Lightweight SLAM and Navigation with a Multi-Camera Rig

Gerardo Carrera Adrien Angeli Andrew J. Davison
Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK

***Abstract*— An interesting recent branch of SLAM algorithms using vision has taken an appealing approach which can be characterised as simple, robust and lightweight when compared to the more established and complex geometrical methods. These lightweight approaches typically comprise mechanical odometry or simple visual odometry for local motion estimation; appearance-based loop closure detection using either whole image statistics or invariant feature matching; and some type of efficient pose graph relaxation. However, such algorithms have so far been proven only as localisation systems, since they have not offered the semantic demarcation of free space and obstacles necessary to guide fully autonomous navigation and exploration. In this paper we investigate how to apply and augment such approaches with other lightweight techniques to permit fully autonomous navigation and exploration around a large and complicated room environment. Our approach uses only odometry and visual sensing, the latter being provided by a rig of multiple standard cameras without the need for precise inter-camera calibration. A particular focus of our work is to investigate the camera configurations which are most valuable to permit the capabilities needed by autonomous navigation, and our solution neatly assigns each camera a well-defined specialist role.**

***Index Terms*— Computer Vision, Robotics, SLAM, Multi-Camera Rig.**

I. Introduction

Computer vision is an increasingly appealing sensing modality for mobile robotics, and a number of successful SLAM systems involving vision as the primary outward-looking sensor have been recently presented. In particular, besides the more standard feature-based reconstruction approaches, there have been several systems which have solved the visual SLAM problem using much 'simpler' lightweight techniques which combine local trajectory estimation (via either wheel or simple visual odometry), visual place recognition, and pose graph optimisation (e.g. [13]). However, in either paradigm few systems have gone further than tackling localisation, and few real visual SLAM systems have been proven in autonomous navigation.

In this paper we present a method based on a combination of lightweight vision techniques which permits robust, automatic and repeatable mapping and navigation around a large room. In this scenario, there are several demands placed on the robot's sensing systems:

- Local trajectory estimation.
- Place recognition to detect re-visits (loop closures) and permit global map optimisation.
- Detection and avoidance of obstacles in the robot's close proximity.
- Global free-space mapping for path planning.

Here we describe a novel combination of lightweight methods to provide all of these capabilities using odometry together with multi-camera visual sensing. Our use of a rig of multiple standard cameras is based on the observation that while a wide field of view is often desirable for SLAM and relocalisation, the other requirements placed on a vision system when the goal is full autonomous navigation are not easily satisfied by a single omnidirectional camera. These cameras often have a limited and low resolution view of downward angles below the horizontal necessary for free space and obstacle detection, which, when looking more to the future, could be a limitation for tasks like object recognition or human interaction.

Pre-calibrated multi-camera rigs in a single unit which offer good resolution across a wide field of view are available, such as Point Grey's Ladybug, but they are expensive, as well as inflexible since they cannot be reconfigured. The advantage of multiple specialised sensors in high performance autonomous systems has been proven in robots such as the DARPA Grand Challenge vehicles or Willow Garage's PR2. In our work, we choose to use just cameras rather than other outward-looking sensor types, but retain the idea of multiple cameras mounted in a weakly coupled way and which provide specific functions as part of a whole navigation solution, without requiring a tedious inter-camera calibration procedure.

We demonstrate automatic localisation and mapping, and autonomous navigation in a goal-directed scenario where the robot is able to move repeatably between any pair of points indicated in the map. Further, we demonstrate full autonomous exploration; the robot is dropped into the room with no knowledge or a large cluttered room and is able to explore autonomously to build a consistent map of the whole area suitable for autonomous navigation.

II. Related Work

A. Lightweight Vision-Based SLAM

The lightweight approaches we study in this work have at their core a coarse metric-topological environment model, but they can still enable accurate and autonomous navigation.

The single most important function provided by vision within such a system boils down to image retrieval, where the most recent image is compared against all previous locations in the environment model. The use of colour and texture information encoded in global histograms for image representation has proven successful for topological robot localisation [16, 18]. More recently, purely appearance-based topological place recognition approaches based on the "Bag of words" paradigm and inspried by information retrieval techniques have been proposed [2, 4]. These are however more computationally demanding, as they require expensive feature descriptor calculations and matching, and we consider them unnecessary in the context of our robot's local navigation problem.

Although a rig of multiple standard cameras is seemingly good alternative to a single omnidirectional device, the use of such rig for SLAM has been subject of little research, presumably because of the difficulty of extrinsic calibration. In our previous work [3] we showed that autonomous external calibration of a camera rig with no overlapping views was indeed possible, and [10] demonstrated SLAM using an eight camera rig. While such geometric work on fused multi-camera SLAM will doubtless continue, in the present paper we aim to use each of the cameras for one or more specialised tasks without the need for global extrinsic calibration.

B. Integrated Visual SLAM and Autonomous Navigation

There have only been few visual SLAM systems using standard cameras which enable fully autonomous navigation. Davison and Murray's early visual SLAM system based on fixating active stereo [6] was used for real-time position-based navigation, but the feature map generated was too sparse to permit reliable reasoning about free and occupied areas of space. With the advent of feature-based robot SLAM systems with much increased feature density, there have recently been some attempts at performing free-space mapping based on semi-dense point clouds. Notably, systems like [15] offer good possibilities for free-space detection (at least in highly textured areas with many feature points), and enable autonomous navigation and exploration.

C. 2D Free Space Mapping using Vision

On the assumption that our robot moves on a ground plane, 2D free space mapping is critical to allow obstacle avoidance and path planning.

This problem has been studied not only for robot navigation but also in road detection for vehicles to aid autonomous driving. Some approaches attempt to define the drivable area either using offline machine learning techniques such as Support Vector Machines [17] or by a combination of geometry information and offline learning [1]. Such techniques cannot however be directly transposed to the indoor 2D free-space detection problem, because in this context there is no such thing as a general geometric pattern for the boundary of the drivable area that can be retrieved in each image based on (potentially learned) a priori information.

Successful methods for obstacle avoidance and free-space mapping for mobile robot navigation rely solely on the use of colour information [9], or infer a planar homography mapping image pixels to 2D floor coordinates under a ground plane assumption [19, 11]. The authors of [12] propose to make use of multiple cues to calculate horizontal lines defining the boundary between floor and walls, which is well suited to corridors, but presumably not adapted in case of more complicated structure.

III. METHOD

We divide our whole framework into five main sections which are all interconnected:

- Rig camera placement.
- Map representation and loop-closure detection.
- Global map relaxation.
- Free space mapping.
- Autonomous navigation, obstacle avoidance and path planning.

A. Rig Camera Placement

Our robot's vision capability is to be provided by a rig of up to four standard cameras, each with a lens offering approximately 80° horizontal field of view. There are many possible choices for the configuration of these cameras, since they must support the various tasks required by loop closure detection, free-space mapping and exploration, and we have examined the trade-offs of different set-ups. Notably, the final configuration chosen is adapted to the characteristics of the office environment used, where movements can be quite restricted and the distance to the objects relatively short (e.g., when traversing narrow corridors between two desks).

In principle, an extremely wide field of view, up to the maximum full cylindrical field of view offered by a single omnidirectional camera, is well suited to relocalisation: not only does this enable the capture of a good variety of appearance data to act as a fingerprint for a location, but it also permits recognition of previously-visited places independent of the orientation of the robot. However, when this wide field of view is provided not by a single omnidirectional camera but by a rig, we found that additional difficulties arose.

We experimented extensively with histogram-based place recognition (see the next section) with an ad-hoc four-camera rig designed such that the cameras were mounted horizontally with maximum angular spacing (i.e. at the corners of a square). However, the performance in recognising locations with different robot orientations was disappointing. This was partly due to the fact that the four cameras used left gaps or 'blind spots' in the cylindrical field of view which would not necessarily align when the robot had made a rotation. Another significant point was that the actual distance between the camera centres on the robot was often significant compared to the distance to objects and furniture in the indoor scene, so unmodelled parallax effects came into play.

We found that a good pragmatic solution to camera placement for loop closure detection in an indoor scene is to have one horizontal camera facing forwards, and one backwards (see Figure 1). This configuration is able to detect the vast majority of loop closure events the robot will encounter, because in restricted spaces, when a robot revisits a place it is very likely to do it either moving in the same or exactly opposite direction to before. Yes, there will be possible loop closure events sometimes missed when the robot crosses a previous path at right angles. But in fact, such a crossing might well be difficult to reliably detect in any case, as it may correspond to a wide open area where several significantly distant positions at the centre of it have similar appearance, making recognition ambiguous and localisation inaccurate.

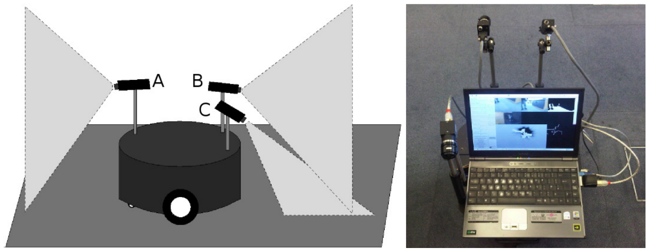

Fig. 1. Robot camera configuration consisting of a three camera rig where camera C is used for obstacle avoidance and free space mapping and the cameras A and B are used for loop closure detection.

Together with the front/back camera combination for loop closure detection, we added a third camera pointing down at the ground in front of the robot for free space mapping. As detailed later, we experimented with the downward angle of this camera where there is a trade-off between immediate, local obstacle detection capability and a more forward view suitable for planning. There is an interesting feedback in play here between the image matching for relocalisation required by SLAM and free space mapping which permits autonomous robot guidance. A free space detection solution which is working well will enable the robot, when exploring or revisiting, to navigate in a precise way by for instance moving repeatably half-way between gaps or along corridor passages. This makes loop closure detection easier, since the robot is likely to very precisely revisit locations.

B. Map representation and Loop Closure Detection

Our approach does not assume any prior knowledge about the environment except that is traversable by a wheeled robot and that its visual appearance is descriptive enough.

As the robot autonomously explores or is being driven through an environment, it builds a topological or graph representation of the area. A topological map is a very practical and desirable model for navigation, since it imposes a discrete structure on a continuous space, and because it easily enables the use of different low level (graph relaxation) and high level (path planning) algorithms.

In this undirected graph $G = (V, E)$, a vertex $V = (I_t^N, X_t)$ represents a physical location in the environment which stores all the images I from N cameras of the rig at time t as well as the global position of the robot $X_t = (x_t, y_t, \theta_t)$ (2D position plus heading direction). Each edge E_t in the graph stores the relative 2D transformation between nodes X_t and X_{t+1}. A new vertex is initialised when the distance traveled from the previous position is greater than some threshold β (using the internal odometry of the robot), or when the dissimilarity between consecutive images is above a threshold α.

Our approach for image comparison relies on a global descriptor implemented in the form of a 1D grey-level histogram. Such minimalistic single-signature place characterisation enables reasonable discrimination, while on the other hand only requiring frugal computational resources. The signature of a location is obtained by sticking the images of both forward and backward cameras next to each other into a single mosaic which serves as the support for the calculation of the descriptor. Different methods for comparison were tested as well as multi-dimension histograms including cues such as gradients, color or intensity, obtaining the best results using EDM [14] over 1D histograms of grey intensities.

Loop-closure detection is achieved by comparing the most recent location signature against all the previously visited locations. Thanks to the compactness and simplicity of place characterisation, such an exhaustive retrieval procedure can be executed efficiently. Once a candidate for loop closure is found, time consistency is ensured to cope with perceptual aliasing. To this end, we perform a comparison of the appearance of 7 locations $L_{\text{recent}} = \{V_{i-3}, \ldots, V_i, \ldots, V_{i+3}\}$ centred in time around the location of interest V_i, with 7 locations $L_{\text{previous}} = \{V_{j-3}, \ldots, V_j, \ldots, V_{j+3}\}$ similarly sampled around the potentially loop-closing location V_j. This yields a 7×7 matrix $C = \sum_{i,j=1,\ldots,7} C_{ij}$, where each entry corresponds to the distance (in appearance-space) between locations $V_i \in L_{\text{recent}}$ and $V_j \in L_{\text{previous}}$. Note that the procedure has to be postponed until the locations V_{i+1}, V_{i+2} and V_{i+3} are visited and added to the map.

Asserting the time-consistency of a potential loop-closure requires an evaluation of the entries of C which takes care of the heading direction of the robot (see Figure 2). However, because of the configuration of the rig retained in our approach, we only need to distinguish between situations where the relative orientation from one passing to another is either $0\,\text{deg}$ or $180\,\text{deg}$. Therefore, we review the scores of all the elements on both diagonals of C, and only if they are all below some threshold for one diagonal the loop-closure is accepted. This is enforcing the consistency of the appearance over neighbouring locations in time around both the location of interest and the potential loop-closing location in such a way that the relative orientation of the robot between the 2 passings is properly taken into consideration.

C. Graph Relaxation and Map Optimization

Due to the topological nature of our map representation we can optimize the poses by minimising the error between robot positions every time a loop closure is found. Every time a new graph relaxation is applied the map becomes more consistent to the real metric map and therefore becomes usable for navigation and path planning. In our approach we used TORO [8] which provides a highly efficient gradient descent-based error minimisation for constrained graphs.

In order to enable accurate obstacle avoidance and path planning, a global free space map M of the environment is being built incrementally as the robot navigates (see Section III-D for details about free space detection). To this end, we associate to every keyframe of the robot trajectory a relative local free space map M_t, every vertex of the graph G being now represented as $V_t = (I_t^N, X_t, M_t)$, which is a simple 2D occupancy grid with a resolution of $1cm^2$ per cell and whose origin is anchored according to the position and orientation of X_t. To ensure the consistency between topological and global

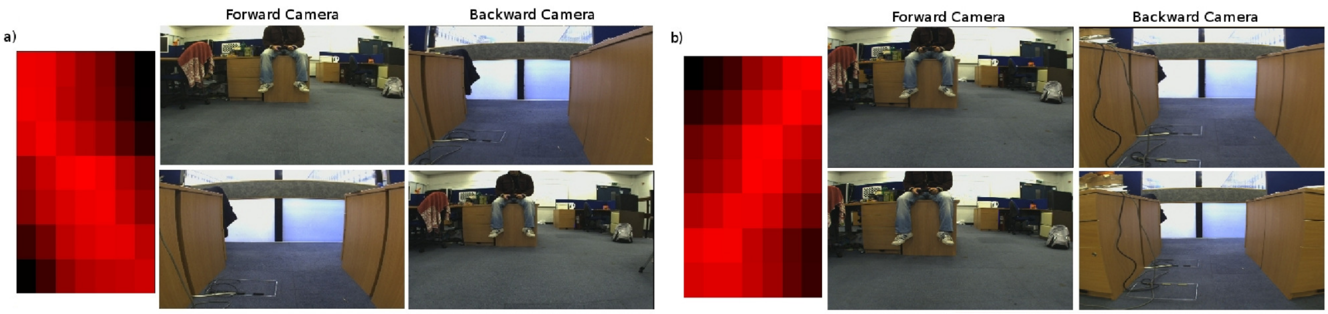

Fig. 2. Loop Closures using two cameras: forward and backward. a) represents a loop closure found with a difference in robot orientation of 180 deg with respect to the matching node. b) represents a loop closure found with the same robot orientation as the matching node. The top row in figures a) and b) indicates the current frame and the bottom row the matching node or keyframe. The single image at the left of figures a) and b) shows the matrix consistency check, with the bright red indicating strong matching along either the forward or backward diagonal.

free space maps, M is updated based on the optimised vertex positions whenever the topological graph is relaxed.

It is true that with our featureless technique for loop-closure detection, an accurate position of the robot is not obtained right away. Yet, this is important to impose precision which improves the graph of poses after relaxation. Therefore, to obtain a good relative position estimate at loop-closure between current X_t and past X_p poses, we approximate the displacement $\Delta P = (\Delta x, \Delta y, \Delta\theta)$ between them by aligning the contours of their respective local free space maps M_t and M_p, which can be simply done by solving a cost function minimising the distance between points in those contours:

$$F^{C_1 C_2} = \min(\sum_{i,j} \mathrm{dist}(P_i^{C_1}, P_j^{C_2})), \tag{1}$$

where the function $\mathrm{dist}(\cdot,\cdot)$ is evaluated obtaining the 2D Euclidian distance between all the points in $P_i^{C_1}$ and $P_j^{C_2}$. When the robot revisits a location with an opposite direction to the one of the previous passing, it is difficult to match the free space maps, as they do not overlap very much. In such situation though, we can still approximate the relative orientation between the two views by calculating the mean horizontal optical flow between the corresponding images (obtained by the loop closure detection) and this relative orientation is used in the loop closure constraint.

D. Free Space Mapping

We have developed an accurate algorithm which incorporates geometry and colour information. Our solution assumes that the robot is moving on a plane which is the same over the whole room, and that the floor is made of large regions with homogeneous colour.

Under the floor planarity assumption, it is possible to map the image pixels of a downward looking camera to the 2D cells of a local free space map of the visible floor in front of the robot. This transformation H_f is a homography which is calibrated once each time the is fixed to the robot by the simple procedure of matching four artificial landmarks on the floor with the corresponding pixel positions in the image.

The inverse mapping H_f^{-1} can be employed to retrieve from the image the RGB value X_i associated to any floor cell i visible in front of the robot, so as to determine if this cell is free of obstacles or not. This is done here by calculating the log-likelihood ratio of occupancy, as follows (statistical dependency on the model is omitted for simplicity):

$$\ln \frac{P(C_i = F|X_i)}{P(C_i = O|X_i)} = \ln \frac{P(X_i|C_i = F)}{P(X_i|C_i = O)} + \ln \frac{P(C_i = F)}{P(C_i = O)} \tag{2}$$

where C_i is the class of cell i (F for "floor", O for "obstacle"). We assume constant empirically fixed priors, and a uniform "obstacle" likelihood model, while the "floor" likelihood model is a mixture of L gaussians similar to the one used by Thrun *et al.* [5] for road detection:

$$P(X_i|C_i = F) =$$

$$\frac{1}{\sum_{j=1}^{L} w_j} \sum_{j=1}^{L} w_j \frac{1}{(2\pi)^{\frac{3}{2}} \sqrt{|\Sigma_j|}} \exp^{-0.5(X_i-\mu_j)'\Sigma_j^{-1}(X_i-\mu_j)} \tag{3}$$

where μ_j, Σ_j, w_j respectively are the mean, covariance matrix and mixing coefficient parametrising gaussian i. When the occupancy ratios have been calculated for every cell, the model parameters are updated according to the procedure proposed in [5], using only those cells whose ratio is above some threshold. Initially, the gaussians are learned in the very first frame using only a small region at the bottom-centre of the image (that is assuming that at startup, the corresponding area on the floor is free of obstacles and is a good representative sample of the overall appearance of the floor).

For better robustness and more efficient planning during navigation, the local occupancy grids corresponding to several consecutive robot poses are fused together (see Figure 3): not only does this provide a medium-sized free space map around the robot which is more suited to navigation, but also makes it possible to filter out incorrect obstacle detections due to inaccurate probability scores in the presence of noise in the image or illumination changes in the scene. Similarly, as already mentioned, a global free space map M of the explored environment can be computed at anytime by fusing all the individual local occupancy grids (see Figures 5 and 6).

Fig. 3. The image on the left shows part of a map which has been incrementally built, the region inside the marked area is the current frame being mapped which corresponds to the image on the right.

E. Autonomous Navigation and Exploration

We use the Dynamic Window Approach (DWA) [7] to plan smooth motions of the robot around nearby obstacles. Having estimated the free space around the robot based on a local window of a fixed number of recent keyframes, the local occupancy map is binarized to obtain boundaries between free-space, obstacles and unexplored areas. Every cell detected as an obstacle boundary in our map is used in DWA to plan the next velocity commands. DWA plans the best route from a current robot position to a target location, avoiding obstacles, and therefore requires the definition of robot goals. We have considered it outside the scope of this paper to investigate ideal exploration strategies, finding that a heuristic technique was sufficient in our application. A goal is randomly selected relative to the robot position and around a square window of $3m \times 3m$. If the robot has spent much time around the same area then a goal is selected further away.

It is also important to obtain a balance between mapping unexplored areas and obtaining accurate maps. Every time the robot has mapped an area completely it revisits previous places within the mapped area to find potential loop closures. By doing this we try to correct drifts in the odometry and also every cell is corrected according to the new robot position.

IV. Experiments and Results

We have developed our experiments in an office of size $10\times$ 10m using a Pioneer robot platform. As can be seen in figure 4, this environment represents a challenging scenario with strong perceptual aliasing and multiple narrow spaces (88cm), making both localisation and autonomous navigation difficult. The quality of our approach is demonstrated by performing manual and autonomous navigation with incremental and real-time free space mapping and loop closure detection.

Our first experiments consider the effect of loop closure detection over map precision: it is well known that odometry-based robot position estimation will continuously drift over time, until a loop-closure is detected and the inconsistency is compensated for. To safely navigate, it is therefore extremely important to correct the map accordingly every time as possible. As can be observed in the top left image of Figure 5,

Fig. 4. Office environment with strong perceptual aliasing and challenging navigable spaces.

Ground truth comparison (cm)			
Robot Pos	Ground truth	Robot Pos	Ground truth
(182, 0)	(180, 0)	(96, 445)	(90, 450)
(212, 145)	(210, 150)	(-246, 487)	(-240, 480)
(425, 143)	(420, 140)	(89, -325)	(90, -330)
(512, -271)	(510, -270)	(-280, -326)	(-270, -330)

TABLE I
Evaluation of Localisation Accuracy.

a constructed map using only odometry information with no loop-closure detection is highly inaccurate, and it would be almost impossible to navigate autonomously or to use the map for further high-level tasks. When loop-closures are detected with only one camera (top right image), then it can be observed that both the robot trajectory and the map are significantly more accurate. When using two cameras, the number of loop closure detections increases, and so the map becomes even more accurate (see Figure 5, bottom).

We have investigated the impact of the downward looking angle of the camera used for free space detection on the quality of the map (Figure 5, bottom images). We have found that if the camera is oriented such that it is only covering a very restricted region (i.e., within 20cm to 110cm) in front of the robot, then, the obtained map is very detailed, enabling very accurate obstacle localisation. However, such very downward looking angle penalises motion planning, as it only provides very local information, with obstacles being detected very late, only when the robot is very close to them (see the bottom right part of the figure). When the camera is covering a larger region (i.e., within 60cm to 450cm) in front of the robot, further away obstacles can be detected, enabling more efficient motion planning, over a wider time window, leading to smoother robot trajectories (bottom left of the figure).

To measure the metric quality of our free space map, we have picked 8 random robot positions distributed over the whole room, and compared their coordinates in the map with ground truth obtained by manually measuring the coordinates of these points on the floor. This comparison, presented in Table I, demonstrates the accuracy of our solution, with a mean 2D error of 6.39cm, and a max error of 10.77cm.

With our system the robot was able to successfully achieve several autonomous navigation runs of approximatively 20min. Figure 6 shows examples of global free space maps, providing

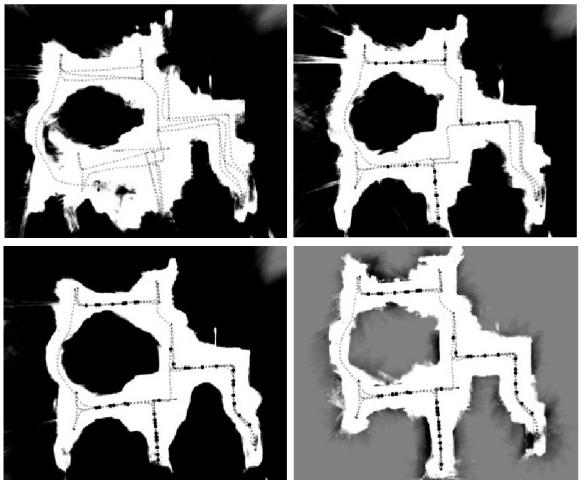

Fig. 5. Free Space Map built by driving the robot manually (700 keyframes). Top left: map without loop closures, top right: map with loop closures coming from the camera facing forward, bottom left: map with loop closures from both cameras, bottom right: map built using a camera facing to the floor.

a comparison of the two downward looking angles already used earlier, again proving the superior accuracy of the proximal sensing configuration. The purpose of this experiment is to demonstrate the reliability of our complete solution comprising simultaneous localisation, mapping, free space detection and navigation: the exploration strategy here is deliberately simplistic, and used only as a proof of applicability of our solution (more advanced policies would certainly result in more efficient exploration).

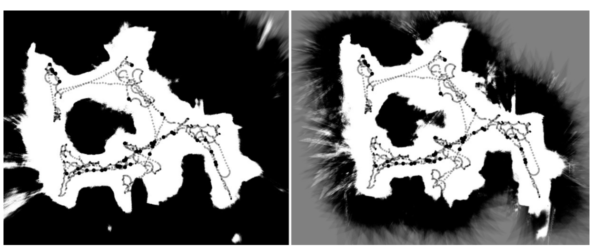

Fig. 6. Free Space Map built autonomously using 2500 keyframes

V. Conclusions

We have shown that lightweight vision-based techniques, previously shown to be effective for surprisingly high performance localisation, can be augmented to be similarly effective for fully autonomous robot navigation. Our system puts together odometry-based trajectory estimation, front/back vision-based loop closure detection, free space identification from a local obstacle detection camera and a forward-looking camera, and local and global occupancy mapping. Our approach relies on an ad-hoc rig of multiple standard cameras, and we have investigated the role each camera should effectively play for the best performance, but in future work it may be worth considering taking the specialisation route further and creating a system with heterogeneous camera types with even more elaborated configurations and roles.

Acknowledgments

This work was supported by a CONACYT scholarship from the Mexican government to G. Carrera and European Research Council Starting Grant 210346.

References

[1] Y. Alon, A. Ferencz, and A. Shashua. Off-road path following using region classification and geometric projection constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 689–696. IEEE, 2006.

[2] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Real-time visual loop-closure detection. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[3] G. Carrera, A. Angeli, and A. J. Davison. SLAM-based automatic extrinsic calibration of a multi-camera rig. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[4] M. Cummins and P. Newman. Highly scalable appearance-only SLAM — FAB-MAP 2.0. In *Proceedings of Robotics: Science and Systems (RSS)*, 2009.

[5] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In *Proc. of Robotics: Science and Systems (RSS)*, 2006.

[6] A. J. Davison and D. W. Murray. Mobile robot localisation using active vision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 1998.

[7] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33, 1997.

[8] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems (RSS)*, 2007.

[9] J. Hoffmann, M. Jungel, and M. Lotzsch. A vision based system for goal-directed obstacle avoidance. In *ROBOCUP2004 SYMPOSIUM, Instituto Superior Técnico, Lisboa, Portugal*, 2004.

[10] M. Kaess and F. Dellaert. Probabilistic structure matching for visual SLAM with a multi-camera rig. *Computer Vision and Image Understanding (CVIU)*, 114:286–296, Feb 2010.

[11] Y. Kim and H. Kim. Layered ground floor detection for vision-based mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 13–18. IEEE, 2004.

[12] Y. Li and S.T. Birchfield. Image-Based Segmentation of Indoor Corridor Floors for a Mobile Robot. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2010.

[13] M. Milford and G. Wyeth. Persistent navigation and mapping using a biologically inspired SLAM system. *International Journal of Robotics Research (IJRR)*, 29(9):1131–1153, 2009.

[14] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the 1998 IEEE International Conference on Computer Vision*, pages pp. 59–66, 1998.

[15] R. Sim and J. J. Little. Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2006.

[16] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[17] Q. Wu, W. Zhang, T. Chen, V. Kumar, et al. Camera-based clear path detection. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1874–1877. IEEE, 2010.

[18] C. Zhou, Y. Wei, and T. Tan. Mobile robot self-localization based on global visual appearance features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 1271–1276, 2003.

[19] J. Zhou and B. Li. Robust ground plane detection with normalized homography in monocular sequences from a robot platform. In *Image Processing, 2006 IEEE International Conference on*, pages 3017–3020. IEEE, 2007.