 respectively. However, it is important to understand that none of these individual textural metrics were able to accurately highlight the cerebellum, but only a combination and weighting of them resulted in effective visualization of the center region of the cerebellum.

Figure 12 shows the results of partitioning the volumes with $\rho = 2$, $\sigma = 4$ and highlighting individual textural metrics. None of the individual metrics was able to accurately highlight different structures. Figure 13 compares different volume rendering techniques. The top-left image shows how even multi-dimensional transfer functions are limited in accurately enhancing the cerebellum from the gray matter. In addition, the top-right image shows how the equally weighted TbTFs and automatic classification were not able to effectively illustrate the different structures. Only, the use of weighted texture-based transfer functions were able to enhance and assign different rendering properties of the gray matter and the cerebellum.

Our results with weighted textures should not be taken as a definite solution to extract the cerebellum. In such situations, a segmentation technique might be more suitable. However, when trying to enhance a region of the cerebellum with high textural differences, local texture analysis and weighted textures will work.

6 CONCLUSION

In this paper, we present a novel way to visualize different materials in volumetric data by using texture-based transfer functions. The main advantage of using TbTF is that the user can effectively visualize features with similar voxel properties differently given that the technique is not attached to a specific intensity value, but rather local properties.

The presented system has some limitations. First, the process of estimating all the textural metrics for each voxel or region of interest could be a time-consuming process for relatively large volumes. That is the primary reason why we suggested that for small values ρ and large volumes, the process of estimating and generating the vector image V' should be an offline process. Second, our current implementation and graphical interface shown in Figure 14 does not use the graphics hardware and texture memory to store the vector image V' . In the future, we plan to expand our system so the lookup for the textural properties can be done within the graphics card.

We had considered multi-resolution texture analysis and multi-scale pyramids to better enhance different structures. However, we found that our current classification process does not benefit from such techniques. First, in a pyramid, by reducing the size of the volume or applying smoothing operation to the volume, small features characteristic of particular patterns are lost, thus reducing the accuracy of

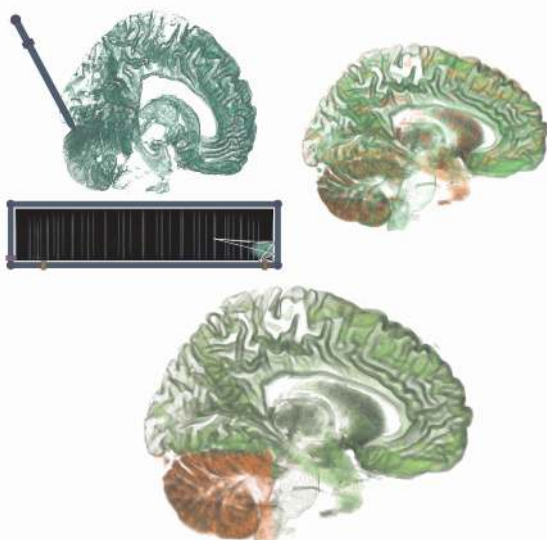


Fig. 13. (Top left) Visualization generated using the dual-domain interaction in Simian. The cerebellum cannot be localized using this technique since it shares intensity and gradient similarities with other regions in the volume, even though it has a unique pattern. (Top right) Results with our uniformly weighted TbTF with $\rho = 2$, $\sigma = 4$, and $k = 3$. (Bottom) Weighted TbTF applied to the brain data. The cerebellum is clearly visualized in red while the rest of the brain structures are seen in green.

the classification process. Second, estimating textural properties in a multi-resolution fashion can reduce the classification given that small volume features might have a very similar descriptor to high-level features. We believe that those two techniques can be used in the future to better classify the data, but special consideration has to be taken.

In our system, we use a combination of first-, second- and high-order statistics to characterize textural features. We found that the enhancement of different features was best performed when metrics were estimated in a 5^3 region. In particular, we found that for $\rho = 2$ and $\sigma = 4$ the computed textural properties led to more accurate feature identification. Our results regarding the textural parameters that more effectively captures local statistics are based on visual estimates. In the future, we plan to more accurately find the texture size and overlap by using a supervised training and testing system.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their comments and suggestions. We would like to thank Alark Joshi for his support and help comparing our technique with previous work. This work was supported in part by NSF grant 0121288 and the 2008-2009 IBM PhD Fellowship Award.

REFERENCES

- [1] S. Bruckner and M. E. Gröller. Style transfer functions for illustrative volume rendering. *EuroGraphics, Computer Graphics Forum*, 26(3):715–724, Sep 2007.
- [2] C. Elkan. Using the triangle inequality to accelerate kmeans. In *International Conference on Machine Learning*, pages 147–153, 2003.
- [3] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 3, pages 610–621, 1973.
- [4] H. Hauser, L. Mroz, G. I. Bisch, and M. E. Gröller. Two-level volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):242–252, 2001.
- [5] M. Hauta-Kasari, J. Parkkinen, T. Jaaskelainen, and R. Lenz. Generalized cooccurrence matrix for multispectral texture analysis. In *13th International Conference on Pattern Recognition*, volume 2, pages 785–789, 1996.

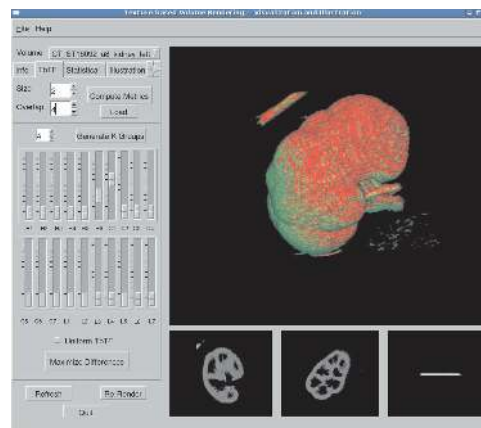


Fig. 14. Changing individual metrics is possible with our graphical interface. The user can pick a threshold of individual metrics, then during the raycast function, if the voxel under consideration is within the range, different opacity and/or color properties are assigned.

- [6] Q. Ji, J. Engel, and E. Craine. Texture analysis for classification of cervix lesions. In *IEEE Transactions on Medical Imaging*, volume 19, pages 1144–1149, 2000.
- [7] G. Kindlmann and J. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of the conference of Volume Visualization, 1998*, pages 79 – 86, 1998.
- [8] G. L. Kindlmann, R. T. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. *Proceedings of IEEE Visualization 2003*, pages 513–520, October 2003.
- [9] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. *Proceedings of IEEE Visualization 2001*, pages 255–262, October 2001.
- [10] V. A. Kovalev and M. Petrou. Texture analysis in three dimensions as a cue to medical diagnosis. In *Handbook of Medical Imaging Processing and Analysis (New York: Academic Press, 2002*.
- [11] M. Levoy. Display of surfaces from volume data. In *IEEE Computer Graphics and Applications*, volume 8, pages 29–37, 1988.
- [12] P. Lung, A. Ynnerman, and C. Lundstrom. Local histograms for design of transfer functions in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1570–1579, November 2006.
- [13] E. B. Lum, J. Shearer, and K.-L. Ma. Interactive multi-scale exploration for volume classification. In *The Visual Computer: International Journal of Computer Graphics*, volume 22, pages 622–630, 2006.
- [14] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 27, pages 1226–1238, 2005.
- [15] M. Petrou and P. Garcia. *Image Processing: Dealing With Texture*. Wiley, 2006.
- [16] M. K. Pietikainen. *Texture Analysis in Machine Vision*. World Scientific Publishing Company, 2000.
- [17] D. Raicu, J. Furst, D. Channin, D. Xu, and A. Kurani. A texture dictionary for human organs tissues’ classification. In *Proc. The 8th World Multi-Conference on References: Systemics, Cybernetics and Informatics*, July 2004.
- [18] S. Roettger, M. B. M., and M. Stamminger. Spatialized transfer functions. In *EuroGraphics - IEEE VGTC Symposium on Visualization*, pages 271–178, 2005.
- [19] Y. Sato, C.-F. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis. Tissue classification based on 3D local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):160–180, 2000.
- [20] X. Tang. Texture information in run-length matrices. *IEEE Transactions on Image Processing*, 7(11):1602–1609, 1998.
- [21] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma. A novel interface for higher-dimensional classification of volume data. In *IEEE Visualization 2003*, pages 505–512, 2003.