2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Depth Completion by Rescaling Monocular Depth Estimates via Compressed Sensing

Daoxin Zhong[1], Jun Li[1], Yeshas Thadimari[1], Meng Yee (Michael) Chuah[1]

***Abstract*— Depth completion is the challenge of recovering a dense depth map from an RGB image and corresponding sparse depth measurements. Many modern depth completion strategies often rely on deep neural networks, using a monocular depth estimation (MDE) backbone to generate an initial dense depth map from the RGB image. This estimate is then further refined with the help of auxiliary network components that utilise the sparse depth measurements to improve accuracy and restore fine-grained depth details. However, such approaches introduce additional model parameters and require domain-specific fine-tuning, making them impractical for resource-constrained robotics applications. In this paper, we propose an alternative refinement strategy based on compressed sensing. Using the Discrete Cosine Transform (DCT) as our basis, we construct a ratio matrix that rescales the estimated depth map to align with measured ground truth data. Our experiments demonstrate that this method can significantly reduce the RMSE and MAE of the initial MDE estimate by more than a factor of 15. Furthermore, the proposed approach can outperform state-of-the-art depth completion models at sampling ratios above 50 percent, while also reducing the overall GPU VRAM requirements. This pipeline is modular and compatible with any existing MDE model with no additional training, making it particularly suitable for deployment on GPU-constrained robotic platforms in previously unseen environments.**

I. Introduction

RGB-D cameras have become ubiquitous sensors on many robotic platforms, driven in part by their simpler mechanical design and greater affordability compared to LiDAR systems. They can detect light from the infrared (IR) spectrum, using both the IR patterns projected by their emitters and other RGB features to measure depth via stereo matching. Their compact active-pixel sensors are able to resolve more points within the physical scene, thus producing denser depth maps compared to time-of-flight sensors found in LiDARs.

However, despite these advantages, the performance of IR depth cameras are limited by many practical constraints. Stereo matching can fail under adverse lighting conditions or when observing uniform and featureless surfaces. This is especially true in outdoor environments, where the projected IR pattern becomes saturated under sunlight. Therefore, the measured pixel depth map is almost never complete, as there will always be some missing patches with empty depth information near the boundaries of objects or in areas under strong illumination.

Fortuitously, the widespread availability of depth cameras has led to the creation of numerous RGB-D image datasets [1], [2], generating research interest in learning-based depth estimation models using RGB images alone, a technique more commonly known as Monocular Depth Estimation (MDE) [3]. Because the RGB image is usually more complete than the depth image, its depth estimate will contain fewer spatial gaps and exhibit more coherent structural details. However, despite recent improvements in model architectures [4] and the availability of large training datasets [5], this estimate is only accurate up to an unknown scale and can struggle with forced perspective scenes, hence limiting the reliability of RGB-only approach to depth estimation (See Fig 1).

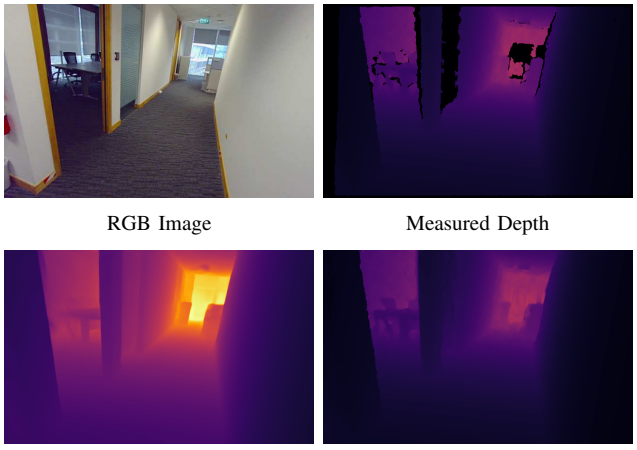

Fig. 1: Example of an indoor scene captured with an RGB-D camera. Note the missing pixels in the measured depth image as stereo matching fails along object boundaries and transparent surfaces. In contrast, the predicted depth from the MDE model (Metric3D-Small) is more complete but fails to estimate the room's scale, hence the brighter pixels at the further points in the scene. Using the available depth measurements, our proposed pipeline can compute a ratio matrix using compressed sensing that rescales the MDE depth to better align with the ground truth.

To address these limitations, researchers have turned to a related problem known as Depth Completion [6][7], which aims to reconstruct a dense depth map using an RGB image in conjunction with sparse depth measurements. This hybrid approach helps correct scale ambiguity and improve robustness in complex scenes. However, since many of these methods are built upon some MDE backbones, they include additional model parameters that require domain-specific fine-tuning. This also increases their GPU VRAM usage at runtime, making them more computationally intensive than pure MDE models. For these reasons, depth comple-

This research is supported by the Agency for Science, Technology and Research (A*STAR).

[1]Authors are with the Institute for Infocomm Research (I^{2}R), A*STAR, Singapore. michael_chuah@a-star.edu.sg

tion models are usually less suitable for real-time robotics applications, where power consumption restrictions limits computational capacity.

To address these limitations, we asked: Given an RGB image with some sparse depth measurement, *can the MDE depth estimate be rescaled via algorithmic methods directly?* Our goal is to estimate a ratio matrix that allows us to perform pixelwise rescaling of the estimated depth map without introducing any new model parameters or fine-tuning. To this end, we draw inspiration from the classical signal processing technique known as Compressed Sensing. Proposed by *Candè et al* [8], this is a strategy that enables signal reconstruction at sampling rates below even the Nyquist limit, achievable by exploiting the fact that many real-world signals are sparse or compressible in some known basis. We postulate that the ratio matrix between estimated and ground truth depth can be similarly represented in the 2D Discrete Cosine Transform (DCT), allowing for its recovery through sparse depth measurements. This enables us to easily rescale the output of any MDE model, thus presenting a viable, practical alternative to depth completion for robotics applications.

In summary, in this paper, we have

- proposed a compressed sensing-based pipeline that uses sparse depth measurements to rescale the output of an MDE model;
- benchmarked its performance using different MDE models and against a state-of-the-art depth completion model; and
- analysed the practical benefits and limitations of our approach.

II. Background

Compressed sensing was a widely used approach for RGB [9] and depth in-painting [10] prior to the emergence of modern deep learning techniques. It assumes that natural signals, such as images, are highly compressible and can be written in an appropriate basis with only a few modes active, thus reducing the number of values needed for accurate representation [11]. This sparsity constraint allows the image to be reconstructed from a number of sparse, incoherent measurements that satisfy the Restricted Isometry Property (RIP) [12]. In practice, it is usually sufficient to sample these measurements at random, as this has been known to satisfy RIP with high probability [11][13]. A powerful technique with broad applications across various domains, research within the realm of image processing has focused primarily on evaluating the merits of representing the 2D image using different basis functions, such as Truncated Nuclear Norms [14] or Discrete Daubechies Wavelets [15]. Other works examine how the minimisation of other auxiliary cost functions, such as the signal's Total Variation [10][16] or ℓ_0 gradient norm [17] can improve the quality of the final reconstructed image. To the best of our knowledge, no prior work has investigated the utility of this strategy for rescaling MDE outputs.

However, despite its strengths, there exist practical limits on the effectiveness of the compressed sensing approach, especially when large blocks of pixels are missing in the image [14], as there is simply not enough local information to estimate the signal in empty regions. Thus, as GPUs become more readily available, modern learning-based neural networks have become the preferred approach to depth completion. Earlier works in the field were focused exclusively on MDE, leveraging learned representations from large datasets to generate disparity maps via RGB images [18]. Since purely monocular methods fail under forced perspective or scale ambiguity, depth completion models emerged as a strategy to ground predictions using sparse depth measurements, thereby improving both model robustness and accuracy. While the utility of earlier models was limited by the accuracy of their MDE backbones [19], this changed with the introduction of Visual Transformer-based architectures [20]. They proved highly effective as an encoder for depth estimation tasks [21], leading to significant improvements in the accuracy of both MDE [22][23] and depth completion models [24]. However, end-to-end depth-completion models come with notable drawbacks. They tend to have more parameters than their MDE backbones, requiring additional domain-specific fine-tuning[24][25]. This also increases both latency and GPU resource demands at run-time, making them impractical for deployment on resource-constrained robotic platforms. This was the motivation behind our investigation into a more modular alternative: to replace the final rescaling or fusion stage with an algorithmic method instead of learned components, thus removing the need for any fine-tuning or more GPU resources.

III. Methodology

A. Compressed Sensing

Let $x \in \mathbb{R}^n$ be a signal consisting of n elements. Suppose that the signal is sampled at a subset of indices, represented by the set $\Omega \subset \{1, \ldots, n\}$. Its remaining entries are assumed to be zero, and their indices form the complement set Ω^c. Let $\mathcal{P}_\Omega$ be the orthogonal projection operator onto the span of the matrices vanishing outside of Ω [14] so that

$$(\mathcal{P}_\Omega(x))_i = \begin{cases} x_i & \text{if } i \in \Omega \\ 0 & \text{if } i \in \Omega^c \end{cases} \tag{1}$$

The goal of compressed sensing is to recover elements in Ω^c given the known values in Ω. It assumes that x has a sparse representation in some known basis or dictionary, Ψ, such that $x = \Psi\theta$ where θ has only a few non-zero entries. This sparsity constraint aims to minimise the ℓ_0-norm of θ, which is usually more efficiently solved through the ℓ_1-norm minimisation [8]. Together with the constraints imposed by the measured values, the resulting optimization problem is commonly referred to as the Basis Pursuit Denoising (BPDN) problem.

$$\min_\theta ||\mathcal{P}_\Omega(\Psi\theta - x)||_F + \lambda||\theta||_1 \tag{2}$$

where λ is some regularisation parameter.

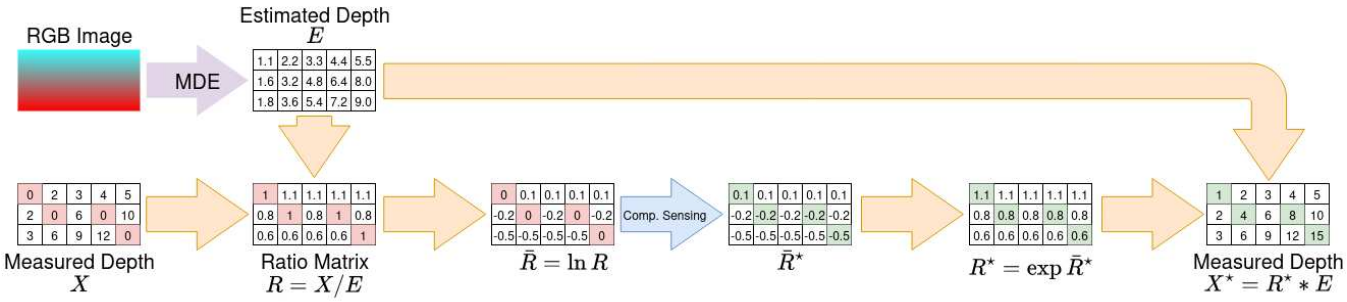

Fig. 2: Schematic diagram of our full rescaling pipeline. Coloured cells represent pixels with missing depth informaton.

Note that when working with 2D signals, such as depth images in our case, the signal is usually better represented as a matrix $X \in \mathbb{R}^{m \times n}$. This does not affect the structure of the optimization problem, as X and Θ can be flattened into vectors during computation.

B. Our approach

Given some RGB-D image, let $X \in \mathbb{R}^{m \times n}$ represent the depth image. The indices of its measured depth pixels constitute the set $\Omega \subset \{1, \dots m\} \times \{1, \dots, n\}$. The corresponding RGB image of X is first used to generate an estimated depth map, $E \in \mathbb{R}^{m \times n}$ using some arbitrary MDE model. A ratio matrix $R \in \mathbb{R}^{m \times n}$ is then calculated via the element-wise operation $R_{ij} = X_{ij}/E_{ij}$. This represents the per-pixel scaling factor that scales the estimated depth to the real depth. Each R_{ij} falls within the domain $[0, \infty)$ and for pixels with missing depth entry, we set $R_{ij} = 1$ for all $(i, j) \in \Omega^c$, effectively assuming that the estimated depth is accurate.

To estimate the scaling ratio for missing pixels in Ω^c, we first take the natural log of R to generate a new matrix $\bar{R}$ such that $\bar{R}_{ij} = \log(R_{ij})$, re-mapping the domain of each scaling factor to $(-\infty, \infty)$. This zeros the mean of the signal, helping to preserve higher-frequency features in the recovered signal after compressed sensing. Using the known ratios in Ω, we can solve the BPDN problem in Equation 2 to find the optimal $\bar{R}^\star$. Taking its natural exponent, we can recover the optimal ratio matrix $R^\star$, which is then multiplied element-wise against E to give the final recovered image $X^\star$. Fig 2 gives a schematic overview of the full rescaling pipeline.

We chose the weights of the 2D Discrete Cosine Transform (DCT) as our basis, given its proven versatility in JPEG compression [26]. The forward 2D DCT converts the log ratio matrix $\bar{R} \in \mathbb{R}^{m \times n}$ to its component bases $\Theta \in \mathbb{R}^{m \times n}$ via the following equation.

$$\Theta_{ij} = \frac{2}{c_i c_j \sqrt{nm}} \sum_{k=1}^{m} \sum_{l=1}^{n} \cos \frac{\pi i (2k+1)}{2m} \cos \frac{\pi j (2l+1)}{2n} \bar{R}_{kl} \tag{3}$$

where

$$c_u = \begin{cases} \sqrt{2}, & \text{if } u = 0 \\ 1, & \text{otherwise} \end{cases}$$

Conversely, the inverse DCT allows us to recover $\bar{R}$ from Θ.

$$\bar{R}_{ij} = \sum_{k=1}^{m} \sum_{l=1}^{n} \frac{2}{c_k c_l \sqrt{nm}} \cos \frac{\pi k (2i+1)}{2m} \cos \frac{\pi l (2j+1)}{2n} \Theta_{kl} \tag{4}$$

To solve the BPDN problem, we used the solver written by Robert Taylor and Thomas Calmart[27]. This is a Python wrapper for the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm implemented by Naoaki Okazaki[28].

Our approach combines the strengths of traditional model-based methods with modern learning-based strategies. Unless the camera is operating under adverse lighting conditions, the RGB image is almost always more complete and detailed than its matching depth image. Therefore, the MDE model can generate an initial depth estimates for the large areas with missing depth information, which are difficult to recover through compressed sensing. Using the physical measurements available, compressed sensing can ground the MDE predictions, enhancing global consistency and accuracy across the entire depth map. Applying compressed sensing on the ratio matrix rather than the depth image directly is also a more practical choice. In principle, an ideal MDE model outputs a metric depth image that is correct up to some scale factor. Therefore, the resulting ratio matrix is expected to be simpler and less feature-dense than the raw depth image, making it more amenable to reconstruction via compressed sensing.

Compared to end-to-end depth completion models, a key strength of our pipeline is the clear separation between the MDE and numeric optimisation steps. This modularity enables flexible selection of any MDE model based on specific engineering requirements, without the need for any additional model fine-tuning. Users can therefore integrate any existing MDE model based on the resource constraints or operational domain of their robotics platform.

IV. Experiments

A. Rescaling MDE outputs

To demonstrate the modularity of our pipeline, we integrated it with two different families of MDE models: DepthAnythingV2 [22] and Metric3DV2 [23]. We used

the NYUv2 dataset [1] to benchmark the accuracy of our rescaled depth predictions. Each RGB image was cropped at the edges to produce an input of size 608×456 pixels, a common approach in other depth completion works to remove invalid data near the image border [24]. We used a sampling ratio of $r = 0.5$, where r denotes the fraction of the depth image assumed to be available and sampled at random. These experiments were performed on an Nvidia RTX 3050.

To quantify the speed of our pipeline at different values of λ, our experiments were repeated using 3 different regularisation parameters, $\lambda = 5 \times |\bar{R}|_F, 0.5 \times |\bar{R}|_F, 0.05 \times |\bar{R}|_F$, where $|\bar{R}|_F$ is the Frobenius norm of the log ratio matrix. The final rescaled image was evaluated against ground truth using three standard metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Threshold Ratio δ_1, defined as the ratio of pixels that satisfies $\max\left(\frac{\hat{d}}{d}, \frac{d}{\hat{d}}\right) < 1.25$, where d is the ground truth and $\hat{d}$ is the predicted depth. We also calculated the metrics for the initial MDE estimate and a naive proportional rescaling approach, which uniformly scales the estimated depth by a common factor that minimises the MAE over the available depth measurements.

Table I tabulates the results of our experiments. We first note that the initial MDE estimate demonstrates limited metric accuracy, since even the better performing DepthAnythingV2 model family exhibited average errors of $40.99 - 42.01$ cm RMSE and $22.74 - 23.07$ cm MAE per frame. Using proportional rescaling, this error can be reduced by a factor of approximately $1.65 - 2.31\times$. However, the final error remains relatively high, with $22.18 - 25.06$ cm RMSE and $9.97 - 12.20$ MAE cm per frame. In comparison, the rescaled depth image from our compressed sensing pipeline exhibits significant improvements in metric accuracy. These are most pronounced at the smallest λ, where both RMSE and MAE improve by a factor of $14.9 - 28.6\times$ to $2.56 - 2.87$ cm and $0.97 - 1.09$ cm respectively. This is expected since a smaller λ helps relax the sparsity constraint on the signal, reducing the error of the reconstructed ratio matrix $R^\star$, and therefore a more accurate final rescaled depth. However, this improvement comes at the cost of an increased computational overhead, as the optimisation solver introduces substantial latency as λ decreases. Looking at the smallest and fastest DepthAnythingV2-Small model, while the model itself ran at 5.95 frames per second, the full pipeline is $16\times$ slower, running only at 0.371 frames per second. In the absence of a faster GPU, a compromise is to use a larger value of λ. Although the improvement in accuracy is more modest at $4.6\times$, the pipeline can now run at more than 1 FPS, making it more practical in scenarios where speed is more critical than accuracy. Fig 4 illustrates a few examples of the final rescaled images at different values of λ.

We also observed that within each family, models built on smaller Visual Transformer (ViT) encoders tend to achieve better metrics. We hypothesize that smaller encoders tend to produce less detailed depth estimates, ignoring the lower-level features of the scene. Consequently, the corresponding $\bar{R}$ derived from these depth estimates is also more uniform, making them more amenable to recovery via compressed sensing. This effect is evident when examining the average magnitudes of the bases of $\bar{R}$ in the NYUv2 dataset, as shown in Fig 3. When ranked in descending order, the magnitude of bases calculated using the small ViT encoder tends to roll off faster than those of the large ViT encoder, allowing the optimal $\bar{R}^\star$ to be more accurately recovered by compressed sensing. This can be advantageous for robotics applications, as models with smaller ViT encoders are more efficient in terms of both GPU memory requirement and run-time, helping to reduce power consumption and latency delays in robotics platforms. Conversely, this also means that there is a fundamental limit to the accuracy achievable through compressed sensing rescaling, rendering it unsuitable for more precise depth estimation tasks.

Regularisation Parameter	RMSE (↓)	MAE (↓)	δ_1(↑)	Freq (fps)
DepthAnythingV2-Small				
Initial MDE Estimate	0.4127	0.2274	0.8847	5.945
Proportional Rescaling	0.2506	0.1220	0.9529	5.737
$\lambda = 5 \times \|\bar{R}\|_F$	0.0904	0.0334	0.9963	1.811
$\lambda = 0.5 \times \|\bar{R}\|_F$	0.0383	0.0128	0.9996	1.143
$\lambda = 0.05 \times \|\bar{R}\|_F$	0.0256	0.0097	0.9999	0.371
DepthAnythingV2-Base				
Initial MDE Estimate	0.4099	0.2283	0.8901	2.468
Proportional Rescaling	0.2298	0.1068	0.9636	2.449
$\lambda = 5 \times \|\bar{R}\|_F$	0.0918	0.0327	0.9959	1.313
$\lambda = 0.5 \times \|\bar{R}\|_F$	0.0398	0.0132	0.9995	0.976
$\lambda = 0.05 \times \|\bar{R}\|_F$	0.0276	0.0105	0.9999	0.367
DepthAnythingV2-Large				
Initial MDE Estimate	0.4201	0.2307	0.8958	0.820
Proportional Rescaling	0.2218	0.0997	0.9662	0.811
$\lambda = 5 \times \|\bar{R}\|_F$	0.0931	0.0324	0.9956	0.588
$\lambda = 0.5 \times \|\bar{R}\|_F$	0.0407	0.0134	0.9995	0.549
$\lambda = 0.05 \times \|\bar{R}\|_F$	0.0287	0.0109	0.9998	0.274
Metric3DV2-Small				
Initial MDE Estimate	0.8493	0.5621	0.5826	5.737
Proportional Rescaling	0.2394	0.1216	0.9564	5.274
$\lambda = 5 \times \|\bar{R}\|_F$	0.1190	0.0459	0.9940	1.321
$\lambda = 0.5 \times \|\bar{R}\|_F$	0.0529	0.0178	0.9991	1.371
$\lambda = 0.05 \times \|\bar{R}\|_F$	0.0248	0.0085	0.9999	0.746
Metric3DV2-Large				
Initial MDE Estimate	1.6800	0.6569	0.5355	0.853
Proportional Rescaling	0.4503	0.2343	0.9021	0.838
$\lambda = 5 \times \|\bar{R}\|_F$	0.2403	0.0501	0.9930	0.664
$\lambda = 0.5 \times \|\bar{R}\|_F$	0.0643	0.0193	0.9986	0.606
$\lambda = 0.05 \times \|\bar{R}\|_F$	0.0279	0.0092	0.9999	0.438

TABLE I: Results of our experiments on the NYUv2 dataset using different MDE models and λ. Full-res 608×456 pixels image is provided to each MDE model at $r = 0.50$ depth sampling ratio.

B. Depth Prompting

To evaluate the efficacy of our rescaling pipeline against fully learning-based depth completion models, we decided to benchmark our approach against Depth Prompting [24], a state-of-the-art depth completion model built on the DepthFormer backbone [4]. Although novel in its use of prompt engineering to generalise depth predictions across various sensor types and scene configurations, the model requires domain-specific fine-tuning and only accepts RGB images at fixed resolution. For indoor environments, the model was fine-tuned using the NYUv2 dataset, downsampled to 304×228 pixels [24]. Additionally, the model demands substantial GPU resources at runtime, making it ill-suited

Sampling Ratio, r	0.90			0.80			0.50			0.20			0.10			0.05		
	RMSE (↓)	MAE (↓)	δ_1(↑)	RMSE (↓)	MAE (↓)	δ_1(↑)	RMSE (↓)	MAE (↓)	δ_1(↑)	RMSE (↓)	MAE (↓)	δ_1(↑)	RMSE (↓)	MAE (↓)	δ_1(↑)	RMSE (↓)	MAE (↓)	δ_1(↑)
304 × 228 Pixels																		
Depth Prompting	0.0304	0.0026	0.9996	0.0324	0.0044	0.9996	0.0345	0.0089	0.9995	**0.0409**	**0.0135**	**0.9994**	**0.0485**	**0.0169**	**0.9991**	**0.0577**	**0.0209**	**0.9986**
DepthFormer + CS	**0.0058**	**0.0010**	**1.0000**	**0.0099**	**0.0024**	**1.0000**	0.0267	0.0109	**1.0000**	0.0671	0.0361	0.9991	0.1024	0.0594	0.9970	0.1407	0.0851	0.9930
DepthAnythingV2-Small + CS	0.0095	0.0016	1.0000	0.0154	0.0038	1.0000	0.0361	0.0146	0.9997	0.0782	0.0414	0.9980	0.1126	0.0640	0.9952	0.1496	0.0883	0.9905
DepthAnythingV2-Base + CS	0.0115	0.0020	1.0000	0.0185	0.0046	0.9999	0.0422	0.0171	0.9994	0.0869	0.0457	0.9970	0.1211	0.0680	0.9936	0.1563	0.0906	0.9888
DepthAnythingV2-Large + CS	0.0126	0.0022	1.0000	0.0202	0.0050	0.9999	0.0454	0.0183	0.9993	0.0919	0.0478	0.9965	0.1268	0.0701	0.9928	0.1624	0.0923	0.9876
Metric3DV2-Small + CS	0.0097	0.0016	1.0000	0.0150	0.0034	1.0000	0.0326	0.0120	0.9999	0.0729	0.0360	0.9986	0.1102	0.0600	0.9956	0.1518	0.0876	0.9903
Metric3DV2-Large + CS	0.0084	0.0013	1.0000	0.0130	0.0029	1.0000	0.0295	0.0109	0.9999	0.0699	0.0352	0.9989	0.1104	0.0618	0.9959	0.1598	0.0958	0.9892
608 × 456 Pixels																		
DepthAnythingV2-Small + CS	0.0075	0.0012	1.0000	0.0116	0.0027	1.0000	0.0256	0.0097	0.9999	0.0544	0.0272	0.9993	0.0795	0.0430	0.9979	0.1072	0.0604	0.9956
DepthAnythingV2-Base + CS	0.0079	0.0013	1.0000	0.0123	0.0029	1.0000	0.0276	0.0105	0.9999	0.0591	0.0296	0.9989	0.0855	0.0461	0.9972	0.1136	0.0634	0.9944
DepthAnythingV2-Large + CS	0.0081	0.0013	1.0000	0.0128	0.0030	1.0000	0.0287	0.0109	0.9998	0.0615	0.0308	0.9988	0.0887	0.0476	0.9968	0.1172	0.0648	0.9938
Metric3DV2-Small + CS	0.0078	0.0011	1.0000	0.0118	0.0025	1.0000	**0.0248**	**0.0085**	0.9999	0.0521	0.0242	0.9993	0.0770	0.0392	0.9980	0.1049	0.0561	0.9957
Metric3DV2-Large + CS	0.0084	0.0012	1.0000	0.0129	0.0026	1.0000	0.0279	0.0092	0.9999	0.0637	0.0276	0.9988	0.1039	0.0465	0.9968	0.1699	0.0691	0.9933

TABLE II: Performance metric on the NYUv2 dataset using Depth Prompting as well as different monocular depth estimation backbone with compressed sensing (CS). We set the regularisaiton parameter $\lambda = 0.05 \times |\bar{R}|_F$. The best performing method for each metric is highlighted in **bold**.

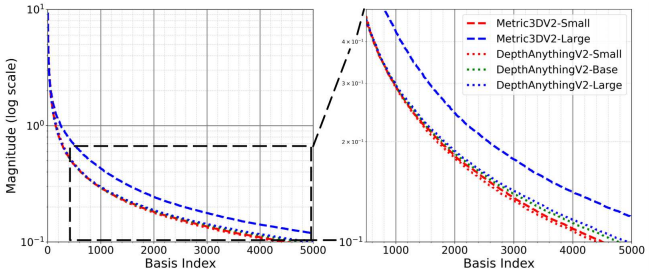

Fig. 3: Average magnitude of the bases of $\bar{R}$ in the NYUv2 dataset, sorted in descending order. This was calculated using ground truth depth and the MDE estimate from each scene. Note how the average magnitude decays more quickly for models with Small ViT encoders (red), making them more readily recoverable by compressed sensing.

for many computationally constrained robotics platforms. For our experiments, the 4 GB of VRAM on the RTX 3050 was insufficient to accommodate all parameters, so we had to move our trial to an NVIDIA Titan XP instead.

Using the initial depth estimate from the DepthFormer backbone, we attempted to recover the ground truth depth using our proposed pipeline at six different sampling ratios, $r = 0.90, 0.80, 0.50, 0.20, 0.10, 0.05$. Table II tabulates the results of our experiments on the NYUv2 dataset [1]. As expected, the Depth Prompting model significantly outperforms our compressed sensing approach as the sampling ratio decreases. This is because the learning-based method is designed to perform under extreme conditions with only a limited number of input points. In contrast, our compressed sensing approach struggles to identify the dominant basis of the underlying signal and is thus unable to accurately reconstruct the structure of the scene. However, at higher sampling ratios, the performance of Depth Prompting deteriorates. As shown in Fig. 5, our compressed sensing pipeline begins to outperform the Depth Prompting model at sampling ratios between 20 and 50 percent for each of the three metrics. We suspect that the prompting engineering strategy to depth completion is unable to maintain physical consistency in the reconstructed scenes, an issue that becomes more pronounced as the number of input samples grows. In contrast, compressed sensing benefits from the information found in additional measurements, which generates a more accurate ratio matrix $R^\star$. This improvement is notable given that our approach uses fewer model parameters and does not require additional tuning. Fig. 6 illustrates a scene from the NYUv2 dataset and visualises how the reconstruction error from both the Depth Prompting model and our compressed sensing pipeline varies at different sampling ratios.

C. Limitations

In strict theoretical terms, for a signal to be recoverable by compressed sensing, its samples must satisfy the Restricted Isometry Property. Usually, in most practical applications, this condition is satisfied with high probability through random sampling. However, since depth readings from stereo cameras are produced via feature matching, the available depth pixels are not truly random. Typically, most features are distributed near the physical edges and corners of the scene, or along surfaces closer to the camera. In outdoor scenes under strong illumination from sunlight, there is a further bias toward shaded regions where the emitted IR pattern remains unsaturated and thus can be more reliably detected. As a result, sparse depth measurements are unevenly distributed throughout the image, often leaving large regions with little or no data. Fig 7 illustrates a few examples of such scenes, note how the depth measurements tend to cluster along floor patterns. In these cases, it becomes difficult to correctly recover the scale in these missing regions. This non-uniform depth distribution biases the estimation of the ratio matrix towards higher-frequency basis functions, resulting in the appearance of artifacts in the final rescaled depth. Thus, for practical applications in outdoor environments, it may be necessary to sample the depth measurements using a secondary depth sensor.

V. Conclusion

In summary, we have shown that compressed sensing can serve as an effective strategy for rescaling MDE estimates to better match ground truth depth. Unlike fully learning-based depth completion models that introduce additional model layers that require fine-tuning, our algorithmic method is entirely training-free, making it a practical alternative on GPU-constrained robotic platforms operating in previously

Fig. 4: Depth maps from selected NYUv2 scenes, comparing ground truth, initial MDE estimate and the rescaled depth using our compressed sensing pipeline at different values of λ

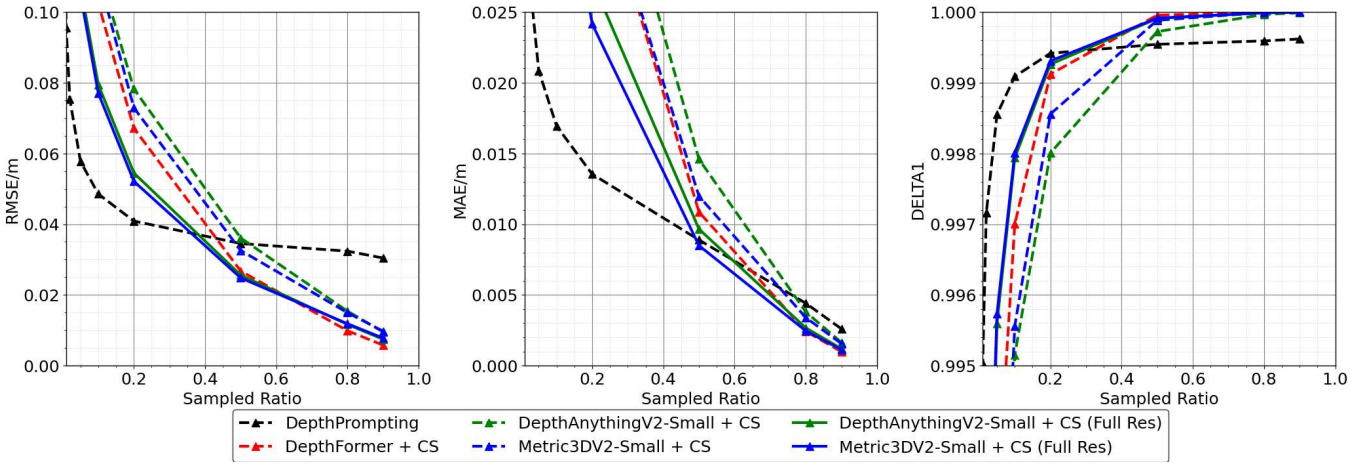

Fig. 5: Graphs of performance metrics against sampled ratio for Depth Prompting model and our compressed sensing pipeline using different MDE backbones

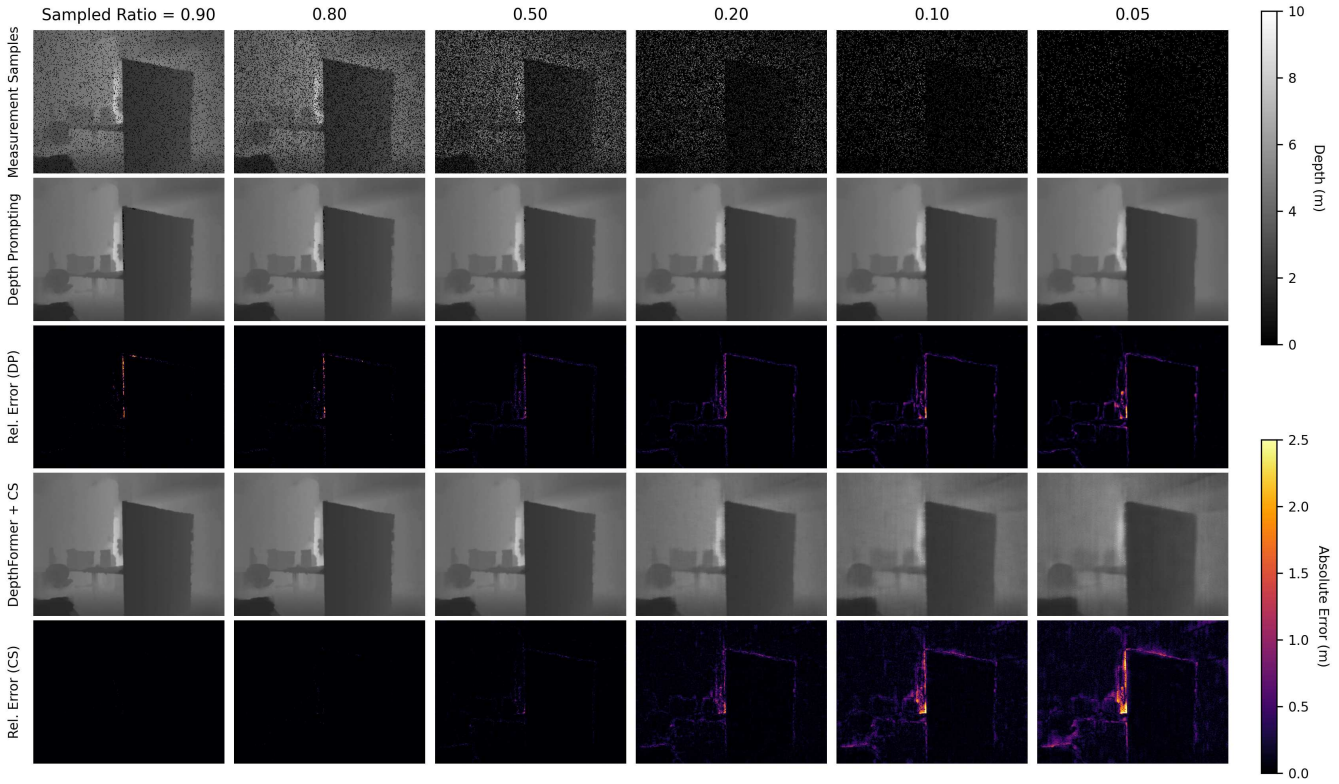

Fig. 6: Images in the top row illustrates the depth map of sampled points using for various sampling ratios. Images in subsequent rows illustrates the final estimated depth and the relative error with respect to ground truth. Note the errors present in the final estimate generated by the Depth Prompting model at higher sampling ratios. This error disappears in our compressed sensing (CS) approach.

unseen environments. Moreover, the compressed sensing step is modular and can be easily integrated with any arbitrary MDE backbone. Experimental results show that rescaling through compressed sensing can improve the metric accuracy of the initial depth estimate by a factor of 15. At higher sampling ratios above 50 percent, our approach can even outperform state-of-the-art depth completion methods, producing a more accurate final depth map.

However, there remain some limitations to our pipeline. There exists a performance ceiling to the level of detail recoverable by compressed sensing, as the high-frequency details tend to be filtered out by the method as noise. Furthermore, the current use of DCT bases combined with the L-BFGS solver is relatively slow, limiting the utility of the method for real-time applications. Although runtime can be reduced slightly by increasing the λ regularisation parameter, it comes at a notable cost to the final rescaled depth accuracy. We also observe that the pipeline's performance degrades in outdoor environments, where the distribution of available depth samples is inherently ill-suited for compressed sensing. This limitation may necessitate the use of auxiliary depth sensors to ensure reliable performance in such settings.

Moving forward, our goal is to improve both the speed and accuracy of our pipeline. Specifically, we plan to explore alternative basis representations for the ratio matrix, such as the Discrete Daubechies Wavelet Transform, which has been shown in previous work to be an effective basis for image reconstruction at lower sampling ratios [15]. We also want to test whether the minimisation of other cost functions, such as the signal's Total Variation, can help remove some of the artifacts we observe [10]. We hope that these enhancements can further improve the accuracy of the rescaled image while preserving the core advantages of our pipeline.

References

[1] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[2] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 567–576, 2015.

[3] A. Bhoi, "Monocular depth estimation: A survey," 2019.

[4] Z. Li, Z. Chen, X. Liu, and J. Jiang, "Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation," *Machine Intelligence Research*, p. 837–854, Sep. 2023.

[5] A. Lopes, R. Souza, and H. Pedrini, "A survey on rgb-d datasets," *Computer Vision and Image Understanding*, vol. 222, p. 103489, 2022.

[6] Y. Zhang and T. Funkhouser, "Deep depth completion of a single rgb-d image," 2018.

[7] M. A. U. Khan, D. Nazir, A. Pagani, H. Mokayed, M. Liwicki, D. Stricker, and M. Z. Afzal, "A comprehensive survey of depth completion approaches," *Sensors*, vol. 22, no. 18, p. 6969, Sep. 2022.

[8] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, p. 1207–1223, Mar. 2006.

[9] A. Majumdar and R. K. Ward, "Compressed sensing of color images," *Signal Processing*, vol. 90, no. 12, pp. 3122–3127, 2010.

[10] F. Shi, J. Cheng, L. Wang, P.-T. Yap, and D. Shen, *Low-Rank Total Variation for Image Super-Resolution*. Springer Berlin Heidelberg, 2013, p. 155–162.

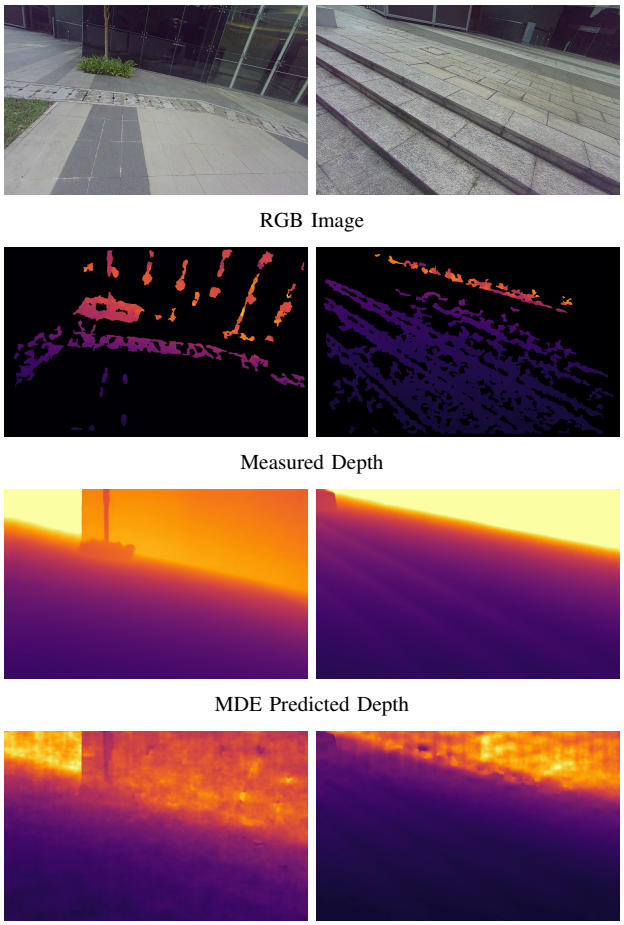

Fig. 7: Results of our compressed sensing pipeline on outdoor scenes captured using an Orbbec Gemini 2L [29]

[11] S. Brunton and J. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[12] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[13] E. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" 2004.

[14] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, p. 2117–2130, Sep. 2013.

[15] N. Dwork, D. O'Connor, C. A. Baron, E. M. I. Johnson, A. B. Kerr, J. M. Pauly, and P. E. Z. Larson, "Utilizing the wavelet transform's structure in compressed sensing," *Signal, Image and Video Processing*, vol. 15, no. 7, p. 1407–1414, Mar. 2021.

[16] H. Xue, S. Zhang, and D. Cai, "Depth image inpainting: Improving low rank matrix completion with low gradient regularization," *IEEE Transactions on Image Processing*, vol. 26, no. 9, p. 4311–4320, Sep. 2017.

[17] R. M. H. Nguyen and M. S. Brown, "Fast and effective l0 gradient minimization by region fusion," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Dec. 2015, p. 208–216.

[18] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," 2014.

[19] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," 2017.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020.

[21] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," 2021.

[22] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *arXiv:2406.09414*, 2024.

[23] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen, "Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[24] J.-H. Park, C. Jeong, J. Lee, and H.-G. Jeon, "Depth prompting for sensor-agnostic depth estimation," 2024.

[25] Y. Zhang, X. Guo, M. Poggi, Z. Zhu, G. Huang, and S. Mattoccia, "Completionformer: Depth completion with convolutions and vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 527–18 536.

[26] International Telecommunication Union and Joint Photographic Experts Group, *Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines (Recommendation ITU-T T.81 / ISO/IEC 10918-1)*, ITU and ISO/IEC Std. T.81 / 10 918-1, Sep. 1992.

[27] R. Taylor and T. Calmant, "Pylbfgs," https://github.com/dedupeio/pylbfgs, 2019.

[28] N. Okazaki, "libLBFGS: A C port of the L-BFGS optimization algorithm," https://chokkan.org/software/liblbfgs/, 2014.

[29] Orbbec, "Gemini 2l specification," https://www.orbbec.com/products/stereo-vision-camera/gemini-2l/, 2025.