MULTIPLE HYPOTHESES DATA ASSOCIATION PROPAGATION FOR ROBUST MONOCULAR-BASED SLAM ALGORITHMS

Mauricio Soto-Alvarez

Pattern Analysis & Computer Vision
Istituto Italiano di Tecnologia (IIT)
via Morego 30, 16163, Genoa, Italy

Petri Honkamaa

VTT Technical Research Centre of Finland
Espoo, Finland

ABSTRACT

Data Association is probably the most important step of every monocular Simultaneous Localization and Mapping (SLAM) algorithm because it provides the basic information to the estimation module, independently on the estimation algorithm of choice. Although important, it is also a difficult task because the analytic solution is NP-Hard. The usual approximation is obtaining only one data association hypothesis per frame which affects the robustness of the result [1][2][3][4][5]. In this paper, a data association approach is presented, where multiple hypotheses are propagated between frames using a probabilistic framework. Experimental results, using real and synthetic data, show that the proposed algorithm produces promising results with respect to other state of the art methods.

Index Terms— SLAM, data association

1. INTRODUCTION

SLAM algorithms concurrently solve two interrelated problems: what is the current pose of the sensor (localization) and what does the environment looks like (mapping). Solving these two problems is difficult because localization depends on mapping and mapping depends on localization, therefore, errors in any of these steps lead to wrong map estimation and failure. SLAM is important because it is the core algorithm in many applications such as augmented reality, autonomous navigation in robots and aerial vehicles, among others. In SLAM applications, the most commonly used sensors are range-lasers, cameras and structured light sensors (e.g. Kinect). Nevertheless, monocular-based localization and mapping is still a very active topic of research because cameras are able to provide good resolution while being inexpensive portable devices. For example, the camera on a mobile phone can be turned into an augmented reality port where additional information can be displayed to the user.

Monocular SLAM algorithms generally rely on four modules: feature detection, data association, localization and mapping. From these modules, data associations is one of the most important because it provides the essential information for estimating the current pose of the sensor (localization) and it defines which features can be added as new landmarks (mapping). It is also important because once data association has been decided, it defines a lower bound on the localization error, i.e. localization estimation error grows when there are data association mistakes independently of the localization algorithm that it is being used.

In this paper, a data association algorithm for monocular is proposed. It is called *Multiple Hypotheses Data Association* (MHDA). The main novelty of this paper is showing how the performance of localization and mapping can be improved by propagating multiple data association hypotheses between frames instead of propagating only one strong hypothesis [1][2][3][4][5].

This paper is structured as follows: the state of the art and the algorithm are explained in sections 2 and 3. Then experiments and results are shown in section 4. Finally, some conclusions and final remarks are presented in section 5

2. STATE OF THE ART

Data association is the name for the module that deals with the problem of associating, at each time step, landmarks from our map to observed features from the image. Data association is sometimes called matching or finding correspondences. In principle, and without any prior information, every feature detected in each image is a possible candidate for associating it with each of the landmarks in the map. The simplest algorithms in data association are based on single candidate *Nearest Neighbor* approaches. For example, A.Davison et al. pioneer work on monocular SLAM[6] used an *Appearance Nearest Neighbor* approach based on Normalized Cross-Correlation (NCC) patch distance. Later, J.Civiera et al.[3] improved on Davisons work by filtering the output of the appearance Nearest Neighbor (NN) with spatial information in the so called *1-point-RANSAC*. The method of Civiera works in three stages: (i) one correspondence is randomly sampled from the set of probable matches and the pose is updated. Based on the new pose, the rest of the matches are classified into inliers and outliers. (ii) The pose is refined using the information of all inliers and the outliers set is searched again for correspondences that comply with the new pose. This process is repeated several times for different random correspondences in a RANSAC fashion. (iii) Finally, the data association hypothesis set with highest number of inlier correspondences and lowest jointly spatial error is chosen.

Single candidate approaches are appealing in real-time setups because they are computationally inexpensive. Nevertheless, they do not provide robust results because spatially close candidates may be incorrect due to occlusions and pose estimation errors. Conversely, there are false positives in appearance matches due to similar texture in the image.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

Algorithm	Spatial Coherence	Appearance Coherence	Multiple Candidates	Intra-Frame Multip. Hypoth.	**Inter-Frame Multip. Hypoth.**
Spatial Nearest Neighbor	x				
Appearance Nearest Neighbor[6]		x			
1-point-RANSAC[3]	x	x		x	
Active Search[2]	x	x			
Active Matching[4]	x	x	x	x	
Scalable Active Matching[5]	x	x	x	x	
Joint Compatibility Branch and Bound (JCBB)[1]	x		x	x	
MHDA (**proposed**)	x	x	x	x	x

Table 1. Characteristics of different Data Association algorithms in literature. The first two columns describe which feature distance are exploited by the algorithm. The upper rows show approaches that use only one candidate matching per feature while lower rows show algorithms that use multiple candidate hypotheses per feature. The bottom row shows the characteristics of the proposed algorithm MHDA.

A.Davison also proposed a clever single candidate approach called *Active Search* [2] based on Mutual Information. He showed that it is possible to pre-calculate what is the amount of information, with respect to the pose of the sensor, that one could gained by measuring each of the landmarks. The gain in information is in bits. He also defined a metric called *measurement efficiency* which is the ratio between the mutual information associated to a landmark and the work that one has to do in order to search this landmark according to its uncertainty. Active Search is an iterative algorithm that works as follows: It begins by searching, using patch appearance, the landmark associated with the largest measurement efficiency coefficient in a Nearest Neighbor fashion. Associating the chosen landmark to a likely patch candidate immediately reduces the uncertainty on the position of other correlated features, and therefore the area required to search them. Active Search requires less image processing and has better performance than other single candidate approaches. However, it is greatly affected by false positives (repetitive texture) due to the use of appearance NN approach.

M.Chli and A.Davison presented and enhanced version of the Active Search algorithm called *Active Matching*[4]. The main improvements over their previous work were to employ multiple candidates matching for each landmark and to propagate multiple hypotheses within one frame. This new approach was able to better handle ambiguities that arise within a frame while using a computationally efficient algorithm based on Gaussian Mixture Model.

In the multiple candidate group, one can find the important work of J. Neira and J.D. Tardós called *Joint Compatibility Branch and Bound (JCBB)*[1] which has been widely adopted in the SLAM community. JCBB casts the data association problem into a tree search problem based solely on spatial coherence. This algorithm does work very well in practice but it tends to be computationally expensive (beyond the real time usage). JCBB is a conservative approach which avoids exhaustive enumeration of the data association hypotheses by cutting off some branches of the tree that fall under a lower bound on the number of matches that can be achieved. Finally, the association set with largest number of matches and lower spatial distance errors is chosen.

In the last years, one branch of SLAM research has focused in single candidate approaches by improving the reliability of appearance descriptors such as *Scale Invariant Feature Transform (SIFT)*[7], *Speeded-Up Robust Features (SURF)*[8] and more recently *Binary Robust Invariant Scalable Keypoints (BRISK)*[9]. Efforts have also been held into increasing the precision of SLAM estimation by including large number of features. A.Handa et al. proposed an algorithm based on Active Matching called *Chow Liu Active Matching (CLAM)* and *Subset Active Matching (SubAM)*[5] that is able to scale well with the number of landmarks. Despite the efforts into making SLAM algorithms faster and more precise, not much has been done into making them more robust by propagating multiple data association hypotheses among different frames. In this paper, an algorithm inspired on object tracking probabilistic modeling and *Active Search*[2] is proposed. The presented approach relies on the particle filter framework in order to keep multiple data association hypotheses within one frame and between different frames.

Table (1) summarizes the characteristics of some important algorithms proposed in literature for data association in SLAM applications.

3. PROPOSED ALGORITHM

In the first part, the theoretical grounds of the probabilistic modeling will be briefly described. This section explicitly states the assumptions that are made in the probabilistic modeling of the problem.

3.1. Probabilistic Modeling

Lets assume that at each time step t, $\mathbf{x}$ is the total state of the system modeled as a Gaussian random stacked vector formed by the pose of the system $\mathbf{x}_v$ and N tracked landmarks $\mathbf{x}_i = (\mathbf{x}_1, ..., \mathbf{x}_N)^T$. The measurement process is governed by

$$\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}) + \mathbf{n}_i \tag{1}$$

with $\mathbf{h}_i$ a function that describes the transformation of the state space into the observation space $\hat{\mathbf{z}}_i$ and $\mathbf{n}_i$ is a zero mean Gaussian variable with covariance $\mathbf{R}_i$ which represents the noise in the sensor measurement. Lets define the mean vector of the state $\hat{\mathbf{x}}_m$ and covariance matrix $\mathbf{P}_m$ as

$$\hat{\mathbf{x}}_m = \begin{pmatrix} \hat{\mathbf{x}}_t \\ \hat{\mathbf{z}}_1 \\ \hat{\mathbf{z}}_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}}_t \\ \mathbf{h}_1(\hat{\mathbf{x}}) \\ \mathbf{h}_2(\hat{\mathbf{x}}) \\ \vdots \end{pmatrix} \tag{2}$$

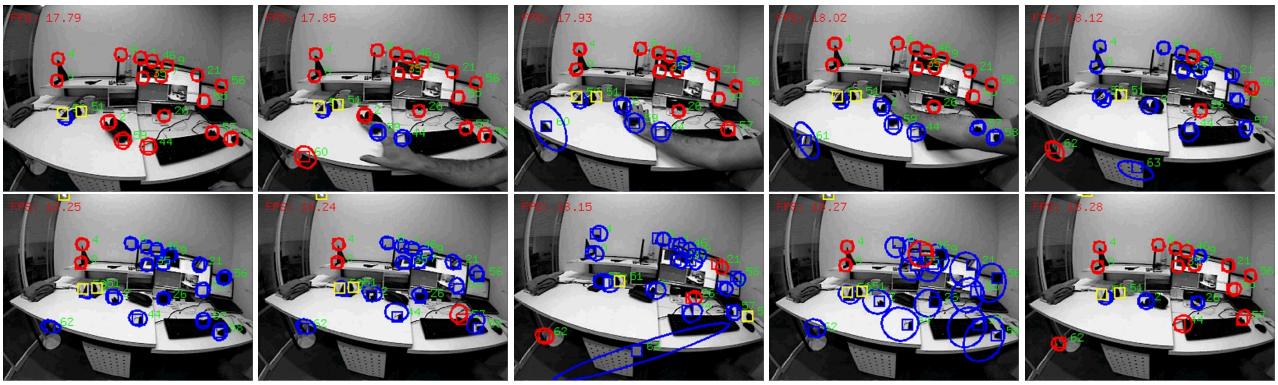

Fig. 1. The image sequence shows an example when the proposed algorithm recovers after modifying the position of an object in the scene. The ellipses represent the uncertainty on the position of different features. Red and blue ellipses represent landmarks that are successfully and unsuccessfully matched in the current frame, respectively.

$$\mathbf{P}_{\mathbf{x}m} = \begin{bmatrix} \mathbf{P}_x & \mathbf{P}_x\frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} & \dots \\ \mathbf{P}_x\frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} & \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}}\mathbf{P}_x\frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} + \mathbf{R}_1 & \dots \\ \mathbf{P}_x\frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} & \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}\mathbf{P}_x\frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} & \dots \\ \vdots & \vdots & \end{bmatrix} \tag{3}$$

$$\mathbf{P}_{\mathbf{x}m} = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy_1} & \Sigma_{xy_2} & \dots \\ \Sigma_{y_1x} & \Sigma_{y_1y_1} & \Sigma_{y_1y_2} & \dots \\ \Sigma_{y_2x} & \Sigma_{y_2y_1} & \Sigma_{y_2y_2} & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{4}$$

Vector $\hat{\mathbf{x}}_m$ and covariance $\mathbf{P}_m$ are used in order to calculate the the Mutual Information described later in Subsection 3.2. The lower-right part of the matrix from eq.3 and eq. 4 is known as the *innovation covariance* S in the Kalman Filter.

3.2. Algorithm

The Particle Filter consists in approximating the marginal probability distribution at time t with a sum of P weighted delta-Diracs called particles

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \sum_{j=1:P} w_t^{(j)}\delta(\mathbf{x}_t - \mathbf{x}_t^{(j)}) \tag{5}$$

where the weights can be calculated using Sequential Importance Sampling (SIS) as

$$w_t^{(j)} \propto w_{t-1}^{(j)} \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)} \tag{6}$$

Equation (6) is the traditional way to calculate the weights in the particle filter framework but it does not take into account the problem of data association. In this algorithm, an object tracking probabilistic model similar to the one proposed in [10] is used. In particular, the weights are calculated as

$$w_t^{(j)} \propto w_{t-1}^{(j)} \prod_{i=1}^{N_t} p(\mathbf{z}|\mathbf{x}, \mathrm{r})p(\mathrm{r}|\mathrm{N_C}, \mathbf{x}_{\mathrm{found}})p(\mathrm{N_C})p(\mathbf{x}_{\mathrm{found}}) \tag{7}$$

Where r is a random vector for associating the landmarks to different observations, $p(\mathrm{N_C})$ is a Poisson probability distribution with parameter $\mathrm{N_C}$ for modeling the number of clutter observations in the image, $p(\mathbf{x}_{\mathrm{found}})$ is a binomial probability distribution for modeling the fact that we do not detect the landmarks in every frame and $p(\mathbf{z}|\mathbf{x}, \mathrm{r})$ is a multinomial probability distribution that models the likelihood of associating each landmark to the different observation candidates. More precisely, the probability of a landmark to be found is given by

$$p(\mathbf{x}_{\mathrm{found},i}) = [\mathrm{P_{present}} * \mathrm{P_{search}}]^{\mathbf{x}_{\mathrm{found},i}} \times [(1 - \mathrm{P_{present}}) + \mathrm{P_{present}} * (1 - \mathrm{P_{search}})]^{(1-\mathbf{x}_{\mathrm{found},i})} \tag{8}$$

In practice, the probability of finding a landmark is modeled as the probability ($\mathrm{P_{present}}$) that it is present in the image and it is inside the search area ($\mathrm{A_{search},}_i$). Where $\mathrm{A_{search},}_i$ is defined as the area around the predicted position of the landmark containing $\mathrm{P_{search}}$ probability mass. An example of the landmark search area can be seen as ellipses in fig.(1). The probability of not finding the landmarks is the sum of the probabilities that either the landmark is not visible or it is present but outside of the search area $\mathrm{A_{search},}_i$. $\mathrm{P_{present}}$ is a parameter of the algorithm that can be seen as a measure of how likely is a landmark to appear at every frame while $\mathrm{P_{search}}$ is a parameter that sets a trade-off between robustness and processing time, i.e. values close to 1 for $\mathrm{P_{search}}$ makes the search area $\mathrm{A_{search},}_i$ to be large so it is more likely to find the landmarks if they are present while low values of $\mathrm{P_{search}}$ makes $\mathrm{A_{search},}_i$ to be smaller and more likely to miss some associations.

3.2.1. *Managing the particles*

At each time step, P particles indexes are sampled from eq. (5) according to their weights. The number of drawn indexes for each particle indicates how many "child particles' should be spanned. For example, lets suppose that there are two particles, each one with weight $w_t^{(0)} = w_t^{(1)} = 0.5$. Lets also consider that the indexes $idx = \{0, 1\}$ are drawned from (5)). This means both particles continue to be alive. Then suppose that particles have weights

Algorithm	Position Error	Orientation Error	Landmarks Error	Averg. time per frame (ms)
Spatial Nearest Neighbor	0.55381	0.02576	5.17722	40.0957
Appearance Nearest Neighbor[6]	3.72404	0.11035	313.843	35.5125
1-point-RANSAC[3]	0.07079	0.00691	0.93737	97.9692
Active Search[2]	1.70737	0.11729	19.29290	20.04350
Active Matching[4]	0.28598	0.010324	0.97682	58.621
JCBB[1]	0.13856	0.01011	1.19577	1380.81
MHDA (**proposed**)	0.00687	0.00724	0.22388	51.545

Table 2. Results for first 200 frames of "Over the table' sequence (publicly available[5]). Image dimension are 640x480

$w_t^{(0)} = 0.1$ and $w_t^{(1)} = 0.9$ and indexes $idx = \{1, 1\}$ the drawn instead. This means that the first particle cease to exist due to its low weight and the second particle should span a new "child particle'.

3.2.2. Measuring the landmarks in each particle

The Mutual Information(M) and *measurement efficiency* are calculated as proposed by A.Davison in Active Search[2]

$$\mathrm{M}_{\mathrm{eff},i} = \mathrm{M}_i / \mathrm{A}_{\mathrm{search},i} \tag{9}$$

With the Mutual information of landmark $\mathbf{x}_i$ calculated as

$$\mathrm{M}_i = 0.5 * log_2 \frac{|\Sigma_{xx}|}{|\Sigma_{xx} - \Sigma_{y_i x} \Sigma_{yy_i}^{-1} \Sigma_{xy_i}|} \tag{10}$$

Even if mutual information is also used in MHDA for guiding the search in the image, the iteration over the landmarks is performed differently. Instead of iterating the landmarks deterministically as in [2], landmarks are iterated in a random way by sampling one landmark proportionally to $\mathrm{M}_{\mathrm{eff}}$. So landmarks that have large values of $\mathrm{M}_{\mathrm{eff}}$ are still more likely to be measured first. Nevertheless, since different particles iterate over the landmarks in a different way, it makes the algorithm more robust to ambiguities within the frame.

3.2.3. Difference with other approaches

Another main difference of the proposed algorithm with respect to the original Active Search algorithm[2] is that it relies on a combined score from spatial and appearance coherence. Furthermore, MHDA propagates different association hypotheses in different particles which makes it more difficult to fall in errors due to ambiguities in one landmark measurement. Multiple Hypotheses Data Association (MHDA) is also different to Active Matching in the sense that it is based on parallel inference using particles instead of multiple hypotheses using a mixture of Gaussians.

4. EXPERIMENTS AND RESULTS

The proposed algorithm has been implemented on top of a *inhouse* version of the publicly available MonoSLAM[11] ported to windows by using Eigen Library and Mobile Robot Programming Toolkit (MRPT) Library[12]. Additionally, *inverse depth parametrization*[13] and patch warping have been ported from J.Civiera matlab code found in [14]. All of the algorithms in the comparison of table 2 are implemented in c++. Some of them have been found already implemented in c++ and all others have been ported from matlab such as the 1-Point RANSAC algorithm. For JCBB, it has been used a version of the algorithm found in the MRPT Library which has been ported into c++ from the original matlab code.

The parameters used in all of the experiments are $\mathrm{P}_{\mathrm{present}} = 0.9$ and $\mathrm{P}_{\mathrm{search}} = 0.95$ which corresponds to approximate 2.5 standard deviations area search. The clutter mean N_{C} has been set to 20 which corresponds to having and average of 20 false positive observations per image.

4.0.4. Synthetic Dataset

The experiments are performed on a publicly available dataset called "over the table"[5]. This dataset is a synthetic photo-realistic image sequence generated by POVRay with a 640x480 image resolution. This dataset provides ground truth pose of the camera and depth distance for every pixel at every time frame. Therefore, quantitative results from camera pose and the landmarks reconstruction can be obtained. The sequence is in an office context and it contains different challenging situation such as repetitive texture (such as keyboards, telephone buttons, etc), objects at different depth planes and several short and long term occlusions. Table 2 summarizes the obtained results. It is possible to see that the two flavors of NN together with Active Matching have the largest error values. Specially, the algorithms that rely only in appearance have problems due to repetitive texture in the scene. One exception is the 1-Point RANSAC which provides good results over all. Nevertheless, it has been observed that 1-Point RANSAC algorithm has difficulties in converging to the real 3D position of the landmarks caused by the difference in depths planes in the scene, i.e. the depth plane that has the majority of landmarks biases the estimation and landmarks in different depth planes are always taken as outliers. In general, Active Matching and JCBB have similar performances but the former one is in average 26 times faster. The proposed MHDA algorithm with only 2 particles provides the best performance while requiring low processing time. Since the processing done by MHDA in each particle is simpler than the Gaussian mixture propagation done in the Active Matching algorithm, the time required to run one particles of MHDA is faster than running AM in a frame basis. Nevertheless, propagating multiple weak hypotheses over the frames is able to produce better results over time. Although the used implementations were from different sources and some of them are perhaps more optimized than others, we believe that the results still give a good general understanding of the relative speeds.

4.0.5. Real sequence

The experiments with real sequences were performed with real-time tracking using a webcam with a wide angle lens (See fig. 1) at a 480x320 image resolution. This resolution has been chosen because it allows real-time tracking. From all the tested algorithms, only Active matching and the proposed algorithm MHDA are able to keep up with real-time usage. The performance of Active Matching and MHDA is similar when there are not many association ambiguities. Nevertheless, the proposed algorithm is able to recover for ambiguous situations that can last a couple of frames due to the fact that it keeps multiple hypotheses over time. MHDA is able to run at about 30fps with around 50 features using non-optimized c++ code.

5. CONCLUSIONS

It this paper, an algorithm called *Multiple Hypotheses Data Association* (MHDA) has been proposed. This approach is inspired on object tracking probabilistic modeling and A.Davison Active Search algorithm. It has been shown, using synthetic and real sequences, promising results by propagating multiple "weak" hypotheses over time instead of propagating only one strong hypothesis. It has also been shown that the proposed algorithm is suitable for real-time monocular SLAM applications running with a 480x320 resolution. Further work will focus on optimizing the code using a GPU in order to process the particles in parallel.

6. REFERENCES

[1] J. Neira and J.D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. Vol. 17, no. No. 6, pp. pp. 890 – 897, December 2001.

[2] Andrew J. Davison, "Active search for real-time vision," in *In Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 66–73.

[3] Javier Civera, Oscar G. Grasa, Andrew J. Davison, and J. M. M. Montiel, "1-point ransac for ekf-based structure from motion.," in *IROS*. 2009, pp. 3498–3504, IEEE.

[4] Margarita Chli and Andrew J. Davison, "Active matching for visual tracking," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1173 – 1187, 2009, Inside Data Association.

[5] Ankur Handa, Margarita Chli, Hauke Strasdat, and Andrew J. Davison, "Scalable active matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[6] A.J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. International Conference on Computer Vision, Nice*, Oct. 2003.

[7] D Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 60no2pp91–110, 2004.

[8] H Bay, a Ess, T Tuytelaars, and L Vangool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, pp. 346–359, 2008.

[9] Stephan Leutenegger, Margarita Chli, and Roland Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[10] M.S. Alvarez, L. Marcenaro, and C.S. Regazzoni, "Efficient framework for extended visual object tracking," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 1831–1838.

[11] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse, "Monoslam: Real-time single camera slam," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[12] Jose-Luis Blanco et al., "The mobile robot programming toolkit," `http://www.mrpt.org/`, 2013.

[13] Javier Civera, Andrew J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular slam.," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[14] OpenSLAM, "Openslam," `http://www.openslam.org/`, 2012.