

# Transfer Function-Guided Saliency-Aware Compression for Transmitting Volumetric Data

Ji Hwan Park, Ievgeniia Gutenko, and Arie E. Kaufman, *Fellow, IEEE*

**Abstract**—We introduce a transfer-function-guided 3D block-based saliency-aware compression scheme for volumetric data that is both content- and spatially-scalable. Salient 3D volumetric blocks are identified and weighted with the help of a transfer function which is used to render the data. We describe our method in the form of a framework for processing, progressive transmission, and visualization of volumetric data on a target device, such as a mobile device with limited computational resources. In particular, we address the transmission bottleneck incurred when transferring 3D volumetric data. Identified salient regions are progressively transmitted to the target device. The received data is rendered progressively in the respective order with a predefined or user-defined transfer function. Our method is developed with medical applications in mind, where preservation of all information is essential for clinical diagnosis. Because our method is integrated into a resolution scalable coding scheme with an integer wavelet transform of the image, it allows the rendering of each significant region at a different resolution up to fully lossless reconstruction. We perform a thorough qualitative and quantitative evaluation of the saliency detection method and the resulting saliency-aware compression schemes. Our results show reduced error in representation of the volumetric data with our method.

**Index Terms**—Compression, saliency, volume visualization, wavelets, discrete cosine transform.

## I. INTRODUCTION

WITH the size of volumetric data constantly growing, demand is increasing for efficient compression, storage, transmission, and visualization of volumetric data. There have been several approaches in the multimedia domain to progressively transmit a compressed 3D mesh with minimum packet loss [1] and to compress depth data [2] and 3D meshes [3, 4]. Unlike 3D mesh data, volumetric data consists of voxels, and often only a certain region or a sub-volume of a large dataset is of interest to the user. Extraction of this region can help in reducing compression, transmission, and visualization time, as well as storage space. For example, non-uniform encoding allocates fewer bits or completely disregards less

significant regions, thus providing a higher compression ratio. Another option is the transmission of the region/volume of interest (ROI/VOI) first at low quality or resolution and then progressively refining the result.

These content-aware techniques require prior analysis of the relevant visual importance or relevance of the data. In this paper, we consider the situation when the relevance of specific regions within the volume is unknown but can be derived from the underlying data values. We suggest reordering data based on a computational model of the transfer function (TF)-guided visual saliency. We construct a 3D block-based saliency map from weighted averaging coefficients such as the 3D discrete cosine transform (DCT) by analyzing low-level features: intensity, color differences, texture energy, orientation, and gradient. Our technique is best suited for applications where the TF is fixed or easily generated via a semi-automatic method [5] based on the material properties, such as data from computed tomography (CT) in medicine and industry. For example, medical scanners produce large volumes of data per patient at high resolution. Not only has the handling and storage of such data become an issue, but remote transmission to a medical expert for analysis is also a challenge. Since not all of the obtained information is essential for diagnosis, both compression and transmission techniques that handle VOI are particularly important.

Our method considers two main requirements of the target applications: the need for lossless reconstruction of the data and a transmission bottleneck associated with volumetric data. One of the main target domains for our framework is compression of 3D medical images. Such compression methods should be visually lossless [6]. The 3D block-based saliency is integrated into a fully reversible integer wavelet transform (IWT) and a quality scalable coding technique, thus allowing progressive transmission up to lossless reconstruction. Encoding and transmission start with the salient regions of the volume and proceed in several iterations. Selection of IWT coefficients is performed in the frequency domain, thus requiring only one transform on the entire volume. Additionally, selected voxels are grouped based on spatial proximity, which allows for reconstruction of more volumetric blocks with fewer coefficients.

In our work, we analyze the problem of user-adaptive informative visualization for progressive compression and transmission of volumetric data. Our contributions are:

- A 3D block-based saliency analysis for automatic computation of VOI.
- A framework design, integrating the 3D view-independent visual saliency representation of the data and grouping vox-

This work has been partially supported by National Science Foundation grants IIS0916235, CNS0959979, IIP1069147, CNS1302246, NRT1633299, CNS1650499 and by the Center of Excellence in Wireless and Information Technology (CEWIT) at Stony Brook University. The Foot, Bonsai, Engine, and the Visible Korean dataset are courtesy of Phillips Research, University of Stuttgart, General Electric, and HuminTec Inc., respectively. All other datasets are courtesy of Stony Brook University Hospital.

J. Park, I. Gutenko, and A. Kaufman are with the Computer Science Department, Stony Brook University, Stony Brook, NY 11794-2424 USA (e-mail: jihwpark@cs.stonybrook.edu; igutenko@cs.stonybrook.edu; ari@cs.stonybrook.edu).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. The material includes a document with a proof. Contact jihwpark@cs.stonybrook.edu for further questions about this work.

els into any scalable compression scheme to progressively compress, transmit, and visualize volumetric data.

- Quantitative measurement of the processing speed of the saliency detection and compression rates in comparison to methods without saliency.
- A qualitative and quantitative evaluation of the saliency detection method based on the ground truth volumetric saliency as defined by a gaze tracking study.
- A user study of subjective qualitative evaluation of the visual information with saliency-aware compression against the methods without saliency that shows reduced error in the representation of volumetric information.

## II. RELATED WORK

### A. Saliency Detection

Salient regions are the most conspicuous areas of the image to the human eye. As shown by Itti et al. [7], low-level features, such as color, intensity, and orientation can determine saliency of the image regions. Many applications make use of the salient information for region of interest (ROI) extraction, image retargeting and resizing for arbitrary size displays [8, 9], and visual enhancement for visualization [10, 11]. Computation of saliency, even for 2D images [7, 12], is often expensive and requires choosing parameters for optimal performance.

The use of the coefficients of a DCT has been studied and used for enhancement of perceptual quality of compression [13], definition of edge structures [14], object localization [15], and comparison of image similarity [16]. Fang et al. [17] use JPEG compressed images for saliency detection in adaptive image retargeting [18]. In this work, DC coefficients are converted from YCbCr to RGB to compute intensity, red/green and blue/yellow double opponency. Texture computation utilizes only the Y component frequencies. Final feature maps are computed by the Euclidean distance weighted by a Gaussian with a coherent normalization-based fusion method. The idea to determine saliency from the transform coefficients has been extended to the wavelet transform domain by Imamoglu et al. [19] through local and global saliency of the images. Saliency is also used in reducing attention-grabbing coding artifacts and adjusted in ROI and non-ROI in an H.264 video coding [20], which improves the visual quality of the coded video. The spatio-temporal saliency model serves for fast detection of salient regions in video sequences [21]. Statistical information in the wavelet compressed domain was used for volumetric data quality assessment [22]. Saliency-cognizant schemes in video compression can be used to conceal errors [23]. When high resolution blocks are not matched, low resolution blocks with a good saliency match are used as a replacement.

Unlike the work described above, we use the saliency map to guide the transmission of volumetric data and visualize the data. For this purpose, we perform construction of the full 3D saliency map for the weighted averaging which makes it computationally affordable for volumetric data and, given our implementation, potentially parallelizable. Our method utilizes multiple features by extending them from 2D to 3D and more importantly adding unique 3D features that allow us to

prioritize significant areas occupied by 3D salient objects, and not only their borders and edges.

### B. Volumetric Data Compression

Typically, image usage determines the choice of a compression method. When some degradation of image quality is permitted, lossy compression techniques achieve higher compression ratios. Lossless techniques are used when any degradation is undesirable and allow for much smaller compression ratios. In particular, for medical images, loss of information can affect the reliability of the diagnosis. Lossy-to-lossless compression techniques adapt to low network bandwidth. In such schemes, significant information is embedded at the beginning of the bitstream, thus allowing for the progressive transmission from initial lossy up to a lossless representation.

Discrete wavelet transform (DWT) or IWT based methods allow for *resolution* or *quality scalable* coding based on a decomposition of the wavelet and enhancement layer of the coding algorithm [24]. Among them are 3D set partitioning in hierarchical trees coding (SPIHT) [25], embedded block coding with optimized truncation (EBCOT) [26], embedded subband coding with optimized truncation (ESCOT) [27], and other modifications of those schemes. There are some methods to visualize data with limited memory via tensor approximation [28], volumetric compressed sensing [29], and dictionary learning with sparse coding technique [30]. These methods can significantly reduce the amount of data to represent input volumes, but they are lossy compression approaches.

The structural similarity (SSIM) index [31] and a foveated just-noticeable-distortion model [32] have been used to achieve the better perceptual quality of an encoded video in an H.264 video coding. These approaches improved the perceptual quality of the encoded videos for lossy compression and even can achieve perceptually lossless compression [32].

VOI compression for medical images has been explored for both 2D images and volumetric data. General ROI-based scaling (GSB) and MAXSHIFT methods are part of JPEG2000. These are the basic variations of ROI compression methods that consider regions of arbitrary form with full or partial loss of the background information. Medical image coding also utilized 3D ROI or VOI with a scalable 3D SPIHT algorithm [33]. This method uses a mask to determine transform coefficients that also undergo compression. Additionally, the resulting VOI can become prioritized during visualization with either iso-surface or ray casting.

In the client-server scenario, the data can be compressed once, and upon the client's request, the bit stream is reordered to allow rendering of the VOI with fading quality of the background [34, 35]. Unlike [34, 35], in our work, we automatically determine and transmit the VOI as the most salient region in the volume without user specification and eventually transmit all the data. Additionally, our approach can utilize any wavelet-based scalable compression scheme, such as SPIHT, EBCOT, and ESCOT.

### C. Visualization of the Compressed Data

Volume rendering directly from the DCT compressed domain has been proposed by Chieh et al. [36] and Yeo and

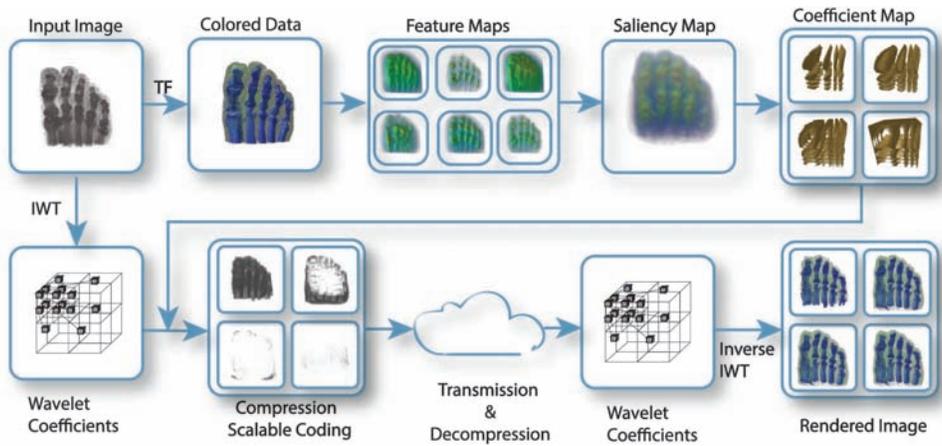


Fig. 1: Overview of our framework. After selecting a pre-defined transfer function (TF) or generating a TF semi-automatically, the saliency map is computed and the integer wavelet transform (IWT) coefficients are split according to the coefficient maps to represent subvolumes that are compressed for transmission and decompressed and progressively rendered on the receiving device.

Bede [37]. Fout and Ma [38] have proposed a block-based transform coding scheme for fast decompression and progressively rendering a volume by approximation. Ljung et al. [39] have used a TF to guide the selection of the level of detail scheme during data decompression. This technique reduces the amount of data to be decompressed and rendered but does not assist in informative rendering, meaning it does not present overview information about the data. A compression-based multiresolution rendering scheme has been suggested for rendering data on a regular grid using wavelet representation [40]. Wavelet decompression of volume data for visualization in a block-based scheme has been proposed by Nguyen et al. [41] for rapid visualization. Some methods address the complexity that is associated with compression and decompression and preserve features of the data while distorting the grid [42]. For a more detailed review of state-of-the-art approaches, we refer the reader to the survey papers [43, 44].

Volumetric rendering on mobile devices is generally subdivided into two approaches: thin client (remote) rendering and rendering directly on the device. In our work, rendering occurs directly on the device and we can use any efficient volume rendering technique for mobile devices such as [45]. The most similar work is by Campoalegre et al. [46] who use a TF aware lossy compression scheme and user-defined ROI for multiresolution visualization on mobile devices. However, their method is for lossy compression and the TF cannot be changed on the mobile devices. Here we also refer the reader to additional techniques surveyed by Noguera et al. [47].

Unlike the papers mentioned above, which lose some information, our method first compresses and transmits the most salient regions and then progressively adds less salient regions. Therefore, we can losslessly reconstruct salient regions with a relatively small size coefficient map. Moreover, users can modify the TF on the receiving device for already received data at any time.

### III. OVERVIEW

We introduce saliency into a scalable coding scheme [25, 26] for progressive transmission and rendering. This method

serves prioritization of salient volumetric regions as well as their quality refinement up to fully lossless reconstruction. First and foremost, our framework (Fig. 1) is based on the semi-automatic design of a TF for the volumetric data as it is intended to be viewed by the user. The following two steps of our framework are executed concurrently. In one step, weighted averaging is performed on the volume colored by the TF. Weighted averaging (3D DCT) blocks are analyzed to construct the 3D block-based *saliency map*. At the same time, the fully reversible IWT is applied to the volumetric scalar data. The *coefficient map* defines a correspondence between the IWT coefficient voxels and the DCT saliency map. Transmission of data is prioritized based on the computed saliency map, and, similarly, progressively decompressed and rendered on the receiving side.

Our approach focuses on domain specific scenarios with known data types. Such applications have a focused scope and are often used by non-computer experts, and thus require a pre-defined or semi-automatically generated set of TFs. The user is provided with an interface to change between the TF presets. In this paper, we use an approach which generates an optimal TF by only evaluating small images generated with similar potential TFs [5]. Here, we show two examples:

1. *Medical CT and CTA application.* Medical software used in clinical practice utilizes a set of pre-set TFs and facilitates a simplified interface for their adjustment. Analysis of CTA data generally follows a protocol, for example, CTA of the thoracic aorta [48] includes pre-set TFs in the software, such as that of TeraRecon (San Mateo, CA). In the user interface of the software, the default value of the TF is specified in two ways. First, the default TF is grayscale, set as a linear ramp from black to white (window width 324, window center 186) at full opacity. Second, the TF is set in a red to yellow range on a triangle ramp (window width 200, window center 106) at opacity 0.4. Properties of CT allow for such fixed differentiation between tissues types (Fig. 2 (a)) [49]. Fig. 2 (b) shows a use case, where we select identical TFs for two datasets, similar to those used in the medical software, and

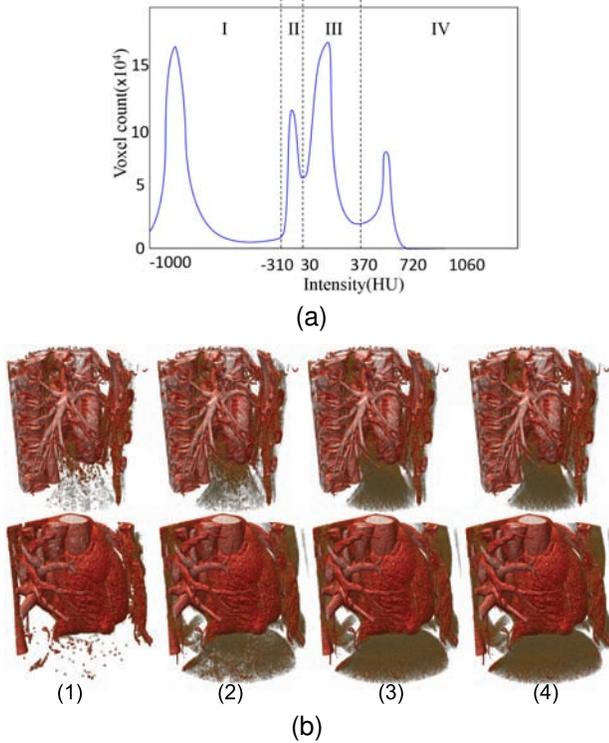


Fig. 2: (a) A histogram of a heart dataset. The histogram is separated into components based on the material properties: I - Pulmonary parenchyma; II - Pericardium, diaphragm; III - Myocardium, hepatica parenchyma, coronary arteries without contrast; IV - Ventricles, coronary arteries with contrast, vein, aorta, bone, pulmonary vessels. (b) A use case for two  $512^3$  heart datasets (top and bottom) with the same fixed TF used for both datasets. The results of progressive transmission and rendering at the first through fourth iteration (1-4).

produce progressive volume rendering, on the target device.

2. *Industrial CT applications.* In industrial manufacturing of engine blocks and cylinder heads, manufactured parts undergo a CT scan for quality assurance. One of the ways to approach differentiation of materials on the CT scan considers isovalues in the histogram of data values. For this engine dataset, there are three known materials: two alloys at isovalues 120 and 165 (regions II & III on Fig. 3 (a)) and air [50]. Known values of alloys can be used for TF initialization, as shown in Fig. 3 (b).

It is important to note that while only one TF is used for the data compression, it can be freely modified on the receiving device for the already received data at any time. Fig. 1 illustrates the entire process of our framework, and each stage of the pipeline is detailed below.

#### IV. MAPS FOR COMPRESSION

Saliency and coefficients support the compression process. The *saliency map* is computed in the transform domain of weighted averaging (DCT) by extracting features of each block, weighting them, and combining them into a single saliency map. The *coefficient map* is computed from the relation between each saliency region and the position of a wavelet coefficient in the tree structure of the wavelet transform.

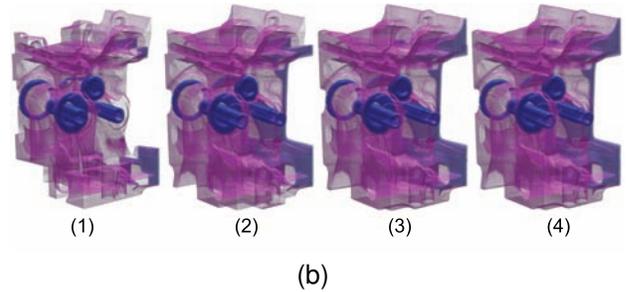
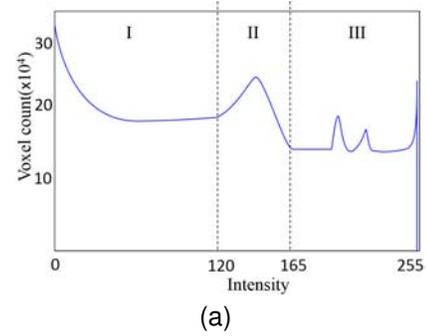


Fig. 3: (a) A histogram of a Engine dataset. The histogram is separated into components based on the material properties: I - air, II and III - two alloys, where air is considered as noise and then assigned low opacity in the TF. (b) A use case for a  $256 \times 256 \times 128$  Engine dataset with a fixed TF. The results of progressive transmission and rendering at the first through fourth iteration (1-4).

The combination of these maps allows for prioritization and transmission of a reduced amount of initial information.

##### A. Weighted Averaging Based Saliency Map

Our method is based on the saliency detection in the compressed domain. We have extended an approach introduced by Fang et al. [17] (for 2D still images) to create a 3D saliency map considering volumetric features. The benefit of using DCT is twofold: first, the properties of the DCT allow us to extract the required features; and second, the block-based transform allows us to shrink the saliency map by orders of magnitude compared to the size of the original volume.

We consider six types of low-level features derived from the DCT and the input volume: *intensity*, *two color components*, *texture*, *orientation*, and *gradient*. While the significance of these features has been shown for 2D images [7, 17], we highlight the differences and feature importance for volumetric data. To construct the saliency map we use the DC coefficients of the DCT that represent an average of the block, and the AC coefficients that show frequency information.

*Intensity and color features.* Most commonly, the volumetric data colored with a TF is rendered in RGB color space, which we convert to YCbCr. The intensity and color features are obtained from their average (DC) in each block. By definition, the DC coefficient establishes local distinctness in the neighborhood (saliency as distinctness [51]). Thus, a distance-weighted difference between DCT blocks describes saliency of the entire volume. As a result, we obtain the first

two types of features: *intensity I* and the *color features Cb* and *Cr*:

$$I = Y_{DC}, Cb = Cb_{DC}, Cr = Cr_{DC}, \quad (1)$$

where  $Y_{DC}$  is the DC coefficient of the  $Y$  component,  $Cb_{DC}$  is the DC coefficient of the  $Cb$  component, and  $Cr_{DC}$  is the DC coefficient of the  $Cr$  component. We consider two color channels to be one feature in the map.

**Texture features.** Well-defined boundaries of objects that attract users' attention [52] can be captured with a texture's high-frequency information. DCT coefficients can be used to obtain such information in both color and gray-scale 2D images [18, 53, 54]. The AC coefficients show a texture feature in a particular direction. We consider only the  $Y$  component of the blocks because the others contain little high-frequency information [55]. In fact, we fully exclude low frequency components (similar to Tong and Venetsanopoulos's approach [13]) as they can be associated with noise [52]. Uniquely to volumetric data rendering, we can assume that the TF assigns low opacity (value less than 10) to noise and exclude the corresponding blocks. Here the range of opacity is between 0 to 255, and salient parts are assigned an opacity value of at least 20.

Finally, the energy of texture block  $T$  is the difference between the sums of all AC coefficients and the low frequency AC coefficients:

$$T = \sum_{i \in D} AC_i - \sum_{i \in L} AC_i, \quad (2)$$

where  $D$  is all the AC coefficients, and  $L$  is all the AC coefficients with low frequency components.

**Orientation and gradient features.** Computation of edges in 3D volumetric data is essentially similar to 2D images with an added complexity of a third dimension. In this work we simplify previously proposed approaches to accommodate volumetric and save computation time.

The orientation of an edge can be computed in gray luminance space [56]. Considering only the  $Y$  component of the DCT blocks we further save computation time when extending this approach to 3D volumetric data.

An *orientation feature* component  $Or$  of each  $8^3$  block along the  $x$ ,  $y$ , and  $z$  axes could be estimated from eight  $4^3$  sub-blocks (by extending a 2D method [14]). Ignoring non-significant components can further simplify each orientation, thus making it suitable for the volumetric data.

As a result, each orientation is approximated as a single AC coefficient:

$$\begin{aligned} Or_x &= 2|\sqrt{2}X_{dct}(1, 0, 0)/32| \\ Or_y &= 2|\sqrt{2}X_{dct}(0, 1, 0)/32| \\ Or_z &= 2|\sqrt{2}X_{dct}(0, 0, 1)/32| \end{aligned} \quad (3)$$

where  $X_{dct}(i, j, k)$  is a DCT coefficient from the  $Y$  component, and  $Or_m$  with  $m \in \{x, y, z\}$  indicates the orientation along the  $m$  axis. For details see the supplementary material.

In 3D volumetric data, gradient differentiates homogenous regions and regions of change, and because of that is frequently used for TF assignment. Kniss et al. [57] showed

the use of a TF for separation of materials and boundaries using a simple 2D TF. Additionally, gradient is used to extract iso-surfaces in volumetric data, thus differentiating structures. As our work builds upon the TF definition, we incorporate a *gradient feature* into the saliency map.

We already compute the orientation of each axis, so we can easily obtain the gradient of each block  $G$  as follows:

$$G = \sqrt{Or_x^2 + Or_y^2 + Or_z^2} \quad (4)$$

where  $Or_m$  with  $m \in \{x, y, z\}$  indicates orientation along the  $m$  axis.

**Opacity.** Opacity is an important and unique feature to visualization of volumetric data. During transfer function design, noise voxels are generally assigned to be more transparent, thus leaving important voxels more opaque. In this case, only visible parts in the volumetric rendering can be salient to the viewer.

Since opacity has a strong influence on all the features, they are weighted by opacity. The opacity  $Oc$  of each block is computed by averaging the opacity values of the individual voxels in that block:

$$F = Oc \cdot F' \quad F, F' \in \{I, Cb, Cr, T, Or_x, Or_y, Or_z, G\} \quad (5)$$

where  $Oc$  is opacity, and  $F'$  and  $F$  are features before and after being weighted by the opacity, respectively.

**Final saliency map.** For each of the five types of features (or six distinct features) extracted from the DCT blocks, we compute a *weighted feature map*. The weights are based on the  $L^2$  norm of the standard deviation and the distance between the values weighted by a Gaussian distribution [17] in the DCT feature space. It is important to note that computation of the distance in the volumetric space is more expensive and may not represent the difference between the features.

In volumetric data, groups of materials/tissues have similar feature properties, for example color and opacity. This enables grouping of DCT blocks based on their quantized color and opacity values. We compute the average of each feature, the standard deviation, and the position of a group. In our experiment, it significantly reduced the computational time for a saliency map, which is shown in Section 6.

In a weighted feature map, the difference  $D_{ij}^t$  between a feature  $t$ ,  $t \in I, Cb, Cr, T, G$  of two clusters can be defined as:

$$D_{ij}^t = |C_i^t - C_j^t| \quad (6)$$

where  $C_i^t$  and  $C_j^t$  represent the value of the  $i$ th and  $j$ th cluster in a feature  $t$ , respectively. For orientation features that consist of directions, the distance is computed as:

$$D_{ij}^{Or} = \sum_{t' \in Or_x, Or_y, Or_z} |C_i^{t'} - C_j^{t'}| \quad (7)$$

where  $C_i^{t'}$  and  $C_j^{t'}$  represent the value of the  $i$ th and  $j$ th cluster oriented along the  $x$ ,  $y$ , or  $z$  axis, respectively. The final feature value  $F_i^t$  is defined as:

$$F_i^t = Oc_i \cdot \sum_{j \neq i} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{d_{ij}^2}{2\sigma^2}} D_{ij}^t, \quad i, j = 1 \dots (M/8)^3 \quad (8)$$

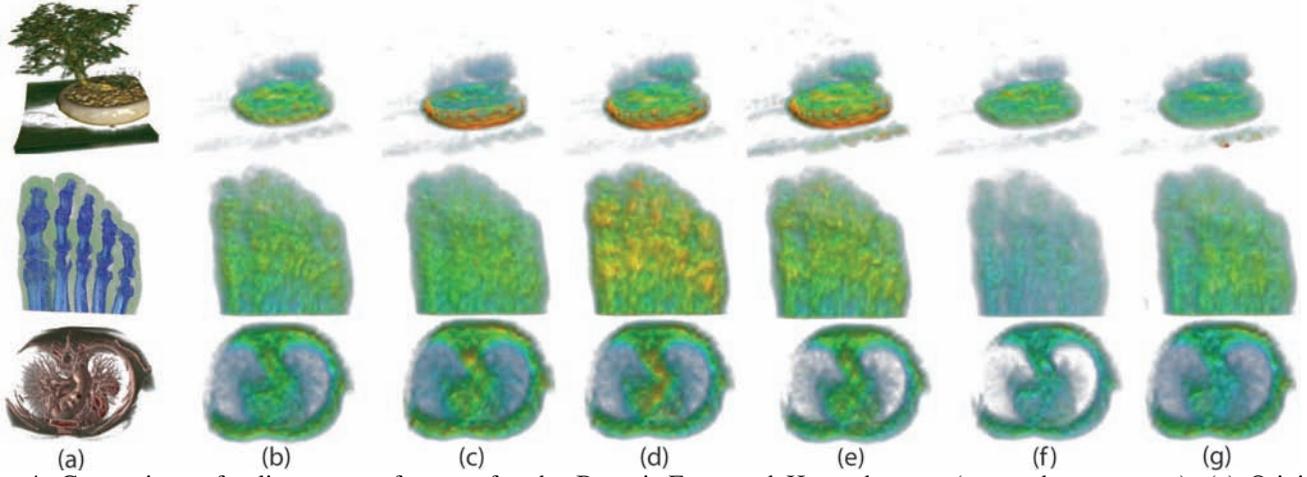


Fig. 4: Comparison of saliency map features for the Bonsai, Foot, and Heart datasets (top to bottom rows). (a) Original dataset, with applied TF and saliency maps for (b) gradient, (c) intensity, (d)  $Cr$  color component, (e)  $Cb$  color component, (f) orientation, and (g) texture features.

where  $\sigma$  is the standard deviation of the Gaussian,  $d_{ij}$  is the sum of the  $L^2$  norm of the standard deviation and the Euclidean distance between two clusters in the extracted feature space, and  $D_{ij}^t$  is the difference of a feature  $t$ ,  $t \in I, Cb, Cr, T, Or, G$  between two clusters defined in Eqs. 6 and 7. Here,  $\sigma$  represents the relative locality of the feature maps, depending on the size of the input volume, where the decrease of  $\sigma$  emphasizes smaller neighboring regions while the increase of  $\sigma$  considers a more global structure of the feature values for saliency. We set  $\sigma$  to 5 to balance local and global structures of the maps in our examples.

Finally, all of the above feature maps are linearly combined into a *saliency map*  $S$ :

$$S = \frac{1}{n} \sum_{i \in t} F^i, t \in \{I, Cb, Cr, T, G, Or\}, \quad n = 6 \quad (9)$$

To illustrate the contribution of each feature to the saliency map, we show each component separately in Fig. 4. The TF color implies the saliency value, with the highest being red, then green, and the lowest blue.

The saliency map values are sorted based on the ranges for each iteration. In the next step, we create coefficient maps corresponding to the range of each transmission.

### B. Coefficient Map

The saliency map guides the selection of the transform coefficients for progressive transmission. As a single voxel in the input volume decomposes into multiple coefficients, such selection is not straightforward and is enabled with the help of a coefficient map.

The vector of the wavelet transform coefficients consists of detail and approximation coefficients. At each level of the transform in the 1D case, a detail coefficient  $i$  is partially affected by  $2^i$  elements in the input data.  $(\log_2 M + 1)^3$  coefficients are required to reconstruct one voxel in the  $M^3$  volume data. Essentially, the same coefficients are needed to reconstruct two adjacent voxels at positions  $v_{2i}$  and  $v_{2i+1}$ .

**Algorithm 1** : Coefficient selection method for each block corresponding to a saliency value

---

```

1:  $lvl =$  level of wavelet transform
2: for all possible approximation coefficients at last level ( $lvl$ ) do
3:   if a coefficient affects a block of saliency region then
4:     Save the index of a current coefficient
5:   end if
6: end for
7: for each level from last( $lvl$ ) to 1 do
8:   for all possible detailed coefficients at current level do
9:     if a coefficient affects a block of saliency region then
10:      Save the index of a current coefficient
11:     end if
12:   end for
13: end for

```

---

We exploit this redundancy when selecting the transform coefficients that are representative of the salient regions (as shown in Algorithm 1).

With the increase in the number of wavelet transform levels, the redundancy between neighboring voxels of different saliency ranges also increases. In a single-scale coefficient map, volumetric blocks are selected solely based on their saliency values. The multi-scale coefficient map reduces this redundancy by grouping neighboring voxels and blocks. The multi-scale map has a maximum resolution of  $2^{max_{wav}}$ , where  $max_{wav}$  is the maximum wavelet transform level.

The multi-scale coefficient map is constructed as follows:

- For each voxel  $S_{i,j,k}$  in a saliency map within a current saliency range, we first check whether the voxel was processed in the previous saliency range. If yes, we skip the voxel. Otherwise, we check the neighboring voxels,  $S_{i+i',j+j',k+k'}$ ,  $i', j', k' \in \{0, 1\}$ . If more than half of the neighboring voxels are within the range, we group them all together.
- Repeat the scan  $(\log_2 \omega - 3)$  times, doubling the range of  $i', j', k'$  each time, where  $\omega$  is the defined resolution.

Fig. 5 compares the structure of single- and multi-scale maps. The two maps are similar even though there are more voxels in its quantized version of multi-scale maps.

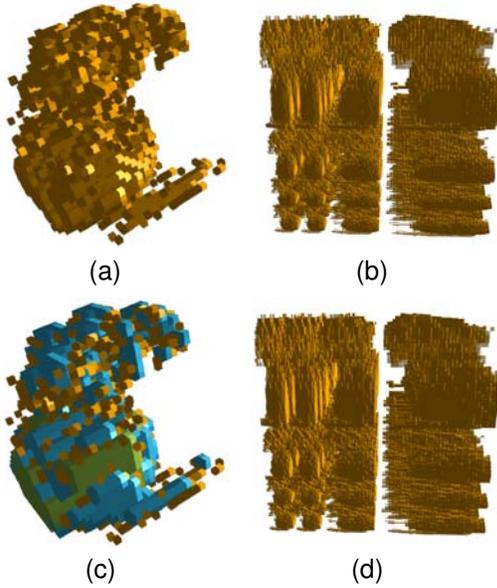


Fig. 5: Correlation between quantized saliency maps (left column) and coefficient maps (right column) of the Bonsai tree dataset. The first row shows a single-scale (a) quantized saliency map and (b) coefficient map. The second row shows a multi-scale (c) quantized saliency map and (d) coefficient map. In the multi-scale quantized saliency map,  $8^3$ ,  $16^3$ , and  $32^3$  resolutions are represented by each color: yellow, blue, and green, respectively.

In our method, we use the 3D Haar wavelet transform with a lifting scheme as it has been shown to be the most optimal. Since one of our target applications is medical imaging, we compare its performance with Cohen-Daubechies-Feauveau (CDF) 9/7 and 5/3 wavelets, used in that domain. Additionally, this approach can be further extended to include Daubechies wavelets D4 and D6. With our method, there is no additional overhead information about the position of the region that has to be transmitted, as in VOI methods, because only selected coefficients are encoded. Thus, our coefficient maps can utilize any wavelet-based scalable compression scheme for progressive transmission. In our study, we have used quality scalable codecs: 3D SPHIT [25], EBCOT [26], and 3D ESCOT [27].

## V. PROGRESSIVE TRANSMISSION

Progressive transmission is an inherent part and goal of scalable coding algorithms, such as SPIHT, EBCOT, and ESCOT. Based on similarities between subbands in a wavelet decomposition of an image, SPIHT creates a spatial orientation tree (SOT) and then performs initialization, a sorting pass, a refinement pass, and quantization-setup update. EBCOT is a block-based compression algorithm. In EBCOT, each subband from the wavelet transform is divided into small blocks, namely code-blocks, which are coded independently. EBCOT has three passes: significant propagation pass, magnitude refinement pass, and clean up pass. ESCOT is an extension of EBCOT, where the main difference is the way to code each subband. These methods first encode the information that

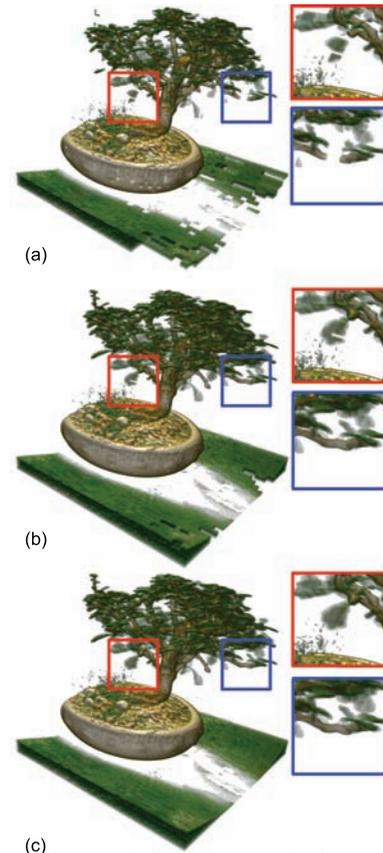


Fig. 6: Comparison of the progressively rendered Bonsai volume after the (a) first, (b) second, and (c) fourth (last) transmission. After the last transmission a lossless reconstruction is obtained.

yields the highest distortion reduction by placing coefficients with larger magnitude at the beginning of the bitstream. If the receiver truncates the bitstream at any point, the highest possible mean-squared error (MSE) up to that point will be achieved.

In addition to prioritization of the bitstream by magnitude as designed by the coding algorithm, we organize the data stream based on the saliency intervals from the computed maps. We split IWT coefficients of the original volume into several different parts based on the saliency intervals of the computed maps and encode them separately. Hence, with every step of the transmission, we have the same scalability benefit of the coding scheme. The most salient blocks will be fully transmitted first, followed by the sets of blocks from the remaining intervals.

The saliency intervals can be either set as constant (i.e. constant determined by the data type, 8-bit, 16-bit, 32-bit), or determined dynamically. In the latter case, the range of the salient region can be computed dynamically from prominent peaks in the histogram plot of intensities of the saliency map. Local minimum points on the plot can serve as transition points between the saliency levels. Since the number of such “separation” points can be pretty high, one can establish a slope threshold. If the change between two points on the histogram is less than a given threshold, then the transition point can be omitted.

The most significant difference is noticed only during the

TABLE I: Computation time (sec) of DCT, averaging  $8 \times 8 \times 8$  voxels (Avg.), and saliency maps (Sal.). We also compare decoding time (sec) for multi-iteration (Iter. 1-4) transmission based on our saliency-guided scheme, as well as the total combined sum of all, and the first three iterations with lossless compression for the full dataset using Haar wavelet and SPIHT (top: performed on LG Nexus 4; bottom: performed on Microsoft Surface Pro).

Dataset	Size	DCT	Avg.	Sal.	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Combined (Iter.1-4)	Combined (Iter.1-3)	Full dataset
Engine	256×256×128	0.16	0.10	0.45	0.98	0.52	0.07	1.82	3.38	<b>1.56</b>	2.55
					0.59	0.29	0.06	1.12	2.06	<b>0.94</b>	1.74
Bonsai	256×256×256	0.41	0.22	1.05	2.02	0.52	0.34	2.42	<b>5.30</b>	<b>2.88</b>	5.65
					1.38	0.35	0.21	1.40	3.33	<b>1.93</b>	2.9
Foot	256×256×256	0.33	0.22	0.94	3.09	0.75	0.11	2.52	6.47	<b>3.95</b>	5.33
					1.80	0.48	0.09	1.48	3.85	<b>2.38</b>	3.44
Heart	256×256×256	0.36	0.22	1.07	3.56	1.92	0.67	2.50	8.65	6.15	5.97
					1.76	1.17	0.36	1.10	4.39	<b>3.42</b>	3.76
Korean	256×256×256	0.35	0.19	1.09	3.63	0.62	0.08	2.30	6.62	<b>4.33</b>	4.99
					1.90	0.36	0.04	1.10	3.40	<b>2.31</b>	3.08
Vismale	256×256×256	0.32	0.22	0.94	1.64	0.35	0.07	1.65	3.70	<b>2.06</b>	2.94
					1.01	0.24	0.05	0.95	2.24	<b>1.29</b>	2.01
Heart1	512×512×512	2.88	1.56	7.09	16.14	4.84	1.56	147.01	<b>169.54</b>	<b>22.53</b>	278.86
					9.32	3.01	1.03	79.10	<b>92.46</b>	<b>13.36</b>	173.64
Heart2	512×512×512	2.92	1.52	6.94	17.99	12.66	6.84	65.90	<b>103.39</b>	<b>37.49</b>	265.70
					10.38	7.86	4.54	35.46	<b>58.24</b>	<b>22.78</b>	165.45

first step of the transmission, as later the entire volume is fully reconstructed. Fig. 6 shows progressive refinement after consecutive transmissions of the Bonsai tree.

## VI. RESULTS

In this section, we discuss and compare the construction of the saliency maps, their quality based on both objective and subjective perceptual metrics, and the performance of the coding algorithms. Compression is done on an Intel Xeon E5-2620 2.0GHz workstation with 8 GB RAM and GeForce GTX 670 graphics with 2 GB memory. Decompression is performed on two mobile devices: LG Nexus 4 (Qualcomm Snapdragon S4 Pro 1.5GHz and 2 GB RAM) and Microsoft Surface Pro (Intel i5-3317U 1700.0 MHz, Intel HD4000). The only requirement of the method is support for decoding algorithms such as SPIHT, EBOCT, or ESCOT.

### A. Objective Evaluation

For objective quality evaluation, we first assess the performance of the saliency maps. We then evaluate the visual quality of the rendered objects based a visual quality metric. Lastly, we compare the compression rates between our method and SPIHT, EBCOT, and ESCOT, and decoding times between our method and SPIHT.

#### *Computation time and performance of saliency maps.*

In order to justify the use of saliency maps, the overall computation overhead of these maps should be insignificant. Table I shows the computation time for the saliency maps. Although our implementation for creating saliency maps is not fully optimized, the average computation times of the saliency map for a  $256^3$  volume and  $512^3$  volume are 1.02 and 7.01 seconds, respectively.

Three out of six features (intensity and two colors) can be obtained by averaging voxels. Hence, we compare the computation time for the DCT-based saliency map and the simple voxel averaging based method (Table I). Essentially,

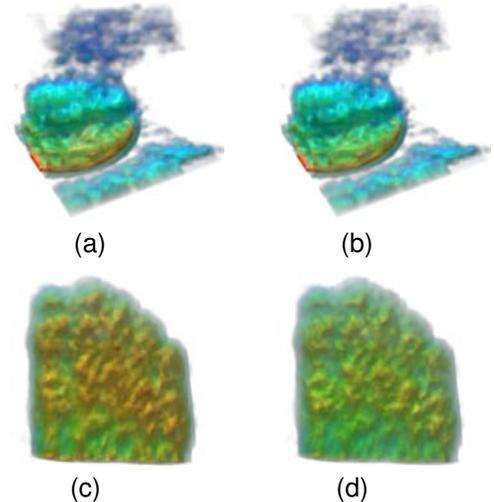


Fig. 7: Comparison of DCT-based (left column) and averaging-based (right column) saliency maps. (a) and (b) Bonsai dataset, and (c) and (d) Foot dataset. Generally, DCT based saliency maps capture more details than averaging based saliency maps.

averaging voxels results in some reduction of time, however, it is only suitable for simple objects (Fig. 7).

**Compression rates, decoding time, and decoding CPU samples.** Progressive transmission and compression of the volumetric data according to the constructed maps enables the initial examination of the most salient regions. Refinement of the data in the order of its saliency allows the viewer to explore a large amount of information much faster. Our experiments have shown that separate salient regions result in a good compression ratio, and carry a sufficient amount of the information for data exploration, as determined by the user who selected or designed the TF.

Table II shows the compression rates for each saliency region. For comparison, we set the number of saliency regions/iterations to four and define ranges of the regions to be

TABLE II: Compression rate (bits per voxel) comparison. We compare the compression rate for our multi-iteration (Iter.) transmission based on our saliency-guided scheme, as well as the total combined sum of all, and the first three iterations with the lossless compression ratio for the full dataset. Three coding algorithms (SPIHT, EBCOT, ESCOT) have been evaluated with Haar wavelet transform. All the datasets are in 8-bit precision.

Dataset	Size	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Combined (Iter. 1-4)	Combined (Iter. 1-3)	Full dataset
Engine (SPIHT)	256×256×128	0.77	0.39	0.08	1.45	2.70	<b>1.24</b>	2.22
Engine (EBCOT)	256×256×128	0.60	0.26	0.03	1.24	2.35	<b>1.12</b>	1.87
Engine (ESCOT)	256×256×128	0.59	0.26	0.03	1.22	2.09	<b>0.88</b>	1.83
Bonsai (SPIHT)	256×256×256	0.87	0.24	0.41	0.78	2.30	<b>1.52</b>	1.85
Bonsai (EBCOT)	256×256×256	0.67	0.12	0.23	0.54	1.57	<b>1.02</b>	1.44
Bonsai (ESCOT)	256×256×256	0.65	0.13	0.24	0.67	1.68	<b>0.01</b>	1.42
Foot (SPIHT)	256×256×256	1.19	0.32	0.06	0.97	2.54	<b>1.57</b>	2.23
Foot (EBCOT)	256×256×256	1.00	0.21	0.02	0.79	2.01	<b>1.22</b>	1.92
Foot (ESCOT)	256×256×256	0.94	0.20	0.02	0.84	2.00	<b>1.16</b>	1.91
Heart (SPIHT)	256×256×256	1.15	0.78	0.25	0.73	2.90	<b>2.18</b>	2.40
Heart (EBCOT)	256×256×256	0.97	0.62	0.16	0.60	2.34	<b>1.74</b>	2.11
Heart (ESCOT)	256×256×256	0.83	0.54	0.15	0.80	2.32	<b>1.51</b>	2.09
VisMale (SPIHT)	256×256×256	0.68	0.15	0.02	0.46	1.31	<b>0.85</b>	1.17
VisMale (EBCOT)	256×256×256	0.48	0.10	0.01	0.45	1.03	<b>0.58</b>	0.97
VisMale (ESCOT)	256×256×256	0.48	0.10	0.01	0.49	1.09	<b>0.60</b>	0.95
Heart1 (SPIHT)	512×512×512	0.71	0.24	0.08	1.36	2.40	<b>1.03</b>	2.04
Heart1 (EBCOT)	512×512×512	0.37	0.11	0.03	0.97	1.47	<b>0.51</b>	1.34
Heart1 (ESCOT)	512×512×512	0.48	0.13	0.03	1.07	1.71	<b>0.64</b>	1.30
Heart2 (SPIHT)	512×512×512	0.64	0.46	0.25	0.96	2.30	<b>1.35</b>	1.93
Heart2 (EBCOT)	512×512×512	0.35	0.24	0.12	0.70	1.41	<b>0.71</b>	1.26
Heart2 (ESCOT)	512×512×512	0.49	0.30	0.13	0.80	1.73	<b>0.93</b>	1.22
Heart (SPIHT)	512×512×1024	0.78	0.14	0.03	1.43	2.27	<b>0.96</b>	2.03
Heart (EBCOT)	512×512×1024	0.48	0.14	0.04	1.00	1.65	<b>0.65</b>	1.52
Heart (ESCOT)	512×512×1024	0.60	0.13	0.04	1.01	1.78	<b>0.77</b>	1.34

TABLE III: CPU sample comparison for decoding by using the sampling profiling method on a mobile device (Microsoft Surface Pro). We compare the decoding CPU samples for our multi-iteration (Iter.) transmission based on our saliency-guided scheme, as well as the total combined sum of all, and the first three iterations with the lossless compression ratio for the full dataset, using the SPIHT coding scheme. All the datasets are in 8-bit precision.

Dataset	Size	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Combined (Iter. 1-4)	Combined (Iter. 1-3)	Full dataset
Abdomen CT	256×256×256	396	11	11	370	788	<b>418</b>	764
Bonsai	256×256×256	278	20	22	304	624	<b>320</b>	549
Heart1	512×512×512	9,930	2,681	860	50,971	<b>64,442</b>	<b>13,471</b>	109,638

constant across datasets. At the first iteration, the amount of data to be transmitted is often less than half compared to the lossless compression of the entire dataset. The time to decode the data (Table I) at the first iteration takes less than half or one tenth of the lossless decoding time for  $256^3$  datasets and  $512^3$  datasets, respectively. However, the viewer can already see most of the data without any additional noise (Figs 8(b) and 2(b)(1)) after the first iteration. The rendering is progressively refined further as the remaining data is received. Furthermore, we moved voxels with non-zero opacity in the last iteration to the third iteration because most of the voxels in the last iteration of our progressive transmission have zero opacity. Thus, all of the voxels in the last iteration have zero opacity. As a result, on the target device, 3D rendering with a TF used for creating the saliency map will result in visually lossless rendering, while the remaining data is transmitted and received in the background. Additionally, we measured CPU samples for decoding by using the sampling profiling method because we can predict the performance of our method by using it, and it has a significant effect on power consumption [58]. As shown in Table III, for the first three iterations, the amount

of CPU samples for decoding is less than half compared to the lossless compression of the entire dataset while we can achieve visually lossless rendering of the input dataset. As data accumulates on the receiving device, regions are merged, and lossless reconstruction of the entire volume is obtained at effectively no cost due to the lack of redundancy between the regions.

**Compression with and without saliency maps.** To evaluate visual quality of the rendered objects objectively, we used the percentile based multi-scale structural similarity index (PMS-SSIM) metric [59] between input volume colored by the TF and transformed data (Table IV). We have constrained the bit budget of the compression without saliency maps to the number of bits required for the transmission of the corresponding saliency interval and computed the PMS-SSIM metric on the decompressed volumes. In the first iteration, compression without saliency maps has given quantitatively better results in most cases, except for the abdomen CT and brain MRI datasets. However, in the second iteration, the visual quality of more than half of the datasets with our method is similar to or better than one without saliency maps.

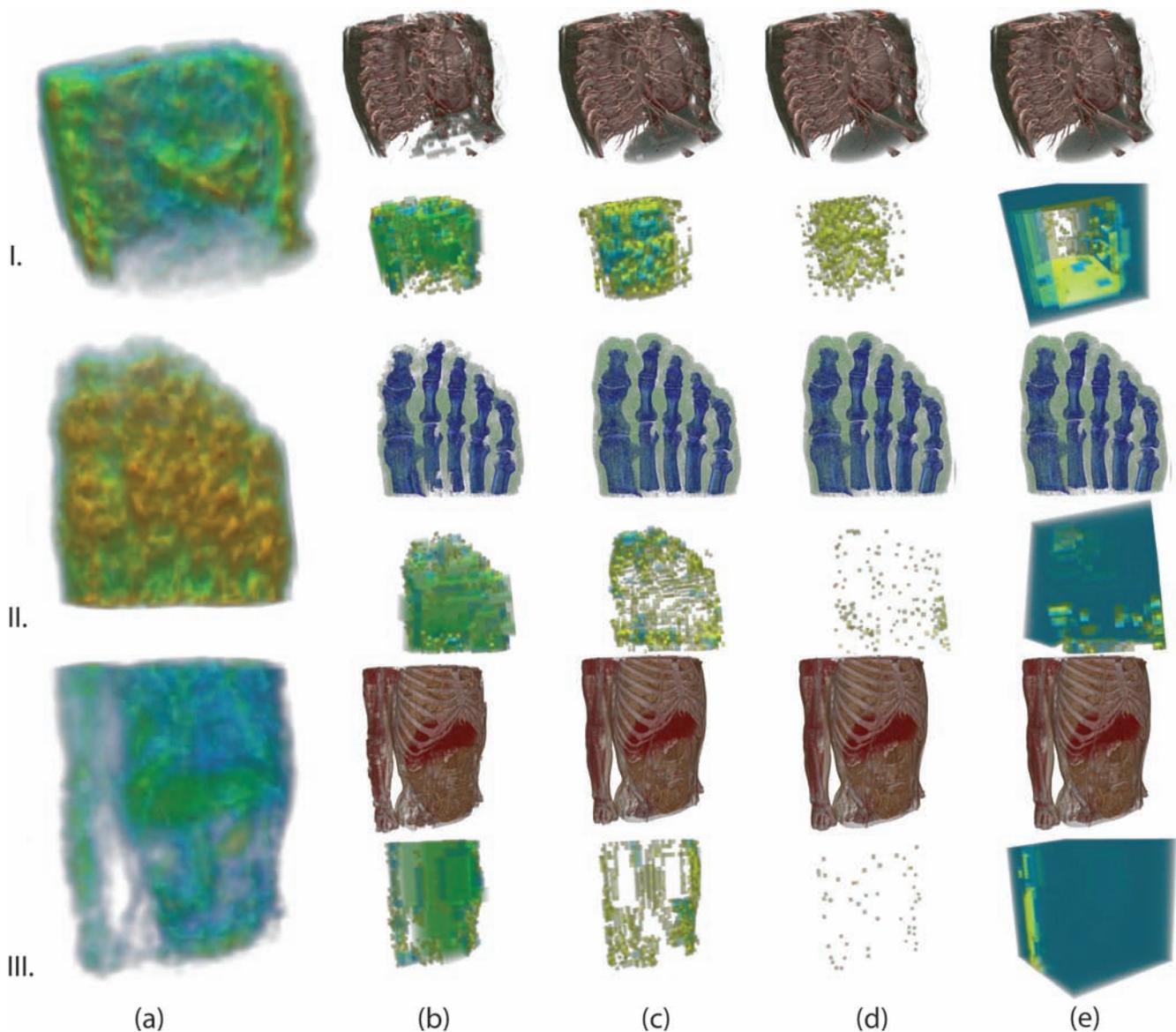


Fig. 8: Progressive rendering of the sample  $256^3$  datasets: I. Heart, II. Foot, III. Visible Korean. (a) Saliency maps and the results of transmission (b) at the first iteration, (c) second iteration, (d) third iteration, and (e) fourth iteration. Top: rendered image; bottom: corresponding blocks in quantized saliency maps.

TABLE IV: Comparison of the percentile based multi-scale structural similarity index (PMS-SSIM) [59] for the entire volume (Vol) in each iteration (Iter.) and the signal-to-noise ratio (SNR) and the peak signal-to-noise ratio (PSNR) for the blocks of a given saliency interval (Blk) in the first iteration, using the SPIHT coding scheme with and without our saliency-aware method (Sal.) constrained by the same bit budget for compression. Bit budget was determined based on the budget required by the corresponding iteration of our method.

Dataset	Vol: SSIM						Blk: SNR (dB)		PSNR (dB)
	Iter. 1		Iter. 2		Iter. 3		Iter. 1		
	W/ Sal.	W/O Sal.	W/ Sal.	W/O Sal.	W/ Sal.	W/O Sal.	W/ Sal.	W/O Sal.	W/O Sal.
Abdomen CT	0.9975	0.9868	0.9975	0.9869	1	0.987	<b>Inf</b>	27.86	44.90
Brain MRI	0.9984	0.9883	0.9984	0.9983	1	0.9884	<b>Inf</b>	31.98	44.10
Bonsai	0.7745	0.9649	0.9373	0.9671	1	0.9704	<b>Inf</b>	27.51	42.36
Engine	0.4186	0.9418	0.815	0.9447	1	0.9532	<b>Inf</b>	32.11	43.94
Foot	0.8016	0.9678	0.9919	0.9743	1	0.9754	<b>Inf</b>	22.44	41.37
Heart CT	0.6814	0.9694	0.893	0.9771	1	0.9884	<b>Inf</b>	24.25	41.85
VisKorean	0.6609	0.9787	0.8889	0.9811	1	0.9817	<b>Inf</b>	31.27	42.83
VisMale	0.8353	0.9649	0.9586	0.9698	1	0.9702	<b>Inf</b>	17.06	41.12

Moreover, in the third iteration, our approach is better than the compression without saliency maps. Additionally, we used

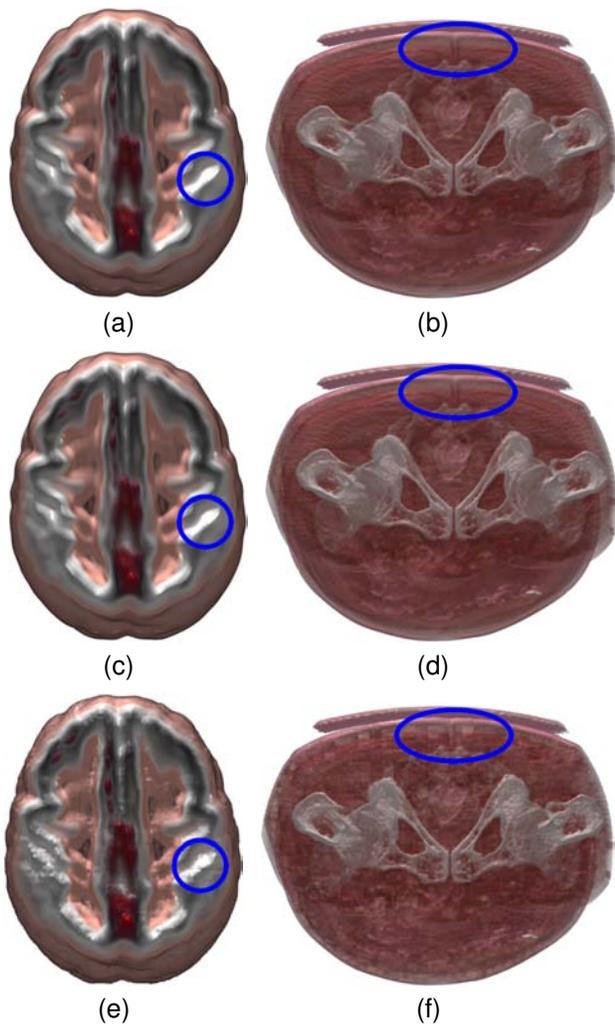


Fig. 9: Compression with and without saliency maps. (a), (c) and (e) Brain MRI and (b), (d) and (f) abdomen CT. (a) and (b) show the original data. (c) and (d) show the result of SPIHT with a saliency map. (e) and (f) show the results from SPIHT without a saliency map. The size of the compressed volume from both of our methods and the SPIHT without saliency are the same.

the signal-to-noise ratio (SNR) and the peak signal-to-noise ratio (PSNR) to measure noise only for the blocks within a given saliency interval in the first iteration. Our method achieves perfect reconstruction in the salient regions while the standard method without saliency suffers from noise in those areas (Table IV). PMS-SSIM might not be an optimal way of qualitative comparison of the decompressed volumetric data. We think that it may be subjective to the viewer whether an image with additional noise is acceptable (Fig.9), so we included this aspect in our subjective evaluation.

**Wavelet-based scalable coding vs. H.264 & HEVC.** Our method is based on a wavelet-based scalable coding. Recently, H.264 and HEVC have been used for 3D lossy compression and even for 3D lossless compression [60]. We ran H.264 and HEVC (Main-Rext profile) with inter-frame prediction mode and computed the compression rates (Table V). For

TABLE V: Compression rate (bits per voxel (BPV)) comparison. We compare the compression rate between H.264 and HEVC. We also compute the BPV difference between HEVC and a wavelet based scalable coding (EBCOT). All the datasets are in 8-bit precision.

Dataset	Size	H.264	HEVC	$\Delta$ BPV
Engine	256×256×128	2.00	1.74	-0.13
Bonsai	256×256×256	1.35	1.16	-0.27
Foot	256×256×256	2.05	1.82	-0.09
Heart	256×256×256	2.26	1.99	-0.11
VisMale	256×256×256	0.89	0.80	-0.15
Heart1	512×512×512	1.93	1.58	<b>0.24</b>
Heart2	512×512×512	1.85	1.51	<b>0.25</b>
Heart	512×512×1024	1.89	1.54	<b>0.03</b>

TABLE VI: Compression rate (bits per voxel) comparison. We compare the compression rate for multi-iteration (Iter.) and transmission based on our saliency-guided scheme, as well as the total combined sum of all iterations to the lossless compression ratio for the full dataset using Haar, CDF 5/3 and 9/7 wavelets, using the SPIHT coding scheme. The datasets are 8-bit,  $256^3$ .

Wavelet	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Combined	Full dataset
Haar	0.68	0.15	0.02	0.46	1.31	1.18
CDF 5/3	0.58	0.38	0.05	0.73	1.74	1.04
CDF 9/7	0.86	0.70	0.12	1.10	2.78	1.41

volumetric datasets, HEVC outperformed H.264. HEVC outperformed a wavelet-based scalable coding (EBCOT) for small volumetric datasets ( $\leq 256 \times 256 \times 256$ ). However, for large volumetric datasets ( $\geq 512 \times 512 \times 512$ ), the wavelet-based scalable coding was better than HEVC in terms of compression rates. Medical data, one of our target domains, contains more noise as compared to many other 3D volumetric data. This might explain why the wavelet based coding is better than HEVC for medical datasets with high resolution. Modern medical scanners produce large volumes of data per patient at high resolution, so the wavelet coding can be optimal for these large medical datasets.

**Types of wavelet transforms.** Earlier, we mentioned the selection of the Haar wavelet transform. While this transform can be replaced for analysis of spatial and temporal redundancy, in our case, it provides a better compression ratio. It is intuitively correct because the Haar wavelet results in a higher proximity of transform coefficients. Construction of the coefficient maps directly benefits from the proximity of the coefficients, as more of them can be grouped together. Table VI shows a numeric comparison of the resulting compression ratios for the Visible Korean dataset based on the Haar wavelet transform and CDF 5/3 and 9/7 wavelet transforms when used with the SPIHT coding scheme.

### B. Subjective Evaluation

To evaluate the quality of saliency maps subjectively, we conducted a user study consisting of two parts: an eye tracking based study of the data saliency and a subjective quality evaluation of the results of progressive transmission. A total of 17 participants took part in the study (3 female), ages 23 to

TABLE VII: Performance (AUC) comparison for saliency maps between the proposed model and other existing methods [7, 19].

Dataset	Orientation	Itti et al. [7]	Imamoglu et al. [19]	our model
Abdomen CT	1	0.73	0.73	0.77
Abdomen CT	2	0.71	0.69	0.76
Bonsai	1	0.76	0.77	0.77
Bonsai	2	0.79	0.80	0.79
Heart CT	1	0.75	0.77	0.77
Heart CT	2	0.72	0.76	0.77

37 (mean  $29.4 \pm 4.5$ ). All of the participants were asked about their experience in visualization (mean  $3.06 \pm 1.56$ ) and image processing (mean  $3.06 \pm 1.3$ ) on a scale from 1 (novice) to 5 (expert).

**Eye tracking based evaluation results.** In the eye tracking experiment, participants were asked to look at three datasets such as bonsai, heart, and abdomen CT from two different orientations each for 5 seconds. We used the Pupil Labs eye-tracker which can record eye movements continuously at 120Hz. The study was conducted on a 15-inch LCD monitor with a resolution of  $1920 \times 1080$ , placed at a distance of 60cm from the subjects. Between each viewing, a square with a calibration circle in one of the corners was shown for 2 seconds to the user to avoid any artificial center bias [61]. The locations of the circles were counterbalanced between participants. Since there are no standard metric to evaluate similarity between eye fixation density maps and 3D saliency maps, we used the area under the receive operating characteristics curve (AUC) exploited in previous work [62] for evaluating stereoscopic 3D visual saliency. Table VII shows that our method is as good/better as previously proposed methods [7, 19]. Our goal is to transmit salient parts/objects first, so we need to detect salient objects, not salient points/voxels. However, the eye fixation density maps can provide us with only gazed points, so evaluation with the eye fixation density maps can reveal the effectiveness of our saliency maps partially. For example, our method selects more region of the heart, as compared to [19] (Fig. 10).

**Subjective quality evaluation results.** In this part of the study participants were asked to compare the resulting renderings of the datasets compressed with and without saliency against the original data. The study was conducted on a 27-inch LCD monitor with a resolution of  $2560 \times 1440$ , placed at a distance of 60cm from the subjects. For each comparison the user was asked the following questions:

- Which rendering is visually better as compared to the original?
- Is a rendering image similar to the original?
- Can you see salient parts clearly as compared to the original?

The results of the later two questions were recorded on a 7 point Likert scale for both renderings with and without saliency. Each participant was asked to perform 72 comparisons (3 datasets x 2 rotations x 2 clipping planes x 2 zoom factors x 3 iterations).

Table VIII summarizes selections broken down by dataset, their rotations, zoom factor, presence of the clipping plane, and iteration. A two-sided Chi-square test was performed on

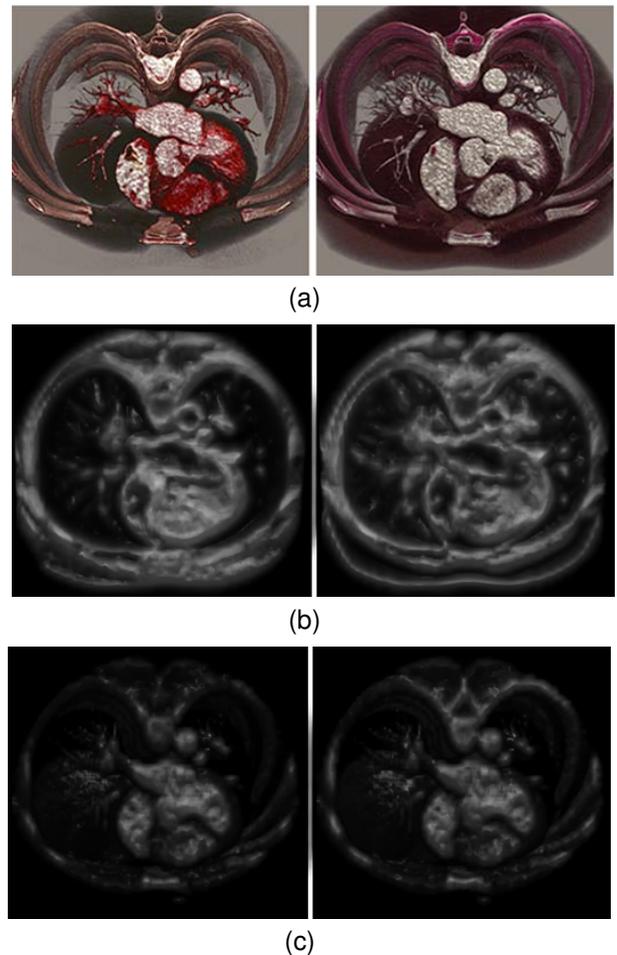


Fig. 10: Comparison of saliency maps obtained with two different TFs (left and right columns). (a) Rendered datasets, (b) with averaged IWT-based saliency maps, and (c) with our DCT-based saliency maps.

all of the levels of independent variables (in Table VIII) to determine whether there is no preference between methods with and without saliency. All of the variables have shown to have statistical significance ( $p < 0.001$ ). Hence, two methods cannot be considered to have the same subjective quality. As the percentages in Table VIII show, participants have indicated more preferences towards the method with saliency. On average, the rendering resulting from compression with a saliency-based method was selected to be visually better as compared to the original in 74.7% of the cases. Participants were more likely to select the saliency-based compression method on the third iteration (82%) of the transmission. However, the selection varied among the datasets.

Likert scores were analyzed using two-way repeated ordinal regression with cumulative link mixed models (CLMM). While the goal of our analysis is to evaluate the effect of the iteration on the Likert score among two methods (with and without saliency), there was a difference in total number of user preferences across the datasets as well. Hence, for this analysis, first, we consider two independent variables, iteration and method, as well as possible differences between the participants (a blocking variable), assuming that some participants may be consistently lower/higher in all responses. Second, we

TABLE VIII: Summary of the user preferences in the subjective evaluation study: percent of selection of the saliency-based compression method (number of selections/number of renderings at this independent variable level).

Variables	Levels		
Iteration	I	II	III
	70% (287/408) **	72% (294/408) **	82% (335/408) **
Dataset	Abdomen CT	Heart	Bonsai
	95% (388/408) **	67% (275/408) **	62% (253/408) **
View	Zoomed	Clipped	Rotated
	75% (457/612) **	75% (456/612) **	76% (462/612) **

\*\* =  $p < 0.001$

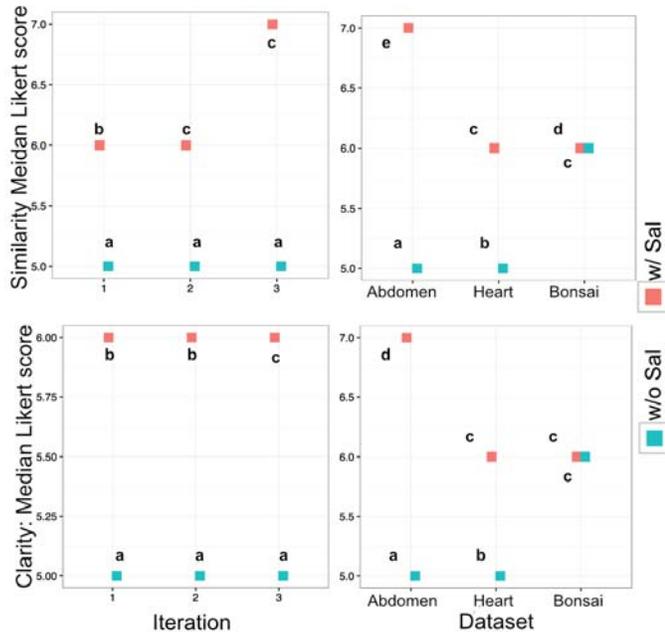


Fig. 11: Median Likert score for interactions between iteration/dataset and method.

also analyze the effect of dataset on Likert scores among two methods. We perform analysis using the R language *clmm* function from the ordinal package.  $p$ -values for the main and interaction effects were obtained using ANOVA analysis of the deviance table. When a significant effect was determined, a post-hoc analysis was performed using least square means for multiple comparisons with Tukey adjustment.

**Clarity of the rendering.** ANOVA has shown a significant main effect of both iteration ( $\chi^2(df = 2) = 13, p = 0.002$ ) and method ( $\chi^2(1) = 320, p < 0.001$ ) on user selection based on rendering clarity. Additionally, there was an interaction between method and iteration ( $\chi^2(2) = 25, p < 0.001$ ). In this and all of the analysis below  $N = 2448$ .

ANOVA has shown a significant main effect of both dataset ( $\chi^2(2) = 50, p < 0.001$ ) and method ( $\chi^2(1) = 331, p < 0.001$ ) on user selection based on rendering clarity, and an interaction between method and dataset ( $\chi^2(2) = 303, p < 0.001$ ).

**Similarity of the rendering.** ANOVA has shown a significant main effect of both iteration ( $\chi^2(2) = 27, p < 0.001$ ) and method ( $\chi^2(1) = 468, p < 0.001$ ) on user selection based on similarity of the renderings, and interaction between dataset and iteration ( $\chi^2(2) = 25, p < 0.001$ ). Additionally, ANOVA

TABLE IX: Mean Likert score for the zoom factor (zoom and no zoom) in the Bonsai dataset with (W/) and without (W/O) saliency.

Question	Dataset	Iter. 1		Iter. 2		Iter. 3	
		W/	W/O	W/	W/O	W/	W/O
Similarity	Zoom	<b>5.20</b>	5.19	<b>5.46</b>	5.25	<b>6.02</b>	5.16
	No Zoom	5.12	5.46	5.22	5.49	<b>5.96</b>	5.09
Clarity	Zoom	5.22	5.53	5.35	5.56	<b>6.10</b>	5.47
	No Zoom	5.15	5.60	5.27	5.59	<b>5.99</b>	5.47

has shown a significant main effect of both dataset ( $\chi^2(2) = 102, p < 0.001$ ) and method ( $\chi^2(1) = 476, p < 0.001$ ) on user selection on similarity of the renderings, and an interaction between method and dataset ( $\chi^2(2) = 396, p < 0.001$ ).

**Post-hoc analysis for clarity and similarity.** Because interaction terms in all of the models were significant, we plot interaction between the variables (Fig. 11). Interactions sharing a letter in each chart are not significantly different at the  $\alpha = 0.05$  level. From this it is easy to see that median Likert score for method with saliency is consistently higher for all iterations (Fig. 11 top and bottom left charts). Our method with saliency on the second and third iteration is actually considered to be within the same group (Fig. 11 top left chart), even though the median Likert scores are different. Thus, there is an indication that the our method with saliency produces equivalent results on second and third iteration. The Abdomen CT dataset appears to be in a different group from both the Heart CT and Bonsai (Fig. 11 bottom right chart). It is interesting to see this pattern because this may indicate dependence of the method not only on the data type (medical CT or other data CT) but also on the transfer function used. In Fig 11, there is little difference between two methods on median Likert scores of similarity and clarity for the bonsai dataset. However, when we zoomed the images, participants expressed that the compression with saliency was closer to the original datasets for all iterations, as compared to the one without saliency (Table IX). Moreover, at the third iteration, the compression with saliency was overall better than the one without saliency in terms of similarity and clarity.

## VII. DISCUSSION

There are several important modification that one might consider for the proposed saliency-based compression method. First, the question becomes, why not exploit IWT for the saliency computation [19] as it is already computed for the lossless compression which is the target of our application? In order to compare the IWT based saliency map to the DCT based map at the same scale, we downsampled the IWT map by averaging to the size of the DCT one (Fig. 10 (b)). Thus, the size of the computed wavelet saliency map is equal to the size of our DCT-based map, which is only  $(M/B)^3$  ( $M =$  volume size,  $B =$  block size). Selection of IWT transform results in a different type of saliency map. As Fig. 10 (b) shows, the averaged wavelet-based saliency map assigns higher saliency values to edges of structures, which is what is to be expected due to wavelet properties. The DCT-based map selects the entire region of the heart to be more salient against the lungs due to its region or block-based nature. The latter will result

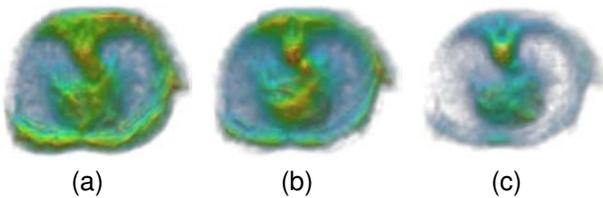


Fig. 12: (a) and (b) Saliency maps from different TFs for a heart dataset, and (c) a saliency map from averaging (a) and (b).

in the selection of the heart and areas around it to be coded and transmitted first. Note that the selection of the TF applied to the scan is arbitrary and for an illustrative purpose and does not carry any diagnostic significance.

Secondly, a modification of DCT, called overlapped DCT, may be used to reduce decompression artifacts for encoding in some applications. Our method employs DCT only for the computation of a volumetric block saliency and not for the encoding itself. Hence, such modification is not suitable in this case.

In general, the saliency of the volumetric data should be defined for the final image on the screen, so it needs to account not only for the volumetric data itself, but also for the TF that has been applied to it, and, most importantly, camera position. However, when a camera position is used, it is necessary to recompute a saliency map whenever a user interacts with the volumetric object and the camera position changes. In our scenario, this puts a significant burden on the transmission channel instead of alleviating it. Hence, we approximate the saliency of objects and generate a 3D camera-independent saliency map that fits the requirement for progressive rendering.

Saliency map generation relies on a TF which can be predefined for a specific application (for example, medical CT data), or presented in the form of thumbnail images for user selection.

There are several options to create saliency maps based on TFs. For example, one can generate a saliency map based on a single TF. Another method applies different TFs to the same dataset and averages the intensity and color from different TFs to one volume. In the latter method, the computation time for a saliency map from the final volume increases slightly (2% in Fig. 12), but the result of the method highlights common salient objects (Fig. 12) among different TFs.

### VIII. CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel framework for progressive saliency-based visualization of compressed volumetric data. Our method detects the 3D volumetric salient regions by incorporating a block-based weighted averaging (3D DCT) with a predefined or user-defined TF. The size of the created saliency map is orders of magnitude smaller than the original volume, which results in a block structure of the map. In order to utilize these maps for progressive transmission and rendering, we first separate the volume into several parts. These salient volume parts are constructed based on the ranges of the values in the saliency map. The ranges can

be constant or vary dynamically. More importantly, we reduce additional redundancy in the transform coefficients before the coding step of the compression by combining blocks with their neighbors depending on the saliency similarity. Compression and transmission of the 3D volumetric data is scheduled based on the ranges of the saliency values starting from the highest. The receiver obtains only a set of data from one saliency range at a time without losing any details, and decompresses and renders it after combining it with previously obtained data. Users can see most of the data in the first two ranges, which reduces the size of the compressed data and the decompression time compared to the lossless compression of the entire data.

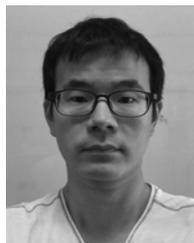
In our method, the TF must be set by the user before starting the computation of the saliency map. We plan to study a way to detect saliency of data based on clustering voxel values, gradient magnitudes, curvatures, and/or occlusion without applying the TF. An alternative to our step-by-step transmission with several separate bitstreams would be reordering a single bitstream based on the saliency order. Hence, we would obtain volumetric scalability by saliency, which is similar to scalability by VOI. Lastly, we will apply our method to other types of data such as 3D mesh and depth data. For example, users can set the priority of objects in a scene, which is similar to assigning opacity to voxels in designing TFs. Objects then are progressively transmitted based on their priorities.

### REFERENCES

- [1] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Optimal packet loss protection of progressively compressed 3-D meshes," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1381–1387, 2009.
- [2] J. Fu, D. Miao, W. Yu, S. Wang, Y. Lu, and S. Li, "Kinect-like depth data compression," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1340–1352, 2013.
- [3] R. Mekuria, M. Sanna, E. Izquierdo, D. C. A. Bulterman, and P. Cesar, "Enabling geometry-based 3-D teleimmersion with fast mesh compression and linear rateless coding," *IEEE Trans. Multimedia*, vol. 16, no. 7, pp. 1809–1820, 2014.
- [4] A. S. Lalos, I. Nikolas, E. Vlachos, and K. Moustakas, "Compressed sensing for efficient encoding of dense 3D meshes using model-based bayesian learning," *IEEE Trans. Multimedia*, vol. 19, no. 1, pp. 41–53, 2017.
- [5] T. He, L. Hong, A. Kaufman, and H. Pfister, "Generation of transfer functions with stochastic search techniques," *IEEE Vis.*, pp. 227–234, 1996.
- [6] L. Pantanowitz, K. Dickinson, A. Evans, L. Hassell, W. Henricks, J. Lennerz, A. Lowe, A. Parwani, M. Riben, D. Smith, J. Tuthill, R. Weinstein, D. Wilbur, E. Krupinski, and J. Bernard, "American Telemedicine Association clinical guidelines for telepathology," *Journal of Pathology Informatics*, vol. 5, no. 1, p. 39, 2014.
- [7] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

- [8] V. Setlur, T. Lechner, M. Nienhaus, and B. Gooch, "Retargeting images and video for preserving information saliency," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 5, pp. 80–88, 2007.
- [9] L. Wolf, M. Guttmann, and D. Cohen-Or, "Non-homogeneous content-driven video-retargeting," *IEEE Int. Conf. Comput. Vision*, pp. 1–6, 2007.
- [10] Y. Kim and A. Varshney, "Saliency-guided enhancement for volume visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 5, pp. 925–932, 2006.
- [11] L. Wang and A. Kaufman, "Importance-driven accessory lights design for enhancing local shapes," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 5, pp. 781–794, 2014.
- [12] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006.
- [13] H. H. Tong and A. N. Venetsanopoulos, "A perceptual model for JPEG applications based on block classification, texture masking, and luminance masking," *IEEE Int. Conf. Image Process.*, pp. 428–432, 1998.
- [14] H. S. Chang and K. Kang, "A compressed domain scheme for classifying block edge patterns," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 145–151, 2005.
- [15] Y. Zhong and A. K. Jain, "Object localization using color, texture and shape," *Pattern Recognition*, vol. 33, no. 4, pp. 671–684, 2000.
- [16] J. Wei, "Color object indexing and retrieval in digital libraries," *IEEE Trans. Image Process.*, vol. 11, no. 8, pp. 912–922, 2002.
- [17] Y. Fang, Z. Chen, W. Lin, and C.-W. Lin, "Saliency detection in the compressed domain for adaptive image retargeting," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 3888–3901, 2012.
- [18] —, "Saliency-based image retargeting in the compressed domain," *ACM Int. Conf. Multimedia*, pp. 1049–1052, 2011.
- [19] N. Imamoglu, W. Lin, and Y. Fang, "A saliency detection model using low-level features based on wavelet transform," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 96–105, 2013.
- [20] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 19–33, 2014.
- [21] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 185–198, 2010.
- [22] C. Wang and K.-L. Ma, "A statistical approach to volume data quality assessment," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 3, pp. 590–602, 2008.
- [23] H. Hadizadeh, I. V. Bajić, and G. Cheung, "Video error concealment using a computation-efficient low saliency prior," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2099–2113, 2013.
- [24] Z. Xiong, X. Wu, S. Cheng, and J. Hua, "Lossy-to-lossless compression of medical volumetric data using three-dimensional integer wavelet transforms," *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 459–470, 2003.
- [25] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, 1996.
- [26] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, 2000.
- [27] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, "Three-dimensional embedded subband coding with optimized truncation (3-D ESCOT)," *Applied & Comput. Harmonic Anal.*, vol. 10, no. 3, pp. 290–315, 2001.
- [28] S. Suter, J. Iglesias Guitián, F. Marton, M. Agus, A. Elsener, C. Zollikofer, M. Gopi, E. Gobbetti, and R. Pajarola, "Interactive multiscale tensor reconstruction for multiresolution volume visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2135–2143, 2011.
- [29] X. Xu, E. Sakhaee, and A. Entezari, "Volumetric data reduction in a compressed sensing framework," *Comput. Graph. Forum*, vol. 33, no. 3, pp. 111–120, 2014.
- [30] E. Gobbetti, J. A. Iglesias Guitián, and F. Marton, "Covra: A compression-domain output-sensitive volume rendering architecture based on a sparse representation of voxel blocks," *Comput. Graph. Forum*, vol. 31, no. 3pt4, pp. 1315–1324, 2012.
- [31] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao, "SSIM-motivated rate-distortion optimization for video coding," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 22, no. 4, pp. 516–529, 2012.
- [32] Z. Chen and C. Guillemot, "Perceptually-friendly H.264/AVC video coding based on foveated just-noticeable-distortion model," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 20, no. 6, pp. 806–819, 2010.
- [33] D. Agrafiotis, D. R. Bull, and N. Canagarajah, "ROI coding of volumetric medical images with application to visualisation," *IEEE Int. Symp. Image & Signal Processing & Analysis*, vol. 1, pp. 217–222, 2003.
- [34] K. Krishnan, M. W. Marcellin, A. Bilgin, and M. S. Nadar, "Efficient transmission of compressed data for remote volume visualization," *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1189–1199, 2006.
- [35] V. Sanchez, R. Abugharbieh, and P. Nasiopoulos, "3-D scalable medical image compression with optimized volume of interest coding," *IEEE Trans. Med. Imag.*, vol. 29, no. 10, pp. 1808–1820, 2010.
- [36] T.-C. Chiueh, C.-K. Yang, T. He, H. Pfister, and A. Kaufman, "Integrated volume compression and visualization," *IEEE Vis.*, pp. 329–336, 1997.
- [37] B.-L. Yeo and B. Liu, "Volume rendering of DCT-based compressed 3D scalar data," *IEEE Trans. Vis. Comput. Graphics*, vol. 1, no. 1, pp. 29–43, 1995.
- [38] N. Fout and K.-L. Ma, "Transform coding for hardware-accelerated volume rendering," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 6, pp. 1600–1607, 2007.
- [39] P. Ljung, C. Lundstrom, A. Ynnerman, and K. Museth, "Transfer function based adaptive decompression for volume rendering of large medical data sets," *IEEE Symp. Vol. Vis. & Graphics*, pp. 25–32, 2004.

- [40] S. Guthe and W. Strasser, "Advanced techniques for high-quality multi-resolution volume rendering," *Comput Graph*, vol. 28, no. 1, pp. 51–58, 2004.
- [41] K. G. Nguyen and D. Saupe, "Rapid high quality compression of volume data for visualization," *Comput. Graph. Forum*, vol. 20, no. 3, pp. 49–57, 2001.
- [42] Y.-S. Wang, C. Wang, T.-Y. Lee, and K.-L. Ma, "Feature-preserving volume data reduction and focus+context visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 2, pp. 171–181, 2011.
- [43] M. Balsa Rodríguez, E. Gobbetti, J. Iglesias Guitián, M. Makhinya, F. Marton, R. Pajarola, and S. K. Suter, "State-of-the-art in compressed GPU-based direct volume rendering," *Comput. Graph. Forum*, vol. 33, no. 6, pp. 77–100, 2014.
- [44] J. Beyer, M. Hadwiger, and H. Pfister, "State-of-the-art in gpu-based large-scale volume visualization," *Comput. Graph. Forum*, vol. 34, no. 8, pp. 13–37, 2015.
- [45] M. Moser and D. Weiskopf, "Interactive volume rendering on mobile devices," *Proc. Vision, Modeling, and Vis.*, pp. 217–226, 2008.
- [46] L. Campoalegre, P. Brunet, and I. Navazo, "Interactive visualization of medical volume models in mobile devices," *Personal and Ubiquitous Computing*, vol. 17, no. 7, pp. 1503–1514, 2013.
- [47] J. Noguera and J. R. Jimenez, "Mobile volume rendering: Past, present and future," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 2, pp. 1164–1178, 2016.
- [48] D. Fleischmann and D. C. Miller, "Clinical 3D and 4D Imaging of the Thoracic Aorta," *Diseases of the Heart, Chest Breast Diagnostic Imaging and Interventional Techniques*, pp. 119–130, 2007.
- [49] S. Gu, R. Gupta, and I. Kyprianou, "Computational high-resolution heart phantoms for medical imaging and dosimetry simulations," *Physics in Medicine and Biology*, vol. 56, no. 18, p. 5845, 2011.
- [50] M. Kraus and T. Ertl, "Topology-guided downsampling," *Vol. Graph*, pp. 223–234, 2001.
- [51] D. Gao and N. Vasconcelos, "Bottom-up saliency is a discriminant process," *IEEE Int. Conf. Comput. Vision*, pp. 1–6, 2007.
- [52] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," *IEEE Conf. Comput. Vision & Pattern Recognition*, pp. 1597–1604, 2009.
- [53] Y.-L. Huang and R.-F. Chang, "Texture features for DCT-coded image retrieval and classification," *IEEE Int. Conf. Acoustics, Speech, & Signal Process.*, vol. 6, pp. 3013–3016, 1999.
- [54] H. Li, G. Liu, and Z. Zhang, "A new texture generation method based on pseudo-DCT coefficients," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1300–1312, 2006.
- [55] C. Theoharatos, V. Pothos, N. Laskaris, G. Economou, and S. Fotopoulos, "Multivariate image similarity in the compressed domain using statistical graph matching," *Pattern Recognition*, vol. 39, no. 10, pp. 1892 – 1904, 2006.
- [56] A. Chalechale, G. Naghdy, and A. Mertins, "Sketch-based image matching using angular partitioning," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 1, pp. 28–41, Jan 2005.
- [57] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional transfer functions for interactive volume rendering," *IEEE Trans. Vis. Comput. Graphics*, vol. 8, no. 3, pp. 270–285, 2002.
- [58] R. Murmura, J. Medsger, A. Stavrou, and J. M. Voas, "Mobile application and device power usage measurements," *IEEE Int. Conf. Software Security and Reliability*, pp. 147–156, 2012.
- [59] A. K. Moorthy and A. C. Bovik, "Visual importance pooling for image quality assessment," *IEEE J. of Sel. Topics Signal Process.*, vol. 3, no. 2, pp. 193–201, 2009.
- [60] M. Zhou, W. Gao, M. Jiang, and H. Yu, "HEVC lossless coding and improvements," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 22, no. 12, pp. 1839–1843, 2012.
- [61] A. Borji, D. N. Sihite, and L. Itti, "Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 55–69, 2013.
- [62] J. Wang, M. P. D. Silva, P. L. Callet, and V. Ricordel, "Computational model of stereoscopic 3D visual saliency," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2151–2165, 2013.



**Ji Hwan Park** received his BS degree in computer engineering from Hongik University, Seoul, South Korea, in 2004. Currently, he is a PhD candidate in the Department of Computer Science at Stony Brook University. His research interests include visual analytics, scientific visualization, information visualization, and human computer interaction.



**Ievgeniia Gutenko** has received her PhD in computer science at Stony Brook University, Stony Brook, NY and BS in computer science and applied mathematics at Northeastern Illinois University, Chicago, IL. Her research interests include the handling, mining, and visualization of medical patient data, including segmentation, compression, and rendering.



**Arie E. Kaufman** is a Distinguished Professor and Chair of Computer Science, Director of the Center for Visual Computing (CVC), and Chief Scientist of the Center of Excellence in Wireless and Information Technology (CEWIT) at Stony Brook University. He received his PhD in Computer Science from Ben-Gurion University, Israel (1977). He is internationally recognized for his pioneering and seminal contributions to visualization, graphics, virtual reality, and their applications, especially in biomedicine. He is Fellow of IEEE, Fellow of ACM, member of European Academy of Sciences, recipient of IEEE Visualization Career Award, and was inducted into LI Technology Hall of Fame. He was the founding Editor-in-Chief of IEEE Transaction on Visualization and Computer Graphics (TVCG), 1995-1998.