Taking the Scenic Route to 3D: Optimising Reconstruction from Moving Cameras

Oscar Mendez
University of Surrey
Guildford GU2
o.mendez@surrey.ac.uk

Simon Hadfield
University of Surrey
Guildford GU2
s.hadfield@surrey.ac.uk

Nicolas Pugeault
University of Exeter
Exeter EX4
n.pugeault@exeter.ac.uk

Richard Bowden
University of Surrey
Guildford GU2
r.bowden@surrey.ac.uk

Abstract

Reconstruction of 3D environments is a problem that has been widely addressed in the literature. While many approaches exist to perform reconstruction, few of them take an active role in deciding where the next observations should come from. Furthermore, the problem of travelling from the camera's current position to the next, known as pathplanning, usually focuses on minimising path length. This approach is ill-suited for reconstruction applications, where learning about the environment is more valuable than speed of traversal.

We present a novel Scenic Route Planner that selects paths which maximise information gain, both in terms of total map coverage and reconstruction accuracy. We also introduce a new type of collaborative behaviour into the planning stage called opportunistic collaboration, which allows sensors to switch between acting as independent Structure from Motion (SfM) agents or as a variable baseline stereo pair.

We show that Scenic Planning enables similar performance to state-of-the-art batch approaches using less than 0.00027% of the possible stereo pairs (3% of the views). Comparison against length-based pathplanning approaches show that our approach produces more complete and more accurate maps with fewer frames. Finally, we demonstrate the Scenic Pathplanner's ability to generalise to live scenarios by mounting cameras on autonomous ground-based sensor platforms and exploring an environment.

1. Introduction

One of the frontiers in vision is its use in autonomous vehicles. Vision-Based 3D reconstruction has the capability to provide autonomous agents with detailed reconstructions that can be used to reason about the environment. However, as datasets have grown, and real-time computer-vision has become prevalent, the focus has shifted from *how to* reconstruct, to *what to* reconstruct. A number of recent techniques have been proposed, that take an active role in the data collection process, reducing computation time by pre-selecting the most informative views of the scene. However, unless the dataset has been collected offline, the sensor must travel between these viewpoints. During this travel time, the sensor may not be contributing significantly to the model. In this paper we propose an active approach to visual reconstruction, that considers not only the next best viewpoint, but the path (or sequence of viewpoints) for it to take. We achieve this by adapting techniques from the robotic pathplanning literature; introducing a computer-vision based cost space and applying pathplanning to this cost space rather than the traditional Euclidean world.

We also propose an extension to this, which allows us to use multiple monocular cameras to operate in a *collaborative* manner. The naive solution is to have multiple sensors operating independently in the same physical space and then fusing the resulting maps. However, this does not fully exploit the potential for collaboration. Reasoning jointly about the cameras' observations allows us to exploit valuable information. Collaborative building of the map, by two or more cameras, has the potential to dramatically increase reliability, while reducing the time needed to perform the reconstruction.

To summarise, we present a novel approach that is capable of using multiple mobile cameras in order to automatically reconstruct a scene from monocular images. Our main contributions are a *Scenic Pathplanner* that efficiently searches Special Euclidean Space (SE(3)) for paths of high information gain and an *Opportunistic Collaboration* framework that determines the behaviour of the cameras jointly during the pathplanning stage, to act either as a wide-baseline stereo pair, or as independent SfM agents. It is important to note that every step along the Scenic Route is not necessarily a local optima. Instead, the Scenic Pathplanner trades information gain against path length. In order to validate our approach's capabilities, we perform guided reconstruction of a room from an offline dataset consisting of $\sim$8500 images. To validate online capabilities, we mount cameras on mobile ground-based robots and autonomously reconstruct a room in real time.

2. Related Work

Collaborative sensors capable of intelligently reconstructing an environment have four fundamental problems to over-

come. First, they must be able to reconstruct their environment. Second, they require the ability to decide where in that map they should go to next. Third, autonomous entities should be able to negotiate an environment to reach their goal. Finally, they should be able to decide whether collaboration with another sensor is in their best interest.

Reconstruction algorithms can be divided into online and offline approaches. Online approaches tend to be sparse both in time and space, while offline approaches are usually denser and can deal with unstructured datasets.

In terms of online approaches, Klein and Murray[20] introduced the concept of splitting pose estimation and mapping into independent threads. This allowed for robust Simultaneous Localization and Mapping (SLAM) algorithms to run in real time. More recent contributions, such as that by Mur-Artal *et al*. [28] and Engel *et al*. [7] add an explicit loop closure thread and are generally more robust. Online systems are good for pose estimation and stabilisation, but are generally not dense enough to provide scene understanding and/or detailed reconstructions.

Offline approaches, commonly referred to as Multi-View Stereo (MVS), typically find pairwise stereo correspondences and use large optimisations to estimate dense and accurate reconstructions, such as work by Snavely *et al*. [34]. Denser reconstructions were achieved by Furukawa and Ponce [11] who use sparse feature matching and patch growing, along with photometric and visibility constraints to produce dense reconstructions. Jancosek *et al*. [17] extend [11] by attempting to actively select views in a Next-Best View (NBV)-like approach to make large datasets feasible by estimating feasible stereo pairs, but provide no results on partial-image reconstruction. Hornung *et al*. [16] use an octree-like hierarchical volumetric reconstruction along with graph cut minimisation. More recently, Galliani *et al*. [12] expand the patch-matching idea proposed by Bleyer *et al*. [3] to use more than two views.

However, the computational cost for dense reconstruction of large structures can be prohibitive, preventing their use online, and lack the ability to choose 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.

In order to perform efficient reconstruction that maximises quality and coverage using a minimum amount of data (such as in [1] and [24]), it is necessary to actively select where the NBV is. NBV estimation can be divided into two main categories: *exploration* and *refinement*.

Exploratory NBV aims to generate the most complete map of the (unknown) scene. It is generally based on the concept of a frontier, for example in the work by Heng *et al*. [14]. This approach uses a precomputed lattice and defines frontier locations as edges between 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*. [30] similarly uses coverage and distance to the NBV. Sim *et al*. [33] evaluate hard-coded exploration strategies to create a visual map. Most similar to our work is Bourgault *et al*. [4], who use an occupancy grid and a measure of information to perform adaptive robotic exploration on a 2D laser-scan map. These approaches 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 to select poses that improve the 3D model accuracy. For example, Forster *et al*. [9] use depth uncertainty to estimate the best areas of the map to explore. Hoppe *et al*. [15] create a full network of poses for an Unmanned Aerial Vehicle (UAV), but assume prior knowledge of the environment. Sadat *et al*. [32] and Mostegel *et al*. [27] plan optimal paths for a monocular Visual Odometry (VO) system, but require a set endpoint. Mauro *et al*. [25] focus on offline datasets, while Banta *et al*. [2] and Potthast and Sukhatme [31] focus on single object NBV. Our work is most similar to Mendez *et al*. [26], who use a joint octree and pointcloud approach to NBV but are limited to offline datasets and perform a brute-force search of the available views.

In this paper, we present an approach that is capable of sampling the camera pose-space to define not only an NBV, but the path to it.

Moving directly from the current view to the NBV discards a lot of useful information along the way. Not only that but, in the case of online sensors, it leaves the navigation to the user. The problem of estimating a path between the current and goal state is known as pathplanning in the field of robotics. The current state-of-the-art for pathplanning are stochastic tree-based algorithms such as Probabilistic Road Map (PRM) [19] and Rapidly-exploring Random Tree (RRT) [22]. Generally speaking, they work by sampling a state space in order to provide collision-free trajectories from a start state to a goal state. PRM [19] is better suited to multi-query scenarios where the same roadmap resolves various queries. On the other hand, RRT [22] algorithms build a tree for every query. However, these algorithms do not guarantee optimality. Work done by Karaman and Frazzoli [18] extended these approaches to guarantee asymptotic optimality and renamed them PRM* and RRT*. More recently, informed sampling has become the state-of-the-art. Work such as that by Gamell *et al*. [13] reduce the state-space of the problem by only sampling from regions that are capable of reducing the cost of the current solution.

In robotics, it is often assumed that the cost of an individual state in the configuration space is intrinsically linked to the pose alone. However, from a computer vision point of view, we know all images from a moving camera provide information important to reconstruction. Therefore, our work

breaks this assumption by relating the cost of a state not only to the pose, but also to the geometry of the scene. Since the geometry of the reconstructed scene is constantly changing, we focus on RRT* since the tree can be built and discarded as needed. To our knowledge, we are the first to implement a pathplanning algorithm that defines the optimum path as one that traverses areas of high information gain (which we refer to as "the scenic route") while optimising a stereo arrangement with other sensors, thereby making the pathplanning algorithm enforce soft collaboration constraints.

Our approach is designed to generalise to more than one sensor. Therefore, it is not only necessary to merge each sensor's interpretation of the world, but to observe the emergent behaviours given the sensors' knowledge of each other. Forster *et al*. [8] use two independent cameras mounted on UAVs that create sparse maps along with an overlap detector to merge the pose-graphs of both cameras. Lazaro *et al*. [23] merge their maps in a decentralised agent-to-agent mode. When the sensors are in the vicinity of each other they share a local version of the map. Each sensor then augments its own SLAM pose graph with a small amount of relative poses to other robots. Similarly, Cunningham *et al*. [6] use local maps, neighbouring sensor information and robust data association to provide a decentralised approach.

To our knowledge, we are the first to implement what we refer to as "opportunistic collaboration" between sensors. This is a higher level form of collaborative behaviour where the cameras come to a consensus during the initial NBV planning stage. The sensors will agree on an initial interpretation of the world and then choose to either act as a variable baseline stereo pair, or to explore independently, depending on the scene properties.

3. Methodology

In this section we describe our approach which creates dense maps using opportunistically collaborative cameras travelling along the scenic route. We use an octree representation of the world, along with Next-Best View (NBV) and Next-Best Stereo (NBS) costs. However, we propose a novel Sequential Monte-Carlo (SMC) approximation to these costs, which allows development of path-planning algorithms that operate directly in the cost space.

In section 3.1 we use images in a state-of-the-art reconstruction algorithm based on dense correspondences obtained from Deep Learning. This reconstruction is then added to a map using an octree structure to perform data association. Section 3.2 describes the NBV cost. In our first contribution, section 3.2.1, we create an approximation of the view quality cost-space using the NBV in a novel SMC formulation. This is done in order to efficiently find the goal state. In our second contribution, section 3.3, we describe the scenic route pathplanner which uses the cost-space approximation of section 3.2.1 to perform an RRT-based search between the current sensor position and the goal-state. Finally, our third contribution is the joint planning of opportunistic collaboration described in section 3.3.3. The cameras jointly plan a number of paths based on various collaborative or independent behaviours. They then execute a combination of behaviours which is expected to collectively maximise the information gained from the environment.

3.1. 3D Reconstruction

To reason about informative views, we must have an interpretation of the current scene geometry. Images from the cameras are used to estimate dense, bidirectional matches using a deep-learning based approach [35]. These dense correspondences are triangulated to obtain a cloud of 3D points and their covariances (i.e. their uncertainty). These pointclouds provide a detailed representation of the scene; however, they are simultaneously too dense to plan navigation and perform data association while also being too sparse for geometric operations such as ray casting. Filtering and storing the points within an octree data structure facilitates efficient lookup and geometric operations. We store a set of voxels $V = (V_o \cup V_e \cup V_u)$, comprising *occupied* (V_o), *empty* (V_e) and *unobserved* (V_u) voxels. *Occupied* voxels contain geometry, *empty* voxels are empty space the sensor can occupy and *unobserved* voxels are unknown areas. Data association is performed by searching an octree for a match for each new point, then updating this match with the new observation. If no match is found, the point is added to the map.

3.2. Next-Best View (NBV) Goal Estimation

In order to explore a 3D environment, each sensor needs to be able to make decisions about where in space they are going to next; the goal state. Since the goal of the sensors is to model the environment, the goal state is defined as the pose in SE(3) (position + orientation) that maximises the potential information gain of the map i.e. the Next-Best View (NBV).

3.2.1 Approximate View Quality Cost-Space

It would be intractable to attempt an exhaustive search for the NBV in SE(3), even if this is done on the discretised octree. Instead, we propose a SMC sampling method that uses information contained in the octree to approximate the distribution of NBV costs across the scene.

It would be counter-productive to sample from voxels we know contain points ($v \in V_o$) as there is a high probability of a collision. Therefore, the first step is to extract the empty voxels $v \in V_e$. We then uniformly sample from these voxels and, for every voxel sampled, randomly assign an orientation. Orientation sampling is application dependant, and we discuss techniques in section 4. The weight w_i of each sample $i \in I$ is then estimated as

$$w_i = 1 - C^i_{nbv} \tag{1}$$

where C_{nbv} is the NBV cost for that view. Our approach is agnostic to the underlying NBV cost, with the exception that it must lie in the range $[0, 1]$. In this paper, we use the NBV cost defined by Mendez *et al*. [26], which is briefly defined in section 3.2.2.

Once the weights have been estimated, it is necessary to perform a resampling stage to better model the underlying cost function. This resampling is done in three steps. First, we propagate a small percentage of the best particles (I_p). Second, we do a weighted resampling (I_g) from the set of particles and apply gaussian noise. In most SMC applications this would be enough to make the solution converge over time. However, in this case the location of the NBV can change drastically as observations are added. Therefore, it is necessary to uniformly sample a smaller number of new particles (I_u) from the empty voxels, to ensure detection of newly emerging peaks in the cost function. The complete set of particles I is then $I = I_p \cup I_g \cup I_u$. Note that during resampling we do *not* want the samples to converge on a single location, as we need an approximation of the full cost function to plan the scenic route.

3.2.2 Next Best View Cost

For the experiments in this paper, the NBV is estimated by casting a set of rays, S_r, from each candidate pose through the image plane and into the scene. It is important to note that S_r is a set of rays cast from the same candidate pose. The candidate pose generation is done as shown in section 3.2.1. The NBV cost of the candidate pose is the average cost of all the rays. The cost of each ray directly depends on what it intersects. In the case that the rays intersect a voxel that contains points ($v \in V_o$), the cost of each point can be calculated as

$$\phi(r, p) = e^{-||\lambda_p e_p \times r||}\,, \tag{2}$$

where $r \in S_r$ is the ray cast from a candidate pose, $v \in V_o$ is the voxel on which the ray is incident and $p \in P_v$ is a point in v. λ_p and e_p are the largest eigenvalue and eigenvector, respectively, of the covariance Σ_p of p. Consequently, the cost of a ray is defined as the average of all the points $p \in P_v$ contained in the intersected voxel

$$\psi(r, v) = \frac{1}{|P_v|} \sum_{p \in P_v} \phi(r, p)\,. \tag{3}$$

If the ray does not intersect an occupied voxel, we assign it a cost of $\gamma \in [0, 1]$, which is a user-defined parameter that biases the cost-space towards exploration 0, or refinement 1. Finally, the NBV cost of a particular pose is defined as

$$C_{view} = \frac{1}{|S_r|} \sum_{r \in S_r} \begin{cases} \psi(r, v) & \text{if } v \in V_o \\ \gamma \in [0, 1] & \text{else } v \in V_u\,. \end{cases} \tag{4}$$

Note that equation 2 will give the lowest cost when r is perpendicular to e_p, meaning the camera is ideally positioned to decrease the uncertainty of that point.

3.3. Scenic Pathplanning

A sensor should also be capable of negotiating a trajectory to its goal. This implies smooth continually updated motion planning, collision avoidance and cost minimisation. A traditional robotics approach would see the path length minimised. Indeed, most planners perform precisely this kind of operation. However, if the goal is reconstruction, then taking the shortest path might result in unfavourable poses for both localization and reconstruction. The sensor will also miss good views along the way to its goal.

In this section, we describe a novel approach that allows the estimation of a "scenic" route. The scenic route is defined as the path that will maximise the potential information gain in the map, both in terms of accuracy and coverage.

3.3.1 Next-Best View Pathplanning

Naively, iterative NBV estimation could be treated as a path. However, this would have no guarantees over the path length or optimality. Instead, a tree-based approach such as RRT can be used to bias the search towards the goal, optimize path length as well as scenic value, and guarantee asymptotic optimality.

An RRT implementation such as [13] would not only be expensive and inefficient, but it would also be biased towards finding short paths between the start and goal. This is because RRT-based methods are designed to explore large Voronoi regions of the pose space with no regard to the cost of that area. Unfortunately, it is ill-defined to solve a problem when the cost is not intrinsically linked to the pose, but is a function of the pose and the reconstructed geometry. Instead, what is needed is a method that biases the search towards areas rich in good views, while minimising the stereo cost of the path.

We can define a tree in SE(3) space (i.e. both position and orientation) as a collection of nodes $Q = \{q\}$, where the root node is defined as $x_{init} \in Q$. The task of growing a tree to get from start to goal would usually be done in a standard RRT by first drawing a sample q_{rand} from SE(3). Second, finding the nearest vertex q_{near} in the tree from that sample. Third, adding a new vertex q_{new} a predefined step Δ_q in direction q_{near} to q_{rand}. The edge cost C_{edge} is then the Euclidean distance $|q_{near} - q_{new}|$.

Instead, we present a novel method that combines the high-dimensional exploration of RRTs* with a bias towards pre-computed areas of high information gain. Algorithm 1 shows how the scenic pathplanning tree would be formed in an RRT* context. We first define the start state (x_{init}) as the current position. The goal state (x_{goal}) is the current peak of the NBV cost function, as estimated in section 3.2. Instead of SE(3), our approach samples from the prior distribution of good NBV candidates estimated in section 3.2.1. Stochastically sampling from this distribution biases the growth of the tree towards areas with good NBV cost.

This novel formulation allows us to estimate paths with

Algorithm 1 RRT version of Scenic Pathplanner.

```
1:  function BUILDSCENICRRT(x_init, x_goal)
2:      G.ADD_VERTEX(x_init)
3:      while dist(x_goal, G) ≥ Δ_q do
4:          q_rand ← SAMPLENBVCOSTSPACE( )
5:          q_near ← NEARESTVERTEX(q_rand, G)
6:          q_new ← NEWVERTEX(q_rand, q_near, Δ_q)
7:          C_edge = |q_near − q_new|
8:          G.ADD_VERTEX(q_new)
9:          G.ADD_EDGE(q_near, q_new, C_edge)
10:     end while
11:     return G
12: end function
```

high information, from the current pose to the NBV. However, we also want to keep the sensor trained on the geometry during the trajectory. More importantly, we want to allow agents to plan collaborative paths. In order to do this, it is necessary to define a cost-function to replace the Euclidean distance of the graph edges.

3.3.2 Stereo Pair Pathplanning

In this section, we define the cost of the graph nodes as a stereo-pair cost. The further away a pair of views are from an "ideal" stereo pair, the higher the cost. The stereo-pair can be made up of successive poses along the path, or be a collaborative stereo-pair with another agent. Either way, the quality of a particular configuration always depends on the same parameters. Namely, the stereo camera baseline, vergence angle and the distance between the known geometry and vergence point.

Perhaps the most important aspect of a stereo pair is its baseline. It must be short enough to allow for robust correspondence estimation, while being large enough to provide good depth estimates. Since the cameras are fully mobile, it makes little sense to enforce a particular baseline. Instead, we parameterise the baseline as a fraction of the distance to the intersection of the rays (r_L, r_R) cast through the principal point of both cameras. That is, we enforce

$$d_{LI} = d_{RI} = \alpha d_B \,, \tag{5}$$

where I is the intersection of both rays, $|r_{LI}| = d_{LI}$ and $|r_{RI}| = d_{RI}$ are the distances from the cameras (Left and Right) to the intersection point I, and d_B is the stereo baseline. We implement this as a soft constraint with the cost function

$$C_B = \frac{|d_{LI} - \alpha d_B|}{\alpha d_B} + \frac{|d_{RI} - \alpha d_B|}{\alpha d_B} + \frac{|d_{LI} - d_{RI}|}{d_B} \,. \tag{6}$$

Figure 1 shows a sample camera configuration, where this soft constraint is formed by the red lines and the baseline. Enforcing this has a two fold effect. First, it makes the baseline variable with the distance to the point being imaged. A camera close to an object will prefer to have a small baseline, while large baselines will be preferred for distant

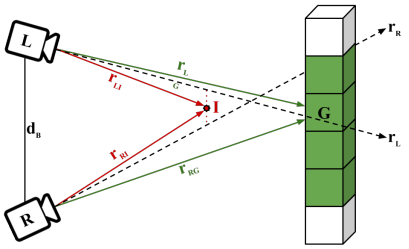


Figure 1: Sample stereo pair geometry.

objects. Second, equation 5 also implicitly enforces a viewing angle as it can be shown that a triangle with sides $d_{LI} = d_{RI} = \alpha d_B$ has an angle $\beta = acos\left(1 - \frac{1}{2\alpha^2}\right)$. To handle the case where the principal rays do not intersect, we penalise large angles between the rays (r_L, r_R) and the rays to the intersection point (r_{LI}, r_{RI})

$$C_T = acos\left(\frac{|r_L \cdot r_{LI}|}{\|r_L\|\|r_{LI}\|}\right) + acos\left(\frac{|r_R \cdot r_{RI}|}{\|r_R\|\|r_{RI}\|}\right) \,. \tag{7}$$

These costs enforce a good stereo arrangement for anything near the intersection point I. However, having a good configuration is useless if the geometry being imaged is not taken into account. Therefore, we define $G \in V_o$ as the closest occupied voxel to the intersection point I and (r_{LG}, r_{RG}) as the rays from the left and right (respectively) to G. These are the green rays in figure 1. Penalising large distances between I and G would be unfavourable to imaging from far away (large baseline). Instead, we penalise having a large angle between the rays to I and G

$$C_G = acos\left(\frac{|r_{LI} \cdot r_{LG}|}{\|r_{LI}\|\|r_{LG}\|}\right) + acos\left(\frac{|r_{RI} \cdot r_{RG}|}{\|r_{RI}\|\|r_{RG}\|}\right) \tag{8}$$

$$C(L,R) = \sigma_1 C_B + \sigma_2 C_\beta + \sigma_3 C_T + C_G \tag{9}$$

These costs ensure that successive poses in our tree are trained on similar geometry. This allows easy SfM for monocular sensors and/or data association for active/stereo sensors. However, we can also leverage the same cost in order to plan "collaborative" paths where more than one sensor is trained on the same geometry.

3.3.3 Opportunistic Collaboration

Until now, we have considered a single camera performing guided reconstruction of its environment. However, if there are multiple cameras, the proposed techniques can be extended to perform joint pathplanning of all cameras simultaneously. However, we do not want to constrain the cameras to act collaboratively. Therefore, we can grow separate trees depending on the mode of operation. This allows the sensors to automatically select the best path from both trees and become *opportunistically collaborative*.

In the case of two monocular sensors, this is performed as follows: during initial pathplanning we treat the robots as

being completely independent from each other. We assume that each sensor only knows the current position of the other robot. Using this information it is possible for each agent to independently grow two different trees and extract a path for each camera.

First, we grow a collaborative stereo tree. In this case, we use the other agent's last known position, along with each new tree node q_{new}, in equation 9 to estimate the cost. This tree will attempt to find a path through the space that maintains a good stereo pair (with the other agent), while also travelling through areas of high information gain.

The second SfM tree is grown in order to optimise the stereo configuration of each successive node along the path. That is, the cost of each new node q_{new} is computed from equation 9 between it and its parent node q_{near}.

In both cases, the cost for each path is computed by simply adding all the successive stereo pairs until the goal is reached. Since we are trying to estimate a "scenic" path to the goal, it is also important for the estimation to have some notion of path length. We enforce this by estimating the path cost integral, where the cost of each edge can be computed as

$$C_{edge} = \frac{C_{q_{new}} + C_{q_{near}}}{2} |q_{new} - q_{near}|. \qquad (10)$$

Finally, once all paths have been estimated, the agents make an autonomous decision about what the best course of action is. They each share their path costs and the path with the minimum cost will dictate how the sensors operate. There are two possible scenarios. In the first, one agent will remain static while the other moves to a position of vantage to collect more data. In the second, they both move towards independent goals while performing SfM. Once the next observation(s) are obtained, a new goal and path are estimated (for each agent) and the process is repeated. This approach doesn't account for the overlap in observations over the whole trajectory. However, this is mitigated by the fact we only use the first pose in the path before re-planning.

4. Results

The contributions of this paper have focused on allowing a pair of mobile cameras to opportunistically and collaboratively explore an unknown area and rapidly create a 3D reconstruction of the scene. An effective system should be able to plan a path which can rapidly explore and refine the map using a small number of maximally informative views. To demonstrate this, we first present qualitative and quantitative evaluation on an online dataset followed by evaluation on a live system that can autonomously reconstruct a scene.

4.1. Offline Dataset Reconstruction

We collect a dataset that consists of several minutes of a UAV moving around a room. This footage is extremely dense in the pose space, as we move the camera multiple times over the same area but with different orientation. We then use this footage to extract 8500 images from the camera and run them through a state-of-the-art batch reconstruction algorithm [36][5]. This provides us with a set of images with their respective pose in 6-Degrees of Freedom (DoF) space. In order to obtain ground truth information, we use a depth sensor running Kinect Fusion[29].

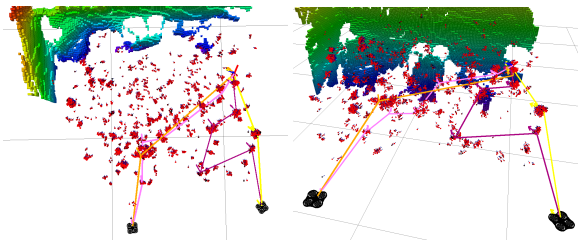


Figure 2: UAVs Pathplanning. The purple tracks show SfM paths, the yellow tracks show Collaborative Stereo paths.

4.1.1 Experimental Setup

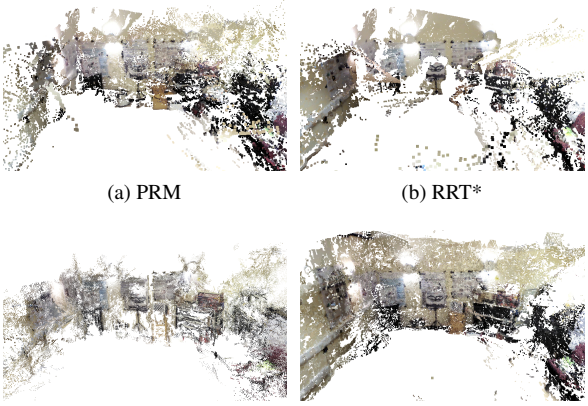
Since the objective of these experiments is to map an unknown environment, we start the process with absolutely no knowledge of the scene. We only provide the algorithm with a pair of images which are used to initialise the reconstruction (and octree). After that, the approach is entirely autonomous. At each iteration, we perform i) Stereo/SfM Reconstruction (section 3.1), ii) Goal Estimation via SMC (section 3.2) and iii) Scenic Pathplanning and Opportunistic collaboration (3.3). Once the decision on whether to collaborate has been made, we take the first pose in the estimated scenic path, and repeat the operation. The goal estimation is done on a 4-DoF manifold of SE(3); this allows us to only sample the yaw angle of the camera, as views looking at the ceiling and/or floor are not very informative. The scenic pathplanning is done in full SE(3). It is only once the poses have been selected that, for the purposes of this evaluation, we select the closest pose in the dataset. This allows repeatability during tests. For these experiments we set $\alpha = 3$, $\gamma = 0.7$ and the various cost weightings to 1.

4.1.2 Qualitative Analysis

The SMC is performing a weighted sample. This means the larger the grouping of particles, the more benefit the sensor would get from visiting it. Therefore, we expect the scenic pathplanning to prefer these clusters as it makes its way to the goal. Figure 2 shows the four different paths estimated from the cameras to the goal pose. As expected, the paths show a bias towards areas of high particle concentration, thereby making the sensor take a more scenic route. In these figures, the paths computed in yellow and orange are the collaborative stereo paths, those in purple are for SfM. Notice how the SfM paths make their way towards the goal in a zig-zag fashion. This happens because the pathplanning is aiming to minimise the stereo costs and therefore prefers wider baselines than a direct path would afford. In addition, the zigzags can be seen

(a) Kinect Fusion [29] (b) Proposed

Figure 3: Close up of the reconstruction performed by Kinect Fusion and the proposed Scenic Route Reconstruction



(a) PRM (b) RRT*

(c) VSFM (d) Proposed

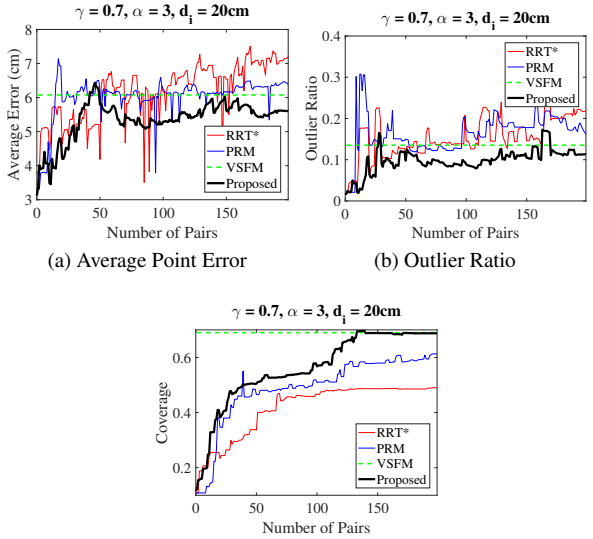
Figure 4: Comparison of the reconstructions done by the different pathplanning algorithms, and the batch approach.

to flow into areas with high particle density. A close-up of the resulting reconstruction can be seen in figure 3b, with a corresponding ground truth reconstruction in figure 3a. Note that we are able to extract a similar level of scene coverage, while maintaining low depth error. A more complete reconstruction, using 150 stereo pairs, can be seen in figure 4, where we show results from two other pathplanning approaches and an online batch approach. Figures 4a and 4b show reconstructions done by PRM and Rapidly-exploring Random Tree (RRT*), respectively. Since these approaches are not trying to optimise the reconstruction during navigation, they lead to either high noise (PRM) or low scene coverage (RRT*). Figure 4c shows the reconstruction obtained by 8500 frames of VisualSFM+CMVS [36][5][10]. Notice that it is not as dense, and has considerably more noise than the proposed method.

4.1.3 Quantitative Analysis

We demonstrate that the proposed scenic pathplanning leads to significantly better reconstructions than generic pathplanners. Each pathplanner is integrated within the same reconstruction framework and is evaluated based on the average point error, number of outliers and coverage. We also compare against VisualSFM+CMVS [36][5][10] and show that we achieve comparable results with a fraction of the data.

The outlier ratio is computed as the fraction of reconstructed points which are more than a threshold distance d_i (set to 0.05 in these experiments) from any part of the ground



(a) Average Point Error (b) Outlier Ratio

(c) Coverage Ratio

Figure 5: Reconstruction performance measures plotted against number of image pairs, for various different pathplanning algorithms (and the baseline batch system).

truth. The average reconstruction error is calculated over all inlier points. Finally to compute the coverage, we find the fraction of ground truth points which are represented by at least one point in the reconstruction (within d_i).

In order to make the comparison fair, we give PRM and RRT* our computed goal-state rather than selecting a random one. However, PRM and RRT* are both optimising the path length to the goal state. This makes these algorithms incapable of enforcing stereo constraints. As such, the robots tend to observe different regions for most of the reconstruction.

For the VisualSFM+CMVS baseline, we use the full dataset to perform the reconstruction. This provides a baseline value for each metric. The dataset consists of over 72 million possible stereo pairs. While some of these pairs might be trivially discarded by an algorithm that has access to the image and pose data, we explicitly do *not* use any of this information. To simulate a live robotic navigation task, the planning is done on the SE(3) manifold and is only related back to the dataset when choosing the nearest-neighbour pose. Therefore it is important to keep in mind that selecting 200 stereo pairs, as shown in figure 5, is still $< 0.00027\%$ of the possible pairs.

Figure 5c demonstrates we can achieve coverage that is comparable to VisualSFM+CMVS - nearly 70% of the ground truth - using under 150 pairs. More importantly, the proposed method explores the space faster than both competing pathplanners while also achieving a higher final coverage. Also notice that we exhibit a "stepped" behaviour in the curve, which corresponds to autonomous switching between exploration and refinement.

Figure 5a shows how the average point error progresses

(a) Proposed (b) RGB-D SLAM [21]

(c) Proposed (d) RGB-D SLAM [21]

Figure 6: Reconstruction comparison for our pathplanning algorithm and state-of-the-art RGB-D SLAM [21].

with the number of frames. The scenic pathplanner consistently outperforms PRM and is only worse than RRT* for a short period between frames $30 - 50$. This is because, as shown in figure 5c, that period corresponds to rapid exploration that RRT* does not perform. The scenic pathplanner in general is significantly more accurate than VisualSFM+CMVS. Areas where VSFM outperforms the proposed technique correspond directly to the periods of exploration, when coverage grows rapidly. In fact, in areas of low coverage growth (refinement behaviour), the error decreases below that of VisualSFM+CMVS (frames $60 - 100$) and only grows larger during an exploration period ($100 - 130$).

Finally, in figure 5b, the proposed method can be seen to consistently exhibit fewer outliers than all other pathplanners. Indeed, apart from failure cases at frames 30 and 160 (which added noisy measurements to the map) we also outperform VisualSFM+CMVS, maintaining around 10% outliers.

Having qualitatively and quantitatively validated our approach on online datasets, we now validate on a live system.

5. Online Reconstruction

In this section, we first discuss implementing a live reconstruction system that uses the proposed approach to perform an intelligent, dense 3D reconstruction of its environment. Qualitatively, we show how the results compare to a dense RGB-D SLAM [21] approach. Quantitatively, we show that the scenic pathplanner is not only capable of autonomously reconstructing the environment, but that setting the value of γ will either encourage or discourage exploration. Further examples can also be found in the supplementary material.

5.1. Experimental Setup

In order for the sensors to autonomously navigate their environment, we perform vision-based SLAM to obtain a consistent pose estimate. This pose estimate is then used in a sensor-fusion framework, along with the Inertial Measurement Unit (IMU) and wheel odometry to obtain a robust pose estimate for each camera. While this is enough for a single agent to perform reconstruction, we are interested in multi-agent reconstruction. Therefore, we perform a reprojection-error based pointcloud alignment on the sparse visual landmarks from each SLAM system. This allows us to estimate a similarity transform between the cameras, effectively putting them in the same coordinate frame. Once the sensors are operating in the same coordinate frame, the current image and pose of each camera is used to initialise the reconstruction (and octree). For these experiments, we set $\alpha = 7$ and $d_i = 10$cm. This enforces a narrower baseline which makes it easier for the SLAM system to keep track of the pose (less pure rotation). Since these experiments consist of a ground-based sensors, we also limit the sampling for NBV and pathplanning to SE(2). While this is not strictly necessary, it reduces complexity and increases performance.

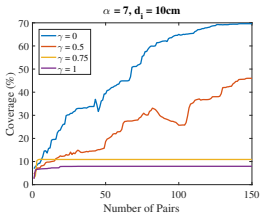


Figure 7: Coverage plotted against number of image pairs, note how lower values of γ are more exploratory.

In figure 6, we show that our approach autonomously reconstructs pointclouds that are both dense and detailed. The level of detail is comparable to the "ground truth" obtained using an RGB-D camera. Our approach also computes the navigability of the space it reconstructs. Therefore, it knows which areas of the map the sensor can realistically reach and which are out of bounds.

In figure 7 we quantitatively demonstrate that our approach is capable of autonomously exploring an environment. The system is run with different values of γ and the achieved coverage is shown. This demonstrates that the sensor's willingness to explore its environment is impacted significantly by γ. When $\gamma = 0$, the system begins exploring rapidly. As γ increases towards 1, the reward of exploration decreases until there is no benefit to it at all; in this case, the sensors will prefer to look at the same geometry from different angles.

6. Conclusion

In conclusion, we have presented a novel approach that can coordinate at least two cameras in an *opportunistically collaborative* way, creating a dense reconstruction of their environment. We leverage the approximate NBV cost distribution to bias a random tree-based search method toward areas of large information gain. This explores SE(3) to find a *scenic path* between the camera and the NBV. In the future, it would also be interesting to adapt this approach to use depth sensors.

Acknowledgements

This work was funded by the SNSF Project *SMILE* grant CRSII2_160811.

References

[1] T. Arbel and F. P. Ferrie. Informative Views and Sequential Recognition. In *ECCV*, 1994.

[2] J. E. Banta, L. M. Wong, C. Dumont, and M. a. Abidi. A next-best-view system for autonomous 3-D object reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans.*, 30(5), 2000.

[3] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *BMVC*, 2011.

[4] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *IROS*, 2002.

[5] W. Changchang. Towards Linear-Time Incremental Structure from Motion. In *3DV*, 2013.

[6] A. Cunningham, K. M. Wurm, W. Burgard, and F. Dellaert. Fully distributed scalable smoothing and mapping with robust multi-robot data association. In *ICRA*, 2012.

[7] J. Engel, T. Schöps, J. Sturm, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *ECCV*, 2014.

[8] C. Forster and S. Lynen. Collaborative monocular SLAM with multiple micro aerial vehicles. In *IROS*, 2013.

[9] C. Forster, M. Pizzoli, and D. Scaramuzza. Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicles. *Robotics: Science and Systems*, 2014.

[10] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010.

[11] Y. Furukawa and J. Ponce. Accurate, Dense, and Robust Multi-View Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8), 2010.

[12] S. Galliani and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, 2015.

[13] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot. Informed rrt*: Optimal incremental path planning focused through an admissible ellipsoidal heuristic. In *IROS*, 2014.

[14] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys. Efficient Visual Exploration and Coverage with a Micro Aerial Vehicle in Unknown Environments. In *ICRA*, 2015.

[15] C. Hoppe, A. Wendel, S. Zollmann, K. Pirker, A. Irschara, H. Bischof, S. Kluckner, and S. C. Technology. Photogrammetric Camera Network Design for Micro Aerial Vehicles. *Computer Vision Winter Workshop*, 2012.

[16] A. Hornung and L. Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *CVPR*. IEEE, 2006.

[17] M. Jancosek, A. Shekhovtsov, and T. Pajdla. Scalable multi-view stereo. In *ICCV*, 2009.

[18] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7), 2011.

[19] L. Kavraki, L. Kavraki, P. Svestka, P. Svestka, J.-C. Latombe, J.-C. Latombe, M. Overmars, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensionalconfiguration spaces. *IEEE Robotics & Automation Magazine*, 12(4), 1996.

[20] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *ISMAR*, 2007.

[21] M. Labbe and F. Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *IROS*, 2014.

[22] S. M. LaValle. Rapidly-exploring random trees a new tool for path planning. Technical report, 1998.

[23] M. Lazaro, L. Paz, J. Castellanos, and G. Grisetti. Multi-robot SLAM using condensed measurements. In *IROS*, 2013.

[24] É. Marchand and F. Chaumette. Active vision for complete scene reconstruction and exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1), 1999.

[25] M. Mauro, H. Riemenschneider, A. Signoroni, R. Leonardi, L. J. Van Gool, and I. Brescia. A unified framework for content-aware view selection and planning through view importance. In *BMVC*, 2014.

[26] O. Mendez, S. Hadfield, N. Pugeault, and R. Bowden. Next-best stereo: extending next best view optimisation for collaborative sensors. In *BMVC*, 2016.

[27] C. Mostegel, A. Wendel, and H. Bischof. Active Monocular Localization : Towards Autonomous Monocular Exploration for Multirotor MAVs. In *ICRA*, 2014.

[28] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5), 2015.

[29] R. a. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. *ISMAR*, 2011.

[30] L. Paull, S. Gharahbolagh, M. Seto, and H. Li. Sensor driven online coverage planning for autonomous underwater vehicles. *IROS*, 2012.

[31] C. Potthast and G. S. Sukhatme. A probabilistic framework for next best view estimation in a cluttered environment. *Journal of Visual Communication and Image Representation*, 25(1), 2014.

[32] S. A. Sadat, K. Chutskoff, D. Jungic, J. Wawerla, and R. Vaughan. Feature-rich path planning for robust navigation of MAVs with Mono-SLAM. In *ICRA*, 2014.

[33] R. Sim and G. Dudek. Effective exploration strategies for the construction of visual maps. In *IROS*, 2003.

[34] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics*, 25(3), 2006.

[35] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013.

[36] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *CVPR*, 2011.