UNSYS digital

International Journal of Robotics and Mechatronics

Vision Based Tracking of Moving Target in an Autonomous Ground Vehicle Framework

S.M. Vaitheeswaran[†], Sangram Behara[‡], M.K. Bharath[†], and M. Gokul[§]

[†]CSIR- NAL, ALD, Bangalore, India

[‡]NITK Surathkal, Mangalore, India

[§]Amrita VishwaVidyapeetham, Coimbatore, India

***Abstract*—**This paper proposes a vision-based algorithm to autonomously track a moving target with an autonomous ground vehicle. The proposed approach, which is to estimate the target's position and orientation, is built on a robust colour based tracker using the Continuously Adaptive Mean Shift (CAMShift) algorithm. The object tracker can handle occlusions, lighting and environment effects in a single framework when combined with Multiple Kalman Filters. The information is then used from the visual tracker to control the position and yaw angle of the UAV in order to track the object and keep it in the field of view. The system is practically implemented and tested using the Arduino platform and a cheap low cost web based camera.

***Keywords*—**CAMShift, mean-shift, probability distribution, robust control, pulse width modulation, gimbal control.

I. Introduction

AUTONOMOUS vehicles are being researched and developed in the world with a larger aim of replacing the human or human operated vehicles, to provide situational awareness, increased safety and assistance features, optimal routing, vehicle to vehicle communication, communication with ground infrastructures among others. In order for these vehicles to accomplish the mentioned objectives, a multitude of problems in perception, navigation, and control need to be addressed.

An autonomous vehicle uses a multitude of sensors such as laser, radar, Light Detection and Ranging [LIDAR], computer vision and Radio Frequency Identification [RFID], Global Positioning System coordinates fusion techniques to perceive the vehicle's local environment. The image analysis including the changing dynamics from these vehicles, in general, fall into the following categories [1]: background subtraction methods, sparse features tracking methods, background modeling techniques and ego motion models.

These techniques have limitations when the sensor such as a gimballed camera is mounted on the moving vehicle. The environmental conditions like lighting or colors are permanently changing, and there are a lot of static as well as dynamic objects in the scene to be taken into account. Typical approaches to handle the limitations have included monocular [2], stereo [3], and lasers [4], to determine motion of the vehicle and construct maps of the terrain using optical flow. vectors, color features and stereo pair disparities are commonly used as feature vectors. An example approach is to apply frame differencing to detect pixel motions in homogeneous regions and accumulate them [5, 6]. A second method has been to know apriori the motion of the camera and map optical flow vectors onto a sphere [7]. A third approach has been to obtain occupancy grids and determine direction and the speed of the dynamic objects for safe driving environments [8].

For objects that have simple and constant appearance, the CAMShift algorithm has been proposed for kernel-based visual object tracking. It has been used as a basic component with a number of advanced trackers for a number of applications, but the method is prone to tracking failures for rapidly changing scene conditions, occlusions, objects with same color etc. To overcome this, a combined Kalman filtering with CAMShift tracking approach has been proposed [9] to enable track recovery after full occlusions. A background-weighted histogram is used to distinguish the target from the background and from other targets. However, a real time implementation which considers all the issues of rapidly changing background, lighting, color, occlusions integrated to practical control system remains to be addressed.

In this paper a single real time framework for moving object detection and tracking that also accounts for occlusion, lighting, background subtraction and noise is presented. The CAMShift algorithm is used to identify the moving target based on the color distribution of the target histogram. The CAMShift algorithm adaptively adjusts the track window`s size and the color distribution pattern of targets during tracking and uses a background-weighted histogram to distinguish the target from the background. The Bhattacharya coefficient is used to handle lighting by comparing the target histogram with a reference histogram. To account for occlusions multiple Kalman filters are used in conjunction with the CAMShift algorithm a control scheme is designed for pan and tilt control to keep the tracked object in the field of view. The system is practically implemented and tested using the Arduino platform and a cheap low cost gimbal stabilized camera.

Corresponding author: S.M. Vaitheeswaran (e-mail: smvaithu@nal.res.in).
This paper was submitted on March 26, 2014; revised on September 28, 2015; and accepted on November 5, 2015.

The rest of the paper is organized as follows: Section II presents the original mean shift and CAMShift algorithms. Combined CAMShift-Kalman filter scheme is outlined to handle occlusions. Section III discusses the implementation of a Pulse Width Modulation (PWM) scheme to control the gimbal system of the camera to keep the object in the field of view over a wide angular range. Experimental results are presented in Section IV. Section V concludes the scheme.

II. Detecting and Tracking Target Using Fixed Camera

The continuous adaptive mean-shift (CAMShift) algorithm is used to detect and track the target from the video frame of fixed camera. CAMShift algorithm is the modified version of the mean shift algorithm. Mean shift is a kernel-based tracking method which uses density-based appearance models to represent targets. The method tracks the targets by finding the most similar distribution pattern in a frame sequences with its sample pattern by iterative searching.

CAMShift basically works with a multicolor histogram to represent the target. This accumulated histogram is used to compute the probability distribution of the corresponding target in every frame for CAMShift tracking. A popular approach is to generate the target color histogram with a weighted scheme.

CAMShift basically works as follows. Firstly, initial search window of the target is selected and its color histogram is computed. Each frame of the sequence afterwards is converted to a probability distribution image relative to the target`s histogram. Then the new size and location of the target are computed via mean-shift from this converted image, and are used as the initial size and location of the target for the next iterations of the algorithm.

The principles of our algorithm are shown in **Figure 1** and can be summarized in the following steps. [10]

1. Initialize the search window`s location and size.
2. Calculate the histogram of initial search window.
3. Calculate the probability distribution image of the current frame.
4. Calculate the new location and size of the target search window using mean-shift.
5. Use the new location and size obtained in step 4 to reinitialize the search window in the new frame, and jump to step 2.

1) Search window initialization

In the CAMShift algorithm, initial location of the target is selected manually by user. After manually locating the target by the surrounding rectangle, its two dimensional color histogram is calculated for further processing in the next steps.

2) Color histogram generation

To obtain robust results, HSV (Hue Saturation Value) color space is used in our algorithm for the color histogram generation. HSV color space separates out color (H) from its saturation(S) and brightness (V) values, which would improve our tracking performance. For target representation, only hue (H) color channels that represent the target`s main color features is used. In our project work, red blob is chosen as moving target. The histogram of red blob is shown in Figure 2.

3) Probability distribution image generation

A common method to generate probability distribution image is histogram back projection. Histogram back projection of the target histogram with a frame generates a probability distribution image in which each pixel`s value associates with the corresponding bin of the target histogram. In step 3 chosen to calculate the back-projection of the target histogram with the resulting image of step 2. This will produce a probability distribution image which will be used by mean shift to calculate the target`s new position.

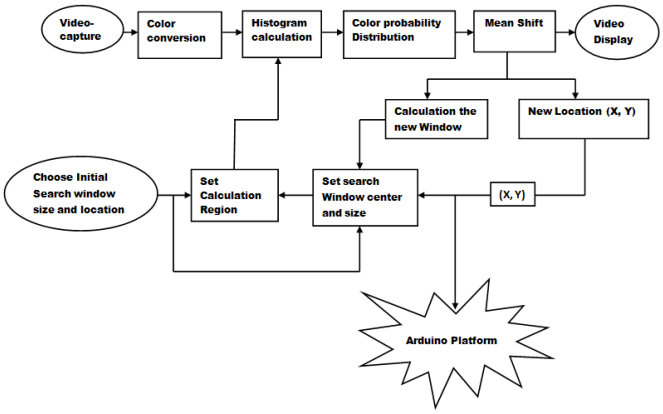

Figure 1: Flowchart of the proposed algorithm

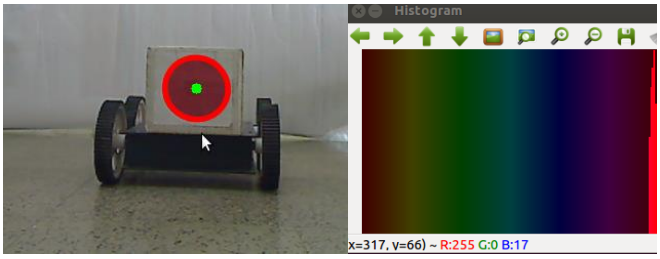

Figure 2: Histogram of the red blob during tracking

4) Mean-shift application

To calculate the new location of a target, the mean-shift algorithm is used. Mean-shift takes probability distribution image and an initial search window, computes the window`s center of mass, and then re-centers the window at the computed center of mass. This movement will change what is under the window, and so the re-centering process is repeated until the movement vector converges to zero. The last calculated center of mass will be the new location of the target. [10]

The following equations are used to calculate the search window`s center of mass(x_c, y_c):

$$x_c = \frac{M_{10}}{M_{00}}$$
$$y_c = \frac{M_{01}}{M_{00}} \quad (1)$$

Here the zeroth and first moments are calculated as:

$$M_{00} = \sum_x \sum_y I(x, y)$$
$$M_{10} = \sum_x \sum_y xI(x, y) \quad (2)$$
$$M_{01} = \sum_x \sum_y yI(x, y)$$

where I(x, y) is the intensity value of point (x, y) in the probability distribution image.

Search window`s new size can be computed as follows:

$$l = \sqrt{\frac{(a+b)+\sqrt{b^2+(a-c)^2}}{2}}$$
$$w = \sqrt{\frac{(a+b)-\sqrt{b^2+(a-c)^2}}{2}} \quad (3)$$

where l and w are the long and short axes of the search window respectively, and a, b, and c are obtained with

$$a = \frac{M_{10}}{M_{00}} - x_c^2$$
$$b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right) \quad (4)$$
$$c = \frac{M_{20}}{M_{00}} - y_c^2$$

And the second order moments are calculated from

$$M_{20} = \sum_x \sum_y x^2 I(x, y)$$
$$M_{02} = \sum_x \sum_y y^2 I(x, y) \quad (5)$$
$$M_{11} = \sum_x \sum_y xyI(x, y)$$

After the computation of M_{00}, M_{10} and M_{01} moments in each cycle, the resulting gradient vector is used to re-center the search window for the next iteration. When the slope becomes zero (i.e. the vector length is zero) the window is considered

Centred and the iteration stops. Following the convergence of the window the M_{20} and M_{02} moments are computed which are next used to compute the target size and aspect information using (3) and (4).

A. *Occlusion handling*

The basic CAMShift algorithm fails to track targets in practical situations such as tracking objects which are fully or partially occluded. To circumvent this problem multiple Kalman filters are used along with the CAMShift algorithm. One Kalman filter is used to track the area returned by the CAMShift algorithm and monitor the change in area due to occlusion, The second Kalman filter is used to track the target position from frame to frame to locate the target in the next frame, The third Kalman filter is used to track the target position and velocity as it passes through occlusions. The moving objects are usually in inertial Newton's systems. Therefore, the states of moving objects in image sequence can be established. The most important states of Newton's system are position, velocity and acceleration. In other words, the second-order model of target object for Kalman filter is established.

From CAMShift, the position of the target object in the image sequence P_t is $P_t = P_{t-1} + (P_{t-1} - P_{t-2})$. Therefore, the velocity and acceleration is obtained by every capturing interval. Assume that the Newton's system with measurement is described as

$$\text{System state:} \quad s(t) = \Phi(t-1)s(t-1) + w(t) \quad (6)$$

$$\text{Measurement:} \quad \mathbf{C}(t) = \mathbf{H}(t)s(t) + v(t) \quad (7)$$

where Φ is the state transition matrix, H is measurement matrix, w(t)and v(t) are noise. The two major steps of Kalman filter are "Prediction" and "Correction".

Prediction (time update):

$$\tilde{s}(t) = \Phi(t-1)\tilde{s}(t-1) \quad (8)$$
$$\tilde{P}(t) = \Phi(t-1)\tilde{P}(t-1)\Phi(t-1) + \mathbf{Q}(t-1)$$

Correction (measurement update):

$$K(t) = \tilde{P}(t)\mathbf{H}^T(t)\left[\mathbf{H}\ (t)\hat{P}(t)\mathbf{H}^T(t) + \mathbf{R}(t)\right]^{-1}$$
$$\tilde{s}(t) = \hat{s}(t) + \mathbf{K}(t)\left[\mathbf{C}(t) - \mathbf{H}(t)\hat{s}(t)\right] \quad (9)$$
$$\tilde{P}(t) = \left[I - \mathbf{K}(t)\mathbf{H}(t)\right]\hat{P}(t)$$

where $\hat{S}$ and $\tilde{S}$ are a priori estimate and posterior estimate, respectively; $\hat{P}$ and $\tilde{P}$ are a priori estimate error covariance and posterior estimate error covariance, respectively. **K** is the Kalman gain and **C** is measurement vector in (t); **R** and **Q** are the measurement error covariance and process noise covariance.

The Process Noise and Measurement Noise covariance matrices for the Kalman filters are shown below

$$Q = \begin{bmatrix} \frac{2p^2\Delta^3}{3\tau} & \frac{p^2\Delta^2}{\tau} & 0 & 0 \\ \frac{p^2\Delta^2}{\tau} & \frac{2p^2\Delta^2}{\tau} & 0 & 0 \\ 0 & 0 & \frac{2p^2\Delta^3}{3\tau} & \frac{p^2\Delta^2}{\tau} \\ 0 & 0 & \frac{p^2\Delta^2}{\tau} & \frac{2p^2\Delta^2}{\tau} \end{bmatrix} \quad (10)$$

where p=8.0 and m=3 for the case considered

$$R = \begin{bmatrix} m^2 & 0 \\ 0 & m^2 \end{bmatrix} \quad (11)$$

where m=3 for the Kalman filter with no occlusion and m=30 for the occlusion case. Both **Q** and **R** are assumed to be independent of each other, white (noise) and with a zero mean and normal probability distribution.

The occlusion detection logic is based on checking the CAMShift target area and area rate of change. If the target is close to or fully occluded the area is less than the threshold (50% of last known tracked value) or is decreasing at a rate of greater than 25%/frame. If an occlusion is detected, the Kalman filters are updated with their own prediction. The search box for the next CAMShift operation is then increased in size based on the rate of change of the slow Kalman tracker, and the next image is read.

If there is no occlusion, the Kalman filter is updated with the CAMShift coordinates and then a prediction is made for the next frame. The prediction is used to center the search window for the CAMShift operation. A new candidate histogram is created for the current ROI and compared to the current target histogram using Bhattacharyya coefficient [11].

B. Lighting/ illumination effects:

After each frame is read and no occlusion is detected a candidate histogram of the target is computed similar to the original histogram. The target and original histograms are compared using the Bhattacharya coefficient for any change in color or lighting. The Bhattacharyya coefficient will return a value from 0 to 1, with 0 being a perfect match and 1 being a total mismatch of the histograms. The equation for the Bhattacharyya coefficient $d_{bhattacharya}$ that compares a target histogram H_t with a candidate histogram H_c is as follows:

$$d_{bhattacharya} = \sqrt{1 - \sum_i \frac{\sqrt{H_t(i) - H_c(i)}}{\sqrt{H_t(i).H_c(i)}}} \qquad (12)$$

III. Tracking the Moving Target Using Pan-Tilt Gimbal Control

Tracking of the moving target is achieved using PAN-TILT gimbal mechanism. Gimbal structure refers to arrangement of servos to achieve 2-degrees of freedom. The two degrees of freedom refers to PAN and TILT movement of servos. PAN servo is responsible for moving the camera in horizontal direction, and TILT servo is responsible for moving the camera in vertical direction. Using CAMShift algorithm the target blob is tracked and the centroid of the detected target is obtained and this value is sent to the microcontroller which controls the two servo motors to keep the target in the field of view of the camera. Communication of the centroid value of target from PC to microcontroller is achieved by using UART (Universal Asynchronous Receive and Transmit) protocol.

A. Servo Motors with PWM Control

Figure 3 indicates how different pulse widths correspond to different position of the motor. The servo receives a pulse every 20 milliseconds (.02 seconds), the length of the pulse width determines how much the motor will rotate. When the PWM is less than 1.5 ms the motor will move to the 0 positions and hold. When PWMtON = 1.5 ms, the motor will rotate to the 90 degree position and for PWM greater than 1.5 ms, the motor will rotate to the 180 degree position. For every 5 microsecond PWM signal there is one degree of rotation.

B. Serial Communication

The tracked centroid of the target is communicated to microcontroller using UART protocol. The centre consists of two values X and Y. At the receiver the controller verifies that corresponding X and Y values are received to control the respective motors. Thus time sharing is achieved to control the motors accurately with respect to the centroid of the target in order to keep it in the field of view of the camera.

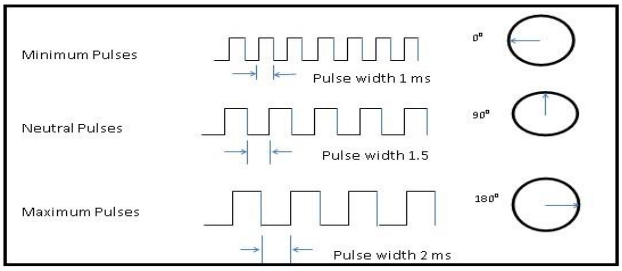

Figure 3: PWM signal to control servo motor

C. Position Control

The objective of position control is to always keep the chosen target within the reference frame of the image as it drifts away from the reference frame.

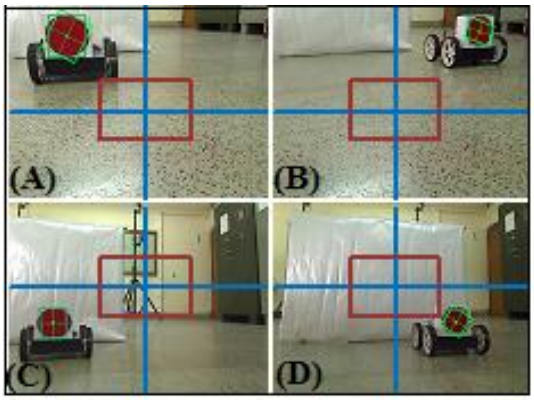

Figure 4: Different location of object on video frame (A) 1st quadrant (B) 2nd quadrant (C) 3rd quadrant (D) 4th quadrant

A reference frame is initialized before tracking procedure. The reference frame can be varied based on the target size as required. The reference input cannot be a single point since it is

almost impossible to stabilize the system on this single point. So a small frame is defined as a reference input which is highlighted in orange color on the video seen as shown in **Figure 4**. Typical coordinates of the reference input are given in **TABLE I** which exactly represents a frame at the center of the video. The differences between reference input seen in the **TABLE I** and current position of the target object are defined as error signal in code. According to this error value, the driving direction of the gimbal is chosen. In other words, according to results of this comparison, gimbal can be driven right-left or up-down.

TABLE I REFERENCE INPUT

Axis	X	Y
X_{refmin}	250	-
X_{refmax}	390	-
Y_{refmin}	-	170
Y_{refmax}	-	310

TABLE II TARGET POSITION IN VIDEO FRAME, MOVING DIRECTION OF GIMBAL AND PERCENTAGE LEVEL OF PWM

Target Position	Gimbal moving direction	PWM pulse width (in ms)
X<125	Left	$1 < t_{ON} < 1.25$
125 < X < 250	Left	$1.25 < t_{ON} < 1.5$
250 < X < 390	Reference frame	$t_{ON} = 0$
390 < X < 515	Right	$1.5 < t_{ON} < 1.75$
515 < X < 640	Right	$1.75 < t_{ON} < 2$

Target Position	Gimbal moving direction	PWM pulse width (in ms)
Y < 85	Up	$1 < t_{ON} < 1.25$
85 < Y < 170	Up	$1.25 < t_{ON} < 1.5$
170 < Y < 310	Reference frame	$t_{ON} = 0$
310 < Y < 395	Down	$1.5 < t_{ON} < 1.75$
395 < Y < 480	Down	$1.75 < t_{ON} < 2$

For instance, if the current position of the target object is close to the center of X axis and far from the Y axis, it means that the gimbal should be made a thrust to up or down in order to centralize Y axis. If it is close to the center of Y axis and far from the X axis, the gimbal should be made a thrust right or left in order to centralize X axis. After deciding right-left or up-down pairs, the algorithm chooses one of the right or left pairs and one of the up or down pairs to decide the moving direction of the gimbal.

After deciding the motion direction of the gimbal, error signal goes through the controller. The controller output is used to decide the pulse width of the PWM signal. There is a larger frame experimentally set around the reference frame. Typical boundaries of the larger frame, the reference frame and gimbal moving direction control through PWM signal used are shown in **TABLE II**.

IV. EXPERIMENTAL RESULT

Practical experiments are conducted in order to test the performance and robustness of the proposed vision based target following which are described in the forthcoming sections.

The experimental setup consists of an autonomous leader and a vision based follower with a PAN-TILT gimbaled camera mounted on the follower. The initialization of the follower vehicle is performed and the tracking of the target is initialized as illustrated in **Figure 5(A)**.

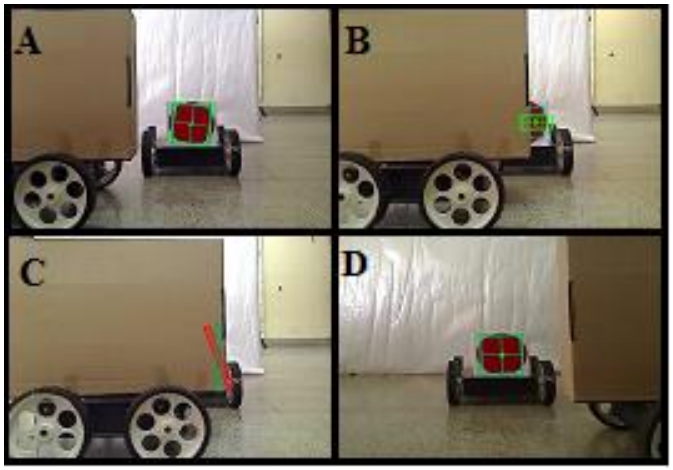

Figure 5 Occlusion detection and recovery of target vehicle in real time

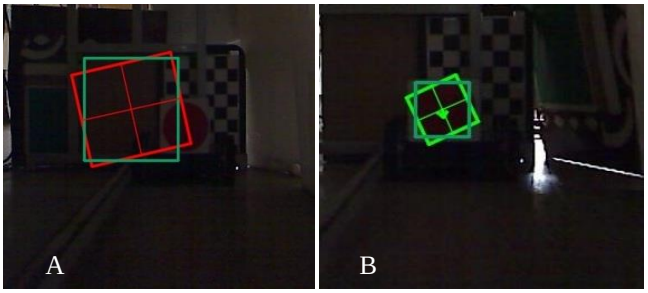

Figure 6: Target tracking response in dark region. (A) CAMShift (B) CAMShift with Kalman filter

In order to demonstrate the occlusion detection an intrusion vehicle with different velocity is passed between the two vehicles. **Figure 5(B)**, and **(C)** illustrates the target occlusions

while the intrusion vehicle is crossing the formation vehicles. The recovery from occlusion is illustrated in **Figure 5 (D)**. This visual representation of the algorithms shows the success and effectiveness for the methods described above for the detection and tracking of objects through occlusions are robust and useful. This process is repeated for multiple environments, as well as in real-time, and the algorithm works robustly for all of the test cases studied, illustrating the overall validity of the assumptions made.

Figure 6 illustrates effects of illumination when the vehicle moves from a bright area into a shaded area. Even though there is a sharp change in illumination, the vehicle is tracked well. The CAMShift algorithm with no Kalman filter (**Figure 6(A)**) fails to track the vehicle when it moves from bright to dark region, but localizes the vehicle correctly when it moves back into the bright region. The CAMShift algorithm with Kalman filter (**Figure 6 (B)**) tracks the vehicle when it moves from bright to dark region, and also localizes the vehicle correctly when it moves back into the bright region.

In all cases the control algorithm is able to keep the leader vehicle in its field of view.

V. Concluding Remarks

The work successfully demonstrates the implementation of the Back Projection, CAMShift, and Kalman estimator algorithms to track objects and maintain track in a video sequence or in real time in the presence of occlusions, and to reacquire the objects when they reappear. The Kalman filter enhances the CAMShift function reducing the number of cycles needed for convergence and proves to be very effective and works well for the majority of situations studied. Overall, the algorithm proposed is both a useful and powerful demonstration of tracking objects, especially when they become occluded and under different lighting conditions. The PWM controller aids vision based convoy–leader follower formation by keeping the leader in its field of view at all times, although the ability to maintain distance between them is not discussed in the current work and will be the subject of another paper.

Acknowledgment

The authors SMV, MKB NAL gratefully acknowledge the funding received from AR&DB, GoI under the NPMICAV program to carry out the above work. The work is carried out at National Aerospace Laboratories, CSIR, Bengaluru, Karnataka, India.

References

[1] A. Talukder, L. Matthies. "Real-Time Detection of Moving Objects from Moving Vehicles Using Dense Stereo and Optical Flow". IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), Sendai, Japan, 28 September–2 October 2004;Volume 4, pp. 3718–3725. CrossRef

[2] P.Tissainayagam, D. Suter," Object tracking in image sequences using point features", Pattern Recogn. 2005, 38, pp. 105–113. CrossRef

[3] A. Milella, R. Siegwart," Stereo-Based Ego-Motion Estimation Using Pixel Tracking and Iterative Closest Point", IEEE International Conference on Computer Vision Systems (ICVS '06), New York, NY, USA, 5–7 January 2006; pp. 21–21. CrossRef

[4] M.Leslar, J. Wang, B. Hu, " Comprehensive utilization of temporal and spatial domain outlier detection methods for mobile terrestrial lidar data", Remote Sens. 2011, 3, pp. 1724–1742. CrossRef

[5] C.H. Huang, Y.T. Wu, J.H. Kao, M.Y. Shih, C.C.Chou, " A Hybrid Moving Object Detection Method for Aerial Images", Advances in Multimedia Information Processing (PCM 2010);

[6] G.Qiu, K. Lam, H. Kiya, X.Y. Xue, C.C. Kuo, M. Lew, Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6297, pp. 357–368.

[7] H. Samija, I. Markovic, I. Petrovic, " Optical Flow Field Segmentation in an Omnidirectional Camera Image Based on Known Camera Motion", 34th International Convention MIPRO, Opatija, Croatia, 23–27 May 2011; pp. 805–809.

[8] N. Suganuma, T. Kubo, Fast Dynamic Object Extraction Using Stereovision Based on Occupancy"Grid Maps and Optical Flow" 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Budapest, Hungary, 3–7 July 2011; pp. 978–983. CrossRef

[9] Z.Jia, A. Balasuriya, S. Challa, "Sensor fusion-based visual target tracking for autonomous"vehicles with the out-of-sequence measurements solution", Robot. Auton. Syst. 2008, 56, pp. 157–176. CrossRef

[10] J. G. Allen, R. Y. D. Xu and J. S. Jin, "Object tracking using CamShift algorithm and multiple quantized feature spaces", In ACM International Conference Proceeding Series, Vol.100, pp. 3-7, 2004.

[11] D. Comaniciu, P. Meer, Mean shift: a robust approach to feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 603.Adaptive Feature Selection and Scale Adaptation", ICIP 2007. IEEE International Conference on Image Processing, 2007.