Mobile Robot Motion Tracking Using Descriptor Matching and Sensor Fusion

Jake Chittle
Engineering Systems and Computing
University of Guelph
Guelph, Ontario, Canada
Email: jchittle@uoguelph.ca

S. Andrew Gadsden
Mechanical Engineering
Unviersity of Guelph
Guelph, Ontario, Canada
Email: gadsden@uoguelph.ca

Mohammad Biglarbegian
Mechanical Engineering
University of Guelph
Guelph, Ontario, Canada
Email: mbiglarb@uoguelph.ca

***Abstract*—This paper presents fast tracking of a mobile robots 2D pose in a plane using the open source computer vision library(OpenCV). This can be useful for setting up experiments to study mobile robot control, robot formation or conflict resolution. Here the feature detectors SIFT, AKAZE and ORB are tested for their speed and accuracy for tracking a robot on a plane of size 2.7m x 2.1m. To determine the accuracy that can be achieved they are compared against an edge-based template matching algorithm which has a known accuracy. First the accuracy vs detection time is studied on different size images. Then sensor fusion is studied by combining the extended Kalman filter (EKF) and unscented Kalman filter (UKF) with odometry to see what gains can be made. Root mean squared pose errors of less than 3mm in translation and less than 1 degree in heading are achieved at a object detection times of less than 50ms.**

I. INTRODUCTION

In order to develop accurate control strategies for mobile robots or self driving cars, their position must be estimated accurately [1], [2]. This can be achieved in several ways. If a kinematic model of the robot is known, the position can be derived by using sensors on the robot, a method known as odometry. One of the oldest techniques is to use encoders on the motor drives, such as the robot in [3]. Encoders placed on the wheels can suffer from wheel slippage. Therefore other techniques have been tried. In [4] an optical mouse is used. While it was shown to have a high resolution, it suffers from problems during non-straight displacements of the robot and depends on the surface.

Using odometry depends on certain dimensional parameters which cannot be known with infinite precision. Therefore any method based on odometry alone will suffer from an error accumulating with time known as drift. Due to this, it is usually complemented with at least one other sensor such as gyroscopes [5], ultrasonic [6] and vision [7].

While a great deal of research is focused on localization or mapping in unknown environments, this paper considers the problem of localization when the robot is confined to a known arena. This is particularly useful for experiments studying conflict resolution and formation strategies such as in [8] and [9]. In these situations, tracking must be fast so the robot can make quick decisions. A couple of examples are provided next. The work in [10] used light emitting diode's on the robots and an overhead camera for localization. Based on a blinking pattern the algorithm determines which robot is in view. The absolute accuracy and detect time was not mentioned. Jarupat in [11] uses 2 overhead cameras to detect robots playing soccer. A blob detection algorithm was used to localize the robot. Positioning errors of 91-105mm in translation and 13-18degrees were reported.

In this paper, descriptor based matching is utilized in order to accurately track a mobile robot. Descriptor based methods have been shown to be fast and accurate. The work in [12], commonly known as the scale-invariant feature transform (SIFT) provided a fast recognition framework shown to be invariant to projective transformations. The work in [13], known as ORB showed similar performance to SIFT but was much faster. More recently, the AKAZE detector [14] was shown to have better performance than ORB and similar performance to SIFT with recognition speeds somewhere in between ORB and SIFT.

II. CONTRIBUTIONS

Since no literature could be found on the absolute pose accuracy that descriptor methods could obtain for tracking objects in a plane, this paper attempts to quantify it. As a ground truth, the edge-based method in [15] is used since it was shown to have accuracies of 1/22 of a pixel and 1/100th of a degree. Later in the paper the use of this algorithm as a ground truth is justified. The results can be used as a reference point for other researches setting up similar experiments.

Finally we study if the results of fusing vision together with odometry using an extended Kalman filter (EKF) and unscented Kalman filter (UKF) to see what gains can be made.

III. BACKGROUND

This paper is based on the kinematics for a differential drive mobile robot that moves around in a plane. The robot has 3 degrees of freedom. It is able to move to a position on a plane which can be described using three coordinates (x, y, θ). These three coordinates can be obtained using odometry from

wheel encoders, or by using machine vision techniques. They can also be fused together using a Kalman filter, which is discussed next.

IV. Kalman Filtering

Odometry and machine vision can be combined using a Kalman filter to obtain a better estimate of the robots position than either technique could yield individually [16], [17]. A prediction can be made to estimate the robots pose at time $(t+1)$ given information from time t [18]. Since the prediction equations are non-linear, the extended Kalman filter (EKF) as well as the unscented Kalman filter (UKF) were chosen. Only the EKF is described here. The reader is referred to [19] for the details of the unscented Kalman filter. The prediction state of the EKF can be written as:

Prediction

$$\hat{\mathbf{x}}_{k+1|k} = f(x_{k|k}, y_{k|k}, \theta_{k|k}) \tag{1}$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^\intercal + \mathbf{Q} \tag{2}$$

where $\hat{\mathbf{x}}$ represents the states, f is a function of the state variables, $\mathbf{P}$ is the state covariance matrix, $\mathbf{Q}$ is the system noise covariance matrix and $\mathbf{F}$ is the state transition matrix. The matrix $\mathbf{F}$ is obtained by taking the Jacobian of the state variables and can be written as

$$\begin{aligned}\mathbf{F} &= \begin{bmatrix} \frac{\partial x_{t+1}}{\partial x_t} & \frac{\partial x_{t+1}}{\partial y_t} & \frac{\partial x_{t+1}}{\partial \theta_t} \\ \frac{\partial y_{t+1}}{\partial y_t} & \frac{\partial y_{t+1}}{\partial y_t} & \frac{\partial y_{t+1}}{\partial \theta_t} \\ \frac{\partial \theta_{t+1}}{\partial x_t} & \frac{\partial \theta_{t+1}}{\partial y_t} & \frac{\partial \theta_{t+1}}{\partial \theta_t} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -\ell_c \sin(\theta_t) \\ 0 & 1 & \ell_c \cos(\theta_t) \\ 0 & 0 & 1 \end{bmatrix}\end{aligned}$$

The prediction state can be updated using a vision measurement and can be written as:

Update

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_{k+1|k}\mathbf{H}^\intercal(\mathbf{H}\mathbf{P}_{k+1|k+1}\mathbf{H}^\intercal + \mathbf{R})^{-1} \\ \mathbf{P}_{k+1|k+1} &= (1 - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}_{k+1|k}(1 - \mathbf{K}_{k+1}\mathbf{H})^\intercal + \\ &\quad \mathbf{K}_{k+1}\mathbf{R}\mathbf{K}_{k+1}^\intercal \\ \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \mathbf{H}\hat{x}_{k+1|k})\end{aligned}$$

where $\mathbf{K}$ represents the Kalman gain, $\mathbf{H}$ is the measurement mapping matrix, $\mathbf{R}$ is the measurement noise covariance matrix and $\mathbf{z}$ is the measurement. Since the vision measurement maps linearly to the state variables, it can be written as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The matrices $\mathbf{Q}$ and $\mathbf{R}$ are discussed in the results section.

V. Experimental Setup

An overview of the testing environment is shown in Figure 1. A mobile robots position is tracked in an area of 2.7m by 2.1m. A 2.5 mega-pixel camera captures the scene from above.

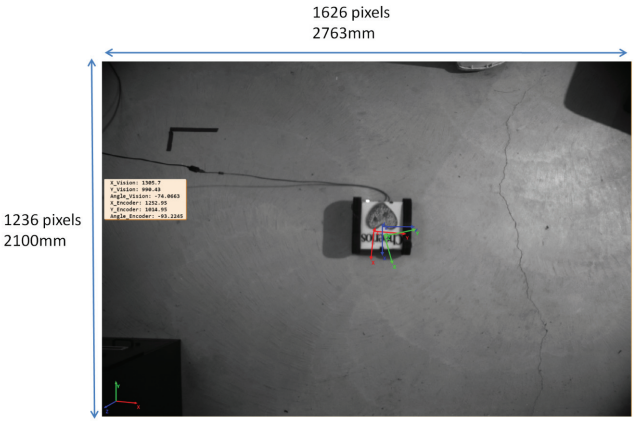

Fig. 1: System Overview. A tracked mobile robots position is estimated using odometry (red triad), machine vision (green triad) and Kalman filtered (blue triad). The world coordinate system is shown in the bottom left corner

A. Robot

Here the details of the robot are discussed. A tracked mobile robot was used with a baseline distance between the wheels of 242mm and a wheel radius of 51.3mm. Each wheel contained an encoder with a resolution of 12 pulses/revolution. Using the gear ratio of 64, the expected resolution using encoders $\mathrm{R}_{\text{encoder}}$ can be calculated as:

$$\begin{aligned}\mathrm{R}_{\text{encoder}} &= \frac{(2)(\pi)(51.3\text{mm})}{12 * 64\text{pulses}} \\ &= 0.419\text{mm/pulse}\end{aligned}$$

B. Vision

Machine vision was used to get the robots ground truth pose. Steger's method outlined in [15] was chosen for its accuracy and speed. The specific implementation can be found in the commercial software Halcon by Mvtec. Using a least square's refinement strategy Steger's method has been demonstrated to be accurate to within 1/22 of a pixel and 1/100 of a degree. Over the field of view this gives a translation resolution of:

$$\mathrm{R}_{\text{Steger}} = \frac{1}{22}\text{pixels}\frac{2763\text{mm}}{1626\text{pixels}} = 0.077\text{mm}.$$

The number of pyramid levels for searching was set to 5. A greediness parameter can speed up the matching and has been set to 0.75.

The camera was calibrated using the method outlined in [20]. The perspective-n-point algorithm given in [21] was used to compute a world coordinate system, which is shown in the bottom left corner of Figure 1.

Descriptor based methods were compared against Steger's to determine how accurate they are. Specifically, the OpenCV implementations of SIFT, AKAZE and ORB were tested. For SIFT and ORB, default parameters were used. For AKAZE, the response threshold was changed from 0.001 to 0.0001 to allow it to detect enough key-points on reduced-size images. To match descriptors, a brute force matcher was used. The matcher returns the two closest points using k-nearest neighbors. The points can be kept or rejected by examining a distance metric and comparing it to a user defined threshold. For SIFT, the L2 norm is used to compute the distance between descriptors. Since AKAZE and ORB both use binary descriptors the hamming distance was used to compute the distance. Descriptor methods work well when there is a lot of texture on the object. Therefore a cereal box has been placed on the robot for easier matching.

For computing the rigid transform between the template and the image the OpenCV implementation of 'estimateRigidTransform' was used. Data from the encoders was gathered using the on-board Arduino micro controller. All Kalman filtering was done using Matlab.

VI. Results and Discussion

Here the results are discussed. First the results from odometry are discussed without sensor fuson. It is shown how odometry results in poor tracking over long periods due to the drift. Next pose estimation using machine vision is discussed. In particular, descriptor based methods are compared against each other to see which one is best in terms of accuracy and speed. Finally the results of two sensor fusion simulations are discussed.

A. Encoder Results

Figure 2 shows the result of encoder translation and Figure 3 shows the angle estimation. The encoder drift can easily be seen. By the end of the run, the robot is off by a distance of 85mm from ground truth and over 10 degrees.

B. Vision Results

For vision testing, two different scenarios were tested. First tracking was performed on the full size image. This results in high accuracy but also high detection time. Then the image was reduced using OpenCV's 'resize' function by 2.5 times (or multiplied by 0.4). This results in a loss of accuracy but tracking time is reduced. During the testing it was verified visually that the descriptor based methods were all less accurate than Steger's method. This was done manually by zooming in on the found position in the image. Steger's method was the only one that did not show any coordinate variance while the robot was motionless. This can seen by looking at Figure 4 which shows coordinate systems from both Steger's method (blue) and AKAZE (red). Here the robot is stationary. The blue coordinate system (Steger's) does not move. However the red coordinate system (AKAZE) bounces around. This justifies the choice of using Steger's algorithm as the ground truth.

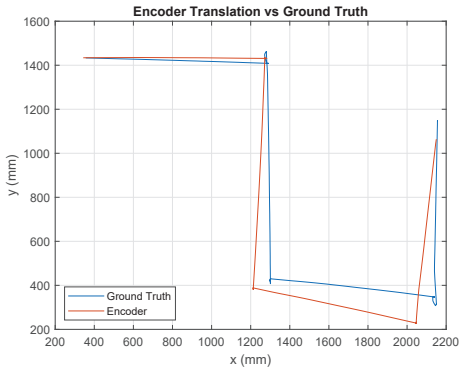

Fig. 2: Translation Estimation using odometry from wheel encoders.

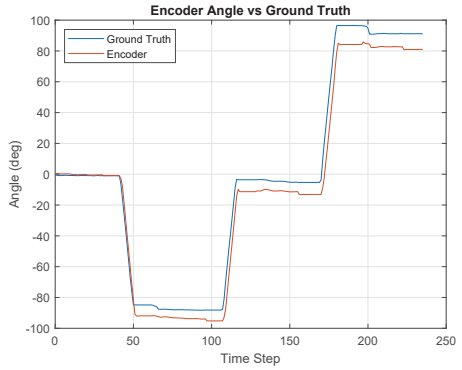

Fig. 3: Angle Estimation using odometry from wheel encoders.

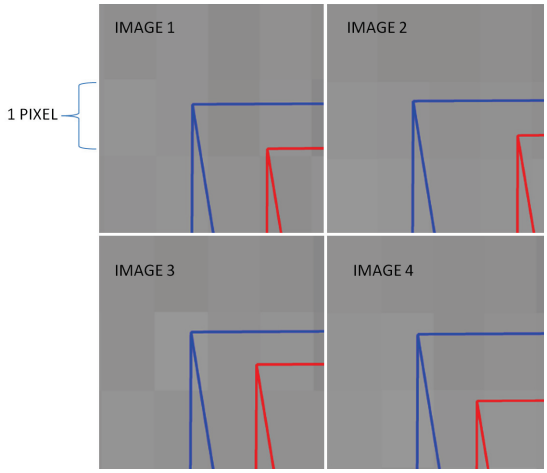

Fig. 4: Zoomed in images of a stationary robot. The blue coordinate system represents Steger's algorithm. The red is AKAZE. Note that no movement can be detected on the blue coordinates system. It remains at the same location. However movement can easily be seen on the red coordinate system, indicating it is not as accurate as the blue.

Figure 5 shows the root mean squared error (RMSE) results of vision-only tracking using three descriptor methods. The results are also displayed in Table I where L2 indicates the L2 norm. This was used to combine the errors in both the X and Y directions. The accuracies are reported relative to Steger's edge gradient method. The left part of the graph shows the accuracies on a full size image. The right side shows what happens if the image size is reduced by 2.5 times (multiply by 0.4). ORB is clearly the worst of the three. SIFT and AKAZE show similar accuracies. This result compares favorably with recent research shown in [22] which compares the differences between SIFT and AKAZE.

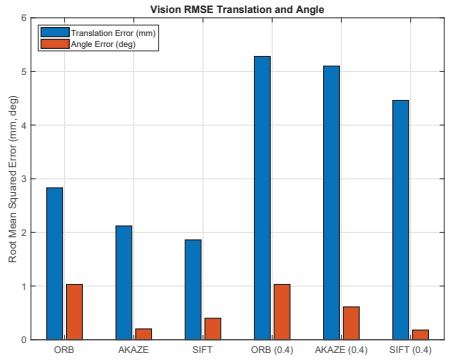

Fig. 5: Results of vision tracking. The left side bars indicate accuracy on a full size image. The right side bars show accuracies on reduced-size images (image size decreased by a factor of 0.4)

	Full Image		Reduced Image	
	L2 (mm)	Angle (deg)	L2 (mm)	Angle (deg)
ORB	2.82	0.68	5.28	1.03
AKAZE	2.11	0.20	5.1	0.61
SIFT	1.86	0.40	4.46	0.17

TABLE I: RMSE Errors

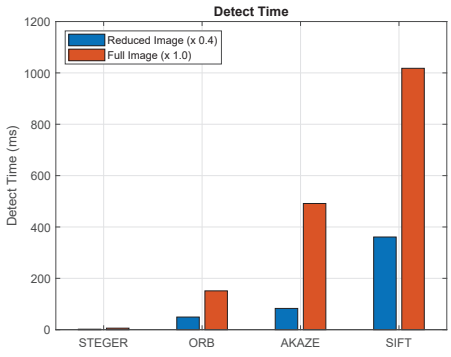

Fig. 6: Detect times for the various methods. The times indicate how fast the each algorithm can detect and locate in pixel coordinates. Two instances for each are shown: one on a full size 2.5Mp image, and one on a reduced size 1Mp image

(a) Full Image

	Time (ms)
Steger 1.0	6.0
ORB 1.0	151.2
AKAZE 1.0	491.4
SIFT 1.0	1017.9

(b) Reduced Image

	Time (ms)
Steger 0.4	1.95
ORB 0.4	49.3
AKAZE 0.4	82.8
SIFT 0.4	360.9

TABLE II: Detect Times

Not surprisingly, the RMSE increases almost proportionately to the image size. The only exception to this seems to be the angle of SIFT, of which the error has dropped by half. Figure 6 shows the detect times for the descriptor-based methods as well as Steger's for both the full image and reduced size images. Steger's method easily runs in real-time on the full size as well as reduced-size images. ORB is the next best. AKAZE can only run in real time on the reduced image. Although SIFT is the most accurate of the three descriptor methods, it cannot run in real time on a full image or reduced-image.

In summary, these results show that ORB is the only method that is suitable for real-time tracking for this particular problem. If the image size can be reduced, AKAZE could achieve real-time tracking. SIFT, although the most accurate of the three descriptor based methods, is not suitable for fast tracking. For planar applications, the descriptor based methods all have a difficult time achieving the sub-millimeter and sub-degree accuracy that Steger's edge based method achieves. Although for most tracking applications, the descriptor based methods achieve an accuracy that should be sufficient.

C. Kalman Filter Simulation 1 - High Vision Variance

It was shown in section V-B that the vision system has a resolution that is approximately 5.5 times better than the encoder. This might not always be the case. For example, if the encoder had a resolution of over 230,000 pulses per revolution like many industrial servo motors do, it could reach resolutions of over 50 times what the vision system could do. In order to simulate the scenario where the vision and encoder resolutions are more closely matched, Gaussian white noise is added to Steger's ground truth vision measurement. Specifically, translation and rotation noise with a standard deviation of 55mm and 10 degrees are added to the vision measurement, respectively. Then the EKF and UKF filters are applied.

Since the noise was added to the vision measurement, the matrix measurement covariance matrix noise R is given as

$$\mathbf{R} = \begin{bmatrix} 3062.2 & 0 & 0 \\ 0 & 2534.3 & 0 \\ 0 & 0 & 1.035 \end{bmatrix}.$$

The system covariance matrix Q was computed by comparing the ground truth with the odometry prediction at each time step and then calculating the variance of each state. This was

used as a starting point and then it was manually modified. The final matrix is given as

$$\mathbf{Q} = \begin{bmatrix} 16.74 & 0 & 0 \\ 0 & 21.51 & 0 \\ 0 & 0 & 0.000486 \end{bmatrix}.$$

Figures 7 and 8 show the translational and angular results of applying the EKF and UKF. Table III shows the root mean squared errors (RMSE) for all methods. The EKF and UKF show similar results. The EKF does slightly better in translation and the UKF does slightly better in angle. Overall the EKF and UKF show much better results than using vision or odometry individually.

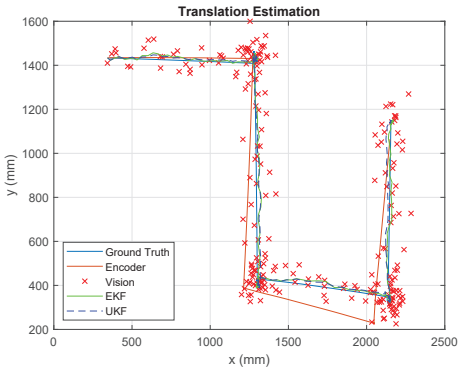

Fig. 7: Simulation 1 - EKF and UKF filters for translation tracking using vision and odometry.

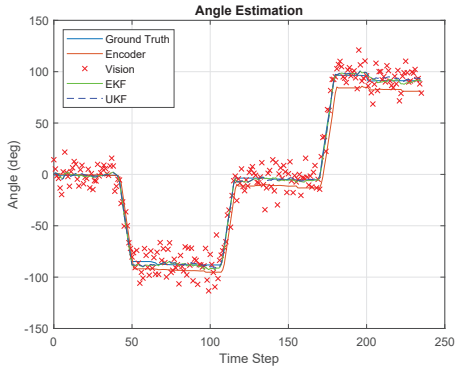

Fig. 8: Simulation 1 - EKF and UKF filters for angle tracking using vision and odometry.

	RMSE			
State	EKF	UKF	Odometry	Vision
X (mm)	16.1222	16.3718	72.5548	54.3846
Y (mm)	16.2234	17.1214	68.982	53.4756
Angle (deg)	2.68276	2.07099	8.76642	10.6263

TABLE III: Simulation 1 - High Vision Variance RMSE

D. Kalman Filter Simulation 2 - Low Vision Variance

The accuracy of descriptor based methods was shown in section VI-B. This section investigates whether the pose estimation can be improved further by using information from the encoders. In this simulation, translation and rotation noise with a standard deviation of 6mm and 4 degrees are added to the vision measurement, respectively. This produces vision RMSE results that are close to what the descriptor methods produced on a reduced size image shown in Figure 5. It will be shown how much these errors can be reduced by using the EKF and UKF.

The measurement noise covariance matrix R and the system noise covariance matrix Q were computed in a similar manner to the last simulation and are given as:

$$\mathbf{R} = \begin{bmatrix} 45.7 & 0 & 0 \\ 0 & 45.7 & 0 \\ 0 & 0 & 0.0055 \end{bmatrix}$$

and

$$\mathbf{Q} = \begin{bmatrix} 25.47 & 0 & 0 \\ 0 & 24.52 & 0 \\ 0 & 0 & 0.00082 \end{bmatrix}.$$

Figures 9 and 10 show the translational and angular results of applying the EKF and UKF. The EKF and UKF show very similar results. Although the EKF and UKF show better results than using odometry or vision individually, the result is not as pronounced as it was in the high variance simulation previously. In this simulation, the vision system is already very accurate. Nevertheless, both the EKF and UKF are able to use the prediction from odometry to obtain a better result than pure vision.

Table IV shows how much the EKF and UKF are able to lower the RMSE compared to using vision by itself. For the translation the RMSE is lower by about 1.2mm. For the angle it is lower by about 1 degree. In fact, by complementing vision with odometry, the accuracy of the full-sized image is obtained from the left side bars of Figure 5. This means that a reduced image can be used to speed up the vision recognition time while keeping the accuracy that was obtained using a full size image.

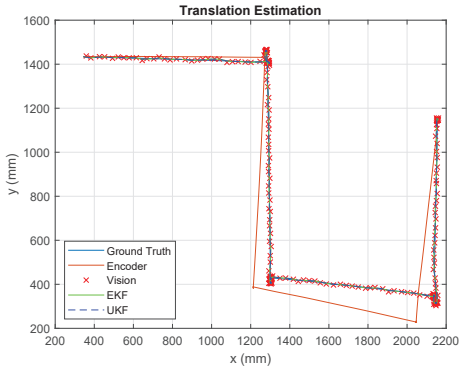

Fig. 9: Simulation 2 - EKF and UKF filters for translation tracking using vision and odometry.

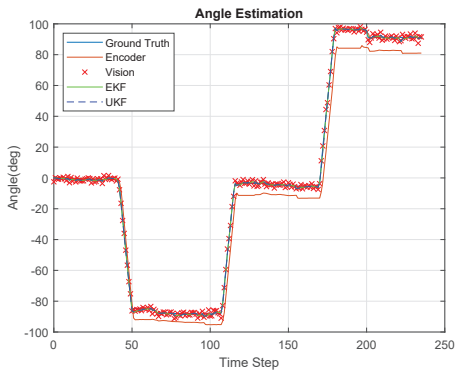

Fig. 10: Simulation 2 - EKF and UKF filters for angle tracking using vision and odometry.

	RMSE			
State	EKF	UKF	Odometry	Vision
X (mm)	2.67621	2.7069	72.5548	3.90613
Y (mm)	3.15117	3.16597	68.982	4.38743
Angle (deg)	0.817416	0.794677	8.76642	1.34938

TABLE IV: Simulation 2 - Low Vision Variance RMSE

VII. CONCLUSIONS

This paper showed the accuracy and detection time that could be obtained using descriptor based matching for tracking a mobile robot in a plane. RMSE of less than 3mm and 1degree accuracy were reported. Of the three descriptor based methods, ORB seems to be the best choice for planar tracking. Although not as accurate as SIFT and AKAZE, its accuracy should be sufficient for most planar tracking applications. Furthermore it runs over 3 times faster than AKAZE and almost 7 times faster than SIFT. If the vision results are complemented with the EKF and UKF, the accuracy results can be improved, though only marginally. This is because using descriptor based methods is already much more accurate than odometry. The effect may be more pronounced if higher resolution encoders are used.

VIII. ACKNOWLEDGEMENT

We would like to thank our industrial partner, Dr. Robot (Markham, Ontario), for their technical support and guidance. Dr. Robot provides cost-effective, high performance robots for industrial, commercial, and academic end-users alike. Product and contact information for Dr. Robot may be found here: www.drrobot.com

REFERENCES

[1] S. Bonadies, N. Smith, N. Niewoehner, A. S. Lee, A. M. Lefcourt, and S. A. Gadsden, "Development of proportional-integral-derivative and fuzzy control strategies for navigation in agricultural environments," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 6, 2017.

[2] S. A. Gadsden, "An adaptive pid controller based on bayesian theory," *2017 ASME Dynamic Systems and Control Conference*, 2017.

[3] J. Palacin, J. A. Salse, I. Valganon, and X. Clua, "Building a mobile robot for a floor-cleaning operation in domestic environments," *IEEE Transactions on instrumentation and measurement*, vol. 53, no. 5, pp. 1418–1424, 2004.

[4] J. Palacin, I. Valganon, and R. Pernia, "The optical mouse for indoor mobile robot odometry measurement," *Sensors and Actuators A: Physical*, vol. 126, no. 1, pp. 141–147, 2006.

[5] S. K. Hong and S. Park, "Minimal-drift heading measurement using a mems gyro for indoor mobile robots," *Sensors*, vol. 8, no. 11, pp. 7287–7299, 2008.

[6] B.-S. Cho, W.-J. Seo, W.-s. Moon, and K.-R. Baek, "Positioning of a mobile robot based on odometry and a new ultrasonic lps," *International Journal of Control, Automation and Systems*, vol. 11, no. 2, pp. 333–345, 2013.

[7] X. Huang and N. Houshangi, "Lane following system for a mobile robot using information from vision and odometry," in *Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on*, pp. 001009–001013, IEEE, 2011.

[8] M. Shahriari and M. Biglarbegian, "A new conflict resolution method for multiple mobile robots in cluttered environments with motion-liveness," *IEEE transactions on cybernetics*, 2016.

[9] Y. Dai, K. S. Choi, and S. G. Lee, "Adaptive formation control and collision avoidance using a priority strategy for nonholonomic mobile robots," *International Journal of Advanced Robotic Systems*, vol. 10, no. 2, p. 140, 2013.

[10] E. Johnson, E. Olson, and C. Boonthum-Denecke, "Robot localization using overhead camera and leds.," in *FLAIRS Conference*, 2012.

[11] J. Jisarojito, "Tracking a robot using overhead cameras for robocup spl league," *School of Computer Science and Engineering, The University of New South Wales, Tech. Rep*, 2011.

[12] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2, pp. 1150–1157, Ieee, 1999.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*, pp. 2564–2571, IEEE, 2011.

[14] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.

[15] C. Steger, "Occlusion, clutter, and illumination invariant object recognition," *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, no. 3/A, pp. 345–350, 2002.

[16] S. A. Gadsden and S. R. Habibi, "A new robust filtering strategy for linear systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 135, no. 1, 2013.

[17] S. A. Gadsden, Y. Song, and S. R. Habibi, "Novel model-based estimators for the purposes of fault detection and diagnosis," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, 2013.

[18] S. A. Gadsden and A. S. Lee, "Advances of the smooth variable structure filter: Square-root and two-pass formulations," *Journal of Applied Remote Sensing*, vol. 11, no. 1, 2017.

[19] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[20] C. Steger, M. Ulrich, and C. Wiedemann, *Machine vision algorithms and applications*. John Wiley & Sons, 2017.

[21] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.

[22] O. Andersson and S. Reyna Marquez, "A comparison of object detection algorithms using unmanipulated testing images: Comparing sift, kaze, akaze and orb," 2016.