SRA: Fast Removal of General Multipath for ToF Sensors

Daniel Freedman[1], Yoni Smolin[1], Eyal Krupka[1],
Ido Leichter[1], and Mirko Schmidt[2]

[1] Microsoft Research, Haifa, Israel
[2] Microsoft Corporation, Mountain View, CA, USA
{danifree,t-yonis,eyalk,idol,mirko.schmidt}@microsoft.com

Abstract. A major issue with Time of Flight sensors is the presence of multipath interference. We present Sparse Reflections Analysis (SRA), an algorithm for removing this interference which has two main advantages. First, it allows for very general forms of multipath, including interference with three or more paths, diffuse multipath resulting from Lambertian surfaces, and combinations thereof. SRA removes this general multipath with robust techniques based on L_1 optimization. Second, due to a novel dimension reduction, we are able to produce a very fast version of SRA, which is able to run at frame rate. Experimental results on both synthetic data with ground truth, as well as real images of challenging scenes, validate the approach.

1 Introduction

The field of depth sensing has attracted much attention over the last few years. By providing direct access to three-dimensional information, depth sensors make many computer vision tasks considerably easier. Examples include object tracking and recognition, human activity analysis, hand gesture analysis, and indoor 3D mapping; see the comprehensive review in [11].

Amongst depth sensing technologies, Time of Flight (ToF) imaging has recently shown a lot of promise. A phase modulated ToF sensor works by computing the time – measured as a phase-shift – it takes a ray of light to bounce off a surface and return to the sensor. ToF sensors are generally able to achieve very high accuracy, and – since they use light in the infrared spectrum – to operate in low illumination settings.

The main issue with ToF sensors is that they suffer from *multipath interference* (henceforth simply "multipath"). Since rays of light are being sent out for each pixel, and since light can reflect off surfaces in myriad ways, a particular pixel may receive photons originally sent out for other pixels as well. An illustration is given in Figure 2. Significant multipath is observed, for example, in scenes with shiny or specular-like floors.

The key problem is that multipath results in corrupted sensor measurements. These corruptions do not look like ordinary noise, and can be quite large, resulting in highly inaccurate depth estimates; see Figure 1. Removing the effect of multipath is therefore a crucial component for ToF systems.

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 234–249, 2014.

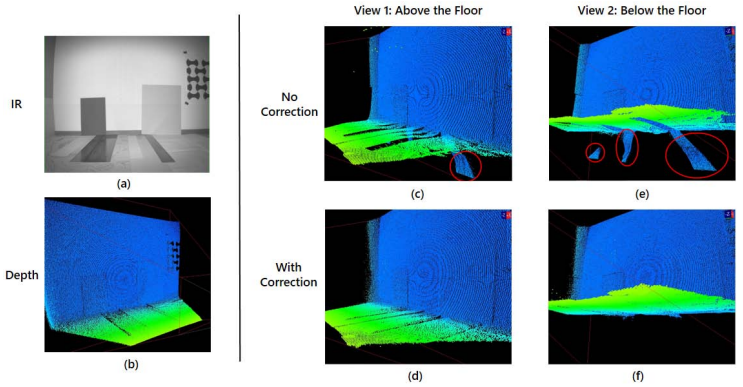


Fig. 1. Effect of Multipath and its Removal. Multipath is caused by specular floor materials. (a) IR image. (b) Depth reconstruction using our proposed SRA algorithm, rendered from a side-on viewpoint; the floor is shown in green, the wall in blue. (c, e) Results of not correcting for multipath, shown from two viewpoints – above and below the floor; gross errors are circled in red. (d, f) Output of our proposed SRA algorithm.

1.1 Contributions

Our work addresses two areas in which we improve on the state-of-the-art:

1. More General Multipath. As we will discuss more explicitly in Section 2, prior work mostly falls into two categories. The first class of algorithms focuses on the case with diffuse multipath, arising from Lambertian surfaces. The second class of algorithms focuses on the case of "two-path" multipath, which arises from specular surfaces.[1] But multipath can often be more general than this: specular multipath with more than two paths is possible, as are combinations of diffuse and specular multipath.

We formalize the problem of general multipath estimation as an L_1 optimization problem; this is the basis for our Sparse Reflections Analysis (SRA) approach. SRA is posed in such a way as to admit the computation of a global optimum, which is crucial for the robust cancellation of general multipath even in the presence of considerable measurement noise.

2. Speed. Prior work targeting diffuse multipath is very slow, typically requiring a few minutes per frame. By contrast, SRA is able to run at frame-rate, i.e. 30 fps. This speed would not be possible with a pure L_1-based approach; we accelerate SRA through the use of a novel dimension reduction which allows for a look-up table based approach. This gives extremely fast performance in practice.

2 Prior Work

Earlier work proposed removing multipath by using additional sensors based on structured light [7,6], while more recent work has attempted to remove the

[1] An exception is the contemporaneous work [1], see Section 2.

multipath directly from the sensor measurement itself. A summary of this more recent work is given in the following table.

Paper	**Multipath Type**	**Running Time Per Frame**	**Other Constraints**
Fuchs [8]	Diffuse only	10 minutes	
Fuchs *et al.* [9]	Diffuse only	60-150 seconds	
Jiménez *et al.* [13]	Diffuse only	"Several minutes"	
Dorrington *et al.* [5]	Two-path only	No information	
Godbaz *et al.* [10]	Two-path only* (see text)	No information	Requires 3 or 4 modulation frequencies
Kirmani *et al.* [15]	Two-path only	"Implementable in real-time"	Requires 5 modulation frequencies

The works of Fuchs [8], Fuchs *et al.* [9], and Jiménez *et al.* [13] all model only diffuse multipath (arising from Lambertian surfaces). [8] estimates the scene by a point cloud and updates the multipath from all pixels to all pixels by ray tracing. A single pass approximation is performed, whose complexity is quadratic in the number of pixels. [9] is a generalization of [8] to a spatially varying, unknown reflection coefficient. It requires an iterative solution consisting of multiple passes. [13] performs a somewhat different iterative optimization of a global function involving scene reconstruction and ray tracing. As is noted in the table, none of these methods are close to real-time, requiring anywhere between 1-10 minutes of processing per frame.

The works of Dorrington *et al.* [5], Godbaz *et al.* [10], and Kirmani *et al.* [15] all model two-path multipath, arising from specular surfaces. All of these methods work on a per pixel basis, using either closed form solutions [10,15] or optimizations [5]. Thus, while they do not report on their running times explicitly, it is reasonable to expect that they may be close to real-time. [15] requires 5 modulation frequencies; one of the two methods presented in [10], which is based on a Cauchy distribution approximation to the backscattering of a single return, requires 4. Several commercial ToF sensors, including the variant of Microsoft's Kinect for Windows beta sensor (henceforth "K4W") on which we perform our experiments, use only 3 modulation frequencies; thus, these methods are rendered impracticable for such sensors. By contrast, the second method presented in [10], which uses a more standard delta-function approximation to a single return, only requires 3 modulation frequencies. Given the additional fact that this method runs in or near real-time, it is therefore our nearest competitor.

We also note the extremely recent work of Bhandari *et al.* [1], whose publication was simultaneous with our submission to ECCV, and which we became aware of after submission. The formulation of the problem in [1] is quite similar to the formulation proposed in this paper; however, the solution is different. This is due to a crucial difference in the setups: while we assume 3 modulation frequencies, [1] assumes an extremely large number of modulation frequencies – 77, in fact. Such a massively large number of frequencies is infeasible in many scenarios of interest: for example, in dynamic scenes with fast movement, especially if there is high pixel resolution. [1] reports an integration time of 47 ms for

an image with 19,200 pixels; this means that purely based on integration time – i.e., even if the depth computation is instantaneous – the frame-rate would top out at 20 fps, and this on a very small image. By contrast, in our case the K4W integration time is 10 ms for 217,088 pixels – that is, the integration time is nearly 5 times smaller for 11 times as many pixels. This is possible because only 3 modulation frequencies are used. (We note that all of the competing methods except for [1] use between 2 and 5 modulation frequencies, for related reasons.)

Given the use of 77 modulation frequencies, [1] successfully solves for a sparse solution through a greedy approach, Orthogonal Matching Pursuit (OMP). When restricting to a small number of modulation frequencies – 3, rather than 77 – we found that a greedy optimization approach based on OMP is not effective. Instead, our method is based on an L_1-style global optimization, which works considerably better in practice. The drawback of L_1 is that it tends to be slow, which is why our novel LUT-based approach is crucial; it allows us to both gain from the accuracy of L_1, while not suffering from its speed disadvantage.

Finally, we note [14] and [12], which use related signal representations.

3 Sparse Reflections Analysis

3.1 The Multipath Representation

The ToF Measurement. We begin by describing the vector which is measured by a ToF sensor. For a given pixel, the sensor emits infra-red (IR) light modulated by several frequencies. The light bounces off a surface in the scene, and some of the light (depending on the reflectivity and orientation of the surface) is returned to the detector. For each of m modulation frequencies, this light is then integrated against sinusoids with the same frequency, such that the phase of the measurement v is based on the distance to the surface:

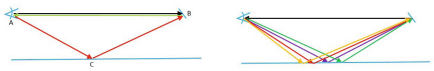
$$v \in \mathbb{C}^m, \text{ with } v_k = xe^{2\pi id/\lambda_k}, \quad k = 1, \ldots, m \tag{1}$$

where d is the distance to the surface, $\lambda_k = c/2f_k$ is half of the wavelength corresponding to the k^{th} modulation frequency f_k, and x is a real scalar corresponding to the strength of the signal received. A typical choice for the number of frequencies is $m = 3$; this is generally sufficient to prevent aliasing effects.

Multipath. Equation (1) assumes that there is no multipath in the scene. If there is single extra path (the "two-path" scenario), as shown in Figure 2(a), then the above equation is modified to

$$v_k = x_1 e^{2\pi id_1/\lambda_k} + x_2 e^{2\pi id_2/\lambda_k} \tag{2}$$

where d_1 and d_2 are the distances of the two paths, and x_1 and x_2 give the strengths of the two paths. If $d_1 < d_2$, then d_1 is the true distance and d_2 is the multipath component; and the ratio x_2/x_1 gives the strength of the multipath.



(a) (b)

Fig. 2. Illustration of Multipath. (a) Two-Path Multipath. The camera (A) and surface (B) are shown in blue, and light rays in other colours. The correct path is A-B-A; the second, incorrect path is A-C-B-A. (b) Diffuse Multipath. This results from small reflections from many nearby points, shown illustratively as four colored paths.

Of course, one can have more general multipath, including: three or more paths; diffuse plus two-path; and so on. Equation (2) generalizes naturally as

$$v_k = \sum_{j=1}^{n} x_j e^{2\pi i d_j/\lambda_k} \tag{3}$$

where d_j is over the relevant interval. In typical examples, we take the range of object distances to be 20 cm to 450 cm, with increments of 1 cm. Thus, in this case $n = 431$. The vector x is referred to as the **backscattering**.

Equation (3) includes the case of diffuse multipath. An ideal Lambertian surface receives light from a given direction, and reflects infinitesimal amounts in all directions. In fact, an infinite number of nearby points on the surface reflect infinitesimal amounts, and the result is finite. This is shown in Figure 2(b). Diffuse multipath typically has the form $v_k = x_\ell e^{2\pi i d_\ell/\lambda_k} + \sum_{j=\ell+\Delta}^{n} x_L(d_{j-\Delta}) e^{2\pi i d_j/\lambda_k}$ for some $\Delta \geq 0$. The shape of $x_L(\cdot)$ can be determined by looking at simulations of diffuse multipath, and turns out to be well approximated by $x_L(d) \approx Ad^\alpha e^{-\beta d}$, where α and β depend on the geometry of the underlying scene.

We can rewrite Equation (3) in vector-matrix form as $v = \Phi x$, where $\Phi \in \mathbb{C}^{m\times n}$ and $x \in \mathbb{R}^n$. We further turn complex measurements into real ones, by stacking the real part on top of the imaginary part, and abuse notation by denoting the $2m$-dimensional result also as v. We do the same with Φ, yielding

$$\Phi \in \mathbb{R}^{2n\times m}, \quad \text{with} \quad \Phi_{kj} = \begin{cases} \cos(2\pi d_j/\lambda_k) & \text{if } k = 1,\ldots,m \\ \sin(2\pi d_j/\lambda_{k-m}) & \text{if } k = m+1,\ldots,2m \end{cases}$$

Then we may still write

$$v = \Phi x \tag{4}$$

but now all quantities are real.

A Characterization of the Backscattering. Let us now characterize the class of backscatterings x which capture the multipath phenomenon.

Property 1: Non-Negativity The first property is that x is non-negative: $x \geq 0$. That is, there can only be positive or zero returns for any given distance.

Property 2: Compressibility The second property is more interesting. We saw that the two-path scenario involved an x which was zero at all indices except for two (corresponding to d_1 and d_2 in Equation (2)). Such an x is sparse.

On the other hand, the diffuse multipath, which has the form $x_L(d) \approx Ad^{\alpha}e^{-\beta d}$, is not sparse. Rather, its discretized version has the following property: when the x coefficients are sorted from greatest to smallest, the resulting vector falls off quickly to 0. This property is referred to as compressibility.

Formally, given a vector $x = (x_1, \ldots, x_n)$, let $(x_{I(1)}, \ldots, x_{I(n)})$ denote the vector sorted in descending order. Then x is compressible if $x_{I(i)} \leq Ri^{-1/r}$ with $r \leq 1$. That is, the sorted entries of x fall off as a power law.

3.2 The SRA Algorithm

We have represented multipath via the backscattering x, and have further characterized the important properties of the backscattering – non-negativity and compressibility. We now go on to show how to use this information to cancel the effects of multipath, and hence find a robust and accurate depth estimate from the raw ToF measured vector.

Multiplicity of Solutions. We are given v, and we know that a backscattering x has generated v; i.e. following Equation (4), we have $v = \Phi x$. Given that $x \in \mathbb{R}^n$ and $v \in \mathbb{R}^{2m}$ where $n \gg 2m$, there are many possible x's which can generate v. But our characterization of the backscattering says that x is non-negative and compressible, which leads to a much more restrictive set of possible backscatterings.

In fact, due to sensor noise we will not have $v = \Phi x$ exactly. Rather, we may expect that $v = \Phi x + \eta$, where η is generally taken to be Gaussian noise, with zero mean and known covariance matrix C.[2]

L_0 Minimization. Let us suppose, for the moment, that x is *sparse* rather than compressible – that is, x has a small number of non-zero entries. The number of non-zero entries of x is often denoted as it's 0-norm, i.e. $\|x\|_0$. In this case, one would like to solve the following problem:

$$\min_{x \geq 0} \|x\|_0 \quad \text{subject to} \quad (\Phi x - v)^T C^{-1} (\Phi x - v) \leq \epsilon^2 \|v\|^2 \tag{5}$$

for some parameter ϵ, which we fix to be 0.05 in our experiments. (Note that we have indicated the non-negativity of x under the min itself.) That is, we want the sparsest backscattering x which yields the measurement vector v, up to some noise tolerance. Unfortunately, the above problem, which is combinatorial in nature, is NP-hard to solve.

L_1 Minimization. However, it turns out that subject to certain conditions on the matrix Φ, solving the problem

$$\min_{x \geq 0} \|x\|_1 \quad \text{subject to} \quad (\Phi x - v)^T C^{-1} (\Phi x - v) \leq \epsilon^2 \|v\|^2 \tag{6}$$

will yield a similar solution [3,2,4] to the optimization in (5). Note that the only difference between the two optimizations is that we have replaced the 0-norm

[2] Due to the physics of the sensor, there is often a shot noise component involved. We will ignore this consideration, though our method can be adapted to handle it.

with the 1-norm. The key implication is that the optimization in (6) is *convex*, and hence may be solved in polynomial time.

In fact, the conditions mentioned in [3,2,4], such as the Restricted Isometry Property, are generally not satisfied by our matrix Φ. This is due to the redundancy present in nearby columns of Φ, i.e. nearby columns tend to have high inner products with each other. Nevertheless, we can use the optimization in (6), with the understanding that certain theoretical guarantees given in [3,2,4] do not hold. Note that this is in the same vein as other computer vision work, such as the celebrated paper by Wright *et al.* on robust face recognition [16], which used L_1 optimization under conditions which differed from those specified in [3,2,4].

Until now we have been assuming that x is sparse, rather than compressible. It turns out, however, that even if x is compressible and not sparse, then solving the L_1 optimization in (6) still yields the correct solution [3,2,4].

L_1 with L_1 Constraints. Although the optimization (6) is convex, it is a second-order cone program which can be slow to solve in practice. We therefore make the following modification. Note that the L_2 constraint above may be written $\|C^{-1/2}(\Phi x - v)\|_2 \leq \epsilon \|v\|_2$. We may consider approximating these constraints by their equivalent L_1 constraints, i.e. $\|C^{-1/2}(\Phi x - v)\|_1 \leq \epsilon \|v\|_1$. In this case, the resulting optimization becomes

$$\min_{x \geq 0} \|x\|_1 \quad \text{subject to} \quad \|C^{-1/2}(\Phi x - v)\|_1 \leq \epsilon \|v\|_1 \tag{7}$$

The advantage of this formulation over (6) is that it can be recast as a linear program. As such, it can be solved considerably faster. To perform the conversion, first notice that since $x \geq 0$, $\|x\|_1 = \sum_{i=1}^{n} x_i = \mathbf{1}^T x$. Second, note that the constraint $\|z\|_1 \leq \gamma$ for $z \in \mathbb{R}^\ell$ can be converted into the set of linear constraints $Q_\ell z \leq \gamma \mathbf{1}$, where Q_ℓ is a $2^\ell \times \ell$ matrix, whose rows consist of all elements of the set $\{-1, +1\}^\ell$. While this might be prohibitive for ℓ large, in our case $\ell = 2m$, and we generally have $m = 3$; this leads to 64 extra constraints, much fewer than the number of non-negativity constraints. This yields the linear program

$$\min_{x \geq 0} \mathbf{1}^T x \quad \text{subject to} \quad Ax \leq b$$

where $A = Q_{2m} C^{-1/2} \Phi$ and $b = Q_{2m} C^{-1/2} v + \epsilon \|v\|_1 \mathbf{1}$.

Computing Depth from Backscattering. The various optimization problems we have just described yield the backscattering x. Of course, in the end our goal is an estimate of the depth; we now explain how to extract the depth from x.

The main path must have the shortest distance; this results from the geometry of the imaging process. Thus, we have simply that the depth corresponds to the first non-zero index of x, i.e. the index $i_1(x) \equiv \arg\min_i \{i : x_i > 0\}$. Then the depth is just $\delta = d_{i_1(x)}$. In practice, due to numerical issues there will be many small non-zero elements of x. Thus, we take $i_1(x) \equiv \arg\min_i \{i : x_i > c \max_{i'} x_{i'}\}$ for some small c; typically, we use $c = 0.01$.

If we have a reasonably accurate noise model, we can be more sophisticated. For each peak of the backscattering x, we can compute the probability that the peak

is generated by noise rather than signal. Then the probability that the first peak is the true return is just one minus the probability that it is generated by noise; the probability that the second peak is the true return is the probability that the first peak is generated by noise times one minus the probability that the second peak is generated by noise; and so on. If no return has probability greater than a threshold (e.g. 0.9), we can "invalidate" the pixel – that is, declare that we do not know the true depth. We will make use of this kind of invalidation in Section 5.

4 Fast Computation

SRA allows us to compute the backscattering x from a sensor measurement v; from the backscattering, one can compute the depth. The issue that now arises is related to the speed of the computation. Solving the optimization in (7) typically requires about 50 milliseconds per instance on a standard CPU. Given that a ToF image may consist of several hundred thousand pixels, this yields on the order at least an hour per frame. To achieve a frame rate of 30 Hz, therefore, a radically different approach is needed. The method we now describe allows for SRA to run at frame rate on ordinary hardware, for images of size 424×512.

Dimension Reduction: Motivation. Real-time computation is often aided by performing pre-computation in the form of a look-up table (LUT). If we construct a LUT directly on the measurement vector v, then the table will be $2m$-dimensional, as this is the dimensionality of v. It is easy to see that multiplying v by a scalar does not change the results of any of the SRA optimizations, except to scale x by the same scalar. Thus, one can easily normalize v so that its L_2 norm (or L_1 norm) is equal to 1, yielding a reduction of a single dimension. The resulting table will then be $(2m-1)$-dimensional.

Our goal is a further reduction of a single dimension, to $(2m-2)$ dimensions. Recall that the size of an LUT is *exponential* in its dimension; thus, the reduction of two dimensions reduces the total memory for the LUT by a factor of L^2, where each dimension has been discretized into L cells. This reduction makes the LUT approach feasible in practice.

A Useful Transformation. Let us return to the complex formulation of the problem; this will make the ensuing discussion easier, though it is not strictly necessary. Let us define the $m \times m$ complex matrix $F_{s,\Delta}$ by

$$F_{s,\Delta} = s \cdot \mathrm{diag}(e^{-2\pi i \Delta/\lambda_1}, \ldots, e^{-2\pi i \Delta/\lambda_m})$$

where $\Delta \in \mathbb{R}$; $s > 0$ is any real positive scalar; and diag() denotes the diagonal matrix with the specified elements on the diagonal. Then we have the following theorem, from which we can derive our dimension reduction.

Theorem 1. *Let x^* be the solution to the optimization (6), and let $x^*_{s,\Delta}$ be the solution to (6) with $F_{s,\Delta}v$ replacing v and $F_{s,\Delta}\Phi$ replacing Φ. Suppose that the covariance C is diagonal, and satisfies $C_{jj} = C_{j+m,j+m}$. Then $x^*_{s,\Delta} = x^*$.*

The proof is included in the supplementary material. The implication of Theorem 1 is that multiplying both the measurement v and the matrix Φ by the

matrix $F_{s,\Delta}$ does not change the backscattering (assuming the theorem's conditions on the covariance matrix hold, which is a reasonable model of sensor noise). Of course, the corresponding range of distances has been shifted by by $-\Delta$, so in extracting the depth from $x^*_{s,\Delta}$, one must add on Δ afterwards.

Before going on, we note that Theorem 1 applies to optimization (6) rather than (7), which we use in practice. However, as (7) is a reasonable approximation to (6), we proceed to use Theorem 1 to construct our dimension reduction.

The Canonical Transformation. The Canonical Transformation is derived from $F_{s,\Delta}$ by a particular choice of s and Δ. Let $k \in \{1, \ldots, m\}$ be a specific frequency index; then Canonical Transformation of v, $\rho^{(k)}(v)$, is given by

$$\rho^{(k)}(v) \equiv F_{s,\Delta} v \quad \text{with} \quad s = \|v\|^{-1}, \ \Delta = \lambda_k(\angle v_k / 2\pi)$$

where $\angle v_k$ denotes the phase of v_k, taken to lie in $[0, 2\pi)$. It is easy to see that $\rho^{(k)}$ has the following property. The k^{th} element of $\rho^{(k)}(v)$ is real, i.e. has 0 phase. Furthermore, the k^{th} element of $\rho^{(k)}(v)$ may be found from the other elements of $\rho^{(k)}(v)$ by $\rho^{(k)}_k(v) = \left(1 - \sum_{k' \neq k} |\rho^{(k)}_{k'}(v)|^2\right)^{1/2}$.

In other words, in the Canonical Transformation, one of the elements is *redundant*, in that it is completely determined by the other elements. Hence, this element can be removed without losing information. Of course, the component is complex, meaning that we have removed two real dimensions, hence enabling the promised dimension reduction from $2m$ to $2m - 2$. A LUT can be built on the remaining $2m - 2$ dimensions, simply by discretizing over these dimensions.

Note that having transformed v by $F_{s,\Delta}$ (with s and Δ given by the Canonical Transformation), we must also apply $F_{s,\Delta}$ to Φ in order to use Theorem 1. In fact, this is straightforward: this transformation simply "shifts" the columns of Φ. So if before they represented distances in the range $[D_{min}, D_{max}]$, they now represent distances in the range $[D_{min} - \Delta, D_{max} - \Delta]$. Rather than actually do the shifting, we simply enlarge this range. Note that the minimal value that Δ can take on is 0, while the maximal value is λ_k; thus, after the Canonical Transformation the potential distances can now fall between $D_{min} - \lambda_k$ and D_{max}. Thus, the matrix Φ is now enlarged to have columns corresponding to distances in the range $[D_{min} - \lambda_k, D_{max}]$. A natural method for choosing k is to keep the above range as small as possible, and hence to choose the k corresponding to the smallest half-wavelength λ_k.

5 Experiments

5.1 Running Time

As one of the main claims of this paper is a fast algorithm, we begin by presenting the speed of the algorithm. We have benchmarked SRA on images of size 424×512, which are the standard for Microsoft's Kinect for Windows beta sensor ("K4W"). The code is run on an Intel Core i7 processor, with 4 cores, 8 logical processors, and a clockspeed of 2.4 GHz. The code runs in 31.2 milliseconds per

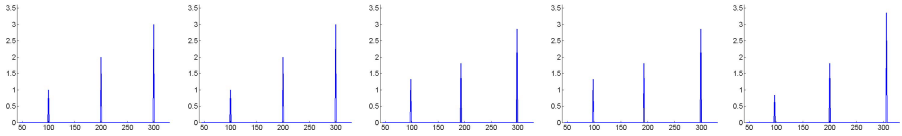
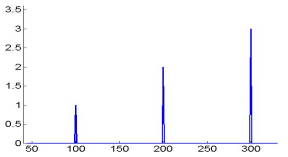
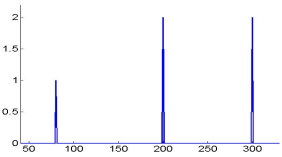


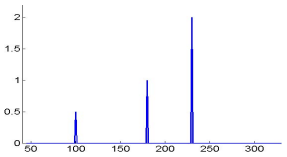
Fig. 3. Three-Path Simulation. Left: true backscattering. Following 4 plots: SRA reconstruction of the backscattering, for SNR= $\infty, 20, 10, 5$. See discussion in the text.



SNR	SRA	Godbaz	ML
∞	**0.0**	15.0	111.0
20	**1.9**	114.7	111.0
10	**3.7**	119.3	111.0
5	**8.1**	109.2	109.0



SNR	SRA	Godbaz	ML
∞	**0.0**	8.4	29.0
20	**2.1**	108.4	29.0
10	**4.1**	117.4	29.0
5	**8.6**	111.9	30.0



SNR	SRA	Godbaz	ML
∞	**0.0**	8.0	29.0
20	**8.7**	91.7	29.0
10	**17.4**	86.2	29.0
5	31.7	76.4	**31.0**

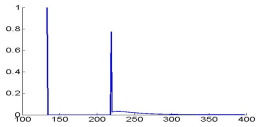
Fig. 4. Three-Path Simulation. Each column shows the true backscattering (top), and the median absolute error in cm of three algorithms under various noise levels (bottom). The best algorithm is indicated in bold. See discussion in the text.

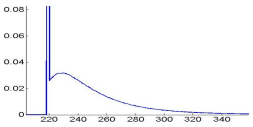
frame, which is real-time given a frame-rate of 30 fps. Note that our code is largely unoptimized Matlab (the only optimization we make is to use the CPU's 8 logical processors for parallelization); the speed comes from the LUT-based approach. It is to be expected that optimized C code would be even faster.

5.2 General Multipath: Examples

Specular Three Path. We begin by motivating the relevance of general multipath. Specular multipath with three or more paths results naturally from simple scene geometries. Suppose that we have the geometry shown in Figure 2(a), where the object (B) lies on a Lambertian surface and the scene element (C) is taken to be purely specular. Then it can be shown that by varying the position and normal of the scene element, we can generate any relative amplitudes we wish between the direct (A-B-A) and interfering (A-C-B-A) paths, as well as any pair of path distances; please see the supplementary material for more details. Generating three (or more) paths then becomes straightforward, by adding an extra (or multiple extra) specular surfaces to the scene.

Figure 3 shows an example of three path specular interference. The leftmost plot shows the true backscattering, corresponding to object distances of 100, 200, and 300 cm, with amplitudes in the ratio 1:2:3. That is, the multipath is $2 + 3 = 5$ times stronger than the initial return. Moving from left to right, the following four plots show the backscattering computed by the SRA algorithm under different levels of noise: SNR= $\infty, 20, 10, 5$. Note that the backscattering extracted is exactly correct for the case of SNR= ∞, and remains fairly close to





SRA	Godbaz	ML
1.9	32.7	5.9

Fig. 5. Combined Diffuse and Specular Multipath. Left: true backscattering. Middle: detail of the backscattering, showing the diffuse part. Right: Absolute error in cm.

the true backscattering as the noise level increases. Indeed, for the highest noise level of SNR= 5, the peaks have moved to 97, 200, and 306 cm (movements of -3, 0, and 6 cm resp.), which yields an error of 3 cm in depth estimation. (Note the amplitudes have changed as well, moving less than 20% in all cases.)

We now move to a more quantitative comparison of SRA with other alternatives in the case of three paths. As our aim is to show the need to model more general multipath, we run against two strong two-path alternatives: the algorithm of Godbaz *et al.* [10] (our most natural competitor, for reasons described in Section 2) and the maximum likelihood (ML) two-path solution. Note that the ML solution is not a practical algorithm, as it requires a slow, exhaustive examination of all pairs of paths; but we include it as it represents the best possible two path solution. Figure 4 shows three separate configurations; the top row shows the true backscatterings, while the tables below show the performance of SRA vs. the two alternatives. The performance is given by the median absolute error of the depth; as we are adding noise, we average over 1,000 samples.

We note that SRA outperforms the two alternatives, and does so by a wide margin once even a small amount of noise is added. In the first example, the accuracies of SRA are very high, staying under 4 cm for SNR levels up to 10; the second example is similar. The third case has been chosen to be more difficult for SRA: the multipath is 6 times stronger than the original path, and the second and third returns are fairly closely spaced. SRA's errors here are higher, though still considerably lower than the alternatives (except in the case of SNR = 5, where performance is similar to ML). In all three cases, Godbaz gives reasonable results in the noiseless case, but fails once even a small amount of noise (SNR = 20) is added. ML is much more resistant to noise, but does not give very high accuracy in any of the examples, regardless of noise level.

Combined Diffuse and Specular Multipath. We now show an example which leads to a combination of diffuse and specular multipath. The geometry is again simple, and consists of an object and a single plane; both object and plane have both specular and diffuse reflectivity. The backscattering, which we generate by use of our own light-transport simulator, is shown in Figure 5; note the fact that the backscattering is no longer sparse (but is still compressible).

The depth estimate errors are shown in Figure 5. The method of Godbaz generates fairly large errors. ML is considerably better: it turns out that there is a reasonably good two path approximation to the measurement v produced by this backscattering. SRA produces the lowest error: not surprisingly, allowing for a more complex backscattering – as SRA does – leads to the best result.

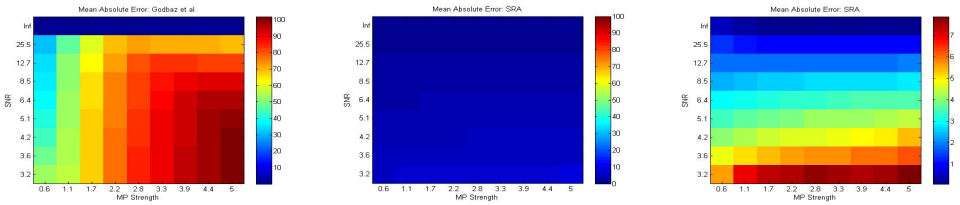


Fig. 6. Two-Path Simulation: Mean Absolute Error (cm). Left: Godbaz. Middle: SRA, with the same scale as Godbaz, i.e. [0, 100] . Right: SRA, with errors rescaled to [0,8].

5.3 Comprehensive Two-Path Evaluation

Setup. We have shown the ability of SRA to deal with general multipath. However, standard two-path interference is a very important case, and we would like to show SRA's capabilities in this regime. We challenge SRA in two ways: by simulating high multipath, up to a factor of 5 times as high as the direct return; and by simulating high noise regimes. We again compare against Godbaz *et al.* [10], our most natural competitor (for reasons described in Section 2).

In particular, we simulate returns of the form $v_k = x_1 e^{2\pi i d_1/\lambda_k} + x_2 e^{2\pi i d_2/\lambda_k} + \eta_k$; the noise η_k has independent real and imaginary components, and is taken to be Gaussian with variance σ^2 for each component. There are two critical parameters: (1) Multipath Strength is defined as x_2/x_1, and takes on values in the set $\{0.6, 1.1, 1.7, 2.2, 2.8, 3.3, 3.9. 4.4, 5.0\}$. (2) SNR is defined as $x_1/\sqrt{6}\sigma$, since there are 6 independent noise components to the measurement. It takes on values in the set $\{\infty, 25.5, 12.7, 8.5, 6.4, 5.1, 4.2, 3.6, 3.2\}$.

We also allow d_1 to vary over values between 20 cm and 380 cm, and the return separation $d_2 - d_1$ to vary between 40 cm and 250 cm; and each instance is generated with many noise vectors. In total, we generate 261,000 examples.

We visualize the results in Figure 6, in which we show the mean absolute error (MAE) of the depth estimates as a function of multipath strength and SNR. Each square corresponds to a (multipath strength, SNR) pair; we average over all examples falling into the square to compute the MAE.

Discussion. In general, SRA's behavior is as we would imagine: as multipath strength increases, MAE increases; likewise, as SNR decreases, MAE increases.

There are two notable facts about the results. First, the MAE is quite small for "realistic" values of multipath strength and SNR. Focus on the upper left rectangle of Figure 6 consisting of SNRs from ∞ to 8.5 and multipath strengths from 0.6 to 2.2; note that these are still quite challenging values. In this regime, the MAE is low: it is less than 2.6 cm in all squares, with an average of 1.4 cm.

Second, SRA's performance degrades in a graceful way: the MAE increases gradually in both dimensions. In fact, even when the multipath is 5 times stronger than the true return and the SNR has a low value of 3.2, the MAE is 7.9 cm, which is a very reasonable error in such circumstances.

It is interesting to compare SRA's MAE with that of Godbaz *et al.* [10], recalling that Godbaz's algorithm was designed with two-path multipath in mind.

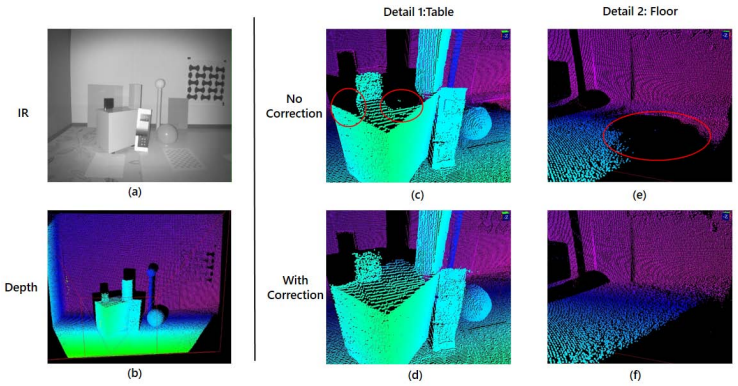


Fig. 7. "Geometric". (a) IR image. (b) SRA depth estimate. (c,e) Optimal depth estimates without multipath correction; errors are circled in red. (d,f) SRA depth.

Godbaz gives good results when there is no noise: the MAE is nearly 0. Once even a small amount of noise is added in, however, Godbaz's performance drops significantly. For example, with SNR = 25.5, the errors range from 30.8 cm (MP Strength = 0.6) to 69.7 cm (MP Strength = 5). Performance worsens significantly as SNR decreases.

5.4 Real Images

We now discuss the results of running SRA on three different challenging images. The images are collected using a variant of the K4W sensor, which has $m = 3$ modulation frequencies – 15, 80, and 120 MHz – and a resolution of 424×512. The images have been placed in the publicly available repository `http://research.microsoft.com/sparsereflections/`; researchers should also be able to acquire similar images on their own, using the K4W sensor.

Unfortunately, we are not able to compare with any of the real-time competing methods due to their incompatibility with the K4W imaging setup. Specifically, Godbaz *et al.* [10] and Dorrington *et al.* [5] require a very particular relation between the modulation frequencies, which is not satisfied by K4W's modulation frequencies. Kirmani *et al.* [15] requires 5 modulation frequencies, while a second method described in [10] requires 4; K4W uses only 3 modulation frequencies. Thus, for comparison purposes in this section, we run SRA against a variant of SRA which looks for the optimal single path which best describes the sensor measurement, which we call "Opt-Single".

"Strips" Image. See Figure 1. This image is a simple scene – a floor and a wall; however, the floor is composed of strips of different materials, each of which has different reflectance properties. This can be seen clearly in the IR image in Figure 1.

The more specular materials tend to lead to very high multipath: a path which bounces off the wall, and from there to the floor, will generally have strength higher than the direct return from the floor. This is due to the fact that the direct path is nearly parallel to the floor, leading to a weak direct return.

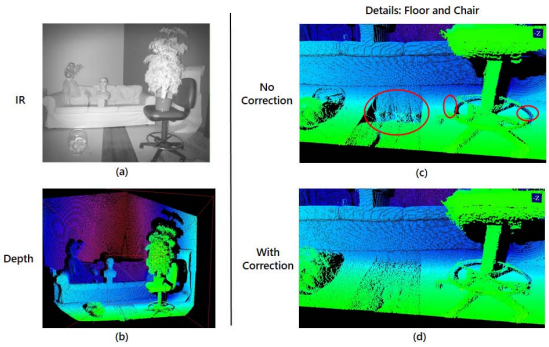


Fig. 8. "Living Room". (a) IR image. (b) SRA depth estimate. (c) Optimal depth estimate without multipath correction; errors are circled in red. (d) SRA depth estimate.

Results of the depth reconstruction are shown in Figure 1. To see the effect of multipath removal, we compare SRA's depth estimate to Opt-Single. In Figure 1, one can see details of the scene which show the effect of "specular floor" multipath: the wall is effectively reflected into the floor, leading to grossly inaccurate estimates for the floor. By contrast, SRA reconstructs a very clean floor, which is seen to be almost completely flat, the exceptions being a few small depressions of less than 4 cm.

Note in Figure 1 that SRA invalidates a number of pixels at the top right part of the image, corresponding to a patterned wall-hanging; this is due to the low reflectivity of this part of the scene, leading to very noisy measurements.

"Geometric" Image. See Figure 7. This image consists of a similar set-up to "Strips", but with various geometric objects inserted. These objects include many sharp angles, as well as several thin structures. SRA reconstructs the scene quite well, see Figure 7. Again, note the scene details in Figure 7, which show the performance of SRA vs. Opt-Single. Much of the surface of the rectangular table is lost without accounting for multipath, whereas SRA is able to reconstruct it. And the corner where the wall joins the floor is lost – actually reflected into the floor – without accounting for multipath, whereas SRA recovers it.

"Living Room" Image. See Figure 8. This image consists of a couch as well as a number of objects inserted. As in "Strips", there is a strip of reflective material on the floor; Opt-Single has trouble accounting for this strip, as can be seen in Figure 8. Perhaps more interestingly, the thin structure which is the wheel at the base of the swivel chair is largely reconstructed by SRA, whereas large portions of it, on both the right and left sides, are lost by Opt-Single.

6 Conclusions

We have presented the SRA algorithm for removing multipath interference from ToF images. We have seen that the method is both general, dealing with many

types of multipath, and fast. SRA has been experimentally validated on both synthetic data as well as challenging real images, demonstrating its superior performance.

References

1. Bhandari, A., Kadambi, A., Whyte, R., Barsi, C., Feigin, M., Dorrington, A., Raskar, R.: Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization. Optics Letters 39(6), 1705–1708 (2014)
2. Candes, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Communications on Pure and Applied Mathematics 59(8), 1207–1223 (2006)
3. Candes, E.J., Tao, T.: Decoding by linear programming. IEEE Transactions on Information Theory 51(12), 4203–4215 (2005)
4. Donoho, D.L.: Compressed sensing. IEEE Transactions on Information Theory 52(4), 1289–1306 (2006)
5. Dorrington, A.A., Godbaz, J.P., Cree, M.J., Payne, A.D., Streeter, L.V.: Separating true range measurements from multi-path and scattering interference in commercial range cameras. In: IS&T/SPIE Electronic Imaging, pp. 786404. International Society for Optics and Photonics (2011)
6. Falie, D., Buzuloiu, V.: Further investigations on ToF cameras distance errors and their corrections. In: European Conference on Circuits and Systems for Communications (ECCSC), pp. 197–200 (2008)
7. Falie, D., Buzuloiu, V.: Distance errors correction for the time of flight (ToF) cameras. In: IEEE International Workshop on Imaging Systems and Techniques, IST 2008, pp. 123–126. IEEE (2008)
8. Fuchs, S.: Multipath interference compensation in time-of-flight camera images. In: International Conference on Pattern Recognition (ICPR), pp. 3583–3586 (2010)
9. Fuchs, S., Suppa, M., Hellwich, O.: Compensation for multipath in ToF camera measurements supported by photometric calibration and environment integration. In: Chen, M., Leibe, B., Neumann, B. (eds.) ICVS 2013. LNCS, vol. 7963, pp. 31–41. Springer, Heidelberg (2013)
10. Godbaz, J.P., Cree, M.J., Dorrington, A.A.: Closed-form inverses for the mixed pixel/multipath interference problem in amcw lidar. In: IS&T/SPIE Electronic Imaging, pp. 829618. International Society for Optics and Photonics (2012)
11. Han, J., Shao, L., Xu, D., Shotton, J.: Enhanced computer vision with Microsoft Kinect sensor: A review. IEEE Transactions on Cybernetics 43(5), 1318–1334 (2013)
12. Heide, F., Hullin, M.B., Gregson, J., Heidrich, W.: Low-budget transient imaging using photonic mixer devices. ACM Transactions on Graphics (TOG) 32(4), 45 (2013)
13. Jiménez, D., Pizarro, D., Mazo, M., Palazuelos, S.: Modelling and correction of multipath interference in time of flight cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 893–900 (2012)

14. Kadambi, A., Whyte, R., Bhandari, A., Streeter, L., Barsi, C., Dorrington, A., Raskar, R.: Coded time of flight cameras: sparse deconvolution to address multipath interference and recover time profiles. ACM Transactions on Graphics (TOG) 32(6), 167 (2013)
15. Kirmani, A., Benedetti, A., Chou, P.A.: Spumic: Simultaneous phase unwrapping and multipath interference cancellation in time-of-flight cameras using spectral methods. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2013)
16. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), 210–227 (2009)