Next-Best Stereo: Extending Next-Best View Optimisation For Collaborative Sensors

Oscar Mendez[1]
O.Mendez@surrey.ac.uk
Simon Hadfield[1]
S.Hadfield@surrey.ac.uk
Nicolas Pugeault[2]
N.Pugeault@exeter.ac.uk
Richard Bowden[1]
R.Bowden@surrey.ac.uk

[1] Centre for Vision Speech and Signal Processing
University of Surrey Guildford, UK

[2] Department of Computer Science
University of Exeter
Exeter, UK

Abstract

Most 3D reconstruction approaches passively optimise over all data, exhaustively matching pairs, rather than actively selecting data to process. This is costly both in terms of time and computer resources, and quickly becomes intractable for large datasets.

This work proposes an approach to intelligently filter large amounts of data for 3D reconstructions of unknown scenes using monocular cameras. Our contributions are two-fold: First, we present a novel approach to efficiently optimise the Next-Best View (NBV) in terms of accuracy and coverage using partial scene geometry. Second, we extend this to intelligently selecting stereo pairs by jointly optimising the baseline and vergence to find the NBV's best stereo pair to perform reconstruction. Both contributions are extremely efficient, taking 0.8ms and 0.3ms per pose, respectively.

Experimental evaluation shows that the proposed method allows efficient selection of stereo pairs for reconstruction, such that a dense model can be obtained with only a small number of images. Once a complete model has been obtained, the remaining computational budget is used to intelligently refine areas of uncertainty, achieving results comparable to state-of-the-art batch approaches on the Middlebury dataset, using as little as 3.8% of the views.

1 Introduction

The 3D reconstruction of scenes and objects from 2D images is an extremely important part of many tasks, such as robot navigation, scene understanding and surveying. Approaches based upon Bundle Adjustment (BA) have become popular within the literature with extremely impressive results that allow dense and high fidelity models to be reconstructed from unordered image collections[1]. However, such approaches are computationally expensive, which limits their application. For example, a mobile robot with a single camera can easily overwhelm systems that attempt exhaustive optimisations over all images. Furthermore, reconstruction needs to be online (not batch) as navigation may well be dependent upon reconstruction. Therefore,

online reconstruction algorithms that are capable of selecting data that maximises performance, while reducing computational time are necessary to perform reconstruction in the real world.

In this paper, we propose an autonomous 3D reconstruction framework that is capable of creating dense, accurate maps using a fraction of the available images. This is done by intelligently and efficiently selecting images to act as variable-baseline stereo pairs. In section 3.1 we present a method of reconstructing a scene using a robust dense matching algorithm based on Deep Learning[24]. In section 3.3 we present a novel method for estimating the Next-Best View (NBV) to observe the scene from, based on the current reconstruction of the geometry. Section 3.4 extends this by selecting a stereo pair for the NBV, the Next-Best Stereo (NBS), that will produce the best reconstruction. Finally, in section 4 we evaluate our system using the Middlebury dataset.

2 Related Work

Fully autonomous reconstruction of unknown scenes and objects is a challenging problem, especially for a monocular camera. While there exist numerous online approaches such as Parallel Tracking and Mapping (PTAM)[16], that uses sparse features, and LSD-SLAM[9], that uses semi-dense depth maps filtered by gradient, the most accurate systems tend to use offline optimisation frameworks.

2.1 Multi-View Stereo (MVS)

Offline approaches, commonly referred to as MVS, typically find pairwise stereo correspondences and use large optimisations to estimate dense and accurate reconstructions, such as work by Snavely *et al*. [23]. Denser reconstructions were achieved by Furukawa and Ponce [6] who use sparse feature matching and patch growing, along with photometric and visibility constraints to produce dense reconstructions. Jancosek *et al*. [14] extend [6] by attempting to actively select views in a NBV-like approach to make large datasets feasible by estimating feasible stereo pairs, but provide no results on partial-image reconstruction. Hornung *et al*. [11] use an octree-like hierarchical volumetric reconstruction along with graph cut minimisation. More recently, Galliani *et al*. [7] expand the patch-matching idea by [2] to use more than two views. Seminal work by Seitz *et al*. [22] established the Middlebury benchmark to compare MVS approaches by providing a calibrated dataset of camera poses and ground truth.

However, the computational cost for dense reconstruction of large structures can be prohibitive, preventing their use online, and lack the ability to chose views dynamically during data capture. In this work we propose a novel approach capable of actively choosing the best locations to improve the reconstruction/model or map. More importantly, it is capable of significantly reducing computational cost by selecting a small number of key views to use.

2.2 Next-Best View (NBV) Estimation

NBV estimates a new pose in order to improve the existing reconstruction, and can be divided into two main categories: *exploration* and *refinement*. Exploratory NBV estimation aims at generating the most complete model of the (unknown) scene. It is generally based on the concept of a *frontier*, for example in the work by Heng *et al*. [9]. Heng uses a precomputed lattice and defines frontier locations as edges between cells where structure has been observed,

and unobserved cells. Frontier pose configurations are then selected based on the information gain they provide and the cost to reach that configuration. Paull *et al*. [19] similarly uses coverage and distance to the goal. Potthast and Sukhatme [20] use a raycasting method that estimates potential information gain for a pose. These systems rely on depth sensors to perform the reconstruction and thus make no attempt to reduce the noise in the scene.

In contrast, refinement NBV estimation aims at selecting poses that improve the 3D model accuracy. For example, Forster *et al*. [6] use depth uncertainty to estimate the best areas of the map to explore but are limited to relatively simple scenes. Hoppe *et al*. [10] create a full network of poses for an Unmanned Aerial Vehicle (UAV), but assumes prior knowledge of the environment. Sadat *et al*. [21] and Mostegel *et al*. [18] plan optimal paths for a monocular Visual Odometry (VO) system, but require a set endpoint. Uniquely, we propose a unified approach that can balance the two competing objectives of exploration and refinement by probing the current estimate of geometry using raycasting and a voxel based representation.

The approaches most similar to ours are Dunn and Frahm [3], Mauro *et al*. [17] and Hornung *et al*. [12]. Dunn and Frahm use a similar raycasting and eigenvalue technique, but require a partial input 3D model, and perform an offline batch optimisation method that must be converted into a mesh at every iteration. Mauro *et al*. [17] define the NBV as the camera that maximises a view importance metric of aggregate quality features (Density, Uncertainty and Saliency), but require an offline reconstruction algorithm. Finally, Hornung *et al*. [12] build a similar voxel-based proxy model and selects the views based on maximising the number of visible low quality voxels. They then refine regions with bad photo-consistency by adding more views of these areas. However, both [17] and [12] require expensive pointcloud reprojection that scales with the size of the scene. Our work proposes a novel unified formulation that is capable of prioritising both exploration and refinement to perform online reconstruction, while using sparse, fast, calculations for NBV and NBS that scale well with map size.

3 Method

In this section we describe our method for creating a dense, accurate reconstruction of a scene using the smallest number of views possible. Our incremental approach to reconstruction relies on using the partial reconstruction at iteration i to intelligently select new images to be added to the reconstruction at iteration $i+1$. We show how a 3D reconstruction is triangulated from two images using a robust dense matching algorithm, and how a map is then created by discretising this dense pointcloud into an octree structure (Section 3.1). We then describe how the reconstructed geometry is used in our novel method to estimate the NBVs for a monocular camera by finding the pose that minimises error in the existing map while also extending it into unexplored areas (Section 3.3). We then discuss how a NBS pair can be estimated, based on the geometry and sensor arrangement, to provide the most benefit to reconstruction (Section 3.4).

3.1 Reconstruction of Dense 3D Structure

Before we describe our NBV and NBS formulation, we describe our approach to MVS. We generate a high-resolution map of the scene by reconstruction from two images in a stereo arrangement. This is achieved in three steps: dense matching, triangulation and temporal accumulation. In the first step, a dense pixel match between the two images is estimated using a deep learning-based approach [24]. This dense matching is done bidirectionally, allowing us to discard inconsistent correspondences and improve robustness and accuracy. In the second step, a dense 3D point cloud Z containing points $z \in Z$ is reconstructed from classical 3D reconstruction equations (see, eg, [8]). In addition, the uncertainty of each reconstructed point z is also estimated from the triangulation error, and stored as a 3×3 covariance matrix $\Lambda_z =$

$B\bar{\Lambda}B^{\top}$. Where B is the Jacobian of the projection function, and $\bar{\Lambda}$ are the uncertainties for each point in the cloud. The matching used in this paper does not provide uncertainties, therefore, matching errors are assumed to be normally distributed ($\bar{\Lambda} = I$). These two steps allow the reconstruction of dense 3D point clouds describing the scene structure and uncertainties from the two cameras. The final step, integrates the 3D scene information from subsequent pairs of views using the covariance matrices, to obtain the final estimate of 3D geometry.

3.2 Octree encoding of scene structure

The triangulated 3D points provide a detailed representation of the scene; however, such a large point cloud is very inefficient for the purpose of reasoning about scene geometry and too large for efficient data association. At the same time it forms discrete data embedded in a continuous space, making it too sparse for meaningful geometric calculations (such as ray casting). In order to solve these two issues, the space is further discretised using an octree data structure to which the points are added at leaf nodes.

We use a modified version of OctoMap[13], which uses trees with depth of 16, to keep track of *occupied* (V_o), *empty* (V_e) and *unobserved* (V_u) voxels. We define the set of all voxels in the octomap as $V = (V_o \cup V_e \cup V_u)$. *Occupied* voxels represent areas with reconstructed geometry, *empty* voxels are unoccupied space with no geometry and *unobserved* voxels are areas of the scene with no observations indicating membership of (V_o) or (V_e), as such $V_o \cap V_e \cap V_u = \emptyset$. We also define the leaf nodes in the tree as $v \in V$. Finally, each leaf node v has a set of points stored in it, therefore, we define the set of points in v as P_v and the individual points as $p \in P_v$.

The octree structure is used to allow efficient point matching across different stereo-pairs of poses. A naive exhaustive match of all points from both point clouds using Mahalanobis distance would be prohibitive, therefore we make use of the octree. For every new pair of frames, we obtain a pointcloud Z that must be fused into the octree. Z contains points $z \in Z$ and covariance matrices Λ_z. If we define $p \in P_v$ as an existing point in the voxel with covariance matrix Σ_p, we can estimate the update thusly. First, we query the octree to find the leaf node each point z lands in. If the leaf node is empty, the point and its covariance matrix Λ_z are added. Otherwise, we find the top 3 Euclidean nearest-neighbours $\delta(z, P_v)$. We find the Mahalanobis distance for the these 3 neighbours and if the closest Mahalanobis distance falls within a 95% confidence ($\chi^2 < 7.81$) the new point is used to perform a Kalmann update on the neighbour point and its covariance matrix Σ. We define the Kalman gain K_g as $\mathrm{K}_g = \Sigma_p(\Sigma_p + \Lambda_z)^{-1}$ and the update is performed as follows:

$$\Sigma_p = (I - K_g)\Sigma_p \tag{1}$$

$$p = p + K_g(z - p) \tag{2}$$

3.3 Next-Best View Optimisation

In order to filter out unnecessary information, we need a system that is capable of choosing what the next position of the two sensors will be, in a way that will increase its scene knowledge optimally. In this section we propose a novel criterion for NBV optimisation based on a compromise between the competing objectives of coverage and accuracy. The coverage objective will drive the system to collect views of previously unobserved parts of the scene (*e.g.*, due to restrictions on the field of view or occlusion), whereas the accuracy objective will drive the system to choose the next pose to reduce the point cloud's uncertainty.

These two criteria are optimised jointly, making use of the octree structure and the dense point cloud. The octree allows for quick and efficient calculations on scene geometry, while the dense cloud (and covariances) allow for more detailed calculations about scene noise and and viewing angle.

The NBV is calculated as follows: Given a Configuration Space (CS) of sensor poses, the cost of each pose can be estimated by casting a set S_r of random rays from the camera centre through the image plane. In practice, we use around 500 rays. Each ray will continue until it hits either an occupied (V_o) or unobserved (V_u) voxel, ignoring empty (V_e) voxels. When a ray $r \in S_r$ is incident on an occupied voxel $v \in V_o$, we can estimate a cost for each point $p \in P_v$ as

$$\phi(r,p) = e^{-||\lambda_p e_p \times r||}, \tag{3}$$

where λ_p and e_p are the largest eigenvalue and eigenvector, respectively, of the covariance Σ_p. Consequently, the cost of a voxel is defined as the average point cost

$$\psi(r,v) = \frac{1}{|P_v|} \sum_{p \in P_v} \phi(r,p). \tag{4}$$

Finally, the NBV cost of a particular pose x is defined as

$$C_x = \frac{1}{|S_r|} \sum_{r \in S_r} \begin{cases} \psi(r,v) & \text{if } v \in V_o \\ \gamma \in [0,1] & \text{else } v \in V_u. \end{cases} \tag{5}$$

In this equation, γ is a parameter that can encourage or discourage exploration. A γ of 1 will always give the highest cost to unobserved voxels, prefering to reduce the uncertainty of observed voxels, while 0 will give them the lowest, preferring to get more observations.

Finally, $\arg\min_x(C_x)$ finds the pose that will provide the most benefit to the existing map.

3.4 Next-Best Stereo Optimisation

When there are multiple collaborating sensors available, we can extend NBV to also optimise the stereo arrangement of the sensors. This can be achieved by selecting another view, with respect to the NBV, to create the best possible stereo pair. Therefore, we now demonstrate how the stereo arrangement of sensors can be optimised such that it is advantageous for both dense matching and 3D reconstruction. Actively selecting stereo pairs allows sensors to be positioned to allow an optimal vergence and baseline, respective to the observed parts of the scene.

This implies several requirements: First, the baseline of the cameras must be scaled depending on the distance to the observed geometry. This is necessary since the baseline is proportional to the depth error. Second, the vergence angle should be minimised to allow the dense matching to be performed with the least amount of error possible. Finally, the distance between the vergence point and the nearest geometry should be minimised, to ensure that the sensors are trained on actual scene geometry and not empty space. Therefore, if L and R are the 6-Degrees of Freedom (DoF) poses of the sensors, the rays r_L and r_R are vectors from each camera centre, through the principal point and represent the viewing direction of each camera. The intersection (I) of these two rays can be calculated as a triangulation similar to the one used in section 3.3. Finally, G is the centre of the occupied octree voxel $v \in V_o$ that is closest to the intersection point I.

3.4.1 Baseline (B) Vergence Angle (β) optimisation

Optimising the baseline can be done by enforcing the following constraints:

$$d_{LI} = d_{RI} = \alpha d_B \tag{6}$$

Where d_{LI} and d_{RI} are the distances to I (from left and right cameras respectively). d_B is the baseline and α is the desired ratio between the baseline and the intersection point. These constraints can be enforced by finding the pose that minimises equation 7.

$$C_B = \frac{|d_{LI} - \alpha d_B|}{\alpha d_B} + \frac{|d_{RI} - \alpha d_B|}{\alpha d_B} + \frac{|d_{LI} - d_{RI}|}{d_B} \tag{7}$$

This enforces a triangular structure defined by alpha (α), where the ratio defines the expected angle of vergence, β, which can be shown to be

$$\beta = acos\left(1 - \frac{1}{2\alpha^2}\right). \tag{8}$$

3.4.2 View Triangulation Optimisation

In 3D, the rays r_L and r_R rarely have an exact point of intersection, instead the triangulation finds the point that is closest to both rays. This implies that there will be a distance between the actual principal point and the reprojection of I, called the reprojection error. While it is possible to measure this error, it is a relatively expensive operation. Instead, the cost is calculated as the angle between the rays r_L and r_{LI}

$$C_T = acos\left(\frac{|r_L \cdot r_{LI}|}{\|r_L\|\|r_{LI}\|}\right) + acos\left(\frac{|r_R \cdot r_{RI}|}{\|r_R\|\|r_{RI}\|}\right) \tag{9}$$

3.4.3 Rotational Optimisation

Since we are computing dense, per-pixel, matches we cannot assume that the process is rotationally invariant. In order to increase the performance of any matching algorithm, we penalise large differences in the orientation of the image. That is, we penalise roll. If we assume the gravity vector $v_g = [0,0,1]$, the roll is then penalised as the difference in the angle between the gravity vector transformed into each camera's coordinate frame

$$C_R = acos\left((R_L v_g)^\top R_R v_g\right) \tag{10}$$

where, R_L and R_R are the rotation matrices of each sensor.

3.4.4 Optimising vergence on scene structure

So far, the costs defined will arrange the cameras to perform an accurate stereo triangulation of anything on or near the plane that contains the vector $r_{LI} \times r_{RI}$. However, this has not yet been coupled with the existing geometry, which should be done in order to avoid situations where the sensors have a vergence point that is behind or in front of the geometry. This is done by minimising the angle between the rays from both cameras to intersection I and the closest point of known geometry G

$$C_G = acos\left(\frac{|r_{LI} \cdot r_{LG}|}{\|r_{LI}\|\|r_{LG}\|}\right) + acos\left(\frac{|r_{RI} \cdot r_{RG}|}{\|r_{RI}\|\|r_{RG}\|}\right) \tag{11}$$

Where, G is the nearest occupied voxel and r_{LG} and r_{RG} are rays from the camera centres to this point.

3.4.5 Stereo poses optimisation

The final cost function can then be defined as

$$C = C_B + C_T + C_R + C_G \tag{12}$$

This cost can be efficiently computed for thousands of candidate pairs in the CS, and the optimum configuration can then be selected as the pair that minimises this cost.

4 Evaluation

In this section, we aim to demonstrate that our algorithm can automatically select the best views and sensor configuration in order to provide a dense, accurate and complete pointcloud using a significantly reduced number of observations. We evaluate our approach by applying it to the standard Multi-View Geometry reconstruction dataset from Middlebury [22]. Firstly, we compare our approach against other NBV approaches using the Middlebury benchmark, showing that we can outperform the state-of-the-art using less frames. Secondly, we demonstrate how our two main parameters α and γ encourage different behaviours and can therefore be tailored to different applications. Lastly, we present a qualitative analysis of our approach where we visualise our pointclouds against a reference model computed by a state-of-the-art reconstruction [6] using all views.

The Middlebury dataset consists of 2 figurines, dino and temple, imaged in a dome-like pattern. For each figurine there are 3 modalities of the images: full ($\sim$300 images), ring ($\sim$50 images) and sparse ring ($\sim$15 images). In terms of NBV selection, the full datasets are more challenging because they present more possible stereo pairs ($O(300^2)$). In order to perform a valid evaluation of our approach, it is necessary to measure the performance as the number of used stereo pairs increases. We use the same error metrics as those used in the Middlebury benchmark by Seitz *et al.* [22]. Seitz measures the distance between each point in the reconstructed pointcloud and the reference pointcloud. They then estimate an error distance d such that a certain percentage of the points are within d. The coverage is similarly estimated by measuring distance from the reference to the reconstructed cloud. Several thresholds are selected, and the percentage of points in the reference cloud that contain a neighbour within that distance is reported. We use these metrics to explore our parameter space and see the effects of α and γ. The Middlebury dataset has no publicly available ground truth, therefore, we create a reference model to aid in parameter exploration. This reference model is created from all the images in each dataset using the state-of-the-art MVS reconstruction algorithm from Furukawa and Ponce [6].

4.1 Parameter Exploration

The parameter, α, controls two competing objectives: the baseline and vergence. A low α value enforces a wider baseline, which gives us low depth error. A high α value enforces a more acute vergence angle, allowing better matching. We expect there to be a value that allows a good compromise between both competing objectives. To demonstrate this, we show the results on dino, since the low texture generally makes it more challenging. We first disable the γ parameter, and then let our approach select 40 pairs of frames. We then calculate the average

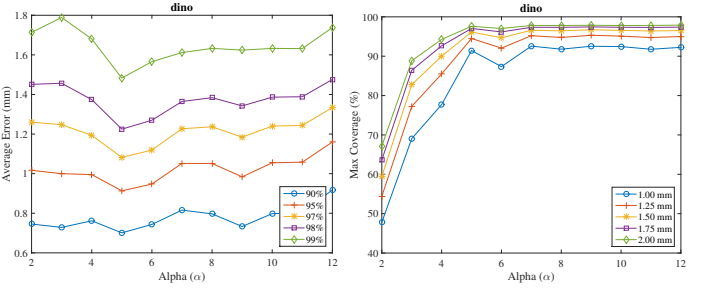


Figure 1: Average Error (Top) and Max Coverage (Bottom) with increasing values of α.

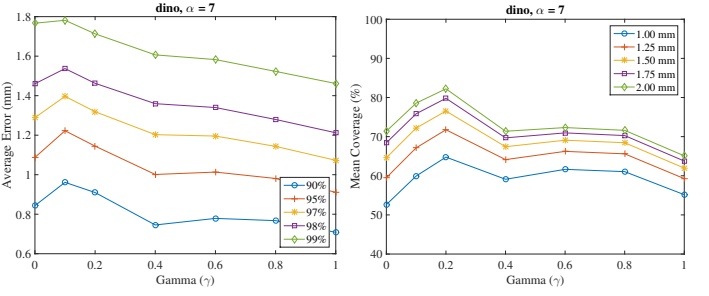


Figure 2: Average Error (Left) and Average Coverage (Right) with different values of γ.

error and the maximum achieved coverage for the entire sequence, with 5 different error thresholds. Figure 1 shows error and coverage curves for different values of α. It can be seen that very low values of α have very low coverage since the wide vergence angles make dense matching difficult. On the other hand, very large values start to suffer from increasing depth error due to the relatively narrow baseline. Choosing values of $\alpha \in [5,7]$, corresponding to a vergence angle of around 9° degrees, achieves high coverage while minimising the average error. It is important to note that these results are *not* dependent on the absolute values of depth, baseline or vergence. Rather, they scale with the scene to provide good stereo configurations. More importantly, α can be tailored to other matching approaches. High accuracy, sparse feature matching can have low values of alpha that allow good depth estimation. Denser, per-pixel methods can use high alphas to encourage easier matching.

In order to explore the effects of γ, we select a value of $\alpha = 7$. This is done because γ only applies to the NBV so we choose a narrow baseline to reduce the effects of mismatching. Furthermore, we are interested in a dense matching approach and the small increase in error is justified by the larger coverage and easier matching.

Gamma (γ) provides the ability to either encourage or discourage exploration. As shown in equation 5, γ controls how favourable it is for the camera to look at unobserved voxels ($v \in V_u$). Setting $\gamma = 0$ assigns the lowest score to unobserved voxels, while $\gamma = 1$ assigns the highest. The error curve in Figure 2 shows the same metrics used previously. The coverage curve shows the average for only the first 5 frames, since otherwise all values of γ converge to high coverage.

It can be seen that as the value of γ goes up, the average error starts to decrease, as our approach prefers refinement over exploration. However, the coverage also decreases as the approach prioritises different views of the same geometry. Note that, despite the general

	Thresholds	Uniform [12]	NBV [12]	NBS [14]	NBS Proposed
Num. Frames	-	41	41	unknown	**26**
Error (mm)	80%	0.64	0.59	0.64	**0.53**
	90%	1.00	0.88	0.91	**0.74**
	99%	2.86	2.08	1.89	**1.68**
Coverage (%)	0.75mm	79.5	82.9	72.9	**87.3**
	1.25mm	90.2	93.0	73.8	**96.4**
	1.75 mm	94.3	96.9	73.9	**98.4**

Table 1: Middlebury Evaluation for different NBV and MVS approaches.

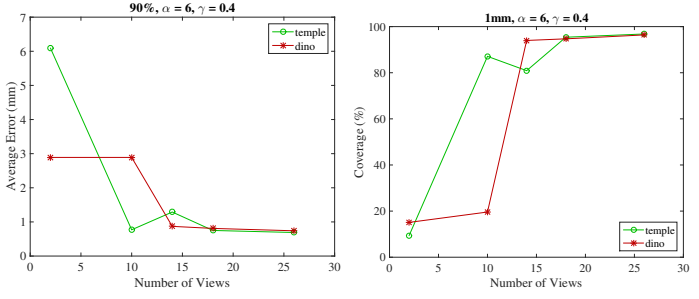


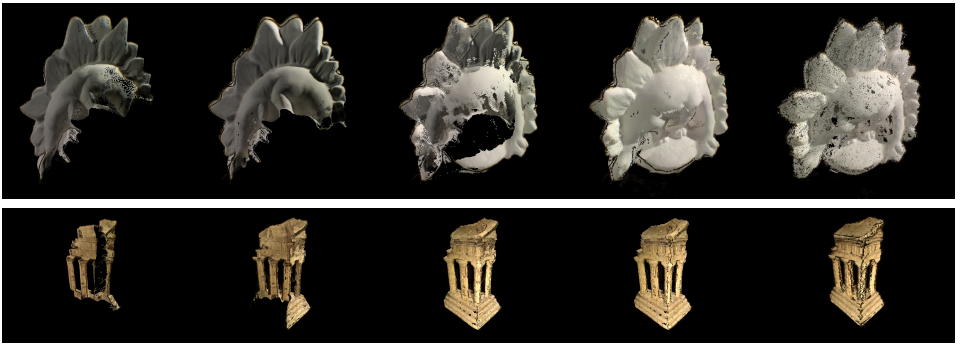
Figure 3: Middlebury Benchmark as number of views go up with $\alpha = 6$ and $\gamma = 0.4$

downward trend, the values of $\gamma < 0.1$ are unstable because the NBV concentrates on areas of the scene where there is no geometry, therefore making the stereo pair selection ill-posed. The same is true with the coverage, where extremely low values of γ encourage looking at the narrowest profiles of the object (since they will include the most unobserved voxels. It is important to notice that all values achieve high levels of coverage. In practice, we use $\gamma = 0.4$, since this allows slight bias towards exploration.

4.2 Quantitative Analysis

Now that we have established good values for α and γ, we can show how we compare against other NBS methods. In order to evaluate against the online ground truth for the Middlebury benchmark, we run our pointclouds through a Poisson Surface Reconstruction[15]. Table 1 shows, from left to right, a comparison against a voxel-based MVS approach with 41 uniformly selected views [12], with their NBV approach and the NBS approach of Jancosek *et al.* [14]. It can be seen that our NBS approach consistently outperforms Hornung *et al.* [12] using both uniform and selected views, while simultaneously using less frames. Furthermore, we outperform the NBS apporach of [14]. Note that we do not compare against the full Middlebury benchmark due to the fact that most approaches are exhaustive MVS optimisations, rather than online NBV or NBS selection approaches, making the comparison unfair.

We also compared our partial results against the ground truth from Middlebury. Figure 3 shows how the error and coverage change as the number of views increases. As expected, we see that our approach can improve coverage whilst simultaneously reducing error. Note that the slight instability at a small number of views is due to problems with the Poisson Reconstruction, not the pointclouds. The pointclouds produced by our approach are clean and accurate.



1 pair 3 pairs 5 pairs 7 pairs Reference (GT)

Figure 4: Results for Middlebury Dino (top) and Temple (bottom) Datasets, with varying numbers of stereo pairs. The final column shows the reference model.

4.3 Qualitative Analysis

Figure 4 shows how as the number of views increases, so does the quality of the reconstructed cloud. By the time our approach has selected 7 views, with their respective stereo pairs, we are capable of producing a pointcloud that is fundamentally complete. Note that both the front and back, of the models have been successfully reconstructed from a maximum of 14 images. This corresponds to 3.8% and 4.4% of the images for dino and temple datasets, respectively. These results are visible in greater detail in the supplementary material.

5 Conclusions

In conclusion, we have proposed an approach that is capable of creating a dense reconstruction of a scene by autonomously selecting images that will provide the largest gain to reconstruction. We have presented a method to reconstruct a dense pointcloud using a joint filtering and discretisation method, and a novel formulation that is capable of actively encouraging or discouraging exploration in the pose selection, using only 0.8ms per pose, and finally established a cost function that allows pose configurations that are beneficial to the sensing framework, using only 0.3ms per pose. We have demonstrated that we are able to achieve state-of-the-art results in our reconstructions, using as little as 3.8% of the views.

6 Acknowledgements

Many thanks to Daniel Scharstein, for maintaining the Middlebury benchmark [22] and comparing our results against ground truth.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54 (10):105–112, 2011. doi: 10.1145/2001269.2001293.

[2] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *British Machine Vision Conference (BMVC)*, 2011.

[3] Jan-Michael Dunn, Enrique Frahm. Next best view planning for active model improvement. In *British Machine Vision Conference (BMVC)*, 2009.

[4] Jakob Engel, Thomas Schöps, Juergen Sturm, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision (ECCV)*, pages 1–16, Zurich, 2014. Springer.

[5] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Appearance-based active, monocular, dense reconstruction for micro aerial vehicles. In *Robotics: Science and Systems Conference*, 2014.

[6] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[7] Silvano Galliani and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *International Conference on Computer Vision (ICCV)*, 2015.

[8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[9] Lionel Heng, Alkis Gotovos, Andreas Krause, and Marc Pollefeys. Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. ISBN 9781479969234.

[10] Christof Hoppe, Andreas Wendel, Stefanie Zollmann, Katrin Pirker, Arnold Irschara, Horst Bischof, Stefan Kluckner, and Siemens Corporate Technology. Photogrammetric camera network design for micro aerial vehicles. In *Computer Vision Winter Workshop*, 2012.

[11] Alexander Hornung and Leif Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2006. ISBN 0769525970.

[12] Alexander Hornung, Boyi Zeng, and Leif Kobbelt. Image selection for improved multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008.

[13] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013. ISSN 09295593. doi: 10.1007/s10514-012-9321-0. URL http://octomap.github.com.

[14] Michal Jancosek, Alexander Shekhovtsov, and Tomas Pajdla. Scalable multi-view stereo. In *International Conference on Computer Vision*, 2009. ISBN 9781424444410.

[15] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013. ISSN 07300301. doi: 10.1145/2487228.2487237. URL http://dl.acm.org/citation.cfm?id=2487237$\delimiter"026E30F$nhttp://dl.acm.org/citation.cfm?doid=2487228.2487237.

[16] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In IEEE and ACM, editors, *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, 2007. doi: 10.1109/ISMAR.2007.4538852.

[17] Massimo Mauro, Hayko Riemenschneider, Alberto Signoroni, Riccardo Leonardi, and Luc Van Gool. A unified framework for content-aware view selection and planning through view importance. In *British Machine Vision Conference (BMVC)*, pages 1–11, 2014. URL http://people.ee.ethz.ch/~rhayko/paper/bmvc2014_mauro_nextbestview.pdf.

[18] Christian Mostegel, Andreas Wendel, and Horst Bischof. Active monocular localization : Towards autonomous monocular exploration for multirotor mavs. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3848–3855, 2014. ISBN 9781479936847. doi: 10.1109/ICRA.2014.6907417.

[19] Liam Paull, Sajad Saeedigharahbolagh, Mae Seto, and Howard Li. Sensor driven online coverage planning for autonomous underwater vehicles. *IEEE International Conference on Intelligent Robots and Systems*, pages 2875–2880, 2012. ISSN 21530858. doi: 10.1109/IROS.2012.6385838.

[20] Christian Potthast and Gaurav S. Sukhatme. A probabilistic framework for next best view estimation in a cluttered environment. *Journal of Visual Communication and Image Representation*, 25(1):148–164, 2014. ISSN 10473203. doi: 10.1016/j.jvcir.2013.07.006. URL http://dx.doi.org/10.1016/j.jvcir.2013.07.006.

[21] Seyed Abbas Sadat, Kyle Chutskoff, Damir Jungic, Jens Wawerla, and Richard Vaughan. Feature-rich path planning for robust navigation of mavs with mono-slam. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3870–3875, 2014. doi: 10.1109/ICRA.2014.6907420.

[22] Steven M Seitz, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[23] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics*, 25(3):835–846, 2006. ISSN 07300301. doi: 10.1145/1141911.1141964.

[24] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. pages 1385–1392, 2013. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.175.