Literature Study on Monocular SLAM & EKF

Addi Djikic – 941028
Jan 2017
addi@kth.se

***Abstract*—This literature study is about describing and analyzing *SLAM*, (Simultaneous Localization and Mapping), especially monocular SLAM where a simple one-eyed camera can be used for the SLAM algorithm. The *EKF*, (Extended Kalman Filter) will also be described and how it interacts and what role it has on the SLAM algorithm. Earlier work and reports about monocular SLAM will be evaluated and results analyzed, along with how they use their algorithms to achieve their results. The research of monocular SLAM has shown to be important and beneficial for future technology, by successfully implementing a distance prediction system for simple cameras, we have room to develop future robots for a much cheaper price that can be used for various future applications. In this study we talk about great results achieved with monocular SLAM and how they will benefit us in future applications.**

I. Introduction

A. SLAM in Brief

SLAM, (*Simultaneous Localization and Mapping*) has been one of the central obstacles to overcome in the field of robotics. Basically the SLAM algorithm is used whenever a robot for example, does not have information about how the map looks of the surrounding environment, along with with no information about its poses [1]. In simplification, SLAM works in a way as its name indicates, that in the case for the robot, it simultaneously draws or acquires a map of its surroundings while at the same time localizes itself in reference to the drawn map. This is done by collecting measurement data $z_{1:t}$, and control data $u_{1:t}$, which are the notations from [1] in which we will use most notations from in this literature report. Will will mostly talk about monocular SLAM and how the EKF relates to it, this is important because SLAM along with monocular SLAM can be used for most future applications such as drones, UAV, autonomous cars, computer vision or war combat robots for instance. Turning a basic camera into a real-time visual compass with distance prediction is a huge step for the implementation of future robots, which we will give more examples off later on.

1) Main problem with monocular SLAM: The task here in this literature report is to show how the SLAM algorithm can be used to successfully permit of real-time 3D localization and mapping by only using a single 'one eyed' camera and predict distances with it. There have been different studies on monocular SLAM and we will bring forward different solutions, among others; [2], where they talk about a real-time single camera with SLAM, [3] where they use monocular SLAM and talk about inverse depth parametrization which we will also discuss, and [4] which talks about drift-free estimation with high frame-rate real-time operation with monocular SLAM. Monocular SLAM has shown great interest in future areas. Areas where robots can simply use a hand held camera to map its own location and analyze its surroundings in distance estimation. [5] talks about different interesting methods for monocular SLAM and compares them with each other, such as large-scale direct monocular SLAM. However we will not go into depth abut different models but instead the process for monocular SLAM and different application in some recent studies in III.

II. Theory & Introduction

1) Two main forms of SLAM: The two main existing forms of SLAM are the so called *online-SLAM problem* and the *full-SLAM problem* [1]. The first online-problem estimates the posterior of the robot over the momentary pose together with the drawn map as (1)

$$p(x_t, m|z_{1:t}, u_{1:t}) \tag{1}$$

where x_t is the pose at a given time and m is the map.

However, it is more convenient to use the full SLAM problem approach, which is given as (2)

$$p(x_{1:t}, m|z_{1:t}, u_{1:t}) \tag{2}$$

and as in this case we estimate the posterior over the whole path $x_{1:t}$ instead of just its current position as in (1). This is preferable because we achieve all there is to know about the map and the pose.

To state the main difference between the two concepts, we say that the online-SLAM is when we integrate over the whole full past poses of the full-SLAM concept, i.e. as given in (3)

$$\begin{aligned} &p(x_t, m|z_{1:t}, u_{1:t}) = \\ \int\int ... \int &p(x_{1:t}, m|z_{1:t}, u_{1:t}) dx_1 dx_2 ... dx_{t-1} \end{aligned} \tag{3}$$

A. EKF & KF

The EKF (*Extended Kalman Filter*) is an extended version of the basic Kalman Filter (KF), which is given by the algorithm in Table 3.1 in [1]. The Kalman filter shows the belief by the moments representation, for given time t the mean μ_t and covariance Σ_t represents the belief [1]. The posteriors are also Gaussian distributed if the following arguments hold (from [1]):

1) Probability of the next state, i.e. $p(x_t|u_t, x_{t-1})$ should be a linear function.
2) The measurements probability, i.e. $p(z_t|x_t)$ also needs to be linear in its arguments with the added Gaussian noise as $z_t = C_t x_t + \delta_t$
3) The initial belief, i.e. $bel(x_o)$ must be normal distributed.

However, in practice, using the regular KF with added Gaussian noise is not that beneficial because of its linearity demands. In real life we mostly deal with non linearity. This is where the EKF takes its place, where it overcomes the linearity assumption of the KF. This is very useful for robots in practice, as an example, if a robot moves with constant translational and rotational velocity it will move in a circular motion, and that cannot be described with linear next state transitions as in the KF case. We will derive the next state probability and measurements as non linear functions g and h [1] as (4)

$$\begin{aligned} x_t &= g(u_t, x_{t-1}) + \epsilon_t \\ z_t &= h(x_t) + \delta_t \end{aligned} \tag{4}$$

The Extended Kalman Filter now gives an approximation of the true belief and approximation by a Gaussians and linearization with Taylor expansion, which is the main difference from the regular Kalman filter. This gives us the upper hand for non linear systems. The algorithm for the EKF is given in Algorithm 1

Algorithm 1 Extended Kalman Filter

1: EKF($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
2: $\quad\bar{\mu}_t = g(u_t, \mu_{t-1})$
3: $\quad\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$
4: $\quad K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$
5: $\quad\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
6: $\quad\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$
7: return μ_t, Σ_t

We observe the difference in line 2 in Algorithm 1 that the state prediction is given as the function g from (4) and the measurement prediction as the function h in line 5, which are both the non linear part of the Kalman filter that gives us the EKF.

B. SLAM & EKF

We have now stated both how SLAM works in brief and the EKF, and perhaps the most influential SLAM algorithms is based on the EKF [1]. In simplifications the EKF-SLAM integrates or applies the EKF to an online-SLAM problem by using maximum likelihood data associations. This leads to a couple of initial conditions from [1]:

1) The EKF approach is based on point landmarks on the map, and it tends to work better if there are less of them. Because of this

condition it leads to that the EKF SLAM requires careful engineering of feature detectors, i.e. for example using artificial beacons as features.
2) Because of the assumption of Gaussian noise in the motion and prediction the EKF SLAM algorithm does, it means that the uncertainty in the posterior must be quite small to avoid large errors.
3) It can only handle positive sightings of landmarks and not negative information from sensor measurements, this is due to the Gaussian belief representation.

The EKF-SLAM approach is quite similar to what the EKF algorithm 1 does, except with the main characteristics that we estimate the position relative of the landmarks that we find with the robot and append it to the state vector of the robot.

I believe it is convenient to display the EKF-SLAM algorithm given by Fig. 1

1: **Algorithm EKF_SLAM_known_correspondences**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t$):

2: $$F_x = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 1 & \underbrace{0\cdots0}_{2N} \end{pmatrix}$$

3: $$\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\mu_{t-1,\theta} + \frac{v_t}{\omega_t}\sin(\mu_{t-1,\theta} + \omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\mu_{t-1,\theta} - \frac{v_t}{\omega_t}\cos(\mu_{t-1,\theta} + \omega_t\Delta t) \\ \omega_t\Delta t \end{pmatrix}$$

4: $$G_t = I + F_x^T \begin{pmatrix} 0 & 0 & \frac{v_t}{\omega_t}\cos\mu_{t-1,\theta} - \frac{v_t}{\omega_t}\cos(\mu_{t-1,\theta} + \omega_t\Delta t) \\ 0 & 0 & \frac{v_t}{\omega_t}\sin\mu_{t-1,\theta} - \frac{v_t}{\omega_t}\sin(\mu_{t-1,\theta} + \omega_t\Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x$$

5: $\bar{\Sigma}_t = G_t\, \Sigma_{t-1}\, G_t^T + F_x^T\, R_t\, F_x$

6: $$Q_t = \begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\phi & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}$$

7: for all observed features $z_t^i = (r_t^i \;\; \phi_t^i \;\; s_t^i)^T$ do

8: $j = c_t^i$

9: if landmark j never seen before

10: $$\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \\ \bar{\mu}_{j,s} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \\ s_t^i \end{pmatrix} + r_t^i \begin{pmatrix} \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \\ 0 \end{pmatrix}$$

11: endif

12: $$\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix}$$

13: $q = \delta^T \delta$

14: $$\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \\ \bar{\mu}_{j,s} \end{pmatrix}$$

15: $$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 & 0 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 & 0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 1 & 0\cdots0 & 0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 0 & 0\cdots0 & 1 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 0 & 0\cdots0 & 0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 0 & \underbrace{0\cdots0}_{2j-2} & 0 & 0 & 1 & \underbrace{0\cdots0}_{2N-2j} \end{pmatrix}$$

16: $$H_t^i = \frac{1}{q} \begin{pmatrix} \sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & -\sqrt{q}\delta_x & \sqrt{q}\delta_y & 0 \\ \delta_y & \delta_x & -1 & -\delta_y & -\delta_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} F_{x,j}$$

17: $K_t^i = \bar{\Sigma}_t\, H_t^{iT} (H_t^i\, \bar{\Sigma}_t\, H_t^{iT} + Q_t)^{-1}$

18: endfor

19: $\mu_t = \bar{\mu}_t + \sum_i K_t^i (z_t^i - \hat{z}_t^i)$

20: $\Sigma_t = (I - \sum_i K_t^i\, H_t^i)\, \bar{\Sigma}_t$

21: return μ_t, Σ_t

Fig. 1. This algorithm is taken directly from the book [1], which shows the EKF-SLAM algorithm, which implies the SLAM problem with a feature based map and requires a sensor on the robot which measures bearing, θ and range.

The EKF-SLAM algorithm in Fig. 1 can be found in [1], and to avoid redundant explanations of every step, all the algorithms notations and variables can be found in [1]. However, a brief explanation of the steps in the algorithm will be provided below.

The algorithm is based of the EKF-SLAM with known correspondence, meaning its based of the continuous part of the SLAM algorithm. As said the state vector now includes the map m and pose x_t which now give us our new combined state space vector seen in (5).

$$\begin{aligned} y_t = (x_t, m)^T = \\ = (x, y, \theta, m_{1,x}, m_{1,y}, s_1, m_{2,x}, m_{2,y}, s_2, ... \\ ..., m_{N,x}, m_{N,y}, s_N) \end{aligned} \tag{5}$$

whereas the position of the robot relative to the map are given from x, y and θ in (5), the $m_{i,x\&y}$ represents the coordinates of the landmark while the s represents the signature and N total landmarks [1].

If we observe Fig. 1, and briefly explain what is going on through the steps of the algorithm:

- The lines 2 to 5 employ the update of the motion.
- Lines 3 to 5 sets the mean and the covariance in the belief distribution of the robot position with respect to the motion model.
- Line 7 to 18 goes through all measurements, and also 6 to 20 incorporates the measurement.
- The condition in line 9 will return true for the landmarks with no initial estimated position, i.e. only new landmarks.
- Given the landmarks from line 9, line 10 then initializes the position for those landmarks from the obtained bearing θ and measurements z_t. Notice that the step is imported for the non-linear case, i.e. the EKF, it is not needed for the simple Kalman filter.
- The Kalman gain is given in line 17 and the

expected measurement is given in line 14.
- The update then goes through the steps in line 19 and line 20.

The EKF-SLAM discussed is when we presume the cognizance of the correspondence. This can now be generalized to a more overall algorithm when we presume unknown correspondence. The main difference now is that the estimator is an incremental maximum likelihood estimate for determining the correspondence. The algorithm is very similar and analogically derived from Fig. 1 and can be found in table 10.2 of the book [1]. For an accurate mathematical derivation of every part of the EKF-SLAM the reader is referred to chapter 10.2.3 of [1].

C. Monocular Camera

A particularly interesting area to apply the EKF-SLAM algorithm is in fact monocular SLAM, to first establish what defines a monocular camera ('one-eyed camera'), that is basically a refracting telescope that can magnify images from distance [6]. A monocular camera captures images in two dimensions while binocular cameras capture in three dimensions. A simple image of such a camera can be visualized in Fig. 2

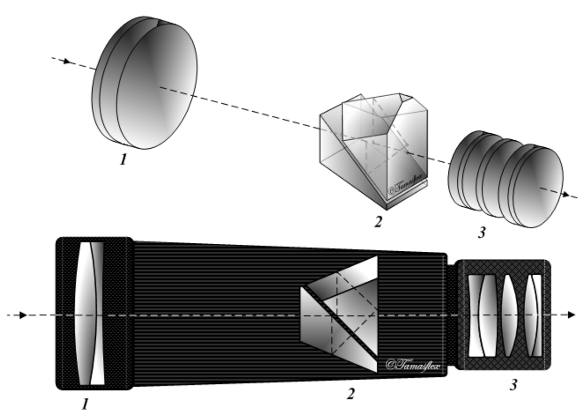

Fig. 2. Monocular camera where 1 is the objective lens, 2 is Schmidth Pechan prism and 3 is the eyepiece. (Image taken from [6])

III. Related Work & Results

As mentioned, various studies have been done around monocular SLAM algorithms, among others, A. Davidson et. al [2] speaks about how they have a successful real-time algorithm that can recover a 3D trajectory from a monocular camera. They acknowledge that they are the first which to accomplish an application of the SLAM methodology from a mobile robotics in pure vision of a uncontrolled camera. As in similar to J. Civera et. al [4], drift free estimation in real time was achieved in both the cases. When using inverse depth parametrization with monocular SLAM as in the paper J.M.M Montiel et. al [3] they also achieve real-time 3D localization and mapping in the pure vision domain with a monocular camera. There will be most weight on the paper J.M.M Montiel et. al [3] from now on due to they have a good explanation on how they use the EKF with monocular SLAM and how the use a parallax effect to measure distances.

A. Measuring Depth in Images With Monocular

The interesting question is how to measure the depth in an image to be able to predict distances. Monocular cameras can measure bearings of image features, this is basically done by letting the camera move, i.e. keep it in motion so it can repeatedly capture the area surrounding it. While it moves it captures ray of lights from the features it sees to the cameras optic center. This method of how to capture features is also described by the paper [3]. There will exist an angle between the rays, which is the features parallax and with the help of it the depth can be estimated. Explanation of a parallax can be seen in Fig. 3

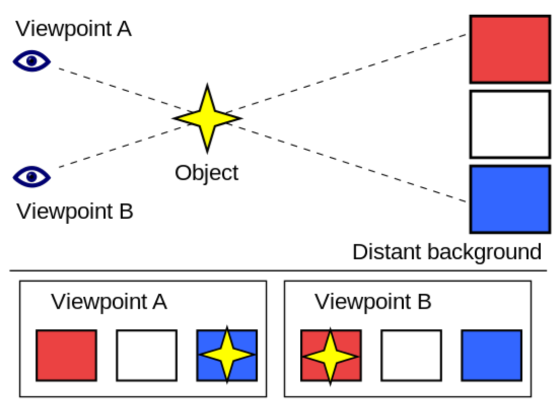

Fig. 3. The parallax effect, when in Viewpoint A it looks like the objects is inside the blue square, and when in Viewpoint B it appears to be inside the red square. (Image taken from [7])

The parallax effect for monocular vision is the key concept for measuring depth so that we can predict distances, unlike in the case for computer vision nowadays the concept of 'stereo vision' is used to measure depth, explained in [8]. The stereo vision will give impression of how the 'cyclopean eye' of the human work, i.e. the concept of using two eyes and make advantage of the distance between the eyes and use epipolar coordinates to combine the vision distance and estimate the location of the viewed object as in Fig. 3. The same principal is used with monocular camera along with the parallax effect. The conception of stereo vision can be visualized in Fig. 4

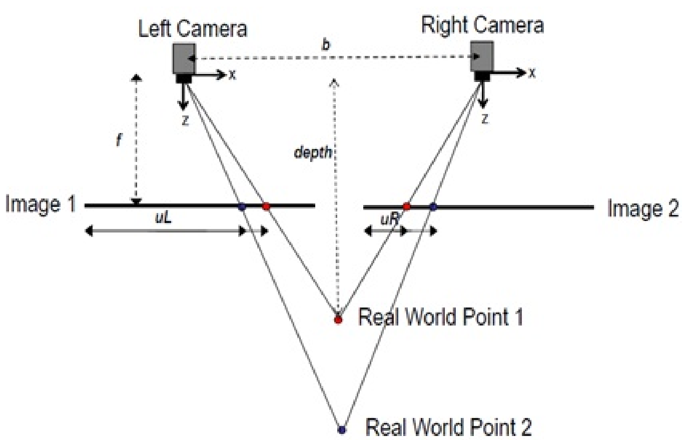

Fig. 4. Image taken from [9], showing a left and right camera and how the combine their single vision to measure the depth, i.e. the distance prediction to the viewed object.

B. Monocular SLAM Distance Prediction

The fascinating concept with using monocular SLAM is that 3D localization and mapping can be done with a monocular and the EKF-SLAM algorithm we described, along with no particular extra sensing. By using the EKF-SLAM and parallax effect as in [3], one can take advantage by setting a mono-camera in motion, and combining the multiple images to predict distances of detected objects and landmarks.

1) Distant Objects: One way to exclude distant objects is to use the approach as in the paper J.M.M Monteil et. al [3], by taking advantage of when no parallax of a feature is observed. Meaning that when no parallax of a feature is detectable after X meters for example, tells us we have a point at 'infinity', somewhat a vanishing point in an image of homogeneous coordinates [8]. This will be decisive for the importance of what features to ignore and which features to keep track of that are important. For object near the mono-cam a few centimeters of movement are sufficient for depth estimation, which is basically standard movement of a robot that simultaneously localizes itself with EKF-SLAM.

2) Inverse Depth: In the paper [3] they use parametrization that can track both near and far features that uses a concept called *inverse depth parametrization* along with the standard EKF-SLAM. It uses a Gaussian distribution to span uncertainty over a finite depth of estimated features, and their algorithm does not need any initial process for the features because that are handled right in the start of their tracking method. Let us simply explain how the inverse depth is intended to be used:

- Given a distance D, we measure the inverse depth $\frac{1}{D}$
- If the inverse depth is equal to zero, we assume distant objects with low confidence, thereof we use the EKF to increase confidence.
- If it gives us an point in *infinity*, we have a close object and we use our homogeneous coordinate system.

C. Parametrization

By using the concept of the paper J.M.M Monteil et. al [3] we can define a 3D point, k, with the new state vector (6) of a 6 dimensional state.

$$y_k = (x_k, y_k, z_k, \theta_k, \phi_k, \rho_k) \tag{6}$$

with x_k, y_k and z_k being the camera optical centre of the observed image and θ_k, ϕ_t the azimuth and elevation angle. While ρ_k is the distance which gives us our inverse depth $d = \frac{1}{\rho_k}$.

The state vector (6) models a 3D point given as (7)

$$(x_k, y_k, z_k)^T = \frac{1}{\rho_k}\boldsymbol{m}(\theta_k, \phi_k) \tag{7}$$

The model parametrization of the state vector (6), the modeled point (7), along with the camera can be view in Fig. 5

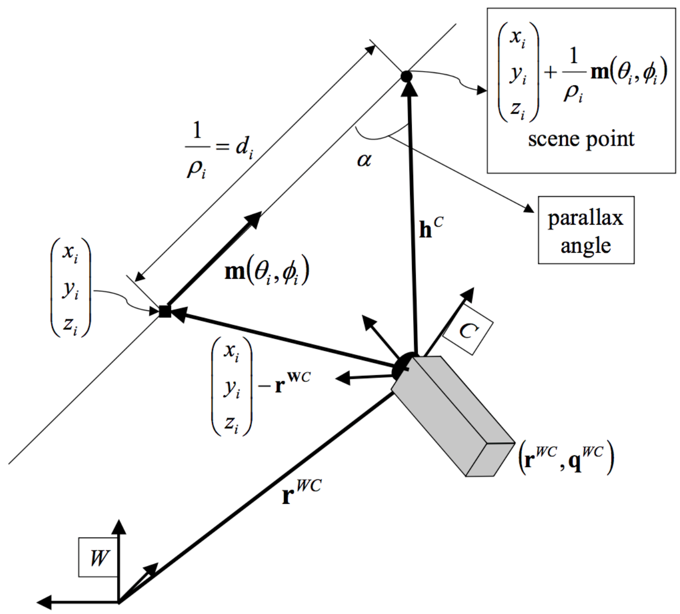

Fig. 5. Image taken from [3], displaying the parametrization for the observed feature and measurements with inverse parametrization.

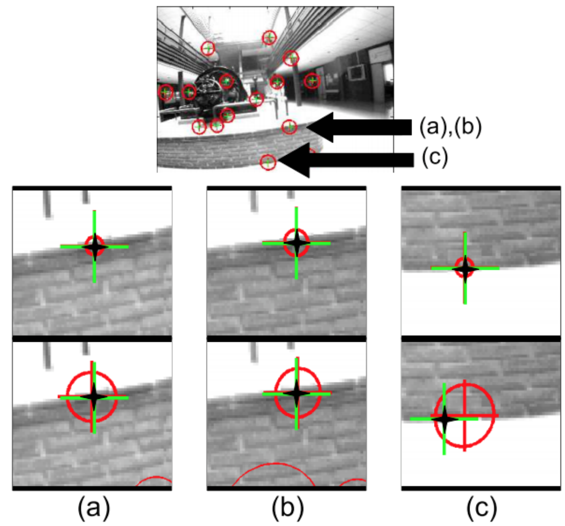

Fig. 6. Image taken from [4], displays different points found in the scene with different depths.

D. Results of Previous Work

Recent work about monocular SLAM has given rise to many positive results, we mentioned a lot about the paper [3] that succeeded with their inverse depth parametrization with a low cost hand held camera at 30Hz, 30fps with just a pixel resolution of 320x240. Many others, as mentioned, the papers [2], [4] among others have all achieved positive results with simple monocular cameras moving through an unknown environment with drift-free estimation at 30Hz. Fig. 6 from the paper [4] visualizes their result of how a monocular camera can detect points with various depths.

This shows us that using simple models we have room for various optimization and development of much more advanced technology of using the monocular SLAM concept. In section V we talk about various application that already use this system and what future applications can make use of it.

IV. Discussion & Purposed Optimization

As mentioned there is a great potential withing the concept of monocular SLAM, though it is a convenient solution that the most basic EKF and SLAM approach works and that the results were beneficial from previous research, we can still 'enlarge' this development. There are more estimation filters than the KF and EKF, for example we have the Unscented Kalman Filter (UKF), which basically uses the same perception as the EKF but chooses a set of points instead of one for linearization to optimize the performance and accuracy of the mean on covariance. In the paper E.A. Wan et. al [10] they speak about the UKF and points out the flaws that exist with the EKF and that the UKF is a more convenient solution by using a so called deterministic sampling approach. One could also mention the Extended Information Filter (EIF), which is described in [1], it is a different parametrization of the EKF depending on what you need to optimize. It is easier to compute the distribution belief with the EIF due to $\Omega = \Sigma^{-1}$ (the new covariance matrix is the inverse of the covariance matrix used in the EKF). Not only can we optimize the estimation filter, but there are different SLAM approaches too, for example we have the FastSLAM theory. In the paper M. Montermerlo, S. Thrun et. al [11], they talk about an algorithm which can recursively use its estimation for full posterior distribution of the robots localization and landmark locations. They state that their algorithm works with up to 50 000 landmarks. It is also worth mentioning the work of E. Eade et. al [12], which states that they are the first ones to introduce a FastSlam particle filter for a monocular camera that allow real-time performance with parallel mapping of great numbers of landmarks. Mentioning these various filters and this particular FastSLAM method, we mark that we have a lot of development to work with and various optimization methods to acknowledge for the sake of performance of monocular SLAM among other methods.

V. Future Applications

There is a vast selection of what we can achieve with the method and basic concept of SLAM and monocular SLAM, in the book [1], they show an example of an underwater vehicle Oberon, which can be observed in Fig. 7.

Fig. 7. Image taken from [1], shows the underwater vehicle Oberon from the University of Sidney.

These type of unmanned vehicles is of a broad interest nowadays and are a prefect candidate to use monocular SLAM. If we can build these kind of robots with high performance and low cost cameras that can localize themselves in unknown and uncharted environments we can explore and achieve progress faster than before, not only wealthy organizations such as the military can use the technology. Even so, the military is one of the organization where this type of technology is moving forward the fastest. If we take for example SAAB (Swedish Airplane Ltd), which are developing war combat airplanes but currently manned with humans, could benefit of a stable technology of unmanned airplanes, or so called UAV:s for example. If a system could be simple enough and have a firm distance prediction system better than the human eye they could potentially exclude humans from their air crafts and keep them completely autonomous.

Perhaps such corporations such as SAAB already have their capital for these type of technology and development. But as mentioned before, the interesting part here would be if one could put the development or make use of the development of monocular SLAM for the regular individual. Meaning that if we can perhaps create apps for mobile devices that uses a typical cellphone camera for monocular SLAM and distance prediction. This is possible, as we discussed the previous work mention in this report, when they used low-resolution monocular cameras and achieved very positive results. One area that could benefit of that regular individuals use monocular SLAM is in areas where one could apply machine learning and deep learning. If we take the example of home care, today their is an global problem of understaffed places that provides home care, such as Sweden and Japan for example. If we could replace humans with robots that take care of the elders instead, we could potentially solve that problem. The problem here is the cost, technology and one of the most important factors, safety. The example of what machine learning could be of use here, is the reason that if the simple individually can use the concept of monocular SLAM, machine learning algorithms could learn and adapt to become more and more safe, which leads to that we can develop more advanced robots and worry less about if they will be unsafe when using methods such as SLAM. This can be done by implementing small robots with monocular cameras, or use cellphone cameras to improve monocular SLAM algorithms. In the long term, this will be a cheap and fast adapting optimized solution.

References

[1] S. Thrun et. al, *Probabilistic Robotics*. Stanford CA, Freiburg Germany, Seattle WA: ACM, 2016, vol. 8.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[3] J. Montiel, J. Civera, and A. J. Davison, “Unified inverse depth parametrization for monocular slam,” *analysis*, vol. 9, p. 1, 2006.

[4] J. Civera, A. J. Davison, and J. Montiel, “Interacting multiple model monocular slam,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 3704–3709.

[5] M. Shridhar and K.-Y. Neo, "Monocular slam for real-time applications on mobile platforms," 2015.

[6] (2016, Dec) Monocular. [Online]. Available: https://en.wikipedia.org/wiki/Monocular

[7] W. Foundation. (2016, Dec) Parallax. [Online]. Available: https://simple.wikipedia.org/wiki/Parallax

[8] K. Castleman, "Digital image processing," 1993.

[9] (2017, Jan) Science behind 3D vision depth Biomechanics. [Online]. Available: http://www.depthbiomechanics.co.uk/?p=102

[10] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. Ieee, 2000, pp. 153–158.

[11] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Aaai/iaai*, 2002, pp. 593–598.

[12] E. Eade and T. Drummond, "Scalable monocular slam," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 469–476.