This work was supported by the Fundação para a Ciência e Tecnologia under the PhD grant PD/BD/113966/2015.

Sparse Reconstruction of Multidimensional Light Transport Signals
César Morais Perdigão
UMinho | 2020

Universidade do Minho
Escola de Engenharia
Departamento de Informática

César Morais Perdigão

Sparse Reconstruction of Multidimensional Light Transport Signals

MAPi Doctoral Programme in Computer Science

November 2020

Universidade do Minho
Escola de Engenharia
Departamento de Informática

César Morais Perdigão

Sparse Reconstruction of Multidimensional Light Transport Signals

MAPi Doctoral Programme in Computer Science

Work supervised by:
Doctor Luís Paulo Peixoto dos Santos
Doctor Thomas Bashford-Rogers

November 2020

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros

Acknowledgements

This work was supported by the Fundação para a Ciência e Tecnologia under the PhD grant PD/BD/113966/2015.

Statement of Integrity

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Sparse Reconstruction of Multidimensional Light Transport Signals

Path space integration rendering algorithms are capable of synthesising high quality images perceptually indistinguishable from the real world. These algorithms require a huge number of light transport paths to be sampled in order to produce realistic imagery. Sampling a huge number of light paths is computationally demanding and methods are needed to reduce this computation time.

The goal of this thesis is to reduce rendering time by reducing the number of sampled light paths and do so by exploiting sparse reconstruction techniques. These techniques allow the recovery of signals even when sampled at unusually low frequencies, in fact, well below the Nyquist limit. This is possible by exploiting some a priori knowledge, namely, the fact that the signal can be sparsely represented in some alternative space, other than the canonical time/space reference frame. Over the last decade sparse reconstruction techniques such as Compressive Sensing and the Sparse Fourier Transform have matured and demonstrated impressive results.

This thesis applies sparse reconstruction techniques to the global light transport problem, including path space integration, in order to reduce the number of samples required to render high quality imagery. The Fourier Space is used as the sparse representation space: high dimensional signals (such as path space) are known to require a reduced number of coefficients to be represented in this space. The main contributions of this thesis are:

- A sparsity analysis methodology for light transport signals, which allows verifying whether the signals being processed exhibit the properties required for successful sparse reconstruction: these include being dense on the sampling basis and sparse on the representation basis.
- A method for efficient integration of multidimensional Fourier signals is proposed. This method treats the integration as a convolution with a box filter and reduces the dimensionality of the problem, calculating the integral values as a lower dimensionality inverse Fourier Transform. This method achieves much lower execution times and memory usage that alternative approaches, both due to the improved algorithmic complexity and the ability to leverage the efficiency of FFT implementations.
- A sparse reconstruction pipeline, using the Sparse Fourier Transform, applied to discrete Path Space in primary sample space was developed. Due to the high sample requirements, however, direct reconstruction of Path Space is not as efficient as classical approaches. These sparse reconstructions can also be used as importance maps, providing a global path guiding mechanism for unbiased Monte Carlo integration, achieving local improvements to caustics rendering.
- A second sparse reconstruction pipeline, based on Matching Pursuit algorithms (CoSaMP, specifically) is also proposed. Since the time complexity of these algorithms is exponential with the number of dimensions, lower dimensionality spaces were used, namely depth of field and indirect lighting, instead of path space. Direct reconstruction of these signals showed no improvements with respect to classical sampling, for the same execution time - note that executing CoSaMP constitutes an additional optimization step that costs a significant fraction of rendering time. An importance sampling approach was applied as well, providing similar local improvements to the previous approach. A new method is also proposed that uses sparse reconstruction results as a distance metric for a bilateral filtering process, which achieves image quality improvements upon classical approaches for complex scenes.

Keywords: Fourier Space, Global Illumination, Monte Carlo Integration, Ray-Tracing, Sparse Reconstruction

Reconstrução Esparsa de Sinais de Transporte de Luz Multidimensionais

Os algoritmos de síntese de imagem baseados na integração no espaço dos trajectos de luz (path space) produzem resultados de elevada qualidade, perceptualmente indistinguíveis do mundo real. Esta qualidade, no entanto, é conseguida à custa de um elevado número de amostras; de fato, a qualidade das imagens é directamente proporcional à raiz quadrada do número de amostras: $\mathcal{O}(\sqrt{N})$. O tempo de execução é directamente proporcional ao número de caminhos de transporte de luz amostrados, logo imagens de alta qualidade exigem tempos de processamento elevados.

O principal objetivo deste trabalho é reduzir o tempo de síntese de imagens de alta qualidade, reduzindo o número de amostras através da aplicação de técnicas de reconstrução esparsa. Estas técnicas permitem a reconstrução de determinado tipo de sinais mesmo quando a taxa de amostragem é muito baixa, de fato, muito inferior ao limite de Nyquist. Tal é conseguido explorando algum conhecimento prévio sobre o sinal, nomeadamente, o fato de este admitir uma representação esparsa numa base alternativa. Ao longo da última década, técnicas de reconstrução esparsa como Compressive Sensing e a Sparse Fourier Transform ganharam maturidade e demonstraram resultados impressionantes.

Esta tese aplica técnicas de reconstrução esparsa ao problema da iluminação global, incluindo integração no espaço dos trajectos de luz, visando a redução o número de amostras necessárias para produzir imagens de elevada qualidade. O espaço de Fourier é usado como base de representação esparsa: sinais de alta dimensionalidade têm sido descritos como aceitando representações com um número reduzido de coeficientes (logo esparsas) neste espaço. As principais contribuições desta tese são:

- Uma metodologia para análise de esparsidade de sinais de transporte de luz, que permite verificar se estes exibem as propriedades necessárias para uma reconstrução esparsa eficaz: sinal denso na base de amostragem e esparso na base de representação.
- Um método para a integração eficiente de sinais multidimensionais no espaço de Fourier. O processo de integração é tratado como uma convolução por um filtro box-car, reduzindo a dimensionalidade do problema e calculando o integral usando uma transformada de Fourier inversa de menor dimensionalidade. Optimizam-se o tempo de execução e a utilização de memória relativamente a abordagens anteriores, o que é conseguido explorando a eficiência da FFT que resulta numa diminuição da complexidade algoritmica.
- Um pipeline de reconstrução esparsa usando a Sparse Fourier Transform (sFT) aplicada ao path space discretizado no espaço primário de amostragem (primary sample space). Devido ao elevado número de amostras exigidos pela sFT a reconstrução direta do sinal não é tão eficiente quanto abordagens clássicas. Reconstruções de menor qualidade, logo usando menos amostras, podem ser usadas como mapas de importância. Estes funcionam como um mecanismo que guia a amostragem do processo de Monte Carlo, exibindo melhorias locais na síntese de cáusticas.
- Um segundo pipeline de reconstrução baseado em algoritmos da família do Matchin Pursuit (CoSaMP, especificamente). Dado que a complexidade destes algoritmos é exponencial com o número de dimensões do sinal, foram usados sinais de menor dimensionalidade, nomeadamente depth of field e iluminação indireta. A reconstrução direta destes sinais não atingiu melhorias sobre abordagens clássicas, para o mesmo tempo de execução. A execução do CoSaMP constitui um passo de otimização adicional que representa uma fração significativa do tempo de execução, anulando os ganhos obtidos com a redução do número de amostras. Foi desenvolvida Uma abordagem de amostragem por importância com resultados semelhantes ao ponto anterior. Adicionalmente, propõe-se utilizar os resultados de reconstrução esparsa como uma métrica de distância para um processo de Bilateral Filtering, atingindo melhorias na qualidade da imagem para cenas complexas.

Palavras-chave: Espaço de Fourier, Iluminação Global, Integração de Monte Carlo, Ray-Tracing, Reconstrução Esparsa

Contents

List of Figures

List of Tables

List of Acronyms

BPT	Bidirectional Path Tracing
BSDF	Bidirectional Scattering Distribution Function
CoSaMP	Compressive Sampling Matching Pursuit
CS	Compressive Sensing
MIS	Multiple Importance Sampling
MLT	Metropolis Light Transport
OMP	Orthogonal Matching Pursuit
pdf	Probability Density Function
PSNR	Peak Signal to Noise Ratio
RMSE	Root Mean Squared Error
SFT	Sparse Fourier Transform
SNR	Signal to Noise Ratio
SSIM	Structural Similarity Index

Chapter 1

Introduction

Photo realistic rendering algorithms provide the possibility of synthesising images of such quality that they become almost indistinguishable from the real world. Applications range from film production to computer aided design, including virtual archaeology, virtual architecture and computer games, among many others. Generating such images, perceptually indistinguishable from real world, has witnessed tremendous developments over the last 35 years, even though some problems remain yet to be solved, particularly those related to performance. In order to create such realistic images, algorithms try to simulate the light transport within a scene to determine the radiance reaching each pixel of the resulting image. This is done by calculating, for each pixel, the integral over the contribution of the relevant light transport paths. Since this integral has no analytical solution, it is usually estimated by Monte Carlo integration, and there are several algorithms that calculate this integral approximation. These algorithms however, require a large number of samples (light paths) in order to provide high quality results. Sampling light paths, by tracing them through the scene, is computationally expensive, so most of these algorithms are not used interactively for complex scenes.

There are several complementary approaches to reduce the execution time of rendering algorithms. These focus either on speeding up the processing of each sample (e.g., by using faster algorithms, improving data structures and increasing computational resources) or on reducing the number of samples required to achieve results of identical quality. This PhD work will explore the latter by exploiting sparse reconstruction algorithms.

This work goal is to reduce the time required to produce high quality photo realistic images by reducing the number of required samples. The hypothesis is that this can be achieved by applying sparse reconstruction algorithms. these have the ability to reduce the number of samples required to produce an image at the expense of an additional computational step. This additional computational step consists in a model fitting process, where the samples taken are fitted into a sparse approximation in a representation basis, in this case, the Fourier basis.

Before using any sparse reconstruction algorithm however, a thorough analysis of the sparsity of the signals of interest is performed in both the Fourier basis and the canonical basis, in order to assess the potential of these algorithms in this context. Such analysis includes specialized metrics such as the Gini Index as well as image quality metrics of hard-thresholded signals.

Compressive Sensing (CS) is a reconstruction technique which states that some signals can be recovered from far fewer samples than conventionally required if the signal is sparse when represented in some appropriate basis [9]. Through this assumption, it is possible to reconstruct a given signal with sampling rates much lower than the Nyquist rate by fitting the acquired samples to a sparse model in the representation basis used. Applying the CS mechanisms to the rendering pipeline could improve rendering time by reducing the number of sampled paths. Current applications of CS to rendering take advantage of the image sparsity in the wavelet domain [63] and multidimensional images (with motion blur, depth of field, among others) considering sparsity in the frequency domain [64].

One alternative to CS algorithms are combinatorial approaches such as the Sparse Fourier Transform (SFT) [34], a sparse reconstruction algorithm that aims to reconstruct a sparse Fourier approximation of a given signal in sub-linear time. Through the use of frequency permutations and filtering, this algorithm hashes the Fourier frequencies into a small number of buckets which represent the sum of a set of coefficients. Using several different hashes it is possible to detect which frequency is hashed to each bucket through a phase shift, while the magnitude can be estimated by taking the median of several hashes. This process allows for an efficient estimation of a sparse Fourier approximation of the signal.

This thesis evaluates both CS approaches using the Compressive Sampling Matching Pursuit (CoSaMP) algorithm and the SFT as a means to reduce sampling requirements for light transport evaluation. In the first test case, a discrete Path Space of limited length was evaluated. Due to the large size of this signal, the SFT algorithm was used given its lower time and memory complexity when compared to CS based approaches, which would require an impractical amount of memory. After the sparsity analysis, a reconstruction pipeline using the SFT algorithm was implemented, which takes the result of the SFT algorithm and integrates it over all dimensions except the image plane in order to generate a final image. Besides direct reconstruction, the results of the SFT provides a global model of the Path Space. This can be used as a multidimensional importance map which can then guide the sampling for a non-biased Monte Carlo integration process.

In order to generate an image from a compressed Fourier representation, it is necessary to integrate over some of the dimensions of the signal. In this thesis an efficient integration method for the Fourier basis with reduced time and memory complexity is proposed. This approach views the integration process as a convolution of the signal with a box-car, which is equivalent to a multiplication in Fourier. Due to the size of the box-car, the equivalent sinc in Fourier becomes a dirac-delta function, which makes the signal zero along every integrated dimension in Fourier. In order to recover the integrated in the original space it is only required to compute a lower dimension inverse Fourier transform.

CS when implemented using Matching Pursuit like algorithms (such as CoSaMP) suffers from the curse of dimensionality. It can not efficiently be applied to a high dimensionality space, such as the previously studied Path Space. Thus, in order to evaluate its performance in this context, CS is applied to low dimensionality case-studies, in particular to depth of field and one bounce indirect lighting, both 4-dimensional signals. After the sparsity analysis of these spaces, pipelines were developed using the CS result for integration and importance sampling. Finally, a new filtering approach is proposed, which uses the CS result as a guide for a bilateral filtering process.

1.1 Contributions

This thesis makes the following five major contributions:

- First, a thorough sparsity analysis of Path Space, Depth of Field and Indirect Lighting, which helps to better understand the possibilities and limitations of sparse reconstruction algorithms applied to the rendering problem. Through the performed analysis, it was learned that while the studied spaces have sparse energy distributions in the Fourier space, the Fourier coefficient magnitude is not so sparse which makes it difficult to differentiate high contribution frequencies. Also, the studied signals presented high sparsity in their canonical representation, which hinders the sparse reconstruction process.
- Second, a thorough understanding of the advantages and drawbacks of applying sparse reconstruction techniques, in particular CS and the SFT, to multidimensional light transport signals.
- The third major contribution is an efficient method for integrating sparse reconstruction results in the Fourier basis, which provide a time and memory efficient method for generating images from multidimensional sparse Fourier reconstructions. The time and memory complexity of this approach is proportional only to the image size (more specifically to the size of the signal's dimensions over which no integration is done), being completely independent of the sparsity and size of the integration domain.
- Fourth, a filtering approach which uses sparse reconstruction results as a guide for a bilateral filter of a Monte Carlo integrated image. This results in a smoother image with less noise for similar rendering times when compared to Path Tracing.
- Finally, an importance sampling technique is proposed, which uses the results of sparse reconstruction to drive a novel path guiding technique. This provides significant improvements when rendering caustics.

1.2 Thesis Structure

- The first chapter of this thesis presents the introduction, context and the main goals for the doctoral work.
- The second chapter provides the background for this work, namely an overview of Path Space Integration rendering algorithms, CS mechanisms are presented, as well as current applications of CS to rendering, and the SFT algorithm.
- The third chapter presents the methodology for the experimental work developed.

- The fourth chapter presents a proposed method for efficiently integrate the results of sparse reconstruction algorithms in order to calculate multidimensional integrals of the Fourier representations with reduced time and memory complexity.

- The fifth chapter presents a sparsity analysis of Path Space, in which a series of metrics for sparsity are evaluated and analysed in order to assess the viability of sparse reconstruction algorithms applied to this problem.

- The sixth chapter presents a reconstruction pipeline using SFT applied to Path Space. The SFT algorithm is used to generate a sparse approximation of Path Space in the Fourier Basis, which after integration provides the final rendered image. Also an Importance Sampling pipeline is proposed, where the result of the SFT algorithm is used as an importance map used for unbiased Monte Carlo rendering.

- The seventh chapter presents a sparsity analysis of depth of field and indirect lighting using the same metrics as the fifth chapter.

- The eighth chapter presents a reconstruction pipeline using CoSaMP applied to depth of field and indirect lighting, as well as a filtering based approach which used the CoSaMP reconstruction as a guide for the filtering process. Also, an importance sampling approach is proposed.

- The ninth chapter presents the conclusions and future work.

Chapter 2

Background

Through this chapter the core background on path space integration algorithms is presented, as well as two alternative sparse reconstruction approaches, CS and the SFT

2.1 Path Space Integration Algorithms

Physically based rendering is the process of generating synthetic images indistinguishable from the perception of the real world. These images are generated according to a model of the scene, including its geometry, materials and light sources. With this information is possible to simulate the light transport process from the light sources to the camera through the scene in order to synthesize an image.

2.1.1 Radiometry

Light sources emit photons, packets of energy, Q, that travel and interact throughout the scene, being reflected and absorbed, until they finally reach the camera sensor where they can be measured. Radiant flux (Φ), or radiant power, measures (in watts) how much energy is transmitted per unit of time.

$$\Phi = \frac{dQ}{dt}$$

Irradiance (E) measures radiant flux arriving or exiting a given point p, and is measured as radiant flux per area unit (W/m^2).

$$E(p) = \frac{d\Phi}{dA}$$

Radiance (L) measures the radiant flux arriving or exiting a given point p through direction ω and is represented as the radiant flux per area unit per solid angle unit ($W/(m^2 sr)$). Solid angle represents the projected area of a surface on the unit sphere centred on p and is measured in steradians (sr).

$$L(p,\omega) = \frac{d\Phi}{d\omega dA \cos\theta}$$

where θ is the angle between ω and the surface normal. The cosine factor present accounts for Lambert's Law, since a non-perpendicular incidence projects a larger area, thus reducing irradiance, and consequently radiance. Radiance is the most important radiometric quantity, since knowing it allows the calculation of all other radiometric quantities by integrating in terms of area and directions. Thus, the goal of photo realistic rendering is calculating the radiance arriving to each pixel on the sensor of the virtual camera [58].

2.1.2 Scene Description

The virtual scene itself is represented by a set of surfaces or shapes defined in a 3-dimensional space, each with a corresponding material, a model which represents how light interacts with its surface. This model, a Bidirectional Scattering Distribution Function (BSDF), measures how much radiance arriving from an incoming direction ω_i is scattered (transmitted or reflected) towards an outgoing direction ω_o. Different models of BSDF can be used and combined to represent different kinds of material lighting interactions such as diffuse, specular, glossy, transmission, among others [24] [59] [55]. Some simple BSDF models include the perfectly diffuse, where light is scattered uniformly toward all directions around the hemisphere centred on the surface normal (equation 2.1).

$$f_r(x, \omega_i \rightarrow \omega_o) = \begin{cases} k_d \frac{1}{2\pi} & n \cdot \omega_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where n is the surface normal at point x and k_d is the diffuse reflection coefficient. Or the perfect specular surface, which is zero on all directions except the ideal reflection direction (equation 2.2).

$$f_r(x, \omega_i \rightarrow \omega_o) = \begin{cases} 1 & \omega_o = \text{reflect}(\omega_i, n) \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

$$\text{reflect}(\omega, n) = -\omega + 2(\omega \cdot n)n$$

Generally, surfaces are not completely diffuse or specular, often assuming an intermediate behaviour. One way of modelling these surfaces is the micro-facet BSDF [66]. This BSDF models the surface as a composition of microscopic V-shaped facets that act as ideal specular surfaces.

$$f_r(x, \omega_i \rightarrow \omega_o)) = \frac{D(\omega_h)G(\omega_o, \omega_i)F_r(\omega_o)}{4(\omega_i \cdot n)(\omega_o \cdot n)} \tag{2.3}$$

The factor D models the distribution of the micro-facet normals. $D(\omega_h)$ gives the portion of micro-facets oriented towards ω_h, the normal vector that provides a perfect specular reflection from ω_i to ω_o. The factor G accounts for geometric attenuation within the micro-facets such as occlusion and shadowing. F_r is the Fresnel term, measuring how much light is reflected given the incidence angle.

2.1.3 Path Space Integration

The problem of physically based rendering can be described as a problem of integration for each pixel over the space of relevant light transport paths, as proposed by Veach [69].

$$\Phi_j = \int_{\Omega_j} f(z) d\mu(z) \tag{2.4}$$

where Φ_j is the value for the pixel j, Ω_j is the space of light transport paths that contribute to Φ_j, $f(z)$ is the contribution function for a path z, and μ is a measure function. A path z is defined by a sequence of points x_0 .. x_n in which x_0 is a point on a light source and x_n is a point on the camera lens.

This high level framework allows the unification of several light transport algorithms into one single framework. Also, this framework helps understand the fundamental differences among rendering algorithms, as some of them are more capable than others at sampling specific path types, such as caustic or reflected caustic paths.

Through this section several light transport algorithms based on Path Space integration will be presented.

2.1.4 Path Tracing

One of the first solutions to the global illumination was introduced by Kajiya [38] which defined the problem of rendering as the solution of an integral equation, also know as the rendering equation.

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} f_r(x, \omega_i \rightarrow \omega_o) L_i(x, \omega_i)(\omega_i \cdot n) d\omega_i \tag{2.5}$$

Where:

- $L_o(x, \omega_o)$ is the outgoing radiance from point x towards the direction ω_o
- $L_e(x, \omega_o)$ is the radiance emitted from point x towards the direction ω_o
- Ω is the unit hemisphere centred around n, which contains all possible directions for ω
- $f_r(x, \omega_i \rightarrow \omega_o)$ is the BSDF, which models how the material scatters light arriving from ω_i towards ω_o
- $L_i(x, \omega_i)$ is the radiance arriving at point x from the direction ω_i
- n is the surface normal at x
- $\omega_i \cdot n$ is the Lambert's attenuation due to incident angle

This equation translates to how much light arrives at a given point from a given direction. This integral equation can not be calculated analytically, so its expected value is calculated through Monte Carlo

Integration, which estimates a given integral as a mean of random samples z_i divided by the Probability Density Function (pdf) of sampling z_i.

$$\int f(z)dz \approx \frac{1}{m}\sum_{i=0}^{m-1}\frac{f(z_i)}{p(z_i)} \tag{2.6}$$

where m is the number of samples, $z_0..z_{m-1}$ are the random samples selected and $p(z_i)$ is the pdf of selecting point z_i [48]. Due to the random nature of Monte Carlo methods, this estimation introduces variance to the result. The variance of a random variable X can be defined as the expected value of the squared deviation from the exact value of X:

$$Var(X) = E[(X - X_e)^2]$$

where $X_e = E[X]$. This means that the estimation may not be exactly the correct result, but converge towards it as the number of samples m increases. Also, the closer the probability distribution function fits the function being integrated, the lower the variance result of the integral estimation for the same number of samples.

Based on this, the goal is to trace a light transport path, with points x_0 .. x_n in which x_0 is a point on a light source and x_n is a point on the camera lens. The outward direction at each intersection usually being importance sampled with a similar distribution to the BSDF, concentrating samples towards directions where the BSDF is maximal. In the case of an uniform BSDF, a cosine sample distribution is typically used.

Commonly, these paths are traced from the camera to the light source in order to ensure that paths sampled fall in the image plane and to capture reflection and refraction paths. However caustic paths have a very low sampling probability, reaching zero if using point light sources. Tracing paths from the light source to the camera [17] however sample caustics more effectively but specular connections to a pinhole camera cannot be sampled.

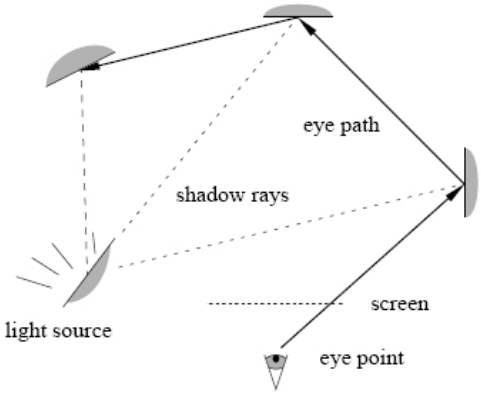


Figure 2.1: Path Tracing Algorithm [46]

2.1.5 Bidirectional Path Tracing

Bidirectional Path Tracing (BPT), proposed independently by Veach and Guibas [70] and Lafortune and Willems [45] combines both forward and backwards path tracing in one single method that can become more robust than the previous two alone. Although with a different mathematical background, both authors propose that the method should sample pairs of sub-paths containing a light sub-path and a camera sub-path. Then, each vertex of one sub-path is explicitly connected to all vertices of the other one, generating a new set of light transport paths.

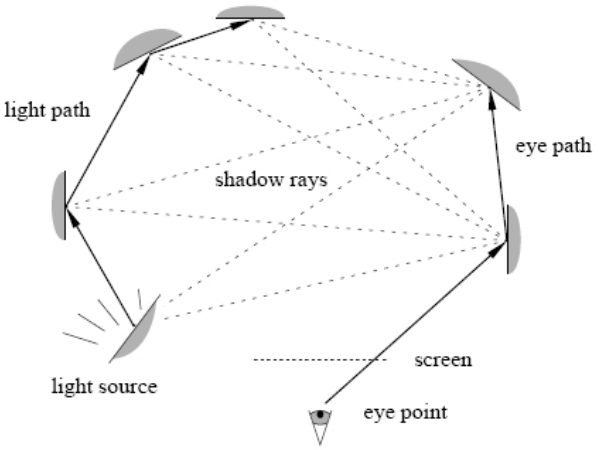


Figure 2.2: Bidirectional Path Tracing Algorithm [46]

In order to make the algorithm more robust, Veach and Guibas [70] proposed the Path Space Integration Framework and Multiple Importance Sampling (MIS). The first consists in converting the problem of light transport from a recursive integral equation (2.5) to a simple integral over the space of all light transport paths, as shown in equation 2.4.

Multiple Importance Sampling is a technique that attempts to combine multiple sampling techniques in a probably good way in order to minimize variance. The Monte Carlo estimator for this approach is given by:

$$F = \sum_{i=1}^{t} \frac{1}{m_i} \sum_{j=1}^{m_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})} \tag{2.7}$$

where t is the number of sampling techniques and m_i the number of samples for technique i. This estimator attributes a weight w to each sample. The balance heuristic is a simple and robust way to calculate these weights and is demonstrated that no other heuristic is much better [71, p. 264], and can be calculated by the following equation.

$$w_i(x) = \frac{n_i p_i(x)}{\sum_k n_k p_k(x)} \tag{2.8}$$

In order to use MIS in Bidirectional Path Tracing one must consider that each path of a given length can be sampled in several ways: a path of length N can be sampled by using whatever light and camera

sub-paths of l and e vertices as long as $N = l + e + 1$. The Bidirectional Path Tracing integrator using Multiple Importance Sampling can be expressed as:

$$I = \sum_{l \geq 0} \sum_{e \geq 0} \omega_{l,e}(x_{l,e}) \frac{f_j(x_{l,e})}{p_{l,e}(x_{l,e})} \tag{2.9}$$

where $p_{s,t}$ is the probability of sampling a given path x through the connection of a light sub-path of length l and a camera sub-path of length e.

This algorithm is much more robust than path tracing, as it can sample many light transport paths efficiently and robustly. Nonetheless, this algorithm is not perfect for every situation, as it has difficulty sampling reflected caustics [19], and is more computationally expensive due to the weight and visibility computation.

2.1.6 Metropolis Light Transport

In an effort to improve the efficiency of light transport algorithms, Veach and Guibas [72] proposed the adaptation of the Metropolis Sampling algorithm [52] [28] to the light transport problem. This algorithm starts from one point in the function domain and generates a random walk such that in the limit, the sampling distribution is proportional to the function value, called the stationary distribution function, providing a global sampling strategy that aims to sample paths with a probability proportional to their contribution. These new samples however, may not always be accepted, and have an acceptance probability given by equation 2.10.

$$a(x \rightarrow y) = min \left\{ 1, \frac{f(y)T(y \rightarrow x)}{f(x)T(x \rightarrow y)} \right\} \tag{2.10}$$

where $a(x \rightarrow y)$ is the acceptance probability from state x to y and T is the the transition probability for the mutation being applied. There is flexibility in choosing T, but it is conceptually similar to the pdf in Monte Carlo sampling. As in the limit, the samples are distributed proportionally to the function value, all samples have the same contribution to the final result.

Applying this algorithm to the light transport problem, the function domain is the space of light transport paths and the function value is the contribution these paths have in the final image. The Metropolis Light Transport (MLT) algorithm starts by generating a light transport path, using BPT for instance, and then apply mutations to that path using the acceptance probability defined in equation 2.10.

As the initial sample in the algorithm was not sampled according to the stationary distribution function, the algorithm exhibits start-up bias, because only in the limit the samples converge to stationary distribution function and until then the results may be heavily influenced by the choice of the initial path. This problem is solved by sampling a set of paths, assigning each a weight $W_0 = f(X_0)/p_0(X_0)$, and then select a subset of these paths with a probability proportional to these weights. Then, all these paths are assigned the same weight that is the average of all previously sampled weights. This solves two problems: removes start-up bias and calculates the normalization weight of the samples, scaling them to the right value.

Primary Sample Space

An important part of the Metropolis algorithm are the mutations ($T(x \rightarrow y)$). In his work, Veach and Guibas [72] proposed a set of mutations that aimed to solve some well known problems of light transport algorithms, namely reflected caustics. However, Kelemen et al. [43] proposed a simpler and robust form of mutation in the primary sampling space. Instead of trying to directly mutate the paths, mutations are applied to the pseudo-random numbers used to generate them, being the magnitude of these perturbation smaller the higher the contribution of the sample, in order to keep the acceptance probability high. One important property of the mutations used in the Metropolis algorithm is that these must be able to transit from any point in the function domain to any other possible point, or else the algorithm would not respect the principle of ergodicity. In order to do that, sometimes it is needed to perform a large step mutation, that is regenerate all the random numbers used.

Path Space integration algorithms build paths from sequences of uniformly distributed random numbers, and then use the probability density function to generate the path according to that distribution.

Considering the unit hypercube containing all the possible sequences of n random numbers U_n, and $S(u)$ the transformation between a given random number sequence and the path generated $z = S(u)$, then we can calculate the previous integral over this new space.

$$\Phi = \int_{\Omega} f(z) d\mu(z) = \int_{U_n} f(S(u)) \left| \frac{dS(u)}{du} \right| du \approx \frac{1}{M} \sum_{i=0}^{M-1} \frac{f(S(u_i))}{\left| \frac{dS(u_i)}{du_i} \right|^{-1}} \tag{2.11}$$

where $|\cdot|$ denotes the determinant of the Jacobian matrix. Any path z generated is unambiguously identified the the sequence of random numbers u used to generate the path. The space of all the possible random number sequences is called Primary Sample Space [43].

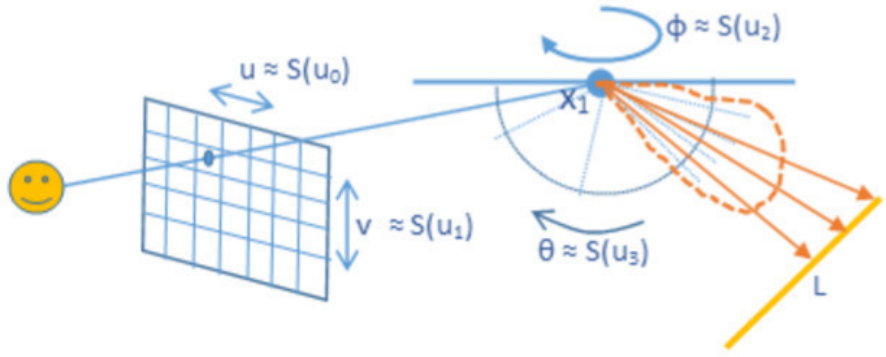


Figure 2.3: Primary Sample Space parametrization of a path of length 2

2.1.7 Path Guiding

Conventional path sampling is a local sampling operation, where each subsequent segment of the path is importance sampled after the next. Although efficient for most scenarios, this approach fails when a high contribution path has a low sampling probability when sampled locally.

Global sampling approaches like Metropolis Light Transport tackle this problem through a Markov chain process in order to converge to a sampling distribution that matches the contribution function as close as possible. However, Metropolis approaches reject a large number of samples in order to achieve this sampling distribution, which lowers the overall efficiency of the algorithm [80].

Another possible approach for global sampling is to store and sample discrete distributions. Lafortune and Willems [47] sample a set of initial paths and store them in a 5D tree of scene positions and directions which are used to guide the sampling process. Müller et al. [53] apply a similar data structure for path guiding, using interactive training during rendering. Jensen [37] and Hey and Purgathofer [31] use Photon Mapping in order to generate the sampling distribution on the hemisphere of surfaces, guiding the samples towards high contribution directions.

It is also possible to guide path samples using a sparse set of distributions which aim to approximate the incident radiance or importance. Bashford-Rogers et al. [3] use a set of cosine lobes to approach and store sampling distributions. Through the use of Expectation-Maximization it is possible to fit a sparse set of Gaussian lobes to particles traced from both the camera and light sources, and then used to sample light transport path in runtime [74]. This approach was further refined by Herholz et al. [30] by combining the Gaussian lobe with BSDF sampling

Zheng and Zwicker [80] however propose an approach that tries to learn this distribution through the use of a neural network applied to primary sample space.

Through the use of a real-valued non-volume preserving (Real NVP) architecture of neural networks, this approach aims to learn, from an initial set of samples, a function Ψ that warps the unit hyper-cube that is Primary Sample Space.

$$\Psi_n : [0,1]^{2(n+1)} \rightarrow [0,1]^{2(n+1)} \tag{2.12}$$

where n is the maximum path length. This warping function aims to give higher sampling probabilities to higher contribution points in this domain, thus reducing variance. Through a change in integration variable it is possible to derive

$$\Phi = \int_\Omega f(z)dz = \int_{U_n} f(S(u)) \left|\frac{dS(u)}{du}\right| du = \int_{U_n} f(S(\Psi(y))) \left|\frac{d\Psi(y)}{dy}\right| \left|\frac{dS(\Psi(y))}{d\Psi(y)}\right| dy \tag{2.13}$$

$$\int_{U_n} f(S(\Psi(y))) \left|\frac{d\Psi(y)}{dy}\right| \left|\frac{dS(\Psi(y))}{d\Psi(y)}\right| dy \approx \frac{1}{M} \sum_{i=0}^{M-1} \frac{f(S(\Psi(y_i)))}{\left|\frac{d\Psi(y_i)}{dy_i}\right|^{-1} \left|\frac{dS(\Psi(y_i))}{d\Psi(y_i)}\right|^{-1}} \tag{2.14}$$

Given this, the goal is to generate a distribution $p_u(u)$ that minimizes variance.

$$p_u(u) = \frac{f(S(u))}{\left|\frac{dS(u)}{du}\right|^{-1}} \tag{2.15}$$

Such distribution closely relates the inverse warp $y = \Psi^{-1}(u)$ by the following change of variable.

$$p_u(u) = \frac{f(S(u))}{\left|\frac{dS(u)}{du}\right|^{-1}} = \left|\frac{d\Psi^{-1}(u)}{du}\right| \tag{2.16}$$

Through a maximum likelihood estimation, the inverse warp Ψ^{-1} is generated such that the warp has a maximum likelihood of producing samples $u\tilde{p}_u$ from uniform samples y_i.

Using RealNVP, the inverse warp Ψ^{-1} is determined from an initial set of samples. The warp function Ψ is then derived from the inverse, and samples distributed according to $p_u(u)$ are obtained by transforming an uniform set of samples by Ψ.

This approach has shown considerable improvements when applied to 4-dimension primary sample space, but can be extended to an arbitrary number of dimensions, albeit with higher training sampling requirements. Because this approach adds a training phase, it adds a constant cost to rendering time, which can be inefficient unless this cost can be amortized over a larger rendering time or applied to higher complexity scenes.

2.2 Compressive Sensing

Reconstructing a signal from a set of samples is a common task often required in various fields of research such as communications, image and sound reconstruction, among others. Most classical techniques to signal reconstruction are based on Shannon's theorem, which in order to completely recover a signal requires sampling it with a frequency of at least twice the largest frequency present in the signal (the Nyquist rate). This often leads to a large amount of information needing to be sampled in order to reconstruct a signal. As a result, it is necessary to compress the signal for efficient storage. Since many signals admit concise representations when transformed to a given basis (most natural images have concise representations in the wavelet basis for example), signals are transformed to these basis and then only the highest coefficients are stored. This leads to a lot of sampling work to be effectively wasted as most signal coefficients are discarded after compression [9] [2] [18].

CS is a sampling technique that states that certain signals can be recovered using far fewer samples than those required by the Nyquist Rate. This derives from the fact that the signal of interest has a sparse representation when transformed to a suitable basis [6] [8] [16]. In practice it is possible to sample a signal proportionally to the amount of information present in the signal rather than its band limit. In order to recover a signal, CS relies on two basic principles: sparsity and incoherence. Sparsity relates to the amount of data required to represent the signal in a given basis. Incoherence implies that a sparse signal in a given basis must be dense in the domain it is sampled, and vice-versa. If these two properties are verified, then it is possible to reconstruct the signal with a smaller amount of samples than required by the Nyquist Rate .

2.2.1 Sparse Signal Recovery

Consider a discrete band limited signal $s \in \mathbb{R}^n$ with a maximum possible frequency of $n/2$. In order to fully reconstruct it one would need to sample all n coefficients. However for some cases it might not be efficient or even possible to do so. In this case, instead of n, only $m \ll n$ samples might be available. In order to reconstruct the signal with only m samples, one would need to solve an undetermined system of equations. Denoting A as the $m \times n$ sensing matrix and y the measurement vector, the system $y = Af$ is undetermined as there are many possible signals f that fit this system. Although complex, it is feasible to solve this problem under two assumptions: sparsity of the signal of interest in a given basis, and incoherence between the sensing basis and the representation basis.

Sparsity

Most natural signals have concise representations when expressed in an appropriate basis. This means that one can discard small coefficients without any relevant information loss.

Considering Ψ a $n \times n$ matrix the orthonormal basis that transforms the concise representation $x \in \mathbb{R}^n$ to its natural representation, $s \in \mathbb{R}^n$.

$$s = \Psi x \tag{2.17}$$

A signal is said to be K-Sparse if at most K of its coefficients are different than zero when represented in the appropriate basis. Completely sparse signals are however rare, so another helpful definition is compressibility. A signal is K-compressible if most of its total energy is focused on K coefficients, the remaining being small enough to be ignored. If x is sparse or at least compressible, one could approximate x by x_K preserving only the K largest coefficients of x, and setting all the others to zero. As Ψ is an orthonormal basis, the magnitude of the difference between the original and the approximation is preserved, so in the canonical domain the perceptual loss is small. This is the basis of many transform compression algorithms such as JPEG-2000 that from the complete signal s compute x and then only store the significant coefficients obtaining a sparse approximation. CS on the other hand takes advantage of sparsity to reduce the number of samples needed to reconstruct the signal s, requiring less samples the sparser the signal is when represented in Ψ [9].

Incoherence

Given two orthonormal basis, Φ, the sensing matrix, and Ψ, the representation matrix, the coherence between these two basis $\mu(\Phi, \Psi)$ is given by:

$$\mu(\Phi, \Psi) = \sqrt{n}\,\text{max}\langle\phi_k, \psi_j\rangle \text{for} k, j \in [1, n] \tag{2.18}$$

where n is the number of points of the signal.

This measures the maximum correlation between the elements of the two bases. The lower this

correlation the lower the coherence. As both Φ and Ψ are orthonormal bases, $\mu(\Phi, \Psi) \in [1, \sqrt{n}]$. For low coherence pairs (coherence is close to one), a sparse signal in one basis has a dense representation in the other and vice versa. In other words, a single point in the sensing basis captures information about all coefficients in the representation basis, hence the importance of incoherence.

A common low coherence basis pair is the point sensing basis (the canonical basis) and the Fourier basis (frequency domain), which have a minimal coherence of one in any number of dimensions. Besides having a low coherence with the point sampling basis, Fourier basis has an analytical representation which can be calculated in runtime, without the need to store the matrix, a useful property for practical implementations. Also, given any fixed basis Ψ, a random orthonormal basis Φ is highly incoherent with high probability [9].

Signal Reconstruction

The goal is to reconstruct the signal $s \in \mathbb{R}^n$, knowing it has a sparse representation $x \in \mathbb{R}^n$ where $s = \Psi x$ from a measurement vector $y \in \mathbb{R}^m$ such that:

$$y_k = \langle \phi_k, \Psi x \rangle, k \in M \tag{2.19}$$

where $M \subset \{1, \ldots, n\}$ with a cardinality $m < n$.

This system of equations is undetermined but with the information that x is sparse, it is possible to estimate x by the sparsest possible $\tilde{x} \in \mathbb{R}^n$ that matches the measured values $y \in \mathbb{R}^m$. This transforms the problem of solving an undetermined system of equations into an optimization problem. Considering $\|x\|_0$ the number of non-zero coefficients of x, the optimization goal is:

$$\text{min } \|\tilde{x}\|_0 \text{ subject to } y_k = \langle \phi_k, \Psi \tilde{x} \rangle, \forall k \in M \tag{2.20}$$

Minimizing $\|\tilde{x}\|_0$ is an $NP - hard$ problem, which is impractical to implement in real applications. However, it has been demonstrated that given enough samples, the result of minimizing the l_1 norm ($\|\tilde{x}\|_1$) provides the same result of minimizing $\|\tilde{x}\|_0$ [5]. The problem of l_1 minimization subject to constraints, unlike $\|\tilde{x}\|_0$ minimization, has a series of efficient algorithms, which makes this a good solution. Also, it is proven that through l_1 minimization, exact recovery of an K-sparse signal occurs with m random measurements as long as

$$m \geq C \cdot \mu^2(\Phi, \Psi) \cdot K \cdot \log n \tag{2.21}$$

for a positive constant C [11].

Also, beyond optimization methods, it is possible to estimate the sparse signal through greedy methods [67], such as matching pursuit based algorithms, that try to quickly find the greatest possible coefficients for x by iteratively selecting a new candidate coefficient to add to x, as well as combinatorial solutions implemented by algorithms such as the Sparse Fourier Transform.

Orthogonal Matching Pursuit

One of these greedy algorithms is the Orthogonal Matching Pursuit (OMP). This iterative algorithm algorithm reconstructs a given signal s from a measuring vector y by adding a new component to the support vector in each iteration. The support vector is the vector containing the indices of the relevant coefficients. The new component chosen is the one with the largest projection of the current residual r in the representation basis.

$$t = max_t(\langle \psi_t, r \rangle) \tag{2.22}$$

After detecting a new component, it is added to the support $c_i = c_{i-1} \cup t$. Afterwards, the current estimate P is calculated by calculating a least-squares estimate $\Psi_c^{-1} P - y$, which aims to minimize the residual, which is updated in the end: $r_i = y - \Psi_c^{-1} P_i$. This process is repeated until a given stopping criteria is achieved, usually until the support size matches the expected K [57] [68].

Algorithm 1: Orthogonal Matching Pursuit Algorithm

Data: y, A, K;
Result: $\tilde{x}$

$r = y$;
$S = \{\}$;
while $\|S\|_0 < K$ **do**
  $i = max_i(A^T r)$; // greatest coefficient projection
  $S = S \cup \{i\}$; // add coefficient to support
  $\tilde{x} = (A_S^T A_S)^{-1} A_S^T y$; // estimate signal through least squares
  $r = y - A\tilde{x}$; // update residual
end

CoSaMP

The CoSaMP algorithm is very similar to the above mentioned OMP but instead of adding only one component in each iteration, the $2K$ largest components in the projection are added to the tentative support. After the estimation phase, the estimation is pruned to the K largest components and the iterative process continues similarly, updating the residual and repeating these steps until a given stopping criteria is reached [54].

During the estimation phase, using direct methods to compute the least-squares is most likely inefficient due to time and memory complexity. Because of this, it is advisable to use iterative methods to

compute the least-squares fit instead [54] [4].

Algorithm 2: CoSaMP Algorithm

Data: y, A, K;
Result: $\tilde{x}$

$r = y$;
$S = \{\}$;
while halting criteria **do**
  $a = A^T r$; // project residual
  $\Omega = supp(a_{2K})$; // detect 2K greatest coefficients
  $S = S \cup \Omega$; // add coefficients to support
  $b = (A_S^T A_S)^{-1} A_S^T y$; // estimate signal through least squares
  $\tilde{x} = prune(b, K)$; // prune to K coefficients
  $S = supp(\tilde{x})$; // update support
  $r = y - A\tilde{x}$; // update residual
end

The main advantage of CoSaMP over OMP is that it allows the removal of wrongly detected coefficients. In OMP once a coefficient is added to the support it is never removed, thus an error in support detection affects the whole process. With CoSaMP however, the support is updated every iteration, allowing the removal of wrongly detected coefficients should they no longer contain meaningful contribution given the current support.

2.2.2 Robustness in Compressive Sensing

In the previous section it was shown that CS can recover K-sparse signals with a set y of samples m. However, in order to be robust, CS must be able to handle signals which are not completely sparse, where most of its coefficients are zero. It must be able to properly handle compressible signals, where most of its coefficients are relatively small and can be ignored. Also, it must be able to handle noise in the measurement process, which in most realistic cases is inevitable.

Restricted Isometry Property

For given matrix A, the isometry constant δ_K is the smallest possible number such that

$$(1 - \delta_K)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_K)\|x\|_2^2 \tag{2.23}$$

is true for every K-sparse vector x [7] [10].

If the constant δ_K is not too close to one, the matrix A obeys the Restricted Isometry Property of order K. In practice this means that every subset of K columns of A is nearly orthogonal and the Euclidean length of an K-sparse vector is approximately preserved [18].

Applying this property to CS, if signal sampled is K-sparse, it would be desirable to have a sensing matrix A which obeys the Restricted Isometry Property of order $2K$. That way, the distances between all K-sparse vectors are approximately preserved:

$$(1 - \delta_{2K})\|x_1 - x_2\|_2^2 \leq \|Ax_1 - Ax_2\|_2^2 \leq (1 + \delta_{2K})\|x - x_2\|_2^2 \tag{2.24}$$

for every K-sparse vector x_1 and x_2. This makes every K-sparse vector distinguishable enough to perform a robust recovery [9].

Non-sparse Signal Recovery

Reconstructing a signal given a sensing matrix A that obeys the Restricted Isometry Property of order $2K$, an accurate reconstruction is obtained through l_1 minimization. If $\delta_{2K} < \sqrt{2} - 1$, then the obtained solution $\tilde{x}$ obeys the following:

$$\|\tilde{x} - x\|_2 \leq C_0 \cdot \frac{\|x - x_K\|_1}{\sqrt{s}} \tag{2.25}$$

$$\|\tilde{x} - x\|_1 \leq C_0 \cdot \|x - x_K\|_1 \tag{2.26}$$

for one positive constant C_0, where x_K is the vector x with all but the K largest coefficients set to zero [12].

Given this, CS is able to handle robustly any kind of signal, sparse or not. If the signal is K-sparse, then $x = x_K$ and the recovery is exact. Otherwise, the recovered signal $\tilde{x}$ is as approximate to x as if the K largest coefficients of x were measured directly [9] [13].

Signal Recovery with Noise

In all previously discussed results, it has been considered that all measurements could be obtained ideally without noise. However, in most realistic scenarios, measurements are affected by noise:

$$y = Ax + z \tag{2.27}$$

where z is an unknown error term.

When dealing with noisy data, the l_1 minimization step is altered, allowing for relaxed constraints, and in doing so, accounting for the effect of noise:

$$\text{min } \|\tilde{x}\|_1 \text{ subject to } \|A\tilde{x} - y\|_2 \leq \epsilon \tag{2.28}$$

where ϵ bounds the amount of noise.

If the constant $\delta_{2K} < \sqrt{2} - 1$ for the sensing matrix A, then the obtained solution $\tilde{x}$ obeys the

following:

$$\|\tilde{x} - x\|_2 \leq C_0 \cdot \frac{\|x - x_K\|_1}{\sqrt{s}} + C_1 \cdot \epsilon \tag{2.29}$$

for some constants C_0 and C_1.

These results show that the error of the reconstruction is bounded by the sum of two factors: one first factor caused by assuming the original signal is completely sparse, and a second factor proportional to the noise in the measurement process [9].

Random Sensing

In order for CS to be robust, the measurement matrix A is required to obey the Restricted Isometry Property. However it is difficult to obtain deterministic matrices that obey this property since calculating the isometry constants for a given matrix is a NP-hard problem, and designing such matrices requires a large number of samples [18]. This limitation can be overcome by using randomness, as it is proven that several types of random matrices such as matrices built by Gaussian or Bernoulli distribution obey the Restricted Isometry Property if

$$m \geq C \cdot K \log(n/K) \tag{2.30}$$

for a constant C.

Also, this property applies to pairs of incoherent orthonormal basis Φ and Ψ with m random measurements as long as

$$m \geq C \cdot \mu^2(\Phi, \Psi) \cdot K(\log n)^4 \tag{2.31}$$

for a constant C [18].

2.2.3 Compressive Rendering

One possible application of CS to rendering is to take advantage of the natural image sparsity in the wavelet domain, as proposed by Sen and Darabi [63]. This approach aims to reconstruct the final image by sampling only a portion of its pixels, using CS to estimate the values of missing pixels, thus saving rendering time.

Since in this context it is only possible to sample directly in the canonical signal domain, the ideal sparse basis would be the Fourier domain because of the low coherence between these basis. However, natural images are not sparse enough in this domain for CS to be efficient. Although images are sparse in the wavelet basis, this is incompatible with the point sampling basis due to high coherence.

In the solution proposed by Sen and Darabi [63], these challenges are overcome by assuming there exists a blurred image which can be sharpened to the original one. This blurred image is still sparse in the wavelet domain but since the sensing matrix now includes a blurred wavelet transform, which is more incoherent with the point sampling basis used, it is possible to apply the CS pipeline. After reconstruction, a blurred image is obtained, which must be sharpened by reversing the blurring filter in order to obtain the original image.

By applying this technique the authors demonstrate it is possible to render an image by sampling only 75% of its pixels, while maintaining image quality and effectively lowering rendering time. Also this technique surpasses other algorithms that estimate missing pixel information such as in-painting or interpolation regarding image quality.

Multidimensional Compressive Rendering

Natural images are commonly sparse in the wavelet domain. This however poses challenges to compressive sensing implementations, since in the context of rendering it is only possible to use the point sampling basis, incompatible with the wavelet basis due to high coherence. However, considering effects like depth of field, motion blur, or even video rendering, which increase the dimensionality of the signal of interest, sparser signals are obtained in the Fourier basis, which is highly incoherent with the point sampling basis used [64]. This increasing sparsity comes from the fact that with each dimension, the amount of data grows exponentially, but the information contained does not increase at the same rate.

By taking advantage of this sparsity increase, the authors demonstrate it is possible to reconstruct the high dimensionality signal using CS, and then integrate over it in order to obtain the resulting image.

The results presented by this approach show that it is possible to perform video rendering (3D signal) by sampling only 25% of the pixels across a video of resolution 512×512 with 128 frames.

Although this method can drastically reduce the number of samples used, at some point in execution it requires storing all the signal coefficients. Since the size of the signal of interest increases exponentially with the number of dimensions, this approach heavily suffers from the effect of the "curse of dimensionality", severely limiting its extension to higher dimension signals.

Compressive Sensing Integration

CS exploits the fact that a signal $f(z)$ admits a sparse representation in a given basis Ψ, allowing a reduction of samples required for reconstruction.

$$f = \Psi c \tag{2.32}$$

where Ψ is a $n \times n$ matrix that represents a set of basis functions, and c is the representation of f in Ψ, so f can be represented as

$$f(z) = \sum_{i=0}^{n-1} c_i \cdot \psi_i(z) \tag{2.33}$$

Since it is assumed that $f(x)$ is K-sparse when represented in Ψ, or at least K-compressible, only the K greatest coefficients are considered.

$$f(z) \approx f_K(z) = \sum_{i=0}^{K-1} c_{s[i]} \cdot \psi_{s[i]}(z) \tag{2.34}$$

where s is the set of relevant K coefficients.

In order to generate an image from the obtained multidimensional space representation, instead of integrating directly in the original domain, this operation can be applied to each of the basis functions in the compressed representation.

$$\int_{\Omega_j} f(z)dz \approx \int_{\Omega_j} f_K(z)dz = \int_{\Omega_j} \sum_{i=0}^{K-1} c_{s[i]} \cdot \psi_{s[i]}(z)dz \tag{2.35}$$

From here, it is possible to derive

$$\int_{\Omega_j} f(z)dz \approx \sum_{i=0}^{K-1} \left[c_{s[i]} \cdot \int_{\Omega_j} \psi_{s[i]}(z)dz \right] \tag{2.36}$$

This shows that integration of $f(z)$ can be calculated by integrating over the same sub domain of the basis functions and then performing an inner product with the representation of $f(z)$ in Ψ. This shows that it is possible to integrate over a compressed space representation [64] [62], and in order to get the value for each pixel a difference of integrals is calculated.

2.3 Sparse Fourier Transform

Sparse reconstruction algorithms provide an efficient way of reconstructing signals using a reduced number of samples. Most of these algorithms however have a time complexity proportional to the total number of points of the discrete signal ($O(N)$). When using the Fourier basis however, it is possible to efficiently reconstruct these signals with sub-linear time and sample complexities [21] [22] [35]

The SFT is an algorithm that aims to estimate the Discrete Fourier Transform of a given signal with a time complexity of $K \log(N)$ for a signal that is K-sparse in frequency domain [27] [26]. Unlike CS algorithms which perform global signal optimization or search, which have a linear time complexity w.r.t. signal size, the SFT leverages some properties of the Fourier Transform in order to produce an approximation in sub-linear time.

This algorithm (4) has two main components: the support detection, in which the high contribution frequencies of the signal are identified, and the magnitude estimation, in which the magnitudes of the previous frequencies are approximated. During both these phases, the signal is sampled several times by generating hashes of the Fourier signal.

The algorithm can also be adjusted by providing it with the expected Signal to Noise Ratio (SNR). By providing the algorithm an expected SNR R, it is possible to adjust the sampling demands since the use of very high quality filters is often an overkill due to the relative tolerance to leakage the post-filtering operations have, as shown in algorithm 3. Given a signal x the SFT algorithm repeatedly performs a support detection and magnitude estimation, keeping only the $2K$ greatest frequencies in the end. In the following iterations, it repeats these steps with a lower SNR, while subtracting from the samples the influence of the currently detected result [34].

Algorithm 3: SFT Algorithm

Data: x, K, R
Result: $\tilde{X}$

$R_0 = R$;
$\tilde{X} = 0$;
$r = O(\text{loglog}R)$;
for $i = 0$ **to** $r - 1$ **do**
  $\tilde{X} = SFTCore(x, \tilde{X}, 3K, R_i)$;
  $\tilde{X} = prune(\tilde{X}, 2K)$;
  $R_{i+1} = c\sqrt{R_i}$;

Algorithm 4: SFTCore Algorithm

Data: x, $\tilde{X_R}$, $3K$, R
Result: $\tilde{X}$

$B = K/\alpha$;
$\tilde{X} = \tilde{X_R}$;
$r = O(\text{loglog}R)$;
for $i = 0$ **to** $r - 1$ **do**
  $K_i = O(K4^{-i})$;
  $S = DetectSupport(x, \tilde{X_R}, B, R)$;
  $\tilde{X} = Estimate(x, \tilde{X_R}, S, B, 13, 3K_t, R)$;

2.3.1 Fourier Hashing

Instead of handling the whole signal, SFT generates several hashes $u_{a,b}$ of the signal, each with a size of B buckets, where B is in the order of the expected signal sparsity K. In each of the B buckets is the sum of N/B random frequencies. A high magnitude frequency is defined as well-hashed if is the only high magnitude frequency hashed to that bucket and that frequency does not fall into the tail of the convolution filter G used, as it may be considered noise. The probability a given frequency is well-hashed is proportional to $\alpha = B/K$ and R. If this is the case, there is little noise to it, and the detection and recovery are trivial.

The hashing process, shown in algorithm 5, is done by first applying random a permutation $P_{a,b}(x)[i] = x[a(i + b)]$ to the signal, being a and b random number in the space of $[1..N]$, with a and N co-prime. This type of permutation results in a similar permutation being applied in the Fourier Space: $F(P_{a,b}(x))[a(i + b)] = F(x)[i]\omega^{aib}$, where $\omega^i = e^{-2\pi i/N}$. This process aims to spread high contribution frequencies in order to avoid bucket collision, and is shown in figures 2.4 and 2.5. By applying a permutation to the signal, the relevant frequencies randomly distributed across the spectrum, increasing the probability of successful detection and reconstruction.

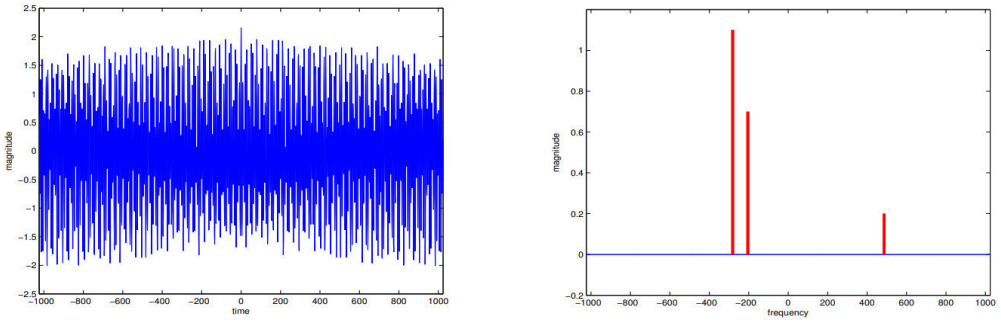


Figure 2.4: Original example signal

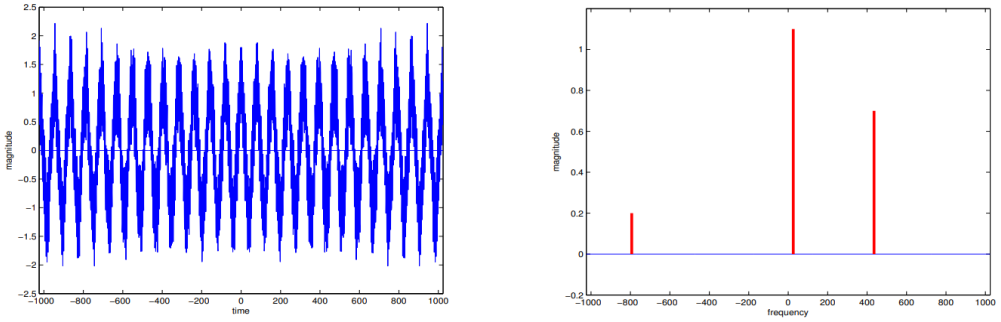
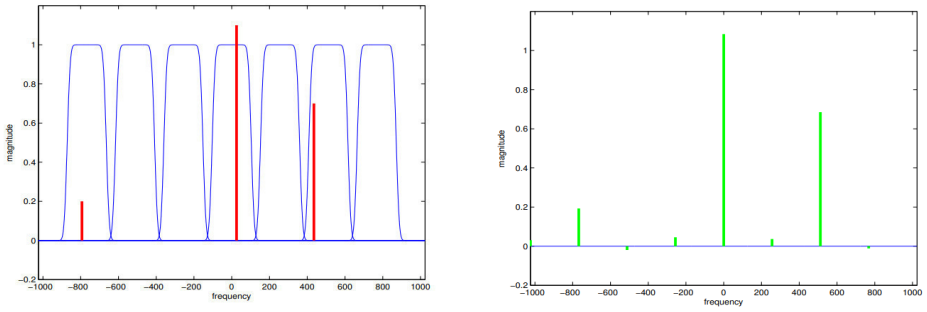


Figure 2.5: Signal from figure 2.4 after applying a permutation

After the permutation the signal is then multiplied by a filter G with a support in the order of B, whose the Fourier Transform is a flat window function. Finally, the signal is then aliased to B terms and the Fourier Transform of the filtered and aliased signal computed. The multiplication and aliasing in time space are equivalent to convolution and sub-sampling in Fourier Space, which cause our final hash $u_{a,b}$ to contain in each bucket the result of the convolution by $F(G)$ at the centre of each bucket, as shown in the figure 2.6.



(a) Filter convolution applied in Fourier Space (b) Final hashing result

Figure 2.6: Example of the filtering process in Fourier Space

Since it is not possible to use perfect filters due to sampling constraints, this process can introduce a small amount of leakage, witch translates in a given frequency contributing to neighbouring buckets other that the one it is hashed to, due to the tail of the filter in Fourier extending beyond the bucket width.

Algorithm 5: HashToBins Algorithm

Data: x, $\tilde{X}$, a, b, B, R
Result: u

$G = WindowFunction(B, R)$;
$u = 0$;
$w = \|G\|_1$;
$y = P_{a,b}(x - \tilde{X})G$;
$F_y = FFT_w(y)$;
for $i = 0$ **to** w/B **do**
  $u = u + F_y[iB; (i+1)B - 1]$

Given the previous permutation, it is possible to know in which bucket it has been hashed to by $h_{a,b}(i) = \text{round}((ai \bmod N)\, N/B)$. We also know the distance each frequency is from the centre of the bucket by $o_{a,b}(i) = (ai \bmod N) - h_{a,b}(i)(N/B)$.

2.3.2 Support Detection

The support detection algorithm used by the SFT (algorithm 6) leverages the time shift property of the Fourier transform. Given two permutations $P_{a,b}(x)$ and $P_{a,b+t}(x)$, with a, b and t randomly chosen, which hash the same frequencies to the same buckets, they differ from each other by a factor of a phase shift, which is dependent on t, the time shift difference between the two hashes, and the frequencies present in each bucket. If a given frequency i is well hashed and the signal is completely sparse, that factor would be exactly ω^{ait}, and the frequency can be detected by calculating the argument of the phase shift and reverting the permutation [26].

In case the signal is not completely sparse however, due to the noise low contribution frequencies add to the phase shift estimation, it is not possible to calculate this phase shift factor through one single time shift. Instead, a process similar to a search is performed, where it is assumed that the high contribution frequency is located at the centre of each bucket, and in each iteration we calculate two new hashes with a decreasing random time shift t between them. The current estimation l is then updated by a factor of $o = \frac{pN/2\pi - (tl \bmod N)}{t}$ where $p = \phi(u_{a,b}/u_{a,b+t})$, an estimate of $(ai \bmod N)t2\pi/N$ [34]. The magnitude of the time shifts t and the amount of iterations required in order to accurately detect a target frequency in each bin is dependent on the contrast of the filter $\gamma = R^{-1/(40\log_2\log_2 R)}$ [34]. In the end of

this iterative process, for each of the B buckets, a candidate frequency is detected.

Algorithm 6: DetectSupport Algorithm

Data: x, $\tilde{X}$, B ,R

Result: S

$N = DIM(x)$;
$c = log(log(N/B))log(\alpha)/log(R)$;
$\gamma = R^{-1/(40log_2(log_2(R))}$;
$\delta = \gamma^{1/10}$;
$t_{max} = O(log_{1/\delta}(N/B))$;
$a = \text{RandCoprime}(N)$;
$S = jN/B : j \in [0, B-1]$;
for $t = 0$ **to** $t_{max} - 1$ **do**
    $f_t = \text{rand}(B\delta^{-t}/8, B\delta^{-t}/4)$;
    for $i = 0$ **to** $c - 1$ **do**
        $b = \text{rand}(0, N-1)$;
        $u_1 = \text{HashToBins}(x, \tilde{X}, a, b, B, R)$;
        $u_2 = \text{HashToBins}(x, \tilde{X}, a, b + f_t, B, R)$;
        $m_i = \text{arg}(u_2/u_1)$
    $m = \text{Median}(m_i) : i \in [0, c-1]$;
    $o = \frac{mN/(2\pi) - f_t S \bmod N}{f_t}$;
    $S = S + o$;

2.3.3 Magnitude Estimation

In the case the signal is exactly sparse, one single hash is required to recover the magnitude of a given well-hashed frequency, since it is only required to divide the value of the bucket a frequency is hashed to by the value of the filter at the distance said frequency is to the centre of the bucket [26].

For the general case, the process is similar but a certain number of random hashes are taken, and for each the estimation above is calculated, being the final result the median of all estimates (algorithm 7) [34]. The median is used instead of the average because it is more tolerant to outliers in situations of

bucket collision.

Algorithm 7: Estimate

Data: x, $\tilde{X}_{old}$, S, B, T ,K, R

Result: $\tilde{X}_{new}$

for $t = 0$ **to** $T - 1$ **do**

  $a = \text{RandCoprime}(N)$;

  $b = \text{rand}(0, N - 1)$;

  $u = \text{HashToBins}(x, \tilde{X}, a, b, B, R)$;

  $\tilde{X}_t = F(G)_{o_{a,b}(i)} u_{h_{a,b}(i)} \omega^{-ai}$ for all $i \in S$;

$\tilde{X}_{new} = \text{Sparsify}(\text{Median}(X_t), K)$ for all $t \in [0, T - 1]$

2.3.4 Filter Construction

The filters used in this algorithm are flat window functions with a width of N/B in the Fourier domain and must have a support in the order of B in time domain. A simple yet effective way of constructing such filter is to convolve a box-car filter of width B with itself an even number of times t proportional to log R, and finally multiply that by a sinc (i/B). This generates a filter in time domain with a width of tB. The Fourier version of the filter is a sinc multiplied by itself t times, convolved with a boxcar filter if width N/B. This process allows the efficient construction of adjustable quality filters, which are sharper and more sample heavy the the higher R, and consequently t is [34].

2.4 Summary

In the first section of this chapter, Path Space integration algorithms were presented, as well as their main advantages and disadvantages. These algorithms provide the foundation for sampling paths to be supplied to sparse reconstruction algorithms.

In the second section of this chapter CS was presented. CS is able to fully recover a signal, even if sampled with a frequency lower than the Nyquist rate, as long as it admits a sparse representation in an appropriate basis, and is sampled in a basis incoherent with its sparse representation basis. It was also presented how CS can robustly handle approximately sparse signals and noise. Finally, the current applications of CS to rendering were presented.

Finally, in the third section of this chapter the SFT algorithm was presented. The SFT algorithms allow the estimation of a sparse signal in Fourier space using sub-linear time and sample complexity. This is achieved by hashing the signal into a much smaller size set of buckets. The support detection is performed through the use of time shifts and evaluating the corresponding Fourier phase shifts. Magnitude estimation uses several hashes of the signal and calculates a median of the hashed result for each frequency. The filters used in the hashing process can be adjusted to the SNR of the signal allowing for a more efficient sampling.

Chapter 3

Methodology

The main goal of this work is to reduce the execution time required to render an image by reducing the sampling rate through the use of sparse reconstruction algorithms. This reduction in sampling rate however requires an additional computational step in order to fit the sparsely sampled data to a model, our representation basis. In order to apply such algorithms, first a sparse representation basis must be selected. Also, the signals of interest must be parametrized and discretised before applying these algorithms. This chapter presents the reasoning behind those design decisions through the developed work, as well as an outline of the proposed experiments.

3.1 Basis Selection

The sampling process of light transport paths is a point-based sampling approach, thus getting a sample which is a linear combination of multiple m points is as costly as getting m point samples, making the point sampling basis the only reasonable approach. For such sampling basis, the Fourier space provides maximum incoherence [9], and thus, more suited for use in CS, as well as admitting sub-linear complexity algorithms as the SFT. Not only that, but according to Sen et al. [64], sparsity in the Fourier space increases with an increase of dimensionality, which for Path Space can be arbitrarily large, so a high level of sparsity is expected. Also, the integration process after reconstruction is very efficient due to the basis being analytically integrable [64], as well allowing the process to be calculated using the convolution theorem (as presented in chapter 4). Motivated by these facts, this thesis uses the Fourier basis for analysis.

3.2 Path Space Parametrization

Before applying the CS pipeline over the path space, it requires some kind of parametrization in order to define how to identify and represent each path in this space. The choice of the parametrization is crucial as it may condition sampling techniques and the efficiency of CS. A path of length n is most commonly described as a sequence of points x_0 .. x_n in which x_0 is a point on a light source and x_n is a point on the camera lens (see section 2.1). This representation is however inappropriate since most of these

sequences of points do not represent valid light transport paths, both due to occlusion between points or non-diffuse light sources or BSDFs. However, there are other parametrisations which have properties that are more amenable for CS reconstruction: the angle space and the primary sampling space.

3.2.1 Angle Space

Defining a light transport path in Angle Space is relatively simple. The path contains two initial coordinates that define in which point it intersects the image plane, and then, for each intersection in the scene two additional coordinates are added, θ and ϕ, that define the angles (over the hemisphere) for the next direction in the path, as shown in figure 3.1.

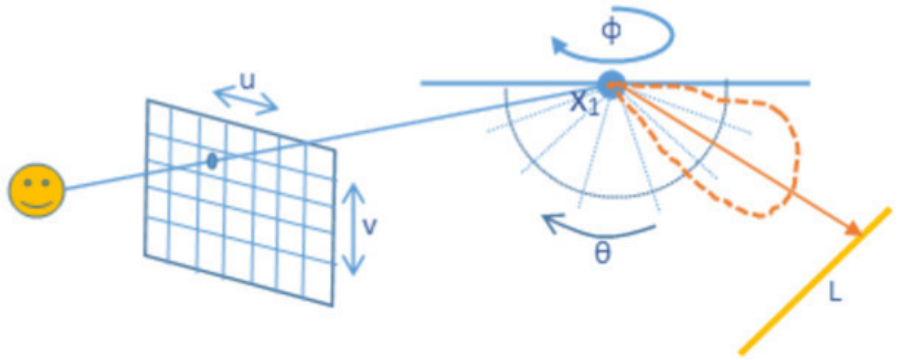


Figure 3.1: Angle Space parametrization of a path of length 2

This parametrization involves no probabilities and is independent of the underlying sampling algorithm. However, this scheme might not be ideal for representing some kinds of surfaces, such as glossy or specular ones, In fact, for these most of the contributions are concentrated over a small angular region around the specular reflection direction, while the remaining angles contain little to no contributions. Resolution would have to be high to enable representation of samples (directions) around the specular directions, but then the majority of data points corresponding to directions away from the specular one would be always zero.

3.2.2 Primary Sample Space

Parametrizing the path space in Primary Sample Space (section 2.1.6) may provide a simple solution for CS, since such an algorithm would operate similarly to classical Monte Carlo rendering algorithms, reconstructing the signal over the Primary Sample Space and using the probability density functions as a factor of conversion between this space and the path space. This way, the discrete signal would possibly have better resolution in areas of interest, when the signal contribution might be higher. However, this implies that the signal representation is completely dependent on the underlying path sampling algorithm.

This parametrization also allows the decomposition and separate analysis of different lighting phenomena. For example, it allows analyzing the path space as a whole up to a given depth, but it also allows studying a particular phenomenon by only selecting a subset of coordinates. I.e., it is possible to study

depth of field by selecting the random numbers used to sample the image plane and lens point, or the indirect lighting by selecting the image plane and secondary direction ray random numbers.

Given all this, the Primary Sample Space was chosen as the coordinate system for our case studies, as it allows a simple implementation of the algorithms, as well as an easy method of discretization: for any given sample, the coordinates in primary sample space are multiplied by the desired discretization level in each dimension and rounded down afterwards, thus obtaining its coordinates in the signal.

3.3 Proposed Experiments

The first thing that needs to be evaluated before attempting to use sparse reconstruction algorithms is the sparsity of the signals studied using appropriate metrics. These metrics must be evaluated in both the original space and in the Fourier basis in order to verify whether sparse reconstruction approaches are viable in this context. Also, image quality metrics of ideal sparse approximations of the signals of interest must be evaluated in order to evaluate the effect of reconstruction on the final image quality.

Based on the results of this evaluation, the sparsity of the studied signals is leveraged through the use of sparse reconstruction techniques, SFT and CS, for both reconstruction and importance sampling. For reconstruction, the results of the sparse reconstruction algorithms are integrated and an image is generated directly. For importance sampling, the results are used as importance maps which are used in an unbiased Monte Carlo process. Due to the different complexity of the algorithms evaluated, they were applied to different case studies. While the SFT was applied to a larger signal, a limited length path space, CS was applied to smaller 4-dimensional signals involving depth of field and indirect lighting.

For direct reconstruction an integration operation is applied upon the K sparse approximation returned by these algorithms. An improved method for integration that takes advantage of the Fourier convolution theorem was developed and compared to previous alternatives regarding time and memory usage.

Chapter 4

Fourier Integration

Suppose that the integral of a multidimensional signal is required across most of its dimensions after a sparse reconstruction algorithm in the Fourier basis is used. For example, the integral calculated for each pixel across every dimension of Path Space after using said reconstruction.

Consider that the signal has D dimensions, each discretized into s_m points dimension $m = 0 \dots D-1$. Let N be the total number of points in the signal domain, $N = \prod_{m=0}^{D-1} s_m$, and let I identify the dimensions over which the integral is calculated (the integration domain). Such integral can be calculated as a sum of every point within the integration domain.

$$\int_{D-I} \int_{D-I+1} \dots \int_{D-1} f(z)dz \approx \sum_{i_{D-I}=0}^{s_{D-I}} \sum_{i_{D-I+1}=0}^{s_{D-I+1}} \dots \sum_{i_{D-1}=0}^{s_{D-1}} f(i_0, i_1, ..., i_{D-1}) \tag{4.1}$$

Let P be defined as the number of points across the non-integrated dimensions, and the number of integrals whose value is required. Calculating the integrals according to equation 4.1 requires the storage of the whole signal (i.e., N values) and a number of sums proportional to N (for each of the P integrals, N/P sums are required), which presents a time and memory complexity of $O(N)$. Besides, since the signal is represented in the Fourier basis, it needs to be transformed into its original domain before integration, which has a complexity of $O(N\log(N))$.

4.1 Fourier Time Integration Theorem

After the execution of the sparse reconstruction process, a compressed representation of the signal is obtained in the form of

$$f(z) \approx f_K(z) = \sum_{i=0}^{K-1} \left[c_{s[i]} \cdot \psi_{s[i]} \right] \tag{4.2}$$

where K is the number of coefficients used in order to represent the signal, c the values of the coefficients of the signal in the Fourier Basis, s the support vector and ψ the basis functions.

In order to calculate the integral of this signal, it is possible to use the sum of the integral of each of

the basis functions [64].

$$\int_{t_0}^{t_1} f(z) \approx \int_{t_0}^{t_1} f_K(z) = \sum_{i=0}^{K-1} \left[c_{s[i]} \cdot \int_{t_0}^{t_1} \psi_{s[i]} \right] \tag{4.3}$$

In the case of the Fourier basis, these are analytically integrable. Which means that in order to integrate the signal it is only required to integrate K basis functions. By the Fourier Time Integration Theorem, we have that the integral of a function in time space corresponds to the following in in frequency space:

$$F\left(\int_{-\infty}^{t} f(z)dz\right)(j\omega) = \frac{1}{j\omega}F(f)(j\omega) + \pi F(f)(0)\delta(\omega) \tag{4.4}$$

where F is the Fourier transform (with $F(f)(j\omega)$ being the Fourier transform of the function f in a given point of the frequency spectrum), j is the imaginary unit, ω is the angular frequency $2\pi k/N$ for frequency k and N the signal size [60]. Finally, δ is a dirac-delta function, an impulse which is zero everywhere except the origin. This is equivalent to:

$$\int_{-\infty}^{t} f(z)dz = F^{-1}\left(\frac{1}{j\omega}F(f)(j\omega) + \pi F(f)(0)\delta(\omega)\right)(t) = \sum_{i=0}^{N} \frac{c_i}{j\omega_i}\psi(t) + F(f)(0)t \tag{4.5}$$

where ω_i is the angular frequency for frequency i.

Since only K frequencies are required to represent the signal, the integral can be calculated as:

$$\int_{-\infty}^{t} f(z)dz = \sum_{i=0}^{K-1} \frac{c_{s[i]}}{j\omega_{s[i]}}\psi_{s[i]}(t) + F(f)(0)t \tag{4.6}$$

By dividing each of the Fourier basis functions by j times their angular frequency along each dimension, the integral is obtained, and the final results can be calculated by a difference of integrals.

$$\int_{t_0}^{t_1} f(z)dz = \int_{-\infty}^{t_1} f(z)dz - \int_{-\infty}^{t_0} f(z)dz \tag{4.7}$$

This process bounds memory usage to $O(K+P)$ and execution time to $O(KP)$ where K is the number of coefficients and P the total number of integrals calculated, in the case image integration, the number of pixels. This is a significant efficiency improvement since $N \gg K$ and $N \gg P$ potentially, since signal size increases exponentially with dimensionality.

However, due to the efficiency of Fast Fourier Transform implementations, calculating each of the basis functions for a given number of points (as required by equations 4.6 and 4.7) may actually be slower than using a full inverse Fourier Transform:

$$\int_{-\infty}^{t} f(z)dz = F^{-1}\left(\sum_{i=0}^{K-1} \frac{c_{s[i]}}{j\omega} + \pi F(f)(0)\delta(\omega)\right)(t) \tag{4.8}$$

This approach has a time complexity of $O(N\log(N))$ and $N \gg K$ and $N \gg P$, but the greatest

drawback is the need to store the whole signal in memory, which for larger signals may not be possible.

4.2 Convolution Fourier Integration

Given a discrete signal $f(z)$ defined over D dimensions, whose integral is wanted over I dimensions $D-I$ to $D-1$.

$$\int_{D-I}\int_{D-I+1}...\int_{D-1} f(z)dz \approx \sum_{i_{D-I}=0}^{s_{D-I}} \sum_{i_{D-I+1}=0}^{s_{D-I+1}} ... \sum_{i_{D-1}=0}^{s_{D-1}} f(i_0, i_1, ..., i_{D-1}) \tag{4.9}$$

This integration process can be viewed as convolution of the signal with a box-car filter of size 1 in the dimensions 0 through $I-1$, and of size s_d for every dimension $D-I$ to $D-1$, where s_d is the number of points of the discrete signal domain along dimension d.

$$\int_{D-I}\int_{D-I+1}...\int_{D-1} f(z)dz \approx f(z) * b(z) \tag{4.10}$$

where b is the box-car filter described above and $*$ denotes the convolution operation.

By the Fourier Convolution Theorem, $F(f*b) = F(f)\cdot F(b)$, this convolution process can be calculated as a multiplication of the signal in Fourier space, which is obtained in a compressed representation through sparse reconstruction algorithms, and the Fourier Transform of the box-car filter.

$$\int_{D-I}\int_{D-I+1}...\int_{D-1} f(z)dz \approx F^{-1}(F(f(z)) \cdot F(b(z))) \tag{4.11}$$

This box-car b is in fact a constant function in every integrated dimension, since it is the same size as the signal itself. The Fourier Transform of such filter is $F(b) = \delta(b)$, a dirac-delta function with the value zero in every coordinate except the origin. This means all non zero frequencies over the integrated dimensions get discarded. Given the signal in its Fourier representation, it is possible to calculate the full integral over I dimensions by calculating the $D-I$-dimension Inverse Fourier Transform of all the Frequencies $(z_0, z_1 ... z_{D-I-1}, 0, 0, ..., 0)$ for all $z_0, z_1 ... z_{D-I-1}$.

$$\int_{D-I}\int_{D-I+1}...\int_{D-1} f(z)dz \approx F_I^{-1}(F(f(z))(z_0, z_1, ..., z_{D-I-1}, 0, 0, ..., 0)) \tag{4.12}$$

This means that in order to calculate the final image, after the reconstruction process it is only necessary to select the frequencies $(x, y, 0, .., 0) \forall x$ and y, and then compute the final image as a 2-dimensional inverse Fourier Transform.

This is a considerable memory and time complexity improvement, since there is no need to reconstruct the whole signal and calculate each pixel as a difference of integrals, but instead a simple 2-dimensional Inverse Fourier Transform which ignores all provided frequencies whose coordinate is different than zero in the 2 first dimensions. This results in a time complexity of $O(P\log(P))$ and a memory complexity of

$O(P)$ where $P \ll N$ is the number of integrals being calculated.

Table 4.1 presents an overview of the time and memory complexities of all integration methods discussed.

Integration Algorithm	**Time Complexity**	**Memory Complexity**
Sum Integration	$O(N)$	$O(N)$
Fourier Time Integration (no FFT)	$O(KP)$	$O(K+P)$
Fourier Time Integration (with FFT)	$O(N\log(N))$	$O(N)$
Convolution Integration	$O(P\log(P))$	$O(K+P)$

Table 4.1: Complexity Comparison of the Analysed Integration Methods

In order to evaluate the improvement of the proposed approach, it was compared with the previously mentioned approach proposed by Sen et al. [64], as well as a naive integration which reconstructs the whole signal and calculates the integral as a the sum over the integration domain.

4.3 Methodology

In order to perform this comparison, a random sparse 4-dimensional signal is generated and integrated over the last two dimensions. The signal generation in Fourier space is given by the pseudo-code presented below:

```
N = s_d ^ 4;
K = N/10000;
support = randperm(N, K); %select the support
values = rand(size(support)); %pick K random values from zero to one
signal = zeros(s_d, s_d, s_d, s_d); %initialize signal to zero
signal(support)=values;  %set values
```

Integration execution times were evaluated with regard to the signal size $N = s^4$, where s is the size of each dimension, which varied from 32 to 128. The sparsity rates for all signals are of $1/10^4$, value chosen given the sparsity rates present in the test light transport signals evaluated (sections 5.4, 7.1.3 and 7.2.2). The number of calculated integrals P is s^2, as all integrals across the last two dimensions are calculated. Memory requirements for each method were also evaluated.

4.4 Experimental Results

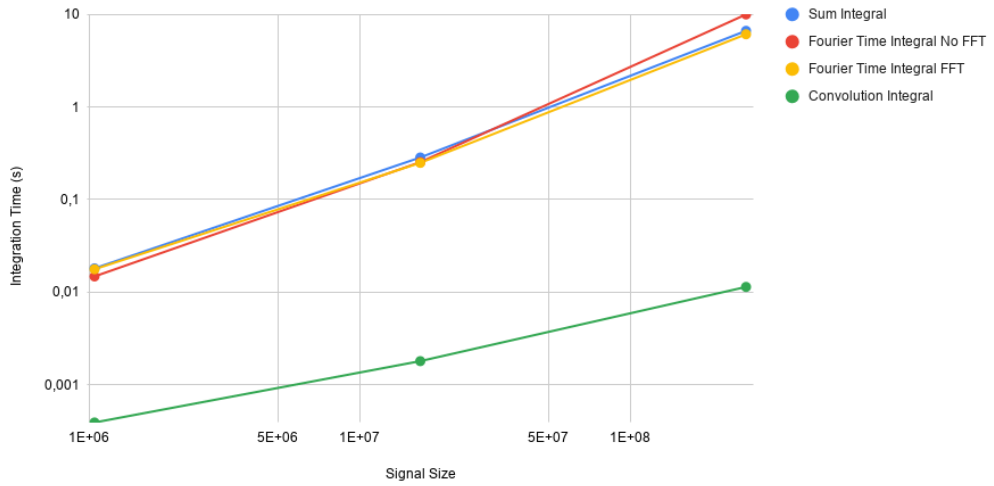


Figure 4.1: Execution time of the integration process with regard to signal size

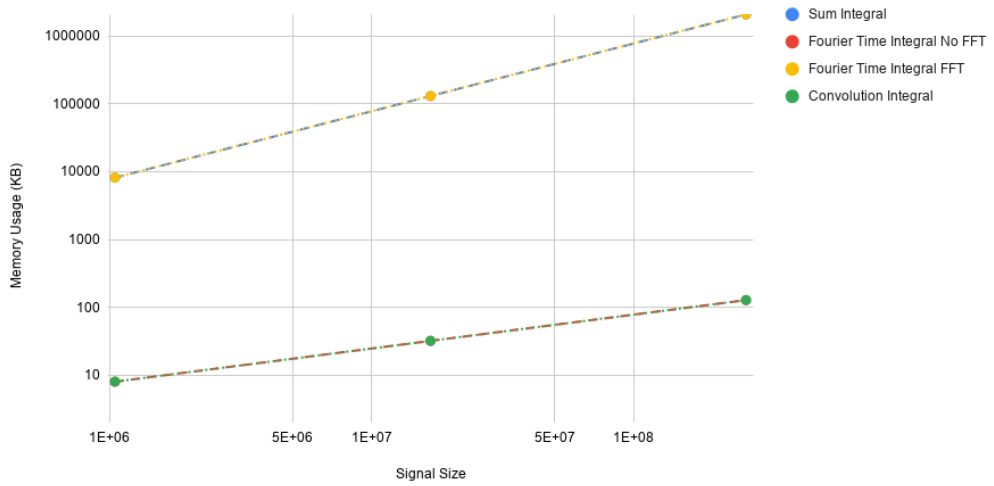


Figure 4.2: Memory usage of the integration process with regard to signal size

Given the results presented in figures 4.1 and 4.2, it is possible to observe that the proposed method achieves great improvements upon previous approaches both in execution time and memory usage. The Fourier Time Integration method, although faster than a naive approach, still requires a full signal representation, and thus not being viable for larger signals. The proposed method allows the integration of potentially larger signals without the need of storing a full reconstruction. Also, the integration for time and memory usage do not depend on the signal sparsity, only on the number of integrals being calculated, pixels in the case of image integration.

4.5 Summary

Through this chapter an efficient method was proposed for integrating sparse Fourier signals using a convolution approach. This approach provides a fast and memory efficient way of integrating sparse reconstruction results when compared to previous approaches. The proposed approach memory usage is proportional to the number of integrals (image pixels) and its execution time is bound by $P\log(P)$, where P is the number of pixels, which is a significant improvement upon previous approaches, as well as managing to do so while leveraging the efficiency of FFT implementations.

Chapter 5

Sparsity Analysis of Path Space

Sparse reconstruction algorithms require the signal of interest to be sparse in a given representation basis, while being dense in the sampling basis. Thus, in order to assess whether a sparse reconstruction approach is feasible, a sparsity analysis was performed beforehand in both the sampling basis (point-based sampling) and representation basis (Fourier).

5.1 Sparsity Metrics

In order to have meaningful results regarding signal sparsity, robust sparsity metrics are required, as well as a methodology to assess how sparse signal reconstruction affects image quality.

Although sparsity can simply be measured as the l_0 norm of a signal, this metric's usefulness is limited only to sparse signals, not applying to compressible signals. Thus, in order to have quantifiable results regarding signal sparsity, some meaningful metric must be used to evaluate this property.

Intuitively, a distribution is sparser the more its energy is concentrated on a small number of coefficients, and an ideal sparsity metric should evaluate this. According to Hurley and Rickard [32], such a sparsity metric should satisfy the following set of properties:

Robin Hood Decreasing the magnitude of high contribution coefficients while increasing the value of low contribution coefficients decreases sparsity.

Scaling Invariance Multiplying all values of the signal by any constant different than zero does not affect sparsity.

Rising Tide Adding a constant value to all coefficients decreases sparsity.

Cloning If there is another signal with similar energy distribution, the sparsity of any signal is the same as the union of the two.

Bill Gates As one coefficient grows infinitely large, sparsity increases.

Babies Adding zero contribution coefficients to an existing signal increases sparsity.

According to Hurley and Rickard [32], the only sparsity metric that satisfies all of these properties is the Gini Index [23] [15].

The Gini Index was proposed as an economics metric of the inequality of wealth distribution and ranges from 0 to 1, where 0 corresponds to an egalitarian distribution of wealth (least sparse distribution possible) and 1 corresponds to one individual possessing all the wealth (sparsest distribution possible). Given a positive vector c ordered ascendantly, the Gini Index $G(c)$ can be calculated as:

$$G(c) = 1 - 2\sum_{i=1}^{N} \frac{c_i}{\|c\|_1}\left(\frac{N - i + \frac{1}{2}}{N}\right) \tag{5.1}$$

Given the ideal set of properties it satisfies, the Gini Index applied to both coefficient magnitude and energy (magnitude squared) was the metric of choice for this evaluation. While the coefficient magnitude is important for detection purposes , thus being able to distinguish two different coefficients and selecting the appropriate one, coefficient energy is the most essential for signal reconstruction, thus the importance of analysing both.

5.2 Image Quality Metric

Since the focus of this work is image reconstruction, it is also important to evaluate the impact of selecting only a small set of frequencies has on image quality. In order to assess this, the reference signals were projected into Fourier space, and while selecting only a fraction of the largest contribution frequencies, every other is set to zero. Afterwards, the image quality of the hard-threshold reconstructions was evaluated using the Structural Similarity Index (SSIM) to compare them to the same reference image.

SSIM is an image quality metric that compares a target image against a reference image and aims to improve upon other image quality metrics such as Root Mean Squared Error (RMSE) and Peak Signal to Noise Ratio (PSNR) by measuring not only the magnitude of the differences between the images, but the structure present in the neighbourhood of each pixel, not only detecting differences in magnitude but also in texture. This metric is first calculated for each pixel using a Gaussian window of 11x11 pixels, for which the following formula is applied:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c2)} \tag{5.2}$$

where:

- μ is the average within the window
- σ_x and σ_y are the variance for the windows in each image
- σ_{xy} is the co-variance x and y
- $c1 = (k_1 L)^2$ and $c2 = (k_2 L)^2$ are two variables to stabilize the division with weak denominator, where L is the dynamic range of the image

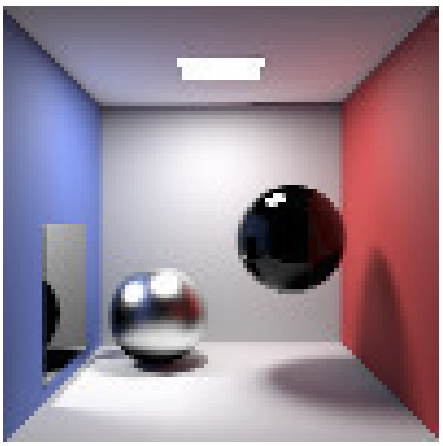
- $k_1 = 0.01$ and $k_2 = 0.03$ by default

The final metric is the average of the similarity across the whole image [77] [78].

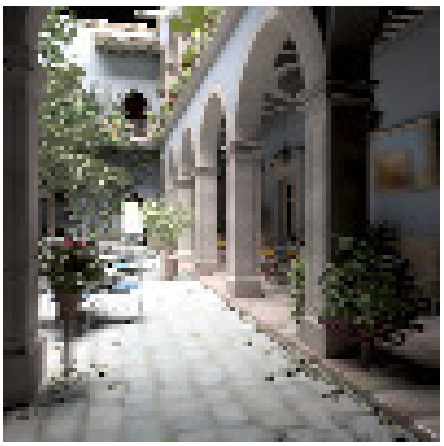
5.3 Methodology

For this analysis, two test scenes were considered, the Cornell Box and San Miguel (see figure 5.1). These were chosen due to their differences in illumination conditions, as well as geometric complexity, and are considered representative of the problem being tackled. For both scenes, a Primary Sample Space of maximum depth 3 was discretised with a resolution of 100x100 in the image plane and 4 in every other dimension. These values were chosen in order to provide a viewable image while still managing to achieve a path depth of 3 within a reasonable memory usage. Considering that for each bounce after the first, two additional rays are generated, one to sample the light source and another to sample the BSDF, each bounce adds 4 new dimensions to the signal. This results in a signal with a total of 10 dimensions and a size of $6,5536 \times 10^8$ points (10^4 for the image plane times 4^8 for the 8 directional dimensions). For both scenes, the analysis of a similar Primary Sample space which does not include direct lighting was also performed, in order to assess whether sparse reconstruction algorithms could be applied only over the indirect lighting component of a scene. This indirect lighting analysis uses the same dimensionality as the first space, instead of sampling the light source on the first bounce, an additional BSDF sample is taken in the last intersection. Each point in these signals was estimated as the average of 64 randomly jittered samples.

The Gini Index for magnitude and energy was measured for this signal in its original space, in order to evaluate its density, as well as for its Fourier Transform in order to evaluate its sparsity. These metrics were taken individually for each colour channel. Finally, several hard thresholds were performed, selecting a small percentage of the highest contribution frequencies, and the image quality of the integration of these hard thresholds was compared against the reference using SSIM.



(a) Cornell Box



(b) San Miguel

Figure 5.1: Test scenes used for the sparsity analysis

5.4 Experimental Results

5.4.1 Gini Measurements

In table 5.1, are shown the sparsity metrics for the Fourier projection of Path Space. Values closer to 1 mean sparser distributions. The same metrics were also taken for the primary sample space, which are shown in table 5.2.

Scene	Channel	Gini Magnitude	Gini Energy
Cornell	R	0,849482	0,999931
	G	0,845120	0,999930
	B	0,848392	0,999931
Cornell Indirect Only	R	0,705334	0,995412
	G	0,689701	0,996581
	B	0,703650	0,997622
San Miguel	R	0,861825	0,999223
	G	0,855371	0,999230
	B	0,840856	0,999225
San Miguel Indirect Only	R	0,874597	0,996286
	G	0,862031	0,995836
	B	0,843055	0,997143

Table 5.1: Sparsity Metrics in Fourier Space

Scene	Channel	Gini Magnitude	Gini Energy
Cornell	R	0,955236	0,990213
	G	0,970621	0,990520
	B	0,955462	0,990208
Cornell Indirect Only	R	0,822949	0,973289
	G	0,891793	0,985055
	B	0,907133	0,973441
San Miguel	R	0,973231	0,974978
	G	0,973341	0,974935
	B	0,972836	0,974283
San Miguel Indirect Only	R	0,954373	0,980880
	G	0,948265	0,977941
	B	0,943414	0,986865

Table 5.2: Sparsity Metrics in Primary Sample Space

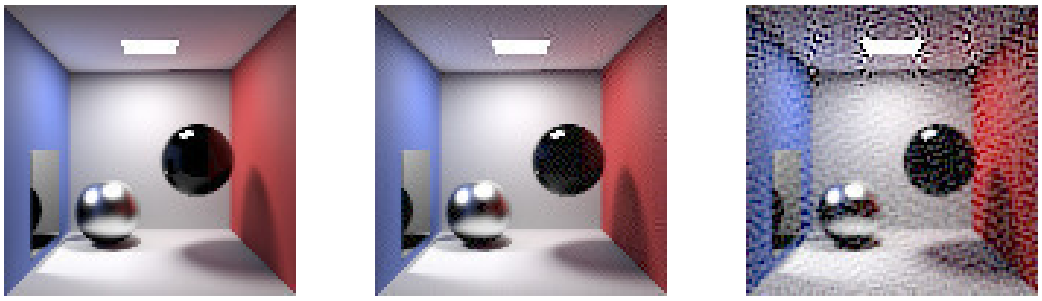
From tables 5.1 and 5.2, it is possible to observe that indirect lighting alone is less sparse than full

light transport in the Fourier domain, which forces the application of sparse reconstruction algorithms to all lighting, direct and indirect. Also, although energy is sparsely distributed in the Fourier space for both scenes, coefficient magnitude is not so sparse. Since the signal energy is sparsely distributed, it should be possible to represent the signal with a small number of coefficients, however, since the coefficients magnitudes are not so sparsely distributed, sparse reconstruction algorithms may struggle to detect and estimate the optimal representation.

On the other hand, Path Space in its original domain seems to be quite sparse in both coefficient magnitude and energy distribution for both scenes. In particular direct lighting takes into account the direct visualization of the light source for the Cornell Box scene, which due to its much greater radiance value compared to the rest of the scene, results on a higher sparsity value. Since sparse reconstruction algorithms require the signals of interest to be dense in their sampling representation, which is not the case in these results, it is expected these techniques do not perform to their fullest potential with these signals.

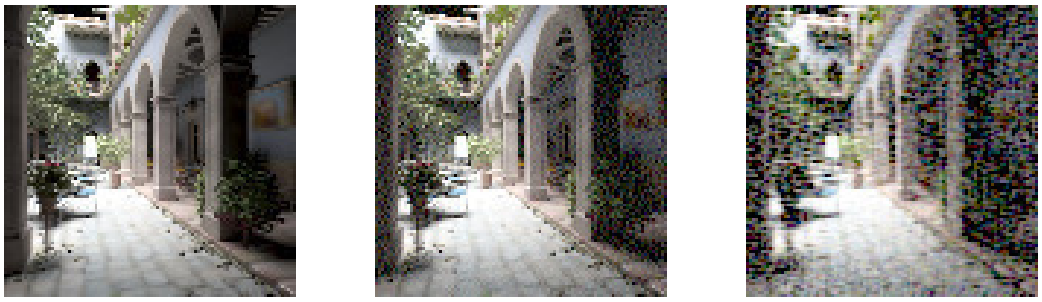
5.4.2 Image Quality Comparison

Although the metrics presented suggest that Path Space admits a compressible representation in Fourier, the image quality of hard thresholded Fourier reconstructions was evaluated. Figures 5.2 and 5.3 present some of these reconstructions, with up to $1/10^4$ coefficients used, that do not show much perceptual difference comparing to the reference.



(a) Reference image (b) $1/10^4$ Reconstruction (c) $1/10^6$ Reconstruction

Figure 5.2: Comparison of hard thresholded Fourier Space for the Cornell Box scene



(a) Reference image (b) $1/10^4$ Reconstruction (c) $1/10^6$ Reconstruction

Figure 5.3: Comparison of hard thresholded Fourier Space for the San Miguel scene

The image quality of these reconstructions was also compared using SSIM, as shown in figure 5.4.

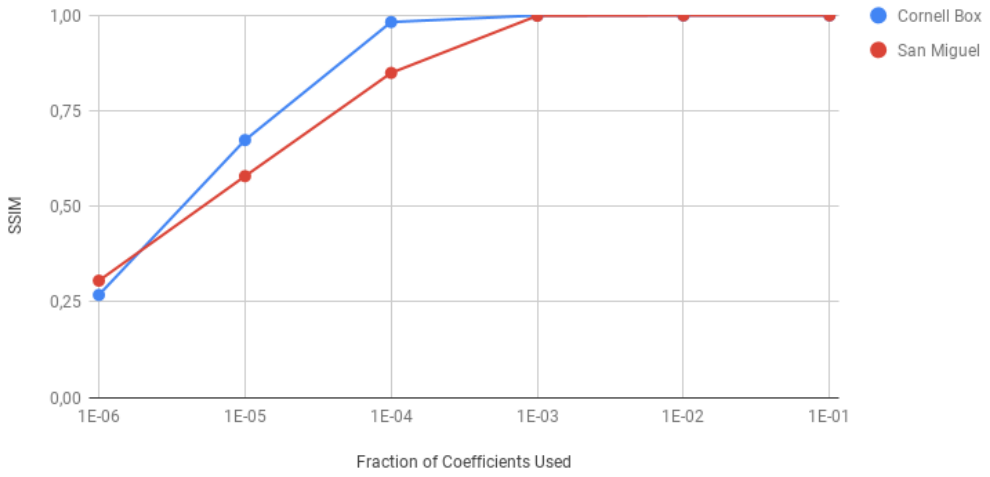


Figure 5.4: Hard Thresholded Fourier Space Reconstruction Quality

These results show that for the tested scenes, it is possible to reconstruct to attain a SSIM above 0.8 for both scenes using only $1/10^4$ of the Fourier coefficients. This provides a best case scenario of what the reconstruction algorithms can achieve, since said algorithms work with incomplete random information regarding the signal, providing only a bound on their success rate.

5.5 Summary

In this chapter, a sparsity analysis of Path Space was performed, taking into consideration meaningful metrics, namely the Gini Index and the SSIM of hard-thresholded signals. These metrics provide an insight on how effective sparse reconstruction algorithms can be in this case, and its possible impact in the final image quality.

From the results of this analysis, it is possible to observe that Path Space admits a compressible representation in Fourier, and in the best case scenario, it is possible to recover acceptable quality images using a small number of Fourier coefficients. The fact that it is heavily sparse in its canonical space is however a cause for concern, since it decreases the success rate of sparse reconstruction algorithms, and thus compromise the proposed approach.

Chapter 6

Sparse Fourier Transform Applied to Path Space

6.1 Path Space Reconstruction

Following the results of the sparsity analysis tests, which suggest that the signals of interest are sparse in the Fourier space, it was decided to test the efficiency of sparse reconstruction methods applied to Path Space. Due to the size of this problem however (10 dimensions for a total of $6,5536 \times 10^8$ points), CS approaches like CoSaMP are infeasible due to time and memory complexity issues, which scale exponentially with dimensionality. Thus, it was decided to apply the SFT algorithm in order to try to reconstruct the previously described Path Spaces, due to its theoretical lower complexity in time and memory.

6.1.1 Methodology

The SFT algorithm was applied to the scenes mentioned in 5.3 using the same parameters: maximum depth of 3, 100x100 image resolution and 4 discretization levels for each other dimension, signal size of $6,5536 \times 10^8$, and an expected sparsity of $1/10^4$. Although there are SFT algorithms applicable to high dimensionality spaces [33] [39] [40] [41], a simpler approach was used, in which the 10-dimensional FFT is computed as a 1-dimensional FFT [49].

By shifting the sampling point using the transform presented in equation 6.1, it is possible to compute a Fourier Transform of any dimension using a 1-dimensional Fourier Transform.

$$\int_{[0..1]^D} e^{2\pi i j x} f(x) dx = \frac{1}{N} \sum_{n=0}^{N} e^{2\pi i j x_n} f(x_n) \tag{6.1}$$

for all $j \in [0, s_d - 1]$ where $x_n = \frac{ng \bmod N}{N}$ and $g = (1, s_0, s_0 s_1, .., \prod_{d=0}^{D-2} s_d)$, being s_d the discretization along dimension d, and $N = \prod_{d=0}^{D-1} s_d$ the size of the signal. Afterwards, the Fourier coefficients can be re-ordered into their original dimensionality by the following isomorphism:

$$n \rightarrow n_0 + n_1 s_0 + n_2 s_0 s_1 + .. + n_{D-1} \prod_{d=0}^{D-2} s_d \tag{6.2}$$

The final integration process uses the method proposed in chapter 4, as it provides the integration result more efficiently than previous approaches.

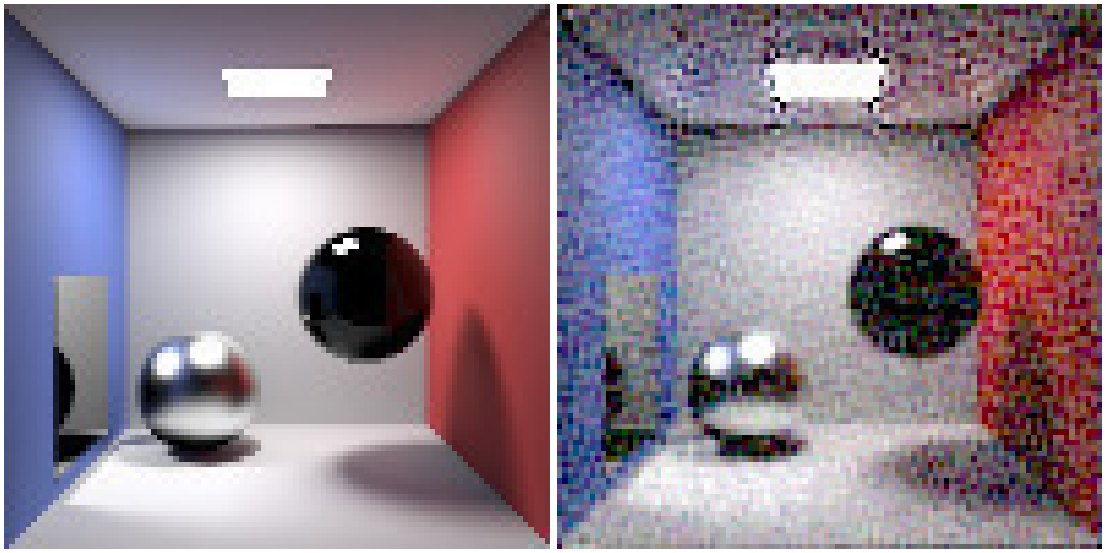
6.1.2 Experimental Results

Table 6.1 presents the sampling requirements of the SFT algorithm for recovering $K = 65536$ coefficients (the sparsity level being 10^{-4} out of $6,5536 \times 10^8$ points in the domain). $\alpha = B/K$ is a parameter of the SFT, where B is the number of buckets used.

α	**Support Detection**	**Magnitude Estimation**	**Total Samples**	**Samples per Pixel**
4	553647600	163577700	717225300	71722
16	1711275624	654311268	2365586892	236558
32	3019898520	1308622692	4328521212	432852

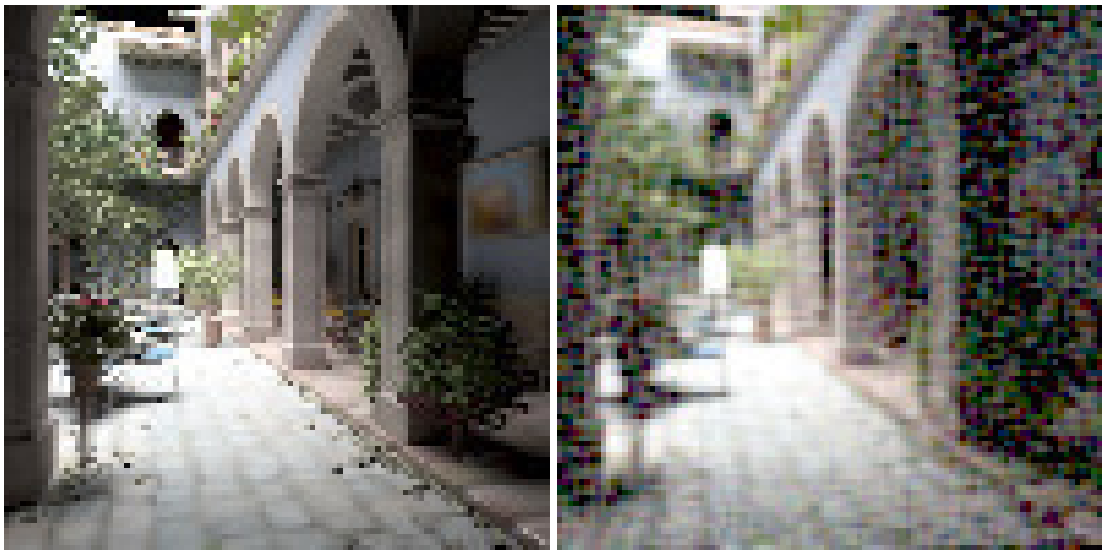
Table 6.1: Sparse Fourier Transform Sampling Requirements

The number of samples is clearly above the acceptable limits, since it exceeds the signal size. Additionally, reconstructed images using the lowest sampling rates (e.g., $\alpha = 4$) exhibit various artifacts (see figures 6.1 and 6.2), which suggests that even if the sampling demands of the SFT could be reduced, image quality would be severely compromised. This is due to the combinatorial approach the SFT follows, which trades higher sampling rates for sub-linear complexity bounds.



(a) Reference Image (b) SFT Reconstruction for an $\alpha = 4$

Figure 6.1: SFT Image comparison for the Cornell Box scene



(a) Reference Image (b) SFT Reconstruction for an $\alpha = 4$

Figure 6.2: SFT Image comparison for the San Miguel scene

Given these results, it is shown that a SFT based reconstruction of path space is infeasible due to the huge number of samples this algorithm requires, and the low quality of the lower sampling reconstructions. Unfortunately, a CS based approach is infeasible as well for such large signals, due to memory and time complexity issues.

6.2 Sparse Fourier Transform based Path Space Importance Mapping

Due to the heavy sampling requirements of the SFT algorithm, the reconstruction process is very inefficient, making it impossible to generate an image in reasonable time.

Sparse signal reconstruction using the SFT can still be leveraged if the algorithm is used to generate a low resolution importance map. This map can then be used to guide sampling of the path space within some unbiased Monte Carlo rendering process.

The importance map is generated (using the SFT) at a much lower resolution than the target image. The hypothesis is that the costs of sampling and running the SFT can be amortized when rendering the final high resolution image. This importance sampled rendering will, hypothetically, require less samples (for the same image quality) than an alternative rendering approach which does not have access to the pre-computed importance map.

Unlike Markov Chain based approaches or PSS warping, the proposed approach generates an importance map that spans all the dimensions of the target path space. Therefore different sampling probabilities are attributed to different segments of the primary sample space, based on the reconstructed values.

6.2.1 Methodology

In order to test this approach, the importance maps were rendered at an image plane resolution of 100x100 and 4 discretization levels along each of the other 8 dimensions, as in the previous tests. This signal was reconstructed using the SFT algorithm using an $\alpha = 8$. Using this reconstruction, the probability distribution function and cumulative distribution function were built.

$$pdf(x) = \frac{R(x)}{\sum_{i=0}^{N-1} R(i)} \tag{6.3}$$

where pdf is the probability distribution function and R is the reconstruction provided by the SFT.

$$cdf(x) = \sum_{i=0}^{x} pdf(i) \tag{6.4}$$

where cdf is the cumulative distribution function.

This importance map was then used to sample the scenes again, first selecting which hyper-volume to sample using the previously built probability distribution function.

$$s(\xi) = x : (cdf(x) \geq \xi \wedge cdf(x-1) < \xi) \tag{6.5}$$

where ξ is an uniformly distributed random number from zero to one. This gives the index of the hyper-volume x to sample.

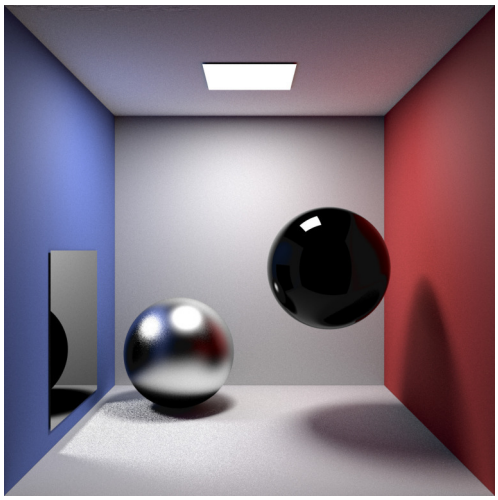
Given the hyper-volume, a sample is selected by randomly jittering the coordinates within this hyper-volume. This process was executed for image resolutions of 1000x1000, 2000x2000 and 4000x4000, using 100 samples per pixel in the importance sampling phase. This approach was compared with a Path Tracing result using the same number of total samples.

6.2.2 Experimental Results

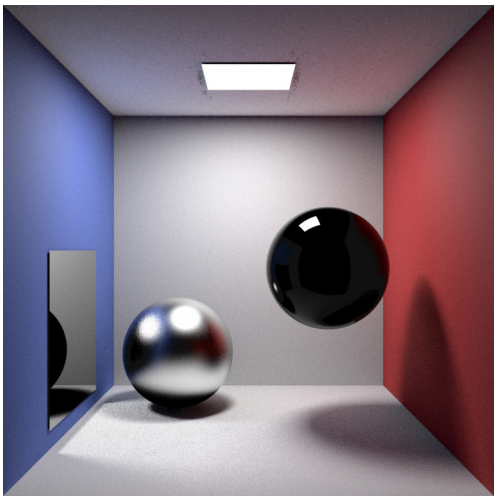
Tables 6.2 and 6.3 present (for the Cornell Box and San Miguel scenes, respectively) the number of samples per pixel, the rendering time for the SFT importance sampling algorithm and the rendering time for path tracing. Results are presented for image resolutions of 1000x1000, 2000x2000 and 4000x4000.

Image Size	Samples per Pixel	SFT Importance Sampling Time (s)	Path Tracing Time (s)
1000x1000	489	238	199
2000x2000	197	387	304
4000x4000	124	930	817

Table 6.2: Execution times for the Cornell Box scene

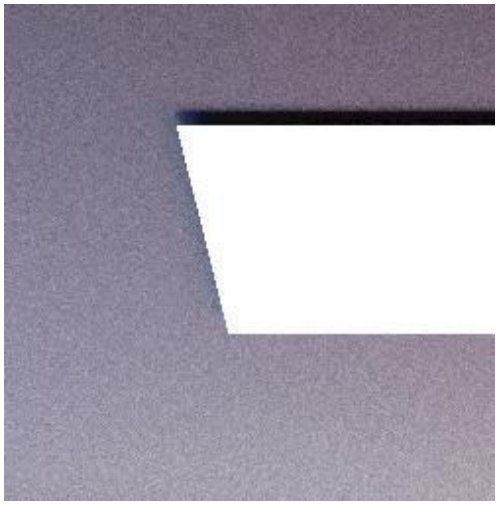


(a) Path Tracing image using 197 samples per pixel



(b) SFT Importance Sampling image using 197 samples per pixel

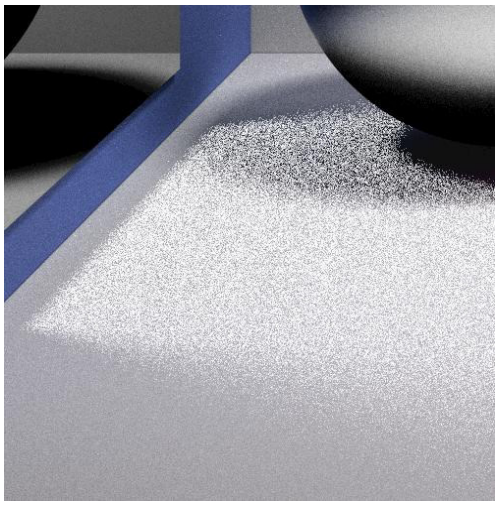
Figure 6.3: Comparison of the proposed implementation with Path Tracing for the Cornell Box scene at 2000x2000 resolution



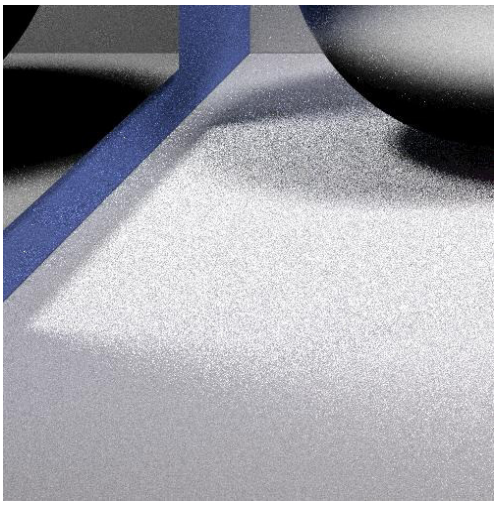
(a) Path Tracing

(b) SFT Importance Sampling

Figure 6.4: Comparison of the proposed implementation with Path Tracing for the Cornell Box scene: light detail



(a) Path Tracing image using 197 samples per pixel



(b) SFT Importance Sampling image using 197 samples per pixel

Figure 6.5: Comparison of the proposed implementation with Path Tracing for the Cornell Box scene: caustic detail

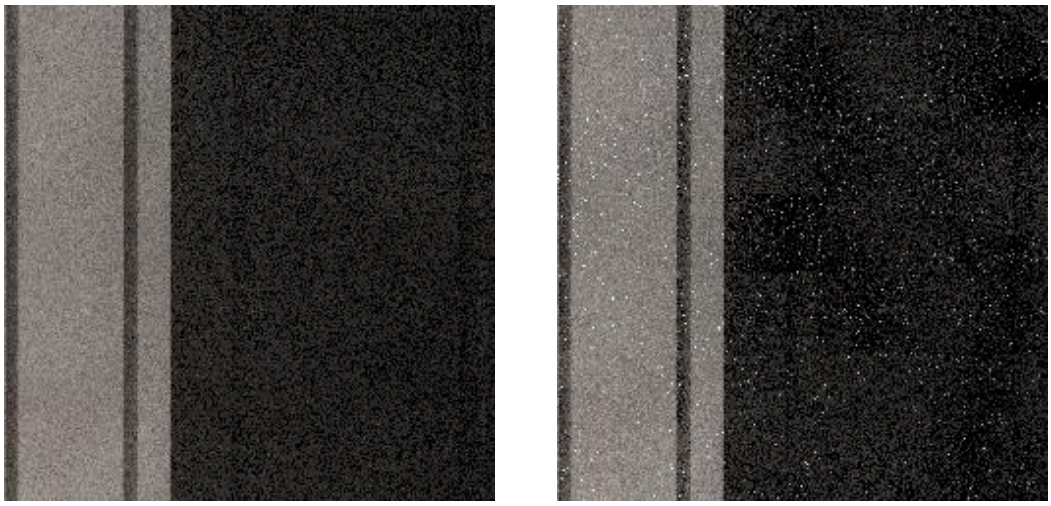
Image Size	Samples per Pixel	SFT Importance Sampling Time (s)	Path Tracing Time
1000x1000	489	1439	1203
2000x2000	197	2216	1969
4000x4000	124	5316	5172

Table 6.3: Execution times for the San Miguel scene

(a) Path Tracing image using 197 samples per pixel

(b) SFT Importance Sampling image using 197 samples per pixel

Figure 6.6: Comparison of the proposed implementation with Path Tracing for the San Miguel scene at 2000x2000 resolution



(a) Path Tracing (b) SFT Importance Sampling

Figure 6.7: Comparison of the proposed implementation with Path Tracing for the San Miguel scene: column detail

From the data in tables 6.2 and 6.3 and images 6.3 and 6.6 it is possible to observe that not only does the SFT add to the execution time, but the images resulting from the process are visibly less converged. Resulting artifacts are made more apparent in figures 6.4 and 6.7. These artefacts are similar in structure to the ones present in figures 6.1 and 6.2 in a smaller scale, which suggests such artifacts are present in the importance maps used as well.

However, the caustics in the Cornell Box scene are visibly more converged for the proposed method, especially in the area shadowed by the sphere (figure 6.5).

Given the quality of the SFT reconstructions in the previous section, it is hypothesized that the importance maps generated do not match the signal closely enough for the importance sampling process to reduce variance as expected.

6.3 Summary

Through this chapter a method for reconstructing Path Space using SFT was tested, and the results did not improve upon Path Tracing due to the heavy sampling demands of the algorithm. In order to try and amortize this issue, another method based on importance sampling was also implemented, using the SFT to generate an importance map which is then used to generate higher resolution images. This method also failed to improve upon Path Tracing but the improvements in caustics suggest that it may be worth going further in this line of research.

Chapter 7

4D Sparsity Analysis

Given the results of the previous experiments, which severely limit the potential of Path Space reconstruction due to the SFT sampling requirements, an analysis of other sparse reconstruction algorithms is required. However, CS algorithms like CoSaMP cannot be applied to high cardinality domains due to its time and space complexity, which when using the Fourier basis is in the order of $N\log N$ for time and N for memory, where N is the size of the signal. To experimentally verify the effectiveness of CoSaMP on a rendering problem it was applied to a smaller domain, in particular 4D domains such as depth of field or indirect lighting. Before testing the reconstruction algorithm, like in the previous case, a sparsity analysis was performed.

7.1 Depth of Field Sparsity

In order to simulate depth of field, Monte Carlo Integration algorithms integrate over the lens area by taking random samples distributed over the lens area. Using the coordinates of the lens point in conjunction with image plane, a 4D space is constructed, upon which CoSaMP can be applied. This case study was chosen due to both its simplicity and widespread application in rendering.

7.1.1 Gamma Mapping

For this case study, the signals of interest are shown to be quite sparse in the canonical domain (see columns "Time" and "Time Energy" and row $\gamma = 1$ in tables 7.1 and 7.2 in section 7.1.3), which limit the application of sparse reconstruction algorithms. This is believed to be due to the high dynamic range nature of the signal, which allows higher contrast between bright and dim portions of the signal. In order to try and mitigate this, the sparsity analysis tests were executed on range compressed signals using Gamma tone-mapping.

Gamma tone-mapping compresses a high dynamic range image to a low dynamic range image by applying a transformation

$$T_i = \left(\frac{I_i}{max(I)} \right)^{\gamma} \tag{7.1}$$

for a parameter $\gamma < 1 : \gamma \in \mathbb{R}$, where T is the tone-mapped image and I is the original image [61]. This effectively compresses the dynamic range of the image and reduces contrast between brighter and dimmer regions of the image.

By applying this tone-mapping process to the signal, it is expected to increase density in the canonical space while achieving even higher sparsity in the Fourier space. This however nullifies the possibility of integrating the reconstructed signal in the Fourier space directly, since the obtained sparse representation corresponds to a tone-mapped signal. Since $a^\gamma + b^\gamma \neq (a+b)^\gamma$, in order to integrate the original signal correctly, it is necessary to reconstruct the whole signal, revert the tone-mapping and then integrate. Even though explicitly representing and processing the complete signal is a drawback, the process is applied to reasonably sized signals: for this particular case study the depth of field signal is 4D and the whole domain will be discretised to an acceptable number of points as detailed in the following section.

7.1.2 Methodology

For this analysis, the 4-dimension space was discretised with a 250x250 resolution in the image plane and a 16x16 resolution on the lens. This results in a signal with 16000000 points in total. The same sparsity metrics from the previous analysis are used here as well as the hard-threshold image quality comparison were executed for varying γ used for tone-mapping. For the image quality analysis, the hard-threshold reconstructions were compared to the reference after reverting the gamma mapping and then integrating the signal. This analysis was performed for the Dragons and San Miguel scenes.

7.1.3 Experimental Results

γ	Time	Time Energy	Frequency	Frequency Energy
1,00	0,6042	0,6574	0,4089	0,9920
0,50	0,5252	0,6042	0,4179	0,9954
0,25	0,4384	0,5252	0,4103	0,9971
0,10	0,3570	0,4141	0,4314	0,9987
0,05	0,3237	0,3570	0,4526	0,9990

Table 7.1: Gini Index for the Dragons scene with different gamma mapping

γ	Time	Time Energy	Frequency	Frequency Energy
1,00	0,7537	0,9493	0,3441	0,9537
0,50	0,5194	0,7537	0,3188	0,9367
0,25	0,3478	0,5194	0,3074	0,9442
0,10	0,2113	0,3064	0,3014	0,9584
0,05	0,1549	0,2113	0,2996	0,9632

Table 7.2: Gini Index for the San Miguel scene with different gamma mapping

The Gini Index applied to the Fourier coefficients magnitude ("Frequency" column) is relatively low for both scenes, even with tone-mapping. Coefficient energy on the other hand is consistently sparse in the Fourier domain, while denser with increasing γ in the original domain. These results shown an increased density in the original domain with increasing γ, as well as increased sparsity in the Fourier space, which may positively affect sparse reconstruction algorithms as suggested by the results presented in figures 7.1 and 7.2.

For each scene the optimal γ values are different due to their initial contrast, and setting a too low value for the parameter may reduce contrast rendering the signal values harder to distinguish, which invalidates a sparse reconstruction despite the increased sparsity achieved.

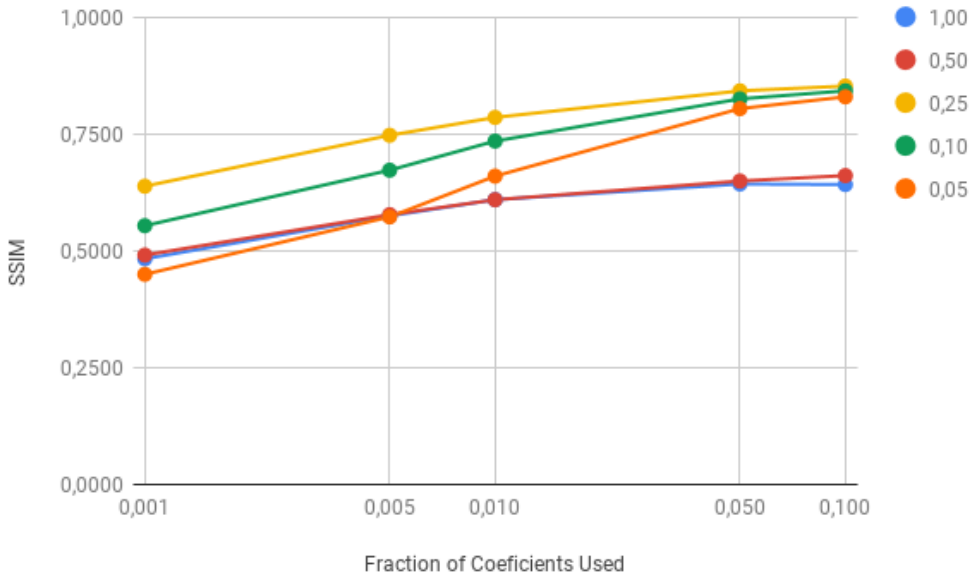


Figure 7.1: SSIM for different Gamma for the Dragons scene

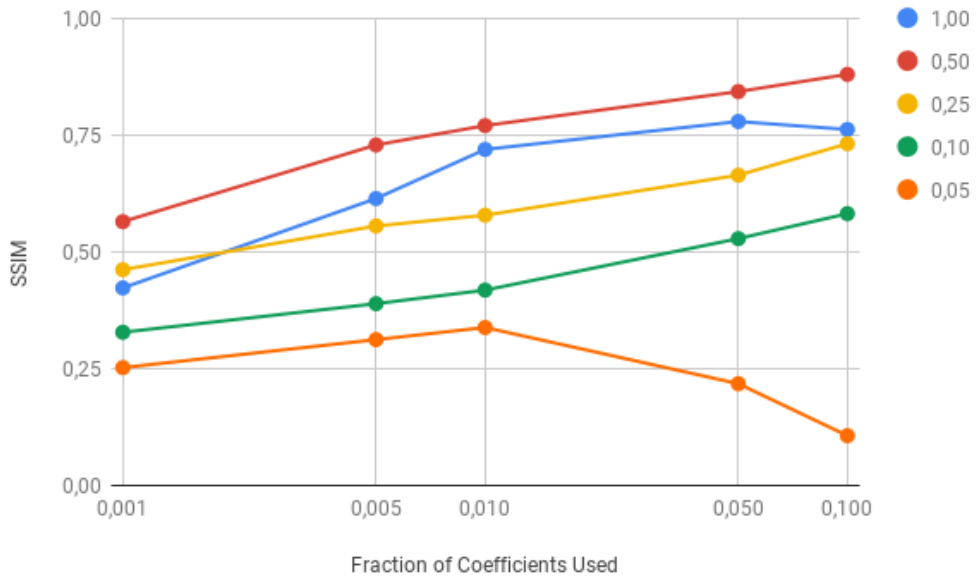
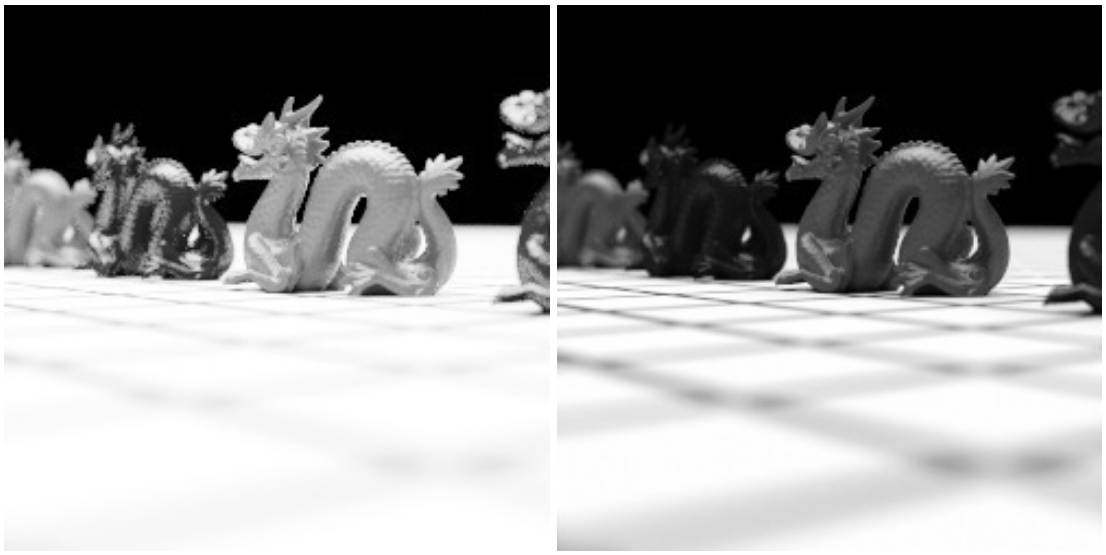


Figure 7.2: SSIM for different Gamma for the San Miguel scene

Although using gamma tone-mapping might increase sparsity in the Fourier space, it reduces the final image contrast of the hard-thresholded signals, not fully capturing the original signal, as seen in figures 7.3 and 7.4.



(a) Reference

(b) Hard-thresholded signal using gamma of 0.25 at 0.05 sparsity rate

Figure 7.3: Comparison of hard-thresholded signal using gamma mapping for the Dragons scene

(a) Reference

(b) Hard-thresholded signal using gamma of 0.25 at 0.05 sparsity rate

Figure 7.4: Comparison of hard-thresholded signal using gamma mapping for the San Miguel Scene

Given these results, it is possible to observe that the image quality of the hard-thresholded signals is heavily impacted by the use of gamma tone-mapping, which significantly improves the SSIM of the final images when at optimal values. This improvement is due to the increased density in the canonical domain and increased sparsity in the Fourier domain. Excessively low values of γ however cause severe image quality loss due to the signal getting closer to a constant as γ tends to zero. There is however a slight shift in contrast with the use of tone-mapping. It is hypothesized that this is due to the Fourier hard-thresholding being applied to a transformed signal instead of the original, which may hinder the reconstruction algorithm ability to fully recover a viable image result.

7.2 Indirect Lighting Sparsity

Another case study selected was the indirect lighting space, with the signal being parametrized by its image plane coordinates and the two random numbers used to generate the direction of the secondary ray. This case study was chosen due to its relevance: 1 bounce indirect lighting is computed in almost every rendering scenario.

7.2.1 Methodology

For the indirect lighting sparsity analysis, a similar process to the depth of field analysis was performed, using a signal with a 250x250 image plane resolution and a 16x16 resolution for the secondary ray direction, as well as an image quality assessment with varying γ. The sparsity results for the San Miguel scene are presented in table 7.3 and the image quality analysis of the hard-thresholded signals in figure 7.5. Given the small indirect lighting component of the Dragons scene, it was not included in this test.

7.2.2 Experimental Results

Gamma	Time	Time Energy	Frequency	Frequency Energy
1,00	0,689	0,9487	0,2435	0,9656
0,50	0,3889	0,689	0,4049	0,9946
0,25	0,2018	0,3889	0,3809	0,9971
0,10	0,0821	0,1624	0,3682	0,9992
0,05	0,0413	0,0821	0,3643	0,9997

Table 7.3: Gini Index for the San Miguel scene with different gamma mapping

As in the depth of field case, the Gini Index applied to Fourier coefficient magnitude is low, while for energy is really high regardless of the γ. In the canonical domain, both coefficient magnitude and energy become denser with decreasing γ. As in the previous example, image quality of hard-thresholded signals significantly improves with the application of the tone-mapping operator, as presented in figure 7.5.

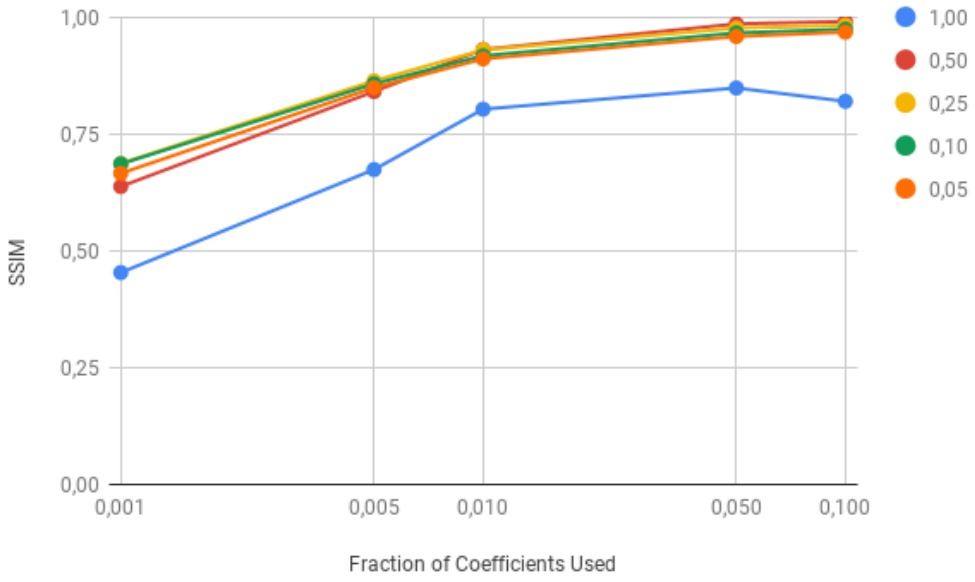


Figure 7.5: SSIM for different Gamma for the San Miguel scene

(a) Reference

(b) Hard-thresholded signal using gamma of 0.25 at 0.05 sparsity rate

Figure 7.6: Comparison of hard-thresholded signal using gamma mapping for the San Miguel Scene

In this test a similar behaviour as in the depth-of-field test was observed: hard-threshold quality improved with tone-mapping, higher density in the canonical domain and higher Fourier sparsity with decreasing γ and a slight contrast variation on the tone-mapped hard-threshold.

7.3 Summary

Through this chapter it was observed that a sparse reconstruction approach applied to a smaller 4-dimensional space may be feasible due to the high sparsity of the Fourier energy, especially with the application of a tone-mapping operator to the signal, which significantly improved the quality of the hard-thresholded signal w.r.t. SSIM.

However, the Fourier coefficients magnitudes are not so sparsely distributed, even with tone mapping. As described in section 5.4, it is expected that sparse reconstruction algorithms, like CoSaMP, may struggle to detect and estimate the optimal representation (i.e, the appropriate coefficients and the respective magnitudes).

Chapter 8

CoSaMP Reconstruction in 4D Space

8.1 Depth of Field Reconstruction

Given the results from the previous analysis, it was decided to test a sparse reconstruction approach using CoSaMP applied to a 4D space, using Gamma tone-mapping in order to increase the density of the original depth of field signals.

8.1.1 Methodology

For the following tests the image quality of the reconstructions using the CoSaMP algorithm applied to the depth of field problem were evaluated. For this effect, for both scenes tested, the execution times of the several parts of the algorithm were measured and the image quality was evaluated using SSIM w.r.t. to rendering time, comparing it to Path Tracing. The goal is to assert whether there is a gain in image quality (as measured by SSIM) for the same computation time. For these tests the gamma tone-mapping is applied using the previously observed optimal values for each scene: 0,25 for the Dragons and 0,5 for the San Miguel, and an expected sparsity rate of 1/200. The algorithm was applied to the luminance channel only, using the original samples provided to the algorithm to calculate the chroma for each pixel, and using the CoSaMP result only to estimate brightness. This was done in order to avoid calculating three CoSaMP instances, one for each channel, thus saving computational time.

8.1.2 Experimental Results

Tables 8.1 and 8.2 present sparse reconstruction times for the Dragons and San Miguel scenes, respectively. For different numbers of samples per pixel (spp) the total rendering time is decomposed into sampling, reconstruction (CoSaMP) and integration times.

SPP	Total	Sampling	CoSaMP	Integration
40	16,7	1,89	14,6	0,182
81	22,4	4,06	18,1	0,195
163	29,0	8,13	20,7	0,183
327	51,6	16,0	35,3	0,182

Table 8.1: Execution Times(s) for the Dragons scene

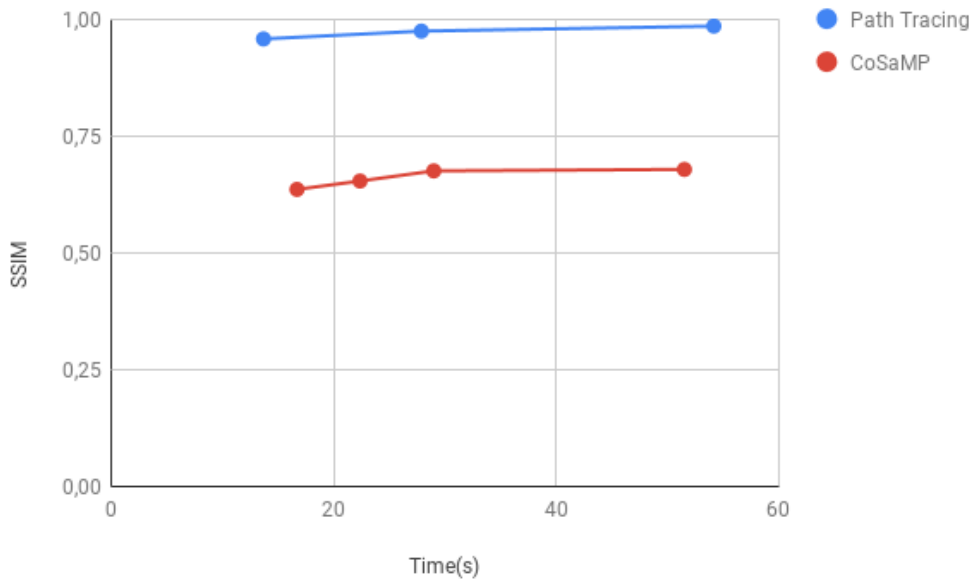
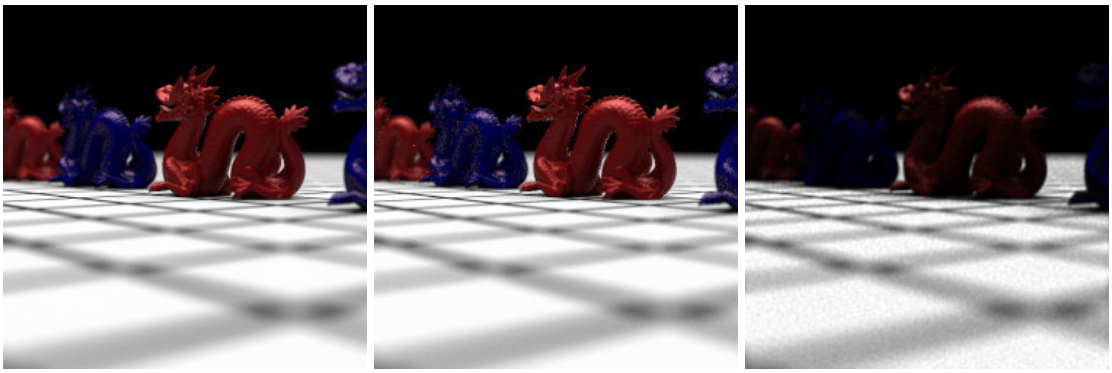


Figure 8.1: Reconstruction quality w.r.t. time for the Dragons scene



(a) Reference (b) Path Tracing 512 SPP (c) CoSaMP Reconstruction 160 SPP

Figure 8.2: Image Comparison of the CoSaMP approach to Path tracing for the Dragons scene for similar rendering times

SPP	Total	Sampling	CoSaMP	Integration
40	26,93	12,49	14,43	0,012
81	38,59	27,11	11,30	0,19
163	63,21	49,50	13,53	0,19

Table 8.2: Execution Times(s) for the San Miguel scene

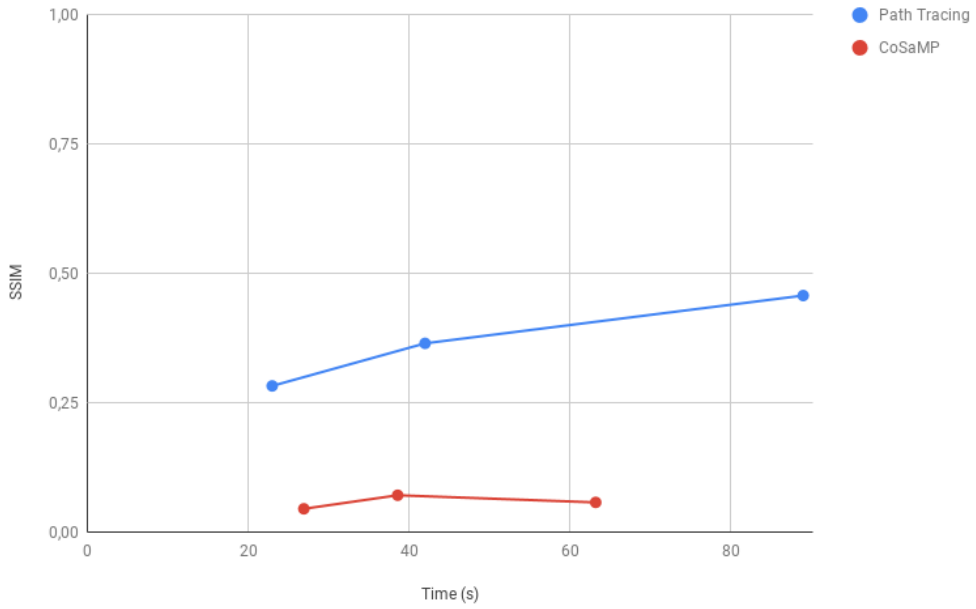


Figure 8.3: Reconstruction quality w.r.t. time for the San Miguel scene

(a) Reference (b) Path Tracing 256 SPP (c) CoSaMP Reconstruction 160 SPP

Figure 8.4: Image Comparison of the CoSaMP approach to Path tracing for the San Miguel scene

As shown in tables 8.1 and 8.2, adding the CoSaMP on top of the rendering pipeline increases the execution time by an approximately constant factor, which is specially significant in the Dragons scene which has low sampling costs due to its simplicity. For all the tests performed, the implemented approach could not outperform the Path Tracing regarding image quality for similar execution times as shown in

figures 8.1 and 8.3. The reconstructed images are considerably darker and have a lower contrast than the originals. This issue is already present in the hard-thresholded signals (figures 7.3 and 7.4), and the CoSaMP process amplified the issue even further. However, although the reconstructions present a much lower contrast and brightness, they are smoother and exhibit less grain than the path traced images.

8.2 Depth of Field Bilateral Filter Reconstruction

Although the results from the CoSaMP do not improve upon the classical approach, there is an efficient way of using them. By leveraging the reconstruction from CoSaMP it may be possible to improve images rendered through Path Tracing and thus generating higher quality images.

The images generated by CoSaMP are darker, lack contrast and are somewhat blurry when compared to the originals. However they are also smooth, not exhibiting the typical grain effect of low sample Path Traced images. Thus, it may be possible to use this generated information to filter a Path Traced generated image in order to obtain a higher quality image with fewer samples and in less time.

8.2.1 Bilateral Filtering

The bilateral filter is an edge-preserving filter that weights neighbouring pixels according to their distance in both image plane and magnitude.

$$I_f(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x) - I(x_i)\|) g_s(\|x - x_i\|) \tag{8.1}$$

being

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x) - I(x_i)\|) g_s(\|x - x_i\|) \tag{8.2}$$

where:

- I_f is the filtered image;
- I is the original input image;
- x are the coordinates of the pixel being filtered;
- Ω is the window centring pixel x;
- f_r is the range kernel for smoothing pixel intensity differences;
- g_s is the spatial kernel for smoothing differences in coordinates.

Pixels farther away and with very different magnitudes are given smaller weights while closer ones are given higher weights. These weights are typically a Gaussian function on the distance between the pixels and their magnitude difference [65].

By using the distance between pixels in the CoSaMP reconstruction as an additional distance metric for the Bilateral Filter, it is possible to smooth the Path Tracing image using a smooth reference generated through sparse reconstruction techniques.

$$I_f(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x) - I(x_i)\|) g_s(\|x - x_i\|) f_r(\|I_{cs}(x) - I_{cs}(x_i)\|) \tag{8.3}$$

where I_{cs} is the CoSaMP reconstructed image, with W_p adjusted accordingly.

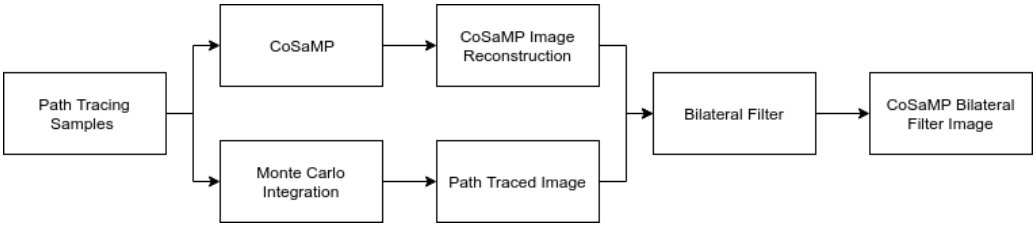


Figure 8.5: CoSaMP Bilateral Filter Flowchart

8.2.2 Methodology

The image quality of this approach is compared to Path Tracing as well as other filters such a Gaussian and Bilateral Filter itself (equation 8.1). For that the SSIM was measured for varying rendering times using the same parameters as in the previous experiments.

8.2.3 Experimental Results

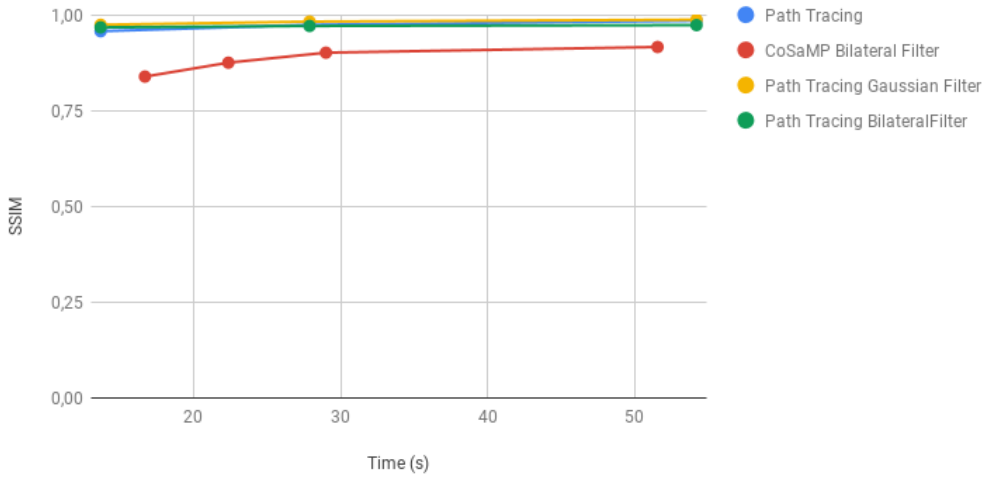
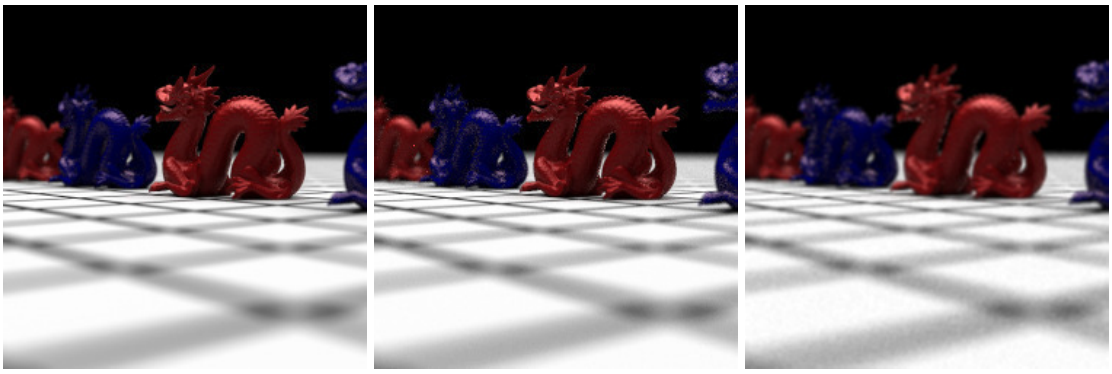


Figure 8.6: Reconstruction quality w.r.t. time for the Dragons scene



(a) Reference (b) Path Tracing 512 SPP (c) CoSaMP Bilateral Filter Reconstruction 160 SPP

Figure 8.7: Image Comparison of the CoSaMP approach to Path tracing for the Dragons scene

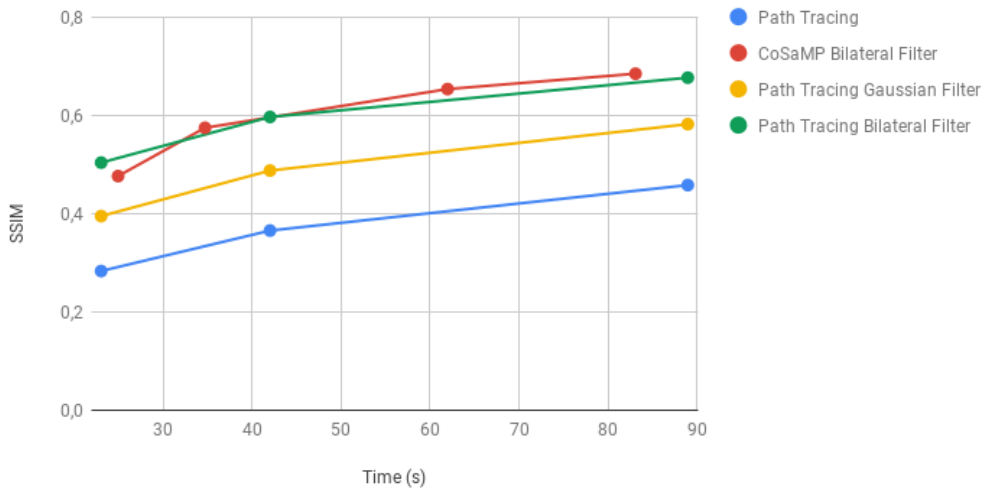


Figure 8.8: Reconstruction quality w.r.t. time for the San Miguel scene

(a) Reference (b) Path Tracing 256 SPP (c) CoSaMP Bilateral FilterReconstruction 160 SPP

Figure 8.9: Image Comparison of the CoSaMP approach to Path tracing for the San Miguel scene

For the San Miguel scene some slight improvements upon were observed for the same rendering times. For the dragons scene however, since the sampling times are so low, the cost of running CoSaMP is not

worth over simply taking more samples. For both scenes, the results are slightly blurred compared to the original, which is typical of a bilateral filter approach.

8.3 Depth of Field Importance Map

Although direct visualization of the CoSaMP result might not be feasible, due to the reduced contrast reconstructed images exhibit, that result may provide some benefit as an importance map to use for classical Monte Carlo integration. By using the result of of the CoSaMP as a probability distribution function, it might be possible to guide samples towards the points in the image plane and lens point with the higher contributions, thus reducing variance in the final image.

8.3.1 Methodology

For this approach, the result of the CoSaMP process is fully expanded into a 4D importance map of the same resolution as the signal, which is then used to importance sample the image plane and lens point using a similar procedure as presented in section 6.2.1.

With a resolution of 250x250 in the image plane and 16x16 on the lens, the importance maps were generated using CoSaMP with the optimal γ values for each scene, and compared with Path Tracing w.r.t. total rendering time. For the Dragons scene two importance maps were generated using CoSaMP 40 and 160 samples per pixel. For the San Miguel scene two importance maps were generated using CoSaMP 40 and 80 samples per pixel.

8.3.2 Experimental Results

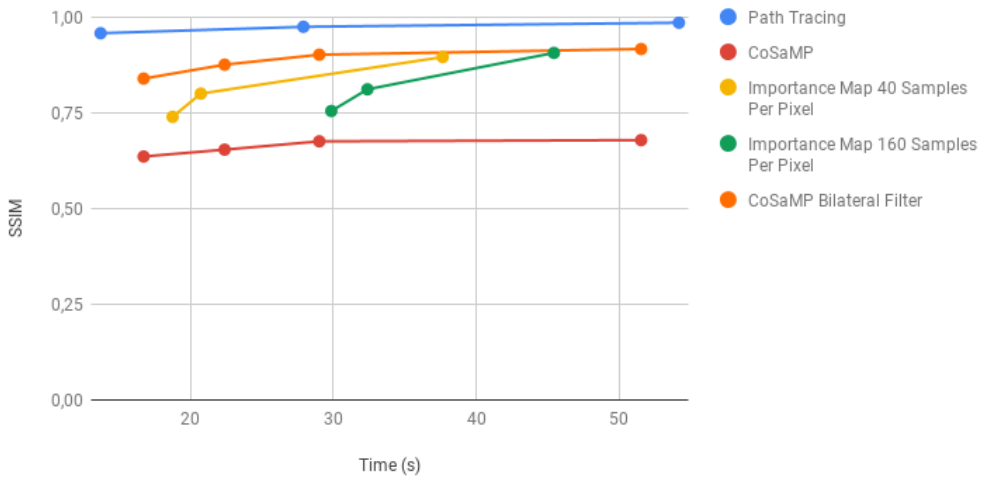
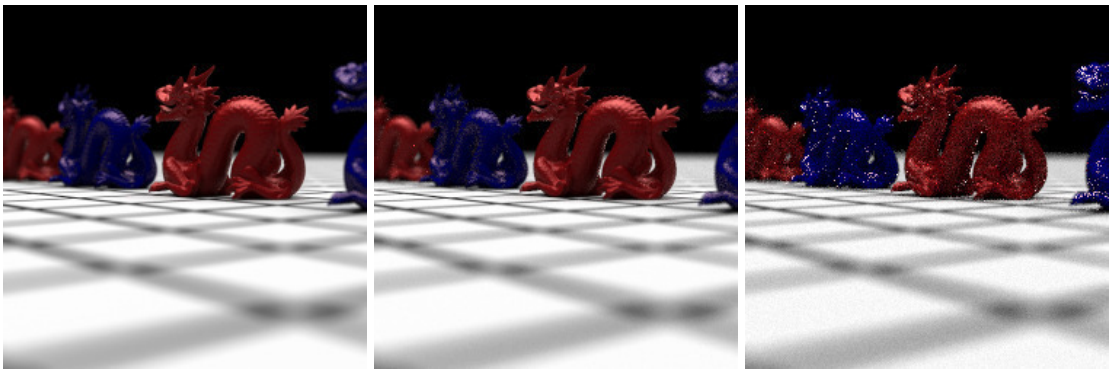


Figure 8.10: Image quality analysis of the importance sampling method for the Dragons scene



(a) Reference (b) Path Tracing 512 SPP (c) CoSaMP Importance Sampling: 160 CoSaMP SPP, 128 importance sampled SPP

Figure 8.11: Image Comparison of the Importance Sampling CoSaMP approach to Path tracing for the Dragons scene

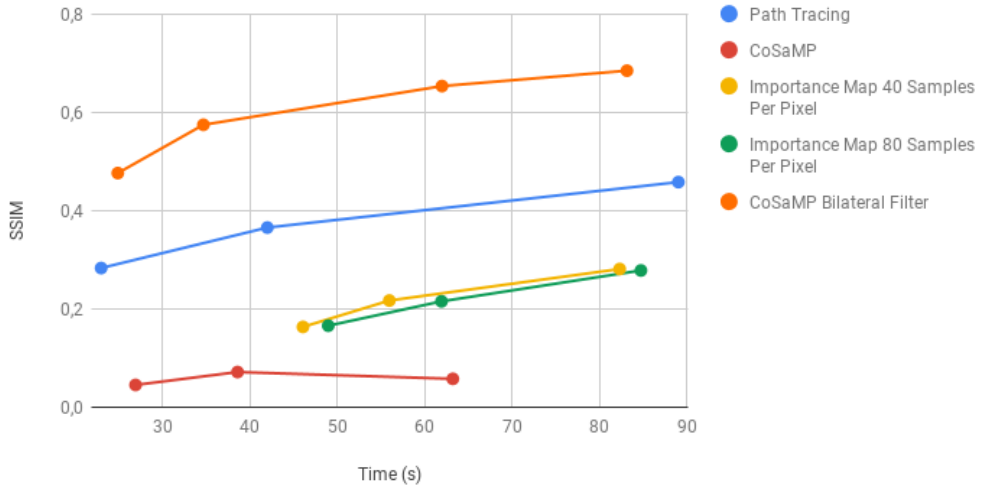


Figure 8.12: Image quality analysis of the importance sampling method for the San Miguel scene

(a) Reference (b) Path Tracing 256 SPP (c) CoSaMP Importance Sampling: 80 CoSaMP SPP, 128 importance sampled SPP

Figure 8.13: Image Comparison of the Importance Sampling CoSaMP approach to Path tracing for the San Miguel scene

As observed from the results in figures 8.10 and 8.12, this method provided no real benefit compared to Path tracing, and given the reconstruction quality of the images presented in 8.1.2 it is hypothesized that the importance maps generated do not match the signal closely enough to provide variance reduction. Also, the added overhead of additional samples and CoSaMP execution time detract the potential gains of this approach.

8.4 4-D Importance Map Bilateral Filtering

Due to the small improvement observed using the CoSaMP Bilateral Filtering approach for the San Miguel scene, it may be worth to apply a similar approach to the Importance Sampling problem. The main goal is to filter a set of samples in the 4-D space, in order to generate an importance map with higher fidelity, and thus reducing variance.

8.4.1 Methodology

By using the reconstruction result of the CoSaMP algorithm as an additional distance metric for the bilateral filter, the initial samples used in for CoSaMP are filtered in the 4-D space, and that filtered result is then used as an importance map used for Monte Carlo integration.

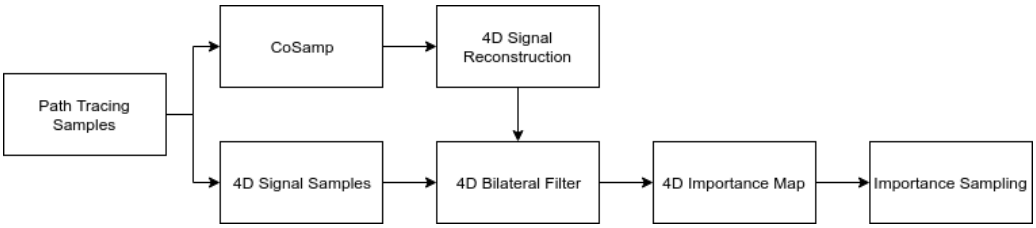


Figure 8.14: Bilateral Filter CoSaMP Importance Sampling Flowchart

8.4.2 Experimental Results

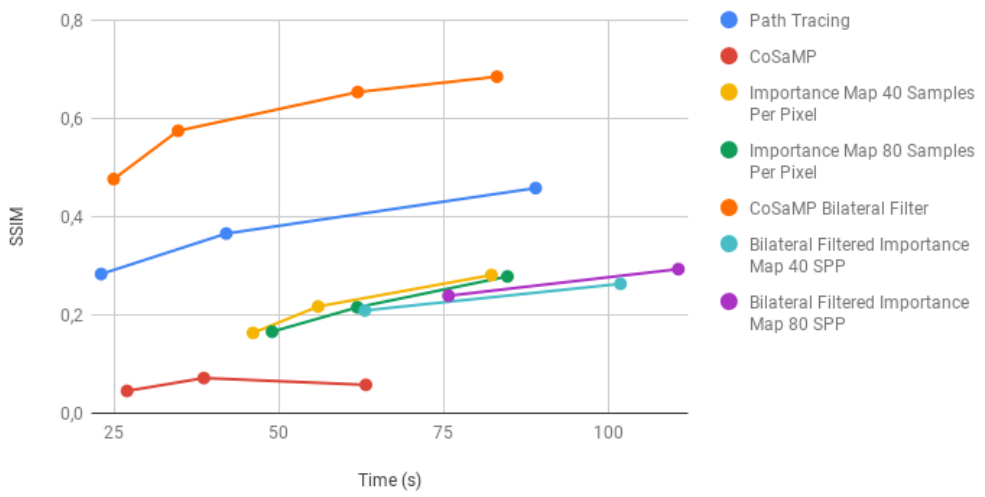


Figure 8.15: Image quality analysis of the importance sampling method for the San Miguel scene

(a) Reference

(b) Path Tracing 256 SPP

(c) Bilateral Filter CoSaMP Importance Sampling: 80 CoSaMP SPP, 128 importance sampled SPP

Figure 8.16: Image Comparison of the Bilateral Filtered Importance Map approach to Path tracing for the San Miguel scene

(a) Path Tracing 256 SPP

(b) Bilateral Filter CoSaMP Importance Sampling: 80 CoSaMP SPP, 128 importance sampled SPP

Figure 8.17: Direct Lighting detail

As observed in figure 8.15, this approach does not bring any improvement upon previous approaches regarding SSIM. However, as observed in the figure 8.17,some areas of the image, namely the brighter directly lit ones, do seem more converged, while the darker ones exhibit more noise than Path Tracing.

8.5 Summary

Although generally no major benefits were achieved through the proposed approaches, some improvement was achieved using the CoSaMP as a bilateral filter guide in a measurable image quality improvement for similar rendering time compared to Path Tracing. The bilateral filter of a 4D importance map, although did not provide measurable image quality improvements, appears to converge faster in brightly lit areas of the image, which may warrant further research.

Chapter 9

Conclusion and Future Work

This thesis' main goal is to contribute to reduce the large rendering times associated with physically based global illumination algorithms, in particular path tracing. The approach taken consists on using sparse reconstruction algorithms, therefore reducing the number of samples (i.e., paths) required to render high quality photo realistic images, compared to regular path tracing. Through this thesis, several methods were proposed for doing so, including sparse reconstruction followed by integration, importance sampling and filtering based approaches.

9.1 Results Discussion

Effective sparse signal reconstruction has two requirements regarding sparsity: the signal of interest must admit a dense representation in its canonical basis and a sparse representation in the selected reconstruction basis. In chapter 5 a sparsity analysis of Path Space signals was performed. A high level of sparsity in the Fourier basis was observed for these signals. High quality hard-thresholded images were obtained using only $1/10^4$ of the Fourier coefficients. However, these signals also exhibited a high sparsity value in the canonical basis. This severely hurts the effectiveness of sparse reconstruction algorithms. In fact, measurements are performed in the canonical basis; since the signal representation in this basis is not dense it will be difficult for the reduced number of measurements to capture enough information to allow for robust sparse reconstruction.

The SFT algorithm was used to perform sparse reconstruction of a 10-dimensional path space, followed by an efficient integration step on Fourier space which allowed for image rendering. However, even for a large number of samples (which defeats the initial goal) the reconstructed images are plagued with artifacts. Sparse reconstructing a lower resolution signal, which is then used as an importance map on an unbiased Monte Carlo path tracing stage, did not improve the results significantly. Promising improvements were observed only in and around caustics, while the remaining image regions still exhibit artifacts. These artifacts are similar to those observed with direct SFT reconstructions. It is, therefore, hypothesized that the artifacts are present in the importance maps as well, thus preventing the importance sampling approach to be effective. All these limitations arise from the huge number of samples required by the

SFT algorithm. The number of samples required exceeds the signal size itself, rendering the approach not competitive, despite the logarithmic complexity bound. The improvements observed in and around caustics might still deserve further investigation.

In order to leverage potential advantages brought about by sparse reconstruction, a different algorithm was investigates. This algorithm, based on the general idea of Matching Pursuit, is CoSaMP. Its complexity is, however, linear with the signal size and exponential on the number of signal dimensions. Very large signals, such as 10-dimensional path spaces, can not be treated due to time and space constraints. CoSaMP was therefore investigated with smaller, 4-dimensional signals, in particular depth of field and one-bounce indirect lighting.

The sparsity analysis of these 4-dimensional path spaces provided mixed results. While the Fourier energy was sparsely distributed for both scenes, the Fourier magnitude was shown to be dense. This is even denser than the canonical space magnitude distribution, which heavily impacts sparse reconstruction.

It was hypothesized that the sparseness in the canonical space (and consequently the density in Fourier space) was caused by the high dynamic range of radiance values carried by the rays. In fact, paths originating at the light sources, for instance, can carry radiance values orders of magnitude larger than paths with smaller contributions. To handle this phenomenon the signal rage was compressed using gamma (γ) correction, in a manner inspired in tone mapping operators. For both scenes tested, the canonical domain density increased with lower γ.

The optimal γ parameter for the hard-threshold reconstructions is however a compromise, since setting it too low may destroy the signal being reconstructed, while setting it too high does not improve sparsity enough for a quality recovery. Also, with range compressed signals, hard-thresholded reconstructions exhibit different contrast than the reference.This is hypothesized to be due to the signal upon which the hard-threshold operation is applied being different than the desired target.

The CoSaMP reconstruction of the 4-dimensional spaces did not achieve a measurable image quality improvement: the contrast on the resulting images is heavily altered. These results were however much smoother than the Path Tracing ones, fact that was leveraged by proposing a bilateral filter approach. The result of the CoSaMP algorithm is used as an additional distance metric for the bilateral filtering of previously taken samples. This approach yielded measurable improvements in image quality for the San Miguel scene. However, given the time overhead this approach implies, such improvement was not observed for the other test scene, the Dragons. Sampling times are much lower, which makes the fixed overhead more costly, i.e., impossible to amortize. The importance sampling approaches applied in 4D yielded similar results to the Path Space importance sampling, with local improvements but overall lower image quality.

9.2 Contributions

The first main contribution of this thesis is a thorough sparsity analysis of the Path Space in the Fourier and canonical basis. By understanding the behaviour of Path Space in this regard, it should be possible to better leverage sparse reconstruction techniques, as well understand their limitation and potential in

this domain. According to the presented analysis, Path Space has a sparse energy distribution in the Fourier space, which implies the possibility of representing the signal in the Fourier space with a small number of coefficients, one of the requirements for sparse reconstruction algorithms to be applicable. Note that this is also the main requirement for sparse signal representation to be possible. However, Fourier space's coefficients magnitudes are not so sparse. This makes it very hard for sparse reconstruction algorithms, such as those inspired in Matching Pursuit, to correctly identify the meaningful coefficients (i.e., the signal support). Also, Path Space presents sparse distributions in its canonical domain, which makes the sampling process less likely to capture high contribution samples.

The second contribution is a thorough understanding of the advantages and drawbacks of applying sparse reconstruction techniques to multidimensional light transport signals. The high sampling requirements of SFT and the linear complexity of CS approaches makes their application to high cardinality domains impractical given the current conditions. Also, given the sparsity profile of lower dimensionality signals, CS approaches require additional work in order to leverage their potential in this domain.

The next main contribution of this thesis is the proposal of a method to efficiently integrate the results of sparse reconstruction algorithms in the Fourier basis. By leveraging the convolution theorem, it is possible to efficiently calculate the integration result for all the pixels of an image using a 2D FFT, which results in an improvement of time and memory complexity to $O(P\log(P))$, where P is the total number of pixels. This result is very important, as it heavily reduces the cost of the integration phase, and decouples its complexity from the result itself, depending only on the number of pixels to calculate.

The fourth main contribution is the CoSaMP bilateral filtering approach, which uses the CoSaMP reconstruction result as an additional distance metric used for bilateral filtering. This approach takes advantage of the smoothness of the reconstructions resulting from CoSaMP in order to guide the filtering process to obtain smooth images with a small number of samples. This proposed approach however adds a constant time overhead that is not justifiable in low complexity scenes. This opens up a possibility of taking advantage of sparse reconstruction results should the possibility of direct visualization of those results prove to be completely infeasible.

This thesis' final main contribution is the sparse reconstruction based importance sampling for path guiding, that uses sparse reconstruction results as an importance map to be used for unbiased Monte Carlo process. Such process provides an unbiased rendering method which takes advantage of the Fourier sparsity of Path Space. Although only local improvements were achieved, with further research these improvements can be leveraged while tackling its shortcomings.

9.3 Future Work

Given the results achieved with the importance sampling approaches, which managed to achieve improvements in some areas of the image, it may be worth to develop some further work into importance sampling mechanisms that take advantage of the areas improved, while detecting areas where other methods provide better results. In particular, techniques for combining sampling strategies such as Multiple

Importance Sampling or Defensive Sampling could provide good results in avoiding the noise present while preserving the improvements achieved with the proposed approaches [56] [29] [44] [42].

Another possible challenge would be to expand the proposed approaches to bidirectional algorithms such as Bidirectional Path Tracing [70]. Mapping these algorithms into a discrete representation could be challenging since the higher degree of freedom in sample acquisition could increase the dimensionality of the signal beyond what might be manageable. This could even be applied to photon mapping based approaches [36] [25] [73] [20] [19] since they can also be represented under the path space integration framework.

Sparsity was used as a parameter for reconstruction through all the implemented solutions. While the optimal values were measured beforehand, this is not practical for real applications. This forces the use of conservative estimates for sparsity and thus needless computational effort. Developing a way of efficient estimation of the Fourier sparsity would allow a fine tuning of this parameter for each scene and thus allow for sample and computational savings [76] [50] [75] [79].

Finally, it would be desirable to allow for a progressive approach, that could produce intermediate results viewable in an interactive environment. Such approach would detect whether or not the current result needs further improvement and refine it by acquiring more samples and adjusting accordingly [51] [14].

9.4 Final Remarks

Through this thesis, the applications of sparse reconstructions to light transport signal were explored, gaining a thorough understanding of the benefits and limitations of such approaches. Two reconstruction pipelines were developed as well as path guiding mechanisms using sparse reconstruction as a base. Also, an efficient method for the integration of multidimensional signals was proposed. Thus this thesis provides a solid foundation for future work developed in this area.

Bibliography

[1] Intel math kernel library. URL https://software.intel.com/en-us/mkl.

[2] Richard Baraniuk, Mark A Davenport, Marco F Duarte, Chinmay Hegde, et al. An introduction to compressive sensing. Connexions e-textbook, 2011.

[3] Thomas Bashford-Rogers, Kurt Debattista, and Alan Chalmers. A significance cache for accelerating global illumination. In Computer Graphics Forum, volume 31, pages 1837–1851. Wiley Online Library, 2012.

[4] A. Bjorck. Numerical Methods for Least Squares Problems. Handbook of Numerical Analysis. Society for Industrial and Applied Mathematics, 1996. ISBN 9780898713602. URL https://books.google.pt/books?id=ZecsDBMz5-IC.

[5] T Tony Cai, Guangwu Xu, and Jun Zhang. On recovery of sparse signals via ℓ_1 minimization. IEEE Transactions on Information Theory, 55(7):3388–3397, 2009.

[6] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory, 52(2): 489–509, Feb 2006. ISSN 1557-9654. doi: 10.1109/TIT.2005.862083.

[7] Emmanuel Candes and Terence Tao. Decoding by linear programming. arXiv preprint math/0502327, 2005.

[8] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE transactions on information theory, 52(12):5406–5425, 2006.

[9] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. IEEE signal processing magazine, 25(2):21–30, 2008.

[10] Emmanuel Candès. The restricted isometry property and its implications for compressed sensing. Compte Rendus de l'Academie des Sciences, 346:589–592, 05 2008. doi: 10.1016/j.crma.2008.03.014.

[11] Emmanuel Candès and Justin Romberg. Sparsity and incoherence in compressive sampling. Inverse Problems, 23(3):969–985, Apr 2007. ISSN 1361-6420. doi: 10.1088/0266-5611/23/3/008. URL http://dx.doi.org/10.1088/0266-5611/23/3/008.

[12] Emmanuel Candès, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. Communications on Pure and Applied Mathematics, 59, 08 2006. doi: 10.1002/cpa.20124.

[13] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. American Mathematical Society, 22:211–231, 01 2009. doi: 10.1090/S0894-0347-08-00610-3.

[14] G. Coluccia, S. K. Kuiteing, A. Abrardo, M. Barni, and E. Magli. Progressive compressed sensing and reconstruction of multidimensional signals using hybrid transform/prediction sparsity model. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2(3):340–352, Sep 2012. ISSN 2156-3365. doi: 10.1109/jetcas.2012.2214891. URL http://dx.doi.org/10.1109/JETCAS.2012.2214891.

[15] H. Dalton. The measurement of the inequity of incomes. Economic Journal, 30:348–361, 1920.

[16] D. L. Donoho. Compressed sensing. IEEE Transactions on Information Theory, 52(4):1289–1306, April 2006. ISSN 1557-9654. doi: 10.1109/TIT.2006.871582.

[17] Philip Dutré, Eric P. Lafortune, and Yves D. Willems. Monte carlo light tracing with direct computation of pixel intensities. In 3rd International Conference on Computational Graphics and Visualisation Techniques, pages 128–137, Alvor, Portugal, December 1993.

[18] Simon Foucart and Holger Rauhut. A mathematical introduction to compressive sensing, volume 1. Springer, 2013.

[19] I. Georgiev, J. Kr□ivánek, T. Davidovič, and P. Slusallek. Light transport simulation with vertex connection and merging. ACM Trans. Graph., 31(6):192:1–192:10, November 2012. ISSN 0730-0301.

[20] Iliyan Georgiev, Jaroslav Krivanek, and Philipp Slusallek. Bidirectional light transport with vertex merging. 12 2011. doi: 10.1145/2077378.2077412.

[21] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, S. Muthukrishnan, and M. Strauss. Near-optimal sparse fourier representations via sampling. In Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing, STOC '02, pages 152–161, New York, NY, USA, 2002. ACM. ISBN 1-58113-495-9. doi: 10.1145/509907.509933. URL http://doi.acm.org/10.1145/509907.509933.

[22] Anna Gilbert, Martin Strauss, and Joel Tropp. A tutorial on fast fourier sampling. Signal Processing Magazine, IEEE, 25:57 – 66, 04 2008. doi: 10.1109/MSP.2007.915000.

[23] Corrado Gini. Measurement of inequality of incomes. The Economic Journal, 31(121):124–126, 1921. ISSN 00130133, 14680297. URL http://www.jstor.org/stable/2223319.

[24] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. SIGGRAPH Comput. Graph., 18(3):213–222, January 1984. ISSN 0097-8930. doi: 10.1145/964965.808601. URL http://doi.acm.org/10.1145/964965.808601.

[25] T. Hachisuka, S. Ogaki, and H. W. Jensen. Progressive photon mapping. ACM Trans. Graph., 27 (5):130:1–130:8, December 2008. ISSN 0730-0301.

[26] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Nearly optimal sparse fourier transform. arXiv preprint arXiv:1201.2501, 2012.

[27] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Simple and practical algorithm for sparse fourier transform. In Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms, pages 1183–1194. Society for Industrial and Applied Mathematics, 2012.

[28] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[29] Hera Y He and Art B Owen. Optimal mixture weights in multiple importance sampling. arXiv preprint arXiv:1411.3954, 2014.

[30] Sebastian Herholz, Oskar Elek, Jir□í Vorba, Hendrik Lensch, and Jaroslav Kr□ivánek. Product importance sampling for light transport path guiding. In Computer Graphics Forum, volume 35, pages 67–77. Wiley Online Library, 2016.

[31] Heinrich Hey and Werner Purgathofer. Importance sampling with hemispherical particle footprints. In Proceedings of the 18th spring conference on Computer graphics, pages 107–114. ACM, 2002.

[32] Niall P. Hurley and Scott T. Rickard. Comparing measures of sparsity. CoRR, abs/0811.4706, 2008. URL http://arxiv.org/abs/0811.4706.

[33] Piotr Indyk and Michael Kapralov. Sample-optimal fourier sampling in any constant dimension. In 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, pages 514–523. IEEE, 2014.

[34] Piotr Indyk, Michael Kapralov, and Eric Price. (nearly) sample-optimal sparse fourier transform. In Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms, pages 480–499. Society for Industrial and Applied Mathematics, 2014.

[35] Mark Iwen. Combinatorial sublinear-time fourier algorithms. Foundations of Computational Mathematics, 10:303–338, 06 2010. doi: 10.1007/s10208-009-9057-1.

[36] H. W. Jensen. Global illumination using photon maps. In Rendering Techniques' 96, pages 21–30. Springer, 1996.

[37] Henrik Wann Jensen. Importance driven path tracing using the photon map. In Rendering Techniques' 95, pages 326–335. Springer, 1995.

[38] J. T. Kajiya. The rendering equation. SIGGRAPH Comput. Graph., 20(4):143–150, August 1986. ISSN 0097-8930.

[39] Michael Kapralov. Sparse fourier transform in any constant dimension with nearly-optimal sample complexity in sublinear time. CoRR, abs/1604.00845, 2016. URL http://arxiv.org/abs/1604.00845.

[40] Michael Kapralov. Sparse fourier transform in any constant dimension with nearly-optimal sample complexity in sublinear time. In Proceedings of the forty-eighth annual ACM symposium on Theory of Computing, pages 264–277. ACM, 2016.

[41] Michael Kapralov, Ameya Velingker, and Amir Zandieh. Dimension-independent sparse fourier transform. In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '19, pages 2709–2728, Philadelphia, PA, USA, 2019. Society for Industrial and Applied Mathematics. URL http://dl.acm.org/citation.cfm?id=3310435.3310603.

[42] Ondřej Karlík, Martin Šik, Petr Vévoda, Tomáš Skřivan, and Jaroslav Křivánek. Mis compensation: Optimizing sampling techniques in multiple importance sampling. ACM Trans. Graph. (SIGGRAPH Asia 2019), 38(6), 2019. doi: 10.1145/3355089.3356565.

[43] Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. A simple and robust mutation strategy for the metropolis light transport algorithm. In Computer Graphics Forum, volume 21, pages 531–540. Wiley Online Library, 2002.

[44] Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. Optimal multiple importance sampling. ACM Trans. Graph., 38(4), 2019. doi: 10.1145/3306346.3323009. URL http://doi.acm.org/10.1145/3306346.3323009.

[45] E. P. Lafortune and Y. D. Willems. Bi-directional path tracing. In Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93), pages 145–153, 1993.

[46] Eric Lafortune. Mathematical models and monte carlo algorithms for physically based rendering. Department of Computer Science, Faculty of Engineering, Katholieke Universiteit Leuven, pages 20–23, 1996.

[47] Eric P Lafortune and Yves D Willems. A 5d tree to reduce the variance of monte carlo ray tracing. In Rendering Techniques' 95, pages 11–20. Springer, 1995.

[48] G. Peter Lepage. Vegas - an adaptive multi-dimensional integration program. CLNS-80/447, 1980.

[49] Pierre-David Letourneau, Harper Langston, Benoit Meister, and Richard Lethin. A sparse multidimensional fft for real positive vectors. arXiv preprint arXiv:1604.06682, 2016.

[50] Miles Lopes. Estimating unknown sparsity in compressed sensing. In ICML (3), pages 217–225, 2013.

[51] Dmitry M Malioutov, Sujay R Sanghavi, and Alan S Willsky. Sequential compressed sensing. IEEE Journal of Selected Topics in Signal Processing, 4(2):435–444, 2010.

[52] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6):1087–1092, 1953.

[53] Thomas Müller, Markus Gross, and Jan Novák. Practical path guiding for efficient light-transport simulation. In Computer Graphics Forum, volume 36, pages 91–100. Wiley Online Library, 2017.

[54] Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. Applied and computational harmonic analysis, 26(3):301–321, 2009.

[55] Michael Oren and Shree K. Nayar. Generalization of lambert's reflectance model. In Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '94, pages 239–246, New York, NY, USA, 1994. ACM. ISBN 0-89791-667-0. doi: 10.1145/192161.192213. URL http://doi.acm.org/10.1145/192161.192213.

[56] Art Owen and Yi Zhou. Safe and effective importance sampling. Journal of the American Statistical Association, 95, 12 1998. doi: 10.2307/2669533.

[57] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In in Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, pages 1–3, 1993.

[58] Matt Pharr and Greg Humphreys. Physically Based Rendering, Second Edition: From Theory To Implementation. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010. ISBN 0123750792, 9780123750792.

[59] Bui Tuong Phong. Illumination for computer generated pictures. Commun. ACM, 18(6):311–317, June 1975. ISSN 0001-0782. doi: 10.1145/360825.360839. URL http://doi.acm.org/10.1145/360825.360839.

[60] M.A. Pinsky. Introduction to Fourier Analysis and Wavelets. Graduate studies in mathematics. American Mathematical Society, 2008. ISBN 9780821847978. URL https://books.google.pt/books?id=PyISCgAAQBAJ.

[61] C. Poynton, Inc Books24x7, and Engineering Information Inc. Digital Video and HD: Algorithms and Interfaces. Electronics & Electrical. Elsevier Science, 2003. ISBN 9781558607927. URL https://books.google.pt/books?id=ra1lcAwgvq4C.

[62] Pradeep Sen and Soheil Darabi. Compressive estimation for signal integration in rendering. In Proceedings of the 21st Eurographics Conference on Rendering, EGSR'10, pages 1355–1363, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association. doi: 10.1111/j.1467-8659.2010.01731.x. URL http://dx.doi.org/10.1111/j.1467-8659.2010.01731.x.

[63] Pradeep Sen and Soheil Darabi. Compressive rendering: A rendering application of compressed sensing. IEEE Transactions on Visualization and Computer Graphics, 17(4):487–499, 2011.

[64] Pradeep Sen, Soheil Darabi, and Lei Xiao. Compressive rendering of multidimensional scenes. In Video Processing and Computational Video, pages 152–183. Springer, 2011.

[65] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In Iccv, volume 98, page 2, 1998.

[66] K. E. Torrance and E. M. Sparrow. Theory for Off-Specular Reflection from Roughened Surfaces, page 32–41. Jones and Bartlett Publishers, Inc., USA, 1992. ISBN 0867202947.

[67] Joel Tropp. Greed is good: Algorithmic results for sparse approximation. Information Theory, IEEE Transactions on, 50:2231 – 2242, 11 2004. doi: 10.1109/TIT.2004.834793.

[68] Joel A. Tropp and Anna C. Gilbert. Signal recovery from partial information via orthogonal matching pursuit. IEEE TRANS. INFORM. THEORY, 2005.

[69] E. Veach. Robust Monte Carlo Methods for Light Transport Simulation. PhD thesis, Stanford, CA, USA, 1998. AAI9837162.

[70] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. Proceedings of Eurographics Workshop on Rendering, 01 1995. doi: 10.1007/978-3-642-87825-1_11.

[71] Eric Veach and Leonidas J Guibas. Optimally combining sampling techniques for monte carlo rendering. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pages 419–428. ACM, 1995.

[72] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '97, pages 65–76, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0-89791-896-7. doi: 10.1145/258734.258775. URL https://doi.org/10.1145/258734.258775.

[73] J. Vorba. Bidirectional photon mapping. In Proc. of the Central European Seminar on Computer Graphics (CESCG'11), 2011.

[74] Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. On-line learning of parametric mixture models for light transport simulation. ACM Transactions on Graphics (TOG), 33 (4):101, 2014.

[75] Jianfeng Wang, Zhiyong Zhou, Anders Garpebring, and Jun Yu. Sparsity estimation in compressive sensing with application to mr images. arXiv preprint arXiv:1710.04030, 2017.

[76] Yue Wang, Zhi Tian, and Chunyan Feng. Sparsity order estimation and its application in compressive spectrum sensing for cognitive radios. IEEE transactions on wireless communications, 11(6): 2116–2125, 2012.

[77] Z. Wang, Eero Simoncelli, and Alan Bovik. Multiscale structural similarity for image quality assessment. volume 2, pages 1398 – 1402 Vol.2, 12 2003. ISBN 0-7803-8104-1. doi: 10.1109/ACSSC.2003.1292216.

[78] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. 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.

[79] Shihong Yao, Arun Kumar Sangaiah, Zhigao Zheng, and Tao Wang. Sparsity estimation matching pursuit algorithm based on restricted isometry property for signal reconstruction. Future generation computer systems, 88:747–754, 2018.

[80] Quan Zheng and Matthias Zwicker. Learning to importance sample in primary sample space. Computer Graphics Forum, 38(2):169–179, 2019. doi: 10.1111/cgf.13628. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13628.

This work was supported by the Fundação para a Ciência e Tecnologia under the PhD grant PD/BD/113966/2015.