Optimization of Overlapped Tiling for Efficient 3D Image Retrieval

Zihong Fan and Antonio Ortega
Signal and Image Processing Institute
Dept. of Electrical Engineering
University of Southern California, Los Angeles, CA, USA

Abstract

Remote visualization of an arbitrary 2-D planar "cut" from a large volumetric dataset with random access has both gained importance and posed significant challenges over the past few years in industrial and medical applications. In this paper, a prediction model is presented that relates transmission efficiency to voxel coverage statistics for a fast random 2-D image retrieval system. This model can be used for parameter selection and also provides insights that lead us to propose a new 3D rectangular tiling scheme, which achieves an additional 10% - 30% reduction in average transmission rate as compared to our previously proposed technique, e.g., a nearly 30% / 45% reduction in the average transmission rate at the cost of a factor of ten / fifteen in storage overhead compared to traditional cubic tiling. Furthermore, this approach leads to improved random access, with less storage and run-time memory required at the client.

I. Introduction

Researchers in many fields (e.g., biomedical imaging, earth sciences, computational fluid dynamics, etc.) need to manipulate and visualize very large datasets. In these kinds of applications, a client-server approach has been often used in order for a personal computer equipped with limited memory to be able to manipulate, visualize, and render the complete dataset. With a client-server approach (e.g., [1]–[6]), the server provides only the data needed for the specific visualization task required at the client end. We focus on situations where lower dimensional portions of a dataset need to be accessed. In this paper, arbitrary oblique planes of a 3D volume may need to be extracted and rendered, as is required in some medical imaging applications. An example of oblique plane intersection with the 3D volume is shown in Fig. 1. In many techniques proposed for volumetric image coding [2]–[6], including approaches such as JP3D [7]–[9], some form of random access is provided via a non-overlapped, independently encoded, cuboid tiling. These approaches can be inefficient in the scenario we focus on, because the only useful voxels[1] for each retrieved cubic tile are those near the intersection between the cube and the desired 2D plane. In our previous work [10], we showed that it is more efficient to use overlapping rotated tiles to represent the dataset, which leads to an increase in the average number of useful voxels per tile so that the total number of tiles to be retrieved is smaller (and hence a lower transmission bitrate is achieved). In this approach, we trade-off increased storage at the server's side for lower bandwidth during interactive access to the data-set. Our proposed system is displayed in Fig. 2 and consists of 5 components: 1) 3D rectangular tiling scheme, 2) mapping algorithm, 3) tile searching algorithm, 4) 3D compression, and 5) random oblique plane reconstruction and display. We have proposed a 3D rectangular tiling scheme and discussed components (2)

[1]A volume element, representing a value on a regular grid in three dimensional space.

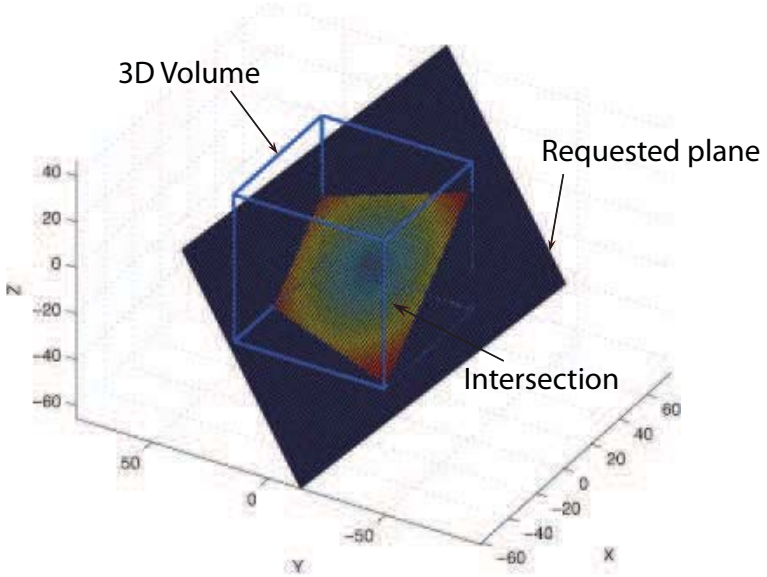


Fig. 1. Random oblique plane acquisition illustration.

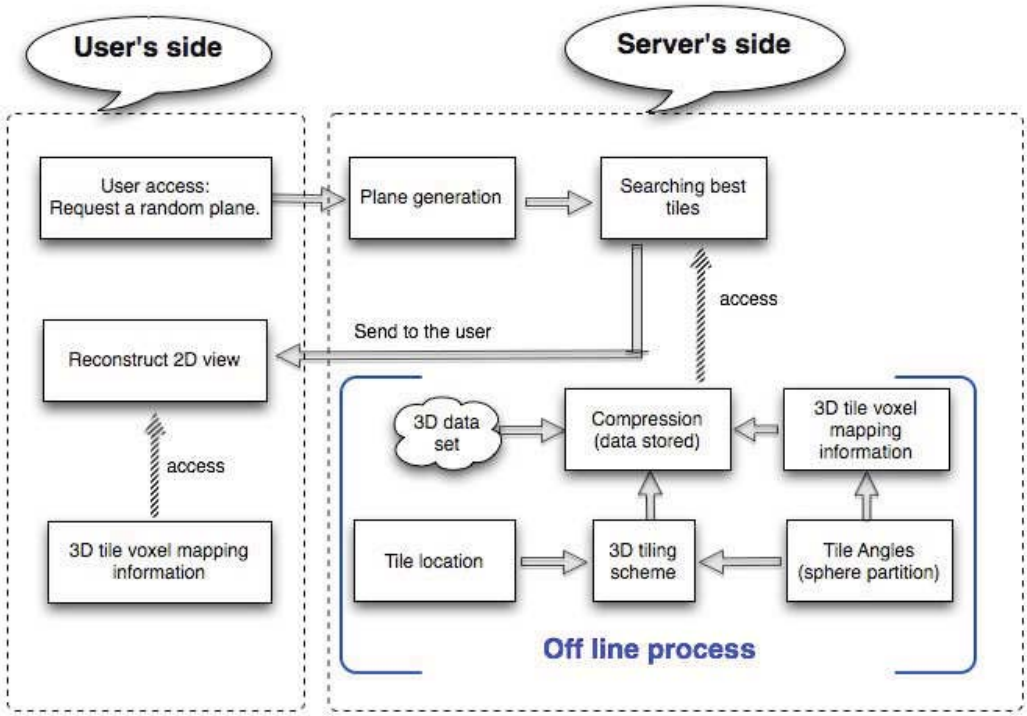


Fig. 2. Retrieval system

and (3) in [10]. The tiling scheme provides multiple redundant tilings of the 3D object, where each tile has a different orientation and is associated with one rotation center. The searching algorithm determines which tiles should be retrieved for a given query while the mapping algorithm enables efficient coding without interpolation of rotated tiles.

In this paper, we first propose a model for our system that allows us to optimize

system parameter selection without the need for simulations of system performance. The proposed model can be used to predict transmission rate and storage, given how the cells are partitioned. Then, based on the insights obtained from the proposed model, a new 3D rectangular tiling scheme is proposed, which leads to better performance and verifies the model. We demonstrate that in exchange for increased server-side storage, significant reductions in average transmission rate can be achieved relative to both the conventional cubic tiling techniques and our previously proposed approach [10]. The run-time memory space can also be reduced accordingly. Each tile is compressed and encoded independently. 3D DCT, a 3D zig-zag scanning order and VLC are used in our compression experiments. Random oblique plane display allows the acquisition and display of the requested plane from the selected overlapped rectangular tiles. Due to space constraints, details of the 3D compression and the random oblique plane reconstruction/display we use here are left for a future paper.

We start by providing a prediction model for parameter selection in Section II, followed by presenting a new 3D rectangular tiling scheme in Section III. Then in Section IV, a performance comparison of the proposed 3D rectangular tiling scheme with the previous tiling scheme [10] is provided. The new tiling scheme achieves an additional 10% - 30% reduction in the average transmission rate compared to the previous rectangular tiling scheme. Finally, we conclude the paper in Section V.

II. PREDICTION MODEL

In our previous work [10], we were able to achieve reductions in transmission rate of 15% - 50%, as compared to the conventional tiling scheme, depending on the desired trade-off with the storage overhead. For a fixed storage overhead, different values of distance between rotation centers and the number of rotation tiles around each rotation center achieve different levels of transmission efficiency. Thus, in designing such redundant storage systems it is important to determine how to select parameter settings in order to achieve the best transmission efficiency for a given storage overhead. In this section, a model is proposed for the parameter selection that provides insight as to why different parameter settings produce different transmission efficiencies with a fixed storage overhead. With this insight we then propose a new tiling scheme (Section III), which leads to improved performance compared to our previous scheme.

We start by noting that with a redundant tiling strategy each voxel (or pixel) is available from more than one tile. Intuitively, since each voxel is equally likely to be selected, it would make sense for all voxels to be available from a similar number of tiles. Thus our model provides a way to estimate relative transmission efficiency by computing the mean and standard deviation of the number of tiles per voxel. While we use our previous tiling scheme to illustrate this model, it can be readily applied to other tiling approaches (e.g., to the new tiling scheme of Section IV). Let $L \times W \times H$ be the size of each tile (with $L = W$ and $L, W > H$ in this paper) and let N be the number of rotation angles we use, chosen using sphere tesselation (refer to [10]), so that the N norm vectors of the tiles are $\tilde{\mathbf{n}} = \{\vec{n}_1, \vec{n}_2, \vec{n}_3, \cdots, \vec{n}_N\}$. Tiles are thus rotated by selecting one of the N angles. In our previous work [10] multiple tiles with different orientations shared the same rotation center, while in our new approach, the tile centers can be different.

In order to compute the mean and standard deviation of the number of tiles per voxel, assume that a tile has a random orientation around a rotation center. More precisely, we

select a tile with norm vector $\tilde{\mathbf{n}}' = \mathbf{R}_\phi \mathbf{R}_\varphi \tilde{\mathbf{n}}$, where φ and ϕ are independent random variables with uniform distribution $\mathbf{U}(0, 2\pi)$. $\mathbf{R}_\phi$ and $\mathbf{R}_\varphi$ are the corresponding rotation matrices. Since φ and ϕ are independent random variables with uniform distribution $\mathbf{U}(0, 2\pi)$, the possible norm vectors of the tiles form a unit sphere.

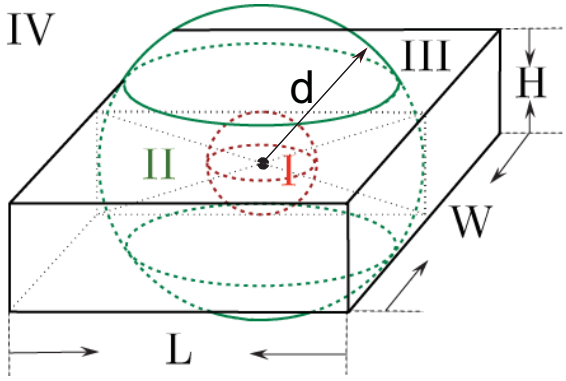


Fig. 3. Model for calculating the average number of tiles covering an arbitrary voxel around a rotation center.

Based on this we derive expressions for the coverage probability, $P(x, y, z)$, i.e., the probability that a voxel at location (x, y, z) will be covered by a randomly rotated tile. Let $d = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}$ be the distance between this voxel and the tile rotation center at (x_c, y_c, z_c). Fig. 3 illustrates the possible cases to consider. Intuitively, the coverage probabilities are higher for voxels closer to the rotation center. For example, as shown in Fig. 3, voxels in the range I (inside the ball with radius H) can always be covered and voxels in the range IV can never be covered, regardless of the rotation angle of the tile. From Fig. 3, notice that the coverage probability of an arbitrary voxel depends only on d: $P(x, y, z)$ can be obtained as the ration between i) the area of the intersection of a sphere of radius d and a tile with arbitrary orientation and ii) the area of the sphere:

$$A = \int_0^{2\pi} \int_\theta^{\pi-\theta} \sin\alpha d\alpha d\varphi, \qquad \text{where} \quad \theta = \arcsin(\frac{H}{2d}) \tag{1}$$

$$P(x, y, z) = \frac{A}{4\pi}. \tag{2}$$

The four cases illustrated in Fig. 3, for different ranges of d, lead to different expressions for $P(x, y, z)$ which are summarized in Table I.

Range	Distance(d)	Coverage Probability
I	$d \leq \frac{H}{2}$	$P(x, y, z) = 1$
II	$\frac{H}{2} < d \leq \frac{L}{2}$	$P(x, y, z) = 1 - \frac{1}{2}(\cos\theta - \cos(\pi - \theta))$, where $\theta = \arcsin(\frac{H}{2d})$.
III	$\frac{L}{2} < d \leq \frac{1}{2} \times \sqrt{L^2 + W^2 + H^2}$	$P(x, y, z) \approx 0$
IV	$d > \frac{1}{2} \times \sqrt{L^2 + W^2}$	$P(x, y, z) = 0$

TABLE I
COVERAGE PROBABILITY FOR ANY VOXEL BY ONE TILE WITH RANDOM NORM VECTOR.

For the tiling scheme of [10], assume we have N tiles around a rotation center, with orientations chosen independently with an identical uniform angle distribution, as discussed before. Then, the average number of tiles covering a voxel is $P(x, y, z) \times N$. Alternatively, assume there are multiple rotation centers in the neighborhood $\mathcal{N}\{j\}$ of voxel j. Here we consider neighboring rotation centers to be those that are within a distance $\sqrt{(L/2)^2 + (W/2)^2 + (H/2)^2}$ of a given voxel. Then the average number of tiles covering voxel j is $T_j = \sum_i P_i(x, y, z) \times N$, where P_i denotes the coverage probability for one of the rotation centers i in $\mathcal{N}\{j\}$ and (x, y, z) corresponds to the location of voxel j. This coverage probability will depend on the distance between the voxel and each of the neighboring rotation centers.

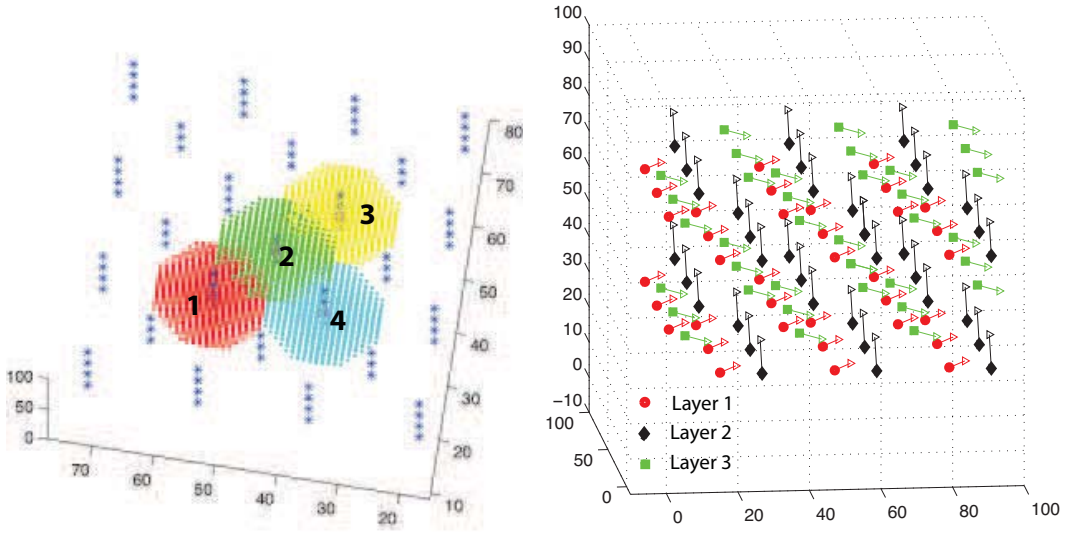
The mean (μ) and standard deviation (σ) of the tile coverage for a whole volume can be computed based on one Voronoi region spanned by the rotation centers, given that regular tilings will be used (i.e., the same tiling structure will be repeated throughout the volume). The Voronoi region associated with rotation center C_i is defined by

$$V(C_i) = \{p : d(p, C_i) \leq d(p, C_j),\ j \neq i, \quad i, j \in I_n\} \tag{3}$$

where p here denotes a voxel at position (x, y, z) in 3D space and $d(p, C_i)$ is the distance between voxel p and rotation center C_i. Denote $M = |V(C_i)|$, the number of voxels in $V(C_i)$. Then μ and σ can be calculated by using one Voronoi region,

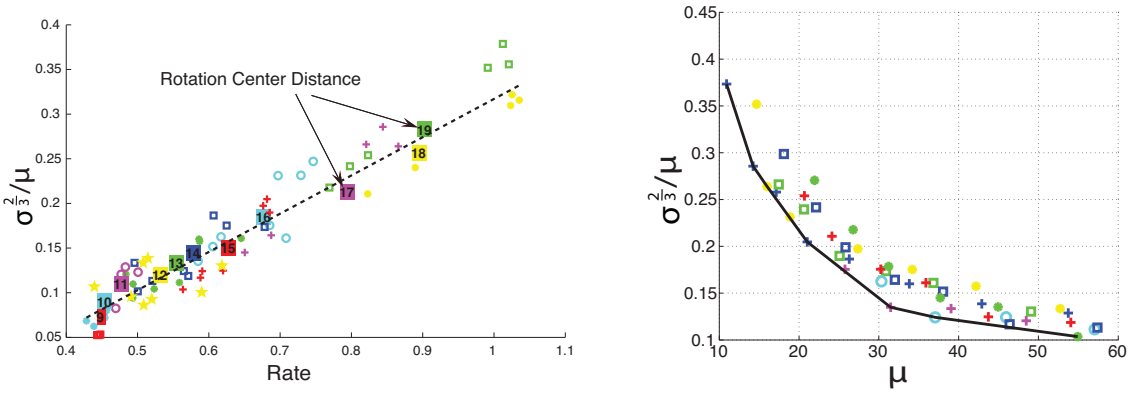
$$\mu = \frac{1}{M}\sum_{j=1}^{M} T_j \qquad \sigma = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(T_j - \mu)^2}. \tag{4}$$

Fig. 4(a) shows the 3D Voronoi regions. Each numbered shaded area represents one Voronoi region. Since the rotation centers are located at points on a regular octahedron grid patten (the rotation centers are located at the regular octahedron vertices), the Voronoi regions are all the same polyhedrons.



(a) (b)

Fig. 4. In (a), 3D Voronoi region: The star points are the locations of the rotation centers. Each number indicates one Voronoi region. In (b), New 3D rectangular tiling scheme. The tile centers for each layer are shown. 3 layers of tiles are in this figure. The arrows show the norm vector $\vec{n}_i$ representing the tile angle.



(a) Linear regression: $\sigma^{\frac{2}{3}}/\mu$ vs. Rate. (b) $\sigma^{\frac{2}{3}}/\mu$ vs. μ, ($\mu < 60$)

Fig. 5. The tile coverage statistics (σ and μ) and the transmission rate.

Using linear regression, Fig. 5(a) shows that there is a highly linear relationship between the ratio $\sigma^{\frac{2}{3}}/\mu$, which characterizes the tile coverage, and the transmission rate[2]. The reason for using the third root of the variance $\sigma^{\frac{2}{3}}$ is the dimensionality of 3D volume data. Fig. 5(b) represents $\sigma^{\frac{2}{3}}/\mu$ as a function of the average storage (μ) for different parameter settings, focusing on the range of $0 \leq \mu \leq 60$. Note that the values of μ shown in Figs. 5(a) and 5(b) are calculated from equations (3) and (4). From Fig. 5(a), we observe that in order to achieve a lower transmission rate, a lower $\sigma^{\frac{2}{3}}/\mu$ is preferred. Hence, for the same value μ, the parameter setting which produces the smallest $\sigma^{\frac{2}{3}}/\mu$ should be chosen. Notice that in our previously proposed tiling scheme [10], for the same value μ, decreasing the rotation center distance is better than increasing the rotation angles (equivalent to increasing the number of tiles per rotation center) because decreasing the rotation center distance will lead to smaller values of $\sigma^{\frac{2}{3}}/\mu$. In order to find the optimal parameter settings considering both μ and $\sigma^{\frac{2}{3}}/\mu$ to be small, from Fig. 5(b), a polynomial may be fit to the lower portion of the convex hull defined by the points in the figure, the associated parameters of which would be preferred for selection.

III. New 3D rectangular tiling scheme

We now use the results of the previous section to motivate our proposed tiling scheme. As shown by in Figs. 5(a) and 5(b), a lower variation in the number of tiles per voxel is desirable. The approach of [10] centered multiple tiles with different orientations around each location center, leading to high variability in coverage (all tiles covered voxels near the rotation center, while only a few tiles covered those voxels further away). As an alternative, we no longer place multiple tiles around each rotation center. More importantly, we propose a technique to search for tile positions and angles *in order to minimize overlap with previously chosen tiles*. This ensures that tiles are more evenly distributed and the overall variance of tile coverage is reduced.

In our new tiling scheme we define multiple "layers" of tiles, where the tiles for each layer have the same angle and are located uniformly throughout the volume, as shown in Fig. 4(b). Each tile is represented by the tile center location p and the norm vector $\vec{n}_i$ representing the tile angle. For each layer, the tiles could be arranged according to a cuboctahedral, octahedral, icosahedral or tetrahedral pattern; in this paper we locate

[2]Rate = transmission bits / points in the plane. The rate is averaged over 80 random planes.

the tiles (corresponding to a set of tile centers $P = \{p_1, p_2, \cdots\}$) at points on a regular octahedron grid patten (the tile centers are located at the vertices of an octahedron). There are N layers in total, each with different tile angles. The N norm vectors (one for each layer) are obtained by using the equal sphere tessellation method [11] and are represented with a common origin and with their end points uniformly spread out on a 3D sphere. When two vectors with exactly opposite directions are part of the tessellation, only one of them is used as the norm vector [10].

The new tiling scheme aims to decrease the standard deviation in the number of tiles that cover a typical voxel. In this scheme, starting with first layer tiles with arbitrary locations on a regular grid pattern and an arbitrary angle, the new layer tiles are determined by minimizing the maximum overlap value between the new tiles and the existing tiles. We then record the new layer of tiles in the existing tile list, and proceed to find the next layer of tiles until all the layers (with different angles) are determined. Intuitively, the 3D rectangular tiling method uses an over-complete set (overlapped rectangular tiles) to represent the 3D dataset, while the traditional method uses an orthogonal set (non-overlapped cubic tiles) to represent the 3D dataset. Minimizing the value of the maximum overlap between the new tile and the existing tiles leads to less correlation within the over-complete set, which then leads to greater transmission efficiency, as shown in Fig. 6(a).

We now describe the algorithm we propose to select tile locations and angles for each of the layers. Let $B \equiv \{1, 2, \ldots, N\}$ be the set of angles indexes. We let (P_i, I_i) denote the selected tiles for layer i, where $P_k = \{p_{k1}, p_{k2}, \cdots\}$ are the k^{th} layer tiles locations and $I_k \in B$ is the k^{th} layer angle index. Let B^k be the set of angles used up to the k^{th} layer so that $B - B^k$ is the set of angles unused up to k^{th} layer. We let A^k denote the set of tiles that have been assigned to the first k layers. Initially $A^0 = \emptyset$ and $B^0 = \emptyset$, but these sets grow as the algorithm assigns tiles to the layers. Note that each angle index will be assigned to one of the layers, such that at the end of the entire algorithm we will have $B - B^N = \emptyset$. For each layer Algorithm 1 searches for the tile positions/angles that minimize the maximum overlap with tiles in all the previously chosen layers. Algorithm 2 is called in order to break ties, whenever multiple angles and positions lead to the same maximum overlap.

Since the tile centers of all layers are on regular octahedron grids (with the grid in one layer typically shifted with respect to those of other layers) it follows that once we know the position of one tile location in a given layer, we know the locations of all the tile centers for that same layer. Fig. 4(a) shows the 3D Voronoi regions, with each color representing a separate region. Since the Voronoi regions are the same within each layer, we only need to determine the location and the angle index of one tile in the center Voronoi region, denoted by V, in order to generate all the tiles for that layer. We let C_k denote the tiles in A_k that are inside or close to the Voronoi region. Let $O(\{p_1, I_1\}, \{p_2, I_2\})$ denote the overlap between two tiles $\{p_1, I_1\}$ and $\{p_2, I_2\}$ and let

$$O_{max}(\{p_1, I_1\}, C^k) = \max_{\{p_m, I_m\} \in C^k} \{O(\{p_1, I_1\}, \{p_m, I_m\})\}$$

be the maximum overlap between tile $\{p_1, I_1\}$ and all the tiles in C^k. We then search

for solutions $\{p_k^*,\ I_k^*\}$[3] that minimize the maximum overlap, i.e.,

$$\{p_k^*, I_k^*\} = \arg\min \max_{\{p_m, I_m\} \in C^k} \{O(\{p_k, I_k\}, \{p_m, I_m\})\},$$

where $p_k \in V$, and $I_k \in B - B^{k-1}$. The new tile $\{p^*,\ I^*\}$ minimizes the maximum overlap with previously chosen tiles. For a given tile location p, it is possible that a continuous range of tile angles may be equally optimal due to the fact that tile overlap is measured by the number of discrete grid points that are contained in the overlapping regions. In this case, Algorithm 2 chooses the median angle index. For example, if tiles $\{p,\ 3\}, \{p,\ 4\}, \{p,\ 5\}, \{p,\ 6\}, \{p,\ 7\}$ are equivalent in terms of overlap, then tile $\{p,\ 5\}$ will be selected. Similarly, if the tile at location p^* minimizes overlap, then it may be possible for some points $(x, y, z) \in \mathcal{N}\{p^*\}$ to achieve the same maximum overlap. Let $\mathbf{S}$ denote the set containing all the points $(x, y, z) \in \mathcal{N}\{p^*\}$ (including p^*) that achieve the minimum max-overlap value. Algorithm 2 chooses the central point of $\mathbf{S}$ as the selected tile location p^{**} with its associated angle index I^{**} to be the new tile in the Voronoi region for the next tile layer. In order to speed up the search, we partition the voronoi region into several 3D subregions. At the center point of each subregion, the maximum overlap value is calculated. Then, we pick the subregion with the minimum value to be used for the next level partition. The partition procedure stops once the selected subregion is very small (e.g., less than 70 voxels within the final selected subregion in our experiments). Algorithm 1 lines 5-7 will then be employed for the final selected subregion, rather than using V. In our experiments, this procedure speeds up the search time dramatically.

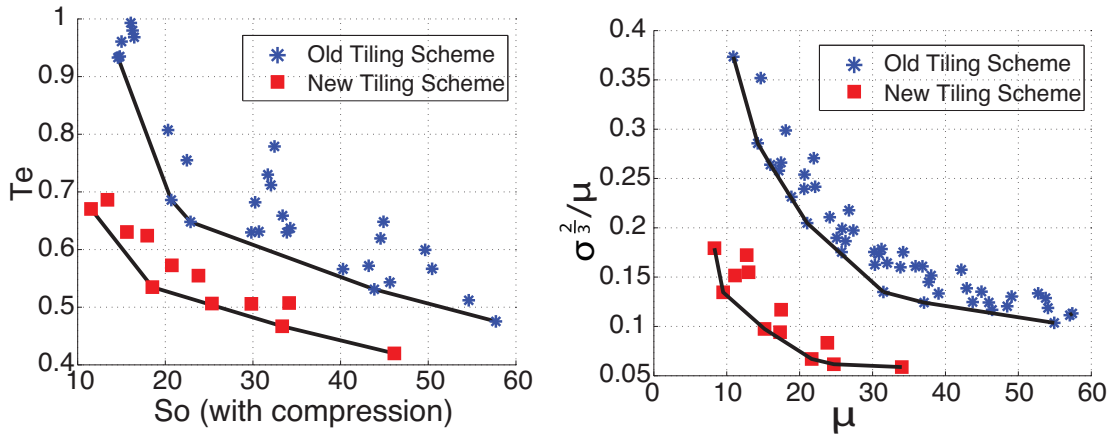
Algorithm 1 New 3D Rectangle Tiling Algorithm

1: Starting with first layer tiles $\{P_i, I_i\}$ with arbitrary locations on a regular octahedron grid pattern and with tile angle index $I_i = 1$. $A^0 = \emptyset$, $B^0 = \emptyset$.
2: **for** k = 1 to N **do**
3: Sort A_k according to the distance from the tile center location to the center of the 3D data.
4: Find C_k.
5: **for** $\forall\ p_k \in |V|$ and $\forall\ I_k \in B - B^{k-1}$ **do**
6: $\{p_k^*,\ I_k^*\} = \arg\min O_{max}(\{p_k, I_k\}, C_k)$
7: **end for**
8: The parameter pairs achieving the minimum value of the maximum overlap may not be unique. Algorithm 2 is used to select the proper one $\{p_k^{**},\ I_k^{**}\}$.
9: Using the new tile $\{p_k^{**},\ I_k^{**}\}$, generate all tiles $\{P_k, I_k\}$ for layer k.
10: $A^k = A^{k-1} \cup \{P_k, I_k\}$. $B^k = B^{k-1} \cup I_k$.
11: **end for**

Algorithm 2 Adjustment Algorithm

for $\forall\ p_k \in |\mathcal{N}\{p_k^*\}|$ and $\forall\ I_k \in B - B^{k-1}$ **do**
$\mathbf{S} = \{\{p^*, I^*\}\ |\{p^*, I^*\} = \arg\min O_{max}(\{p_k, I_k\}, C_k)\}$
3: **end for**
Find the central point p^{**} of $\mathbf{S}$.
if Continuous angles lead the same minimum max-overlap value **then**
6: The median angle index will be chosen as I^{**}.
end if
Return $\{p^{**},\ I^{**}\}$.

[3] * / ** denotes the result before / after the adjustment Algorithm 2, respectively.



(a) 3D case result: T_e versus S_o.

(b) $\sigma^{\frac{2}{3}}/\mu$ vs. μ

Fig. 6. In (a), T_e denotes the ratio of required bandwidths for the rectangular and cubic tiling schemes. S_o denotes the relative storage overhead required by the rectangular tiling scheme. $T_e = T_r/T_s$ $S_o = S_r/S_s$. In (b), μ and σ are the average and the standard deviation of the tile coverage.

IV. Experimental results

The performances of the proposed 3D rectangular tiling scheme and the tiling scheme in [10] are compared in Fig. 6(a). The two tiling schemes are also compared in Fig. 6(b) by using the model proposed in Section II, which shows that the model is also useful for designing new tiling schemes. For all the experimental results in this section, MRI volume data for a patient's head[4] of size $256 \times 256 \times 256$ is used. After mapping the Cartesian grid points to neighboring points in the rotated rectangular tiles, 3D compression is applied independently to each tile and stored on the server.

Performance is shown compared to a standard tiling strategy (with cubic, non-overlapping tiles). The rectangular tile size was fixed at $W = 32$, $L = 32$ and $H = 8$. The side of a cubic tile was 16. T_r and T_s indicate the total numbers of bits transmitted in the rectangular and cubic tiling modes, respectively, and $T_e = T_r/T_s$ denotes the ratio of required bandwidths for the rectangular and cubic tiling schemes. S_r and S_s are the total storage sizes when using the rectangular and cubic tiling, respectively, and $S_o = S_r/S_s$ is the relative storage overhead required by the rectangular tiling scheme. Fig. 6(a) shows the trade-off between the ratio of required bandwidth T_e and the storage overhead S_o while we fix one and vary the other by changing the distance between tile centers and the number of different tile angles N. N varies by using the tile angle selection method in [10]. As shown in Fig. 6(a), the proposed new tiling scheme leads to an additional increase in random access transmission efficiency of 10% - 30% when compared to the previous proposed tiling scheme in [10], depending on the chosen storage overhead (S_0). Intuitively, the 3D rectangular tiling methods use an over-complete set (overlapped rectangular tiles) to represent the 3D dataset. Therefore, less correlation between the over-complete set leads to greater transmission efficiency. In addition, Fig. 6(a) shows that for the same storage overhead (S_o), different transmission efficiencies (T_e) are achieved by using different parameter settings (the distance between the tile centers and the number of tile rotation angles). By using the model in Section II, the parameter setting to achieve the best transmission efficiency (given a fixed storage overhead) can be chosen without experimental tests, shown in Fig. 6(b). Smaller variance will lead to greater transmission

[4]This volume data was downloaded from http://www9.informatik.uni-erlangen.de/External/vollib/.

efficiency. Intuitively, this is because by decreasing the coverage variance across all voxels it follows that all voxels will tend to be covered by a similar number of tiles.

V. Conclusions

In this paper a prediction model was presented that provides a way to relate transmission efficiency to voxel coverage statistics (mean value and standard deviation) for a fast random 2D image retrieval system. This model enables designers to select parameter settings that achieve the best transmission efficiency without performing experimental tests and also provides us with insight on designing a new tiling scheme. We have thus proposed a new 3D rectangular tiling scheme which achieves an additional 10% - 30% reduction in the average transmission rate compared to the previous rectangular tiling scheme, e.g., a nearly 30% / 45% reduction in the average transmission rate at the cost of a factor of ten / fifteen in storage overhead compared to traditional cubic tiling. We have demonstrated that in exchange for increased server-side storage, significant reductions in the average transmission rate can be achieved. Furthermore, compared to the conventional tiling scheme the client side benefits from random data access and relaxed storage requirements. The run-time memory space can also be reduced accordingly. Our system creates a more interactive and responsive interface for humans to access 3-D datasets and has the potential to be used on mobile devices and web applications to allow for fast random access to oblique planes from volume datasets in a variety of potential applications.

References

[1] H. Lalgudi, M. Marcellin, A. Bilgin, and M. Nadar, "Lifting-based view compensated compression of volume rendered images for efficient remote visualization," in *DCC-Proc*, 2008.

[2] I. Ihm and S. Park, "Wavelet-based 3D compression scheme for interactive visualization of very large volume data," *Computer Graphics Forum*, March 1999.

[3] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec using set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, pp. 243–250, Jun. 1996.

[4] Y. Cho, A. Said, and W. A. Pearlman, "Low complexity resolution progressive image coding algorithm: PROGRES(PROGressive RESolution decompression)," in *IEEE ICIP*, 2005.

[5] S. Muraki, "Volume data and wavelet transforms," in *IEEE Trans. Computer Graphics and Application*, July 1993.

[6] Y. Cho and W. A. Pearlman, "Hierarchical dynamic range coding of wavelet subbands for fast and efficient image compression," *IEEE Trans. Image Processing*, vol. 16, no. 2005-2015, Aug 2007.

[7] "Information technology - JPEG 2000 image coding system: Part 10 - extensions for three-dimensional data (jp3d) - fcd v1.0," *ISO/IEC JTC1/SC29/WG1 N4101*, 2006.

[8] T. Bruylants, A. Munteanu, A. Alecu, R. Deklerck, and P. Schelkens, "Volumetric image compression with JPEG2000," in *SPIE The International Society for Optical Engineering*, 2007.

[9] J. P. W. Pluim and J. M. Reinhardt, "Compression of medical volumetric datasets: physical and psychovisual performance comparison of the emerging JP3D standard and JPEG2000," in *SPIE, Medical Imaging 2007: Image Processing*., vol. 6512, 2007.

[10] Z. Fan and A. Ortega, "Overlapped tiling for fast random access of 3-d datasets," in *Proc of Data Compression Conference (DCC)*, 2009.

[11] Coxeter and H. S. M., *Regular Polytopes*, 3rd ed., ser. ISBN 0-486-61480-8. New York: Dover Publications., 1973.