VOLUME VISUALIZATION USING SPARSE NONPARAMETRIC SUPPORT VECTOR MACHINES AND HARMONIC COLORS

Naimul Mefraz Khan⋆ *Riadh Ksantini*† *Ling Guan*⋆

⋆ Department of Electrical and Computer Engineering, Ryerson University.
†School of Computer Science, University of Windsor.

ABSTRACT

In Direct Volume Rendering (DVR), the Transfer Function (TF) to map voxel values to color and opacity values is difficult to obtain. Existing TF design tools are complex and non-intuitive for the end user, who is more likely to be a medical professional than an expert in image processing. In this paper, we propose a volume visualization method where the user directly works on the volume data to simply select the parts he/she would like to visualize. The user's work is further simplified by presenting only the most informative volume slices for selection. Based on the selected parts, all the voxels are classified using our Sparse Nonparametric Support Vector Machine (SN-SVM) classifier, which combines both local and near-global distributional information of the training data to obtain accurate results. The voxel classes are then mapped to color and opacity values using the concept of harmonic colors, which provides easily distinguishable and aesthetically pleasing results. Experimental results on several benchmark datasets show the effectiveness of the proposed method.

***Index Terms*—** Volume Visualization, Medical Imaging, Classification, SVM, Color Harmonization

1. INTRODUCTION

Direct Volume Rendering (DVR) is a technique to reveal interesting structures and regions from raw 3D imaging data, typically obtained through popular medical imaging procedures such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) [1]. DVR makes use of a Transfer Function (TF), which maps one or more features extracted from the data (the feature space) to different optical properties such as color and opacity. The TF design is typically a user-controlled process, where the user interacts with different widgets (usually representing feature clusters or 1D/2D histograms) to manually set color and opacity properties to the feature space. In case of clusering-based TF design, the user can control some low-level properties like number of clusters, cluster variance etc. Most of the recently proposed DVR methodologies [1, 2, 3, 4] are based on these basics.

However, interacting with the feature space is difficult for the end-user, who may not have any knowledge about image processing or clustering. Multi-dimensional feature spaces can not represent distinguishable properties such as peaks and valleys which are important for proper TF generation from histogram [5]. Also, these kind of widgets try to represent the feature space directly, putting a strict restriction on the type of features used and the dimensionality.

In this paper, we propose a rather direct approach to simplify the process of volume visualization. Instead of working with complex widgets for histogram or cluster manipulation, the user simply works on the volume data itself. The user is presented with grayscale form of some slices from the volume data, where he/she can do simple selection on voxels to express his/her intention of how the volume should be classified. To further simplify the process, we carefully pick the most representative slices from the volume and only show those to the user. The slices are picked by sorting them based on image entropy, which provides a measure of information present in one slice [6]. Once the user selection is completed, we treat the selected voxels as training data and extract some high-dimensional features. A recently proposed Sparse Nonparametric Support Vector Machine (SN-SVM) [7] is then used to classify the whole volume. This approach combines the local information available through support vectors [8] and the near-global information available through Kernel Nonparametric Discriminant (KND) [9] to provide an accurate high-dimensional classification. Due to the robustness of the classifier, only a small number of training samples can provide excellent results, as we will see from the experiments. After the voxel classification, the classes are mapped to different color and opacity values automatically by using the concept of color harmonization [10], which can generate easily distinguishable and aesthetically pleasing visualization of the underlying classes.

2. RELATED WORK

TF design for volume rendering has been a matter of intense research since the first 1D TF mapping voxel intensity values to color and opacity was published in [11]. The first 2D TF was proposed in [1], where the intensity and gradient magnitude were used to build a 2D histogram. The user manually assigns different color and opacity values by observing the

E-mail address of first author: n77khan@ee.ryerson.ca

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

histogram. Since then, several similar methods have been proposed, where the histogram is calculated based on different measures, such as LH values [12], curvature measurements [13], structure size [14] etc.

The problem with histogram-based TF generation is that the method is restricted to use low-dimensional (typically upto 2D) features, since the feature space is represented directly with the histogram. Moreover, due to the interdisciplinary use of volume rendering, end-users may not have deep understanding of image processing techniques i.e. histograms. Some recent methods try to use various clustering and classification techniques to remedy this problem, such as kernel density estimation [5], mean-shift clustering and hierarchical clustering [2] etc.

Even with these methods, the end user will still have to manipulate low level cluster parameters and assign the color and opacity values manually. Our proposed method approaches the problem from an *image-centric* viewpoint [15, 16], where the user directly performs operations on the most informative volume slices rather than on histograms or cluster visualizations. We also automate the color and opacity assignment based on color harmonization and user's perception. As a result, the user can focus on the interpretation of the result rather than manipulating complex interfaces.

3. PROPOSED METHOD

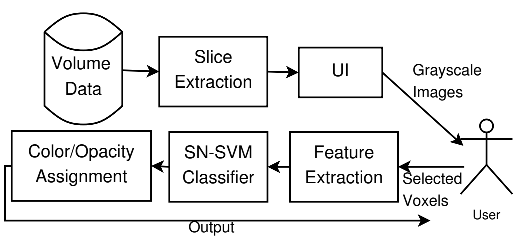


Fig. 1. System diagram for the proposed method.

Figure 1 shows the high level system diagram for the proposed method. As can be seen, the most informative slices of the volume are extracted and presented to the user. The user selects voxels and assigns them in different classes based on what he/she wants to see. Features are extracted from the voxels and used as training data for the proposed SN-SVM classification. The voxel classes are then assigned different color and opacity values based on color harmonization. The various modules of the proposed method are described in the next sections.

3.1. Slice Extraction

Since the user will perform selection operations on volume slices, we need to provide the user with the *most informative* slices. A typical volume can have thousands of slices (along X, Y or Z direction). We need to provide the user with the slices that contain the most variation, since we can safely assume that these slices will contain all the structures that the user might be interested in [6]. For this, we calculate the *image entropy* of each slice. In general, for a set of M symbols with probabilities $p_1, p_2, \ldots, p_M$ the entropy can be calculated as follows [6]:

$$H = -\sum_{i=1}^{M} p_i \log p_i. \tag{1}$$

For an image (a single slice), the entropy can be similarly calculated from the histogram [6]. The entropy provides a measure of variation in a slice. Hence, if we sort the slices in terms of entropy in descending order, the slices with the highest entropy values can be considered as representative slices.

3.2. User Input

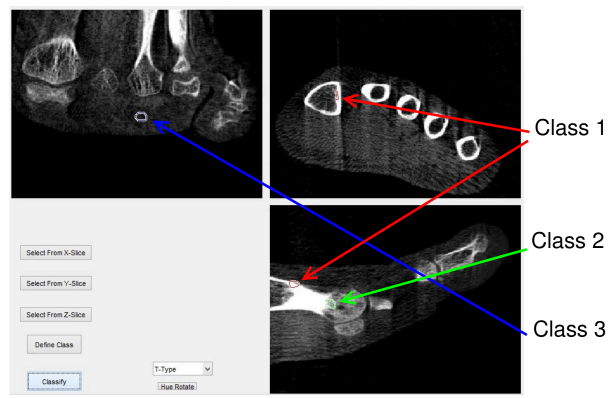


Fig. 2. The GUI with extracted slices. The red, green and blue selections (circles) by the user refer to voxels assigned to Class 1, Class 2, and Class 3, respectively.

The user is presented with a GUI (Figure 2), which shows the three slices with highest entropy values in X, Y and Z direction. The user can then select different voxels and assign them to different classes, which will be treated as training data for the next steps in the method. As we can see from Figure 2, the regions assigned to different classes by the user are marked with different colors. The dataset shown here is the Foot CT (details can be found in Section 4). The objective is to roughly separate the bones (Class 1), joints (Class 2), and the outer layer (Class 3) of the Foot. Please note that although the number of training voxels seems small, our proposed SN-SVM classification method can classify the whole volume from a small training set reliably by combining both local and near-global distributional information [7].

3.3. Feature Extraction

Unlike histogram-based methods, our approach does not restrict the feature dimension. Therefore, we can use a reasonably high-dimensional feature set. The features need to be picked carefully so that in the classifier stage, there is enough distinction between different classes. The intensity value of each voxel is an obvious choice and the most popular feature in histogram-based TF [1]. To emphasize the boundaries between materials [1], we use the 3D gradient magnitude of each voxel as another feature. Finally, we want to capture the local variance in intensity to capture the distinctive properties of each class. As a result, the average neighboring voxel values (in all three directions) are used as other three features. Hence, the 5D feature space consist of:

- the intensity value,
- the 3D gradient magnitude of each voxel, defined by:

$$G = \sqrt{G_x^2 + G_y^2 + G_z^2}, \qquad (2)$$

  where G_x,G_y and G_z are the gradient values along X, Y and Z direction, respectively.

- The average intensity value of the neighboring 8 voxels in X, Y and Z direction, respectively. The three average values along different directions are treated as different features.

These features provide us with reasonable localization of voxel attributes, which helps separating different structures in the volume in the classification stage.

3.4. SN-SVM Classifier

The SN-SVM classifier [7] is motivated by combining the merits of both discriminant-based classifier such as KND [9] and the classical SVM. The KND calculates the within-class scatter matrix by considering the κ-nearest neighbors for each training data point. Thus it considers the "near-global" characteristics of the training distribution. On the other hand, SVM only considers the "local" characteristics (support vectors) to build the separating hyperplane. Both of these sources of information are important for accuracy [7]. In SN-SVM, these two are combined by incorporating the within-class scatter matrix Δ and the between-class scatter matrix ∇ of KND into the SVM optimization problem. Let $\mathcal{X} = \{x_i\}_{i=1}^N$ represents the training data and $\mathcal{T} = \{t_i\}_{i=1}^N$ represents the associated class tags for a two-class problem. We also use the *kernel trick* [8] to map the data points to a higher-dimensional feature space with the function Φ. Then, the SN-SVM formulation can be described by the following optimization problem:

$$\min_{\mathbf{w}\neq 0, w_0} \left\{ \frac{1}{2}\mathbf{w}^T(\eta\Delta(\nabla + \beta I)^{-1}\Delta + I)\mathbf{w} + C\sum_{i=1}^{N} max(0, 1 - t_i(\Phi^T(x_i)\mathbf{w} + w_0)) \right\}. \qquad (3)$$

Here, $\mathbf{w}$ and w_0 are the weight vector and the offset to be optimized. η is the control parameter which dictates the amount of contribution from SVM and KND. By using an appropriate value of η, we can control the direction of the separating hyperplane of the classifier and place it in an optimum way. In [7], we have also shown that the solution provided by SN-SVM is more sparse than the classical SVM, which can be utilized by efficient numerical methods to significantly speed up computation[17].

The value of η is set to 0.3 through experiments in the proposed system. Since the problem can be multi-class, we have used the one-vs-one classification scheme with our SN-SVM method [7].

3.5. Color and Opacity Assignment

Once the classification is complete, different colors and opacity values are automatically assigned to different classes by the system. We use the HSV color space. The H (hue) value in this space determines the actual intensity of the color, while the S and V values respectively determine the saturation and brightness.

To determine the H value, we use a harmonic set [10] which formally specifies the relative position of the colors on a color wheel rather than specific colors themselves. several different templates (T-type, X-type etc.) can be defined. These templates can also be rotated to easily generate a new set of colors in the system. For a dataset, we equally space out the classes on to the selected color template, so that each class is assigned a distinct H value.

We generate the S and V values based on the understanding of the user's perception [10]. Voxel classes that have a small spatial variance occupy a smaller viewing area compared to others and need to be assigned more saturated colors for highlighting [10]. So we calculate the S value for each class based on spatial variance among the class voxels:

$$S_i = \frac{1}{(1 + \sigma_i)}, \quad i = 1, \ldots, Z. \qquad (4)$$

Here, σ_i represents the spatial variance of each class.

Since the rendering results are usually viewed at a distance from the whole volume, we can assume that voxel classes closer to the center of the volume needs to be brighter so that they are not overshadowed by other classes [10]. Hence, The V value can be calculated as follows:

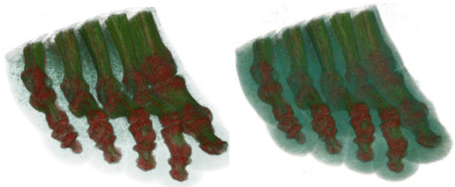
$$V_i = \frac{1}{(1 + \mathbf{D}_i)}, \quad i = 1, \ldots, Z. \qquad (5)$$

Here, $\mathbf{D}_i$ denotes the distance of the centroid of the i–th class to the center of the volume [10].

The last parameter to define the full TF is opacity. Since voxel classes with smaller spatial variances are likely to be obstructed for proper viewing by other classes, we calculate the opacity values based on spatial variances while emphasizing the boundary:

(a) Foot (b) Visual Male Head (c) Engine

Fig. 3. Rendering of the datasets without any classification.



(a) Class 1 & Class 2 (b) All Classes together

Fig. 4. Results for the Foot dataset.

$$O_i^v = (1 - \frac{1}{\sigma_i}) * (1 + F_v), \quad i = 1, \ldots, Z. \qquad (6)$$

Here, σ_i is the spatial variance of the i–th class. F_v is the *Gradient Factor* which helps boundary emphasis:

$$F_v = \frac{G_v}{max_{v \in \mathcal{Z}} G_v}. \qquad (7)$$

Here, $\mathcal{Z}$ denotes the class that the voxel v belongs to. G_v is the 3D gradient magnitude for each voxel (Equation (2)).

These values are converted into *RGBA* texture and passed on to the rendering stage. The Visualization Toolkit (VTK) [18] is used for fast GPU computed rendering.

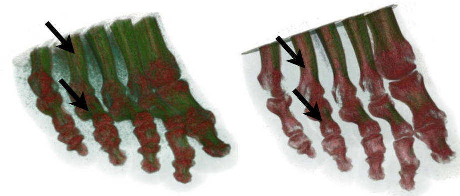
4. EXPERIMENTAL RESULTS

We provide results on three CT scan datasets [19]: Foot, Visual Male Head and Engine (1). To speed up the classification process, we threshold these datasets with a value close to zero so that the air surrounding the actual data are not passed on to the classifier.

Figure 3 shows an intermediate rendering of the datasets with arbitrary colors after thresholding. We can see that no inner structures are visible before classification. These images are presented to provide certain context to the results after classification.

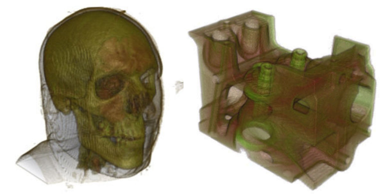
Figure 4 shows the result obtained based on the training data shown in Figure 2. Here, we see that our SN-SVM method can separate the bones, joints and the outer layer of the foot effectively. Figure 4-(b) shows that due to the use of intelligent color and opacity assignment, all three classes can be visualized at the same time and easily distinguishable.

Figure 5 shows comparison between SN-SVM and the classical SVM (based on the same training data). The bones and joints are shown together. We can clearly see the advantage of SN-SVM. In the areas pointed by arrows, the SVM was unable to accurately separate the joints from the bones, while the SN-SVM method was successful. This shows the superiority of the combined approach in SN-SVM. Although the SN-SVM result may look noisy in some areas, the target here is to show the effectiveness in separating the bones and joints. The apparently cleaner output from SVM can actually mislead the user in thinking this is accurate.



(a) SN-SVM (b) SVM

Fig. 5. Comparison between SN-SVM and SVM.



(a) Visual Male Head (b) Engine

Fig. 6. Results for Visual Male Head and Engine dataset.

Figure 6 shows the other two datasets. In both cases, two classes of voxels were defined. We can see that the inner structures are easily separable with the proposed method.

Dataset	Size	# Training Samples	Total Time
Foot	256X256X256	401	12.33
Vis. Male	128X256X256	233	2.27
Engine	256X256X256	180	1.07

Table 1. Dataset details and required times (in *seconds*).

Table 1 lists the training sizes and times required for the SN-SVM method (both training and classification). We see that, even with an small number of training samples, the results obtained are accurate and quick.

5. CONCLUSION

In this paper we have proposed a new image-centric volume visualization approach where the user directly interacts with the data to select interesting structures. Treating the user input as training data, the SN-SVM classifier combined with the concept of color harmonization can generate accurate output showing easily distinguishable structures with aesthetically pleasing colors. Experimental results on several datasets have shown the effectiveness and efficiency of the system.

6. REFERENCES

[1] G. Kindlmann and J.W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," in *Proceedings of the 1998 IEEE symposium on Volume visualization*. 1998, pp. 79–86, ACM.

[2] Binh Nguyen, Wei-Liang Tay, Chee-Kong Chui, and Sim-Heng Ong, "A clustering-based system to automate transfer function design for medical image visualization," *The Visual Computer*, vol. 28, pp. 181–191, 2012.

[3] Yunhai Wang, Wei Chen, Jian Zhang, Tingxing Dong, Guihua Shan, and Xuebin Chi, "Efficient volume exploration using the gaussian mixture model," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 1560–1573, 2011.

[4] M. Selver, M. Alper, and C. Guzeli, "Semiautomatic transfer function initialization for abdominal visualization using self-generating hierarchical radial basis function networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 395–409, May 2009.

[5] R. Maciejewski, I. Wu, W. Chen, , and D. Ebert, "Structuring feature space: A non-parametric method for volumetric transfer function generation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 1473–1480, 2009.

[6] C. Studholme, D.L.G. Hill, and D.J. Hawkes, "An overlap invariant entropy measure for 3D medical image alignment," *Pattern Recognition*, vol. 32, pp. 71–86, 1999.

[7] N.M. Khan, R. Ksantini, I.S. Ahmad, and L. Guan, "SN-SVM: A Sparse Nonparametric Support Vector Machine Classifier," *Signal, Image and Video Processing*, 2012.

[8] M. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK, 2000.

[9] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.R. Mullers, "Fisher Discriminant Analysis with Kernels," *Neural Networks for Signal Processing IX*, pp. 41 –48, aug. 1999.

[10] Y. Liu, C. Lisle, and J. Collins, "Quick2insight: A user-friendly framework for interactive rendering of biological image volumes," in *IEEE Symposium on Biological Data Visualization*. 2011, pp. 1–8, IEEE Press.

[11] M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29–37, May 1988.

[12] P. Sereda, A.V. Bartroli, I.W.O. Serlie, and F.A. Gerritsen, "Visualization of boundaries in volumetric data sets using lh histograms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 208–218, May 2006.

[13] G. Kindlmann, R. Whitaker T. Ross, T. Tasdizen, and T. Moller, "Curvature-based transfer functions for direct volume rendering: Methods and applications," in *IEEE Visualization*. 2003, pp. 513–520, IEEE Press.

[14] S. Wesarg, M. Kirschner, and M.F. Khan, "2d histogram based vol-ume visualization: combining intensity and size of anatomical structures," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, pp. 655–666, May 2010.

[15] F.-Y. Tzeng, *Intelligent system-assisted user interfaces for volume visualization*, Ph.D. thesis, University of California, 2006.

[16] J. Kniss, G. Kindlmann, and C. Hansen, "Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets," in *IEEE Visualization*. 2001, pp. 255–262, IEEE Press.

[17] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351 – 1362, june 2005.

[18] W. Schroeder, K. Martin, and B. Lorensen, Eds., *The Visualization Toolkit*, Kitware, New York, NY, USA, 2006.

[19] Stefan Roettger, "The volume library," Ohm Hochschule Nurnberg, 2013.