Automatic View Selection for Volume Rendering

Udeepta D. Bordoloi and Han-Wei Shen

The Ohio State University, Columbus, USA

Abstract

In fields such as computer graphics and computer vision, viewpoint comparison techniques have been used to find good views of a given scene, or to find a minimal set of representative views which capture the entire scene. In this paper, we present a view selection method designed for volume rendering. We introduce a viewpoint goodness measure based on the formulation of entropy from information theory. The measure takes into account the transfer function, the data distribution and the visibility of the voxels. Our framework can integrate domain specific knowledge about voxel noteworthiness, resulting in behavior tailored to very specific situations. This technique can be used as a guide which suggests good viewpoints for further exploration. We generate a view space partitioning, and select one representative view for each partition. Together, this set of views encapsulates the most important and distinct views of the data. Viewpoints in this set can be used as starting points for interactive exploration of the data, thus reducing the human effort in visualization. In non-interactive situations, this set can be used as a representative visualization of the dataset from all directions. We extend the viewpoint goodness measure to time-varying datasets by including the changes between adjacent time-steps.

1. Introduction

Over the past few years, data sizes have grown quite rapidly. And the trend is apt to continue at the same pace, if not faster. Even with the predicted advances in silicon technology, it is unlikely that computing power and bandwidth will catch up with the data explosion in the foreseeable future. This scenario poses many challenges to the field of visualization, which is defined as the act of a person viewing some representation(s) of the data in a bid to discover new insights from the data. Larger datasets translate to greater amount of work required from the person, and also to greater response times of the computers.

The visualization process frequently involves a hit and trial method of parameter tweaking in an effort to create better representations. This works well for smaller datasets, but for large data, the response times of the visualization system can become uncomfortably large. Many user studies have shown that there is an inverse relationship between human productivity and the response times of the systems [Shn84]. Moreover, longer waiting times are known to significantly increase the anxiety levels of users [Guy88]. For complex tasks, there is evidence that human errors increase when the response takes longer than an optimal time dependent on the tasks [Shn84][BHCL83]. While this problem can be tackled by creating more interactive systems [SBL*02], we take the approach of reducing the hit and trial tweaking the user has to do to create a desirable visualization. Automatic (or semi-automatic) methods for generating transfer functions [MAB*97][KD98] can be thought of as efforts in this direction. In this paper, we focus on helping the user with one specific component of interaction– view selection in the case of volume rendering.

In a typical volume rendering scenario, the user starts with a default viewpoint. After the first image is rendered, she changes the view to look at parts of the dataset that are occluded in the current view. This process continues till she is satisfied. The longer she has to wait for the rendering at the new viewpoint to show up, the less efficient and more frustrated she would become. The problem is exacerbated in highly non-interactive situations, and it is desirable that she quickly find a satisfactory view. This manual view selection method can be specially tricky and time consuming in the case of volume rendering of a time-varying dataset. The user tries to get a better view based on a few time steps, but it is very difficult for her to imagine how the image will change with the viewpoint for all the time steps in the sequence. Our algorithm makes the manual view selection faster by suggesting good viewpoints to the user. These viewpoints can then be used as a starting point for further exploration. Once the user finishes exploration in the neighborhood of one suggested viewpoint, she can pick another suggested viewpoint to explore.

The viewpoints suggested by our technique can also be used to improve image-based volume rendering algorithms [MSHC99][CKT01]. Frequently, IBR methods use the scene properties to create a non-uniform camera placement for the pre-rendered views based on the scene. Our formulation can be used to quantify the change between two volume-rendered views. An adaptive sampling of the view space can be generated by creating more pre-rendered samples in neighborhoods of large view changes and vice-versa. This can help the IBR system achieve better rendering quality with less storage.

We introduce a ‘goodness’ measure of viewpoints based on the information theory concept of entropy, also called average information. We proposition that good viewpoints are ones which provide higher visibilities to the more important voxels, the importance being judged by the opacities assigned by the transfer function. This interpretation leads us to the formulation of viewpoint information presented in this paper. We utilize a property of our entropy definition which indicates that when the visibilities are close to their desired values, the viewpoint information is maximized. This measure allows us to compare different viewpoints and suggest the best ones to the user. Given a desired number N of views, our algorithm can be used to find N good viewpoints over the view space. We use an entropy based similarity measure for views, which is then used to create a view space partitioning and return the best views in each partition. Together, this set of views represents most of what can be seen from all directions. For time-dependent data, we present a modification of the ‘goodness’ measure of a viewpoint by taking into account not only the static information but also the change in each time-step.

2. Related Work

The idea of comparing different views developed much before computer graphics and visualization matured. As early as 1976, Koenderink and van Doorn [KvD76][KvD79] had studied singularities in 2D projections of smooth bodies. They showed that for most views (called stable views), the topology of the projection does not change for small changes in the viewpoint. The topological changes between viewpoints can be stored in an aspect graph. Each node in the graph represents a region of stable views, and each edge represents a transition from one such region to an adjacent one. These regions form a partitioning of the view space, which is typically a sphere of a fixed radius with the object of interest at its center. The aspect graph (or its dual, the view space partition) defines the minimal number of views required to represent all the topologically different projections of the object. A lot of research has been done since the early papers, mainly in the field of computer vision, which extended the ideas to more complex objects. In the case of volume rendering, a similar topology based partitioning can not be constructed. Instead, we find a visibility based partitioning by comparing visibilities of voxels in neighboring views, and clustering together viewpoints that are similar.

Viewpoint selection has been an active topic of research in many fields. For instance, viewpoint selection solutions have been proposed for the problem of modeling a three-dimensional object from range data [WDA99] and from images [FCOL00], and also for object recognition [AF99]. However, the topic has not been well investigated in the fields of computer graphics and visualization, possibly because applications in these domains have relied heavily on human control. Recently, Vázquez et al. [VFSH01][VFSH03] have presented an entropy based technique to find good views of polygonal scenes. They define an entropy for any given view, which is derived from the projected area of the faces of the geometric models in the scene. Their motivation is to achieve a balance between the number of faces visible and their projection areas. The entropy value is maximized when all the faces project to an equal area on the screen. The viewpoint measure presented in this paper is based on the entropy function, but is designed for volumetric data. Each voxel is assigned a visual significance, and the entropy is maximized when the visibilities of the voxels approach the respective significance values. Entropy based methods have been used in a variety of problems, e.g., for calculating scene complexity for radiosity algorithms [FABS99], for object recognition [AF99] and for aiding light source placements [Gum02].

3. Viewpoint Evaluation

This essential goal of this paper is to have a computer suggest ‘good’ viewpoint(s) to the user. This naturally leads us to the question: “what is a good viewpoint?”, or, “what makes a viewpoint better than another?”. The answer will depend greatly on the viewing context and the desired outcome. For example, a photographer will choose the view which best contributes to the chosen mood and visual effect. For this paper, the context is the process of volume rendering, which being used to get visual information from the data. Hence, for our purposes, *a viewpoint is better than another if it conveys more information about the dataset*. In this section, we present a method for quantifying the information contained in a view using properties of the entropy function from information theory.

The information that is transferred from a volumetric dataset to the two-dimensional screen is governed by the optical model which is used for the projection. In this paper, we assume the popular absorption plus emission model [Max95]. The intensity Y at a pixel D is given by

$$Y(D) = Y_0 T(0) + \int_0^D g(s)T(s)ds \tag{1}$$

where, $T(s)$ is the transparency of the material between the pixel D and the point s. We will refer to $T(s)$ as the visibility of the location s. The first term in the equation represents the contribution of the background light, Y_0 being its intensity. The second term adds the contributions of all the voxels along the viewing ray passing through D. A voxel at point s has an emission factor of $g(s)$, and its effect on the pixel intensity is scaled by its visibility $T(s)$. If two voxels have the same emission factor, then the one with a higher visibility will contribute more towards the final image.

The emission factors of voxels are usually defined by the users. They set the transfer function to highlight the group of voxels they want to see, and to make the others more transparent. We use this fact to define a *noteworthiness* factor for each voxel (section 3.2), which captures, among other things, the importance of the voxel as defined by the transfer function. Based on the preceding discussion, we have the following two (not necessarily disjoint) guidelines for defining a good viewpoint:

1. A viewpoint is good if voxels with high noteworthiness factors have high visibilities.
2. A viewpoint is good if the projection of the volumetric dataset contains a high amount of information.

In the following section, we present the details of our view information function and its properties.

3.1. Entropy and View Information

Consider any information source X which outputs a random sequence of symbols taken from the alphabet set $\{a_0, a_1, \ldots, a_{J-1}\}$. Suppose the symbols occur with the probabilities $\mathbf{p} = \{p_0, p_1, \ldots, p_{J-1}\}$. Alternatively, we can think of it as the random variable X which gets the value a_j with probability p_j. The information associated with a single occurrence of a_j is defined in information theory as $I(a_j) = -\log p_j$. The logarithm can be taken with base 2 or e, and the unit of information is bits or nats respectively. In a sequence of length n, the symbol a_j will occur np_j times, and will carry $-np_j \log p_j$ units of information. Then the *average information* of the sequence, also called its entropy, is defined as

$$H(X) \equiv H(\mathbf{p}) = -\sum_{j=0}^{J-1} p_j \cdot \log_2 p_j \qquad \text{bits/symbol} \qquad (2)$$

with $0 \cdot \log_2 0$ defined as zero [Bla87]. Even though the entropy is frequently expressed as a function of the random variable X, it is actually a function of the probability distribution $\mathbf{p}$ of the variable X. We will use the following two properties of the entropy function in constructing our viewpoint evaluation measure:

1. For a given number of symbols J, the maximum entropy occurs for the distribution $\mathbf{p_{eq}}$, where $\{p_0 = p_1 = \ldots = p_{J-1} = 1/J\}$. (See figure 1, which gives an example of the entropy values for a three dimensional distribution.)

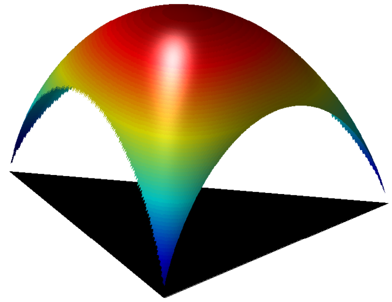


Figure 1: *Entropy Function for three dimensional probability vectors* $\mathbf{p} = \{p_0, p_1, p_2\}$*. The function is defined only over the plane* $p_0 + p_1 + p_2 = 1$*, within the triangular region specified by* $0 \le p_1, p_2, p_3 \le 1$*. The maximum occurs at the point* $p_0 = p_1 = p_2 = 1/3$*, and the value falls as we move away from that point in any direction. So, increasing the entropy has the effect of making the probabilities more uniform.*

2. Entropy is a concave function, which implies that the local maximum at $\mathbf{p_{eq}}$ is also the global maximum. It also implies that as we move away from the equal distribution $\mathbf{p_{eq}}$, along a straight line in any direction, the value of entropy decreases (or remains the same, but does not increase).

We will use probability distributions associated with views to calculate their entropy (average information). For each voxel j, we define a noteworthiness factor W_j, which indicates the visual significance of the voxel. (More details about W_j are given in section 3.2). Suppose, for a given view V, the visibility of the voxel is $v_j(V)$. We are using the term 'visibility' to denote the transparency of the material between the camera and the voxel. It is equivalent to $T(s)$ in equation (1). Then, for the view V, we define the *visual probability*, q_j, of the voxel as

$$q_j \equiv q_j(V) = \frac{1}{\sigma} \cdot \frac{v_j(V)}{W_j} \qquad \text{where, } \sigma = \sum_{j=0}^{J-1} \frac{v_j(V)}{W_j} \qquad (3)$$

where the summation is taken over all voxels in the data. Thus, for any view V, we have a visual probability distribution $\mathbf{q} \equiv \{q_0, q_1, \ldots, q_{J-1}\}$, where J is the number of voxels in the dataset. Then, we define the entropy (average information) of the view to be

$$H(V) \equiv H(\mathbf{q}) = -\sum_{j=0}^{J-1} q_j \cdot \log_2 q_j \qquad (4)$$

The view with the highest entropy is then chosen as the best view. This satisfies the two guidelines presented earlier in section 3:

1. The best view has the highest information content (averaged over all voxels).

2. The visual probability distribution of the voxels is close to the equal distribution $\{q_0 = q_1 = \ldots = q_{J-1} = 1/J\}$, which implies that the voxel visibilities are proportional to their noteworthiness.

To calculate the view entropy, we need to know the voxel visibilities and the noteworthiness factors. Visibilities can be queried through any standard volume rendering technique such as ray casting. The noteworthiness, described in the next section, is view independent, and needs to be calculated only once for a given transfer function.

3.2. Noteworthiness

The noteworthiness factor of each voxel denotes the significance of the voxel to the visualization. It should be high for voxels which are desired to be seen, and vice versa. Considering the diverse array of situations volume rendering is used in, it is practically impossible to give a single definition of noteworthiness that satisfies expectations of all users. Instead, we can rely on the user-specified transfer functions to deliver us a definition which is tailor-made for the particular situation. The opacity of a voxel, as assigned by the transfer function, is part of the emission factor $g(s)$ in equation (1), and controls the contribution of the voxel to the final image. We use opacity as one element of the noteworthiness of the voxel. Another consideration is that some voxels are more visually meaningful to the viewer than other voxels. Consider a simple example: suppose the dataset has a small region of yellow voxels and the rest of the voxels are blue. In this case, the visibility of the yellow region is more important than that of a similar number of blue voxels. When the yellow region occludes part of the blue region, Gestalt principles [PTN98] suggest that the human mind extrapolates the larger object (called ground) behind the smaller one (called figure). If, on the other hand, the yellow region is occluded, the viewer will have no idea of knowing it even exists.

Based on these observations, we construct the noteworthiness W_j of the jth voxel as follows. We assign probabilities to voxels in our dataset by constructing a histogram of the data. All the voxels are assigned to bins of the histogram according to their value, and each voxel gets a probability from the frequency of its bin. The information I_j carried by the jth voxel is then $-\log f_j$, where f_j is its probability (bin frequency). Then, W_j for the voxel is $\alpha_j I_j$, where α_j is its opacity. We ignore voxels whose opacities are zero or close to zero. These voxels are not included in the evaluation of equation (4). Domain specific knowledge can be readily included in our framework by adapting the noteworthiness. Irrespective of the method used to specify the interestingness of the voxels, maximizing the entropy serves to give better visibility to the more interesting voxels.

3.3. A Simple Example

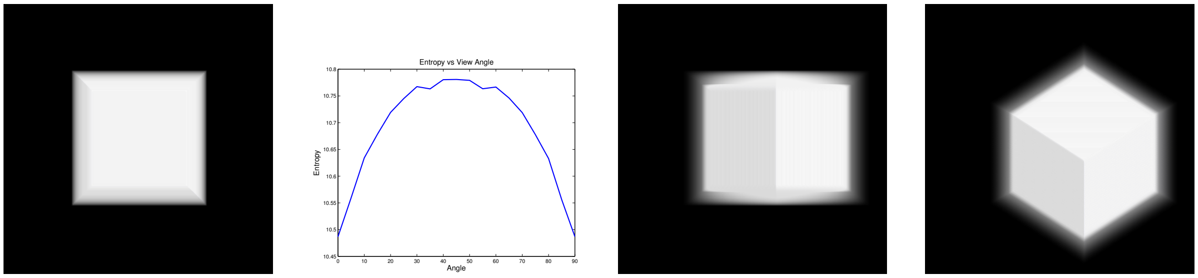
To demonstrate our concept of view information, we constructed the test dataset shown in figure 2. The voxel opacities of the cube dataset increase linearly with distance from the boundary of the cube. Figure 2(a) shows the volume rendering the dataset when the camera is looking directly at one of its faces. If we revolve the camera about the vertical axis of the dataset, (or, equivalently, rotate the dataset in the opposite direction about the vertical axis), the view entropy increases (figure 2(b)) as voxels near the side face start becoming visible. The entropy reaches a maximum when camera has moved by 45°, which is the view that shows the two faces equally (figure 2(c)). Further movement of the camera results in greater occlusion of voxels near the first face, and the entropy begins to drop again. Upon evaluating the entropies for all camera positions around the dataset, the view in figure 2(d) results in the highest entropy. Clearly, this is one of the more informative views about the cube dataset for a human observer.

3.4. Finding the Good View

The dataset is placed at the origin, and the camera is restricted to be at suitable fixed distance from the origin. This spherical set of all possible camera positions defines the view sphere, and represents all the view directions. The view space is then sampled by placing the camera at sample points on this sphere. We create a uniform triangular tessellation of the sphere and place the viewpoints at the triangle centroids.

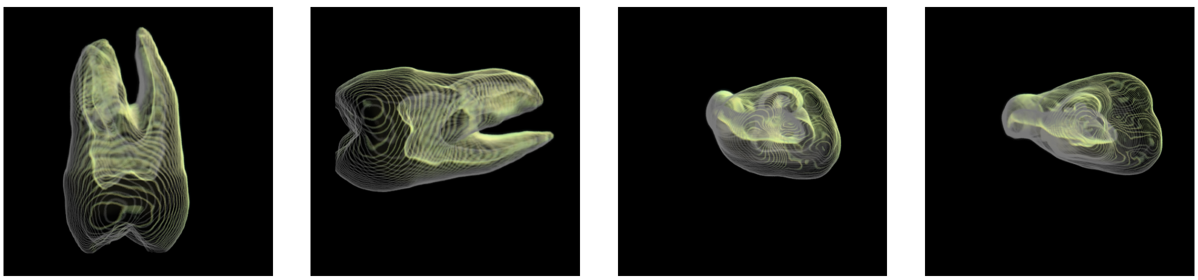
Next, the voxel visibilities are calculated for each sample view position. Our technique is not dependent on any particular volume rendering method, and both software and hardware renderers can be used as long as the voxel visibilities can be queried. For the examples presented here, we have used ray casting. The screen resolution is set high enough so that at least one ray passes through each voxel. Voxels with opacities close to zero (defined by a threshold) are classified as empty space and are not used in the evaluation of equation (4). This reduces the computational and memory requirements for the entropy and similarity calculations (section 4.1), and also raises the possibility of using empty space leaping enhancements for ray casting. Early ray termination is used, and visibilities of voxels not seen (not intersected by any ray) are set to zero. The entropy for the view direction can then be calculated by using the visibilities and the noteworthiness factor.

Figure 3 shows a $128 \times 128 \times 80$ tooth dataset rendered using ray-casting. The view sphere was sampled at 128 points, and the screen resolution used was 512×512. Figures (a) and (b) have the highest view entropy values. Figures (c) and (d) have the lowest entropy, and not surprisingly, are highly occluded views. It is notable that the viewpoints for (c) and (d) are not very far apart, and that (a) and (b) show much of the same voxels. This shows that if the user wants a few (say, N) good views from the algorithm, returning the N highest entropy views might not be the best option. Instead we can try to find a set of good views whose view samples



(a) (b) (c) (d)

Figure 2: *An illustration of the change in view entropy (equation 4) with camera position for a test dataset. Figure (a) shows the initial position of the camera. Figure (b) shows the behavior of entropy as the camera revolves around the dataset (around the vertical axis in the figure). The entropy increases and reaches a maximum for a movement of 45° (figure (c)), and then begins to decrease again. The maximum entropy for the whole view space is obtained for the view in figure (d).*



(a) (b) (c) (d)

Figure 3: *The two highest entropy views for the tooth dataset are shown in (a) and (b), and the two worst ones in (c) and (d).*

are well distributed over the view sphere. The next section presents such a solution.

4. View Space Partitioning

The goodness measure presented in the previous section can be used as a yardstick to measure the information captured by different volume rendering views and select the best view. But, for most datasets, a single view does not give enough information to the user. The user will almost certainly want to look at the dataset from another angle. Instead of a single view, it is desirable to present to the user a set of views such that, together, all the views in the set provide a complete visual description of the dataset. This can also be thought of as a solution to the best N views problem: given a positive number N, we want to find the best N views which together give the best visual representation of the dataset.

We propose to find the N views by partitioning the view sphere into N disjoint partitions, and selecting a representative view for each partition. A similar partitioning is defined by aspect graphs [KvD76][KvD79], where each node (aspect) of the graph represents a set of stable views. Each set shows the same group of features on the surface of the object. However, the aspect graph creation methods deal mostly with algebraic and polygonal models and their topology, and cannot be applied in a straightforward manner to volume rendering. Instead, we compute the partitioning by grouping similar viewpoints together.

4.1. View Similarity

To find the (dis)similarity of viewpoints, we use the visual probability distributions associated with each viewpoint (section 3.1). Popular measures for computing the dissimilarity between two distributions $\mathbf{p}$ and $\mathbf{p}'$ are the relative entropy (also known as the Kullback-Leibler (KL) distance), and its symmetric form (known as divergence) which is a true metric [Bla87]. (Please note that some texts refer to the KL distance as divergence instead.):

$$D(\mathbf{p}\|\mathbf{p}') = \sum_{j=0}^{J-1} p_j \log \frac{p_j}{p'_j} \quad (5)$$

$$D_s(\mathbf{p},\mathbf{p}') = D(\mathbf{p}\|\mathbf{p}') + D(\mathbf{p}'\|\mathbf{p}) \quad (6)$$

Although these measures have some nice properties, there are some issues with these measures that make them less than ideal. If $p'_j = 0$ and $p_j \neq 0$ for any j, then $D(\mathbf{p}\|\mathbf{p}')$ is undefined. In our case, any voxel which is fully occluded (zero visibility) will get a visual probability q_j of zero (equation (3)). If it is visible in one view but occluded in the other, we cannot evaluate equation 5 for these views. Also, $D(\mathbf{p}||\mathbf{p}')$ and $D_s(\mathbf{p},\mathbf{p}')$ do not offer any nice upper-bounds. To overcome these problems, we instead use the Jensen-Shannon divergence measure [Lin91]:

$$JS(\mathbf{p},\mathbf{p}') = JS(\mathbf{p}',\mathbf{p}) = K(\mathbf{p},\mathbf{p}') + K(\mathbf{p}',\mathbf{p}) \quad (7)$$

$$\text{where, } K(\mathbf{p},\mathbf{p}') = D(\mathbf{p}||(\frac{1}{2}\mathbf{p}+\frac{1}{2}\mathbf{p}')) \quad (8)$$

The distance between two views V_1 and V_2, with distributions $\mathbf{q}_1$ and $\mathbf{q}_2$, is then defined as $JS(\mathbf{q}_1,\mathbf{q}_2)$. This measure does not have the zero visual probability problem, since the denominator of the log term is zero iff the numerator is zero. It is also nicely bounded by $0 < JS(\mathbf{q}_1,\mathbf{q}_2) < 2$. Moreover, it can be expressed in terms of entropy [Lin91], which allows us to reuse the view information calculations given in equation (4):

$$JS(\mathbf{q}_1,\mathbf{q}_2) = 2H(\frac{1}{2}\mathbf{q}_1+\frac{1}{2}\mathbf{q}_1) - H(\mathbf{q}_1) - H(\mathbf{q}_2) \quad (9)$$

4.2. Partitioning

Once the visual probability functions (q) and their entropies (H) are calculated as described in section 3, we use the JS-divergence to find the (dis)similarities between all pairs of view samples. We then cluster the samples to create a disjoint partition of view sphere. The number of desired clusters can be specified by the user. Each partition represents a set of similar views, i.e., these views show the voxels at similar visibilities. If desired, the JS-measure can be weighted using the physical distance between the view samples to yield tight regional clusters.

The best (highest entropy) views within each partition are selected as representatives of the cluster and displayed to the user. Together, this set of images give a good visualization of the dataset from many different viewpoints. Sometimes, it might happen that the selected representatives of two neighboring partitions lie on the common boundary and next to each other. If the separation between two selected view samples is less than a threshold, we use a greedy approach and select the next best sample.

Figure 4 shows the results of a 5-way partitioning of the view space for the tooth dataset. 128 view samples were used with a JS-divergence measure. The largest partition contains 39 samples, while the smallest one has 18. The representative views from four of the partitions are shown. The view for the fifth partition is figure 3(a). Figures 3(a) and 3(b) both lie in the same partition. In fact, the top ten high entropy viewpoints fall in the same partition, illustrating the need for selecting representative views from different partitions.

5. Time Varying Data

Suggestion of good views becomes all the more useful in the case of time dependent data. The time required to compute a volume rendering animation of the dataset grows with the number of time steps. In an interactive setting, this creates a large lag between a viewpoint update and the completion of all the frames. Moreover, it takes more tries by the user to find the desired viewpoint because the data changes with time, and the user has to consider not only the current time step but also the previous and future ones. The user's job is made harder by cases where an interesting view in a few time steps turns out to be a dull view in the rest.

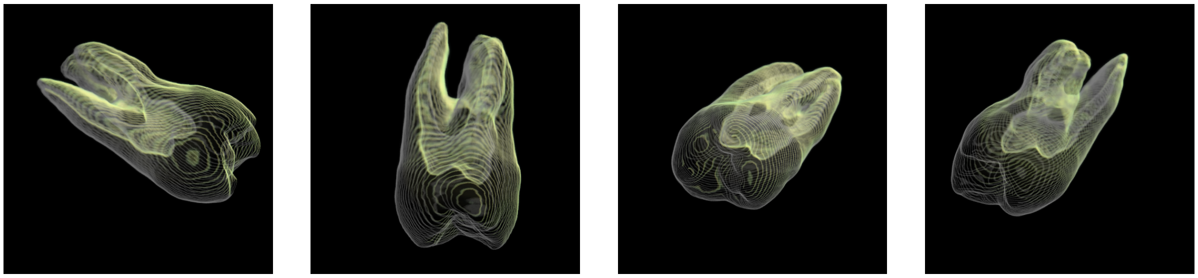
In section 3, we discussed the notion of a good view and presented a measure of view information for a volume dataset. For time-dependent data, using equation (4) separately for each time-step is not the desired solution– it can yield viewpoints in adjacent time-steps that are far from each other, thus resulting abrupt jumps of the camera during the animation. A natural solution is to constrain the camera, but it still does not guarantee the most informative viewpoint. For instance, it can result in a viewpoint which has a high information value for each individual time-step, but does not show any time-varying changes. It is contrary to what is expected from an animation– it should show both the data at each time-step, and also the changes occurring from one frame to the next. In the next section, we present an alternate version of viewpoint information tailored to capture the view information present in one time-step, taking into account the information present in the previous step.

5.1. View Information

Consider two random variables X and Y with probability distributions $\mathbf{r}$ and $\mathbf{s}$ respectively. If X and Y are related (not independent), then an observation of X gives us some information about Y. As a result, the information carried by Y, conditional on observing X, becomes $H(Y|X) \equiv H(\mathbf{s}|\mathbf{r})$. Then the information carried together by X and Y is $H(X,Y) = H(Y,X) = H(X) + H(Y|X)$, as opposed to $H(X)+H(Y)$. We will use this concept to create a modified viewpoint goodness measure for time dependent data.

Suppose there are n time-steps $\{t_1,t_2,\ldots,t_n\}$ in the dataset. For a given view V, we denote the entropy for time-step t_i as $H(V,t_i) \equiv H_V(t_i)$. The view entropy for all the time-steps together is $H_V(t_1,t_2,\ldots,t_n)$. We will assume a Markov sequence model for the data, i.e., the data in any time-step t_i is dependent on the data of the time-step t_{i-1}, but independent of older time-steps. Then the information measure for the view, for all the time-steps taken together, is given by equation (11). (10) is a standard relation [Bla87], and (11) follows from the independence assumption.

$$\begin{aligned} H(V) &= H_V(t_1,t_2,\ldots,t_n) \\ &= H(t_1) + H(t_2|t_1) + \ldots + H(t_n|t_1,\ldots,t_{n-1}) \quad (10) \\ &= H(t_1) + H(t_2|t_1) + \ldots + H(t_n|t_{n-1}) \quad (11) \end{aligned}$$



(a) (b) (c) (d)

Figure 4: *Representative views for a 5-way partitioning of the view-sphere for the tooth dataset. The view for the fifth partition is figure 3(a).*

The conditional entropies will be defined following the same principles outlined in section 3. We consider a view to be good when the visibilities of the voxels are in proportion to their noteworthiness. But in the time-varying case, the significance of a voxel is derived not only from its opacity, but also from the change in its opacity from the previous time-step. For the time-step t_i, we then define the noteworthiness factor of the jth voxel as $W_j(t_i|t_{i-1})$ =

$$\{k \cdot |\alpha_j(t_i) - \alpha_j(t_{i-1})| + (1-k) \cdot \alpha_j(t_i)\} \cdot I_j(t_i) \quad (12)$$

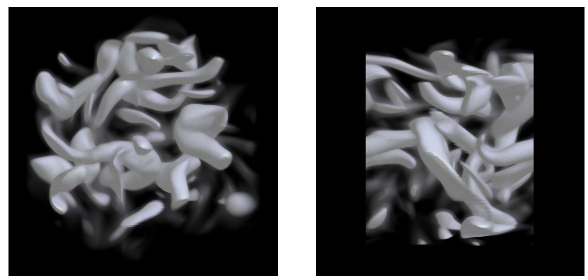
where, $0 < k < 1$ is used to weight the effects of voxel opacities and the change in their opacities. A high value of k will highlight the changes in the dataset. Suppose the visibility of the voxel for the view V is $v_j(V, t_i)$. Then, the conditional visual probability, $q_j(t_i|t_{i-1})$, of the voxel is

$$q_j(t_i|t_{i-1}) \equiv q_j(V, t_i|t_{i-1}) = \frac{1}{\sigma} \cdot \frac{v_j(V)}{W_j(t_i|t_{i-1})} \quad (13)$$

where, σ is the normalizing factor as in equation (3). The entropy of the view V is then calculated using equations (11) and (13). Voxels with both low opacities and small changes (as defined by thresholds) are ignored for these calculations.

6. Results and Discussion

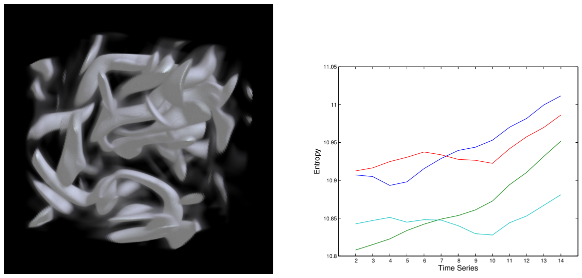
We have implemented our technique using a traditional ray casting algorithm with early ray termination. 128 sample views were used for each dataset. The camera positions were obtained by a regular triangular tessellation of a sphere with the dataset place at its center. View selection results for the $128 \times 128 \times 80$ tooth dataset have been shown in figure 3. Figure 4 shows the results of a 5-way view space partitioning for the dataset using the *JS* divergence measure. The partitioning helps to avoid selection of a set of good views which happen to be similar to each other. Even though we have not considered the physical distance between the viewpoints during partitioning, it forces the selected viewpoints to be well distributed over the view sphere. Figure 5 shows view evaluation results for a 128-cube vortex dataset. Both high and low quality views are shown for comparison. This method can be extended to large datasets by computing the view entropies at a lower resolution.



(a) (b)

Figure 5: *View Evaluation results for a 128-cube vortex dataset. Figure (a) shows the recommended view with a high entropy value, (b) shows a bad view for comparison.*

For time-varying data, we used the view information measure presented in section 5. A sequence of 14 time-steps of the 128-cube vortex data was used. The entropy for each view was summed over all the time-steps, as given by equation (11). The conditional entropy for each time-step was calculated with $k = 0.9$ in equation (12). A high value of k gives more weight to the voxels which are changing their values with time compared to high opacity voxels which remain relatively unaltered. Figure 6(a) shows the view with the best cumulative entropy for the time-series. Although the summed entropy gives a good overall view for the whole time-series, there might be other views which are better for particular segments of the time-series. Figure 6(b) plots the conditional entropies ($H(t_n|t_{n-1})$) for four selected views of the vortex dataset. The best overall view (figure 6(a)), which is represented by the blue curve (highest curve on the right boundary), is not the best choice for the first half of the series. For long time sequences, it might be beneficial to consider different good views for different segments of time.



(a) (b)

Figure 6: *View Evaluation for time-varying dataset. (a) The best overall view for 14 time-steps. (b) The conditional entropies of four selected views for each of the 14 time-steps. The view in (a) is represented by the blue plot (highest curve, top-right corner).*

7. Conclusion and Future Work

We have presented a measure for finding the goodness of a view for volume rendering. We have used the properties of the entropy function to satisfy the intuition that good views show the noteworthy voxels more prominently. The user sets the noteworthiness of the voxels by specifying the transfer function. Our algorithm can be used both as an aid for human interaction in not-so-interactive systems, and also as an oracle to present multiple good views in less interactive contexts. Furthermore, view sampling methods such as IBR can use the sample similarity information to create a better distribution of samples.

References

[AF99] ARBEL T., FERRIE F.: Viewpoint selection by navigation through entropy maps. In *Proc. of the 7th IEEE International Conf. on Computer Vision (ICCV-99)* (1999), vol. I, IEEE, pp. 248–254. 2

[BHCL83] BARBER R. E., HENRY C. LUCAS J.: System response time operator productivity, and job satisfaction. *Comm. of the ACM 26*, 11 (1983), 972–986. 1

[Bla87] BLAHUT R. E.: *Principles and practice of information theory*. Addison-Wesley Publ. Co., 1987. 3, 5, 6

[CKT01] CHEN B., KAUFMAN A., TANG Q.: Image-based rendering of surfaces from volume data. In *Proc. of IEEE Workshop on Volume Graphics* (2001), IEEE. 2

[FABS99] FEIXAS M., ACEBO E., BEKAERT P., SBERT M.: An information theory framework for the analysis of scene complexity. *Computer Graphics Forum (Eurographics'99 Proc.) 18*, 3 (1999), 95–106. 2

[FCOL00] FLEISHMAN S., COHEN-OR D., LISCHINSKI D.: Automatic camera placement for image-based modeling. *Computer Graphics Forum 19*, 2 (2000), 101–110. 2

[Gum02] GUMHOLD S.: Maximum entropy light source placement. In *Proc. of IEEE Visualization '02* (2002), pp. 215–222. 2

[Guy88] GUYNES J. L.: Impact of system response time on state anxiety. *Comm. of the ACM 31*, 3 (1988), 342–347. 1

[KD98] KINDLMANN G., DURKIN J. W.: Semi-automatic generation of transfer functions for direct volume rendering. In *Proc. of IEEE Symposium on Volume Visualization '98* (1998), pp. 79–86. 1

[KvD76] KOENDERINK J. J., VAN DOORN A. J.: The singularities of the visual mapping. *Biological Cybernetics 24* (1976), 51–59. 2, 5

[KvD79] KOENDERINK J. J., VAN DOORN A. J.: The internal representation of solid shape with respect to vision. *Biological Cybernetics 32* (1979), 211–216. 2, 5

[Lin91] LIN J.: Divergence measures based on the shannon entropy. *IEEE Trans. on Information Theory 37*, 1 (January 1991), 145–151. 6

[MAB*97] MARKS J., ANDALMAN B., BEARDSLEY P., FREEMAN W., GIBSON S., HODGINS J., KANG T., MIRTICH B., PFISTER H., RUML W., RYALL K., SEIMS J., SHIEBER S.: Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proc. of SIGGRAPH 1997* (1997), ACM. 1

[Max95] MAX N.: Optical models for direct volume rendering. *IEEE Trans. on Visualization and Computer Graphics 1*, 2 (June 1995), 99–108. 2

[MSHC99] MUELLER K., SHAREEF N., HUANG J., CRAWFIS R.: Ibr-assisted volume rendering. In *Late Breaking Hot Topics, IEEE Visualization Conf.* IEEE, 1999. 2

[PTN98] PESSOA L., THOMPSON E., NOË A.: Finding out about filling-in: A guide to perceptual completion for visual science and the philosophy of perception. *Behavioral and Brain Sciences 21*, 6 (1998), 723–748. 4

[SBL*02] SILVA C., BARTZ D., LINDSTROM P., KLOSOWSKI J., SHROEDER W.: High-performance visualization of large and complex scientific datasets. Tutorial M9, Super Computing 2002, Baltimore, 2002. 1

[Shn84] SHNEIDERMAN B.: Response time and display rate in human performance with computers. *ACM Computing Surveys 16*, 3 (1984), 265–285. 1

[VFSH01] VÁZQUEZ P. P., FEIXAS M., SBERT M., HEIDRICH W.: Viewpoint selection using viewpoint entropy. In *Proc. of Vision, Modelling, and Visualization '01* (2001), pp. 273–280. 2

[VFSH03] VÁZQUEZ P. P., FEIXAS M., SBERT M., HEIDRICH W.: Automatic view selection using viewpoint entropy and its application to image-based modeling. *Computer Graphics Forum 22*, 4 (2003), 689–700. 2

[WDA99] WONG L., DUMONT C., ABIDI M.: Next best view system in a 3-d modeling task. In *Proc. of International Symposium on Computational Intelligence in Robotics and Automation (CIRA)* (1999), pp. 306–311. 2