Optical Flow based Obstacle Avoidance for Real World Autonomous Aerial Navigation Tasks

ROBRECHT JURRIAANS
UNIVERSITEIT VAN AMSTERDAM

UNIVERSITEIT VAN AMSTERDAM

Optical flow based obstacle avoidance for real world autonomous aerial navigation tasks

R.C. JURRIAANS

5887380

BSc THESIS

Credits: 15 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam

Faculty of Science

Science Park 904

1098 XH Amsterdam

Supervisor:
Arnoud Visser

Room C3.237
Science Park 904
NL 1098 XH Amsterdam

8th August 2011

Abstract

In this thesis a method is proposed for aerial autonomous navigation tasks based on monocular stereo vision using the optical flow between frames. First, the optical flow is derived which serves as the basis for creating a disparity map that can be used as a path finder. The 3D reconstruction can be determined up to a projective transformation. To estimate ranges in the 3D reconstruction a time to contact method is proposed. This method provides depth estimates for a limited number of points in the camera view. The quality of this method is compared to standard stereo vision. The robot used for these tasks is the Parrot AR.Drone and the tasks are part of the IMAV 2011 competition.

Keywords: Optical Flow, Stereo Vision, SLAM, Navigation, Monocular Vision

Contents

1 Introduction

> *In regione caecorum rex est luscus.*
> *(In the land of the blind, the one-eyed man is king)*
> — DESIDERIUS ERASMUS, ADAGIA (III, IV, 1496)

This is a famous quote by Dutch philosopher Desiderius Erasmus referring to the superiority of the entity with the mildest disadvantage within a group of entities with greater disadvantages. This greatly applies to autonomous robots that are usually equipped with noisy, difficult to interpret sensors, which give an incomplete perception of the environment. But these robots have to make do, because without these sensors they would be incapable of any meaningful autonomous behaviour. However, in recent years sensors have become increasingly better performing, smaller and especially cheaper [4]. For most types of robots this is especially prevalent, since there are little to none constraints to the amount and types of sensors to equip. This is not the case when it comes to micro aerial vehicles (MAV), for stability and centre of gravity are far more important for correct movement. It is therefore desirable to have as few or as small as possible sensors equipped to the MAV.

Having as few sensors as possible leads to the thought whether or not it would be possible to use only one sensor. One range finder would essentially not be enough, since this would only allow the MAV to determine its distance to a possible obstacle without any form of recognition of the type of obstacle. It is possible to use GPS-like positioning, but this would require external help while navigating indoors. This leads to the thought that it is necessary to incorporate a camera. When using one camera it becomes possible to calculate optical flow which provides a basis for monocular stereo vision.

1.1 Motivation

From a practical point of view using optical flow for both the sensor model and the motion model in autonomous navigation liberates engineers from incorporating multiple sensors onto an autonomous vehicle, since optical flow can be derived using only one camera. Only needing one sensor has two very major advantages: cameras are quickly becoming cheaper and optical flow can be calculated from only one camera; cameras are also becoming smaller and lighter, which makes it relatively easy to add to an autonomous vehicle.

Apart from needing less physical sensors on the autonomous vehicle this could potentially also reduce computational complexity, since optical flow only has to be computed once for one sensor instead of combining multiple sensors and computations. Optical flow is already used to aid autonomous navigation, but mainly for estimation of the motion model and for lower scale navigational problems such as aerial stability and time-to-collision calculation.

Optical flow has been shown [17, 10, 14] to be capable of determining the rotational quantity of a motion. This would still leave the translation unknown,

but this can be estimated from marker tracking. It was shown that optical flow is also capable of marker tracking. Combining these different capabilities of optical flow to create a hybrid technique for determining structure with the limited resolution of the AR.Drone camera and using this 3D map of the environment for autonomous navigation in an aerial navigation task has not been done before.

In this paper the quality of optical flow based stereo vision will be examined to provide an essential component to autonomous behaviour using one camera. The robot that is used to provide the images is the AR.Drone as seen in figure 1 and the task to be completed is given by the IMAV 2011 competition.

FIGURE 1: *Top view of the AR.Drone*

1.2 IMAV 2011 Indoor Navigation Task

The IMAV 2011 competition exists from three different competitions, two of which are indoor tasks and one which takes place outdoors. One of the indoor tasks and the outdoor task are based on path following around obstacles. The other indoor task is based on navigation through an unknown environment. The IMAV competition provides an effective and established forum for dissemination and demonstration of recent advances in MAV-technology.

Although the dimensions of both the first wall and the building, as described in the scenario, are known, the exact configuration is not. This means that for true autonomous behaviour it is necessary to map the environment or perhaps it is possible to use context-aware obstacle avoidance to keep track of the subtasks.

Scenario: your team is asked by law enforcement agencies to secure

evidence held in a private building on a guarded compound. Your team's MAV is small enough to slip past the guards and enter the building. Based on intelligence information you must first confirm that you have entered the right building by visual recognition of known features in the building. Then, your team will proceed to pick up a piece of evidence from the desk and bring it outside the building. A retreat through the opening in the roof is advised to escape detection. Once outside the building the evidence must be placed in a designated zone where it will be picked up by an agent. While the evidence is analysed the MAV is required to land and wait for further instructions at a predefined spot. As the sun rises the MAV will quickly heat up and you are advised to anticipate this change. After a short waiting period, your MAV will be instructed to fly again and end the mission.[1]

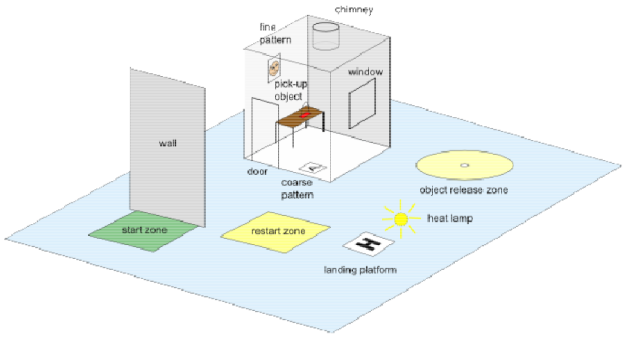


FIGURE 2: *Layout of the course for the navigational task*

Completing the various subtasks awards points, these points are then combined with the level of autonomy and the size of the MAV to yield a score. There are no points to be awarded for remote-controlled MAV's in order to promote autonomous behaviour.

Overview of Paper First, some of the methods for navigating using only one camera are examined. Then in section 3 the algorithm for stereo vision using optical flow is explained. In section 4 The use of the algorithm is described together with some of the main issues with the algorithm. Section 5 will give the results of the algorithm. Finally, in section 6 and section 7 a conclusion is drawn and discussed.

[1] http://www.imav2011.org/

2 Related Work

There are various ways to navigate using only one camera. Each has its own advantages and disadvantages.

Blob-based Obstacle Avoidance Blob-based obstacle avoidance does not result in a map of the environment and does not allow the agent to localise itself [27]. What it does do is that it enables the agent to navigate past various obstacles. It does this by sampling the lower part of the image and assuming that this pattern or plain colour is a uniform representation of the floor. If a binary image is created where the pattern of the floor is separated from other colours and patterns in the image it becomes possible to determine the furthest reachable point by simply taking each column and finding the first zero-value from the bottom.

This method can be applied continuously and in real time but has some issues for most real world tasks. One of the main problems with this method is the assumption that the floor is uniformly one pattern or one colour. Another important assumption is that the blob representing the floor is in fact the largest blob. This is already a confining assumption but for autonomous flight even more so, because of the fact that aerial vehicles are above the ground and therefore the floor is almost certainly not the largest blob.

Single Image Perspective Cues Another possibility to use a single camera for navigation would be to use the Canny edge detector and apply a probabilistic Hough Transform to acquire line segments. The resulting line segments can be used to distinguish certain environments [3] and choosing a goal within that environment. When the agent is in a hallway many lines will intersect at the same point, being the vanishing point. Finding this vanishing point not only reveals the nature of the environment, but also provides an excellent cue for planning. Staircases can be distinguished in a similar fashion as having many horizontal lines. The mean of all the end points in the set of horizontal lines provides the directional cue for navigating a staircase, as can be seen in figure 3 which have been obtained using the AR.Drones front facing camera in flight.

FIGURE 3: *Various staircases and their directional cues, Courtesy [3]*

This method requires fairly distinct and confined environments in order to work. Since the IMAV 2011 indoor competition does not require the following of hallways and has complexer environments it becomes difficult to distinguish the various environments and to extract directional cues.

Stereo Vision It is possible to use two or more cameras and estimate the depth map using stereo vision. Although less robust than most range sensors [4], it only requires two cameras. Another great advantage of using cameras is the capability to apply various vision algorithms such as object recognition without the need for additional sensors.

To extract three dimensions from 2D images it is necessary to have more than one 2D image. In stereo vision multiple cameras are used to get different views of the same environment. If more than one 2D image is used it becomes possible to compare the relative position of objects in both images and if the relative position of the cameras is known it becomes possible to find the distance to the object. To find the distance to an object it is necessary to first remove radial and tangential lens distortion in order to have images which are pure projections. It is then important to adjust for the angles and distances between cameras to rectify the images so that the images are coplanar and row-aligned. It is true that for some robots the assumption that both images are coplanar and row aligned is viable. However, this is not the case for aerial robots such as the AR.Drone which rely on rotation to move around skewering the images in the process. When the images are undistorted and rectified it is possible to find correspondences between images where the disparities are the differences in x-coordinates on the image planes of the features. When the disparity between images is known we can find the distances by means of triangulation [26, 8, 23, 13] which can be seen in fig 4. The disparity map gives the differences in x-coordinates on the image planes. Therefore it is necessary to have all images row aligned, so that the same feature has the same y-coordinate in each image.

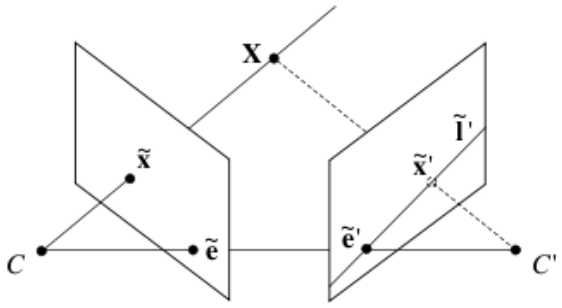


FIGURE 4: *Triangulation of a point X given two views provided by cameras C and C', Courtesy [28]*

To undistort the images it is necessary to both remove radial distortions, which exist as a result of the shape of the lens, and tangential distortions, which arise

from the alignment of the lens and the imager. In practise these distortions can be characterised in the first terms of a Taylor series expansion around $r = 0$ [8] which is the radial distortion at the centre of the camera. Usually only two parameters are used for both the radial and the tangential distortion, but since cheap cameras often require an extra parameter for the radial distortion it is best to find five parameters by means of calibration. The method of calibration works by viewing a known object with recognisable features from different angles. In practise a chessboard is used since the dimensions can be easily described.

After the distortion has been removed from the images it is important to make sure that both images are coplanar and aligned so that all rows have the same y-coordinate across all images. This is done by finding the rotation and translation between cameras and combining this essential matrix with the intrinsic parameters of the cameras, in order to get the fundamental matrix which operates in the image pixel coordinates as opposed to the essential matrix which operates on the physical coordinates. The images can then be projected so that their rows are aligned.

The problem here is that it still requires at least two cameras, as opposed to the initial thought to have just one camera as a sensor. Having two cameras means that there is twice as much data to be transferred, twice as much weight to be carried around by the vehicle and twice as much power consumed for the cameras.

Monocular Stereo Vision From biology we can learn that it is in fact not necessary to equip two cameras: birds are capable of perceiving the environment in 3D, but their eyes are positioned so that binocular depth vision is impossible [25]. There is a plethora of methods to extract 3D information from one camera. Such monocular depth cues include occlusion, texture gradients, size and optical flow [25]. It has already been shown that optical flow based stereo vision, which basically treats two images obtained from one camera with a specific time interval as two cameras for stereo vision, is capable of extracting depth information.

It is possible to use only one camera for stereo vision if we introduce the assumption that the environment is static. When the environment is static it means that the time between consecutive frames is not important and that the two frames behave as if they were taken by two different cameras. The assumption here is somewhat limiting for most applications, since any moving object in any of the frames will violate this. The problem of having only one camera is that it is necessary to find the transformation between the consecutive frames for each new frame. This results in the necessity of finding the fundamental matrix for every two consecutive frames. Apart from the computational complexity, this is difficult, for finding the fundamental matrix requires points in both images to be linked so that the transformation can be estimated. Finding these corresponding points and using them for stereo vision is known as structure from motion [28]. A possibility is to use markers scattered around the environment and use these to find matching points between frames. Another possibility is to use optical flow, which basically tells us where a pixel has gone to in the next frame [25, 20, 5, 29, 15].

Since optical flow plays such a significant role in biology it is also not surprising that a lot of research [17, 11, 22, 10, 16, 14] has been done on implementing optical flow estimation in computer vision. Optical flow has many practical uses in computer vision with the most obvious being motion detection. Optical flow can also be used to track markers, which can be used to estimate the translational quantity of the egomotion [17]. Another important use of optical flow is object segmentation which is possible since moving objects can easily be recognised from a sequence of images in which the camera has not moved. Another possibility of object segmentation happens when the camera is moving but the scene is static, because objects closer to the camera move further between frames than objects further away from the camera. The problem here is that this is only accurate when the egomotion between frames is known which reduces the task to a stereo vision task which has been extensively studied.

Using the optical flow based sensor and motion model it becomes possible for an aerial robot to navigate an indoor space [19, 9]. The aerial robot used for this is the AR.Drone, a quadcopter with a camera mounted on the front which can be seen in figure 1. The AR.Drone also has a camera and two sonar sensors pointing downwards for stability. Furthermore, the AR.Drone has three accelerometers and three gyroscopes as internal sensors [18].

With this navigation method the AR.Drone can perform autonomous indoor navigation to complete a sequence of tasks. The sequence of tasks to be completed are given by the IMAV indoor competition[2] in which an aerial vehicle has to navigate an indoor space by performing tasks such as flying through a window.

[2]http://www.imav2011.org/

3 Theory

In this section the various stages of the monocular stereo vision algorithm are explained. First, the calculation of the optical flow is explained. Then it is explained how the optical flow vectors can be used for stereo vision.

3.1 Optical Flow Calculation

Optical flow is the apparent motion between an observer and the environment. The optical flow resulting from one step forward, in a simulated environment, can be seen in figure 6. It serves to estimate the motion field, but does not necessarily correspond to actual motion, as can be seen in figure 5. There are various ways of calculating optical flow. A short description can be found in appendix A.

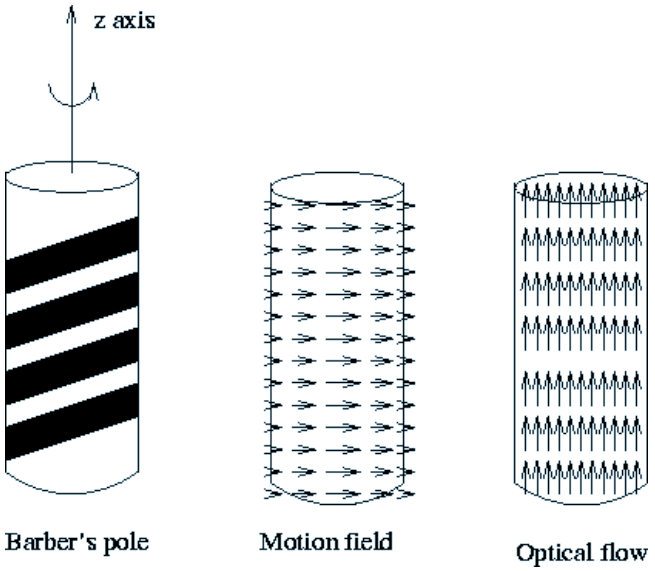


FIGURE 5: *Optical flow does not necessarily corresponds to the motion field, Courtesy [7]*

When calculating optical flow what we actually want is the location of each pixel from one frame in the second. When we try to estimate the optical flow in a continuous fashion, we are actually calculating dense optical flow. The problem here is that it is very difficult to find corresponding pixels, since many pixels can have the same colour and even have surrounding pixels with matching colours. To calculate dense optical flow it is usually needed to introduce the assumption that there is very little movement between images, so that no pixel has moved more than one pixel in distance. This leads to problems when used for robotics [2], since the usually robots move in a speed where any delay in sending or interpretation of images results in shifts over larger distances.

What usually is done is that features are tracked across images [24]. This is called sparse optical flow. For most algorithms, as described in appendix A,

the assumption that movement between frames is very small is still present. This is also true for the Lucas-Kanade algorithm. However, the pyramidal implementation of Lucas-Kanade [7] solves this problem and allows any distance between features.

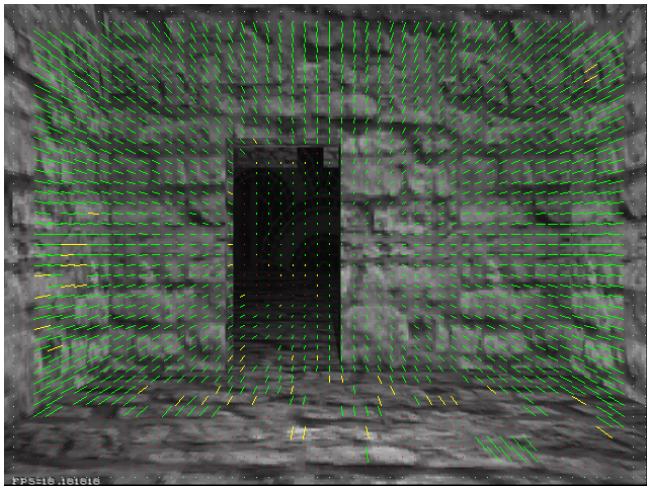


FIGURE 6: *Optical flow in a simulated environment after taking one step forward*

3.1.1 Lucas-Kanade Pyramidal Implementation

The Lucas-Kanade algorithm assumes that distance between frames is small, so that if small windows are taken from the image the optical flow equation 1 holds. Where q_i are the pixels and $I_x(q_i)$, $I_y(q_i)$ and $I_t(q_i)$ are the partial derivatives of the image I for positions x and y and time t for the point q_i.

$$\begin{aligned} I_x(q_1)V_x + I_y(q_1)V - y &= -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V - y &= -I_t(q_2) \\ &\vdots \\ I_x(q_n)V_x + I_y(q_n)V - y &= -I_t(q_n) \end{aligned} \tag{1}$$

These equations can be put in the matrix form $Av = b$ as described in equations 2, 3 and 4.

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_1) & I_y(q_1) \\ \vdots & \vdots \\ I_x(q_1) & I_y(q_1) \end{bmatrix} \tag{2}$$

$$v = \begin{bmatrix} V_x \\ V_y \end{bmatrix} \tag{3}$$

$$b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \tag{4}$$

Because this equation results in less unknowns than equations it is in practise over-determined. Lucas-Kanade uses a compromised solution as given by the least squares fit as seen in equation 5.

$$v = (A^T A)^{-1} A^T b \tag{5}$$

The equation 5 gives the same importance to each pixel within the window. However, it is usually better to give more weight [2] to pixels near the centre of the window by adding a weight matrix W which is a matrix containing all weights for each pixel. The resulting equation can be found in equation 6.

$$v = (A^T W A)^{-1} A^T W b \tag{6}$$

By gradually increasing the size of the window it becomes possible to find optical flow which occurs over larger distances. This is true since Lucas-Kanade finds the least square fit for the pixels.

3.2 Monocular Stereo Vision

Another thing that optical flow gives is corresponding pixels between images. Using these point pairs it becomes possible to calculate the fundamental matrix in a similar fashion as one would with stereo vision. The problem here lies in the fact that with stereo vision the configuration of the cameras is fixed and therefore it is only necessary to calculate the fundamental matrix once. Another problem that arises here is the lack of physical distances between points, since the points are provided by optical flow instead of a known object such as a chessboard[8]. This is an important problem, because if the distances between the found features are unknown it is impossible to determine the scale unless the algorithm is aided by either enriching the environment by adding markers with a set size [17] or by adding another range finding technique which is capable of determining the scale of the movement. Consider two objects which have the same shape and appearance but one is significantly larger than the other one. When the two objects are placed in a non-descriptive space such as an image of the object with a white background, it suddenly becomes impossible to determine the size of the objects [8, 26] and in effect it becomes impossible to determine which is greater in size.

Since this is the case we can only use it for obstacle avoidance because, although the depth maps will be relatively correct, combining multiple maps will be impossible due to their relative size being unknown.

4 Algorithm

In this section the various algorithms that are used for monocular stereo vision are discussed.

4.1 Shi-Tomasi

Lucas-Kanade works by tracking certain features, instead of trying to find the location of each pixel in the other frame. An algorithm that can provide adequate features [8] is the Shi-Tomasi algorithm[24]. They propose the tracking of corners to avoid the aperture problem. Shi-Tomasi define a good feature in the same manner as Harris[12] which relies on the matrix of the second-order derivatives of the image intensities. By calculating autocorrelation of the second derivative over small windows around each point we get a good description of the window. Shi and Tomasi found that a good corner could be easily described as long as the smaller of the two eigenvalues was greater than a minimum threshold.

Since we have low resolution it is necessary to calculate the sub-pixel corner locations [8, 26], because the location of the intensity peak is almost never centred on a pixel. We can fit a parabola on the intensity of the window and find the peak of this parabola to determine the sub-pixel location of the feature.

4.2 RANSAC

The fundamental matrix contains the relation between two frames in that it can give the location of each pixel in one frame to the location of that pixel in the other frame. If the physical relation between the frames is known, the fundamental matrix can be used to obtain the essential matrix which provides the same information only in physical coordinates instead of pixel-coordinates. In order to estimate the fundamental matrix RANSAC is used. RANSAC works by using a small amount of points and fitting the model on these points. After this is done the model is evaluated by calculating whether the remaining points are inliers or outliers of this model. If the model has a sufficient number of inliers the model is re-estimated based on all inliers. This model is then re-evaluated and the error of the model is compared to previous iterations. If the error is smaller the new model is kept. This method has as an advantage that it works well with outliers in the data which is necessary due to the high amount of noise in the Lucas-Kanade algorithm.

4.3 Hartley's algorithm

Once the fundamental matrix is calculated we can rectify the images without calibration by using Hartley's algorithm. Hartley's algorithm attempts to find homographies [26] that map the epipoles to infinity while minimizing the computed disparities. This is done by matching points between the two image pairs which implicitly contains the camera intrinsics. The main problem that

arises is that Hartley's algorithm does not calculate scale [8]. This results in that the 3D reconstruction can only be determined up to a projective transformation. This also means that different projections of an object can appear the same to us when only looking at the configuration of the feature points.

Hartley's algorithm by first calculating the epipoles using the relations $Fe_1 = 0$ for the left epipole and $(e_r)^T F = 0$ for the right epipole [26]. The first homography H_r will map the right epipole to the 2D homogeneous point at infinity. In this case the homography has seven constraints since we can not compute scale. We have four degrees of freedom, for we need only three constraints to map to infinity. These four degrees have to be carefully selected since most choices would result in highly distorted images.

If we calculate the translation T that will take a selected point of interest to the origin of the right image and the rotation R that will take the epipole to $(e_r)^T = (f, 0, 1)$ we can calculate the homography as seen in equation 7.

$$H_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/k & 0 & 1 \end{bmatrix} RT \tag{7}$$

We then need the homography that will do the same for the left epipole to align the rows. This is easily done by aligning the rows as such that the total distance between all matching points is minimised between the images.

5 Experiments & Results

To test out the algorithm, images were taken by the AR.Drone at a resolution of 320 by 240 pixels. Optical flow was calculated over these images and these vectors were used to calculate the fundamental matrix using RANSAC, which in turn served as the input for the Hartley algorithm. A typical example of the algorithms output can be seen in figure 7. In the resulting disparity map, as seen in the upper right image, the posts of the door have been found as well as a small portion of the floor, which is darker and thus further away. The disparity map has a low quality with many pixels missing. This is a result of the sparseness of optical flow vectors, as seen in the bottom left of the image, which have been found due to the wall being a single colour and therefore lacking features found by the Shi-Tomasi algorithm [24] which are used for the Lucas-Kanada algorithm.

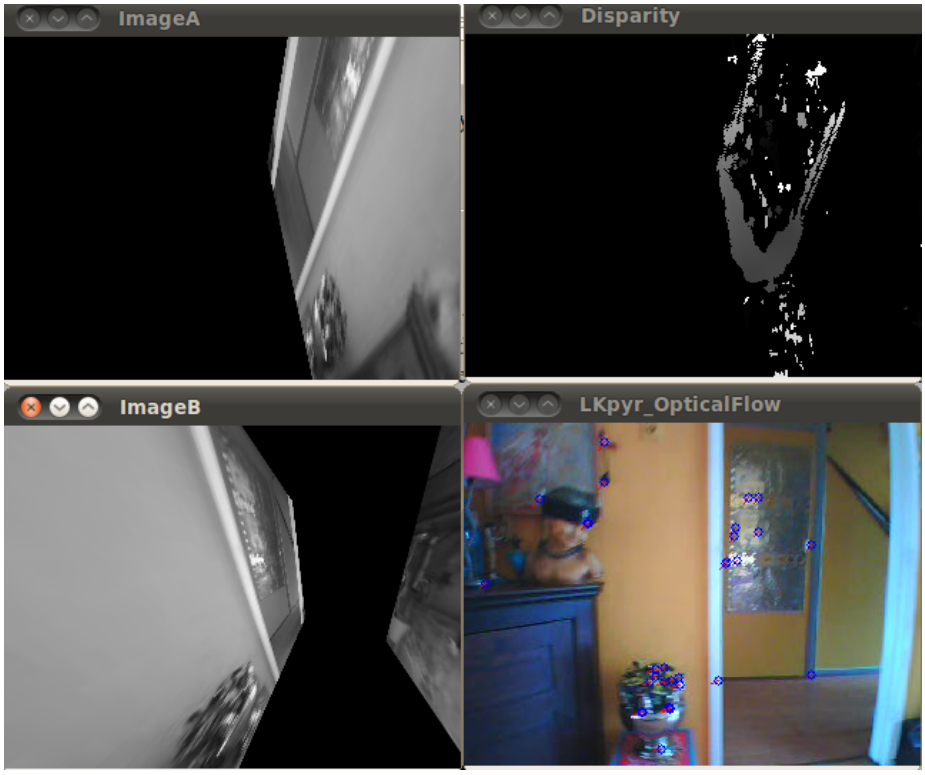


FIGURE 7: *Disparity map generated between two consecutive frames*

5.1 Added texture

By adding texture to the wall, which simulates the textures as used in the IMAV 2011 competition as seen in figure 12, the Shi-Tomasi algorithm [24] finds more features to track resulting in a higher number of optical flow vectors as seen in figure 8. The disparity map now shows more detail and the post of the door

is clearly lighter with the hallway being substantially darker. Another thing to note is the gradient from light to dark to the right of the door post in the disparity map which shows the wall.

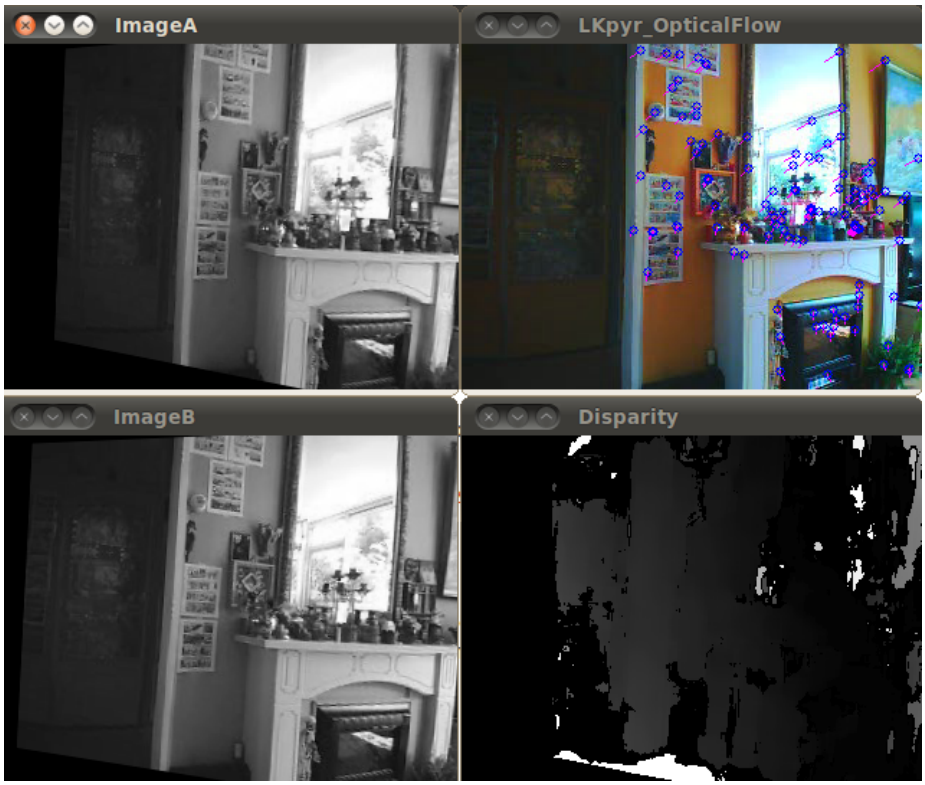


FIGURE 8: *Disparity map generated between two consecutive frames using additional textures*

Although there are now more features to track, the algorithm still has difficulty in finding features due to the low resolution of the camera on the AR.Drone and an automatic white-balancing method in the driver of the camera.

5.2 Higher resolution

By taking images with a higher resolution camera, the amount of features that can be tracked increases. This results in a more robust estimation of the fundamental matrix and increases the quality of the disparity map, as can be seen in figure 9. Although many pixels are still black in the disparity map, the pixels that have been found are relatively free of noise as opposed to the lower resolution images. The door can be clearly distinguished as well as the fireplace.

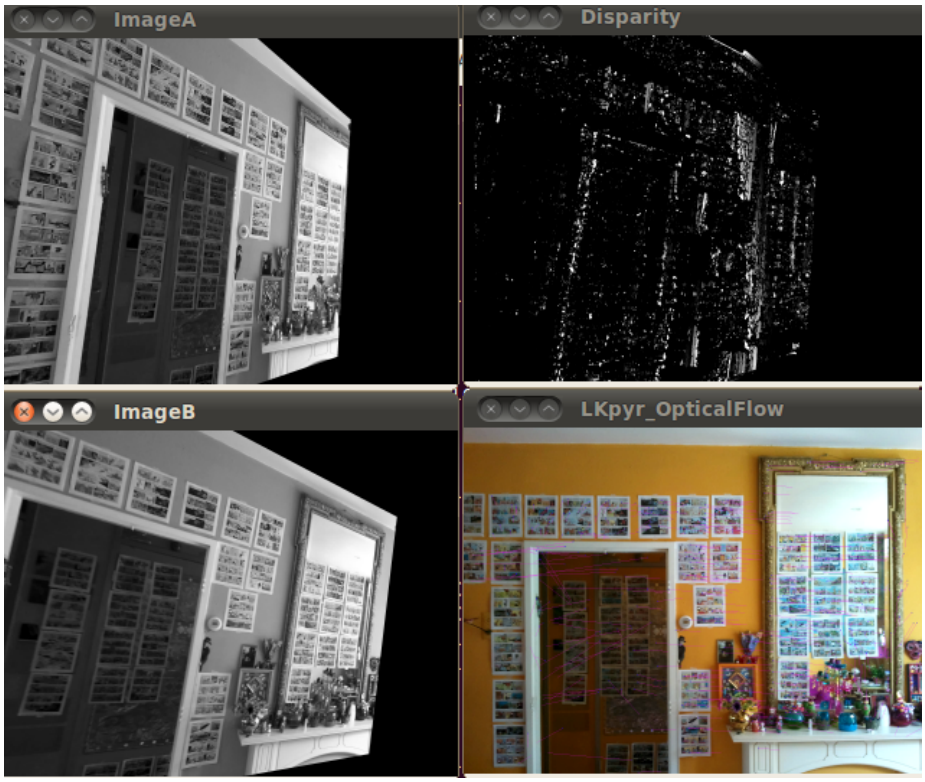


FIGURE 9: *Disparity map generated between two consecutive frames using additional textures and a camera with a higher resolution*

5.3 Limitations

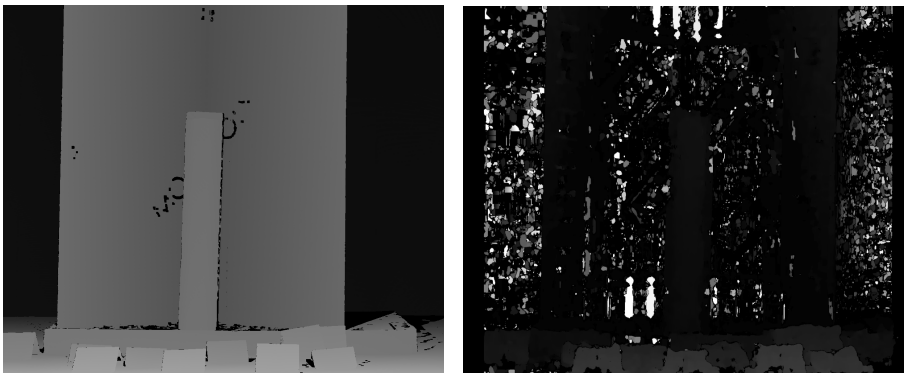
Although the algorithm is capable of creating disparity maps it is not as robust as standard stereo vision. For stereo vision only 18 point pairs are needed to solve the calibration problem [8] and find the intrinsic and extrinsic parameters. However, it is wise to use more point pairs to get a more robust estimation. In practice 10 views of a 7 by 8 chessboard are advised resulting in 560 point pairs. This is quite a lot more than Shi-Tomasi and Lucas-Kanade produce, as can be seen in table 1. Although even with low resolution and a lack of textures in the scene there is still enough points for some quality. However, because of speed constraints in real time navigation tasks, the algorithms used for the monocular stereo vision need to be set in a way that greatly affects the robustness of these algorithms. In particular the calculation of the fundamental matrix by means of RANSAC, which has to terminate after a smaller amount of iterations. Therefore it becomes difficult to use this algorithm in tasks where there is a lack of recognizable textures.

Another major limitation of the algorithm is that it can not find the scale of the depth map and therefore does not work as a range finder but rather a path finder. This limitation may be overcome by using methods that do give physical distance such as marker detection or time-to-contact methods.

TABLE 1: *The amount of point pairs for stereo vision*

Technique	Points
Stereo Vision, minimum necessary	2 views of 3x3 chessboard
Stereo Vision, advice for high quality	10 views of 7x8 chessboard
Optical Flow - Low texture count	150 corner count
Optical Flow - High texture count	260 corner count
Optical Flow - High resolution	380 corner count

These limitations affect the quality of the disparity map. Although a 380 corner count would be enough to make a robust estimate of the fundamental matrix, the actual corner count that is used is in fact lower, since RANSAC eliminates outliers meaning the actual number of corners that are used is lower. The result of this can be seen when comparing figure 10(a) with figure 10(b). The first is the ground truth of a standard stereo vision experiment based on the images seen in figure 11 whilst the latter is the disparity map as given by monocular stereo vision.



(a) Ground truth of a standard stereo vision experiment, Courtesy [13]

(b) Disparity map as given by the monocular stereo vision algorithm

FIGURE 10: *Comparison of disparity map (b) with ground truth (a)*

Since the used images are taken at a high resolution of 1330 by 1110 pixels and well lit, the Shi-Tomasi algorithm was capable of finding a 1000 corners. A 1000 corners was the maximum allowed corner count. This is mainly due to the images having many distinct features as well as the features having the same intensity in both images. 765 of these corners were found in the second image, resulting in 765 optical flow vectors. When calculating the fundamental matrix, RANSAC only used 481 vectors. The 481 vectors are close to the 560 point pairs needed for robust estimation of the fundamental matrix, but it is clear that RANSAC removes quite a few of the point pairs, namely 50% on average. The total count of good vectors is in fact higher since the images were taken using good lighting making both images identical in colour and intensity of features. This explains why the images taken by the AR.Drone result in a lower quality of disparity maps. The AR.Drone moves around the environment resulting in the images to be further apart and having more features only present in one of the two images. The camera of the AR.Drone also uses white-balancing

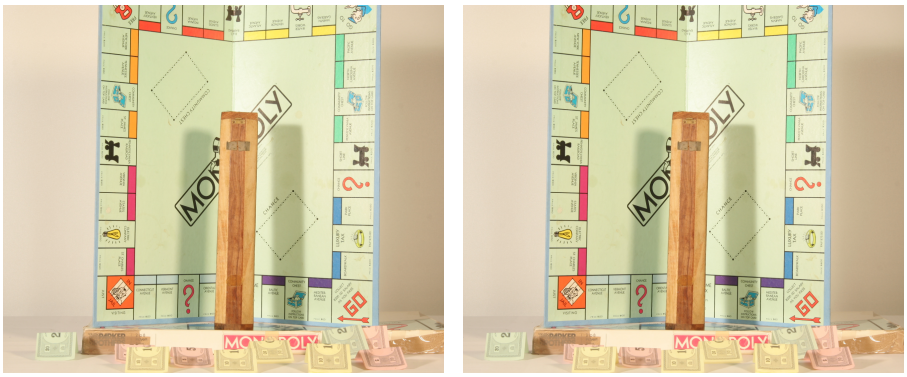


FIGURE 11: *Images used to estimate the disparity map as seen in figure (b)*

resulting in the intensity of the features being different in both images.

When processing the high resolution images, RANSAC removes on average 33% of the optical flow vectors from the calculation. When calculating the fundamental matrix from optical flow vectors in images taken with the low resolution camera of the AR.Drone, RANSAC removes 25% on average. This means that the corner counts as seen in table 1 are higher than the amount of point pairs actually used in the calculation of the fundamental matrix. This explains why the disparity maps created by the AR.Drone are of a lower quality than can be expected from the amount of optical flow vectors found between frames.

6 Conclusion

In this paper a method of using only one camera to obtain depth maps is proposed. Optical flow results in enough point pairs to easily calculate the fundamental matrix although the noise in optical flow lowers the quality of the fundamental matrix. Part of this noise is removed by using RANSAC to calculate the fundamental matrix. Because RANSAC removes the outliers the corner counts are higher than the actual point pairs incorporated for the computation of the fundamental matrix. The low resolution of the camera on the AR.Drone results in the algorithm being unstable because of the lower amount of features that can be tracked by Lucas-Kanade.

Although scale is missing in the fundamental matrix and in effect the disparity maps are only correct up to a coefficient, they are still useful in determining structure in the scene. The disparity map could be used as the basis for a balancing strategy which extracts directional cues from the scene. This would allow the AR.Drone to navigate through doorways and avoid obstacles.

Optical flow is a good basis for autonomous navigation since it is capable of helping navigation of various levels including obstacle avoidance, egomotion estimation as well as providing cues that help the agent to maintain more stability.

FIGURE 12: *Texture of the walls used at the IMAV 2011 competition*

7 Future Research

For the AR.Drone to compete in the IMAV 2011 competition it is necessary to use the current algorithm and enhance it with a balancing strategy to make sure the AR.Drone will not crash. The camera of the AR.Drone also allows another algorithm to take care of the situational awareness necessary for the various stages of the challenge. A method similar to the single image cue method described in section 2 could potentially solve this. The disparity maps provided the algorithm described in this thesis give clues about passages, which can be translated into a directional cue. For instance, the first wall results in the depth map consisting out of one plane whilst once the AR.Drone reaches the top of the wall the disparity map changes into one which provides far more structure. The doorway of the building could then be recognised as being a rectangular darker patch within a larger plane.

Because the fundamental matrix has to be calculated for each pair of consecutive frames this algorithm is fairly slow for real world tasks. It is necessary to tweak the various parameters in the underlying algorithms to find the optimal balance between speed and robustness. Such parameters include the number of iterations RANSAC can take to test models as well as the threshold of certainty necessary to couple features in the two frames.

Time to Contact Since scale is missing from the depth map it is necessary to add an extra range finding algorithm in order to have actual monocular depth estimation. A possible method to derive depth information from a single camera is by calculating the time to contact using optical flow. When the movement between frames has been purely translational and the scene is static all optical flow vectors will originate from a single point [6, 21], the focus of expansion. The time to contact can be calculated as seen in equation 8 where Δ_i is the distance to the focus of expansion and V^t is the magnitude of the optical flow vector [1].

$$TTC = \frac{\Delta_i}{|\vec{V^t}|} \tag{8}$$

Combining this with the velocity vector of the agent it becomes possible to estimate a depth map. This depth map is fairly limited in use, since the various points which get a depth are either very few when a sparse optical flow algorithm is used or very noisy when a dense optical flow algorithm is used.

This method has a very limiting restriction in that only pure translation is allowed. Whilst it is already hard for most ground-vehicles to comply to this restriction, it is downright impossible for a helicopter type vehicle, since this type of vehicle uses rotation to move. It is possible to determine the rotational quantity of motion using optical flow which would allow the AR.Drone to only use TTC when the AR.Drone detects it is moving straight forward. However, since the AR.Drone needs to pivot around its axes it is rare that the AR.Drone only moves in a translation.

The focus of expansion can be calculated using RANSAC by using intersection of two lines as the model and then calculating distance to all other flow vectors.

The result of two typical images can be seen in figure 13 which were taken with the high resolution camera. The images are taken by moving the camera slightly forward. Already it is apparent that even little amounts of rotation already result in difficulties when finding the focus of expansion.

FIGURE 13: *Focus of expansion calculated by RANSAC*

Calculating the focus of expansion is only possible from a forward or backward transformation. When the translation is sideways the focus of expansion lies at infinity and it therefore becomes impossible to calculate the distance to the features.

Although this method is not applicable to a MAV it does provide accurate distance measurements for robots which can satisfy the translational constraint more easily. This effect can be seen in figure 14 where the size of the circle signifies the distance to the object with larger circles being closer to the camera. When using a simple balancing strategy in which the image is divided into four segments, the upper right quadrant would have the least amount of obstacles and thus is a potential candidate for a directional cue.

Optical Flow Another possibility is using just optical flow to estimate distances. This can be done using the time to contact method but this method requires the AR.Drone to move in a pure translation. Because the camera is low resolution, objects that are closer are in theory more defined, since more pixels are used to describe these objects. When the image is divided into multiple windows the count of optical flow vectors could provide some clues about the distance to objects. This is another of the monocular depth cues, namely

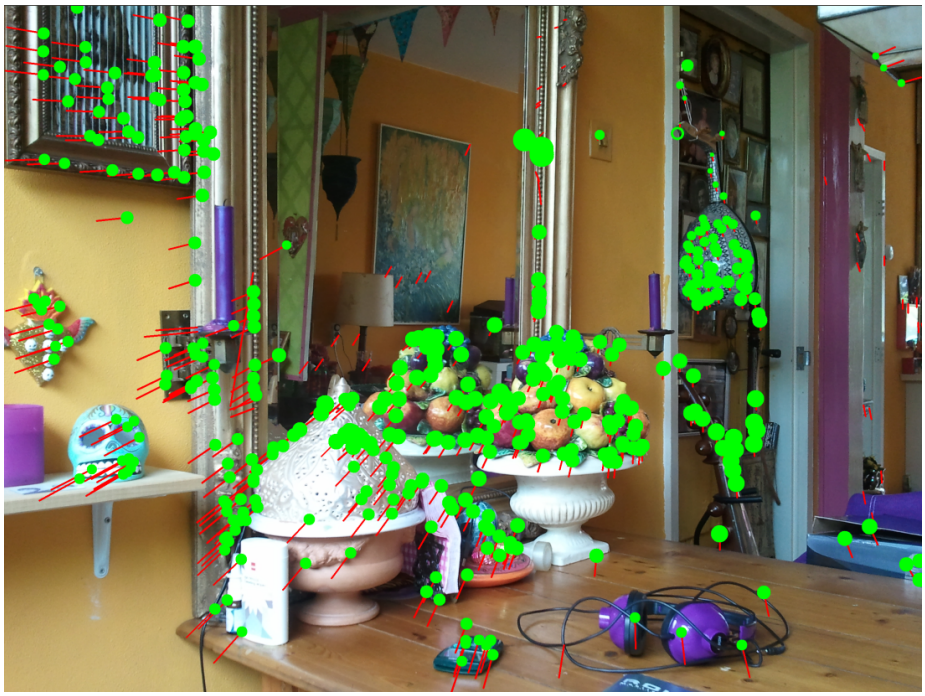


FIGURE 14: *Time to contact as calculated from optical flow and the focus of expansion*

texture gradient [10]. Furthermore, the optical flow vectors will be pointing to the side of the image when objects get closer to the camera. If the image is divided into columns and the direction of all the optical flow vectors as well as their magnitude is taken into consideration, it could be possible to determine which objects are getting closer. These cues are very dependent of a couple of assumptions, including the necessity of the image to have a uniform distribution of texture. The latter needs the rotation of the camera to be negated, since rotation causes all optical flow vectors to be altered in magnitude. For instance, a rotation to the left causes optical flow vectors that would point to the right to be decreased in magnitude whilst optical flow vectors that point to the left are increased in magnitude. Since the rotational quantity of the transformation between frames can be determined from optical flow it is possible to weight the magnitudes of the optical flow vectors so that the effect of the rotation is negated.

A combination of these techniques would allow the AR.Drone to avoid obstacles in most real world environments.

References

[1] N. Ancona and T. Poggio. Optical flow from 1d correlation: Application to a simple time-to-crash detector. In *International Journal of Computer Vision*, pages 673–682, 1993.

[2] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994.

[3] C. Bills, J. Chen, and A. Saxena. Autonomous mav flight in indoor environments using single image perspective cues. *International Conference on Robotics and Automation (ICRA)*, 2011.

[4] F. Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1), 2004.

[5] G. Bleser and G. Hendeby. Using optical flow as lightweight slam alternative. *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, 0:175–176, 2009. ISBN 978-1-4244-5390-0.

[6] C. Born. Determining the focus of expansion by means of flowfield projections. In *In Proc. Deutsche Arbeitsgemeinschaft fur Mustererkennung DAGM'94*, pages 711–719, 1994.

[7] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[8] G. Bradski and A. Kaehler. *Learning OpenCV, Computer Vision with the OpenCV Library*. O'Reilly books, September 2008.

[9] M. J. Brooks, W. Chojnacki, and L. Baumela. Determining the egomotion of an uncalibrated camera from instantaneous optical flow. *Journal of the Optical Society of America A*, 1997.

[10] A. Dev. *Visual Navigation on Optical Flow*. PhD thesis, University of Amsterdam, September 1998.

[11] D. J. Fleet and Y. Weiss. *Mathematical Models in Computer Vision: The Handbook (Optical Flow Estimation)*, chapter 15, pages 239–258. Springer, 2005.

[12] C. Harris and M. Stephens. A combined corner and edge detector. *International Journal of Computer Vision*, 35, 1998.

[13] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2007.

[14] S. J. Huston and H. G. Krapp. Visuomotor transformation in the fly gaze stabilization system. *PLoS Biol*, 2008.

[15] K. Kanatani. Self-calibration from optical flow and its reliability evaluation. In *IAPR Workshop on Machine Vision Applications (MVA2000)*, pages 443–446, 2000.

[16] D. Kane, P. Bex, and S. Dakin. Quantifying "the aperture problem" for judgments of motion direction in natural scenes. *Journal of Vision*, 11(3): 1–20, 2011.

[17] B. Kelly. Structure from stereo vision using optical flow. Master's thesis, University of Canterbury, November 2006.

[18] T. Krajník, V. Vonásek, D. Fišer, and J. Faigl. AR-Drone as a Platform for Robotic Research and Education. In *Research and Education in Robotics: EUROBOT 2011*. Springer, Heidelberg, 2011.

[19] T. Low and G. Wyeth. Learning to avoid indoor obstacles from optical flow, December 2007.

[20] B. D. Lucas and T. Kanade. Optical Navigation by the Method of Differences. In *International Joint Conference on Artificial Intelligence*, pages 981–984.

[21] S. Negahdaripour and B. Horn. A direct method for locating the focus of expansion. *CVGIP*, 46(3):303–326, June 1989.

[22] J. A. Saunders and D. C. Niehorster. A bayesian model for estimating observer translation and rotation from optic flow and extra-retinal input. *Journal of Vision*, 10(10):1–22, 2010.

[23] D. Scharstein and R. Szelisk. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1), April-June 2002.

[24] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[25] S. F. te Pas. *Perception of Structure in Optical Flow Fields*. PhD thesis, University of Utrecht, September 1996.

[26] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, 1998.

[27] I. Ulrich and I. R. Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *AAAI/IAAI'00*, pages 866–871, 2000.

[28] M. Varga. *Practical Image Processing and Computer Vision*, chapter 13. John Wiley & Sons, 2009.

[29] M. Zucchelli, J. Santos Victor, and H. Christensen. Constrained structure and motion estimation from optical flow. pages I: 339–342, 2002.

A Optical Flow Algorithms

Differential Techniques are techniques to compute velocity from spatiotemporal derivatives of image intensity. They work because they have the assumption that intensity is conserved.

- **Lucas and Kanade:** Introduces assumption that flow is constant in the local neighbourhood of each pixel.
- **Horn and Schunck:** Introduction of a global smoothness constraint to solve the aperture problem.
- **Nagel:** Improves the Horn and Schunck method by assigning an orientation to the smoothness.

Region-Based Matching is a set of techniques which define velocity as the shift that has the best ft between image regions at different times.

- **Anandan:** Based on a Laplacian pyramid and a coarse-to-fine SSD-based matching strategy.
- **Singh:** A two-stage matching method which also applies a Laplacian pyramid to center the SSD surface at the true displacement.

Energy-Based Methods are based on the output energy of velocity-tuned filters. These techniques are also known as frequency-based methods.

- **Heeger:** A least-squares fit of spatiotemporal energy to a plane in frequency space.

Phase-Based Techniques are techniques which define velocity in terms of the phase behaviour of band-pass filter outputs.

- **Waxman, Wu and Bergholm:** A method in which edges are tracked in real-time..
- **Fleet and Jepson:** In this method component velocity is defined in terms of the instantaneous motion normal to level phase contours.