# Vision Based Steering on Husqvarna Research Platform

Alice Karlsson
Jakob Nilsson,
Elias Nehmé
Zongshang Pang

*Abstract*— **This report discusses how computer vision techniques are utilized to accurately steer a lawn mower in a straight path under various conditions for systematically covering pre-defined areas such as a lawn. This was implemented by extracting SIFT-points in each video frame and comparing these between frames in order to get the yaw angle of the lawn mower. The yaw estimated by the camera gets fused with the yaw readings of the internal simulated odometer, which depends on a real sensor, the wheel encoder. The fused yaws act as the input to the PID controller, which aims to keep the mower going straight without being affected by surroundings. The resulting control algorithm helped the mower going straight in conditions that it normally would drift for example in a scenario when the diameter of one wheel is changed due to cut grass sticking to it. Throughout the whole implementation, the Husqvarna platform is used as the supporting working environment as well as the interface between the algorithms and the mower.**

## I. Introduction

Nowadays, most of the lawn mowers available on the market base its path algorithm on some randomness, and this is not a very effective approach since it covers the same area more than once before it has covered the whole lawn. By mimicking the methodical way of manually cutting the grass in straight lines, the mowing can be optimized. The first step to achieve this is to ensure that the lawn mower can move in straight lines regardless of external conditions with the aid from a camera. No article of this specific implementation has been found, but multiple articles in related areas have been investigated.

There has been a project that aimed to solve a similar problem. C. Zhu et. al. [1] proposes a solution that uses ultrasonic sensors and triangulation which corrects the path by comparing the robot's current angle with the expected angle.

Another report by B. Madsen et. al. [2] preforms a triangulation using three objects and a single camera to receive the position of the robot relative to the global environment.

C. Pati et. al. [3] present a solution to a related problem. There is a robot that is following another robot with the aid of camera input. The robot identifies the other robot's relative position using a camera and depth sensors. The distance between the robots is then minimized by a PID controller. The PID controller's error term is equivalent to the distance.

Another similar project uses a camera to detect lanes by investigate the contrast in the images and with this information apply a controller in order to follow the lanes correctly [4].

M.Kqbayas et. al [5] have developed a method for using visual servoing for robots. They suggest using landmarks and measuring the distance to the objects with a camera. A controller is implemented and thus strive to minimize the distance.

This project has been mostly inspired by the paper by Y.Erdem et. al. [6]. They present a method for motion estimation using a feature-based approach, by finding different interest points and calculating the velocity vectors for the movement of these points instead of every pixel in the images. The images are divided into quadrants and the motion vectors are added into four final vectors describing the movement of the camera.

SIFT points are commonly used for comparing features in different images, from which the position or orientation of a robot is extracted. To create a person-following robot, T.Sonoura et. al. [7] uses features points and calculates the velocity vector. They also calculate the distance to the feature points and uses the distance and direction data to turn and follow an object.

Another project uses SIFT points to create line segments, with which the relative camera pose is estimated. The change and similarities for the position in two images can be obtained using homography and epipolar constraints [8].

## II. Problem formulation

The purpose of this project was to use camera vision in order to help the Husqvarna research platform navigate in a limited area. The Husqvarna research platform should move in straight lines with guidance from a camera. If no external factors affect the lawn mover it will run straight, however, the system is missing a feedback loop thus making it vulnerable to external factors. These external factors include hard slopes, wet grass or mowed grass sticking to the wheels and thus changing the radius of the wheels that can affect the steering and make to mower drift. The task was to ensure, despite external factors, the mower to run on a straight path making it easier to mow a lawn in a systematic manner.

## III. Theory

This section presents relevant theory for this project and aim to ease the understanding of the project. Most of the concepts presented here, as RANSAC and SIFT-point extraction, are later implemented using the Python module openCV which have all these functionalities.

### A. RANSAC

RANSAC stands for random sample consensus and is a iterative method that finds optimized model parameters despite the data containing outliers. A RANSAC loop is repeated multiple times, first a random set of measurements are used to calculate model parameters. The loss is then calculated using these parameters to see whether they are any good, and the process repeated. The parameters with the lowest loss is kept and this is the optimized model parameters [9].

### B. SIFT points

The SIFT algorithm consists of a feature point detection, feature points localization, orientation assignment and feature descriptor generation. By finding points of interest in images point-to-point correspondences can be obtained. One way to localize features points is to find blobs in the image, by using difference-of-Gaussians filters. It is important that the points of interest have a detectable size and a distinct center. Around each of these points the SIFT algorithm divides the input images into different sections and analyzes the gradient of the different patches. This by creating and analyzing gradient histogram, first the gradient of each pixel in the input image is defined and then classified with regards to orientation and added into eight different bins. This results in a histogram vector and this vector can be used to compare images. The SIFT descriptor is a vector containing all of the histogram vectors for the correspond patch of the input image. This can be used in order to enable feature extraction, by finding the similar descriptors between two images we can find point-to-point correspondences with for example RANSAC [9].

### C. The camera equation

The camera equation that describes the relation between the 2D point in the image and the 3D point in the scene,

$$\lambda x = PX = K[R\ t]X \tag{1}$$

includes of the matrix P that describes the orientation and position of the camera. $P = K[R\ t]$ were $R$ is the rotation matrix and $K$ is the specific camera parameters that transform the image into the real image coordinate system. $X$ is a 3D-point and $x$ is the projection of X in the image [10].

$$K = \begin{pmatrix} \gamma f & sf & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

The K matrix depends on camera parameters such as the focal length $f$, principal point $(x_0, y_0)$, the skew $s$ and aspect ratio $\gamma$. In order to obtain the K-matrix a camera calibration is needed [10].

### D. Estimating homographies from feature correspondences

By using feature points in two different images, $u_i$ and $u_i'$, and using RANSAC the homographies H can be estimated using

$$u_i{}' = Hu_i \tag{3}$$

And then point to point relationship between two images is obtained. The system can be written as

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} * \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u\prime \\ v\prime \\ 1 \end{bmatrix} \tag{4}$$

This relation can be used in a DLT (Direct Linear Transfomration) in order to extract the values of the elements in the homography. To do so four or more correspondences are needed of $u_i'$ and $u_i$ [11].

For each homography the camera matrices can be obtained. And in the normalized case the first camera matrix is always the same. It is a four by four matrix with the identity matrix to the left and zeros in the rest. The second camera matrix cannot be explicitly determined however and up to four solutions are obtained as in Equation 6. The variables $U$ and $V$ are obtained from the single value decomposition of the homography, $W$ are described in Equation 5, and $u_3$ is the last column of $U$.

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$P_2 = \begin{bmatrix} UWV^T & u_3 \end{bmatrix} \text{ or } \begin{bmatrix} UW^TV^T & u_3 \end{bmatrix} \tag{6}$$

$$\text{or } \begin{bmatrix} UWV^T & -u_3 \end{bmatrix} \text{ or } \begin{bmatrix} UW^TV^T & -u_3 \end{bmatrix}$$

To determine which second camera matrix is valid, the 3D points has to be computed and the depth of them checked. If the corresponding depth to the camera matrix is negative, then the camera matrix is invalid [10].

## IV. Methodology

This section gives a summary of the different parts of the whole solution to the control problem using computer vision. The implemented solution with the added ROS nodes and camera algorithm can be found in the repository **hrp-cv** on Github [14]. The whole solution is shown in Figure 1 and the different components are discussed in further detail in this section.

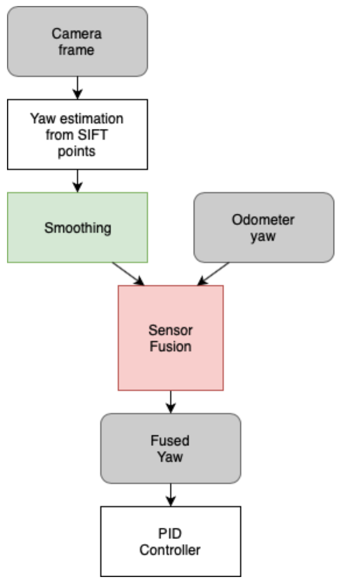

Fig. 1. The architecture of the problem solution.

## A. *The Husqvarna Research Platform*

Husqvarna Research Platform is a lawn mower especially developed for research on autonomous robots with some open source software to control it [12]. The robot is equipped with ROS Kinetic software and has integrated collision sensors. The motors has encoder sensors that read the rotation of the wheels. The platform is also equipped with integrated battery and associated charging station. On this platform a Raspberry Pi 3 was installed to which a camera was connected. The camera used in the project was a Raspberry Pi camera v2 and it was mounted in the front bumper of the lawn mower. The camera module is compatible with all models of Raspberry Pi and has a Sony IMX219 8-megapixel sensor. For video it supports 640x480p90, 1080p30 and 720p60 and for static images it has the resolution of 3280 x 2464 pixels [13]. In this project that resolution was limited to 640x480 with 10 frames per second due to hardware limitations.

## B. *ROS infrastructure*

To effectively communicate with the robot, a ROS system (Robot Operating System) was developed and integrated with the already existing ROS software of the Husqvarna robot. ROS is a high level operating system for all kinds of robots, increasing modularity and re-usability of code. ROS is based on having different nodes that can either subscribe or publish information to each other.

Two computing units were used for this task, a local Raspberry Pi on the robot itself and a Linux computer remotely. The Raspberry Pi had the camera and outputted the video stream to the ROS network. This video stream was then processed on the remote PC with computer vision algorithms, and outputted a yaw estimate back to the robot. The robot then had a controller node that used this yaw and the odometer readings to create a control signal. Due to lack of processing power on the Raspberry Pi, this second computer was needed for the computationally heavy computer vision algorithms.

## C. *Camera calibration*

The camera calibration was preformed using a package in ROS. The camera calibration is essential in order to receive the image center, focal length and lens distortions parameters so the correct camera camera parameters, $K$, can be obtained. These camera parameters where then used in the algorithm to obtain the Euler angles of the robot.

## D. *Yaw estimation algorithm using vision*

There are a few different methods for estimating the Euler angles of the lawn mower from a video stream. All methods use SIFT points in some manner which is explained in the theory section. One method that was tested and implemented split the images in quadrants and computed a mean motion vector in each quadrant of the image. In this way the direction and size of the movement of the lawn mower is estimated but the absolute angle is not obtained which is needed to accurately control the robot.

The other method that was used in the final version of the software computes SIFT-points in each image in the video stream. The amount of SIFT-points extracted in each image was limited to a maximum of 500 points and other parameters to find the SIFT-features were changed for the given application such as the minimum distance between the computed points. The points of the current image in the video stream are then compared to the points in the previous image. From these two sets of points a homography is computed with points that correspond to one another in the two pictures. All points in the two sets of points can not be used since new points can be detected in the most current picture that did not exist in the previous one. This correspondence detection is done with a RANSAC algorithm with a threshold on what is to be considered a inlier of 2 pixels. The homography can then be used to compute the rotation and translation matrices. Here up to four valid solutions are obtained for the rotations and translations. In order to figure out which of these matrices are correct the 3D points had to be triangulated by constructing camera matrices from the rotation and translation. These 3D points are checked to see if they lie in front of the cameras and the rotation and translation matrices with the most 3D points in front of the cameras where used. From the rotation matrix obtained in the previous step Euler angles can be calculated. From these Euler angles a yaw can be obtained. The yaw signal was then smoothed in order to reduce the noise that came from the shaking of the robot when in operation. The pseudo code for the algorithm is shown in Algorithm 1.

**Algorithm 1:** Pseudo code for the yaw estimation.

```
Data: K: The camera parameters
Data: curr_im: The current frame
Data: prev_im: The previous frame
Data: prev_sift: The SIFT points in prev_im
Data: euler_angles: The current orientation, the
      accumulated euler angles.
Result: yaw_angle Is outputted for each new
        image.
1  while camera ACTIVE do
       /* Get the current image                    */
2      Fetch curr_im
       /* Compute the SIFT points in the fetched
          image                                    */
3      curr_sift = find_sift_points(
        curr_im)
       /* Find point correspondances and compute
          the homography between them              */
4      point_corr = RANSAC(curr_sift,
        prev_sift)
5      H = homography(point_corr)
       /* Extract the rotation and translation
          matrices from H, up to four of each can
          be obtained. Here the camera parameters
          K are needed.                            */
6      R_all,t_all = decompose_H(H, K)
       /* Find the valid rotation by computing 3D
          points' depth                            */
7      R = find_valid_rotation(R_all)
       /* Compute euler angles from R              */
8      curr_euler_angels =
        compute_angles(R)
       /* Add the change in Euler angles to the
          total orientation                        */
9      euler_angles += curr_euler_angels
       /* Save image and SIFT-points to next
          iteration                                */
10     prev_im = curr_im
11     prev_sift = curr_sift
       /* Extract yaw from the Euler angles and
          smooth the signal by a Blackman window
          mean.                                    */
12     yaw_angle = smooth(euler_angles(2))
```

## E. Sensor fusion of camera and internal odometer

There are yaw readings from a simulated odometer which gives fairly reliable values most of the time. However, one problem that the odometer has is that it does not yield good readings when the wheels go through abnormal movements, for example, running on a bumpy road. This is because the odometer readings are derived from the data generated by the wheel encoder. While, theoretically, the existence of the camera is to help alleviate the bad effect caused by this situation. Consequently, one strategy to utilize both the sensors is to fuse the data from them. The predication step is defined with a constant angular velocity model. Two update steps for the two sensors are designed, respectively, although the measurements models are the same, i.e., only measuring yaw readings. The state vector thus contains the yaw angle, $\varphi$, and the yaw rate, $\omega$. The motion model can be seen in Equation (7) and the measurement model for the camera in Equation (8) and for the odometer in Equation (9). Since both the motion model and the measurement model are linear, the linear Kalman filter suffices to solve the problem.

$$\begin{bmatrix}\varphi(k+1)\\ \omega(k+1)\end{bmatrix} = \begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix}\begin{bmatrix}\varphi(k)\\ \omega(k)\end{bmatrix} + \begin{bmatrix}q_\varphi(k)\\ q_\omega(k)\end{bmatrix} \tag{7}$$

$$y_c(k) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\varphi(k)\\ \omega(k)\end{bmatrix} + v_c(k) \tag{8}$$

$$y_o(k) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\varphi(k)\\ \omega(k)\end{bmatrix} + v_o(k) \tag{9}$$

With these models, the Kalman prediction is calculated according to Equation (10), while the Kalman updates are estimated according to equation (11).

$$\begin{aligned}\hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1}\\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q\end{aligned} \tag{10}$$

$$\begin{aligned}\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k - H^T x_{k|k-1})\\ P_{k|k} &= P_{k|k-1} - K_k S_k K_t^T\end{aligned} \tag{11}$$

where $K_k = P_{k|k-1}H^T S_k^{-1}$ is the Kalman gain and $S_k = HP_{k|k-1}H^T + R_k$ is the innovation covariance. The prediction is sent into the two update steps sequentially, first through the camera update and then the odometer update. The process and measurement noises $Q, R_{camera}, R_{odometer}$ were then tuned based on the actual performance of the mower.

## F. The controller

For computer vision steering, a new ROS node, `cam-controller` was implemented. This ROS node contains a discrete-time PID controller that outputs velocity signals to the robot via the topic `cmd_vel`. The node takes yaw input signals from the camera through the computer vision node `cam-steer`, and takes in the odometer yaw estimate trough the odometer node `odom`. The `cam-steer` subscribes to the video stream from the Raspberry Pi camera and publishes an yaw angle, published on the topic `cam_yaw`. Refer to the figures folder in the github repository for an overview of the ROS system [14].

## G. Validation

The validation test consists of two parts and both test preformed without cutting of the grass and indoors. The first test is set in one of the hallways and is testing the SIFT algorithm and the whole system under the most ideal conditions. To pass this test the robot should be able to register sift points along its path and stay on a straight path for 10 meters.

The second test is more difficult and tests some of the real conditions for the lawn mower. It is the same as the first test but on one of the wheels a large cardboard piece attached to it, this in order to simulate the cut grass sticking on the wheels and thus chaining the radius of the wheels. To pass this test should be able to register SIFT points along its path and stay on a straight path for 10 meters.

## V. Results

This subsection will present the results of the validation test and the lawn mowers general performance.

### A. Validation test 1

The results from validation test 1 can be seen in Figure 2. Here the internal odometer in the mower drifts by four degrees even though the mover is driving straight. This is however adjusted by the yaw readings from the camera algorithm and the fused yaw is around zero degrees.

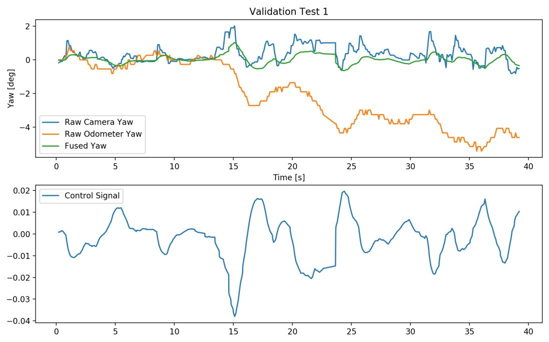

Fig. 2. The first validation test.

The variance parameters for the sensor fusion algorithm were set according to equation (12)

$$\begin{aligned} Q &= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \\ R_c &= 7 \\ R_o &= 20 \end{aligned} \tag{12}$$

The PID controller was tuned according to equation (13)

$$\begin{aligned} P &= 2 \\ I &= 0.1 \\ D &= 0.001 \end{aligned} \tag{13}$$

### B. Validation test 2

The results from validation test 2 can be seen in Figure 3. As expected the internal odometer drifts even more, to about 15 degrees, when the diameter of one wheel is changed but the fused yaw is still around zero degrees.

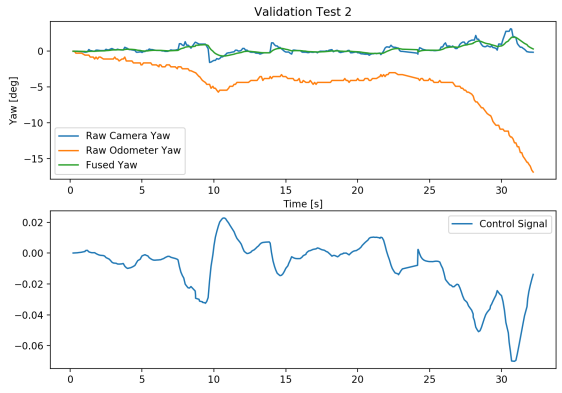

Fig. 3. The second validation test.

The variance parameters for the sensor fusion algorithm were set according to equation (14).

$$\begin{aligned} Q &= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \\ R_c &= 5 \\ R_o &= 10^3 \end{aligned} \tag{14}$$

The PID controller was tuned according to equation (15)

$$\begin{aligned} P &= 2 \\ I &= 0.1 \\ D &= 0.001 \end{aligned} \tag{15}$$

## VI. Discussion

As seen in both validation test 1 and 2, the camera algorithms implemented improved the yaw angle estimation. In validation test 2 where rough terrain was simulated the camera was even more crucial as the odometer was drifting away and the slip error was accumulating over time. On way this SIFT point solution could be improved is to introduce another control parameter to the system, namely the horizontal position of the robot. As only the angle is being controlled, the horizontal position of the robot could potentially drift over time. By implementing an 2 dimensional Linear Quadratic Regulator (LQR) and augmenting the model with horizontal position this could be mitigated. This was not done due to time constraints and since it was not crucial to prove that the concept of computer vision steering is viable.

Even though the results presented show a great improvement over the odometer, it should be noted that without global positioning the result will not be accurate enough as the small errors will accumulate over time. A systematic mowing of a lawn is a big task that takes a lot of time, and without any global localization or positioning the error in the estimated position would accumulate too much. Absolute reference points such as static objects or map localization are the most intuitive and effective solutions to these problems, as they counteract an accumulating error over long time spans. The reason the SIFT-point solution was implemented is due to the short implementation time as well as the robustness of the algorithm. The SIFT-point solution does not have any specific requirements on the surroundings or the surface. The algorithm only required installing one camera to the robot. A natural limitation of the algorithm as mentioned above is however that over time the positional error accumulates due to lack of absolute reference points.

To optimize the cutting time and coverage, a vision algorithm could be implemented on the Husqvarna Research Platform to identify objects and be able to make decisions and steer based on these. The cutting can be optimized using landmarks and performed with object detection using a camera, which should help navigate the mower. For example, a behavior that makes the lawn mower drive towards an object could be developed, and when reaching the object, the mower could make a turn and identify another object, and drive towards it. This can be achieved using the YOLO-algorithm to identify objects such as trees and a fence. Due to time constraints, this was not implemented on the Research Platform and the object detection was only tried on a separate computer.

Another approach to solving this problem is to create a map using the SLAM algorithm and a camera. By mapping the surroundings, the lawn mower movement can be known and optimized using for example an algorithm similar to the Hamiltonian cycle. To utilize this, accurate states of the mower should be provided. This can be achieved using sensor fusion algorithms since we have multiple sensors available at hand, such as the camera, the wheel encoders, and the simulated odometer. Considering the turns the mower has to make, a nonlinear motion model involving angles need to be designed, which requires us to use nonlinear sensor fusion algorithms, such as the UKF, the CKF, and the simple EKF. However, all these potentially beneficial algorithms need more computational power, compared to the crude solution presented in this report.

If the customers desire a stripy lawn similar to how a football field is cut, a grass roller could be attached to the robotic platform. A grass roller bends the blade of the grass and thus creating dark or light ”stripes” depending on which direction the lawn is mowed in. With the aid of a camera, the robot can see where it already has cut the grass since the cut grass will, depending on how the roller is attached, be darker or lighter. With this information, a lane detection algorithm could be implemented to make the lawn mower follow the lanes and move in straight lines as well as not cut the grass on the same spot twice.

## VII. Conclusions

The Raspberry Pi has too little computing power to run the camera algorithm locally. Therefore, the communication between a work station and the raspberry pi has been built in order to run laboring tasks on the former and control the robot by the latter. This method has several defects. Firstly, the communication can only be stable within short distance. For a large lawn, where the mower should freely run, the communication could be lost anytime. Secondly, it is tedious to equip the mower with a specific work station just for processing computer vision algorithms, and this will make buying the mower together with a workstation less attractive.

Moreover, in order to implement this in some sort of production setting the algorithm has to be made more efficient or a larger computing unit has to be used. The camera used in this project seems to be sufficient for this kind of problem since a camera with higher resolution would introduce further issues with the CPU used. However sensor fusion is crucial in order to obtain a good result, not just a camera. Overall this is a viable solution that helps the robot on rough terrain where the odometer can not be relied on. In order to get a more generally robust solution that is reliable over longer time, a localization or navigation system should be implemented to give absolute positional information. Using Object detection is another immediate way to use an absolute reference point to improve the global positioning. By using object detection, the robot can have a static reference point which it follows, which in turn counteracts the problem of an accumulating error over time.

## VIII. Acknowledgment

We would like to thank the supervisor Martin Fabian for guidance throughout the project.

## References

[1] C.Zhu, M.He, P.Chen, K.Sun, J.Wang and Q.Huang, 2018. ”Navigation for Indoor Robot: Straight Line Movement via Navigator”. Mathematical Problems in Engineering Volume 2018. [Online]. Available at: https://www.hindawi.com/journals/mpe/2018/8419384/

[2] C.Madsen, C.Andersen, J.Sorensen, ”A Robustness Analysis of Triangulation-Based Robot Self-Positioning” [Online]. Available at: https://www.cs.cmu.edu/~motionplanning/papers/sbp_papers/integrated4/madsen_triangulation.pdf

[3] C.Pati, R.Kala ”Vision-Based Robot Following Using PID Control”. [Online]. Available at: https://res.mdpi.comtechnologies-05-00034

[4] A.Thomson, ”A PATH FOLLOWING SYSTEM FOR AUTONOMOUS ROBOTS WITH MINIMAL COMPUTING POWER” [Online]. Available at: http://www4.cs.umanitoba.ca/~jacky/Teaching/Courses/74.795-LocalVision/ReadingList/thomson-thesis-path-following.pdf

[5] M.Kqbayas, Y.Miyamqtq, K.Mitsjhashij,Y.Tanaka. ”A METHOD OF VISUAL SERVOING FOR AUTONOMOUS VEHICLES”. [Online]. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=509435&tag=1

[6] Y.Erdem, F.Galip, I.Furkan Ince, Md.Haidar Sharif, ”Estimation of Camera Ego-Motion for Real-Time Computer Vision Applications”. International Journal of Scientific Research in Information Systems and Engineering, Vol.1, no.2, Dec.2015

[7] T.Sonoura, T.Yoshimi, M.Nishiyama, H.Nakamoto, S.Tokura and N.Matsuhira, ”Person Following Robot with Vision-based and Sensor Fusion Tracking Algorithm”. [Online]. Available at: http://cdn.intechweb.org/pdfs/5205.pdf

[8] I.Reisner-Kollmann, A.Reichinger, W.Purgathofer. ”3D Camera Pose Estimation using Line Correspondences and 1D Homographies”. [Online]. Available at: https://www.vrvis.at/publications/pdfs/PB-VRVis-2010-019.pdf

[9] O. Enqvist, ”Lecture Notes in Image Analysis” Chalmers University of Technology.

[10] R.Szeliski. 2010. ”Computer Vision: Algorithms and Applications”. [Online]. Available at: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf

[11] T.Opsahl. ”Lecture 4.3 Estimating homographies from feature correspondences” [Online]. Available at: https://www.uio.no/studier/emner/matnat/its/nedlagte-emner/UNIK4690/v16/forelesninger/lecture_4_3-estimating-homographies-from-feature\-correspondences.pdf?fbclid=IwAR1uNYWcQHTSg-lIi-wKlPywZO_QwXwVXt4cJupimeNw6FgW2AKd38eZrPg

[12] Husqvarna Research Platform. 2017. Husqvarna Group. https://github.com/HusqvarnaResearch/hrp

[13] [Online]. Available at: https://thepihut.com/products/raspberry-pi-camera-module

[14] Project Repo. 2019. https://github.com/Eliacus/hrp-cv