Terrain Classification from UAV Flights using Monocular Vision

Igor S. G. Campos, Erickson R. Nascimento and Luiz Chaimowicz
Department of Computer Science
Federal University of Minas Gerais
Belo Horizonte, Minas Gerais 31270-901
Email: {igor.gama, erickson, chaimo}@dcc.ufmg.br

***Abstract*—With the popularization of small Unmanned Aerial Vehicles (UAVs) and their usage diversification among various fields, such as aerial mapping applications, it is important to develop better terrain following techniques that rely solely on the vehicle's sensing capabilities. The objective of this paper is to evaluate whether is possible to gather information about terrain inclination and elevation from monocular video captured from such aircrafts. Our approach is biologically inspired by trying to reproduce some insects behaviour with the use of optical flow to infer about the terrain. We built an UAV specifically for this research which uses a gimbal stabilized down-facing camera and flew it at a fixed Above Sea Level (ASL) altitude. After performing preliminary analysis on sparse optical flow data and validating the concept, we moved towards a dense optical flow algorithm and created different descriptors to feed multiple decision trees in order to infer about terrain characteristics. We achieved accuracies of 77.34%, 86.75% and 91.85% depending on the evaluated characteristic, showing that our approach is valid.**

I. Introduction

With the use of small Unmanned Aerial Vehicles (UAVs) becoming increasingly popular quite rapidly, many new applications begin to appear, however they are still limited by the vehicle's degree of autonomy. One example is aerial mapping, such applications happen in uneven terrain, such as in [1], where it is necessary to perform terrain following flights, however there are still very few low cost sensors capable of estimating the height of the aircraft and their range is still limited.

In aerial mapping applications, UAV's typically use previous information about the terrain elevation, which allow them to calculate the proper Above Sea Level (ASL) altitude at each coordinates of the flight to keep themselves at the desired Above Ground Level (AGL) altitude. However, this information, such as data from the Shuttle Radar Topography Mission (SRTM), is usually in low resolution, outdated or present incorrect data. This inspired us to evaluate the possibility of using sensors commonly found on small UAVs to gather terrain inclination and elevation information.

Since the focus of this work is on a monocular vision approach to infer the terrain elevation and inclination, we decided to use optical flow. Optical flow is an approximation to the motion flow of objects in a sequence of images. While motion flow refers to the motion of objects, optical flow is the apparent motion of such objects, as a consequence, it does not work well with objects without texture. The optical flow consists of the displacement vectors of the image points between two subsequent images.

In biology, researchers found evidence which supports that insects use optical flow to control their flights, including the desired characteristic of terrain following [2]–[5]. Optical flow allows the insects to perceive relative distance to the objects and control their AGL altitude with relation to their ground speed.

In this work we propose a classification algorithm to infer terrain characteristics based on information provided by monocular vision from a RGB camera set up on a custom built quadrotor UAV.

We captured video sequences of flight over different terrains and ran sparse optical flow algorithms to translate the video in meaningful data. We analysed histograms of the magnitude and direction of the optical flow vectors and found that scenes with different terrain characteristics were quite discernible. This led us to our approach, consisting of feeding such data into a decision tree to determine if there was a slope or not, and if such slope was positive or negative, as well as its direction.

Remaining sections are organized in the following manner. Section II discusses some related works. Section III shows our initial analysis and classification process. Section IV describes our custom built UAV. Section V details our experiments. Finally, Section VI shows our conclusions and further steps of our work.

II. Related Work

Several researchers studied insects and suggested they use optical flow to control their flight. Horridge [2] pointed out that it is a much cheaper way to compute distances and reliable enough for insect to perform their various tasks. He also emphasized the relevance of such discovery for robot vision, indicating that the processing power of a computer back then already surpassed insects' computing capabilities. Lehrer et. al. [3] reinforced such findings by observing trained bees gathering food in various controlled environments. They varied the scale of objects and found that bees completed the task irrelevant to such changes, leading to a conclusion that they navigate based on scale, disregarding the real size of objects. Srinivasan et. al. [4] found that bees can estimate the distance of a given object by analysing its motion across the retina.

In the robotics field, Franceschini et. al. [5] built a synthetic compound eye based on the fly's visual system and conducted

terrain-following flights of a flying robot in a controlled environment. Their controller did not account for a desired height, instead it was automatically regulated according to the horizontal speed of the robot and a target optical flow displacement. Another important finding is that in featureless environments the absence of optical flow makes the robot fly lower and lower until it crashes into the ground.

Chao et. al. [6] conducted a survey of optical flow techniques for UAV navigation applications. They identified works for the most diverse applications that use optical flow as the main technique to achieve the desired goals. The examples are: distance estimation, altitude hold, hovering, velocity estimation, obstacle avoidance, vertical landing, height estimation and terrain following.

Zufferey et. al. [7] performed a collision avoidance experiment of a fixed wing UAV following a predetermined path. The UAV used GPS to follow the path and several optical flow sensors commonly found in computer mice to avoid trees in the path. This demonstrates the potential of optical flow as a mean of sensing for UAV control.

Garrat and Chahl [8] conducted field experiments of an UAV terrain following flight using optical flow. They used a Yamaha RMAX helicopter carrying a camera, a laser range finder to be used as ground truth and a differential GPS unit to account for the camera translation. In their experiments, the flight was conducted over flat terrain and the AGL altitude was only 2 meters. This prevented the analysis of the algorithm behaviour at higher altitudes and at varied relief, such as a mountain.

Differently from these approaches, our methodology is focused on inferring certain characteristics about the terrain, such as elevation and inclination. Which should enable us to anticipate actions of the UAV with relation to these characteristics in future works.

III. Methodology

Our methodology is composed of two main modules: the optical flow computation and the classification. Each of them will be discussed in detail in the following subsections.

A. Optical Flow

Optical flow is the measurement of displacement between image points from two consecutive images. The technique assumes that the brightness is constant between frames captured in short intervals:

$$I(x, y, t) = I(x + u, y + v, t + 1). \tag{1}$$

Which expanding through its Taylor series transforms into:

$$-\frac{\partial I}{\partial t} = \nabla I \cdot \begin{bmatrix} u \\ v \end{bmatrix}. \tag{2}$$

This gives us one equation (2), but that leaves us with an underconstrained system, since the optical flow consists of vectors defined by two independent variables.

There are several approaches to create an additional constraint needed to solve the system. The first one was developed by Horn and Schunck [9] : It imposes global smoothness, assuming that vectors from a given region of the image point have the same direction and magnitude, which is mostly true, but fails at edges and corners. The second approach was developed by Lucas and Kanade [10]: It consists of finding the displacement vector that best fits a windows of pixels, imposing local smoothness, and is well suited for tracking features, such as corners.

To evaluate if optical flow provided sufficient information about the relief we first analysed data from sparse optical flow [10] by looking for discernible histogram patterns of the magnitude and angle of the optical flow vectors. The magnitude histogram consists of 24 bins of fixed size while the angle histogram consists of 180 bins of fixed size, as seen in Figure 1.

Based on our observations the terrain types were clearly discernible when the following assumptions were true:

- the scene is mostly static;
- there is small to no camera rotation.

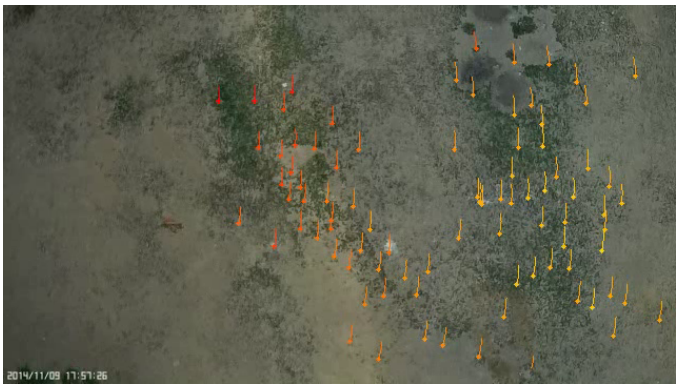
After proving the concept we moved to a dense optical flow approach to evenly distribute the vectors among the image, since we had some issues with areas of low texture. We chose the Farnebäck algorithm (FB) [11], which is based on an approximation of each neighborhood of both frames by quadratic polynomials. The author presents an efficiently and accurate approximation by using the polynomial expansion transform, providing a fast and robust algorithm to estimate the displacement field between two frames. In addition to the small error of optical flow computed by FB, we can efficiently compute dense flows using the CUDA implementation available at the OpenCV library. Despite the aforementioned advantages of FB algorithm, it also fails to find optical flow in areas with low texture. Thus, to minimize the impact of such areas, we only used displacement vectors with a magnitude over a minimum threshold, which was determined empirically.

B. Classification and Feature Vector

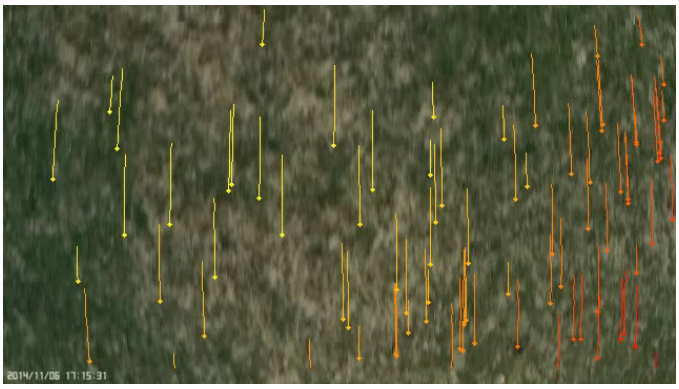
We identified three characteristics we wanted to classify about the terrain. The first one is whether the terrain is flat, inclined or something else (mixed, for instance). The second one is whether the terrain is elevating, dropping or neither with relation to the UAV flying direction. The last one is the direction of the inclination of the terrain, which we discretized into eight possible values representing the direction in intervals of 45 degrees (see Figure 2), there is an additional class representing terrain without inclination.

We chose to use decision trees to classify our data since they are fast to train and classify, allowing us to perform many tests with different feature vector configurations. The decision tree type we used is the ID3 [12], as implemented by OpenCV. Before training the decision trees, we needed to reduce the data from the dense optical flow algorithms. For that, we tried three different descriptors and all possible combinations of them.

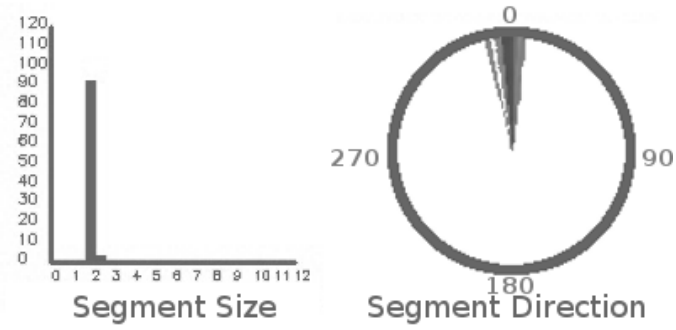
The first descriptor consisted of slicing the image in 8 triangles, each one consisting of half a quadrant. Considering the origin as the camera center pixel, the data would be the average of each bin (Figure 3a). The second descriptor



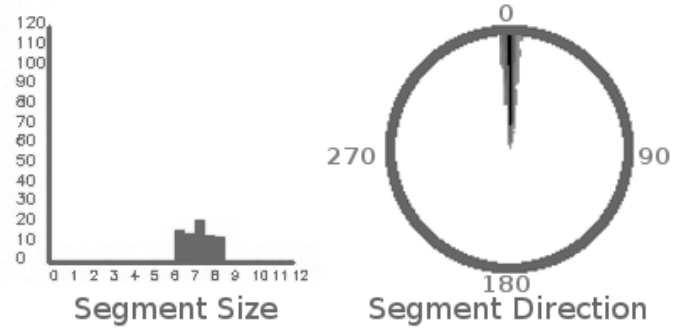
(a) Optical flow of flat terrain image.



(b) Optical flow of inclined terrain image.

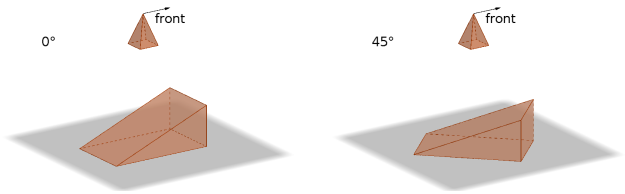


(c) Segment size and direction histograms of flat terrain



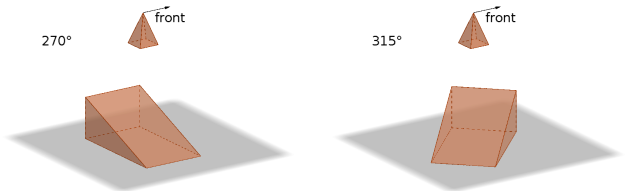
(d) Segment size and direction histograms of inclined terrain

Fig. 1: Sparse optical flow of a flat terrain image (a) and inclined terrain image (b) along with their respective histograms (c) and (d). The colors of the flow vectors vary according to their size, bigger vectors are yellow and smaller vectors are red. The histograms represent the vector magnitude (left) and direction (right), in the later, color intensity represent the distribution among the bins, darker bins representing more vectors laying inside them. In these samples the segment direction is highly concentrated close to 0º, however the segment size histogram show a completely different behaviour, being concentrated between 1,5 and 2 pixels in (c) and spread between 6 and 8,5 pixels in (d). This illustrates the expected difference according to the terrain.



(a) 0º

(b) 45º



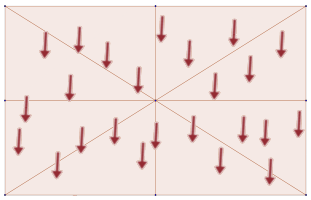
(c) 270º

(d) 315º

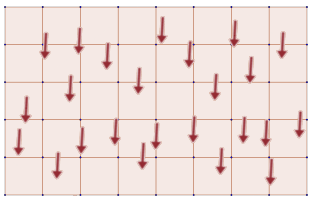
Fig. 2: Images showing some samples of direction of inclination.

consisted of evenly distributed square bins over the image, forming a grid, composed of the average magnitude of each one. Specifically, we used bins of 50 pixels in width and height (Figure 3b). The last descriptor consisted of a history of the 10 last measures of the average magnitude of optical flow over the entire image.

After the preprocessing we fed the data into the decision



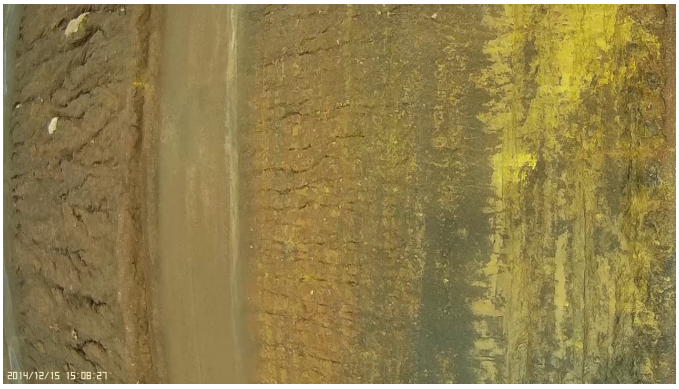
(a) First descriptor.



(b) Second descriptor.

Fig. 3: First (a) and second (b) descriptors. They consist of slicing the image in bins determined by the lines and calculating the average value of each bin.

(a) Flat terrain



(b) Mining site

(c) Buildings

(d) Trees

Fig. 4: Snapshots of videos composing our dataset. (a) Represents an area of flat terrain, (b) an area of inclined terrain, (c) and (d) areas of "other" terrain.

trees as implemented by the OpenCV library and obtained the results presented in Section V-B.

IV. UAV Platform Design

To perform our experiments we built a quadrotor UAV (Figure 5) from *off-the-shelf* parts. It weights less than 2Kg and its approximate size is 572x572x250mm, resulting in a very popular size and weight among other quadrotor UAVs.

Our UAV frame is the DJI F450, a robust and inexpensive one. It features four NTM 2826-1000KV motors attached to 1045 propellers powered by 3S LiPo batteries. The flight controller is the popular ArduPilot Mega 2.6. It features an IMU consisting of accelerometer, gyroscope, magnetometer and barometer together with a Ublox NEO-6M GPS unit and a 433MHz telemetry radio.

The flight controller uses the open-source ArduCopter 3.2.1 firmware, which allows us to log flight data and use the MAVLink communication protocol to monitor and send commands to the UAV. There is a ROS interface driver (roscopter) for such applications, which allows us to override the remote controller transmitter or send more sophisticated instructions like waypoints, take-off and land commands.

Our camera is stabilized by an actuated gimbal (Quanum Q-2D) with a 0.1^{o} accuracy, its control loop runs at 1kHz,

Fig. 5: Custom Built UAV. It is based on the DJI F450 frame and uses an ArduPilot Mega 2.6 as flight controller. In this picture the UAV's camera has been upgraded to a GoPro Hero 3+ Black Edition, but at the time the experiments were performed the vehicle was equipped with a Mobius ActionCam Lens B, which is comparable to the GoPro Hero 3 White Edition. It also uses a brushless motor gimbal to stabilize the image.

the motor driving frequency is 32kHz and it is attached to the quadrotor through four vibration dampening pads. The camera is a Mobius ActionCam Lens B and the UAV also features a 900MHz analogue video transmitter to allow real-time processing outside of the vehicle.

V. Experiments

The goal of this work is to infer about the terrain, thus we processed the data from the camera and flight recorder offline. The computer was a Lenovo y50-70 laptop featuring an Intel core i5 4200H processor, a 2GB NVIDIA GeForce GTX 860M with Maxwell architecture and 8GB 1600MHz primary RAM. We used the OpenCV library compiled with CUDA in the Ubuntu 14.04 64-bit environment to calculate the optical flow more efficiently.

To compute the optical flow we downsized the video in 50% across each dimension and ran the CUDA implementation of the Farnebäck algorithm with a step of 5 frames between comparisons, achieving a 7 FPS processing rate. Since the decision trees classification time is negligible when compared to the optical flow processing we did not account for it.

To perform our tests, we created a dataset from images acquired in our experiments. Further details about the dataset construction are described in the following subsection.

A. Dataset

The experiments we conducted consist of fixed ASL altitude flights over different terrain types, such as hills, flat surfaces, over buildings and over trees (Figure 4). We planned the flight through waypoints and performed an autonomous mission at a configured speed of 3.5 m/s. The missions were executed when there was little wind.

The gimbal stabilized camera features a fixed focus lens and was configured to face down and shoot 720p video at 60 FPS with the narrowest Field Of View (FOV).

Our ground truth is based on previous knowledge of the terrain, which was obtained through previous walks over the flight path and manual selection and annotation of the video scenes over different terrain types. Our dataset is composed of 948 entries as described in Tables I and II.

B. Results

We trained three different decision trees. The first one was meant to classify whether the terrain was flat, inclined or something else, such as over buildings or trees. The second was to classify if the terrain was going up, down or none of those. The last one was to determine the direction of inclination. The trees were trained using 10-fold cross-validation and 80% of the data was used as the training set, the remaining 20% composing the test set. The confusion matrices are shown in Figure 6.

In the first decision tree we achieved an accuracy of 86.75% with the combination of the first and last descriptors, with a higher influence from the first descriptor, which is expected, since it was designed to detect inclination.

In the second decision tree we achieved an accuracy of 91.85% with the combination of the second and third descriptors, where most of the influence comes from the third descriptor. This is a consequence of the elevation being evaluated as a function of time, and as seen in Section II, the optical flow is able to provide height information, thus resulting in elevation information if evaluated in subsequent frames.

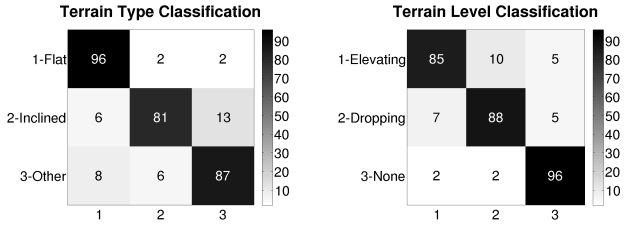
In the last decision tree we achieved an accuracy of 77.34%

TABLE I: Dataset composition for terrain inclination and terrain elevation.

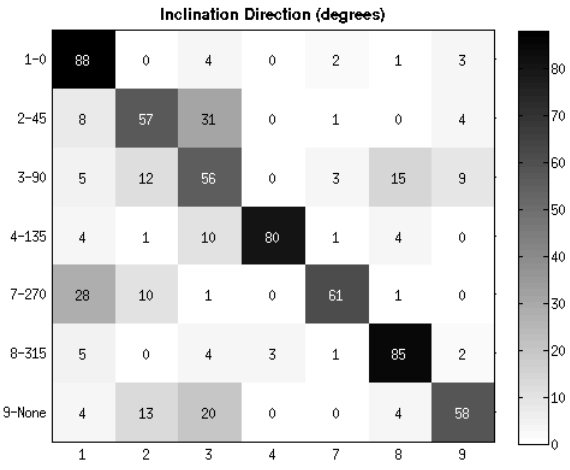
Tree	Class	Amount
1	Flat	264
	Inclined	324
	Other	360
2	Elevating	192
	Dropping	264
	None	492

TABLE II: Dataset composition for terrain inclination direction.

Class	None	0º	45º	90º	135º	270º	315º
Amount	264	36	108	156	72	252	60



(a) Terrain type (b) Terrain level



(c) Terrain inclination direction

Fig. 6: Confusion Matrices. In (c) we omitted two classes (180º and 225º) in which we did not have any data.

with the combination of all descriptors, however, by crossing data from Table II and Figure 6c, we conclude that the low accuracy in some classes is a consequence of few data in them. In this tree all descriptors contribute to classification, with the last one contributing the least, but still providing valuable data for inclination directions closer to 0^o and 180^o.

VI. Conclusion and Future Works

In this work we showed that it is viable to infer about the terrain from optical flow provided by aerial videos and we hope that this brings further improvement to terrain following algorithms. Some benefits could be colision avoidance and taking earlier control decisions based on terrain information instead of waiting for optical flow to reach predetermined thresholds,this way the UAV would behave as if it was looking ahead.

In our dataset our experiments demonstrated promising results, but we have yet to test the robustness of the algorithm with data from other environments, however, since the data analysed is the optical flow, we expect similar results as long as there is enough texture to be tracked on the images.

As future development we are working on an algorithm to estimate the AGL altitude of the UAV and intend to develop better terrain following techniques with the fusion of data provided by this study.

Acknowledgement

The authors gratefully acknowledge CNPq, FAPEAM, CAPES and FAPEMIG for the financial support provided, which enabled the development of this work, and Vale Technological Institute for allowing the recording of videos with the UAV at a deactivated mining site, enriching our dataset.

References

[1] sensefly: Mapping the matterhorn. senseFly. [Online]. Available: https://www.sensefly.com/user-cases/mapping-the-matterhorn.html

[2] G. A. Horridge, "A theory of insect vision: Velocity parallax," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 229, no. 1254, pp. 13–27, 1986.

[3] M. Lehrer, M. V. Srinivasan, S. W. Zhang, and G. A. Horridge, "Motion cues provide the bee's visual world with a third dimension," *Nature*, vol. 332, pp. 356–357, Mar. 1988.

[4] M. Srinivasan, M. Lehrer, S. Zhang, and G. Horridge, "How honeybees measure their distance from objects of unknown size," *Journal of Comparative Physiology A*, vol. 165, no. 5, pp. 605–613, 1989.

[5] N. Franceschini, F. Ruffier, J. Serres, and S. Viollet, *Aerial Vehicles*. InTech, 2009, ch. Optic flow based visual guidance: from flying insects to miniature aerial vehicles.

[6] H. Chao, Y. Gu, and M. Napolitano, "A survey of optical flow techniques for uav navigation applications," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, May 2013, pp. 710–716.

[7] J.-C. Zufferey, A. Beyeler, and D. Floreano, "Autonomous flight at low altitude with vision-based collision avoidance and gps-based path following," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 3329–3334.

[8] M. A. Garratt and J. S. Chahl, "Vision-based terrain following for an unmanned rotorcraft," *Journal of Field Robotics*, vol. 25, no. 4-5, pp. 284–301, 2008.

[9] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *ARTIFICAL INTELLIGENCE*, vol. 17, pp. 185–203, 1981.

[10] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[11] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, ser. Lecture Notes in Computer Science, J. Bigun and T. Gustavsson, Eds. Springer Berlin Heidelberg, 2003, vol. 2749, pp. 363–370.

[12] J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.