Integrating disparity and edge detection algorithms to autonomously follow linear-shaped structures at low altitude

Camiel R. Verschoor*, and Arnoud Visser*
*Intelligent Systems Laboratory Amsterdam
Universiteit van Amsterdam

***Abstract*—One of the main requirements in enabling autonomous flight of Micro Aerial Vehicles is the ability of autonomous navigation. One possible solution to solve this navigation problem is to use vision-based line-following algorithms. Such vision-based algorithm could rely on the various linear structures, which are present in the human constructed environment. Edge detection and disparity estimation have proven to be strong algorithms for the detection of nearby objects. However, these detection algorithms have their weaknesses. The candidate lines found by both algorithms are input for the Probabilistic Hough Transform, which is used to select the best candidate and to determine a directional vector from this line. This paper is a survey for the experimental circumstances to fairly test both algorithms for a line-following task, which is one of the challenges of the Indoor Micro Aerial Vehicle competition.**

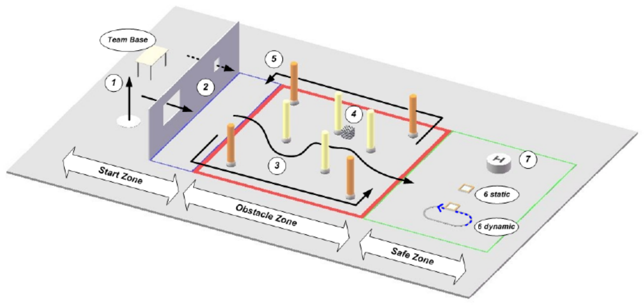

Figure 1. The indoor arena of the International Micro Aerial Vehicle (IMAV) competitions (Courtesy IMAV committee).

I. Introduction

In robotics one of the main goals is to develop mobile robots that can operate autonomously in the real world environment. These autonomous robots have various purposes and are used for a wide range of applications such as inspection, exploration and rescue. Even though reasonable developments have been made in the robotics field, robots cannot operate autonomously in all circumstances the real world can provide.

One of the initiatives to promote the developments in autonomous robots is the International Micro Aerial Vehicle (IMAV) conference and competition, which is the basis of the Iran Open Flying robot competition. The indoor challenge of this competition consists of an arena with several mission elements (see Fig 1). One of the mission elements is 'Following a path' (marked with ⑤ in Fig 1). For this mission element a path is laid out around the, four fixed poles which mark the obstacle zone. The performance of the autonomous robot is scored based on the number of laps done flying over the path.

The experiments described in this paper are performed prior to the publication of the IMAV 2013 competition rules. The major difference is that in our study the line can follow a 3D trajectory, while in the IMAV 2013 competition the line is on the ground. This means that one of our algorithms, based on edge detection, could be applied directly. The algorithm based on disparity has to wait on more advanced challenges in future competitions.

AR.Drone SLAM [1] is a development framework for the Parrot AR.Drone developed and proposed by N. Dijkshoorn. This framework runs off-board and contains a real-time Simultaneous Localization and Mapping (SLAM) implementation based on a down-pointing camera. Therefore, it allows a Micro Aerial Vehicle (MAV) to know its position and movement in the environment by generating a feature map of the environment so the MAV can localize itself on this map. Furthermore, the framework facilitates control for a 3D mouse or a keyboard, and enables the generation of visual and elevation maps. The algorithms described in this study are an extension of this framework.

Indoor navigation is possible based on Visual SLAM, but when clear navigation clues are available in the environment it is beneficial to make use of those clues (with potential advances in efficiency and robustness). Human constructions (walls, fences) have often linear structures, which could be exploited with a line-following algorithm. There are various approaches for line detection based on vision. Edge detection and disparity estimation have proven to be strong algorithms for the detection of nearby objects [2], [3]. Linear structures (i.e. power lines) have a specific brightness and height, therefore, edge detection and disparity estimation are suitable algorithms. Edge detection finds sharp brightness changes in an image. A disparity map indicates the apparent motion in an image and shows the foreground objects brighter, denoting greater apparent motion and lesser distance.

Both algorithms are implemented as pre-processing alternative for the first stage. In the second stage a Probabilistic Hough Transform (PHT) is applied to extract the best line from the pre-processed image. This is a common feature extraction technique used in the field of robotics [4], [5], [2]. PHT determines a directional vector for navigation purposes and can be used to make the platform move accordingly. Based on the output of the approach the platform will navigate according to the instructions. Furthermore, this paper is a survey for the experimental circumstances to fairly test both algorithms for a line-following task.

Therefore, the main research question is to examine how edge detection and disparity estimation can be combined to strengthen each other in a line following task. This main research question is divided up in the following sub-questions:

- What is the optimal configuration for the optical sensors of the platform to follow a line?
- What are the experimental settings that demonstrate the strength and weaknesses of both algorithms clearly?
- What is the performance and robustness of different vision-based methods to navigation over a linear structure in an indoor environment?

In this paper the quality of an edge detection and disparity estimation algorithm will be separately examined to determine where these can strengthen each other when combined. This provides an essential component for autonomous navigation using a single camera.

Section II gives an overview is given over the related research regarding line-following on unmanned aerial vehicles. The approach of this paper is given in section III. In section IV the experiments are illustrated and in section V the results will be presented and discussed. In section VI the conclusion of this paper will be presented and directions for future research will be proposed.

II. Related Work

This section gives an overview of the related research that has been done regarding autonomous navigation for unmanned aerial vehicles. Small stable quadcopters have become affordable and research on this platform is moving towards more intelligent and autonomous applications. Various autonomous navigation methods have already been investigated in the field. Various related autonomous applications are listed and briefly described.

Corridor following is a task performed by robots for autonomous navigation in indoor environments (i.e. an office). Recently, a new navigation method for the MAV [2] was introduced by navigating in the indoor environment based on single image perspective cues. This is in contrast with previous approaches where a 3D model is built before planning and control. This method first classifies the type of indoor environment and then the MAV navigates through the environment using vision algorithms based on perspective cues to estimate the desired direction. The environment is detected through a confidence classifier, where estimates of the stair and corridor algorithms are used to compute the confidence values. The highest confidence value, above a threshold, is deemed as the current environment. The corridor and stair algorithm both use the Canny edge detector and probabilistic Hough transform to acquire the line segments. The corridor algorithm then tries to determine the vanishing point in the image. The stair algorithm on the other hand tries to determine the middle of the stairs by looking at horizontal line segments of the stairs.

Power line inspection is an essential task for the maintenance of the electric grids, which is difficult due to the range of grid distributions. Over the last years rapid development has been made driven by the need for fast, accurate, safe and low-cost power line inspection [6]. Important requirements for power line inspection robots are to maintain position above power lines and to navigate over them, which is also the case for small aircraft. A vision-based power line following method has already been proposed [7]. This algorithm makes use of the Hough transform to detect the line segments in the image. The method makes the assumption that power lines run vertically through the screen. Therefore, the method uses the vertical line segments to adjust its position. The main advantage of power line inspection is that the MAV can make use of the redundancy in power line design as it can follow three parallel lines. Therefore, it is easier to navigate over power lines as when one of the conductors is not detected; the MAV can adjust its position according to the other two.

Road-Following is a task performed by robots for autonomous navigation in outdoor environments. Small autonomous aircraft [4] have already been able to follow a road based on real-time road detection and localization. The algorithm uses multiple vision-based methods to detect the road and the lane markings. First the Bayesian Pixel Classifier, an advanced alternative of the HSV-color filter, is applied on every pixel to see whether it belongs to the road. The classifier makes use of a database of RGB values of over 20000 pixels. Then connected-component analysis is used by labeling pixels in order to remove noise in the image and detect connected regions. After detecting the road in the image the lane markings are detected by the Bayesian pixel classification algorithm. Then a Hough transformation is applied to test multiple candidate lanes. This results in a rough discretization of the lanes. Lastly, robust line fitting, least-trimmed square, is applied to finalize the position and orientation of the center lane markings. This produced encouraging results; however, improvements of the algorithm can still be made.

Elementary Motion Detectors is an approach presented in a previous IMAV competition by the BioMAV team [3]. Their approach was to combine motion information provided by Elementary Movement Detectors (EMDs) with edge detection. EMDs are useful for UAVs due to the detection of temporal and motion effects caused by the flight of the platform. The motion information is generated by the EMD implementation [8], which is barely dependent on differences in contrast and color. In this approach EMDs are applied to improve the image segmentation as borders of relevant objects produce higher responses. Rotational movements of the drone will lead to edge enhancements and translational movements to larger apparent motion of objects closer to the drone. The constant self-motion will provide steady and a reliable source of information. Although the EMD is useful, it rarely gives a complete picture of the environment as they rely on contrast differences. Therefore, the EMDs alone are not enough to segment the image and provide the vehicle with sufficient information. However, the EMDs provide an abundance of additional information to the traditional approach. For this reason, a combination of edge detection and motion detection is used to detect objects in the environment. In this study, the same combination is proposed, although the motion detection is based on a disparity map.

III. Algorithm

In this section the various algorithms that are used for edge and motion detection for this paper are discussed.

A. Main Approach

The main approach of this paper for line-following navigation consists of three phases. These are the following phases:

In the **pre-processing phase**, the edge detection and disparity estimation algorithms are applied to the image in preparation for the feature extraction phase.

The **feature extraction phase** extracts objects, in this case lines, from the pre-processed image by using a probabilistic Hough transform.

In the **navigation phase**, the lines found are interpreted and controls are given to the platform.

The system is a closed loop system, as shown in Fig. 2, and will loop through these phases, so that the system keeps on following the line. The approach is real-time and integrated in the AR.Drone SLAM [1] development framework. In this paper the edge detection and disparity estimation were independently implemented in order to examine their strengths and weaknesses. In the following sections these phases will be described.

$$\begin{tikzcd}
Camera \arrow[r, "Frame"] & Pre-processing \arrow[d, "Binary\ image"] \\
World \arrow[u, "Input"] & Feature\ extraction \arrow[d, "Line\ segments"] \\
Flight\ controller \arrow[u, "Flying"] & Navigation \arrow[l, "Controls"']
\end{tikzcd}$$

Figure 2. Schematic overview of the approach

B. Pre-processing

There are two types of pre-processing method implemented in this paper. In this section the implementation of the edge detection and disparity estimation will be described.

1) Edge Detection: In this paper edge detection is implemented in the following way: The **color filter** is applied to the image and filters for the color of the line. The resulting binary image is combined with the result of the Canny edge detector. A color filter was chosen as this paper focuses on simple linear structures. However, a texture filter would also be an appropriate choice since the system flies at a low altitude, textures can easily be detected.

Gaussian smoothing is applied to prepare the image for the Canny edge detector so that small distortions will be removed. The resulting image is provided to the Canny Edge Detector.

Subsequently, the **Canny edge detector** is applied to the image in order to find the edges indicated in a binary image. The Canny edge detector was chosen as it already was successful in previous studies regarding MAV navigation [2]. For this reason, this method is suitable to be surveyed for use in combination with motion detection.

By **combining** the resulting images of the color filter and Canny edge detector the number of candidates for the line can be reduced. Combining the two techniques filters out noisy edges.

The result of this algorithm is a binary image containing the edges of the images. On this binary image feature extraction will be applied.

2) Disparity estimation: In this paper disparity map is estimated based on the motion of the platform:

Finding features can be done by applying the Shi-Tomasi algorithm [9] on the first image. This algorithm has a decent feature definition for the next (matching) stage where the optical flow is determined.

Optical flow can be calculated by providing the found features to the Lucas-Kanade pyramidal implementation in order to find equivalent features in the next frame. Due to the pyramidal implementation, the algorithm can find features with any distance between them. This circumvents the assumption that motion between frames is very small. This algorithm results in a set of feature pairs. The average optical flow is also computed to examine the average motion.

The **fundamental matrix** is the estimation of the constraints on the 3D motion of the camera based on the 2D motion of the pixels in the image. The fundamental matrix can be used to rectify the images. The fundamental matrix is determined in combination with the RANSAC algorithm as the optical flow algorithm can provide noisy data. RANSAC filters out this noise and tries to find the best model for the given set of features. This results in an optimal fundamental matrix for the given features.

Rectification is warping one image from its image plane towards another image plane. The fundamental matrix is required for rectifying the images without calibration. The algorithm used for this is Hartley's algorithm, which attempts to find homographs that map epipoles to infinity while minimizing the computed disparities, resulting into rectified frames of the original frames.

Stereo Matching can be performed when a common image plane is found. After determining the fundamental matrix and rectification of the original frames a fast pass stereo matching algorithm [10] is applied to compute the disparity map. This allows keeping the implementation real-time. This step results into a disparity map indicating foreground object brighter denoting greater motion and lesser distance.

Foreground objects can be found by applying a certain threshold over the disparity map, which in this case is the line on a certain height above the ground.

The result of this image is a binary image containing the foreground objects that have passed the threshold. On this binary image feature extraction will be applied.

C. Feature Extraction

The feature extraction method that is applied in this paper is the probabilistic Hough transform. This method is computationally fast and deals with noise cause by the pre-processing stage. In the case of this paper the probabilistic form of the Hough transform [11] is used to extract lines from the image. This form takes a only a subset of the found edges to extract features. The algorithm results into a array of lines that satisfy the criteria of the method.

D. Navigation

To determine the controls for the platform the found lines examined. In this paper a simple algorithm is used that merges lines that have a similar slope and a short distance between them. The merging is done by taking the average of both lines. This reduces the amount of lines found in the image by the feature extraction method. Thereafter, the line that is the longest and has most confidence is selected. This is measured by calculating the length of the line and counting the amount of lines this line represents. The weighted sum is taken to determine the best line found in the image.

After finding the line in the image the platform is required to navigate towards the line and follow it. This is the process of monitoring and controlling the movement of the vehicle. In order to navigate towards the line the trajectory has to be calculated. The line is kept in the middle of the screen while navigating over it, so that the line is not lost during navigation. This gives the system the following tasks:

1) Move towards the line in the correct orientation
2) Navigate over the line while making adjustments to keep it in the middle of the screen.

The line that is found gives coordinates (x_1, y_1) and (x_2, y_2). The linear equation $y = mx + b$ can be determined from these coordinates. Given the formula of the line the adjustment can be calculated, which results in an angle θ and a translation x. However, flying towards the line can be done in several ways. In this paper the platform first flies on top of the line and then corrects its orientation parallel to the line. When the platform is hovering in the correct orientation above the line it will move forward, while correcting its orientation accordingly to the line. If the platform moves too far from the line the platform will do the above step again. This is the control strategy for line-following navigation.

Furthermore, a region of interest is defined when the line is found based on the movement of the platform and location of the line. This is done to speed up the algorithm by decreasing the data it processes.

IV. Experiments

This section investigates the experimental settings that demonstrate the strength and weaknesses of both algorithms clearly. The proposed experiments aid to answer the previously determined research questions (see section I). The various configurations, experiments and evaluation criteria will be discussed in this section.

A. Platform

For this paper, the Ascending Technologies Pelican and Parrot AR.Drone were both considered for the evaluation of the vision-based algorithms. To evaluate the algorithms a stable platform with on-board stabilization was required. The Pelican has several advantages such as: on-board processing, modular design and high payload. However, the system has no indoor on-board stabilization and had technological difficulties with the wireless connection. Therefore, the Parrot AR.Drone was chosen for the experiments.

B. Optimal Camera Configuration

To perform vision-based line-following navigation the optimal camera configuration has to be determined. In the current configuration of the AR.Drone it has a bottom and front camera. For optimal line-following navigation the camera has to point obliquely to the ground in front of the MAV. The platform should look ahead so it can adjust its course on time. Neither the bottom nor front camera provides this view as the bottom camera has a small field of view and the front camera looks too far ahead. This makes it impossible for the platform to navigate over a line as the platform cannot adapt itself to changes in time and some cases it is not even able to detect the line. Therefore, the current configuration of the cameras is not suitable for navigation.

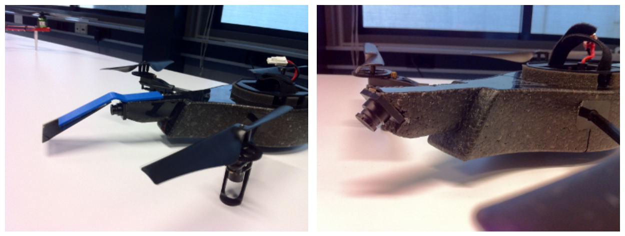
Figure 3. Possible solutions to solve the camera configuration problem, the mirror construction (left) and the modification of the AR.Drone (right)

The optimal solution for the camera configuration would be the pan-tilt camera of the Pelican. The pan-tilt camera can change the angle of the camera and therefore it is suitable for flying at various altitudes. Since the experiments are only indoors the platform only flies at low altitudes between 1-2 meters. Therefore, an angle of approximately 45° is considered and tested in this paper. To change the angle, this paper constructed and tested the following two solutions (see figure 3) to change the angle:

A **mirror construction** that changes the view of the front camera. This construction can be placed on top of the front camera. The mirror has an angle towards the ground, which gives the camera the mirrored ahead view.

A **modification of AR.Drone** with respect to the position of the front camera. Due to the fact that the AR.Drone is made from Styrofoam its simple to modify the angle of the front camera. By cutting away Styrofoam it is possible to set the camera under a different angle.

C. Experiments

In order to investigate the strengths and weaknesses of both vision algorithms this paper examined two type of experiments. The experiments (see figure 4) challenge both the algorithms. The following experiments examine the strengths and weaknesses of both algorithms: In the first experiment the weakness of the disparity estimation algorithm is challenged by an orange line on the ground. In the second experiment the edge detection algorithm is challenged by a hanging blue colored line in the air above a blue background.

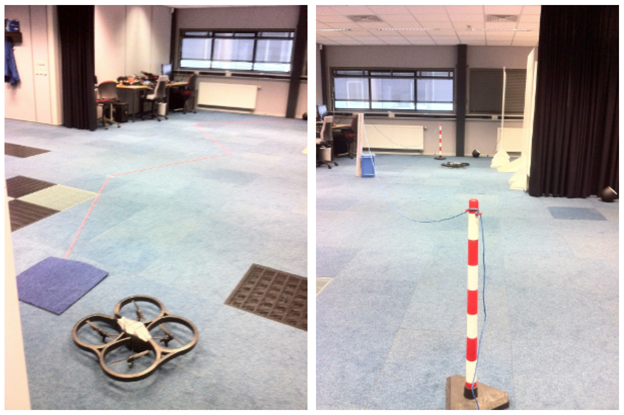
Figure 4. On the left the setup of experiment and on the right the setup of experiment two

These experiments are challenges for both algorithms as the motion detection algorithm finds the line on the basis of distance. In contrast with the edge detection algorithm that finds differences in brightness intensity. The strengths are also illustrated by these two experiments. In experiment one there is high contrast between the object and the background and in experiment two there is difference in height of the object and the background. These are possible situations where the two algorithms can strengthen each others weaknesses.

D. Evaluation Criteria

To evaluate the edge and motion detection algorithms criteria are set, to compare their performance. In this paper the methods are evaluated based on the following two criteria:

- The number of lines detected in the images.
- The number of lines with the correct direction.

Both algorithms will be evaluated on the same recorded datasets[1] so both algorithms experience the same circumstances. The datasets are recorded, while flying the platform manually in order to prevent inaccuracies caused by navigation errors of the simple navigation method.

V. Results and Discussion

In this section the results of the various experiments will be presented and discussed.

A. Camera Configuration

Before carrying out the experiments the optimal camera configuration of the platform was determined. This is done because the standard camera configuration is not suitable for line-following navigation. The following two solutions were constructed and tested in this paper:

Mirror Construction
The mirror construction was designed and made on the basis of the field of view of the AR.Drone. The construction consists of a tailor made mirror that is mounted to a 3-Dimensional printed frame. The frame is placed on top of the AR.Drone as shown in figure 3. This configuration resulted into distorted images during flight and a not optimal angle towards the ground. The distorted images were caused by the vibrations of the platform during flight. The material of the frame is constructed with a thermoplastic material called Acrylonitrile Butadiene Styrene (ABS). The ABS material is flexible, which causes the construction to vibrate rapidly. Furthermore, the camera cannot be set to an angle of $45°$ without losing part of the image as the field of view of the camera is too large for this angle. Due to this the camera overlaps with the bottom camera and a smaller area is covered by the camera.

Modification of AR.Drone
The modification to the AR.Drone was based on the field of view of the AR.Drone. For the modification the Styrofoam of the AR.Drone was cut away under an angle of $45°$. This gave the AR.Drone the configuration as shown in figure 3. This configuration resulted into clear images during flight and an optimal angle towards the ground. The Styrofoam compensates for the vibrations of the platform. Therefore, the platform takes sharper images comparing to the mirror construction. Additionally, by modifying the AR.Drone the camera can be put at any angle. However, the modification cannot be undone, which is a disadvantage.

B. Edge Detection

After determining the camera configuration, the edge detection algorithm was tested on several images. The color filter and the Canny edge detector in combination with a Gaussian blur were applied over these images. The result of the color filter and the Canny edge detector were combined in an image, which was provided to the probabilistic Hough transform. This resulted into an array of possible line segments. The best line was determined and indicated on the image. In figure 5, typical results can be seen of this algorithm.

These results look promising; however, the line is detected simply as there are high intensity differences in the image. The combination of the color filter seems to work well for filtering out the noisy edges. Note that the probabilistic Hough transform is performing well on the pre-processed images.

C. Disparity estimation

The motion detection algorithm was also tested before being evaluated. The optical flow was calculated on the basis of the features that were found by the method of Shi-Tomasi [9]. From the corresponding points the fundamental matrix was determined using RANSAC to discriminate the outliers. The fundamental matrix served as input for Hartley's algorithm that rectified the images. These images were used to compute the disparity map, which was thresholded to find the closest foreground object. The resulting image served as input for the probabilistic Hough transform, which extracts the lines

[1]The datasets are publicly available.

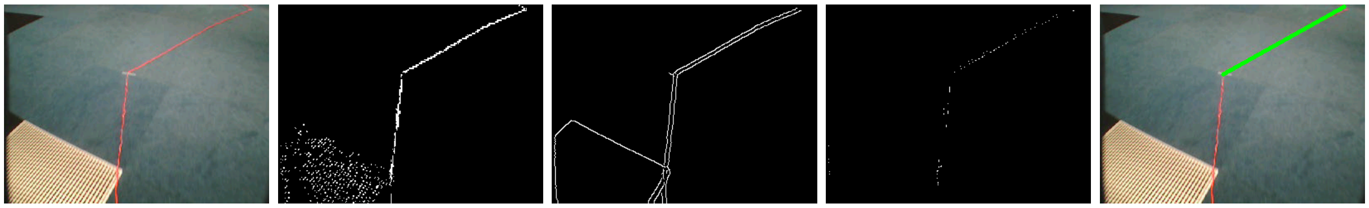

Figure 5. Respectively: Original Image, color Filter and Canny Edge Detector, Combined Image and the Probabilistic Hough Transform result.

from the provided images. The best line was determined and indicated. In figure 6, a sample result of this algorithm is shown.

The result of the motion detection algorithm looks reasonable. Nevertheless, this was tested on an ideal situation where the line is hanging close to the platform. It seems that the algorithm sometimes is unable to detect the line due to the lack of motion between the frames. This is in line with the theory behind Monocular Stereo Vision, as then it is hard to see any motion.

D. Experiment One

In experiment one the platform was acquired to fly over an orange line on the ground. The edge and motion detection were both evaluated on the same dataset. The vision algorithms were evaluated on the amount of times a line was detected and whether the detected line had the correct direction. This gave the following results:

Edge Detection

Performance	True	False
Detected	528 (97.8%)	12 (2.2%)
Direction	503 (93.1%)	37 (6.9%)

Motion Detection

Performance	True	False
Detected	53 (9.8%)	487 (90.2%)
Direction	37 (6.9%)	503 (93.1%)

These results evidently show that this experiment is a weakness of the Motion Detection algorithm. All the features in the image move in the same motion, therefore, the algorithm only detects background. This results in a disparity map without any depth. On the other hand the performance of the edge detection algorithm is satisfying. The results obviously show the strength of edge detection and the weakness of motion detection. Therefore, edge detection can strengthen motion detection.

E. Experiment Two

In experiment two the platform was acquired to fly over a hanging blue colored line with a blue background. The edge and motion detection were both evaluated on the same dataset. The vision algorithms were evaluated on the amount of times a line was detected and whether the detected line had the correct direction. This gave the following results:

Edge Detection

Performance	True	False
Detected	69 (29.0%)	169 (71.0%)
Direction	58 (24.4%)	180 (85.6%)

Motion Detection

Performance	True	False
Detected	96 (40.3%)	142 (59.7%)
Direction	87 (36.6%)	137 (63.4%)

These results show that motion detection is performing better than edge detection. However, the motion detection algorithm is not as robust as edge detection in experiment one. This is because between some frames in the dataset was not enough distance making the method unable to detect motion in the image. This is causing the weak performance of the algorithm as it was able to detect continuously the line. Furthermore, the results show that low gradient intensity differences are hardly detected by an edge detector. Due to the fact the motion detection algorithm is able to detect the line when there is enough motion it can strengthen edge detection.

All in all, the above results show the weaknesses and strengths of the edge and motion detection algorithms. Edge detection is good at detecting differences in gradient intensity of the image. On the other hand motion detection is good at detecting foreground objects assuming that there are enough motions in between the frames. The results show that combining the two would strengthen others weaknesses. Therefore, fusing these two algorithms is a suitable approach for the pre-processing stage of a probabilistic Hough transform.

VI. Conclusion

In this paper a two stage approach is demonstrated, that combines edge detection or disparity estimation with a probabilistic Hough transform. Where edge detection finds sharp brightness changes, disparity estimation finds the foreground objects. The results of the experiments show the strengths and weaknesses of both algorithms. The results also show that there is no optimal vision-based approach. Therefore, fusing algorithms will make vision-based methods more robust.

The optimal configuration of the camera for the line-following task was determined in this paper. The front camera was set to an angle of 45° by modifying the AR.Drone. This was the best and most stable solution in order to solve the camera configuration problem. A mirror construction was also examined. However, this was unstable in the sense that the mirror was vibrating during flights. Additionally, this construction could not obtain the optimal angle.

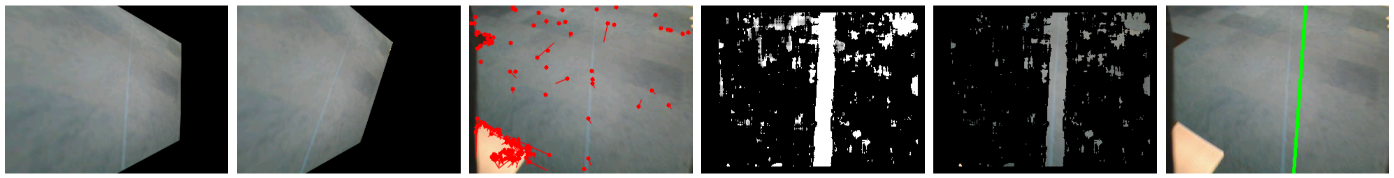

Figure 6. Respectively: Left Image, Right Image and Optical Flow, Disparity Map, Disparity Map on Image and the Probabilistic Hough Transform result.

Two experiments where designed to show the strengths and weaknesses of the edge and motion detectors. In the first experiment, motion detection was challenged and in the second experiment, edge detection. The results of the experiments show that these algorithms strengthen each other's weaknesses.

The performance and robustness of both vision-based algorithms showed promising results and can be used for a line-following task. However, when edge and motion detection are combined this can lead to even more robust vision-based algorithms.

Conclusively, this paper examined the strengths and weaknesses of two different kind of pre-processing techniques, namely, edge and motion detection. From this examination can be concluded that indeed edge and motion detectors can strengthen each other. Therefore, this paper recommends a combination of these two algorithms to autonomously follow linear shaped structures in a landscape.

VII. Future Research

For a MAV to be able to navigate autonomously in an indoor or urban environment it is necessary to use a navigation method as proposed in this paper. For navigation accurate controls are required, which are provided by the visual system. The controls provided by the visual system should be accurate as it is not affordable to crash. The visual system of the MAV can be more robust by combining the strengths of these vision-based algorithms. Also other methods can aid the robustness of vision-based algorithms: Probabilistic methods such as the Kalman Filter [12] can aid keeping track of the location of the line. A Kalman Filter is an algorithm, which uses a series of measurements observed over time, containing noise, produces estimates of unknown variables that tend to be more precise than those that would be based on a single measurement. The navigation method should be improved so it can handle multiple lines. A smarter navigation module leads to more intelligent applications. The vision-based algorithms should be tested in a wide range of environments to determine their robustness. The Iran Open flying robot competition is an excellent opportunity to benchmark the edge detection algorithm prior to the international competition.

Acknowledgements

This research is partly funded by the EuroStars project SmartINSIDE. We would like to thank Gerald Poppinga for his support and guidance. Furthermore, we are grateful to Nick Dijkshoorn for providing of his development framework, AR.Drone SLAM. We also like to thank Rob van Holstein for the construction of the 3-dimensional printed frame of the mirror construction. Likewise, we would like to thank Christian Muller for helping collecting the dataset.

References

[1] N. Dijkshoorn, "Simultaneous localization and mapping with the ar.drone," Master's thesis, Universiteit van Amsterdam, July 2012.

[2] C. Bills, J. Chen, and A. Saxena, "Autonomous mav flight in indoor environments using single image perspective cues," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 5776 –5783.

[3] P. Gerke, J. Langevoort, S. Lagarde, L. Bax, T. Grootswagers, R.-J. Drenth, V. Slieker, L. Vuurpijl, P. Haselager, I. Sprinkhuizen-Kuyper, M. Otterlo, and G. de Croon, "Biomav: bio-inspired intelligence for autonomous flight," in *Proceedings of the International Micro Air Vehicle conference and competitions, 12-15 September (IMAV 2011)*, 2011.

[4] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta, "Vision-based road-following using a small autonomous aircraft," in *IEEE Aerospace Conference Proceedings*, vol. 5, March 2004, pp. 3006 – 3015.

[5] Z. Li, Y. Liu, R. Hayward, J. Zhang, and J. Cai, "Knowledge-based power line detection for uav surveillance and inspection systems," in *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, nov. 2008, pp. 1 –6.

[6] J. Katrasnik, F. Pernus, and B. Likar, "A survey of mobile robots for distribution power line inspection," *Power Delivery, IEEE Transactions on*, vol. 25, no. 1, pp. 485 –493, jan. 2010.

[7] I. Golightly and D. Jones, "Visual control of an unmanned aerial vehicle for power line inspection," in *Advanced Robotics, 2005. ICAR '05. Proceedings., 12th International Conference on*, july 2005, pp. 288 – 295.

[8] T. Zhang, H. Wu, E. Borst, K. Kühnlenz, M. Buss, and T. U. München, "An fpga implementation of insect-inspired motion detector for high-speed vision systems," in *IEEE International Conference on Robotics and Automation (ICRA*, 2008, pp. 335–340.

[9] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on.* IEEE, 1994, pp. 593–600.

[10] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proceedings. 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '99)*, 1999, pp. 318 –325.

[11] N. Kiryati, Y. Eldar, and A. Bruckstein, "A probabilistic hough transform," *Pattern Recognition*, vol. 24, no. 4, pp. 303 – 316, 1991.

[12] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina, Chapel Hill, NC, USA, Tech. Rep., 1995.