Monocular Depth Estimation for Semi-Transparent Volume Renderings

Dominik Engel, Sebastian Hartwig and Timo Ropinski

Abstract—Neural networks have shown great success in extracting geometric information from color images. Especially, monocular depth estimation networks are increasingly reliable in real-world scenes. In this work we investigate the applicability of such monocular depth estimation networks to semi-transparent volume rendered images. As depth is notoriously difficult to define in a volumetric scene without clearly defined surfaces, we consider different depth computations that have emerged in practice, and compare state-of-the-art monocular depth estimation approaches for these different interpretations during an evaluation considering different degrees of opacity in the renderings. Additionally, we investigate how these networks can be extended to further obtain color and opacity information, in order to create a layered representation of the scene based on a single color image. This layered representation consists of spatially separated semi-transparent intervals that composite to the original input rendering. In our experiments we show that adaptations of existing approaches to monocular depth estimation perform well on semi-transparent volume renderings, which has several applications in the area of scientific visualization.

Index Terms—Volume rendering, depth compositing, monocular depth estimation.

1 INTRODUCTION

Visualization of volumetric data is a common necessity in many applied sciences, to explore new phenomena, understand the acquired measures or to make diagnoses based on them. Methods like direct volume rendering (DVR) are commonly used in practice nowadays, and countless visualizations have been created using DVR or similar techniques. In general, DVR is the preferred method to visualize 3D data using semi-transparent structures, allowing for inspection of multiple structures at once that would normally occlude each other. However, such volume renderings usually have to be created with great care and often require a significant amount of manual labor to find good rendering parameters, such as the transfer function (TF), which assigns optical properties to the data. In practice, parameters like the TF that are essential in order to reproduce such a visualization, are often not shared, or even the underlying original data is unavailable, which makes volume renderings impossible to reproduce or change in a meaningful way.

In this work, we investigate the abilities of neural networks to extract additional relevant information from single RGB images of such visualizations, with the goal of making these rendered images re-usable through composition. As basis for our approach, we investigate existing approaches for monocular depth estimation and adapt them to semi-transparent scenes. For this, a less ambiguous definition of depth is required, as semi-transparent renderings do not have a single depth value as opaque scenes, but rather have multiple relevant structures lying behind each other on a view ray. Looking at prior work, there are some attempts to find the most relevant of these structures in the context of picking (e.g., [1,35]). We employ these techniques to extract depth information for surfaces of interest, and refer to these techniques as depth techniques in the remainder of this paper.

Throughout this paper, we compare the performance of multiple state-of-the-art (SoA) monocular depth estimation networks for different depth techniques and different degrees of opacity in the rendered images. To our knowledge, this is the first study considering monocular depth estimation on volume rendered images. Based on our findings, we modify the SoA models to not only predict the depth of the surface of interest, but also the color and opacity of the semi-transparent structures in front, as well as the color behind this surface of interest (see Fig. 1). Based on the obtained layers, which can be predicted from a
single RGB-only volume rendered image, layer-wise re-compositing of the original renderings becomes possible, such that for instance additional meshes can be faithfully integrated into the volume rendering. All without any knowledge about the underlying volume data, transfer function or any other rendering parameter.

To achieve our goals, we make the following contributions within this paper:

- We evaluate SotA monocular depth estimators on volume rendered images subject to different amounts of semi-transparency.
- We extend the best performing model to predict a layered image representation, containing the depth of a visually dominant surface, as well as color and alpha layers in front and in the back of these surfaces, enabling the re-composition of the single input color image.
- We demonstrate the efficacy of our approach by compositing mesh geometries into existing volume renderings for which we only have access to RGB images.

We further make our code, datasets and trained models publicly available\(^1\) in order to enable other visualization researchers to use and further extend the presented techniques.

2 RELATED WORK

Within this section, we discuss prior work related to our approach. We will first provide an overview about depth techniques in the context of direct volume rendering, before we briefly recap the SotA in monocular depth estimation.

**Volume rendering depth.** Traditionally, volume rendered images are generated using the emission absorption model, which is a physics-inspired model underlying the popular volume rendering integral [24]. While this approach leads to faithfully volume rendered images of the volume data at hand, the cloudy nature of the resulting imagery let researchers investigate methods to visually emphasize more well-defined and delineated objects, to support object detection and quantification. The most seminal work in this direction is most likely Levoy’s early paper on gradient-based surface extraction from volume data [20]. Kindlmann’s and Durkin’s build up on this idea, to obtain gradient-based transfer functions [14], in which they exploit the gradient magnitude at a sample in order to modulate the sample’s opacity. Further, inspired by this approach, several other transfer function approaches followed, that exploit particular properties to modulate opacity and thus obtain more surface-like representations (e.g. [4, 7, 15, 27]). Naturally, with the opportunities to represent well delineated objects in volume rendered images, the wish to fuse volume rendered images with geometry meshes arose [13]. While this can be directly achieved when extracting surfaces from volumes [26], the integration of geometry into standard ray-casting based volume renderings [16] required the generation of a meaningful depth map in order to support correct image compositing [21]. The most straightforward approach to obtain depth values for a volume rendered image, is the use of FirstHit depth values, where the depth value represents the depth of the first non-transparent sample along a viewing ray. Needless to say, that a single depth value cannot represent the complexity of a volume rendered image containing semi-transparent structures. The same observation is true for maximum gradient magnitude based depth maps, whereby the depth value represents the depth of the sample with the maximum gradient magnitude along a view ray. Other, more advanced approaches allow for the extraction of multiple depth values along a ray. Lindholm for instance applied a buffer based depth peeling to obtain a complex depth structure, quite faithfully representing the complex nature of the volume rendered object [21]. Bruckner and Gr"oller presented the MIDA method, which is not particularly targeted at extracting depth values, but can obtain meaningful surface representations by exploiting the compositing gradient [3]. Similarly, the what you see is what you pick (WYSIWYP) approach allows for the extraction of several layers, originally meant to support picking of visually salient features [32, 35]. While we suggest using such approaches to compute depth maps during volume rendering, Stoppel and Bruckner have also shown that layers can be used to guide interaction widgets in the same context [31]. Apart from these approaches, which are somehow integrated into the rendering process, also other more advanced models exist, which support the extraction of boundaries. Lindholm et al. for instance propose the improved visualization of boundary surfaces by exploiting material-specific reconstruction in the classification process [22].

**Monocular depth estimation.** After the first approaches started to leverage convolutional neural networks (CNNs) to directly estimate depth from a single input image [8, 9], monocular depth estimation techniques developed in a rapid way. Scores on famous benchmarks such as NYU [30] and KITTI [12] have been surpassed multiple times. While standard CNNs reduce the spatial resolution of feature maps, other work implements multi-layer convolutional networks [17] and multiscale networks [8, 9, 18]. Framing depth estimation as classification task, Fu. et al. introduce deep ordinal regression networks which divide depth values into discret ordinal labels [11], instead of directly regressing depth [19, 37]. Maximov et al. improve depth estimation using domain invariant defocused synthetic images as supervision in order to close the reality gap [25].

Segmentation masks which separate individual objects in the input image pose a strong prior to depth estimation. Wang et al. perform semantic image segmentation and apply a divide and conquer strategy to depth estimation [33]. Independently, for each segment, depth maps are predicted in a canonical space and then recomposed to global space. Zhu et al. use semantic segmentation masks for better depth estimation, especially along object borders [39]. In the work of Ramamonjisoa et al. they also focus on enhancing depth predictions around occlusion boundaries using displacement fields [28]. Each dataset comes with distinct characteristics and biases. Circumventing domain bound depth predictions, Ranftl et al. combine multiple large datasets in order to train a network that is invariant to changes in depth range and scale [29]. Virtual normals have been introduced by Yin et al. enforcing high-order geometric constraints in 3D space for the depth prediction task [37]. Their geometric loss function projects the predicted depth values into 3D, by using a pinhole camera model, enabling to compute virtual normals, to encode geometric constraints. More recently, Zhao et al. trained a network that removes clutter and novel objects from real images, in order to improve depth prediction [38]. In the work of Lee et al. [18], they present a novel local guidance layer, which is used to compute locally-defined relative depth estimations, on different levels of resolution. While very few monocular depth estimation papers also mention transparent surfaces [6], to our knowledge no systematic study as well as possible applications in the context of volume rendering have been investigated.

3 METHOD

To investigate SotA monocular depth estimation in the context of volume rendering, we have created large scale training data sets to train existing estimators. Here we first discuss these datasets, how they were generated, and how we controlled the amount of opacity in them. Furthermore, we detail how we tackle the depth ambiguity apparent in semi-transparent structures, resulting from the lack of clear surfaces. Therefore, we compare multiple depth techniques as mentioned above. We then elaborate on the differences to the common real-world datasets for monocular depth estimation and the resulting modifications of the neural models. Lastly, we detail how we extend the monocular depth estimation models to also predict color and opacity layers in addition to depth, allowing for the de-composition of the input RGB-only volume rendering to enable visualization operations, such as for instance geometry integration.

3.1 Datasets

3.1.1 Dataset Properties

The standard monocular depth estimation task requires predicting a depth map from just an RGB color image. Most existing approaches to monocular depth estimation are trained for real-world scenes, with the...
which consists of 491 CT scans of human heads. Of those, we used 397 as basis for our synthetic dataset, we need a large corpus of volume data and cover five different work and standard datasets to our requirements when applying the task to images containing semi-transparent structures. Mainly, we have identified the following three differences:

**Depth sparsity.** The standard datasets generally provide quite densely labeled depth maps, with only few missing values due to reflections and occlusions in the ground truth acquisition. For the case of synthetic volume rendered images, we usually have much sparser images, as a large percentage of pixels covers background without a relevant depth. During training, this high percentage of irrelevant pixels would dominate the loss, leading to poor training signals. We tackle this problem using loss masking.

**Absolute vs. relative depth.** Another relevant aspect is the depth range in the datasets. The standard datasets commonly denote their depth labels in meters, whereas synthetic renderings often do not encode real-world units at all, but rather use a rendering-centered scaling of the depth values within the view frustum. As a result, we are more interested in correctly predicting depth values relative to each other, as opposed to producing accurate absolute values.

**Depth ambiguity.** Lastly, depth is quite ambiguous to define in semi-transparent renderings, because there are no clear surfaces to choose, as compared to the real-world datasets. In addition to that, there may also be multiple relevant structures in front of each other without full occlusion, allowing for multiple valid choices. In this work, we first focus on predicting the most relevant surface from a visual perspective. For this, we looked at solutions from prior work in the context of picking. A picking layer holds information about what structure or object is "picked" (for each pixel) during a mouse interaction. Naturally, a similar ambiguity problem arises and prior work has proposed methods to decide for the most relevant structure. We make use of this prior work and cover five different depth techniques in Sec. 3.2.

### 3.1.2 Dataset Generation

As basis for our synthetic dataset, we need a large corpus of volume data for which we can generate semi-transparent renderings automatically in order to create a sufficiently large training dataset. We chose to generate our training dataset from the CQ500 dataset by QureAI [5], which consists of 491 CT scans of human heads. Of those, we used 397 to generate training images, 40 for validation and 40 for testing of the neural network. The volumes have a resolution of 512 × 512 per slice and have between 101 and 645 slices per volume.

From the volumes above, we generate 100K training images from random viewpoints and with randomized transfer functions (TF). The viewpoints are drawn at random from a uniform distribution of viewing directions with a random distance between 2.5 and 3.0 to the volume center. This configuration places the volume around the center of the view frustum. The scenes are rendered with a fixed far plane distance \( f = 5.0 \), and we define our depth \( d \) as the normalized distance to the camera \( c \) for a sample position \( p \):

\[
d = |p - c|_2 / f
\]

To vary the appearance, and to control the amount of opacity in the renderings, we propose a transfer function generation scheme, similar to the approach proposed by Engel et al. [10]. This scheme generates piece-wise linear ID transfer functions using randomly generated trapezoids, representing "peaks". For our datasets, we generate TFs containing between one and three of such peaks, while randomizing their widths (along intensity space) and heights (assigned opacity) to achieve a high variation of renderings with a controllable amount of opacity in the rendered images. Controlling the amount of opacity is relevant to investigate how the amount of semi-transparent structures complicates the monocular depth estimation task in comparison to fully opaque scenes. Hence, we generate two different datasets:

Hereby, we classify a pixel as influenced by semi-transparency if its accumulated opacity is in the interval \([0.02, 0.9]\), where an opacity of 0.0 corresponds to fully opaque and 0.0 to fully transparent. For the generation of the \( \text{TRANSPARENT} \) dataset, we decreased the maximum height of the generated trapezoid peaks compared to \( \text{Opaque} \) from 0.6 to 0.4. During generation, we reject all samples that do not satisfy the amount of opacity condition. The resulting \( \text{Opaque} \) dataset has an average of 5.2% semi-transparent pixels and \( \text{TRANSPARENT} \) has an average of 29.8% semi-transparent pixels. During training we further apply the standard image augmentation techniques employed by the respective depth-estimation method.

#### 3.2 Depth Techniques

Due to the ambiguity of surfaces in semi-transparent renderings, it is notoriously difficult to find a definitive depth value for a given pixel in such renderings. Prior work on volume picking offers a variety of approaches that are designed to find "relevant" surfaces or objects from a human perspective [36]. Together with some common choices, we compare a total of five depth techniques that we include in our experiments. Assuming a ray \( r(t) = c + t \cdot \vec{r} \) with ray direction \( \vec{r} \) and local and accumulated opacities \( \sigma_r(t) \) and \( \sigma^*_r(t) \), we use the following depth techniques (compare Fig. 2):

**FirstHit.** This is the simplest strategy. The depth of a pixel is given by the position of the first sample that has a non-zero opacity along a view ray. This strategy completely discards semi-transparency and essentially treats the scene like an opaque object.

\[
d_{\text{FH}} = \min_t \{ t | \sigma_r(t) > 0 \}
\]

**MaxOpacity.** This strategy corresponds to taking the depth of the
sample with the respective opacity along a ray, i.e., the depth of the sample used for Maximum Intensity Projection (MIP).

$$d_{MO} = \arg\max_t \sigma_r(t)$$

**MaxGradient.** The MaxGradient strategy looks for the steepest change in accumulated opacity along a ray. Similar to the following methods, this favors early samples along the rays.

$$d_{MG} = \arg\max_t \frac{\partial \sigma_r(t)}{\partial t}$$

**WYSIWYP.** This approach follows the picking strategy presented by Wiebel et al. [35]. This strategy finds intervals along each ray, defined by the zero-crossing of the second order derivative of the accumulated opacity. An interval begins when the second derivative becomes positive after being negative or zero. The interval ends after the second derivative has plateaued at 0 again, corresponding to regions without opacity accumulation. After defining intervals, the chosen depth value is the distance to the start of the interval that increased the accumulated opacity the most.

$$d_{WYSIWYP} = \arg\max_{t_{s,i}} (\sigma_r(t_{c,i}) - \sigma_r(t_{s,i}))$$

$$t_{s,i} = \min_{t > t_{s,i-1}} \left\{ t \mid \frac{\partial^2 \sigma_r(t)}{\partial t^2} = 0, \frac{\partial^3 \sigma_r(t)}{\partial t^3} > 0 \right\}$$

$$t_{c,i} = \min_{t < t_{c,i}} \left\{ t \mid \frac{\partial^2 \sigma_r(t)}{\partial t^2} = 0, \frac{\partial^3 \sigma_r(t)}{\partial t^3} = 0 \right\}$$

Here $t_s, t_e$ denote the start and end depth of the interval, and the first interval begins at $t_{s1} = d_{FH}$. The intervals are illustrated on top of Fig. 2.

**MIDA.** Lastly, we took inspiration from Maximum Intensity Difference Accumulation proposed by Bruckner et al. [3]. MIDA is a compositing technique to combine benefits of DVR and MIP. In their work, they define a $\beta$ parameter that trades some already accumulated opacity along a ray for new, highly relevant (in a MIP sense) structures. To find a depth value, we use the sample with the lowest $\beta$ parameter, i.e., the sample that increases over the previous MIP (up to this sample) the most.

$$d_{MIDA} = \arg\max_t \left( \sigma_r(t) - \max_{\tau < t} \sigma_r(\tau) \right)$$

### 3.3 Monocular Depth Estimation

As a first step, we investigate whether neural nets can predict depth under the above requirements. For this, we compare five Sota approaches that have shown great success on real-world data.

In the following experiments, we use the method of Laina et al. [17] (LAINA), one of the first deep convolutional neuronal networks for monocular depth estimation. Further, we use the well-known deep ordinal regression network (DORN), the first approach formulating depth estimation as classification problem. Additionally, we picked the approaches of Lee et al. [18], a multiscale approach (BTS) and Yin et al. [37] using virtual normals to enforce geometric constraints (VNL). Both approaches show strong performance on common depth estimation datasets. Finally, we adopt the network of Ranftl et al. [29](MIDAS), which yields robust depth predictions for a combination of multiple datasets, also being scale and shift invariant. In this task, we train the neural nets to predict a depth map for a semi-transparent rendering from just an RGB color image.

To adapt this task to semi-transparent images, we have to make minor adjustments to the originally proposed neural nets. One of the adjustments is to compensate for the fact that the synthetically generated renderings typically do not encode real-world sizes, but rather encode depth as a normalized distance within an enclosed view frustum. Specifically, we adapt the output activation functions, which are usually rectified linear units (ReLU), to sigmoid activations. This scales the network’s output to $(0, 1)$, aligning it with the normalized depth we require. We then trained the networks with their proposed training procedure, including data augmentation, and losses on both the OPAQUE and TRANSPARENT dataset. We report our results in Sec. 4.1.

### 3.4 Monocular Layered Representation Prediction

Since we found that the standard monocular depth estimation task works quite well on semi-transparent data, we further extended the approach, to predict additional information on top of the depth map. Given, that we can predict depths from relevant surfaces in the semi-transparent renderings, we now further ask the network to separate the color and opacity of the scene at those depths. That means we let the network predict the accumulated color and opacity in front of the predicted depth, as well as in the back of the predicted depth. In the following, we discuss our choice of the depth technique and depth estimator for this task and how they need to be adapted. Lastly, we present two novel ideas to further improve our results.

**Choice of depth technique.** For these experiments, we chose to use...
only the BTS [18] network in combination with the WYSIWYP depth. The reason is that depth techniques like MaxOpacity and MaxGradient result in very sparse front layers after separation, while using FirstHit would be meaningless, as front would be 0 by definition. MIDA and WYSIWYP both provide reasonable separation, however they both still coincide with FirstHit depths for many of the pixels. In order to include those thin semi-transparent layers in the front, we adjust the depths to be just behind that structure. While MIDA does not consider the thickness of structures, WYSIWYP readily provides full intervals. To include the first-hit structures, we modify the WYSIWYP approach to use the depth at the interval end, instead of the start, if the first interval would be chosen to represent the depth (compare Fig. 2).

Choice of neural net. We chose BTS [18] for our continued experiments for multiple reasons. Firstly, VNL [37] and DORN [11] disqualify for our further adaptations, because they are unable to scale to the thickness of structures, WYSIWYP readily provides full intervals. To include the first-hit structures, we modify the WYSIWYP approach to use the depth at the interval end, instead of the start, if the first interval would be chosen to represent the depth (compare Fig. 2).

Layered representation. We now ask the networks to separate the semi-transparent structures in a front layer, before the predicted depth values, and a back layer, behind the predicted depth, as illustrated in Fig. 1. The front layer can easily be retrieved during ray casting, by exporting the accumulated RGBA buffer when reaching the depth value to separate at. The back can then be computed from the front and the fully composited rendering. The resulting prediction is essentially a layered representation of the rendered scene, that composites to the original input image. This layered representation allows for visualization-centric modifications of the input image, like compositing additional objects into the scene, as also demonstrated in Fig. 1.

In order to predict such layered representation, we modify the last layers of BTS to output a 9-channel image, instead of a single-channel image. The 9 channels consist of:

- The depth map,
- RGBA of the structures in front of the depth map, and
- RGBA of the structures behind the depth map (back).

With the inclusion of color layers in the prediction, we also remove any color-related data augmentation that is part of the depth estimator’s training procedure. We refer to the resulting network as baseline (compare Fig. 5). While this baseline suffers from blurry layers and often fails to actually compose to the original image, we introduce the following two learning extensions.

Residual image output. In the baseline, we find that predicted front and back layers suffer from blurriness, which we attribute to the decoder of the network. This decoder part is prone to introducing blur, and the skip connections from early layers now have to transport a lot more information compared to when just predicting depth. As a solution to the blurriness, we propose Residual Image Outputs. Instead of directly predicting the front and back RGBA layers, we predict the residual to the input image. Using a tanh-activation we predict how much color and alpha must be removed from or added to the input image, while preserving the input pixel for weakly activated neurons. For the alpha channels, we predict the residuals to the mean of the input color channels, as the input image’s alpha is unavailable during inference. The idea is described by the following equation, given neural net $F$, parameterized by $\Theta$ and input image $x$:

$$\hat{c}_{\text{old}} = F_{\Theta}(x), \quad \hat{c}_{\text{new}} = F_{\Theta}(x) + x$$

Compositing loss. In addition, we introduce a novel, visualization-centered loss function, which we call Compositing Loss. The idea of the loss is to enforce consistency in the layered image representation, by enforcing the front and back layers to alpha-compose to the original input rendering, as described by the following equation, given color and opacity predictions $\hat{c}$ and $\hat{\alpha}$, number of valid pixels $N$ and ground truth color and opacity $c$, $\alpha$:

$$\hat{c} = c_{\text{front}} + (1 - c_{\text{front}}) \cdot c_{\text{back}} \cdot \hat{\alpha}$$

$$\hat{\alpha} = \alpha_{\text{front}} + (1 - \alpha_{\text{front}}) \cdot \hat{\alpha}_{\text{back}}$$

$$L_{\text{Comp}} = \frac{1}{N} \sum_{i} (\hat{c}_i - c_i)^2 + (\hat{\alpha}_i - \alpha_i)^2$$

Hereby we consider pixels as valid if they have an $\alpha > 0$, discarding the pixels showing only background, as they would dominate the loss due to the sparsity of our data. Note that this mask would be unavailable during inference, as we cannot expect to have an alpha channel for the input image. During inference we can instead mask the predicted output using the input color channels.

Overall loss. The overall loss function to train the network for joint depth estimation and color/alpha separation is a weighted sum of the $SILog$ loss [9] on the depth map only, a mean absolute error ($L_1$) on the color and alpha channels, and the compositing loss on the recomposed image, which backpropagates through the color and alpha

| Depth | FirstHit | MaxOpacity | MaxGradient | WYSIWYP | MIDA |
|-------|----------|------------|-------------|---------|------|
| BTS   | 0.963    | 0.981     | 0.987      | 0.026   | 0.937 |
| VNL   | 0.966    | 0.994     | 0.998      | 0.033   | 0.935 |
| LAINA | 0.927    | 0.958     | 0.970      | 0.021   | 0.904 |
| DORN  | 0.859    | 0.948     | 0.960      | 0.050   | 0.578 |
| MiDaS| 0.955    | 0.977     | 0.984      | 0.022   | 0.942 |

Table 1: Quantitative monocular depth estimation results for the different depth techniques, datasets and depth estimation networks (rows). We show depth predictions for the best method on the Transparent dataset in Fig. 3.
We have performed several experiments to investigate the monocular depth estimation, the monocular layered representation prediction, as well as the applicability in the wild. The depth estimation experiments evaluate how well neural networks perform on images of semi-transparent renderings, specifically we (1) measure the effect of the amount of transparency in the input images, (2) compare the different depth techniques presented in Sec. 3.2 to each other and (3) use the predicted depth maps to composite the input image with additional objects.

Based on the results of those first experiments, we choose a neural net and depth technique for our second set of experiments, investigating the monocular layered representation prediction. In this second set of experiments, we (1) compare the proposed baseline for the separation of alpha-blending. Lastly, we apply our approach to renderings taken as screenshots from the seminal VolumeShop paper [2] to demonstrate its applicability in the wild.

**Metrics.** We measure the performance of the networks using metrics commonly used in the domain of monocular depth estimation, as well as image similarity metrics for the RGBA layers:

- **δ-metric:** The $\delta_1, \delta_2, \delta_3$ metrics measure the percentage of valid predictions within a certain margin of relative error (here 25%):

  $$ \delta_k = \frac{1}{N} \left\{ \left| i \left| \max \left( \frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i} \right) < \lambda_k \right. \right\} \right. $$

  with $\lambda = 1.25$

- **$L_1$-metric:** mean absolute error of valid pixels

- **SSIM** [34]: The structured similarity index:

  $$ \text{SSIM}(a,b) = \frac{(2\mu_a\mu_b + \varepsilon_1)(2\sigma_{ab} + \varepsilon_2)}{(\mu_a^2 + \mu_b^2 + \varepsilon_1)(\sigma_a^2 + \sigma_b^2 + \varepsilon_2)} $$

Hereby $\varepsilon_i$ are small constants for numerical stability and $\mu_a, \sigma_a^2$ and $\sigma_{ab}$ are the means, variances and covariances within a local neighborhood for all pixels of (a and b) respectively.

For the $\delta$-metrics and $L_1$ we again only consider the $N$ valid pixels, indexed by $i$, that are not part of the background.

### 4.1 Monocular Depth Estimation

**Experiment 1.** In this experiment, we evaluate the applicability of five SotA monocular depth estimation approaches on the different depth techniques introduced in Sec. 3.2. The five approaches we test are DORN [11], MIDAS [29], LAINA [17], VNL [37] and BTS [18]. We chose this set of networks because they are top performers on the monocular depth estimation leaderboards and have publicly available implementations. The goal of this experiment is to find out if such approaches can be applied to semi-transparent images and how the degree of opacity impacts the neural nets. The quantitative results of this experiment are presented in Table 1. Additionally, Fig. 3 shows the depth predictions of the BTS depth estimator, for each of the depth techniques.

**Discussion.** The first thing to note from the experiment is that all approaches perform reasonably well on the presented data. Especially for the FirstHit depth technique, we get impressive $\delta$-scores, while the networks perform quite similar on the Opaque and Transparent datasets. In contrast, the remaining depth techniques show a bigger drop in performance when decreasing the amount of opacity. The MaxOpacity depth shows the most severe decrease in $\delta$-scores, and we think this is the case, because it is the least biased towards the front and thus the FirstHit depth. MaxOpacity is completely invariant to the position of structures along the ray, whereas all other techniques take the accumulated opacity into account, which increases faster at the beginning of a ray, because there has not occurred so much absorption yet.

**Experiment 2.** Here, we use the predicted depth maps to composite additional objects onto the input image. This experiment does not consider blending of colors and alpha, only showing the color with the lower depth for each pixel. The results are visualized for two different depth techniques in Fig. 4.

**Discussion.** As shown in Fig. 4, the quality of the composition is highly dependent on the depth technique. The FirstHit approach fails in Fig. 4a, as the semi-transparent structures inside the skull make the skull appear flat at the clipping plane, instead of showing the interior of the skull, as WYSIWYP does (see Fig. 4b).

### 4.2 Monocular Layered Representation Prediction

Based on the previous experiments and the adaptability reasons discussed in Sec. 3.4, we choose BTS in combination with the WYSIWYP depth technique for the upcoming experiments.

**Experiment 1.** This experiment evaluates our proposed extended version of BTS quantitatively. We predict the Front, Back and Depth layers from an input RGB image (see Fig. 1) and compare them with the respective ground truths using SSIM and $L_1$. The ablation study for our different variants, with Residual Image Outputs and the Compositing Loss can be seen in Table 2 with the according layered representations illustrated in Fig. 5.

**Discussion.** The resulting layers shown in Fig. 5 make clear that our baseline approach, with just the number of output layers of the network adapted, suffers from blurry predictions. The proposed residual image outputs ($+R$) in contrast are much crisper, as the network receives a pixel-precise signal to its last layer, making it easier to keep sharp boundaries. But still, the $+R$ variant’s re-composition is visually quite
We show the Front layer (+C). Visual results in Fig. 5.

Table 2: Ablation study of the Residual Image Output (+R) and the Compositing Loss (+R +C). Visual results in Fig. 5

| METHOD       | FRONT SSIM ↑ | FRONT L1 ↓ | BACK SSIM ↑ | BACK L1 ↓ | COMPOSED SSIM ↑ | COMPOSED L1 ↓ |
|--------------|--------------|------------|-------------|-----------|----------------|---------------|
| Base         | 0.867        | 0.035      | 0.872       | 0.034     | 0.935          | 0.028         |
| +R           | 0.847        | 0.026      | 0.889       | 0.030     | 0.949          | 0.018         |
| +R+C         | 0.856        | 0.025      | 0.889       | 0.031     | 0.960          | 0.017         |

**Discussion.**

As can be seen in Fig. 7c, the naive approach to compositing an RGB image with additional geometry usually does not produce any meaningful result, as the geometry simply occludes the image. Our approach on the other hand manages to correctly occlude the cylinders behind the bones of the fingers. Additionally, the cylinders are partially occluded behind the semi-transparent skin of the hand. Compared to the ground truth, our approach results in a slightly more opaque and darker skin color, but is visually still very close to the ground truth.

**Experiment 2.** In this experiment, we first predict the layered representation of the input image. We use this layered representation to re-compose the input image with additional geometry. To composite the new geometry with the layered representation, we use the Depth map together with the Back layer to render intersections of the new geometry with the relevant structures, before rendering the Front layer on top. The results can be seen in Fig. 7, where we compare our approach with a naive approach that renderings, by only using a single RGB image extracted as a screenshot without having any access to the renderer. While we have no ground truth for these predictions, it is still quite clear that the quality of the depth maps still leave room for improvement. For example in the depth prediction of the hand, most of the blood vessels are not visible and some of the dark shadows in the split-open head cause holes in the predicted depth map. Despite these limitations, our method already enables meaningful re-compositions of visualizations taken from the wild.

**4.3 Application in the Wild**

In the previous experiments we evaluated our approach on test datasets that are generally close to the training dataset, in the sense that most of the evaluated images contained either renderings of skulls from the CQ500 dataset that we excluded from training, or other human parts, e.g., the hand CT from Fig. 7. To investigate how our approach generalizes beyond this data, we have applied it to renderings of a salamander as well as the engine block (see Fig. 6). While these images still have been generated with the same volume renderer we have used to generate our training data, the ultimate test is to see how our method generalizes beyond such renderings. Therefore, to evaluate such an applicability in the wild, we run our method on screenshots taken from the seminal VolumeShop paper [2]. Those screenshots partially contained labels and zoom ins, used different shading techniques and aspect ratios. We used a simple flood-fill algorithm for a very rough background extraction, removed annotations with deep inpainting [23], and resized the images to 512 × 512 pixels before predicting a layered representation. Finally, we re-composite the input image with additional geometry, before resizing the image to its original resolution. The results are shown in Fig. 8. For the opaque objects, where the predicted depth map is close to the FirstHit depth, we make a slight change to the compositing. If the object is mostly opaque, errors in the Front layer can become visually quite obvious, when the Front is blend over the additional geometry, where it should fully occlude the object. Consequently, we make use of the fact that our layered representation composites to an RGBA version of the input image. Using the original input RGB, the predicted alpha of the re-composition of the layered representation and the predicted depth map, we can perform normal depth-based alpha blending.

Dotted results for non-human volumes shown in Fig. 6 indicate, that our approach rather learns properties of volume rendered CT images than the actual geometry, as we can still generate appealing results for other specimen and even objects. As shown in Fig. 8, our approach allows for compositing new geometry into existing volume renderings, by only using a single RGB image extracted as a screenshot without having any access to the renderer. While we have no ground truth for these predictions, it is still quite clear that the quality of the depth maps still leave room for improvement. For example in the depth prediction of the hand, most of the blood vessels are not visible and some of the dark shadows in the split-open head cause holes in the predicted depth map. Despite those limitations, our method already enables meaningful re-compositions of visualizations taken from the wild.

**5 Conclusions and Future Work**

Within this paper, we have presented the first systematic study investigating the capabilities of deep monocular depth estimation techniques in the context of volume rendered images containing semi-transparent structures. We could show, that SotA monocular depth estimators can be used to predict meaningful depth maps for such images. Motivated by these findings, we have extended the BTS approach, to not only allow for the estimation of depth, but also of a layered image represen-
tation containing color and opacity. Our extension not only modulates the output of the model, but also facilitates residual learning and exploits a novel compositing loss. With the presented approach, we are able to generate meaningful layered representation of RGB-only volume rendered images, and are thus able to modify them according to visualization-centric tasks, such as for instance the integration of mesh geometry.

In the future, we would like to investigate other tasks which could potentially benefit from our approach. We consider the investigation of novel view synthesis, and subsequently changing the transfer function, just to name a few opportunities. We would also like to more systematically evaluate the influence of the underlying shading model on the depth estimation task.

ACKNOWLEDGMENTS

This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG) under grant 391107954 (Inviwo). Some of the figures were produced using the Inviwo framework (www.inviwo.org).

REFERENCES

[1] F. Argelaguet and C. Andujar. A survey of 3d object selection techniques for virtual environments. Computers & Graphics, 37(3):121–136, 2013.
[2] S. Bruckner and M. E. Groller. Volumesharp: An interactive system for direct volume illustration. IEEE, 2005.
[3] S. Bruckner and M. E. Groller. Instant volume visualization using maximum intensity difference accumulation. Computer Graphics Forum, 28(3):775–782, 2009.
[4] J. J. Caban and P. Rheingans. Texture-based transfer functions for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics, 14(6):1364–1371, 2008.
[5] S. Chilamkurthy, R. Ghosh, S. Tanamala, M. Biviji, N. G. Campeau, V. K. Venugopal, V. Mahajan, P. Rao, and P. Warier. Development and validation of deep learning algorithms for detection of critical findings in head ct scans. arXiv preprint arXiv:1803.05854, 2018.
[6] J. Choi, D. Jung, Y. Lee, D. Kim, D. Manocha, and D. Lee. Selfdecop: Self-supervised monocular depth completion in challenging indoor environments. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 467–474. IEEE, 2021.
[7] C. Correa and K.-L. Ma. Size-based transfer functions: A new volume exploration technique. IEEE transactions on visualization and computer graphics, 14(6):1380–1387, 2008.
[8] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE international conference on computer vision, pp. 2650–2658, 2015.
[9] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Advances in neural information processing systems, pp. 2366–2374, 2014.
[10] D. Engel and T. Ropinski. Deep volumetric ambient occlusion. IEEE Transactions on Visualization and Computer Graphics, 27(2):1268–1278, 2020.
[11] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2002–2011, 2018.
[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11):1231–1237, 2013.
[13] A. Kaufman, R. Yagel, and D. Cohen. Intermixing surface and volume rendering. In 3D Imaging in Medicine, pp. 217–227. Springer, 1990.
[14] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In IEEE Symposium on Volume Visualization (Cat. No. 989EX300), pp. 79–86. IEEE, 1998.
[15] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Møller. Curve-based transfer functions for direct volume rendering: Methods and applications. In IEEE Visualization, 2003. VIS 2003., pp. 513–520. IEEE, 2003.
[16] J. Krüger and R. Westermann. Acceleration techniques for gpu-based volume rendering. In Visualization Conference, IEEE, pp. 38–38. IEEE Computer Society, 2003.
[17] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In 2016 Fourth international conference on 3D vision (3DV), pp. 239–248. IEEE, 2016.
[18] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh. From big to small: Multiscale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10126, 2019.
[19] J.-H. Lee and C.-S. Kim. Monocular depth estimation using relative depth maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9729–9738, 2019.
[20] M. Levy. Display of surfaces from volume data. IEEE Computer graphics and Applications, 8(3):29–37, 1988.
[21] S. Lindholm, M. Fält, E. Sundén, A. Bock, A. Ynnerman, and T. Ropinski. Hybrid data visualization based on depth complexity histogram analysis. In Computer Graphics Forum, vol. 34, pp. 74–85. Wiley Online Library, 2015.
[22] S. Lindholm, D. Jönsson, C. Hansen, and A. Ynnerman. Boundary aware reconstruction of scalar fields. IEEE transactions on visualization and computer graphics, 20(12):2447–2455, 2014.
[23] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European conference on computer vision (ECCV), pp. 85–100, 2018.
[24] N. Max. Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics, 1(2):99–108, 1995.
[25] M. Maximov, K. Galim, and L. Leal-Taixe. Focus on defocus: bridging the synthetic to real domain gap for depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1071–1080, 2020.
[26] T. S. Newman and H. Yi. A survey of the marching cubes algorithm. Computers & Graphics, 30(5):854–879, 2006.
[27] J.-S. Pražník, T. Ropinski, J. Mensmann, and K. Hinrichs. Shape-based transfer functions for volume visualization. In 2010 IEEE Pacific Visualization Symposium (PacificVis), pp. 9–16. IEEE, 2010.
[28] M. Ramamonjisoa, Y. Du, and V. Lepetit. Predicting sharp and accurate
Fig. 8: **Applying our approach in the wild.** Here we took screenshots from the seminal VolumeShop paper [2], resized them to $512 \times 512$ and removed the background using a simple floodfill algorithm. Using the input’s color and the alpha of the predicted re-composited rendering together with the predicted depth map allows for realistic compositing with additional geometry. Note that for the hand we additionally used deep image inpainting [23] to remove the zoom-in frame.
occlusion boundaries in monocular depth estimation using displacement fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14648–14657, 2020.

[29] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE transactions on pattern analysis and machine intelligence, 2020.

[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In European conference on computer vision, pp. 746–760. Springer, 2012.

[31] S. Stoppel and S. Bruckner. Smart surrogate widgets for direct volume manipulation. In 2018 IEEE Pacific Visualization Symposium (PacificVis), pp. 36–45. IEEE, 2018.

[32] S. Stoppel, H.-C. Hege, and A. Wiebel. Visibility-driven depth determination of surface patches in direct volume rendering. 2014.

[33] L. Wang, J. Zhang, O. Wang, Z. Lin, and H. Lu. Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 541–550, 2020.

[34] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, Apr. 2004. doi: 10.1109/TIP.2003.819861

[35] A. Wiebel, F. M. Vos, D. Foerster, and H.-C. Hege. Wysiwyp: What you see is what you pick. IEEE Transactions on Visualization and Computer Graphics, 18(12):2236–2244, 2012.

[36] A. Wiebel, F. M. Vos, and H.-C. Hege. Perception-oriented picking of structures in direct volumetric renderings. 2011.

[37] W. Yin, Y. Liu, C. Shen, and Y. Yan. Enforcing geometric constraints of virtual normal for depth prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5684–5693, 2019.

[38] Y. Zhao, S. Kong, D. Shin, and C. Fowlkes. Domain decluttering: Simplifying images to mitigate synthetic-real domain shift and improve depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3330–3340, 2020.

[39] S. Zhu, G. Brazil, and X. Liu. The edge of depth: Explicit constraints between segmentation and depth. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13116–13125, 2020.