IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 4, April 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304169 |

Reviving Monochrome Memories: Automated Image Colorization website for Monochrome Images using OpenCV in Python

Anto Ajay Sharma A, Azeem R, Chandan Rishi S, Rajapriya N

U.G. Student, Department of Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli, India

U.G. Student, Department of Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli, India

U.G. Student, Department of Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli, India

Assistant Professor, Department of Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli, India

ABSTRACT: With direct volume rendering, a novice or inexperienced individual still takes much time to expectantly control the transfer function for the analysis of the volume data. Some studies aimed that the use of deep learning techniques in volume rendering would reduce the difficulty. However, present work cannot apply out-of-the-box with current DCR pipelines. In this work, we propose an image-simulated TF colorization with CNN, which automatically generates a direct volume rendering image similar to a specified target image. The premise system consists of CNN model training, TF labeling, image-based TF generation, and volume rendering performed by image to match the target. The study presents the technique of training CNN and labeling TF's images which are similar to the input volume dataset. We also obtain primary

INDEX TERMS: Volume Rendering, CNN, TF colorization

I.INTRODUCTION

A technique known as direct volume rendering (DVR) is commonly used to display 3D volume data by projecting it onto a 2D plane. To enable the creation of direct volume-rendered images (DVRIs) from the discrete volume data sets, a camera is placed in an environment where there is 3D volume data and each voxel has been assigned different colors or opacities. In this way, the renderer firstly maps the 3D space composed of voxels and seen through the camera lens into a regular grid by means of direct volume rendering techniques such as ray casting. A transfer function (TF) refines optical properties of voxels within the VR during this process. Then, the user moves the TF for coloring and making opacity changes in respect to each voxel being observed by them. Fine-tuned TF helps effectively represent type of information carried by given voxels and their domain range that allows exploring preferable parts in those volumes efficiently.

The most popular two-dimensional transfer function is the Intensity-Gradient Magnitude (I-GM) TF. In I-GM TF, the x-axis of 2D histogram corresponds to volume intensity and the y-axis corresponds to gradient magnitude. With increasing hardness of a medium, its value increases in the case of volume intensity which represents its medium properties. The magnitude of change shows how much a medium varies. For instance, within the same material, the gradient magnitude is low because it has relatively uniform intensity.

Thus, due to these features such as intensity and gradient strength, arc patterns are displayed connecting smallest and largest intensities between similar media for IG-MT TFs. Therefore, users find arches that make up parts of 3D objects or material boundaries and assign optical properties on them in a VRML format file. Sometimes though these arches may overlap each other within a 2DTF; thus will become impossible to differentiate between different media on an I-GM produced image.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304169 |

The right area of interest is to be selected as one adjusts the TF for volume data and this requires an understanding and know-how of the TF but may take a considerable time for a user to specify the appropriate TF setting. As such, there have been studies carried out until recently targeting designing interaction with TFs that use various attributes obtained directly from volumetric data. It can also be seen in the paper by Maciejewski et al., where they propose an attribute space visualization approach of relationships between rendering and volumetric data attributes. This model groups voxels into several clusters which are defined by their attribute space specified by the user. For every cluster, DVRI is generated through rendering using semi-automatically generated TF based on attribute space, representing it's characteristic. In this work we introduce a clustered-TF-based technique for obtaining DVRIs in attribute space inspired by Maciejewski et al.'s work, and classify them with CNNs.

Many a research field has seen the application of state-of-the-art deep learning techniques whereby a model is generated by combining neural network architectures with data. Some volume rendering tech- niques encompass deep learning methods such as volume segmentation , viewpoint estimation , transfer function , lighting , and quality improvement . These experiments utilize Convolution Neural Network (CNN) and Gan, otherwise known as Generative Adversarial Net . However, CNN-based techniques are only applicable for indirect volume rendering (IVR) which can hardly be used in DVR, while GAN-based works take long training time and may have distorted data.

In this work, we suggest to use an image based TF through CNN to generate a DVRI similar to the target image. This paper illustrates our method as illustrated in figure 1. The target image is one way to overcome the problem of manipulating Transfer Function in DVRs. The term "target" refers not only to images made up of DVRIs but also includes actual pictures such as photographs.

In other words, the user not familiar with volume rendering first guesses the name of volume dataset, estimates the data shape of it, and then infers the relationship between the rough shape and target image. While the TF is adjusted in the volume renderer, the user sees TF of adjacent many layers but does not find the TF that shows the target image. As a result, the user sets the target image that can be achieved with anywhere using the keywords from the first guess of the volume dataset name and the second guess of the rough shape of the volume.

II.RELATED WORK

Direct volume rendering (DVR) is a rendering technique by specifying the color and opacity of each voxel in the transfer function (TF). Since setting appropriate TF in DVR increases understanding of volume data and improves visual cue, many TF techniques have been studied to set proper TFs in the volume data domain. Drebin et al. [10] have proposed a direct volume rendering (DVR) with a 1D histogram TF. He et al. have used the 1D TF which classifies voxels based on their scalar intensity values. and Li et al. have introduced the classification of volume rendering by applying a 1D TF to an automobile CT image. The 1D TF is the most common transfer function technique employed in various systems. However, when trying to separate voxels with different features, there are many cases where the intensity ranges of voxels overlap. Therefore, it is limited to distinguish and identify features with only 1D TF

For the overcome of limitations by the 1D TF, 2D or higher dimensional TFs have been proposed by employing additional parameters including scalar value and gradient magnitude, color distance gradient, curvature, spatial distances to boundaries , occlusion spec- trum , and view-dependent occlusion . The high dimensional TFs allow us to distinguish the volume fea- tures more efficiently and achieve better volume render- ing images compared to the 1D TFs. As more parameters are added to various TFs, it becomes possible to refine the volume data features. Typical multi-dimensional trans- fer functions (MDTFs) include the predefined probability distributions, LH domain for material boundary iden- tification , peak values using RBF network , non- parametric kernel density estimation , statistical TF , attribute space , and stochastic neighbor embedding-based dimension reduction. In these studies, multi-dimensional domain properties are employed to the MDTFs, which in- creases the likelihood of separating features within the vol- ume data and enhances the visual cue in the rendering image. However, the MDTFs require much effort to designate suit- able TFs embedding domain characteristics because of the complexity of user interactions as the number of attributes increases.

Unlike the techniques that directly increase the dimension of TF, some studies improve TF manipulation or introduce new TF manipulation techniques. Castro et al. have presented a TF classification technique by coupling metadata with 1D TF to select high-level components, including bones, brains, and muscles. Kniss et al have introduced a set of widgets and interaction tools for the MDTF. Moreover, var- ious interaction techniques have been proposed. Especially, parameterized primitive sets, sorted histogram, fus- ing multiple features, textural metrics, anatomical structure, feature

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304169 |

size-based TF , parallel coordi- nates plot , WYSIWYG , hierarchical modified dendrogram, attribute space exploration , and spatial connectivity of boundaries have been presented.

Instead of manipulating TFs, TFs are also automatically generated in DVR. Fujishiro et al have proposed an automated TF utilizing the Reeb graph topology analysis. Sereda et al. have introduced an automatic classification model by computing the similarity between the LH space and volume. Pfaffelmoser et al. have visualized the uncertainty of the volume surface with the spatial deviation of the surface in the automatically selected TF. Bramon et al. have presented an information weight model to which the information theory is applied to define multiple TFs automatically. Ma and Entezari have shown a technique to identify FOI (Feature of Intensity) utilizing the cell-based feature similarity and automatically generate the TF based on
the selected feature.

Some studies enhance the performance of data analysis by employing machine learning techniques for the transfer function. De Moura Pinto and Freitas have proposed an MDTF design by reducing the dimension of TF space with the self-organizing map. Soundararajan and Schultz have compared five supervised classification techniques, in- cluding Gaussian Naive Bayes, *k* Nearest Neighbor, Support Vector Machines, Neural Networks, and Random Forests for the stochastic TF design. Quan et al. have introduced a high-dimensional feature model using hierarchical multi- scale 3D convolutional sparse coding as a voxel classification technique.

Neural networks facilitate the retrieval of solutions to complex problems by repeatedly fitting the networks to data through multiple neural network structures. Convolutional Neural Network (CNN) and Generative Adversarial Nets (GAN) [53] have been employed to enhance the TF manipu- lations. Cheng et al. have used CNN for the volume visu- alization of complex high-dimensional information as the TF study, where they have divided the data into 65x65 patches and trained the CNN with the ADADELTA solver. 200-dimensional features are extracted, and the volume is visualized with the conventional marching cube algorithm for specific high-dimensional feature values. However, since this approach is limited to indirect volume rendering by marching cube, it is not easy to extend this technique to the transfer function in the DVR. Berger et al. have introduced a volume rendering technique utilizing GAN to compute a model from a large set of volume rendering images at specific viewpoints and certain transfer functions for the opacity and color. Hong et al. have trained a deep neural network combined with GAN and CNN using volume rendering im- ages. Their system synthesizes the volume rendering images with goal-effect rendering. This technique enables volume data browsing without an explicit transfer function.
Although there have been many machines learning tech- niques for the volume rendering, we still find difficulty in producing the rendering images from volume data automati- cally according to the rendering preference. Our work in this paper combines the deep learning technique, TF manipula- tion, and user preference setting for DVR as an automatic DVR pipeline.

III.CNN MODEL TRAINING

CNN model training is a foundation step for applying CNN to the volume rendering. In this step, we label training datasets and train the CNN network. Figure 2 presents the training process for CNN. We build image datasets by crawling images on the web that are similar in the shape and context to the volume data. We divide the image datasets into smaller image patches and label them to generate labeled image patches. The system then trains our CNN network with the generated labeled image patches. The trained CNN is used to classify direct volume rendering images

A. IMAGE DATA COLLECTING

In general, labeled data are required to complete supervised learning. In this section, we introduce how we build labeled image datasets representing the shape and context of the volume data. When rendering a volume dataset, we usually render the overall shape of the data and manipulate the TF to color the voxels. During this procedure, we deduce keywords related to the shape and name of the volume data. We also determine labels with the keywords that represent prominent features in the volume rendering images. The labels are fictitious names assigned only by the overall shapes of the volume rendering image. For the bonsai dataset, for example, we begin with *leaf, stem, soil,* and *pot* as the labels. There- fore, we create the keywords as *bonsai, bonsai leaf, bonsai stem, bonsai soil,* and *bonsai flowerpot* and crawl the image datasets on the internet with these keywords. However, the crawled image datasets include images that are inappropriate for the training, such as graphic edited images, illustrated images, images with watermarks, and images irrelevant to the context. Therefore, we create a set of raw images after removing images that are inappropriate for the training. The raw images contain various objects together. Therefore, in order to classify the raw images into the labels, we divide the raw

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304169 |

images into 32 * 32 and 64*64 patches and classify the patches into the labels. If the patch size is smaller than 32*32, there are too few features, which may interfere with training. If it is larger than 64*64, the patch may contain multiple features.

B. LABELED DATASET GENERATION FOR CNN

For the CNN model training, we generate the training datasets. The objective of this paper is to label the transfer function (TF) using CNN to generate rendering images sim- ilar to the target image colors. Although the main goal of this paper is not the image labeling, the labeled dataset for the training is essential in the deep learning research. Many public datasets are intended to provide labels as creatures or objects, such as a human, dog, cat, or car. However, for the volume rendering, it is necessary to label the data from a more anatomical point of view. For example, it is necessary to label a plant image by dividing it into leaves, stems, and pots in the image, not the plant itself. Since it is not easy to obtain labeled volume image data, we divide the images into smaller patches and classify them manually to evaluate our study. In order to generate a labeled dataset, we develop a labeling system. We have built the labeling system with Python, PyQt5, and Keras, which divides the meaningful parts of an image into patches and enables us to assign labels to the patches simply by mouse clicking & dragging

C. CNN ARCHITECTURE AND TRAINING

Convolutional neural network (CNN) is a class of deep learning mainly utilized to recognize images. As a CNN architecture, many well-designed networks such as AlexNet [56], VGGNet [57], GoogLeNet [58], and ResNet [59] have been studied. In particular, the VGGNet is applied to the volume rendering by Shi and Tao [6] to demonstrate its superiority when measuring the image recognition accuracy. The VGGNet architecture increases the layer depth by overlapping the 3×3 convolution layer multiple times. Since it has fewer parameters than large-sized filter like 5×5 or 7×7, the VGGNet increases computational efficiency and accuracy. In this work, we design the CNN model by adjusting the layer size by referring to the VGG architecture

We modify the architecture and parameters of the VGGNet. We utilize the previously labeled 32×32 or 64×64 patch images for the input to the CNN. The 32 × 32 patch is resized to a 64 × 64 patch to apply the same architecture. The ReLU is used as an activation function, and the batch normalization is performed after every convolution and activation to avoid the data skewness. Moreover, the system gradually reduces the neurons with the 2×2 maxPooling after each stack of convolutional layer is performed. Finally, the neurons are flattened in one dimension and train the network according to the number of labels in the dense layer and the sigmoid activation layer. The trained network labels the input image patches.

We train our deep learning network using labeled patches. However, if the number of images is different for each label, the deep learning network tends to be biased. Therefore, we randomly sample the labeled patches with the same number to generate a representative set of each label. Then, the patches of each label are randomly divided into a training set and a test set to train the deep learning network. Note that 70% and 30% of the patches are used as the training set and the test set, respectively.

We check the performance of training a CNN model with the labeled data generated from 500 bonsai images. Figure 5 shows the loss, accuracy, validation loss, and validation accuracy during 35 epochs. The training time is 3,512 seconds and the accuracy is 98.48 %.

IV. VOLUME RENDERING TECHNIQUES TO REFLECT TARGET IMAGE'S COLOR

In volume rendering, the transfer function setting affects the rendering quality, such as the visibility of valuable information within the volume data. In general, to acquire direct volume-rendered images (DVRIs) from discrete volume data, a camera is placed in a space where 3D volume data exists, and sets of color and opacity are assigned to all voxels. Then, the renderer visualizes the 3D volume data with the camera and voxel information by applying direct volume rendering techniques such as ray casting. In this process, a transfer function (TF) is manipulated to determine the optical characteristics of voxels within the volume renderer. The user then interacts with the TF to adjust the color and transparency of a voxel. Properly manipulated TF can efficiently convey data type and domain characteristics and help the user explore the area of interest in the volume data. Figure 6 (a) presents the 2D transfer function based on the intensity value and gradient magnitude (I-GM TF). In order to visualize the volume data, a user adjusts the TF colors, as shown in Figure 6 (b), and render the data with the colors. Figure 6 (c) displays the direct volume rendering image (DVRI) with the TF in (b). Although the proper TF allows the user to achieve the desired visualization, the user still

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304169 |

needs to adjust the TF repeatedly to explore the volume data until the best TF is discovered. In this study, we propose a novel volume rendering system resembling the target image to shorten the learning curve for the TF setting.

V. RESULT AND DISCUSSION

Our system is built on Intel i7-10700 CPU with 32GB RAM, and Nvidia RTX 2080 Super with 8 GB VRAM. We implement the deep learning module using Python and Keras and integrate it with the volume renderer. The volume renderer is built using Qt and OpenGL with shaders. In order to evaluate the usability of the proposed imagebased TF colorization with CNN, we recruited five novice participants for an experiment, including four students from computer science, and one from the medical field. Since all participants do not know about the volume rendering, we first gave them necessary information about the direct volume rendering and taught them how to use our volume renderer. We provided the participants with the bonsai image as a target image, as seen in Figure 17, and measured the time until they have obtained the volume rendering image similar to the target image. The participants explored the bonsai volume data while looking at the target image. They adjusted the TF and assigned the colors in the rendered image. We requested that the participant informed us when further manipulation was considered pointless as of completion. Figure 17 (a) displays five DVRIs manually generated by the participants with bonsai volume dataset. Although all participants selected the purple to render purple leaves, we can see that the colors recognized by all participants are different. User1 rendered the stem, soil, and moss by mixing colors, and User5 gave up the pot color to better represent the moss.

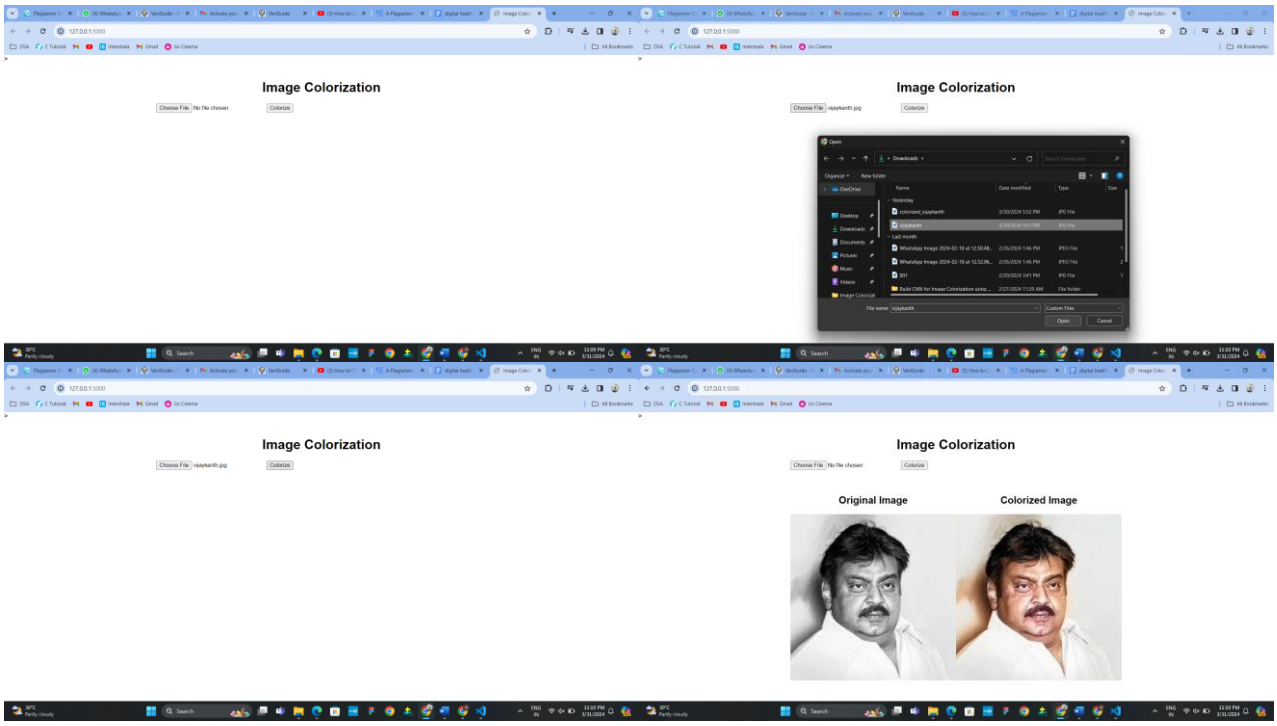
Although all participants rendered bonsai data using the same target image, all participants assigned different colors to the data. The rendering time varies from 10 minutes 24 seconds to 47 minutes 46 seconds, and the average rendering time is about 25 minutes 7 seconds. We interviewed all the participants about the difficulties in the volume rendering after the experiment. User 1 stated, "After dividing the TF into multiple rendering areas, I tried to assign colors. But I could not remember which TF area indicates which part of the rendered image, hence it took a long time to check." User 2, 3, 5 mentioned, "I wanted to divide the TF into smaller ones as I want, but many volume parts were sharing the same TF area, so I had to compromise." User 2, 4, 5 talked about the difficulty in selecting colors and said, "It is difficult to choose colors with proper saturation and contrast. It seems to require aesthetic sense for the color selection." Figure 17 (b) shows the DVRIs for the carp dataset. Different rendering results were produced from the target images because each participant estimated the color values subjectively and had a different region of interest in the volume. After rendering the carp dataset, all participants said, "It was obvious to distinguish carp bones, but it was tough to distinguish between body and bladder." User1 and User4 said, "The body and bladder seem to be located at the same position in the I-GM TF, and I gave up after inspecting the TF by 1 pixel." User2 mentioned that "I checked the shape of each TF area and assigned a color to each area, but when all the selected TF colors were applied, the desired color was not well reproduced because other voxels obstructed the voxels that I want to represent." User3 remarked, "After coloring the bladder, I rotated the volume and found that different part of the body was being rendered with the bladder color at the same time," and stated his experience with perception distortion caused by projecting a 3D space into a 2D image. Figure 17 (c) shows the DVRIs of the X-mas tree dataset. The participants rendered the X-mas tree dataset with the snow-covered pine tree, red ball, and green bottle as the target images. Since all parts of the X-mas tree are made of similar materials, separating each shape with TF alone is not straightforward. All participants pointed out that the decoration objects are indistinguishable and said, "The tree and decorations are separable, but it is difficult to distinguish different decorations such as bottles and balls." User2 revealed the difficulty of distinguishing object boundaries and three-dimensional recognition, saying, "After applying the color to the red ball, the red voxels were hidden and disappeared when a color was applied to the tree." User4 talked about the sensitivity of TF manipulation, saying, "I changed the TF only a little bit, but the colors of all voxels changed." Our approach took less than 4 minutes to render the images. The color scheme of our model is not inferior to the manual rendering images at all, and it was possible to render the volume data faster than the manual rendering by the participants. However, there were some differences in the case of the x-mas tree. Users explored the entire volume despite the difficulty of separating decorations from each other and forcibly divided them into green and red by personal criteria. In contrast, our approach did not distinguish them explicitly and mingled colors in the rendering. In addition, the participants represented the tree in white color, but our approach rendered the tree in a light blue color. It is interpreted that when people perceive colors, the general knowledge they learned beforehand influenced color perception. On the other hand, since there is no information about "snow is white" in our approach, the rendering color is determined by calculating the color values on the target image. We applied various target images to the bonsai volume data to see how differently rendering images are generated according to different target images. Figure 18 shows the colorized volume rendering images generated with three different target images. (a), (b), and (c) show the rendered images with purple maple, orange maple, and green pine tree as the target images, respectively. We classified the target image features into leaf, stem, and pot to render the bonsai data, and our system determined the feature colors.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304169 |

As seen in the figure, the colors in the rendered images are well-matched with the feature colors of the target images. We applied the CNN model to the carp data for the transparent rendering and multiple target images. We first trained CNN by labeling the carp with the body, bone, and air bladder features. During the bonsai data rendering, since different features were found within one target image, it was possible to render the data with only one target image. However, it was difficult to obtain the target images that contain both internal structures and an external shape at the same time. Therefore, we selected different feature images as the target images. Figure 19 (a) presents three target images to render the carp data, including the bone, air bladder, and carp shape images. The CNN model determined each feature color, which is called the label colors in the figure. Once our system automatically generated the rendered image with the colorized TF, we adjusted the global opacity separately to show the inside and outside of the volume simultaneously.



As seen in the image, we can observe that the hard object, which is the man, was explicitly rendered. However, the soft objects were not well separated, which indicates that the bladder and body share similar data values. We will study how to separate these features in the future. Moreover, since the global opacity setting is operated manually, we will investigate optimal opacity values for features by synthesizing the information of the target images as future work. Figure 19 (b) shows the target images, label colors, and rendering images for the automatic X-mas tree rendering. We trained CNN by labeling the X-mas tree with the tree, ball ornament, and bottle ornament features. We applied three target images. In the our approach, we can see that the tree is rendered with the snow color automatically assigned by the CNN model, but the ornaments have mixed colors of all three label colors. Our deep learning model determines the colors by matching the object shapes of the data with the object shapes in the target image. However, the rendering system separates the material features and renders the data accordingly. The ball and bottle ornaments of the X-mas tree data cannot be separated by our TF alone. Therefore, our approach has a limitation in that the colors may not be distinguished if the target image is selected based on the shapes without considering the material features. This problem is also the limitation of the I-GM TF. Datasets that are difficult to classify by the I-GM TFs can be classified with multidimensional TFs or classifying voxels in volume space rather than projected space. However, this paper aims to lower the hurdles for rendering for beginners, so we avoided either applying multidimensional TFs that required complex interactions or approaches that compromised the underlying volume pipeline. We are planning a study using multidimensional TFs as a future work. Table 1 presents the computing time of the CNN model and rendering. We performed the target image-based colorized volume renderings for the bonsai, carp, and X-mas tree datasets. The computing times were measured separately for the CNN model training time, M, and the TF labeling and TF generation time, T F. The CNN model training time includes labeling time, Mlabeling, and training time, Mtraining. The Mlabeling is the time it takes to create labeled patch images from the crawled images introduced in Section III-B. The Mtraining is the time for training the CNN model with the labeled patch images. Note that M is not necessary if CNN model is already trained within the

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304169 |

system. T F is composed of T Fextract and T Fcoloring. T Fextract is the time to extract major colors from target images. T Fcoloring is the time to label TF and map major colors. As seen from T F in Table 1, our system performs the image-based TF colorization within less than 4 minutes. Since the T F is preprocessing operations, it does not affect the performance of direct volume rendering. Table 2 shows the performance of deep models, including VGGNet, ResNet, AlexNet, and LeNet, on the bonsai, carp, and x-mas tree datasets, respectively. VGGNet, ResNet, and AlexNet have accuracy above 0.99 and loss below 0.015. However, the validation loss varies for deep learning models and datasets. VGGNet produces consistently good performance for all datasets. Also, VGGNet has short computation times compared to ResNet and AlexNet. However, since the experiment was evaluated with specific datasets, an experiment with more data is necessary. Therefore, we plan to compare deep learning architectures with more datasets in the future. Also, to improve the model performance, we will investigate how the depth of deep learning architecture affects the performance. In our work, it is possible to reuse the pre-trained CNN model for domains similar to the previously trained domains. However, the CNN approach for different domains demands to retrain the CNN model with new labeled images according to new data domains. To overcome this limitation in labeled data acquisition, we developed a collaborative data labeling system and a semi-automatic labeling procedure as a learning process for a new domain. For the training data labeling, as mentioned in Section III-B, it took about 15 seconds per single image with our in-house software. We found 2,764 bonsai images obtained on the Internet, and the time it took for one person to label all images manually was about 12 hours. The collaborative labeling system took an hour and 46 minutes when ten people labeled the bonsai images together. Although the collaborative system was efficient in reducing the time for the data labeling, the manual data labeling is still cumbersome. Different approaches to avoid manual labeling could be semi-automatic learning and fewshot learning. For the semi-automatic labeling approach, 100 images are labeled manually, and then the labeled images are used to train any CNN model. Afterward, new images are labeled with the CNN model, and a human just reviews the labels and corrects only the wrong labels. Although the initial performance of semi-automatic learning is somewhat poor, if it is combined with the collaborative system, it is possible to obtain a stable model that gradually improves performance. As another approach, we plan future work for a meta-learning-based method that learns metadata constructed from only one image and a method that expands the learning data using a GAN that has learned a target image.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we proposed an image-based TF colorization with CNN to automatically generate a direct volume rendering image (DVRI) similar to a target image color. We created labeled datasets from internet images and trained the CNN model. We defined various TFs, including Grid-TF, AS-TF, and Class-TF, to label the TF using information entropy and a voting technique with Borda count scores. We also extracted label colors from the target image with the CNN model to create a colorized TF for the final volume rendering. For the usability evaluation, we performed an experiment with novice participants and compared manual volume rendering by the participants with our automatic rendering image. We used the bonsai, carp, and X-mas tree volume datasets to evaluate our rendering pipeline and system. However, our approach employs supervised learning, which has the disadvantage that the labeling time might be significantly longer. Besides, there is a limitation in that it is difficult to apply different feature to objects that are not distinguishable in the I-GM TF space. Therefore, we plan to multidimensional TF approach and apply the automatic labeling technique or few-shot learning techniques as future work. Also, we are planning semi-unsupervised learning-based volume rendering in which GAN and CNN are associated as a technique of training a single target image without using multiple labeled datasets in the future.

REFERENCES

[1] S. Arens and G. Domik, “A survey of transfer functions suitable for volume rendering,” in Proceedings of the 8th IEEE/EG international conference on Volume Graphics, pp. 77–83, Eurographics Association, 2010.
[2] P. Ljung, J. Krüger, E. Groller, M. Hadwiger, C. D. Hansen, and A. Ynnerman, “State of the art in transfer functions for direct volume rendering,” Computer Graphics Forum, vol. 35, no. 3, pp. 669–691, 2016.
[3] R. Maciejewski, Y. Jang, I. Woo, H. Jänicke, K. P. Gaither, and D. S. Ebert, “Abstracting attribute space for transfer function exploration and design,” IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 1, pp. 94–107, 2013.
[4] H.-C. Cheng, A. Cardone, S. Jain, E. Krokos, K. Narayan, S. Subramaniam, and A. Varshney, “Deep-learning-assisted volume visualization,” IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 2, pp. 1378–1391, 2018.
[5] L. Fang, J. Liu, J. Liu, and R. Mao, “Automatic segmentation and 3d reconstruction of spine based on fcn and marching cubes in ct volumes,” in 2018 10th International Conference on Modelling, Identification and Control (ICMIC),

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 4, April 2024 ||

| DOI:10.15680/IJIRSET.2024.1304169 |

pp. 1–5, IEEE, 2018.
[6] N. Shi and Y. Tao, "Cnns based viewpoint estimation for volume visualization," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 10, no. 3, pp. 1–22, 2019.
[7] M. Berger, J. Li, and J. A. Levine, "A generative model for volume rendering," IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 4, pp. 1636–1650, 2018.
[8] S. Martin, S. Bruton, D. Ganter, and M. Manzke, "Using a depth heuristic for light field volume rendering," in Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 1: GRAPP,, pp. 134–144, 2019.
[9] S. Weiss, M. Chu, N. Thuerey, and R. Westermann, "Volumetric isosurface rendering with deep learning-based super-resolution," Preprint on fIEEE Transactions on Visualization and Computer Graphics, 2019.
[10] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," in Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '88, (New York, NY, USA), pp. 65– 74, ACM, 1988.
[11] T. He, L. Hong, A. Kaufman, and H. Pfister, "Generation of transfer functions with stochastic search techniques," in Proceedings of Seventh Annual IEEE Visualization'96, pp. 227–234, IEEE, 1996.
[12] J. Li, L. Zhou, H. Yu, H. Liang, and L. Wang, "Classification for volume rendering of industrial ct based on moment of histogram," in 2007 2nd IEEE Conference on Industrial Electronics and Applications, pp. 913–918, 2007.
[13] S. Arens and G. Domik, "A Survey of Transfer Functions Suitable for Volume Rendering," in IEEE/ EG Symposium on Volume Graphics (R. Westermann and G. Kindlmann, eds.), The Eurographics Association, 2010.
[14] T. Fogal and J. Krüger, "Tuvok, an Architecture for Large Scale Volume Rendering," in Proceedings of the 15th International Workshop on Vision, Modeling, and Visualization, November 2010.
[15] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, G. H. Weber, H. Krishnan, et al., "Visit: An enduser tool for visualizing and analyzing very large data," High performance visualization–enabling extreme-scale scientific insight, pp. 357– 372, 2012.
[16] C. Lundstrom, P. Ljung, and A. Ynnerman, "Local histograms for design of transfer functions in direct volume rendering," IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 6, pp. 1570–1579, 2006.
[17] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional transfer functions for interactive volume rendering," IEEE Transactions on Visualization and Computer Graphics, vol. 8, no. 3, pp. 270–285, 2002.
[18] C. J. Morris and D. Ebert, "Direct Volume Rendering of Photographic Volumes Using Multi-Dimensional Color-Based Transfer Functions," in Eurographics / IEEE VGTC Symposium on Visualization (D. Ebert, P. Brunet, and I. Navazo, eds.), The Eurographics Association, 2002.
[19] G. Kindlmann and J. W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," in Volume Visualization, 1998. IEEE Symposium on, pp. 79–86, IEEE, 1998.
[20] J. Hladuvka, A. König, and E. Gröller, "Curvature-based transfer functions for direct volume rendering," in Proceedings of Spring Conference on Computer Graphics and its Applications, pp. 58–65, 2000.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details