Choosing Where To Go: Complete 3D Exploration With Stereo

Robbie Shade and Paul Newman

***Abstract*— This paper is about the autonomous acquisition of detailed 3D maps of a-priori unknown environments using a stereo camera - it is about choosing where to go. Our approach hinges upon a boundary value constrained partial differential equation (PDE) – the solution of which provides a scalar field guaranteed to have no local minima. This scalar field is trivially transformed into a vector field in which following lines of max flow causes provably complete exploration of the environment in full 6 degrees of freedom (6-DOF). We use a SLAM system to infer the position of a stereo pair in real time and fused stereo depth maps to generate the boundary conditions which drive exploration. Our exploration algorithm is parameter free, is as applicable to 3D laser data as it is to stereo, is real time and is guaranteed to deliver complete exploration. We show empirically that it performs better than oft-used frontier based approaches and demonstrate our system working with real and simulated data.**

I. Introduction

After switching a mobile robot on, where should it go? In this work we say it should try and look at, and thus map, every reachable surface. Of course initially only a fraction of the workspace is visible and plans must be made about where the robot should move to further increase the extent of its map. Ideally we would decide upon a smooth path to a new sensor pose which looks into a "maximally promising" area. But how do we define promising? How do we assign numbers to "promising"? It is evidently sensible to try and extend the boundary between explored and unexplored and in particular, to plan to view which is in some sense perpendicular to the boundary, to look directly in to the unseen regions of the workspace. This then implies we should consider the entire boundary between free and unexplored space. We cannot expect a single boundary to exist and so we must entertain the possibility that there could be a multitude of view points which would provide information rich views - which one to choose? Some could be far away and so costly to reach but resting on an extensive free space boundary, some could be close but perched on a narrow boundary. How should we balance these aspects and capture our intuition of a good exploration strategy given above?

II. Approach

In this paper we present a method for the active exploration of unstructured, a-priori unknown 3D environments using a stereo camera. There is no other input to the system - the stereo imagery used to produce depth measurements is also used to drive a visual SLAM system which provides pose estimation.

The authors are with the Oxford University Mobile Robotics Group {rjs,pnewman}@robots.ox.ac.uk

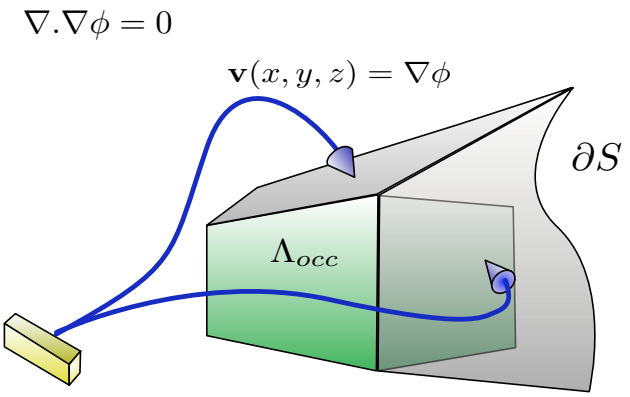


Fig. 1. Moving a camera to explore the world: A solution to Laplace's equation, a harmonic function ϕ, is calculated over the domain of known free space. The boundary of free space – ∂S – is either occupied by an obstacle or is a target for exploration as beyond it lies an unexplored region. The camera is guided to view these unexplored regions by following the gradient of $\nabla\phi$ – two such streamlines are shown.

Our knowledge about the state of the world is encoded in an octree based occupancy grid – a discretisation of space into voxels (volumetric pixels). Voxels are classified in one of three states: occupied, free space, or unknown and we refer to the sets of these voxels as $\mathbf{\Lambda}_{occupied}$, $\mathbf{\Lambda}_{free}$, and $\mathbf{\Lambda}_{unknown}$ resepctively. A frontier voxel is one which lies on the boundary between explored and unexplored space – a free space voxel with a neighbouring unknown voxel. The set of frontier voxels is denoted $\mathbf{\Lambda}_{frontier}$. The volume of explored space, $\mathbf{\Lambda}_{free}$, is enclosed by a boundary $\partial S = \mathbf{\Lambda}_{occupied} + \mathbf{\Lambda}_{frontier}$. Beyond this boundary exist only unknown voxels: $\mathbf{\Lambda}_{unknown}$.

The criterion for exploration termination is the elimination of all frontier voxels, *i.e.* $\mathbf{\Lambda}_{frontier} = \emptyset$. Assuming that our sensor can resolve 3D points at a sufficient density over the surface of the world, then this equates to complete exploration, up to the resolution of the occupancy grid. The key to successful exploration is to reposition our sensor such that ∂S will be expanded outwards into unknown space, increasing $\mathbf{\Lambda}_{free}$ and reducing $\mathbf{\Lambda}_{unknown}$.

The pertinent steps in our algorithm are as follows:

- **Scan** Capture stereo images, process into depth map.
- **Integrate** Raytrace measurements into occupancy grid.
- **Boundaries** Boundary conditions applied.
- **Solve** Calculate solution to PDE.
- **Plan Path** Plan path down gradient of solution.
- **Move** Sensor is moved along path to new pose.

III. Related Work

A. Mapping

Typical mapping approaches use point clouds[1] or regular grids of 2D or 3D volumes (voxels)[2]. Point clouds represent occupied space only and cannot cope easily with noisy measurements. Regular grids are memory intensive, prohibitively so at high spatial resolutions.

To explicitly model free, occupied, and unknown space in 3D, and to do so efficiently requires moving beyond a simple grid structure. An appropriate alternative is an octree representation, see [3] and [4]. An octree is an hierarchical data structure used to partition 3D space by recursively subdiving into eight octants to a given resolution. This structure has a number of advantages over regular grids: lower memory consumption as large contiguous volumes will be represented by a single leaf node; the extent of the map need not be known at runtime, the octree can expand outwards as needed; it is trivial to obtain subdivisions at different resolutions by traversing the tree to a given depth; it can cope with noisy sensors or dynamic objects. In this paper we use OctoMap [5] which is a freely available (GNU-GPL) library for 3D occupancy grid mapping using an octree.

B. Exploration

1) Frontier-based methods: Exploration in an occupancy grid is typically accomplished using a variant of the frontier-based method proposed by [6]. This is typically framed as an information theoretic problem – where to move the sensor to maximise information gain? To do this, new camera poses are randomly hypothesised in the free space of the grid and the number of frontier cells seen from that pose (a measure of expected information) are counted. Such an exploration strategy is employed in [7] in which a voxelised 3D workspace is fully explored with a high DOF robot arm. An interesting approach is that of [8] in which exploration of new areas is balanced with improving localisation and the existing map – new poses are selected which maxmise expected uncertainty reduction.

2) Potential methods: Potential methods have a long history in the path-planning literature, see the early work of [9], [10], [11]. Assigning a high potential to the start position, and a low potential to the goal position, a path can be found by gradient descent through the resulting field, assuming appropriate boundary conditions have been set.

Little attention has been given to adapting such methods for exploration, a notable example being [12]. They demonstrate succesful exploration of a 2D environment with a sonar sensor. They make use of Dirichlet boundary conditions only, which can result in scalar potential fields with close-to-zero gradient in large-scale environments.

IV. Harmonic Functions

The exploration algorithm described in this paper relies on finding a scalar field in the free space of the map which contains no local minima and in which all streamlines through the gradient of this field terminate in boundaries free of obstacles.

Our method leverages the properties of harmonic solutions to Laplace's equation but before describing this formalism in detail it is helpful to present an informal analogy to what we are proposing. Imagine that the workspace of the robot is filled with a gas and a draft flows into unexplored regions. If the boundary between explored (known to be not occupied) and unexplored areas is all at the same pressure the flow will be (largely) perpendicular to the boundary. The far field effect of a particular boundary will be a flow which is function to the total flux crossing the boundary and the distance to it and all other boundaries. This immediately makes us think of trying to solve a steady state flow problem over the known workspace. In the absence of an analog to viscosity, this collapses down to finding a solution to Laplace's equation [1]. All such harmonic functions have properties that are advantageous to our cause: they have no local minima and hence streamlines emanating from the sensor position are guaranteed to lead us to exploration boundaries.

A. Laplace's Equation

Laplace's equation is an elliptic second-order partial differential equation over a scalar field ϕ:

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0. \tag{1}$$

Where ∇^2 is the Laplacian operator, $\nabla^2\phi = \nabla.\nabla\phi$. A function ϕ which satisfies Laplace's equation at every point, $\mathbf{x}$, in the problem domain U is known as a *harmonic function*.

$$\nabla \cdot \nabla\phi(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in U. \tag{2}$$

If U is connected – topologically – then a harmonic function satisfying (1) on U can be shown to have *no local maxima or minima*. We can see that this is true by considering the three differential terms: $\frac{\partial^2\phi}{\partial x^2}$, $\frac{\partial^2\phi}{\partial y^2}$, and $\frac{\partial^2\phi}{\partial z^2}$. At a local extremum the three terms would share the same sign – positive at a minimum, negative at a maximum – and thus not sum to zero.

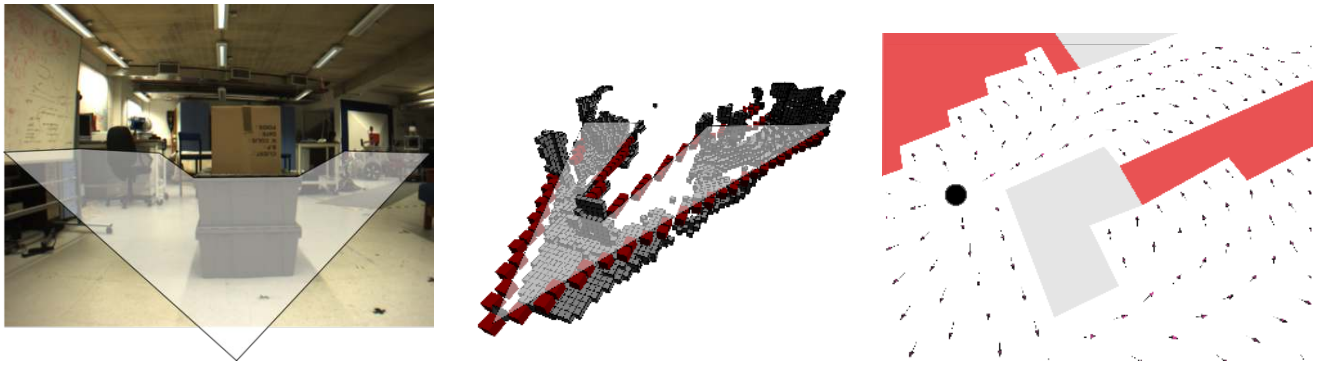
B. Setting Boundary Conditions

With no local minima, descending the gradient of a scalar field satisfying (1) is guaranteed to bring us to a domain boundary. We wish to apply boundary conditions such that solving for ϕ over our occupancy grid results in streamlines leading from the current sensor position to frontier voxels thus expanding ∂S.

Exploration boundaries should have a fixed, low value of ϕ while the current sensor position should have a fixed, high value. This will result in streamlines travelling from sensor source to frontier sinks. Dirichlet boundary conditions are used to enforce these requirements. Dirichlet boundary conditions state that the function's *value* on the domain boundary must take a specific value:

$$\phi(\mathbf{x}) = f(\mathbf{x}) \quad \forall \mathbf{x} \in \partial S_{unknown}. \tag{3}$$

[1]the inclusion of a viscosity term would further amplify the effect of known obstacles



(a) Input image from stereo camera (b) Occupancy grid and exploration boundaries (c) Harmonic function gradient, $\nabla\phi$

Fig. 2. Exploration in two dimensions: An example left image from the stereo camera is shown in (a). This is processed, producing a depth image which is used to fill an occupancy grid, shown in (b). 2D exploration is performed in a single plane of this occupancy grid – exploration boundaries are shown in red. (c) shows the gradient of a harmonic function satisfying $\nabla^2\phi = 0$ over this domain. The sensor position is a black circle, unexplored space in red, and occupied space in grey.

The gradient of ϕ should be orthogonal to $\partial S_{unknown}$ and we enforce this by setting $f(x) = 0$ in (3). This set of fixed values comprise part of an isosurface of ϕ, and thus $\nabla\phi$ is orthogonal to these boundaries as required.

A Dirichlet boundary condition is also used to force $\nabla\phi$ to point directly outwards from the sensor position by setting $f(x) = 1$ at the voxel containing the sensor.

Finally we consider obstacle boundaries. The important difference here is that an exploration boundary is potentially traversable but an obstacle boundary cannot. For $\nabla\phi$ to be parallel to obstacles we require

$$\nabla\phi \cdot \mathbf{n} = 0, \tag{4}$$

where $\mathbf{n}$ is the surface normal vector. This is equivalent to setting Neumann boundary conditions which specify that a function's *first derivative* must take a given value across boundaries:

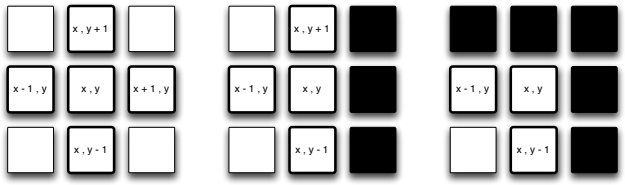
$$\nabla\phi(\mathbf{x}) = g(\mathbf{x}) \quad \forall \mathbf{x} \in \partial S_{occupied} \tag{5}$$

Setting $g(\mathbf{x}) = 0$ in (5) ensures that $\nabla\phi$ is parallel to $\partial S_{occupied}$, and (4) is satisfied.

Fig. 2 illustrates the effect of these boundary conditions. It is difficult to illustrate this in 3D and so we show a 2D slice through the occupancy grid. Figs 2(a) and 2(b) show the input image and a horizontal slice through the resulting occupancy grid. The exploration boundaries are shown in red – all voxels lying within this boundary are free space, those outside are unknown. After applying boundary conditions and calculating a harmonic solution over this grid we can display $\nabla\phi$ as a vector field (Fig. 2(c)).

C. Computing ϕ

To analytically solve Eqn 1 is not feasible for the 3D irregular domains we are dealing with, but the occupancy grid structure lends itself perfectly to the application of the finite difference method (FDM). In FDM the function ϕ is represented by a discrete value in each grid cell, and derivates are approximated by differences between neighbouring cells.



(a) (b) (c)

Fig. 3. FDM in two dimensions: The five-point stencil for the finite difference calculations is shown in (a). A couple of special cases are shown in (b) and (c). These situations require fictitious mirror values to enforce the Neumann boundary conditions. See Section IV-C for more detail.

To apply FDM to Laplace's equation a finite difference approximation to the second derivative of ϕ is required. Consider Fig. 3(a). It shows a point in a 2D grid with its four neighbours in the x and y directions, separated by h (the distance between voxel centres). The value of ϕ at this point is denoted by $\phi_{x,y}$. Taking a Taylor expansion around $\phi_{x,y}$ in the x direction leads us to the second derivative approximation:

$$\frac{\partial^2\phi}{\partial x^2} \approx \frac{\phi_{x+1,y} - 2\phi_{x,y} + \phi_{x-1,y}}{h^2} + O(h^2). \tag{6}$$

Combining Eqn (6) with the corresponding equation in y and the two dimensional version of Laplace's equation (1) we get:

$$\phi_{x,y} \approx \frac{1}{4}(\phi_{x+1,y} + \phi_{x-1,y} + \phi_{x,y+1} + \phi_{x,y-1}). \tag{7}$$

This tells us that if ϕ satisfies Laplace's equation then at any point in the problem domain, *i.e.* at the centre of any free-space voxel, then the value of ϕ at that point is simply the average of the 4 surrounding voxels.

Eqn (7) is trivially extended into three dimensions, and it is this sum over 6 neighbouring values which is used in our work:

$$\phi_{x,y,z} \approx \frac{1}{6}(\phi_{x\pm1,y,z} + \phi_{x,y\pm1,z} + \phi_{x,y,z\pm1}). \tag{8}$$

Of course this relies on having a regular grid structure to operate on - not an octree structure which contains neighbouring cells of different sizes. The conversion to a regular grid is explained later in Section VI.

Consider again Fig. 3. In (b) and (c) we see that we cannot use (7) as is – required cells such as $\phi_{x+1,y}$ lie outside of the PDE domain. The solution is to use fictitious grid points lying beyond the boundary which, if they existed, would result in the specified derivative. The zero derivative Neumann boundary conditions mean these fictitious values are simply a reflectance of the interior values and so the update equation for $\phi_{x,y}$ in Fig. 3(c) becomes:

$$\phi_{x,y} = \frac{1}{4}(2\phi_{x-1,y} + 2\phi_{x,y-1}), \tag{9}$$

and similarly in higher dimensions.

The solution to this bounded PDE is now calculated by iteratively updating each cell. At each timestep (8) is applied to every voxel. When updating $\phi_{x,y,z}$, do we use the neighbours' values from the previous timestep, or the values from this iteration if already computed? Jacobi iteration uses values from the previous timestep, but faster convergence can be achieved if we use the latest available values – the Gauss-Seidel method.

We terminate based on the maximum relative error at each timestep. The relative error for voxel x, y, z at timestep i is defined as:

$$\epsilon(x,y,z) = \left| \frac{\phi^{i}_{x,y,z} - \phi^{i-1}_{x,y,z}}{\phi^{i-1}_{x,y,z}} \right|. \tag{10}$$

Termination occurs when the largest relative error drops below a given precision threshold. A precision of $\epsilon_t = 10^{-4}$ was found to be acceptable.

D. Choosing a path

The resulting harmonic scalar function ϕ can be trivially converted into a vector field by taking its gradient, $\mathbf{v} = \nabla\phi$. We choose the streamline following the path of max flow from the current sensor pose – that is we move down the steepest gradient. Our boundary conditions ensure that any such streamline must terminate in a frontier voxel.

V. Pose Estimation and Mapping

1) Stereo Processing: To obtain depth measurements from the stereo images we use an adaptive multi-window local matching method based on the SAD5 algorithm of [13] – more details of our implementation can be found in [14].

2) 6 DoF Pose Estimation: Accurate knowledge of the depth sensor position in a global coordinate frame is vital. The relative bundle-adjustment visual odometry system of Sibley et. al.[15] gives us 6 DoF pose estimates for the stereo camera. This system gives poses of sub-centimetre accuracy over multi-kilometre scales, and we consider the output to be error-free for the purposes of this work.

VI. Integrating Measurements into the Map

As the sensor moves along the streamline to its new pose we integrate new measurements into the existing map. This description follows the OctoMap paper[5]. Given sensor measurements $z_{1:t}$ the probability of a given voxel being occupied is estimated as:

$$P(n|z_{1:t}) = \left[1 + \frac{1 - P(n|z_t)}{P(n|z_t)} \frac{1 - P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1 - P(n)}\right]^{-1} \tag{11}$$

and this can be converted to the log-odds notation when we assume a uniform prior ($P(n) = 0.5$):

$$L(n|z_{1:t}) = L(n|z_{1:t-1}) + L(n|z_t). \tag{12}$$

The integration of stereo data into the occpuancy grid uses a simple beam-based inverse sensor model. For each pixel in the depth image, a beam is cast from the camera pose to the 3D world point. Raycasting is done using a 3D variant of the Bresenham algorithm[16] and voxels traversed by the ray are updated accordingly:

$$L(n|z_t) = \begin{cases} l_{occ} & \text{- if ray ends in volume} \\ l_{free} & \text{- if ray traversed volume} \end{cases} \tag{13}$$

As suggested in [5], values of $l_{occ} = 0.85$ and $l_{free} = -0.4$ were used.

For the purposes of performing the iterative finite difference calculations we require a regular grid of voxels. To do this a bounding box is found for the octree which will contain within it all occupied and freespace voxels. A regular voxel grid is initialised with corresponding dimensions. If we are interested in only exploring a constrained part of the workspace, these dimensions can be set accordingly. A freespace voxel can easily be identified as a frontier voxel by examining its 6 immediate neighbours ($x \pm 1, y \pm 1, z \pm 1$) – if at least one is unseen space then this voxel is a frontier.

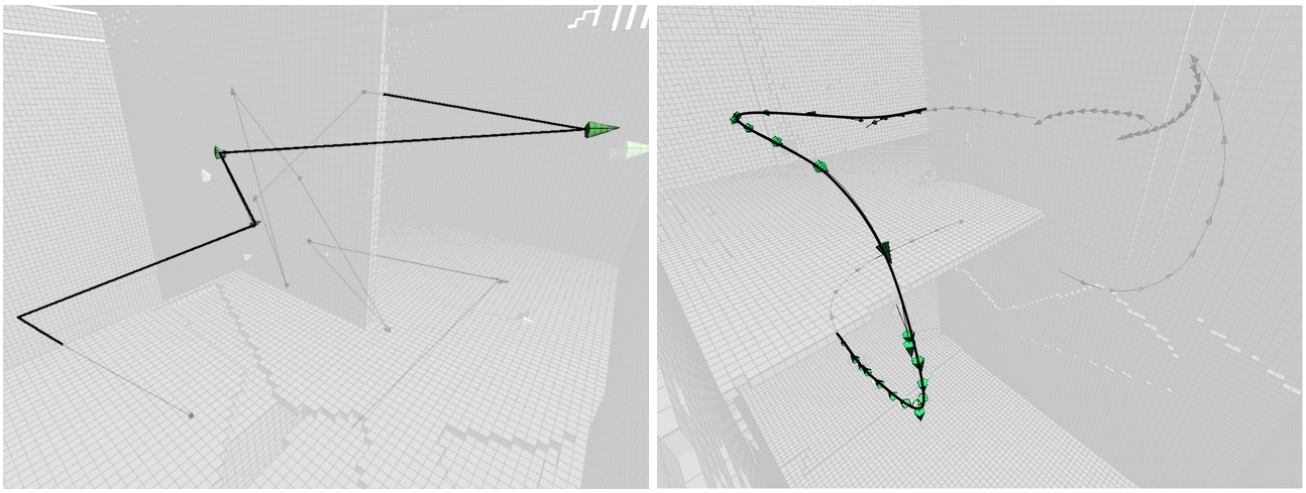
VII. Results

The exploration algorithm is repeated until exploration termination, $\mathbf{\Lambda}_{frontier} = \emptyset$. When this occurs ∂S consists entirely of occupied voxels – $\mathbf{\Lambda}_{occupied}$ forms a closed boundary around $\mathbf{\Lambda}_{free}$.

We use a simulated environment to obtain repeatable quantitive results and to rapidly compare different exploration strategies. We compared our approach with frontier based exploration described in Section III. Real world results in which a stereo camera was used to fully explore a lab environment are also presented. The final occupancy grid can be display as is, or used as a starting point for further model refinements such as surface modelling - in this case a detailed 3D point cloud is shown.

A. Simulation results

The simulated test environment consisted of a $100m^3$ bounding box with internal horizontal and vertical walls, and the highest resolution voxels were 20x20x20cm. Fig. 4 shows



(a) (b)

Fig. 4. Paths taken by exploration algorithms in a simulator. The environment explored by both algorithms is identical – it is shown from two different viewpoints in (a) and (b). (a) shows the result of the frontier based algorithm. (b) shows the path selected by the algorithm described in this paper. Note that although both algorithms resulted in 100% exploration the path in (b) is much smoother than the path in (a), and is significantly shorter in length (40.4m compared to 62.1m). The discontinuities in the path in (b) are due to the planned path being obstructed by previously unseen obstacles.

the exploration of this environment using both exploration methods. Quantitive results are shown in Table 1.

As expected both strategies exhibited complete exploration – 100% coverage of the environment indicating no remaining frontier voxels. Our method achieved this using a noticeably shorter total path length and with fewer scans, but at the cost of higher computational cost.

The approximate computations listed in Table 1 quantify and compare the computational cost of these two approaches. Integration ops is the approximate number of raytracing operations used to fuse depth measurements into the map – it scales linearly with the number of scans.

The computational cost of choosing the next camera pose is harder to compare directly due to the different methodologies involved. Recall that the frontier method hypothesises a number of new camera poses and selects that which is likely to give the highest information gain. To do this for N poses it must perform $N * W * H$ raytracing operations where the sensor image resolution is WxH pixels. In addition to this, once a pose has been chosen a path must be planned to reach it from the current pose – this is done using the A* algorithm over the occupancy grid. The cost of A* is negligible compared to the pose hypothesis step, and so the value recorded in Table 1 is a reflection of the number of hypothesised poses only.

The exploration described in this paper is more computationally expensive when it comes to planning. In our test example we have an occupancy grid with $132651(51^3)$ elements, each of which must be updated iteratively at each exploration step.

A further consideration is path *shape*. Subjectively comparing Figs 4(a) and 4(b) we see that the explorative path generated by our approach is considerably smoother than the angular shapes of the frontier based method. This is beneficial for mobile robots for which sharp turns are undesirable and is a direct consequence of the continual field approach.

	Frontier	**This Method**
Path length	62.1m	40.4m
Scans taken	130	90
Integration ops	25x10^6	18x10^6
Planning ops	51x10^6	99x10^6
Explored	100%	100%

TABLE I

QUANTITIVE COMPARISON OF EXPLORATION STRATEGIES.

B. Real world results

Fig. 5 shows the successful exploration of a lab environment using the 2D version of our exploration algorithm. The first column is the left image from the stereo camera. The middle column shows the gradient of the harmonic function and the streamline chosen for exploration. The third column shows the occupancy grid map at each stage with remaining frontiers marked in red.

A colour PointGrey Bumblebee2 stereo camera with 65° horizontal field of view was used. Images were captured at a resolution of 512x384. This experiment was run live on a laptop, manually moving the camera to follow the streamlines chosen by our algorithm. A grid resolution of $0.01m^2$ was used, covering an area roughly $15mx10m$. 66 scans were taken and Fig. 5 shows a representative sample.

VIII. CONCLUSIONS AND FUTURE WORK

We have presented an exploration and map building approach which relies on nothing but stereo imagery. The 2D version runs at real-time rates in a laboratory sized

environment, and 3D has been shown to work, but with a higher computational cost.

On concluding exploration the result is a 3D occupancy grid of the environment with freespace and obstacles explicitly marked. This can be used for path planning by a mobile robot, or combined with colour information from the source stereo images and displayed.

A GPU implementation may help alleviate some of the problems with scaling this approach to larger scales or higher resolutions. Additionally we plan on investigating strategies for coping with gaps in the stereo disparity images – poorly textured surfaces pose a problem.

IX. Acknowledgements

This work was supported by an EPSRC Industrial CASE studentship with OC Robotics.

References

[1] D. Cole and P. M. Newman, "Using laser range data for 3d SLAM in outdoor environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando Florida USA, May 2006.

[2] H. Moravec, "Robot spatial perception by stereoscopic vision and 3d evidence grids," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-96-34, September 1996.

[3] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6d slam—3d mapping outdoor environments: Research articles," *J. Field Robot.*, vol. 24, no. 8-9, pp. 699–722, 2007.

[4] N. Fairfield, G. A. Kantor, and D. Wettergreen, "Real-time slam with octree evidence grids for exploration in underwater tunnels," *Journal of Field Robotics*, 2007.

[5] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010, software available at http://octomap.sf.net/. [Online]. Available: http://octomap.sf.net/

[6] B. Yamauchi, A. Schultz, and W. Adams, "Mobile robot exploration and map-building with continuous localization," pp. 3715–3720, 1998.

[7] L. Freda, G. Oriolo, and F. Vecchioli, "Sensor-based exploration for general robotic systems," sep. 2008, pp. 2157 –2164.

[8] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters," in *IN RSS*, 2005, pp. 65–72.

[9] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Rob. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[10] C. I. Connolly and R. A. Grupen, "On the applications of harmonic functions to robotics," *Journal of Robotic Systems*, vol. 10, pp. 931–946, 1993.

[11] E. S. Conkur and R. Buckingham, "Manoeuvring highly redundant manipulators," *Robotica*, vol. 15, no. 4, pp. 435–447, 1997.

[12] E. P. e Silva, P. M. Engel, M. Trevisan, and M. A. P. Idiart, "Exploration method using harmonic functions," *Robotics and Autonomous Systems*, vol. 40, no. 1, pp. 25 – 42, 2002.

[13] H. Hirschmüller, P. R. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 229–246, 2002.

[14] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schröter, L. Murphy, W. Churchill, D. Cole, and I. Reid, "Navigating, recognising and describing urban spaces with vision and laser," *The International Journal of Robotics Research*, 2009.

[15] G. Sibley, C. Mei, I. Reid, and P. Newman, "Adaptive relative bundle adjustment," in *Robotics Science and Systems (RSS)*, Seattle, USA, June 2009.

[16] J. Amanatides and A. Woo, "A fast voxel traversal algorithm for ray tracing," in *In Eurographics 87*, 1987, pp. 3–10.

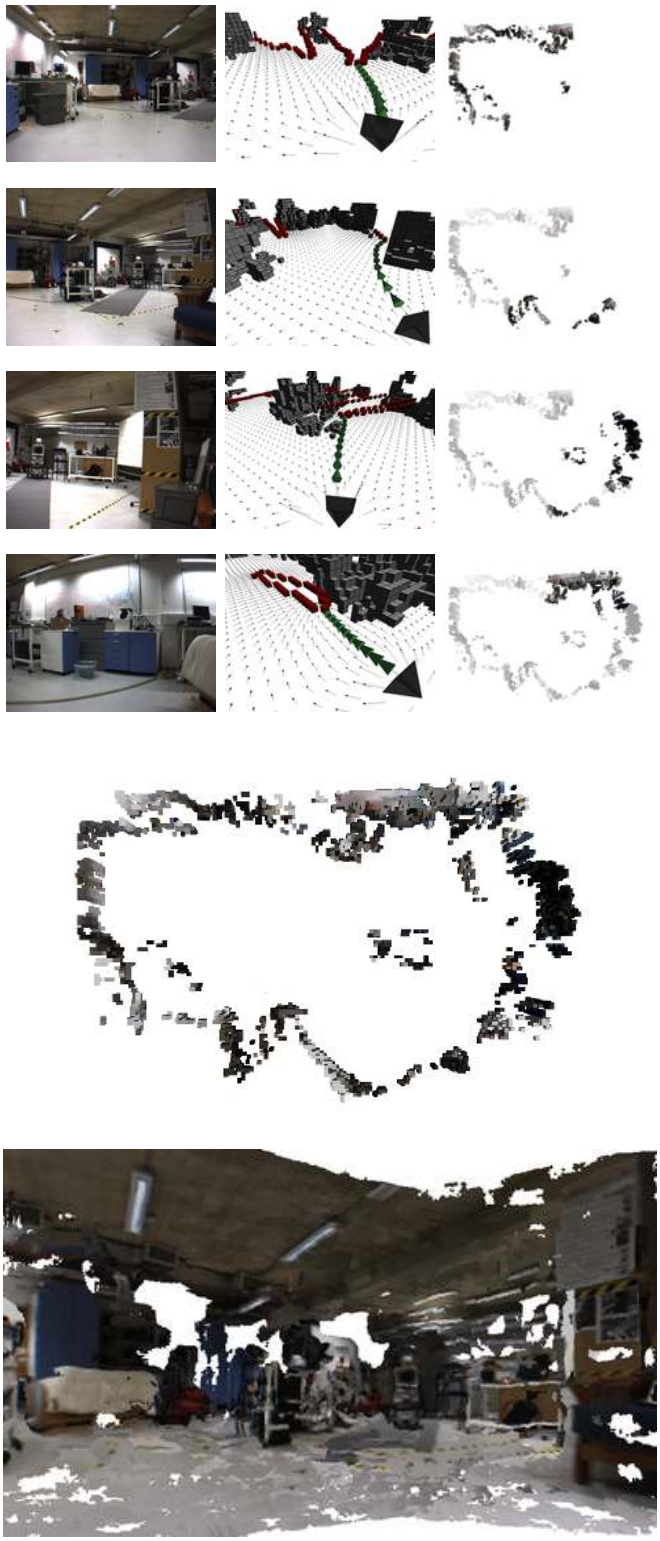


Fig. 5. Representative images from a sequence of 66 scans taken while exploring a lab environment. The first column shows the left image from the stereo camera. The second column shows the gradient of the harmonic function at that pose, and the streamline chosen for exploration. The final column shows the updating map of the world – dark points are those seen from that particular pose. The bottom two images are of the complete map after exploration has terminated, and a point cloud of the environment from stereo data collected during the scans.