Explorable Images for Visualizing Volume Data

Anna Tikhonova[*] Carlos D. Correa[†] Kwan-Liu Ma[‡]

Visualization & Interface Design Innovation (VIDi) Research Group
University of California, Davis

Abstract

We present a technique which automatically converts a small number of single-view volume rendered images of the same 3D data set into a compact representation of that data set. This representation is a multi-layered image, or an explorable image, which enables interactive exploration of volume data in transfer function space without accessing the original data. We achieve this by automatically extracting layers depicted in composited images. The layers can then be recombined in different ways to simulate opacity changes and recoloring of individual features. Our results demonstrate that explorable images are especially useful when the volume data is too large for interactive exploration, takes too long to render due to the underlying mesh structure or desired shading effect, or if the original volume data is not available. Explorable images can offer real-time image-based interaction as a preview mechanism for remote visualization or visualization of large volume data on low-end hardware, within a mobile device, or a Web browser.

Index Terms: I.3.3 [Computer Graphics]: Picture/Image Generation—display algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—color, shading, shadowing, and texture

1 Introduction

Despite the proliferation of 3D applications and the increasing rendering capabilities of contemporary graphics hardware, image-based applications are more popular and familiar to users than 3D rendering software. In volume visualization, image-based rendering methods are necessary to enable interactive exploration of data sets that are too large to store, to render, to explore interactively using 3D software, or when a user does not have access to the original volume data. For example, large volumes usually do not fit into GPU memory and the CPU to GPU bandwidth often inhibits interactive exploration of large data sets. One solution is to use dedicated clusters for image generation and let users interact with images from remote, often low-end clients. In other cases, users may only have a collection of 2D renderings of the data, but not the original volume. In these scenarios, one has to resort to image-based operations in order to manipulate different properties of the data. However, the lack of explicit 3D information prevents the user from changing the color and opacity of the data depicted in the images.

In this paper, we present a technique for manipulating some properties of volume rendered data using 2D images as input. We introduce the notion of an *explorable image*, which is a new type of image that lets users manipulate rendering properties such as color and opacity in image space without re-rendering and without the need for high-end graphics hardware. Although view-dependent, explorable images allow users to navigate the transfer function space of volume data without accessing the original 3D data. Our approach is based on the observation that volume rendered images, despite being 2D projections of 3D data, contain enough information about the different structures in the data and provide cues such as depth, occlusion, and shape. Depending on the number of composited images available, we can decompose the data into different layers corresponding to different structures in the data and synthesize new images with different opacity and color mappings.

To achieve this, we decompose the features comprising a volume given a small collection of composited images as input. These input images depict arbitrary combinations of the different structures in the data, and can be provided (when only the images are available), or generated automatically from a large data set. Our algorithm extracts the contribution of each structure to the input images using Bayesian estimation. The result of this process is a set of extracted layers, corresponding, ideally, to each feature depicted in the input images. Once the contribution of each layer is known, we can recomposite the layers into new renderings, allowing the user to manipulate the color and opacity of each layer. This process is illustrated in Figure 1(a). The automatic multi-layer extraction technique and the concept of an explorable image are the main contributions of this paper. Unlike previous image-based compositing techniques, we extract layers from composited images that include multiplicative attenuation, typical of semi-transparent or translucent materials. This type of attenuation gives us clues about the relative depth and volume of each structure. We exploit this property to synthesize images with different materials and create volumetric cutaways that reveal the internal structure of complex data sets while preserving their context.

There are a number of technical challenges in obtaining explorable images. As a general rule of thumb, we allow the number of input images to be much smaller than the number of layers to extract. This is important for making explorable images as compact as possible. However, this makes the problem ill-defined. We present an optimization-based approach that extracts layers from a small number of input images with additional constraints, such as spatial consistency and opacity distribution, to help arrive at plausible solutions. In other cases, 3D data is available but it is too large to distribute or to store on low-end devices. A more efficient solution is to instead distribute a series of composited images. This requires the generation of a transfer function that is appropriate for our multi-layer extraction process. We describe an opacity mapping generation algorithm that highlights combinations of features such that the number of composited images is much lower than the space requirement for storing individual layers.

We demonstrate that one of the main advantages of explorable images is enabling users to dynamically filter the information presented in resulting rendered images. For example, a user may decrease the opacity of an outer layer to see the inner layers more clearly. We believe that the interactions we provide are useful in helping users understand the spatial relationships between layers and the overall shape of the features of interest. Because of the ability to manipulate the opacity and color properties in image space, explorable images can be far more clear and informative than their

[*] e-mail: tikhonov@cs.ucdavis.edu

[†] e-mail: correac@cs.ucdavis.edu

[‡] e-mail: ma@cs.ucdavis.edu

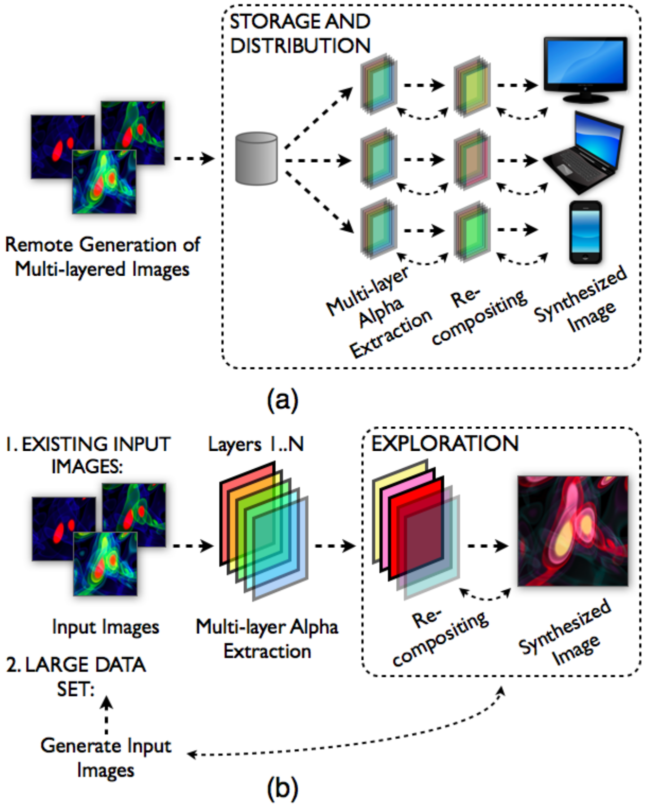


Figure 1: (a) Overview of the two possible usage scenarios. In the first scenario, a user has a set of static input images and in the second, the input images are generated from volume data. Then, multi-layer alpha extraction is performed, followed by interactive exploration. This process can be repeated. (b) Overall process of the storage and distribution of explorable images. The compact image-based representation of a volume is stored and distributed for future exploration. Multi-layer alpha extraction allows to generate layers for synthesizing new images. Users can influence the layer extraction process and to interactively modify the properties of each layer.

static counterparts.

2 Related Work

Layer and Image Based Rendering. Due to the ubiquity and popularity of 2D image processing, image-based rendering has been a popular alternative for view synthesis, 3D geometry reconstruction, re-lighting and multi-layer rendering [12]. Shade et al. propose the use of *layered depth images (LDI)*, whose pixels contain a list of color and depth values [26]. Depth values allow to display the appropriate parallax induced by camera motion. Multi-layered representations have been popularized in commercial rendering software to simulate complex materials on synthetic objects, such as skin and translucency [6, 7]. In volume rendering, images have been used to synthesize new views of volume data [4], to cache results in a remote setting [15], to provide an efficient exploration space of color and lighting parameters [9, 18] or to generate transfer functions [23, 34]. The work in this paper is related to the design of transfer functions in image space. He and al. [9] and Marks et al. [18] allow users to find good transfer functions based on the image results of previous interactions. Ropinski et al. present a sketch-based design where users can combine different layers, obtained from individual component functions, to define a single transfer function [23]. Wu and Qu present a system that generates a transfer function that approximates the visual result of combining different image layers from a volume data set [34]. Rautek et al. use the multi-layer metaphor to combine different rendering styles according to a semantic user specification [21]. In this paper, we address a rather complementary challenge, that of extracting visible layers from existing volume-rendered images, and the subsequent recomposition for new transfer functions. Although the synthesis of new images from layers follows the same ideas proposed before, our unique contribution lies in layer extraction and the fact that we can provide new transfer functions from just a small set of input images.

One of the applications of our approach is recolorization of volume rendered images. Color design is an important part of visualization. Effective color maps help guide users' attention [31], convey motion [32], or improve perception of shapes [22, 14]. It is not in the scope of our paper to apply any particular coloring scheme to the synthesized images, but rather to provide the means to explore the color space of volume rendered images. In that sense, we believe our approach is a powerful complementary technique.

Layer Extraction. The issue of layer extraction has surfaced in the film and video production field as a way to create digital effects. The simplest approach, called *alpha matting*, is used for extracting a single foreground layer from the background of a photograph or a synthetic image. Earlier techniques required either a constant background color, as pioneered by Vlahos (blue-screen matting), a combination of two constant-color backdrops [27], or a known non-constant background [20].

The more general problem of extracting mattes from arbitrary photographs or video streams is known as *natural image matting*. Some of the most successful techniques in this category include Knockout, described in patents by Berman et al. [1, 2], and the technique of Ruzon and Tomasi [24]. Maximum likelihood estimation has been extensively used to solve the matting problem [5, 3, 30]. Other techniques to improve alpha matting using Principal Component Analysis (PCA) [10], gradient fields [28], or spectral decompositions [16]. These techniques, however, are more effective for natural images. In this paper, we generalize the notion of alpha extraction for multiple overlapping layers, useful for decomposing volume rendered images. Alpha matting has been extended for multi-layered images, either to recover layers from photographs of moving objects [29] or for multi-view object reconstruction [33]. In other cases, multiple layers are used to recover lighting effects. Environment matting [36], for example, extracts additional layers corresponding to the refraction and reflection introduced by foreground objects. Chen et al [3] use segmentation to divide the image into several foreground objects. Unlike previous attempts where foreground objects and layers are opaque, we consider the case of volume rendered images, which often contain multiple semi-transparent layers composited together. To the best of our knowledge, this is the first attempt to extract and recomposite different alpha layers directly from volume rendered images.

A more general approach for editing semi-transparent layers is to reconstruct a volume from images, then recomposite them with new material properties. Yamazaki et al. [35] formulate the process as *inverse volume rendering*, where photos of the same object taken against two different constant-color backdrops are taken as input. Seitz and Dyer use voxel coloring to reconstruct opaque objects [25]. The reconstruction of semi-transparent objects, such as fire and smoke, demands a different approach, such as tomographic reconstruction [8, 11]. A similar approach was proposed for reconstructing astronomical objects [17]. Similar to these approaches, we use inverse compositing to combine alpha mattes of individual layers. However, since we are interested in changing the material properties and light interactions of layers, not synthesize new views, we can extract the layers directly from input images without reconstructing a volume. This makes our approach particularly effective for low-powered machines and interactive exploration.

3 Explorable Images

An explorable image is a collection of single-view layered images of a data set. This collection has the following characteristics:

- Each layer is represented in at least one image.
- Each image may contain multiple layers.
- If m is the number of images and n is the number of layers, we allow $m < n$. Generally, we want m to be as small as possible.

Given an explorable image of a data set, it is possible to extract individual layers and to generate many new images with any combination of layers. It is also possible to modify the transparency and color of each layer. The methods for creating and interacting with explorable images are entirely image-based. Neither the original 3D data nor re-rendering are required.

The process of creating and interacting with an explorable image includes the following:

Generation of input images: In the scenario where input images are not provided, this pre-processing rendering step generates a set of multi-layered images from volume data. At the rendering time, the rendering parameters for each layer are stored as additional information. The first step in the input image creation process is the automatic generation of a transfer function. Users can control this process by supplying domain knowledge, such as the range of values of interest. Users can also influence the number of generated images by selecting the number of layers they wish to explore. The number of images can range from a single image for the entire data set to an individual image per layer. There is a trade-off between the storage space or bandwidth requirements and the ability to reproduce all possible interactions between layers. Our method can generate explorable images from any set of input images. Even in the case when only a single image of a data set is available, we can still create an explorable image. Once the input images have been generated, this compact representation of a volume is ready for storage and distribution for future exploration.

Automatic layer extraction: In this step, we extract the layers using *multi-layered alpha extraction* - a method similar to alpha matting techniques.

Interaction/Exploration: The decomposition of 2D images into individual layers allows users to explore the data in image space. The available operations include recoloring, recompositing, and rendering of ghosted views and cutaways. Since these operations are image-based, they are inexpensive and can be performed on low-end hardware, mobile devices, and within browsers.

These stages are illustrated in Figure 1(a) for two scenarios. In the first scenario, the input is a small set of images. After automatic multi-layer extraction, users can explore the dataset by changing opacity, color, or adding volumetric cutaways. In the second scenario, we generate input images from a large data set using the method described below. The extracted layers act as an image cache that can be used to explore the data in transfer function space without re-rendering. Since our method is view-dependent, the process must be repeated when the view changes. Figure 1(b) describes the overall process of storage and distribution of explorable images. The set of multi-layered images (showing arbitrary combinations of layers) is stored and can be distributed for future exploration. Once the user chooses to explore the data set, automatic multi-layer alpha extraction is performed. The layers can be subsequently used for recompositing and synthesizing new images.

4 Generation of Input Images

Our approach does not assume anything particular about the input images except a rough estimate of each layer's color to make them separable. In our experiments, however, we generate images from 3D data pseudo-automatically. Since we allow users to change the color and opacity of individual layers, the initial transfer function is not optimized for visualization, but for layer extraction. Our technique can be used even if a user wants to manually generate input images. The only requirement is that the following additional information is stored in addition to the images: order of layers, approximate color and opacity of each layer.

In our work, we employ the transfer function generation method developed by Kindlmann and Durkin [13]. Automatic transfer function generation is not the focus of this paper and other transfer function generation methods may be considered. Refer to [19] for an overview of different methods. The technique of Kindlmann and Durkin is designed for a class of scalar volume data where the regions of interest are boundaries between different materials. The transfer function that amplifies such boundaries is generated by analyzing the relationship between the following three quantities: the data value and its first and second derivatives along the gradient direction. In our system, user has an option of controlling the amount of boundary blurring; otherwise, it is set to a default value. The red lines in the subsequent images are examples of the transfer functions generated using this technique.

We do not use the result of this method directly, but compute the peaks in the generated transfer function to obtain iso-values describing material boundaries. Our system then selects a sampling of these peaks, such that the peaks are as equally spaced as possible. Our transfer function then consists of a series of Gaussian curves with centers at the selected peaks. The user can control the width, height, and the number of peaks, which in turn controls the number of generated input images. The user can also supply domain knowledge information to the system by specifying the range of values of interest. This additional "hint" is used to control the opacity of the series of Gaussians. The green lines in the subsequent images are examples of such "hints".

Once the transfer function is generated, we can render the input images. We devised a simple scheme for generating combinations of layers. The first image contains only two layers: the innermost and the outermost layer. The second image contains these two layers along with a layer in between. The concept is that if we have information about the two bounding layers, we can reconstruct the data in between. Additional images are generated in the same manner. The heuristic tries to maximize the amount of information we can fit in a small set of images. In general, we are able to represent n layers with approximately $\log n$ images.

5 Automatic Layer Extraction

The light transport equation for a semi-transparent volume defines the color intensity at any pixel as an exponential attenuation of the extinction parameters of each point along a viewing ray:

$$C = \int_0^D c(s) e^{-\int_s^D \tau(t)dt} ds \qquad (1)$$

where $c(s)$ is the radiance coming from light sources and reflected on the surface at a sample s along a viewing ray, and the exponential is the accumulated opacity between the sample and the eye position ($s = D$). Sampling this integral at discrete points results in the well known *over* compositing operator:

$$C = \sum_{i=1}^{n} \alpha_i c_i \prod_{k=1}^{i-1} (1 - \alpha_k), \tag{2}$$

where a_i is the opacity (or extinction) of sample i. However, estimating the extinction or alpha coefficients for each sample point in the object is not possible in the general case for arbitrary opacities. For this reason, we consider the case when the image is the result of compositing several layers, each of which can be described with an homogeneous material. With this new assumption, the light transport equation becomes:

$$C = \sum_{i=1}^{L} A_i c_i \prod_{k=1}^{i-1} (1 - A_k), \tag{3}$$

where c_i is material color of a layer and $A_i = (1 - a_i)^d$ represents the approximate accumulated alpha value of a pixel in layer i along a viewing ray segment of length d. Unlike Equation 2, this compositing occurs over a small number of layers, denoted as L, instead of a large number of samples. The problem of generating an explorable image then becomes a problem of extracting the alpha values A_k from a set of composited images.

5.1 Alpha Estimation

As we mentioned previously, maximum likelihood methods have been extensively used for alpha matting. Chuang et al. [5] introduce the Bayesian approach to digital matting, where the maximum-likelihood criterion is used to estimate the optimal alpha values for each pair of foreground and background clusters. Chen et al. [3] use Bayesian estimation to solve the matting problem for grayscale images. Wang et al. [30] extend the segmentation based approach to situations in which a trimap is difficult to create. They use *belief propagation* to solve the MAP estimation problem, where each pixel's alpha value is estimated using a small user-selected sample of foreground and background colors. Szeliski et al. [29] employ constrained least squares to recover layer images from multiple composited images. Wexler et al. [33] use Bayesian estimation for multiview object reconstruction.

In our work, we are given a set of input images, each of which contains an arbitrary combination of layers. Each pixel in an input image can be represented using Equation 3 above. We formulate the problem of computing alpha mattes for each layer using the Bayesian framework and solve it using the *maximum a posteriori* (MAP) method.

Given a set of m images with observations $\hat{C} = \{C_1, \ldots, C_m\}$ and a set of n layer material colors $\hat{c} = \{c_1, \ldots, c_n\}$, we use Bayesian estimation to recover a set of n layer mattes $\hat{A} = \{A_1, \ldots, A_n\}$. Using the Bayesian framework, we express the problem as the maximization over a sum of log-likelihoods:

$$\begin{aligned} & \underset{\hat{A}}{arg\,max} P\left(\hat{A}|\hat{C},\hat{c}\right) \qquad (4) \\ &= \underset{\hat{A}}{arg\,max} P\left(\hat{C},\hat{c}|\hat{A}\right) \prod_{i=1}^{n} P\left(A_i\right) / \prod_{j=1}^{m} P\left(C_j\right) \\ &= \underset{\hat{A}}{arg\,max} \left(L\left(\hat{C},\hat{c}|\hat{A}\right) + \sum_{i=1}^{n} L\left(A_i\right) \right) \end{aligned}$$

where $L(\cdot)$ is the log-likelihood function, i.e. the log of probability $L(\cdot) = log\left(P(\cdot)\right)$. The term $\prod_{j=1}^{m} P\left(C_j\right)$ is dropped, because it is constant with respect to the optimization parameters A_i. The problem is now reduced to defining log likelihoods $L\left(\hat{C},\hat{c}|\hat{A}\right)$ and $L\left(A_1\right), \ldots, L\left(A_n\right)$. The first term is modeled by measuring the difference between the observed intensity C_j and the intensity that would be predicted by the estimated $\hat{A}$:

$$L\left(\hat{C},\hat{c}|\hat{A}\right) = \sum_{j=1}^{m} \| \left(\sum_{i=1}^{n} A_i c_i \prod_{k=1}^{i-1} (1 - A_k) \right) - C_j \|^2 / 2\sigma_{C_j}^2. \tag{5}$$

This log-likelihood models the sum of errors in the measurement of all C_j. Each term in the sum corresponds to a Gaussian probability distribution with mean $\sum_{i=1}^{n} A_i c_i \prod_{k=1}^{i-1} (1 - A_k)$ and standard deviation σ_{C_j}. Unlike [5, 3], we do not assume σ_{C_j} to be constant. We compute the values σ_{C_j} by fitting the intensity distribution of each input image to a Gaussian curve.

In general, estimating the layers from a set of images is an ill-defined problem, since it may be the case, and is often desired, to have fewer images than layers. For this reason, we introduce additional information in the form of priors to the likelihood models. The first prior attempts to fit a probability distribution to the alphas, while the second prior assumes spatial consistency of the alphas.

5.2 Alpha Distribution Priors

In our system, we constrain each A_i to be in the range $[0, 1]$ and we assume them to be modeled as a Gaussian distribution, similar to the approach by Chen et al. [3].

Wexler et al. [33] found that for natural images containing a single foreground, the alpha matte can be modeled as a beta distribution. This distribution is used as an alpha prior to improve the results of the optimization solver. Our technique is designed to deal primarily with images containing semi-transparent features. Through experimentation, we observed that alpha distribution priors for a semi-transparent feature are best modeled by a Gaussian distribution, with mean $\bar{A}$ and standard deviation σ_{A_i}.

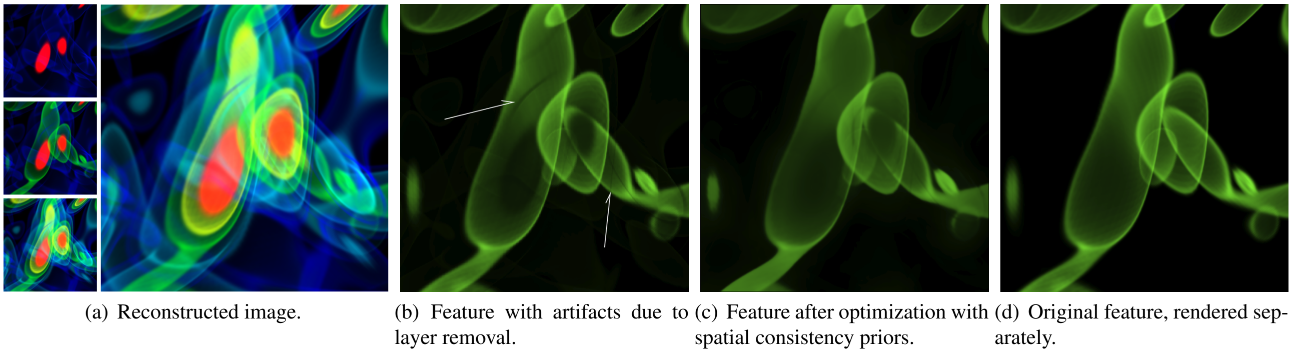
$$L(A_i) = \|\bar{A} - A_i\|^2 / 2\sigma_{A_i}^2, \tag{6}$$

This likelihood is weighted by the user to give a different relative importance in relation to the color likelihood component in Equation 5.

Unlike the approach in [3], where the distribution parameters are given or empirically assigned by the user, we estimate them from the available images. We first compute "initial guesses" of each layer's alpha matte using Equation 2 with layer material color and image background color. This is equivalent to obtaining a "pseudo-matte" for that layer given its expected color. Then we compute the Gaussian distribution parameters, $\bar{A}$ and σ_{A_i}, by fitting the intensity distribution of each "initial guess" to a Gaussian curve. This component makes sure that the extracted alphas follow the expected distribution found in the original images.

5.3 Spatial Consistency

Another assumption is that the alpha layers exhibit spatial consistency. Since the layer extraction problem may be ill-defined, it is sometimes the case that the contribution of two or more layers to the final image is ambiguous. Therefore, since in the general case we do not have all the individual layers, removing a single layer may introduce darker regions (due to overlap) for which we lack sufficient information. Spatial consistency assumes that there is a strong correlation between nearby image pixels and that the neighboring pixels with similar colors should have similar alpha values. Therefore, high frequency artifacts introduced by the removal of a layer or by the lack of information can be solved with this component.


(a) Reconstructed image.

(b) Feature with artifacts due to layer removal.

(c) Feature after optimization with spatial consistency priors.

(d) Original feature, rendered separately.

Figure 2: Spatial consistency priors. (a) Input images on the left allow us to decompose a rendering of a turbulent flow (vorticity). (b) With the extracted layer, we can synthesize a new image of a single feature (green layer). However, with the lack of complete information, some artifacts appear where occluding layers used to be present. (c) Spatial consistency helps alleviate these problems. Compare to the ground-truth image, obtained by re-rendering the individual layer (d).

We begin by using the optimization to obtain first estimates for each layer. Once we have the initial results, we blur each estimated layer and compute its gradient map. Then, we perform optimization again with the spatial consistency priors. The spatial consistency priors are incorporated into the maximum likelihood model as the squared differences of the color of a pixel and its neighbors:

$$\sum_{y \in N(x)} W(x,y)(A(x-y)-A(x))^2 \tag{7}$$

where $N(x)$ is the set of neighbors of x, and $W(x,y)$ is a weighing factor that prevents smoothing across edges. Similar to [33] and non-linear diffusion operators, this weighing factor is defined as a function of the gradient between neighboring points:

$$W(x,y) = exp\left(-\frac{((g(x,y)^{\top}(x-y))^2}{(x-y)^{\top}(x-y)}\right) \tag{8}$$

for two neighboring pixels x and y. The effect of adding the optional spatial consistency prior to the optimization system is shown in Figure 2, for images obtained from a turbulent flow (vorticity) simulation. As input, we have 3 images with different combinations of 5 layers (shown as thumbnails on the left). We can estimate each of these layers and recomposite the image as shown in Figure 2(a). When we isolate a single layer, as shown in Figure 2(b), some artifacts appear due to the limited information about some layers (since we have fewer images than the total number of layers). The spatial consistency component helps resolve some of these artifacts without blurring the stronger edges that define the feature in this layer (Figure 2(c)). Compare to the ground-truth image in Figure 2(d), obtained by rendering the isolated feature. As with other smoothing operators, there is a trade-off between the smoothness of each separate layer and the quality of the original layer details.

6 Exploration Space

The goal of extracting layers from input images is to give users the ability to modify the material properties of a data set and synthesize new images that highlight different parts of the data, using recolorization and operations on the opacities.

Our system allows users to colorize any layer in the explorable image. This is achieved by simply changing the color component of the extracted layer into the desired new color. Figure 3 shows an explorable image of a turbulent flow (vorticity) simulation. The explorable image lets a user change the color of any of the 7 layers.

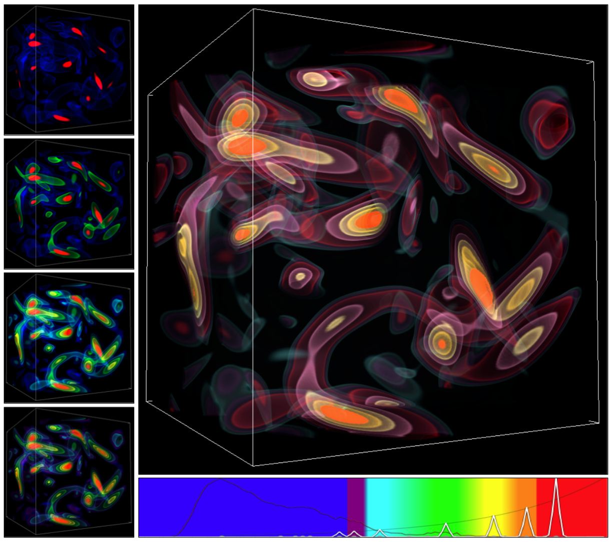


Figure 3: Colorization example. An explorable image with 7 layers is recomposed from 4 input images. The colors of all features are modified. The data set is a turbulent flow (vorticity) simulation.

We provide only 4 images of the volume data as input and the system is still able to recover most of the layers, despite the fact that some of them are not clearly isolated in the input images.

Figures 4, 5, and 6 show examples using a supernova (entropy) simulation data set. One of the challenges is to visualize the turbulent structures near the core of the supernova to understand what triggers an explosion. In Figure 5, we show the 7 layers extracted from only 3 input images and compare them to the ground-truth, individually volume rendered features. We demonstrate that the extraction is accurate, except for the artifacts due to occlusion by opaque features. After the layers are extracted, users can apply different color maps, change opacities of features, and apply volumetric cutaways to reveal the partially occluded turbulent structures.

Since each layer contains opacity information, operations on the alpha component of layers enable us to synthesize new images where we can selectively highlight parts of interest. For example, applying a 2D mask to the opacity component of a layer helps us create cutaways and ghosted views. Figure 4 shows a radial feathered cutaway mask applied to 3 outer layers (blue, cyan, green) of the supernova (entropy) data set, revealing a partially occluded structure

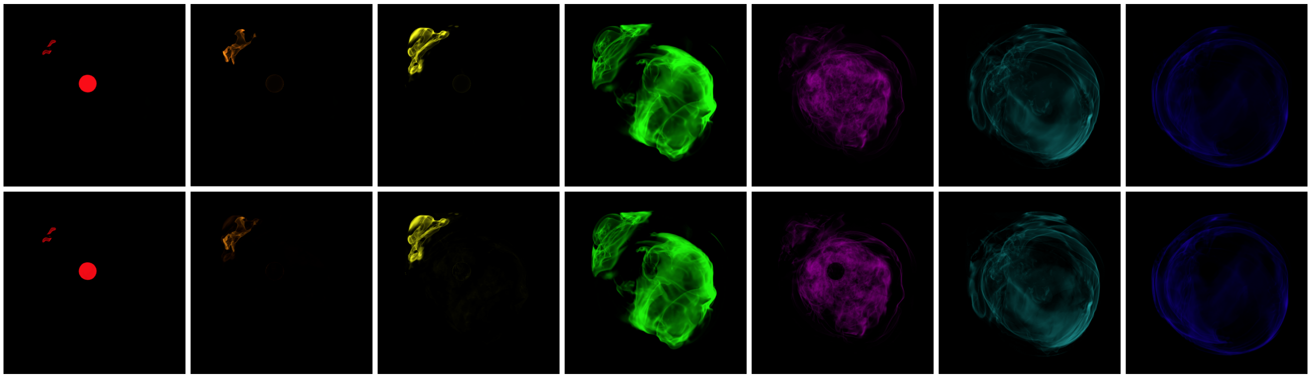


Figure 5: Comparison between the features extracted using our technique (bottom) and the ground-truth, individually volume rendered features (top). The extraction is accurate, except for the structures occluded by opaque features.

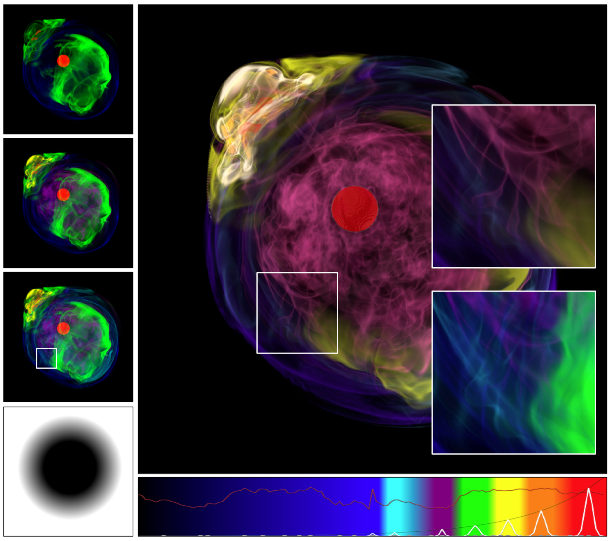


Figure 4: The radial feathered alpha mask is applied to 3 outer layers (purple, cyan, green) to reveal the inner feature (purple). Insets provide a close up view of one of the areas where the partially occluded structure is revealed. The extraction of all 7 layers is performed using only 3 input images.

(purple). The insets highlight one area where recolorization and application of an alpha mask reveal a previously occluded feature.

Unlike traditional layer-based rendering, we estimate the opacity based on a volumetric light transport model, which assumes that light attenuates more as it travels through more samples. Based on this observation, we can obtain volumetric cutaways that give the illusion of depth. By creating a mask that varies in opacity in the area of a cut (i.e., the rim), the recomposition produces images that simulate the relative depth that would occur due to a solid cut. Examples can be seen in Figure 6 and Figure 7. Note the appearance of a thick rim as a result of the alpha mask. In Figure 6, we simulate a triangular volumetric cut on the outer layers of the supernova (entropy) data set to reveal the partially occluded features. Specifically, parts of the 4 outer layers (blue, cyan, green, purple) are removed to reveal the 3 inner features (yellow, orange, red). Figure 7 simulates the application of cutaway planes for several layers of the lifted flame (mixfrac) data set. The scientists are interested in studying the distribution of the scalar field along the main flame surface to aid their understanding of the efficiency of the combustion process. Different cutaways help us visualize the partially occluded features to further the understanding of the scalar field. The first cutaway is applied to the first 2 outer layers (purple, pink) and the second cutaway is applied to the next 2 layers (red, orange), revealing the partially occluded inner (yellow) layer. In both examples, the insets demonstrate how the application of cutaways can help reveal hidden structures - all without re-rendering - thus, enabling meaningful exploration entirely in image space.

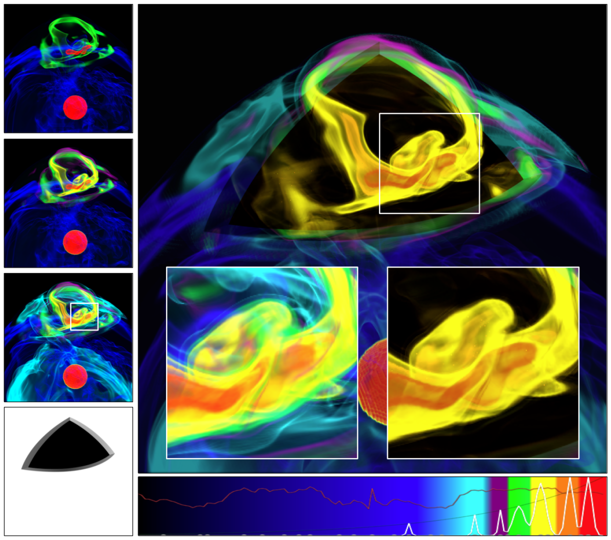


Figure 6: A triangular volumetric cutaway is applied to multiple outer layers of the supernova (entropy) data set. The extraction of 7 layers is performed using only 3 input images. The opacity of the outer layers (blue, cyan, purple, green) is decreased to reveal the inner features (yellow, orange, red). The insets show one of the areas where the finer details of the partially occluded inner structures are revealed with the use of a volumetric cutaway.

7 Discussion

As the number of images decreases, the problem of recovering layers becomes ill-posed and cannot be solved accurately. However, we show that even with a single image, our multi-layer extraction technique lets us obtain an approximation of the contribution of each layer to the input images. Figure 8 shows the results of layer extraction from 4, 3, 2, and a single image. Row (a) shows the recomposition of all the layers and row (c) - the recomposition with 3 out of 7 layers removed. Rows (b) and (d) show the difference between the synthesized and the ground truth (volume rendered)

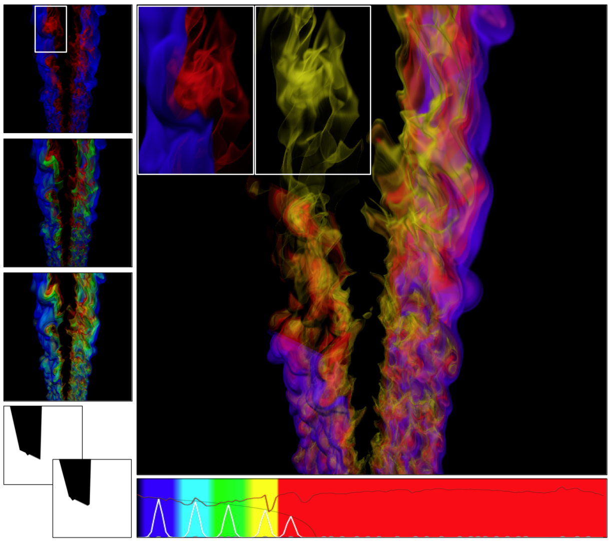


Figure 7: Volumetric cutaways are applied to several layers at a time, simulating cutting planes. Cutaway alpha masks are applied to the outer layers of a lifted flame (mixfrac) data set to reveal hidden structures. The first cutaway is applied to the 2 outermost layers (purple, pink) and the second cutaway is applied to the next 2 layers (red, orange), revealing the partially occluded inner layer (yellow).

images. Note that the difference magnitude is amplified for clarity, specifically, it is increased by a factor of 2. The difference between layer extraction from 3 and 4 images is not noticeable. For 7 layers, only 3 images showing some combinations of layers suffice for accurate decomposition. Row (c) shows that for 2 images and a single image, the method is not able to properly isolate some of the layers. After removing 3 out of 7 layers, we can still see the contribution of some of the removed layers to the image. The actual average pixel error values for each example are plotted in Figure 9. Since the problem is ill-defined, there are naturally cases where accurate decomposition is not possible. For example, the use of attenuation may produce images where certain parts are hidden behind opaque features. Removal of the opaque layers then results in artifacts due to missing information. Although spatial consistency priors and alpha distributions can remove minor artifacts, entire hidden regions cannot be properly reconstructed without hole-filling techniques. Another approach is to include additional layers where the compositing is performed back-to-front instead of front-to-back and to modify the color likelihood equation accordingly.

Our technique can be used if only the images of a data set are available and not the original volume. Figure 10 demonstrates the application of our technique to the CT data set of a chameleon. We extract 3 layers of the data set (skin, muscle, and bone) using only 2 images as input. We approximated the color and opacity of each layer to perform decomposition. We created two synthesized images of the dataset. In the first one, the skin layer is completely removed. In the second image, a volumetric cutaway was applied to the bone layer to reveal a feature hidden inside.

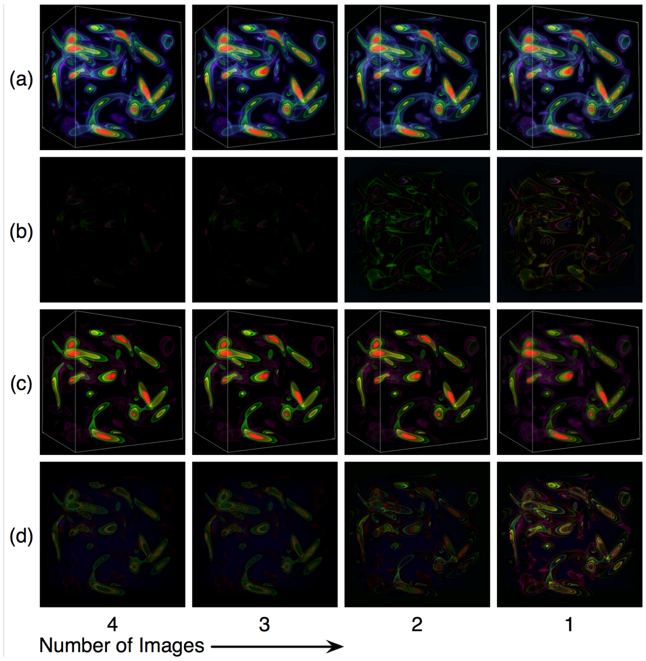


Figure 8: Examples of explorable images created with a different number of input images. There are 7 layers and 4 input images (same setup as in Figure 3). The optimization is performed with 4, 3, 2, and 1 input images (from left to right). (a) Synthesized images, where layer extraction is performed with a different number of images. The error between the synthesized images in (a) and the original rendering is shown in (b). (c) Synthesized images with 3 out of 7 layers fully removed. The error between the synthesized images in (c) and ground truth is shown in (d). Note that the difference magnitude is amplified (increased by a factor of 2), for clarity.

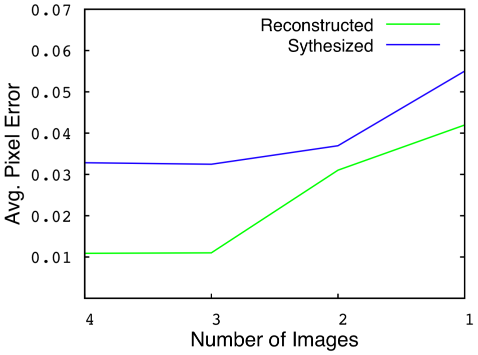


Figure 9: Average error per pixel for explorable images created with a different number of input images. The average error per pixel values are computed from difference images in Figure 8(c) and (d).

8 Conclusion

Volume data visualization has become an important tool in many areas of study and practice from medicine, 3D applications in science and engineering, to industrial non-destructive testing. The concept of explorable images suggests a new approach to interactive volume visualization, leading to a more accessible and affordable solution. Interactive high-quality volume visualization will become available to us on devices from mobile phones, netbooks, desktop computers, to large-scale, high-resolution displays without the need to transfer large data sets for visualization and a high-end rendering engine. Explorable images can also be far more clear, informative, and compelling than their static counterparts. However, to turn explorable images into a truly useful tool for real-world applications, we need to extend explorable images to also support spatial and temporal domains exploration. Consequently, the other essential task is to design an interactive interface for exploration in the resulting multi-dimensional space. We envision explorable images will stimulate a new line of research in the field of visualization.

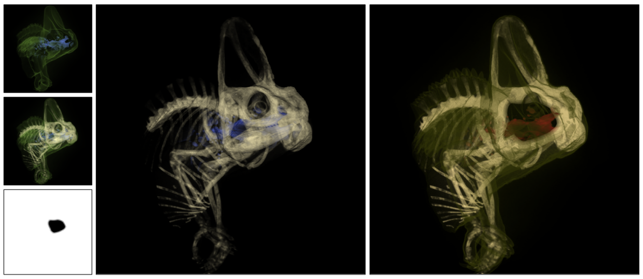


Figure 10: Example of layer extraction from existing input images, with no volume data available. Decomposition is done using 2 input images (left). In the first synthesized image, the skin layer is completely removed. In the second synthesized image, a volumetric cutaway is applied to the bone layer to reveal a feature hidden inside.

Acknowledgements

This research was supported in part by the U.S. National Science Foundation through grants OCI-0325934, OCI-0749217, CNS-0551727, CCF-0811422, OCI-0749227, OCI-0950008, CCF-0938114 and OCI-0850566, and the U.S. Department of Energy through the SciDAC program with Agreements No. DE-FC02-06ER25777 and DE-FG02-08ER54956. Data sets courtesy of Deborah Silver of Rutgers University, John Blondin of North Carolina State University, Jackie Chen of Sandia National Laboratories, and DigiMorph, CTLab, and TACC, University of Texas at Austin.

References

[1] A. Berman, P. Vlahos, and A. Dadourian. Comprehensive method for removing from an image the background surrounding a selected object. U.S. Patent 6, 134, 345, 2000.

[2] A. Berman, P. Vlahos, and A. Dadourian. Comprehensive method for removing from an image the background surrounding a selected object. U.S. Patent 6, 134, 346, 2000.

[3] T. Chen, Y. Wang, V. Schillings, and C. Meinel. Grayscale image matting and colorization. In *Proc. of Asian Conference on Computer Vision*, volume 1, pages 1164–1169, 2004.

[4] J. Choi and Y. Shin. Efficient image-based rendering of volume data. In *Proc. of Pacific Conference on Computer Graphics and Applications*, page 70, 1998.

[5] Y. Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proc. of Computer Vision and Pattern Recognition*, volume 1, pages 264–271, 2001.

[6] C. Donner and H. W. Jensen. Light diffusion in multi-layered translucent materials. In *Proc. of SIGGRAPH*, pages 1032–1039, 2005.

[7] C. Donner, T. Weyrich, E. d'Eon, R. Ramamoorthi, and S. Rusinkiewicz. A layered, heterogeneous reflectance model for acquiring and rendering human skin. In *Proc. of SIGGRAPH Asia*, pages 1–12, 2008.

[8] S. W. Hasinoff and K. N. Member-Kutulakos. Photo-consistent reconstruction of semitransparent scenes by density-sheet decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):870–885, 2007.

[9] T. He, L. Hong, A. Kaufman, and H. Pfister. Generation of transfer functions with stochastic search techniques. In *Proc. of Visualization Conference*, pages 227–ff., 1996.

[10] P. Hillman, J. Hannah, and D. Renshaw. Alpha channel estimation in high resolution images and image sequences. In *Proc. of Computer Vision and Pattern Recognition*, volume 1, pages I–1063–I–1068, 2001.

[11] I. Ihrke and M. Magnor. Image-based tomographic reconstruction of flames. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 365–373, 2004.

[12] S. B. Kang. A survey of image-based rendering techniques. In *Proc. of Videometrics, SPIE*, pages 2–16, 1999.

[13] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proc. of IEEE Symposium on Volume Visualization*, volume 1, pages 79–86, 1998.

[14] G. Kindlmann, E. Reinhard, and S. Creem. Face-based luminance matching for perceptual colormap generation. In *Proc. of Visualization Conference*, pages 299–306, 2002.

[15] E. LaMar and V. Pascucci. A multi-layered image cache for scientific visualization. In *Proc. of IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, pages 61–68, 2003.

[16] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. In *Proc. of Computer Vision and Pattern Recognition*, pages 1699–1712, 2007.

[17] M. Magnor, G. Kindlmann, and C. Hansen. Constrained inverse volume rendering for planetary nebulae. In *Proc. of Visualization Conference*, pages 83–90, 2004.

[18] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proc. of SIGGRAPH*, pages 389–400, 1997.

[19] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. S. A. Avila, K. Martin, R. Machiraju, and J. Lee. The transfer function bake-off. *Computer Graphics and Applications*, 21(3):16–22, 2001.

[20] R. J. Qian and M. I. Sezan. Video background replacement without a blue screen. In *Proc. of International Conference on Image Processing*, volume 4, pages 143–146, 1999.

[21] P. Rautek, S. Bruckner, and M. E. Groller. Semantic layers for illustrative volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1336–1343, 2007.

[22] B. E. Rogowitz and A. D. Kalvin. The "which blair project": a quick visual method for evaluating perceptual color maps. In *Proc. of Visualization Conference*, pages 183–190, 2001.

[23] T. Ropinski, J.-S. Prassni, F. Steinicke, and K. H. Hinrichs. Stroke-based transfer function design. In *Proc. of IEEE/EG International Symposium on Volume and Point-Based Graphics*, pages 41–48, 2008.

[24] M. A. Ruzon and C. Tomasi. Alpha estimation in natural images. In *Proc. of Computer Vision and Pattern Recognition*, volume 1, pages 18–25, 2000.

[25] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. of Computer Vision and Pattern Recognition*, page 1067, 1997.

[26] J. Shade, S. Gortler, L. wei He, and R. Szeliski. Layered depth images. In *Proc. of SIGGRAPH*, volume 1, pages 231–242, 1998.

[27] A. R. Smith and J. F. Blinn. Blue screen matting. In *Proc. of SIGGRAPH*, volume 1, pages 259–268, 1996.

[28] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. *ACM Transactions on Graphics*, 23(3):315–321, 2004.

[29] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *Proc. of Computer Vision and Pattern Recognition*, volume 1, pages 246–253, 2000.

[30] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Proc. of IEEE International Conference on Computer Vision*, volume 1, pages 936–943, 2005.

[31] L. Wang, J. Giesen, K. T. McDonnell, P. Zolliker, and K. Mueller. Color design for illustrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1739–1754, 2008.

[32] D. Weiskopf. On the role of color in the perception of motion in animated visualizations. In *Proc. of Visualization Conference*, pages 305–312, 2004.

[33] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman. Bayesian estimation of layers from multiple images. In *Proc. of European Conference on Computer Vision-Part III*, volume 1, pages 487 – 501, 2002.

[34] Y. Wu and H. Qu. Interactive transfer function design based on editing direct volume rendered images. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1027–1040, 2007.

[35] S. Yamazaki, M. Mochimaru, and T. Kanade. Inverse volume rendering approach to 3d reconstruction from multiple images. In *Proc. of Asian Conference on Computer Vision*, volume 1, pages 408–413, 2006.

[36] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin. Environment matting and compositing. In *Proc. of SIGGRAPH*, volume 1, pages 205–214, 1999.