EUROGRAPHICS 2015 / H. Carr, K.-L. Ma, and G. Santucci
(Guest Editors)

Volume 34 (2015), Number 3

Compressive Volume Rendering

Xiaoyang Liu and Usman R. Alim

University of Calgary, Calgary AB T2N 1N4, Canada

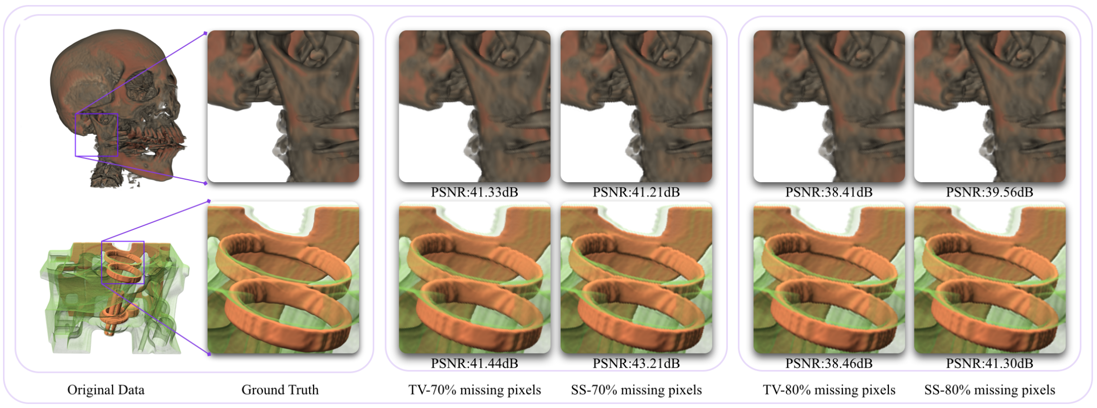

Figure 1: *Compressive Volume Rendering exploits image smoothness to recover images from a small number of rendered pixels. We present two non-adaptive methods that can achieve high quality recovery with as few as 20% of the pixels.*

Abstract

Compressive rendering refers to the process of reconstructing a full image from a small subset of the rendered pixels, thereby expediting the rendering task. In this paper, we empirically investigate three image order techniques for compressive rendering that are suitable for direct volume rendering. The first technique is based on the theory of compressed sensing and leverages the sparsity of the image gradient in the Fourier domain. The latter techniques exploit smoothness properties of the rendered image; the second technique recovers the missing pixels via a total variation minimization procedure while the third technique incorporates a smoothness prior in a variational reconstruction framework employing interpolating cubic B-splines. We compare and contrast the three techniques in terms of quality, efficiency and sensitivity to the distribution of pixels. Our results show that smoothness-based techniques significantly outperform techniques that are based on compressed sensing and are also robust in the presence of highly incomplete information. We achieve high quality recovery with as little as 20% of the pixels distributed uniformly in screen space.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

THe need to to efficiently process and analyze data sets is recognized by various disciplines. As large screen and high density displays are becoming commonplace, we are faced with the additional challenge of adapting our visualization algorithms so that they can produce better quality large size images. In the context of a ray-casting based approach to volume visualization, the total rendering time is usually proportional to the number of pixels in the rendered image which in turn depends on the number of rays cast per

© 2015 The Author(s)
Computer Graphics Forum © 2015 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

pixel. Sophisticated illumination effects, high-quality data filtering and out-of-core data management techniques make the process even more expensive. It is therefore imperative to employ acceleration strategies that reduce the overall cost while maintaining image quality.

In this paper, we explore an image-order acceleration technique that is inspired by the fact that volume rendered images are typically spatially slowly varying and lack sharp fluctuations such as textures. Owing to their inherent smoothness, rendering every single pixel in an image is computationally expensive and wasteful. An alternative is to render a subset of the pixels and infer the missing pixels by leveraging the sparsity of the image in a transform domain such as the Fourier or wavelet domain. This is highly desirable since it not only saves many costly ray-integration operations, it also achieves compression. Initially proposed by Sen and Darabi [SD11], this rendering approach is known as *compressive rendering* and is an application of the theory of compressed sensing (CS) [EK12]. Sen and Darabi [SD11] reported promising recovery results with 75% of the pixel samples.

We build upon the idea of compressive rendering specifically in the context of volume rendering, where we postulate that volume rendered images, owing to their inherent smoothness, should be recoverable from a much smaller fraction of the pixels. Towards this end, we explore recent techniques inspired from the image-processing community. While these techniques can be viewed from different angles (image restoration or completion, compression etc.), our interest is in studying them from the point of view of compressive volume rendering. It is unclear which method is the most suitable for this purpose, specially when the fraction of rendered pixels is very small as compared to the total number of pixels. This paper aims to definitively answer this question so as to guide researchers and practitioners in designing more efficient volume rendering strategies.

Our first technique (Sec. 3.1.3) is an extension of the idea of Sen and Darabi [SD11]. Instead of rendering the original image, we render a subset of the gradient of the image and exploit the sparsity of the discrete Fourier transform of the gradient components to recover a gradient image that is most consistent with the measurements. We then use a Poisson solver to recover the original image. Unlike the approach of Sen and Darabi [SD11], this approach does not suffer from coherence issues since the Fourier basis is inherently incoherent with the canonical sampling basis. Moreover, the gradient components of a volume rendered image are typically more sparse in the Fourier domain as compared to the image itself. Our results show an improvement over the results of Sen and Darabi [SD11]. However, both methods quickly deteriorate when the fraction of missing pixels is high.

In order to guarantee robustness to highly incomplete information, we explore methods that are fundamentally different from CS and employ smoothness instead of sparsity priors. This is a more natural setting for this problem as smoothness is a key property of volume rendered images. Towards this end, we explore two methods namely total variation (TV) minimization (Sec. 3.2) and, variational minimization of a smoothness norm in a spline space (Sec. 3.3). Both of these methods predate CS but have not been explored in the context of compressive rendering. As the name suggests, TV minimization attempts to minimize the total variation norm that is intimately associated with the smoothness of an image [NW13]. The latter method seeks to find a smooth solution in a shift-invariant spline space (SS). This is achieved by minimizing a least-squares type norm that penalizes non-smooth solutions [XAE12]. We show that these smoothness inspired methods are much more resilient to missing information as compared to the CS-based methods (Fig. 1). Moreover, they also have a performance advantage since the minimization process is less expensive as compared to the pursuit strategies typically employed in CS.

The remainder of the paper is organized as follows. After reviewing relevant prior art (Sec. 2), we present a detailed description of our proposed methods (Sec. 3). These methods are then thoroughly compared and contrasted in terms of image quality, sensitivity to the distribution of pixels, and performance (Sec. 4). Even though we focus on image quality in the volume rendering setting, we show that smoothness is also relevant in the general rendering context and leads to significantly better recovery as compared to the state of the art (Sec. 4.5).

2. Related Work

It is important to stress that our approach is fundamentally an image-order acceleration approach and can be used in conjunction with the plethora of object-order acceleration techniques that are available. GPU-based ray-casting is the state-of-the art approach to volume rendering [HKRs*06]. Recent efforts have focused on data management techniques that handle large datasets [BHP14], as well as compression techniques for volumetric data [RGG*12].

In the broader context of Monte-Carlo rendering, adaptive image-order techniques have recently been used to reduce the number of rays traced [RKZ12]. These techniques usually employ multiple passes, incrementally improving image quality. In contrast, compressive rendering is a non-adaptive approach that only traces rays through a subset of the pixels. It can however be extended to a second pass where the placement of pixels is based on the results of the previous pass [SD11]. Our focus however, is on the non-adaptive first pass where we are interested in investigating the effect of the initial non-adaptive distribution of pixels on image quality.

Transform domain strategies are no stranger to graphics and visualization. There are a multitude of approaches that exploit compressibility in well-known transform domains such as the discrete cosine transform (DCT) and the discrete wavelet transform (DWT). Interest in exploiting transform domain sparsity is gaining momentum and started with applications of compressed sensing to light transport [PML*09, SD09]. More recent works have explored these

Method	Property
CS-Wavelet	Based on compressed sensing, assumes **x** is sparse in the wavelet domain [SD11].
CS-Gradient	Based on compressed sensing, leverages the sparsity of the gradient of **x** in the Fourier domain.
TV	Assumes that the **x** exhibits low total variation.
SS	Incorporates a smoothness norm based on the second derivatives of **x**.

Table 1: *Summary of different methods*

ideas in the context of sparsely representing volumetric datasets [WAG*12, GIGM12, XSE14].

There are some other lesser known transforms such as shearlets [GL07] and curvelets [MP10] that are better able to describe anisotropic features such as edges. However, in order to use these in a compressive sensing framework, one needs to employ a sampling basis that is incoherent with these transform basis. Since compressive rendering makes pixel measurements (corresponding to the canonical basis), the choice of transform domain is constrained to the discrete Fourier or cosine transforms.

Besides the approaches presented in this paper, there are other recent approaches to missing data recovery that are not considered in this paper and are a subject of future work. Examples include dictionary learning [Ela10], matrix completion [CR09], and tensor completion [LMWY13]. Some classical approaches to the problem of missing data recovery include radial basis functions [Buh00] and inpainting [BSCB00]. These have already been explored previously in the context of compressive rendering [SD11] and are not considered here.

3. Recovery Methods

Let $\mathbf{x}$ denote the rendered image that is W pixels wide and H pixels high. For convenience, we treat $\mathbf{x}$ as an $N \times 1$ column vector, i.e. $\mathbf{x} = [x_1 \cdots x_N]^T$ where $N = WH$. We also restrict attention to scalar-valued images with the assumption that RGB images can be treated in an independent component-wise manner. Instead of rendering all the pixels in $\mathbf{x}$, we are interested in rendering a small subset of the pixels. The rendered pixels are given by

$$\mathbf{y} = \mathbf{S}\mathbf{x}, \tag{1}$$

where $\mathbf{y} = [y_1 \cdots y_M]^T$ is an $M \times 1$ (typically $M \ll N$) column vector and $\mathbf{S}$ is a $M \times N$ binary sampling matrix. Each row of $\mathbf{S}$ is zero everywhere except for the pixel location that is to be retained. The recovery goal is then to estimate the full image $\mathbf{x}$ from the rendered pixels $\mathbf{y}$. Since the number of rendered pixels is much smaller than the total size of the image, this problem is inherently ill-posed. Some prior assumption about $\mathbf{x}$ needs to be incorporated in order to make the recovery process work. Table 1 summarizes the priors used in the methods presented in this paper.

3.1. Methods Based on CS

This approach is similar to the work of Sen and Darabi [SD11]. For the sake of comparison and completeness, we review briefly the theory of compressed sensing before proposing our solution that exploits sparsity of the gradient components in the Fourier domain. More details on compressed sensing can be found in the recent textbook [EK12].

3.1.1. CS Background

Let $\hat{\mathbf{x}} \in \mathbb{R}^N$ be a sparse vector, i.e. it has a few non-zero entries. Formally, sparsity is quantified by the ℓ_0-norm $||\cdot||_0$, that counts the number of non-zero entries. A vector $\hat{\mathbf{x}}$ is said to be k-sparse if $||\hat{\mathbf{x}}||_0 \leq k$. The sensing mechanism is modelled as a set of linear measurements that yield the vector $\mathbf{y} \in \mathbb{R}^M$. In particular,

$$\mathbf{y} = \mathbf{A}\hat{\mathbf{x}}, \tag{2}$$

where $\mathbf{A}$ is an $M \times N$ sensing matrix with $M \ll N$. Even though this system is underdetermined, it can be solved uniquely using compressed sensing as long as $\mathbf{A}$ meets the Restricted Isometry Condition (RIC):

$$(1-\delta)\,||\hat{\mathbf{x}}||_2^2 \leq ||\mathbf{A}\hat{\mathbf{x}}||_2^2 \leq (1+\delta)\,||\hat{\mathbf{x}}||_2^2, \tag{3}$$

where $\delta \in (0,1)$ and $||\cdot||_2$ indicates the ℓ_2-norm. Intuitively, the RIC states that in a valid sensing measurement matrix $\mathbf{A}$, every possible set of k columns form an approximate orthogonal set. Matrices that have been proven (probabilistically) to meet RIC include partial Fourier or cosine matrices (randomly selected rows from the full discrete Fourier or cosine transform matrix), Gaussian and Bernoulli random matrices [CT06]. Under the condition of RIC, the corresponding recovery mechanism becomes non-linear and can be formulated as the optimization problem

$$\min ||\hat{\mathbf{x}}||_0 \quad \text{subject to} \quad \mathbf{A}\hat{\mathbf{x}} = \mathbf{y}. \tag{4}$$

This is an NP-hard problem. In practice, it is substituted by an ℓ_1-minimization problem which is solved using a pursuit algorithm [TG07]. Provided that the RIC is satisfied, recovery accuracy depends on the sparsity of $\hat{\mathbf{x}}$; various error bounds relating δ and k have been explored [EK12].

3.1.2. CS for Rendered Images (CS-Wavelet)

Usually, signals of interest such as rendered images are not inherently sparse, but are sparse in a transform domain such as the DWT or the DCT. In other words, the transformed representation $\hat{\mathbf{x}} = \Psi\mathbf{x}$ approximates the original image well with a few non-zero coefficients. Here, Ψ is an $N \times N$ orthonormal matrix corresponding to a compression basis. Substituting this transform domain representation into 1, we obtain the measurement equation

$$\mathbf{y} = \mathbf{S}\Psi^{-1}\hat{\mathbf{x}}. \tag{5}$$

Letting $\mathbf{A} = \mathbf{S}\Psi^{-1}$, it is clear that this equation corresponds to the sensing equation 2, and recovery is possible as long as the matrix $\mathbf{S}\Psi^{-1}$ satisfies the RIC. Verifying the RIC is

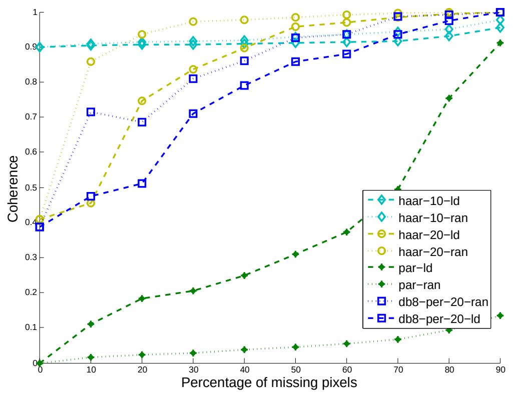

Figure 2: *Coherence results for different sensing matrices ($N = 64^2$). The numbers indicate the standard deviation of the Gaussian blurring filter in the Fourier domain, lower values indicate greater blurring. The other abbreviations are: ld - low discrepancy, ran - random, and par - partial Fourier.*

computationally difficult and a useful related notion is that of coherence. The coherence of a sensing matrix $\mathbf{A}$, $\mu(\mathbf{A})$, is the largest absolute inner-product between any two columns $\mathbf{a}_i$ and $\mathbf{a}_j$:

$$\mu(\mathbf{A}) = \max_{1 \le i < j \le n} \frac{|\mathbf{a}_i^T \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}. \quad (6)$$

Intuitively, the lower the coherence, the better the sparse recovery via ℓ_1 minimization. When $M \ll N$, the coherence is lower bounded according to $\mu(\mathbf{A}) \ge 1/\sqrt{M}$.

Another way to look at coherence is in terms of the sampling matrix $\mathbf{S}$ and the compression matrix Ψ^{-1}. In order to guarantee the RIC, the two must be incoherent. The work of Sen and Darabi [SD11] exploits sparsity of the image in the wavelet domain. Unfortunately, the wavelet domain is *not* incoherent with point sampling. To improve incoherence, they recover a blurred version $\mathbf{x}_b$ of the original image $\mathbf{x}$, where $\mathbf{x}_b = \Phi\mathbf{x}$, and Φ is an $N \times N$ matrix corresponding to the Gaussian blurring filter. Their sensing mechanism can be written as

$$\mathbf{y} = \underbrace{\mathbf{S}\Phi^{-1}\mathbf{W}^{-1}}_{\mathbf{A}} \hat{\mathbf{x}}_b, \quad (7)$$

where $\mathbf{W}^{-1}$ is the inverse DWT matrix, and $\hat{\mathbf{x}}_b$ is the DWT of the blurred image $\mathbf{x}_b$. From the recovered coefficients $\hat{\mathbf{x}}_b$, the final image is obtained according to $\mathbf{x} = \Phi^{-1}\mathbf{W}^{-1}\hat{\mathbf{x}}_b$. Even though, the blurring operation improves the incoherence somewhat, successful recovery is sensitive to the variance of the blurring filter Φ which needs to be adjusted on a case-by-case basis. As our tests show, this method is also very sensitive to the distribution of pixels and breaks down when the fraction of missing pixels is high.

3.1.3. Gradient Recovery via CS (CS-Gradient)

In order to ameliorate coherence problems, we choose to exploit sparsity in the discrete Fourier transform (DFT) domain. The RIC property of random partial Fourier matrices is well-known. In other words, the Fourier domain is inherently incoherent with point sampling measurements. Fig. 2 compares the coherence of the sensing matrices in the CS-Wavelet method for a number of wavelet types as a function of the fraction of missing pixels. Observe that, for all wavelet types, coherence becomes higher as the fraction of missing pixels increases, and the blurring filter only improves incoherence slightly. On the other hand, random partial Fourier matrices exhibit very low coherence.

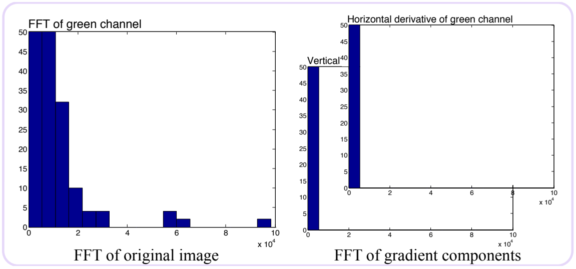

Figure 3: *Histogram of the absolute values of the DFT coefficients of the engine image (left) and its derivatives (right). The image and the derivatives were normalized to lie in the range $[0,1]$ before applying the FFT.*

Rendered images are usually more sparse in the DWT domain as compared to the DFT domain [SD11]. In order to improve sparsity in the DFT domain, we can recover the image gradient rather than the image itself. For images that are slowly varying, we expect that the gradient components are more sparse in the DFT domain as compared to the image itself (Fig. 3). Observe that rendered images are also sparse in the gradient domain itself, and therefore, one can exploit sparsity in the gradient domain. However, the theory of CS dictates that one would need to make point measurements (of the image gradient components) in the DFT domain. This is suitable for applications such as MRI [PMGC12], but cannot be realized in our case as the rendering process (barring applications such as frequency domain volume rendering [TL93]) typically makes pixel measurements.

Thus, instead of rendering the image $\mathbf{x}$ directly, we render the discrete gradient components $\mathbf{x}_1$ and $\mathbf{x}_2$. This can be done by representing $\mathbf{x}$ in a basis spanned by a tensor product (pixel reconstruction) kernel such as the bilinear B-spline or the Mitchell-Netravali cubic kernel [PH10]. The gradient can then be obtained by differentiating the kernel, i.e. instead of weighting the incoming rays with the pixel reconstruction filter, we can simply weight them according to the derivative of the kernel. Alternatively, one can use a box filter for rendering and apply a finite differencing scheme to obtain the gradient. Let $\mathbf{y}_1$ be an $m \times 1$ column vector that contains the horizontal component of the gradient measured at m different locations according to the sampling matrix $\mathbf{S}$. Our sensing mechanism can be formulated as

$$\mathbf{y}_1 = \mathbf{S}\mathbf{F}^{-1}\hat{\mathbf{x}}_1, \quad (8)$$

where $\mathbf{F}$ is the 2D DFT matrix, and $\hat{\mathbf{x}}_1$ denotes the 2D DFT of the horizontal gradient component $\mathbf{x}_1$. Observe that $\mathbf{A} = \mathbf{S}\mathbf{F}^{-1}$ is a partial Fourier matrix provided that the location of the pixels is random. An approximation of $\hat{\mathbf{x}}_1$ is obtained via the following ℓ_1 minimization:

$$\min \|\hat{\mathbf{x}}_1\|_1 \quad \text{subject to} \quad \|\mathbf{S}\mathbf{F}^{-1}\hat{\mathbf{x}}_1 - \mathbf{y}_1\|_2 \le \varepsilon. \tag{9}$$

An approximation of the DFT of the vertical component, $\hat{\mathbf{x}}_2$, is obtained similarly. Computing the pixel values from gradient components is a problem that frequently arises in gradient-domain image processing applications. We follow the approach of Pérez *et al.* [PGB03] and apply a Poisson solver—with Dirichlet boundary conditions—to the divergence of the gradient.

3.2. Recovery via TV minimization (TV)

The total variation seminorm of an image is defined as:

$$\|\mathbf{x}\|_{TV} := \sum_{i,j} |x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|, \tag{10}$$

where $x_{i,j}$ — with slight abuse of notation — is the pixel value corresponding to the pixel with image-space coordinates (i,j). In other words, it is the sum of the ℓ_1 norms of the horizontal and vertical components of the gradient obtained via forward differencing. First proposed by Rudin *et. al* [ROF92], the TV-norm is low for images that are smooth and high for images that have high variability. It is also connected with sparsity; images that are sparse in the gradient domain also exhibit low TV. This property has been empirically known for some time and has been used in several applications such as denoising, inpainting, and recovery from partial Fourier measurements (see e.g [CEPY05]). The precise theoretical connection with CS has only recently been established [NW13].

Despite its success in image processing applications, TV minimization (or regularization) has not been used in the compressive rendering context which, at its core, is an image restoration problem akin to inpainting. Our goal here is to investigate how well this method performs as compared to the CS-based methods described earlier. The precise minimization problem that we wish to solve is given by

$$\min \|\mathbf{x}\|_{TV} \quad \text{subject to} \quad \|\mathbf{S}\mathbf{x} - \mathbf{y}\|_2 \le \varepsilon, \tag{11}$$

where $\mathbf{S}$ and $\mathbf{y}$ are as described in 1. This is a well-studied minimization problem and fast algorithms have recently gained popularity (e.g. split Bergman [GO09] or NESTA [BBC11]).

3.3. Recovery in a Splines Space (SS)

The third method we consider can be regarded as a scattered data approximation problem [XAE12]. It attempts to find a smooth solution in a prescribed space that is spanned by the uniform shifts of a kernel function $\varphi(\mathbf{t})$ where $\mathbf{t} \in \mathbb{R}^2$.

Let $\mathbf{j}_1, \dots, \mathbf{j}_N$ denote the pixel locations corresponding to the pixel values in $\mathbf{x}$, and let $\mathbf{k}_1, \dots, \mathbf{k}_M$ denote the pixel locations corresponding to the pixel values in $\mathbf{y}$. The goal is to find the coefficients $\mathbf{c} = [c_1 \cdots c_N]^T$ of the approximation:

$$f(\mathbf{t}) := \sum_{n=1}^{N} c_n \varphi(\mathbf{t} - \mathbf{j}_n), \tag{12}$$

such that the approximation closely matches the measured pixel values, i.e. $f(\mathbf{k}_m) \approx y_m$ for $m = 1, \dots, M$. Additionally, it is desired that the function $f(\mathbf{t})$ be smooth. A useful notion of smoothness is provided by the second-order Beppo-Levi seminorm. For functions $g(\mathbf{t})$ and $h(\mathbf{t})$, the second-order Beppo-Levi inner-product is defined as

$$\langle g,h\rangle_{BL_2} := \langle \partial_{t_1t_1}g, \partial_{t_1t_1}h\rangle + 2\langle \partial_{t_1t_2}g, \partial_{t_1t_2}h\rangle + \langle \partial_{t_2t_2}g, \partial_{t_2t_2}h\rangle \tag{13}$$

where $\langle \cdot,\cdot\rangle$ denotes the standard L_2 inner-product. The BL_2 inner-product induces a seminorm which we denote as $\|g\|^2_{BL_2} := \langle g,g\rangle_{BL_2}$. In contrast to the TV seminorm introduced earlier, the BL_2 seminorm is defined in the continuous domain and measures smoothness via the second-order derivatives. Smooth functions have a low BL_2 norm and vice-versa. Our minimization problem in this setting can now be formulated as

$$\min_f \quad \lambda\|f\|^2_{BL_2} + \sum_{m=1}^{M} (f(\mathbf{k}_m) - y_m)^2, \tag{14}$$

where the minimization is to be carried out over all functions f that are of the form (12). The first term in the above equation measures the smoothness of the solution $f(\mathbf{t})$ by the energy present in all of its second derivatives, and the second term is a fidelity term that attempts to fit the function $f(\mathbf{t})$ to the available data.

In order to solve this minimization problem, we need to choose a kernel function $\varphi(\mathbf{t})$. There are many choices available such as the bilinear or bicubic B-splines etc. In order to ensure consistency with the other recovery methods, and to obtain good quality approximations, we choose the optimal interpolating cubic B-spline proposed by Blu *et al.* [BTU01]. It is defined as

$$\beta^3_I(t) := \beta^3(t) - \frac{1}{6}\frac{d^2}{dt^2}\beta^3(t), \tag{15}$$

where $\beta^3(t)$ denotes the univariate uniform centered cubic B-spline. The corresponding bivariate kernel is obtained via a tensor product, i.e. $\varphi(t_1,t_2) = \beta^3_I(t_1)\beta^3_I(t_2)$. Observe that, with this choice of φ, the coefficient vector $\mathbf{c}$ is the same as the vector $\mathbf{x}$ (since the kernel is interpolating), and our minimization problem can be written equivalently (see [XAE12] for details) as

$$\min_{\mathbf{x}} \|\mathbf{S}\mathbf{x} - \mathbf{y}\|^2_2 + \lambda \mathbf{x}^T\mathbf{H}\mathbf{x}, \tag{16}$$

where the $N \times N$ matrix $\mathbf{H}$ is defined as

$$\mathbf{H}_{p,q} = \langle \varphi(\cdot - \mathbf{j}_p), \varphi(\cdot - \mathbf{j}_q)\rangle_{BL_2}. \tag{17}$$

Since (16) involves an ℓ_2 norm, we can differentiate with respect to $\mathbf{x}$ to obtain the following least-squares problem

$$(\mathbf{S}^T\mathbf{S}+\lambda\mathbf{H})\mathbf{x}=\mathbf{S}^T\mathbf{y}, \tag{18}$$

whose solution yields the minimizer of (16). This least-squares problem can be efficiently solved using the conjugate gradient method since the matrix $(\mathbf{S}^T\mathbf{S}+\lambda\mathbf{H})$ is symmetric and positive definite. The matrix $\mathbf{H}$ does not need to be explicitly computed or stored since its action on a vector $\mathbf{x}$ is equivalent to a filtering operation; the filter weights are obtained via (17).

4. Results and Discussion

We generated volume rendered test images from datasets using our own volume renderer as well as Paraview. All of the test images were generated at a resolution of 1200×1200 on a workstation with a quad-core, 3.4 Ghz Intel®Core$^{\text{TM}}$i7-3770 CPU with 16GB RAM. All of our recovery experiments were conducted in Matlab where we used the NESTA solver [BBC11] to carry out ℓ_1 (CS-Wavelet, CS-Gradient) and TV minimization. We used Matlab's conjugate gradient solver to solve the least-squares minimization (SS) problem (16). For parameter settings, we used 20 for the standard deviation of the blurring filter Φ for the CS-Wavelet approach which, according to the results of Sen and Darabi [SD11], balances the tradeoff between incoherence and blurring effects. We used the same value of 10^{-2} for the tolerance and stopping criteria of the NESTA solver in our CS-Wavelet, CS-Gradient and TV experiments. This value was empirically chosen to provide a good tradeoff between recovery quality and runtime. For the least-squares solver (SS), we used the value 10^{-2} for the regularization parameter λ as suggested by Xu *et. al* [XAE12].

We recovered the images from a fraction of the pixels. We experimented with different percentages of pixels that are removed via two different pixel distribution algorithms explained in the following section. To measure recovery quality, we computed the peek signal-to-noise ratio (PSNR) values measured in decibels (dB). The PSNR is a well-known quality metric and is a good way to quantify large differences in recovery trends exhibited by the different techniques. We also computed error images in the CIELUV colorspace according to the work of Ljung *et al.* [LLYM04]. In addition to measuring the recovery quality, we also measured the performance of each method with respect to the timing for recovery.

4.1. Choice of Distribution Algorithm

Choosing a pixel distribution algorithm wisely is of significant importance as our goal is to recover volume rendered images from a small fraction of the pixels. A straightforward way of choosing pixels is the random distribution, obtained by randomly drawing a number between 1 and N where N is the total number of pixels. This strategy however leads to inhomogeneous regions (Fig. 4). A better strategy is to distribute the pixels as uniformly as possible so

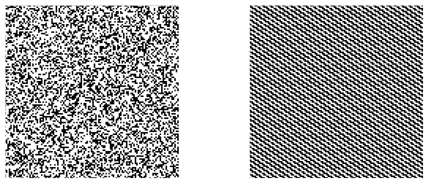

Figure 4: *Masks with 50% missing pixels; left: random, and right: LD via pixel shuffle.*

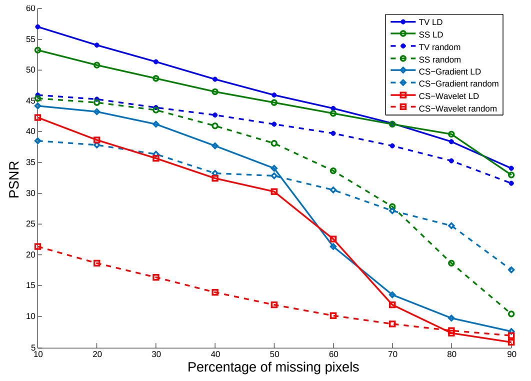

Figure 5: *Random vs. LD distributions.*

that the overall discrepancy [PH10] is low. A potential distribution that achieves low discrepancy (LD) can be obtained by the Poisson disk sampling algorithm [PH10]. This algorithm achieves very good blue-noise distributions. However, a disadvantage is that it is not progressive, i.e. a distribution with a high percentage of coverage does not contain a distribution with a low percentage of coverage. This is an important property for compressive rendering as it allows for the progressive update of an image. A distribution that does satisfy this property is provided by Anderson's pixel shuffle algorithm [And93]. This algorithm is based on the Fibonacci numbers, and attempts to fill in the biggest gaps in the distribution to maintain a low overall discrepancy (Fig. 4).

4.2. Random Distribution vs. Pixel Shuffle

We used the two aforementioned distribution algorithms to compare the recovery quality of all the methods. The quantitative results for the head dataset are shown in Fig. 5, and some of the qualitative results are shown in Fig. 6. We can observe that our CS-Gradient method produces slightly better results compared to the CS-Wavelet method. The CS-Wavelet method seems to be highly sensitive to the distribution of pixels. In our tests, we observed that, when the percentage of missing pixels is high, the random distribution leads to strong speckling artefacts. The LD distribution achieves a lower coherence (Fig. 2) and thefore yields better results. However, it also exhibits directional artefacts (Fig. 6: second column). In comparison, our CS-Gradient method fares much better (Fig. 5). It favours the random distribution when the fraction of missing pixels is high. This is to be expected as the random distribution leads to partial Fourier

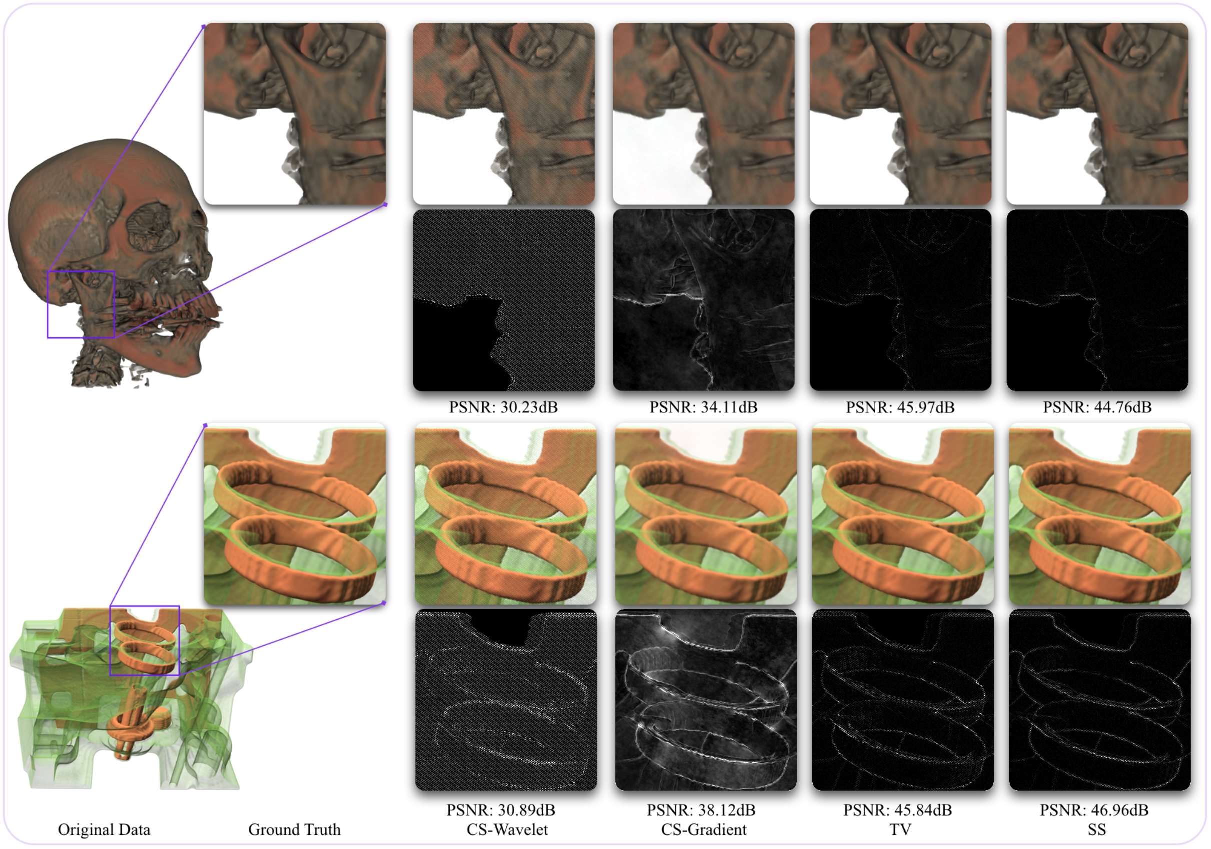

Figure 6: *Recovery results for all methods with LD distribution (50% missing pixels). The grayscale images indicate the magnitude of the error computed in the CIELUV space; the error range range* $[0,20]$ *is linearly mapped to a grayscale colormap.*

matrices with better incoherence when the fraction of missing pixels is high (Fig. 2). Additionally, we also did not notice any objectionable artefacts with this method (Fig. 6: third column). However, notice that both CS-Wavelet and CS-Gradient quickly deteriorate when the fraction of missing pixels is high.

The smoothness-based methods (TV and SS) significantly outperform the CS-based methods. Both seem to favour the LD distribution over the random distribution, the differences are indeed quite stark (Fig.5). In terms of reconstruction quality, the two methods, in conjunction with the LD distribution are quite comparable when the fraction of missing pixels is high (Fig.1 and Fig. 6). The TV method seems to have an edge over SS when the fraction of missing pixels is low. In the following results, we focus on the smoothness based methods and compare them with CS-Wavelet in combination with the LD distribution.

4.3. Sensitivity to Image Content

To compare the sensitivity of the algorithms to images with different smoothness characteristics, we used Paraview to render DVR images of the foot dataset and isosurface images of the aneurysm dataset. Some qualitative results are shown in Fig. 8. In terms of recovery quality, TV and SS produce similar results; SS recovery is somewhat sharper in the boundary regions while the TV method seems to preserve edges better. From Fig. 8, we can see that our TV and SS methods can produce consistent recovery results over different fractions of missing pixels. The higher PSNR values for the foot images corroborate the fact that these methods perform much better when the image content is slowly varying. The results for the aneurysm dataset show some degradation when the fraction of missing pixels is high. However, it is not as severe as it is for the CS-based methods.

In order to investigate the importance of the smoothness of the image, we experimented with a synthetic texture image that has both high and low frequencies alike (Fig. 7). Both TV and SS methods cannot produce acceptable recovery even with 10% missing pixels. As summarized in Table 3, this image lacks smoothness which is a key assumption made by all the recovery algorithms.

4.4. Performance Aspects

The reconstruction time for different fractions of missing pixels is shown in Fig. 9. We reckon that our SS algorithm outperforms all the other methods. This is due to the fact that

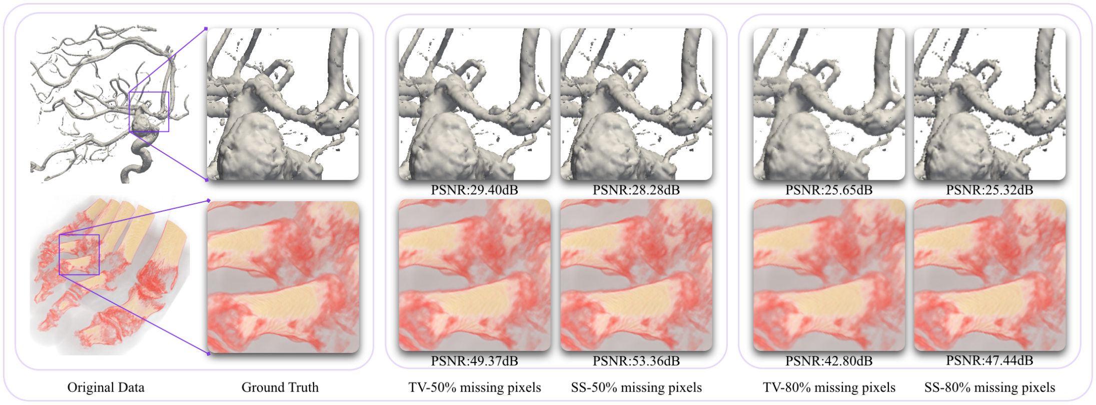

Figure 8: *Recovery results for foot and aneurysm data via TV and SS*

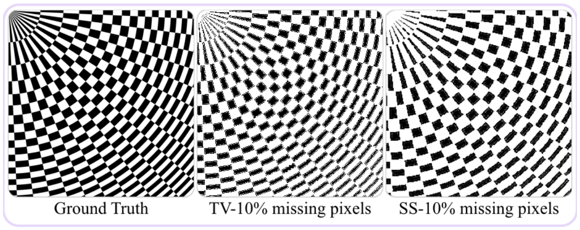

Figure 7: *Recovery results for high frequency texture image.*

	Head	Engine
600×600	1754.92s	1093.61s
900×900	3955.13s	2442.28s
1200×1200	7067.32s	4249.44s

Table 2: *Rendering timing for different resolutions.*

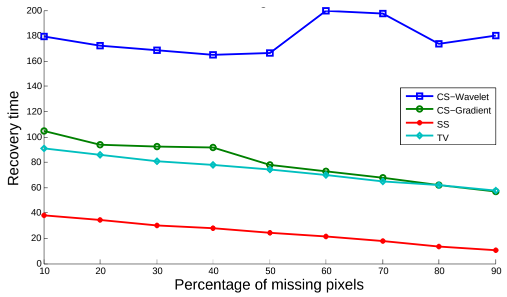

Figure 9: *Timing results for head data via different recovery methods (*1200×1200 *images).*

SS utilizes a least-squares solver which is more efficient as compared to ℓ_1 and TV minimization. CS-Gradients and TV minimization exhibit comparable performance trends while CS-Wavelet lags behind by a wide margin due to the fact that it needs to evaluate the forward and inverse DWT at every iteration which is slower as compared to the FFT.

It should be stressed that these performance results only compare the trends shown by the different algorithms for image recovery. The overall cost of the rendering operation is the sum of the time spent casting rays and the time it takes to recover the full image from the rendered subset. Since rendering time can vary greatly depending on the quality of the renderer, compressive volume rendering only makes sense when the time spent tracing rays is much greater than the image recovery time. As an example, Table 2 shows the total render time for the head and engine datasets using our single-threaded software renderer that uses high-quality tricubic interpolation for both the scalar data and the gradient. Rendering one-fourth of the image pixels cuts down the rendering time by several thousand seconds. The recovery time is only a tiny fraction of this. At the very least, compressive rendering is a viable strategy to accelerate offline volume rendering tasks.

In an interactive environment, one would need to consider additional factors such as the parallelizability of these recovery methods, and their implementation on the GPU. The work of Zach *et al.* [ZPB07] presents a GPU implementation of TV-L1 minimization; they demonstrate real-time optical flow calculation with 30 FPS for a resolution of 320×240 pixels. However, they used an iterative conjugate gradient solver to do the TV-L1 minimization whose convergence is not as fast as compared to methods such as NESTA [BBC11]. Least-squares linear systems appear in various problems in Computer Graphics and have a long histroy of successful GPU acceleration [PF05]. For the SS method, significant speed up can be obtained as the matrix operations in equation 18 can be expressed as fast image filtering operations.

Our focus in this work is on investigating the theoretical aspects of the recovery algorithms in the context of volume rendering. With a careful consideration of the inherent parallelizability of these methods, we anticipate significant gains in terms of efficiency or quality. For instance, one could employ a higher quality renderer and only trace a fraction of the rays to achieve a better overall image for the same computational cost. It is also possible to improve reconstruction quality in a two-pass rendering scheme where the first pass recovers a low quality image that can be used to adaptively trace rays according to the content of the image.

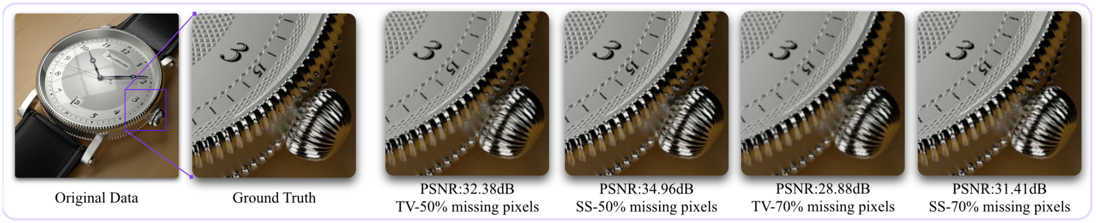

Figure 11: *Recovery results via TV and SS*

Image	Category	TV/SS with LD
Engine, Head and Foot (Fig. 6, Fig. 8)	DVR: very smooth image.	Consistent recovery results over different fractions of missing pixels.
Watch (Fig. 11)	Physically-based rendering: smooth image, very few small scale details.	Good recovery results even with high fractions of missing pixels.
Aneurysm (Fig. 8)	ISR: not so smooth, some small scale details and sharp edges.	Acceptable recovery with about 50% missing pixels but some degradation when the fraction of missing pixels is high.
Synthetic (Fig. 7)	Non smooth: lots of small scale details and edges.	Unacceptable recovery even with a very small fraction of missing pixels.

Table 3: *Performance summary of TV and SS for different types of images.*

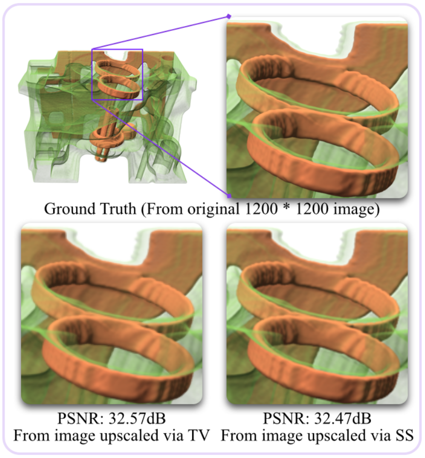
Ground Truth (From original 1200 * 1200 image)
PSNR: 32.57dB
From image upscaled via TV
PSNR: 32.47dB
From image upscaled via SS

Figure 10: *Upscaling results via TV and SS*

4.5. Other Applications

The methods presented in this paper can also be applied to ray-traced images. We tested the recovery results on the watch image generated using the physically-based renderer LuxRender. The original image took several hours to render. Fig. 11 shows the comparisons between different algorithms. As expected, our TV and SS methods can produce good recovery results even with 50% missing pixels. For 70% missing pixels, the results are acceptable but do exhibit subtle artefacts.

We can also use these methods for image up-scaling, i.e. we can render at one resolution but recover at a higher resolution (also known as super resolution). In this case, we generated an image with higher resolution from a full low resolution image. The pixels from the low resolution image are mapped to the high-resolution image and the missing pixels are recovered. Fig. 10 demonstrate the results for this application. The images shown in the bottom row were upscaled from a low resolution 600×600 image, which is equivalent to the original formulation with 75% missing pixels. The results are slightly worse than the case of image recovery at the same resolution (see Fig. 1), but the overall quality remains acceptable.

5. Conclusion

We presented three different methods for recovering images from a subset of the pixels. Our results show that the CS-based approaches are not suitable for this problem as we are restricted to making pixel measurements. The previously proposed approach (CS-Wavelet) exhibits strong artefacts when the fraction of missing pixels in high. Our attempt to improve the method (CS-Gradient) only shows slight improvements. On the other hand, smoothness-based methods (TV and SS) are much more suitable for this task and are able to recover images successfully even when the fraction of missing pixels is high. A summary of the recovery results based on the smoothness-inspired methods is shown in Table 3. For performance considerations, we advocate the SS method as it is based on a least-squares solver. In future, besides exploring the matrix and tensor completion methods mentioned earlier, we are also interested in further comparing and contrasting the smoothness-based methods in an interactive volume rendering environment, both in terms of performance and in terms of quality with respect to perceptual metrics.

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive feedback. This work was supported by the Faculty of Science, University of Calgary.

References

[And93] ANDERSON P. G.: Linear pixel shuffling for image processing: an introduction. *Journal of Electronic Imaging 2*, 2 (1993), 147–154. 6

[BBC11] BECKER S., BOBIN J., CANDÈS E. J.: NESTA: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences 4*, 1 (2011), 1–39. 5, 6, 8

[BHP14] BEYER J., HADWIGER M., PFISTER H.: A survey of GPU-based large-scale volume visualization. In *EuroVis 2014 - State of the Art Reports* (June 2014), The Eurographics Association. 2

[BSCB00] BERTALMIO M., SAPIRO G., CASELLES V., BALLESTER C.: Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH)* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 417–424. 3

[BTU01] BLU T., TH'EVENAZ P., UNSER M.: MOMS: Maximal-order interpolation of minimal support. *IEEE Transactions on Image Processing 10*, 7 (2001), 1069–1080. 5

[Buh00] BUHMANN M. D.: Radial basis functions. *Acta Numerica 2000 9* (2000), 1–38. 3

[CEPY05] CHAN T., ESEDOGLU S., PARK F., YIP A.: Recent developments in total variation image restoration. In *Mathematical Models of Computer Vision* (2005), Springer Verlag. 5

[CR09] CANDÈS E. J., RECHT B.: Exact matrix completion via convex optimization. *Foundations of Computational mathematics 9*, 6 (2009), 717–772. 3

[CT06] CANDES E. J., TAO T.: Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory 52*, 12 (2006), 5406–5425. 3

[EK12] ELDAR Y., KUTYNIOK G.: *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012. 2, 3

[Ela10] ELAD M.: *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, 2010. 3

[GIGM12] GOBBETTI E., IGLESIAS GUITIÁN J. A., MARTON F.: COVRA: A compression-domain output-sensitive volume rendering architecture based on a sparse representation of voxel blocks. *Computer Graphics Forum 31*, 3pt4 (2012), 1315–1324. 3

[GL07] GUO K., LABATE D.: Optimally sparse multidimensional representation using shearlets. *SIAM Journal on Mathematical Analysis 39*, 1 (2007), 298–318. 3

[GO09] GOLDSTEIN T., OSHER S.: The split bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences 2*, 2 (2009), 323–343. 5

[HKRs*06] HADWIGER M., KNISS J. M., REZK-SALAMA C., WEISKOPF D., ENGEL K.: *Real-time Volume Graphics*. A. K. Peters, Ltd., Natick, MA, USA, 2006. 2

[LLYM04] LJUNG P., LUNDSTROM C., YNNERMAN A., MUSETH K.: Transfer function based adaptive decompression for volume rendering of large medical data sets. In *IEEE Symposium on Volume Visualization and Graphics* (2004), IEEE, pp. 25–32. 6

[LMWY13] LIU J., MUSIALSKI P., WONKA P., YE J.: Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 1 (2013), 208–220. 3

[MP10] MA J., PLONKA G.: The curvelet transform. *IEEE Signal Processing Magazine 27*, 2 (2010), 118–133. 3

[NW13] NEEDELL D., WARD R.: Stable image reconstruction using total variation minimization. *SIAM Journal on Imaging Sciences 6*, 2 (2013), 1035–1058. 2, 5

[PF05] PHARR M., FERNANDO R.: *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*. Addison-Wesley Professional, 2005. 8

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Transactions on Graphics 22*, 3 (2003), 313–318. 5

[PH10] PHARR M., HUMPHREYS G.: *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2010. 4, 6

[PMGC12] PATEL V. M., MALEH R., GILBERT A. C., CHELLAPPA R.: Gradient-based image recovery methods from incomplete fourier measurements. *IEEE Transactions on Image Processing 21*, 1 (2012), 94–105. 4

[PML*09] PEERS P., MAHAJAN D. K., LAMOND B., GHOSH A., MATUSIK W., RAMAMOORTHI R., DEBEVEC P.: Compressive light transport sensing. *ACM Transactions on Graphics (TOG) 28*, 1 (2009), 3. 2

[RGG*12] RODRÍGUEZ M. B., GOBBETTI E., GUITIÁN J. A. I., MAKHINYA M., MARTON F., PAJAROLA R., SUTER S. K.: A survey of compressed GPU-based direct volume rendering. In *Eurographics 2013-State of the Art Reports* (2012), The Eurographics Association, pp. 117–136. 2

[RKZ12] ROUSSELLE F., KNAUS C., ZWICKER M.: Adaptive rendering with non-local means filtering. *ACM Transactions on Graphics 31*, 6 (Nov. 2012), 195:1–195:11. 2

[ROF92] RUDIN L. I., OSHER S., FATEMI E.: Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena 60*, 1 (1992), 259–268. 5

[SD09] SEN P., DARABI S.: Compressive dual photography. *Computer Graphics Forum 28*, 2 (2009), 609–618. 2

[SD11] SEN P., DARABI S.: Compressive rendering: A rendering application of compressed sensing. *IEEE Transactions on Visualization and Computer Graphics 17*, 4 (2011), 487–499. 2, 3, 4, 6

[TG07] TROPP J. A., GILBERT A. C.: Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory 53*, 12 (2007), 4655–4666. 3

[TL93] TOTSUKA T., LEVOY M.: Frequency domain volume rendering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH)* (1993), ACM, pp. 271–278. 4

[WAG*12] WENGER S., AMENT M., GUTHE S., LORENZ D., TILLMANN A., WEISKOPF D., MAGNOR M.: Visualization of astronomical nebulae via distributed multi-GPU compressed sensing tomography. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2188–2197. 3

[XAE12] XU X., ALVARADO A. S., ENTEZARI A.: Reconstruction of irregularly-sampled volumetric data in efficient box spline spaces. *IEEE Transactions on Medical Imaging 31*, 7 (2012), 1472–1480. 2, 5, 6

[XSE14] XU X., SAKHAEE E., ENTEZARI A.: Volumetric data reduction in a compressed sensing framework. *Computer Graphics Forum (EuroVis 2014) 33*, 3 (2014), 111–120. 3

[ZPB07] ZACH C., POCK T., BISCHOF H.: A duality based approach for realtime TV-L1 optical flow. In *Pattern Recognition*. Springer, 2007, pp. 214–223. 8