International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue
National Conference "NCPCI-2016", 19 March 2016
Available online at www.ijrat.org

Exploring Robust Vehicle Detection and Tracking Methods for Various Illumination Settings using Infrared Thermographic Images

Rheeya Uppaal
Computer Science Department, Symbiosis Institute of Technology
Email: rheeyauppaal@gmail.com

Abstract-Today's world has been populated by vast amounts and varieties of traffic. In such a scenario, it becomes essential to have intelligent systems which can monitor traffic and extract valuable information from the same. Vehicle data, being necessary for many aspects of transportation and traffic engineering, is thus of utmost importance.

This paper aims to describe a system that shall detect and track vehicles in real time, using near infrared taken from an infrared thermographic camera. While many such systems are currently in existence, most of them use regular cameras with visible light images as input, making the accuracy of such systems dependent of varying lighting and weather conditions. The robust tracking system proposed in this paper aims to overcome this issue by using infrared images instead.

The proposed algorithm specifies a basic series of image processing steps on the input from the infrared thermographic camera to eliminate noise and other unnecessary parts of the image. The processed input is then sent to a tracker based on the Continuously Adaptive Meanshift algorithm, which then tracks the vehicles. Thus, traffic detection and tracking can be achieved.

Index Terms- Intelligent Transportation Systems; vehicle detection; vehicle tracking; infrared

1.INTRODUCTION

Traffic surveillance is extremely important for the development of better infrastructure. It helps increase efficiency of roads, reduce the volume of traffic and develop better infrastructure efficiently. It also allows for better utilization of roads and gives an accurate estimate of the ratio of vehicles to people. [1]

Thus, developing vision based traffic measurements are of utmost importance in the field of Intelligent Traffic Systems (ITS). Factors like detecting vehicle speed and position and tracking vehicles in multilane environments were not possible using conventional methods. [2]

In the present scenario, there exist many vision based traffic measurement systems which successfully detect vehicles. However, these VDS (Vehicle Detection Systems) face certain challenges.

Occlusion poses as a challenge in vehicle detection, as it is hard to detect both vehicles when one occludes the other. In addition, the shadows cast by vehicles also serve as a hindrance. Many algorithms handle occlusion in daytime scenarios, but fail in low lighting. [3]

Noise in the video also serves as an obstacle in the running of accurate VDS. The noise can exist due to various reasons. It can be introduced due to external factors like inefficient handling of video capture, by permitting dust to cover the camera lens or not creating a steady base for the camera. It can also be introduced due to the poor image quality, which is especially true for low light conditions. Apart from this, there is an inherent amount of noise which shall always exist in the video in most circumstances. [3]

One of the hardest challenges in creating efficient and accurate VDS is handling varying lighting conditions. Most daytime methods of detection lose their accuracy when applied to night-time detection. [4] Due to this, existing systems run two separate algorithms for daytime and night-time detection. Typically, headlights or taillights are used for detection during night-time. Such methods yield low accuracy. In addition, these methods do not account for lighting scenarios associated with weather conditions like fog, mist, rain etc.

VDS often require vehicles to be detected and tracked from various angles and positions. This can serve as a challenge, since many systems are specifically designed for exactly one orientation and scenario for detection and tracking. [5]

1.1. *Infrared Detection and Tracking*

Infrared waves are invisible to the human eye. In the electromagnetic spectrum, infrared radiation exists between the visible and microwave regions, with wavelengths between 0.75 and 1000µm. The wavelength region which ranges from 0.75 to 3µm is known as the near infrared regions. The region between 3 and 6µm is the mid-infrared and radiation with a wavelength greater higher than 6µm is called far infrared. [6]

An infrared sensor is defined as an electronic instrument, used to sense particular characteristics in its surroundings. It does this by either emitting or detecting infrared radiation. These sensors can also measure the heat emitted by an object and can detect motion. [6]

There are two basic designs for electronic IR imagers. The first is the scanner, where the scene is scanned in small parts, to finally create the entire image. The other type of IR imaging is the starer, which 'stares' because it focuses the image onto an extended focal plane. Image resolution in a staring scanner is limited by the number of elements in the array, whereas in a scanning system it is limited by the size of the scanning dot. [7]

All objects which have a temperature greater than absolute zero (0 Kelvin) possess thermal energy and are thus sources of infrared radiation. Thus, infrared detection provides a means of 'seeing' objects in both, low light and well-lit conditions. [7]

The images obtained using a thermographic infrared camera or near infrared camera showcase a clear contrast between the vehicles and their lower temperature surroundings, or background, even in poor visibility conditions. [5] This can clearly be seen in Figure 1.

Due to the fact that infrared cameras detect heat, and not visible light, systems using these cameras are robust to illumination changes and are unaffected by weather conditions like fog, mist etc. The tyres, followed by the engines and exhaust pipes, are the hottest parts of the vehicle, and are substantially hotter than the surroundings. When these surroundings contain cooler elements like rain, mist or fog, they are invisible to the infrared camera due to the temperature difference, thus facilitating successful detection, even in low light and bad weather conditions. [8]

However, even using infrared footage for vehicle detection and tracking has certain drawbacks. The images obtained from such cameras can sustain a loss of special and temporal information, due to the fact that infrared cameras get saturated more easily than visible light cameras. There is also a certain level of background-foreground similarity. [8]

2. THE CURRENT SCENARIO

Systems today use two components in their framework when tracking: an *inference framework* (e.g., Kalman filter, particle filter, etc.) and a *target representation* (e.g., linear subspace, sparse representation, etc.). The Kalman filter provides optimal solutions for linear Gaussian models. The Particle Filter, or the sequential Monte Carlo method, is a very popular approach. [3]

There exist a small number of vehicle detection algorithms using infrared images. However, none of these accurately detect vehicle positions in heavy traffic. [9]

Active sensors, like millimetre wave radars, have recently been used in vehicle detection. While they detect vehicles in poor visibility conditions with sufficient accuracy, interference among radar poses an issue. Moreover, radar gives limited information regarding the shape of the object, and can thus not be used for distinguishing. [8]
Yean-Jye et al proposed a vehicle classification method which uses a modified edge sharpening techniqueto filter out noise and locate edges of bridge clusters simultaneously. The classifier includes a modified kNN method, with a linear time complexity. The procedure of image analysis is divided into four stages: image acquisition, data processing, feature extraction and object classification. The data processing involves determining the size of the scanning window, calculating the gradient, determining the beginning and end edge of bright segments, creating bright clusters and discarding noise clusters. [10]

Iwasaki et al proposed multiple methods for using infrared video to detect and track vehicles in low light conditions. The first method uses pattern recognition on Haar feature based classifiers on the windshield area and its surroundings to detect vehicles. However, the accuracy of this method reduces when low temperatures are prevalent. The second method uses the thermal energy reflected by the tyres onto the road, as detection targets. This method was only accurate for vehicles on central lanes. The third method is a modified version of the second method with a higher level of accuracy. [2,4,5,8]

3.EXPERIMENTAL SETTING

There exist three main methods of night vision:

Low-light imaging, Thermal imaging and Near Infrared Illumination. Here, the term 'night vision' is referenced as a technology that provides vision in complete darkness, and improves the quality in low light environments.

Low light imaging is based on the use of image intensifiers, which are commonly used in night vision goggles. More recently, on-chip gain multiplication CCD cameras are being popularly used for low-light security and astronomical observation. Image intensifiers amplify the available light to achieve better vision. While they have high levels of sensitivity in low light conditions, the performance degrades in regular daytime scenarios. [11]

Thermal imaging methods do not require any ambient light to perform. They operate on the principal that all objects emit infrared energy as a function of their temperature. A thermal imager uses infrared radiation emitted from objects to create an electronic image. Due to this, these methods are unaffected by visual obstructions like smoke, fog and haze. Some thermal cameras show images in false colour, which simplifies the task of distinguishing between objects. However, thermal imaging cameras are generally very expensive, and thus impractical to be used in this project. [11, 12]

Near infrared illumination is a popular and inexpensive method, where a device that is sensitive to invisible near infrared radiation is used in conjunction with an infrared illuminator. Infrared illumination then illuminates the otherwise low-light illumination scene, producing reasonable image quality in low-light scenarios. The video is in black and white, with white or black parts representing hotter regions, depending on the camera used. The infrared illuminator can be a filtered incandescent lamp, an LED type illuminator or a laser type illuminator. Near infrared illumination are also referred to as 'starlight' vision systems, due to the fact that they amplify light in low-light scenarios. [11, 12]

This paper makes use of near infrared illumination for the following reasons: [11]

1. It is the cheapest, and thus easiest to practically implement, method of infrared vision.
2. Shadows are eliminated and lettering, numbers and objects can easily be identified.
3. The method can be used for high speed detection, which is crucial to the project as it aims at detecting vehicles in real-time.
4. The video capture is unaffected by night time fog, mist, rain, snowfall etc. It is also not affected by vehicle windows, which permits accurate and efficient tracking to be performed from within a vehicle.
5. The variability of ambient light is eliminated.

For the experiment, a fixed, near infrared camera is used. It is positioned above the road, opposite to the direction of incoming traffic, so that an angled top view of vehicles is obtained in the video footage.

Figure 1. A screenshot from the sample data used for the experiment described in this paper

The distance of the camera from the road is such that accuracy is not degraded.

4.IMAGE PROCESSING AND NOISE REMOVAL

The detection of vehicles is performed solely through image processing methods. This improves the efficiency of the system for working in realtime, as the image processing operations performed on the input are not time consuming.

In the course of this experiment, to process the video in real time, frames are extracted from the video, one at a time. A series of various image processing techniques are then applied to every frame. Various techniques were considered or used to process the video, remove noise and then detect the moving vehicles. These methods have been described as follows.

4.1. *Greyscale Conversion*

The extracted frame is converted to greyscale, primarily to improve computation time of the system. RGB images contain three values for every pixel: the red, green and blue intensities. Greyscale images contain only one value instead of three. Even though the original image is already in greyscale, it has been saved as a regular RGB image and is thus converted to greyscale. This makes no difference to the accuracy performance of the system, but helps reduce computation time.

4.2. *Double Difference*

The Double Difference method, also called the Three-Difference method, is an image processing technique used to identify moving objects in a video. It has been used for vehicle detection in [13] and [14]. The method uses three frames and calculates two difference values. It then calculates a flag value for each pixel. If both difference images have a value greater than a given threshold value, the flag is set to 1. The final image is created by making all pixels with flag=1 white and the remaining pixels black. Mathematically, this can be defined as follows.

Let $\{F_n\}$ be the frames of the input video. The nth difference image is defined as:

$D_n(i, j) = F_{n-1}(i, j) - F_n(i, j)$ Where $F_n(i,j)$denotes the pixel value at (i, j) in the nth frame, $D_n(i, j)$ is the absolute difference.

The flag is defined as:

Flag= $(D_{n+1}(i,j)>T) \wedge (D_n(i,j)>T)$ Where T is a given threshold. When both difference values at coordinate (i, j) in two consecutive difference images are larger than a given threshold T, *flag* is 1.

(a) (b)

Figure 2. The image on the left, (a) is a frame from the original video. (b) shows a frame converted into greyscale. There is no visible difference between both images, but the second image uses less memory space.

The Double Difference image, DDn is defined as

$DD_n(i,j)$= 1, flag=1
0, otherwise [14]

When applying this method to the experiment data, very poor results were achieved, with no accurate detection of vehicles. We believe this is due to large amounts of noise in the data, and varying illumination in the course of the video. Due to this, the double difference method was not used in the final algorithm.

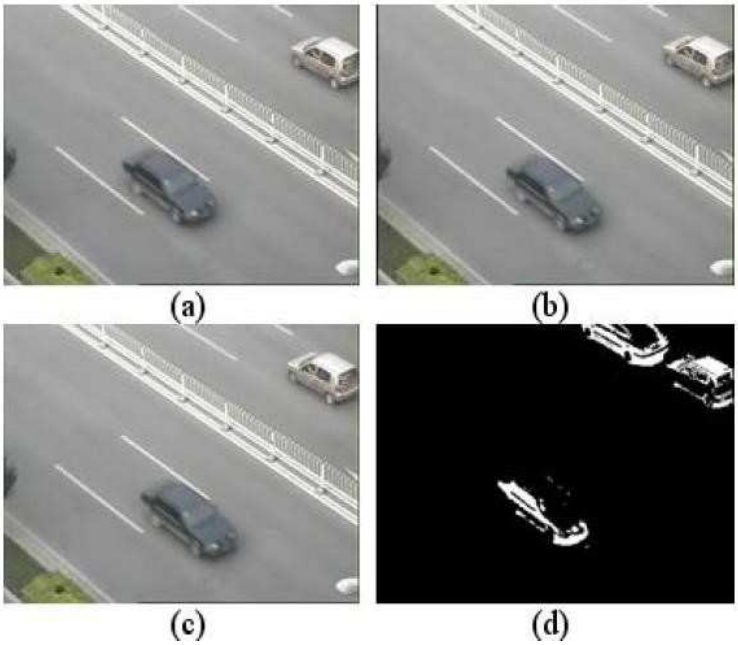


Figure 3. An example of the double difference method. (a), (b) and (c) represent three consecutive frames in a video, while (d) shows the double difference image. The moving objects, i.e. the vehicles, have been highlighted in white while the rest of the image is black. [13]

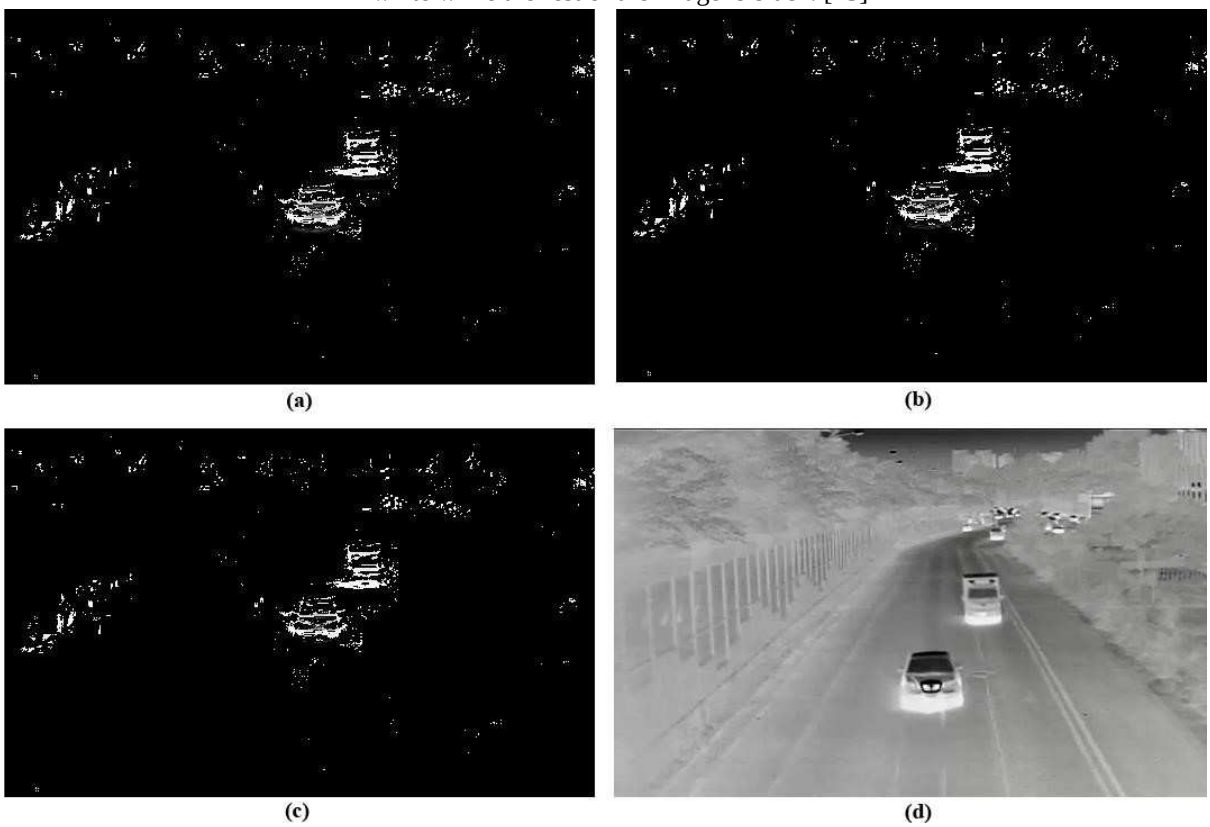


Figure 4. Double Difference method applied on the experiment data with different threshold values. (a) Threshold=10 (b) Threshold=25 (c) Threshold=50 (d) Original frame from the experimental data.

4.3. *Background Subtraction*

The Background Subtraction method described here has been proposed as an alternative to the double difference method. The BackgroundSubtractorMOG2 function created by OpenCV was used for this. [15] The algorithm segments background and foreground on the basis of a Gaussian function, by selecting the appropriate number of Gaussian distribution for each pixel. This provides better adaptability for varying illumination.

The technique generates a foreground mask, which is a binary image containing pixels belonging to moving objects, when using a static camera. This is done by performing a subtraction between the current frame and a background model, which is the static part of the first frame of the video. Background modelling involves two steps: (1) Background initialization, where an initial model of the background model is created, and (2) Background update, where the model is updated to adapt to changes in the scene. [16, 17]

The use of the background subtractor proposed by Zivkovic et al provided far better results than the double difference method described in the previous section. While there exists a large amount of noise, the noise is sparsely distributed. The falsely detected moving parts of the image are very small in size, and can hence not be confused with vehicles.

4.3. *Erode and Dilate*

Eroding and Dilating are two morphological transformations used in conjunction with each other. Their primary purpose is to remove noise from an image. Erosion thins all the parts of the image detected as moving objects, cancelling out the noise completely. Dilation then returns the rest of the detected image, i.e. the moving vehicles, back to their original form by thickening them. [18]

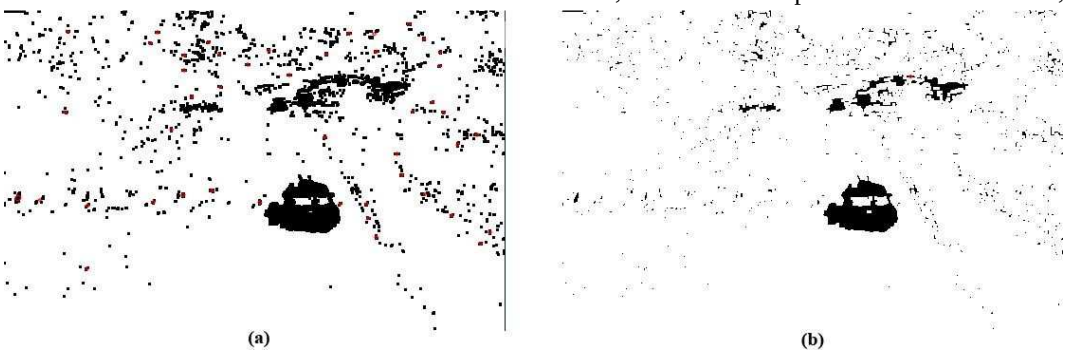
The Erode method computes a local minimum over the area of the kernel. As the kernel scans over the image, the minimal pixel value overlapped by the kernel is replaced by the pixel under the anchor point with that minimal value. Erosion makes the bright areas of the image thinner and the dark parts thicker.

The Dilate method convolutes an image with a kernel which can have any shape or size, using a square or circle. The kernel has a defined anchor point, which is usually the centre of the kernel. As

Figure 5. Background subtraction applied on the experimental data using BackgroundSubtractorMOG2. The method has provided better results than the double difference method. The colours of the image have been inverted to highlight the moving parts of the image in black.

the kernel is scanned over the image, the maximum pixel value overlapped by the kernel is replaced by the pixel in the anchor point position with that maximum value. This causes bright regions in the image to thicken.

The Erode and Dilate method was used in conjunction with Background Subtraction to obtain a cleaner version of the image with lesser noise.



(a) (b)

Figure 6. (a) Erode method used on a frame from the experimental data with Background Subtraction. (b) Erode and Dilate method used on a frame with Background Subtraction.

(a) (b)

Figure 7. (a) Image obtained after background subtraction and thresholding with a threshold set to a pixel intensity of 75. (b) The same image, without thresholding.

4.4. *Thresholding*

The process of thresholding has been included as a method of noise removal. There exist three main types of thresholding: Simple, Adaptive and Otsu's Binarization. [15] In simple thresholding, a pixel is turned on (assigned a particular colour) if its intensity value crosses a user defined threshold. Otherwise, the pixel is turned off (assigned another colour). Adaptive thresholding calculates different thresholds for different parts of an image, and then uses simple threshold for each region of the image. This makes the thresholding more sensitive to illumination changes in the image.

Simple thresholding was selected over adaptive thresholding, on the frames of the experimental data, to reduce computation time. However, there was no noticeable difference in the processed images after applying this method.

It must be noted that the method deteriorates detection performance when using blob features to track vehicles. Applying a threshold on the pixels of a frame makes the windscreens of vehicles white, creating a gap in the 'blob' of pixels which is the vehicle. This can be seen in Figure 7.

4.5. *Blurring*

Blurring, or 'Smoothing', is an image processing operation often used to remove noise. The process involves applying a filter to an image. The output pixel's value is calculated as a weighted sum of input pixel values. Based on the type of filter used, different output images are achieved.

When using a normalized box filter, every output pixel is the mean of its kernel neighbours. The Gaussian filter is useful, but functions slowly. The process convolves each input pixel with a Gaussian kernel and adds the results to create the output image. The median filter replaces each pixel with the median of its neighbouring pixels. The bilateral filter blurs the image, while ensuring the edges of

the image are clear and unaffected. Weights, with two components, are assigned to all neighbouring pixels. The first component is the weighting used is the Gaussian filter. The second takes the difference in intensity of the pixel and its neighbours into account. [15]

Applying various types of blurring effects was considered to improve the image quality of the frames to improve the accuracy of tracking. However, none of the blurring techniques applied improved the accuracy of the tracking mechanism.

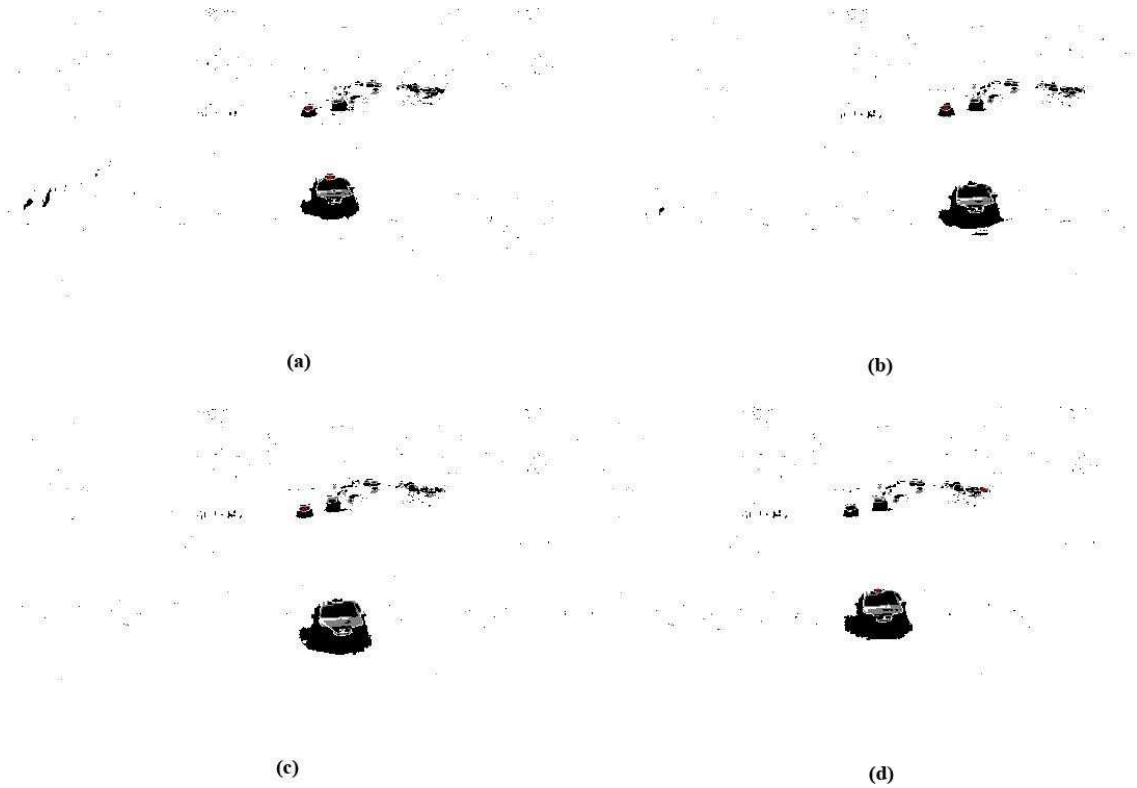


Figure 8. Blurring used in conjunction with background subtraction. The four images show different kernels used for blurring. (a) Normalized Box Filter (b) Gaussian Filter (c) Median Filter (d) Bilateral Filter

5.VEHICLE DETECTION

To summarize the details of the previous section, the algorithm to process every frame of the video is as follows:

(1) Extract one frame from the video
(2) Convert the image to greyscale
(3) Perform background subtraction to highlight moving objects
(4) Apply thresholding on the image to reduce noise

After the image has been processed using the aforementioned method, detection techniques can be accurately applied for the system to detect the moving vehicles in the frames of the input video.

As all the moving parts of the video, i.e. the vehicles, have been highlighted in black, the detection method needs to identify the clusters of black pixels and return their pixel positions. These regions of interest can be detected using different features. Some commonly used features are edges, corners and blobs. [15] The methods considered to detect features of the image have been described as follows.

5.1. *Shi-Tomasi Corner Detection*

A corner in an image is an intersection of two edges, where the directions of both edges change. Due to this, the gradient of the image at a corner has high variation and can easily be detected.

The method finds the difference in intensity for all displacements in all directions. The points with the highest computed values are marked as corners. This method performs better than Harris corner detection as it uses a modified scoring function to determine if a window can contain a corner or not. The OpenCV function cv2.goodFeaturesToTrack performs Shi-Tomasi corner detection to find the N strongest corners in an image. [19]

The method was found to be infeasible for this experiment as the value of N is dynamic and cannot

be predicted. Apart from this, the method detects the corners of vehicles, but the mapping of corners to vehicles is not in the scope of this paper.

5.2. *Blob Detection*

To extract blob features from an image converted to binary, the centres of connected components are calculated by finding the contours of the image. After this, the centres from a series of images are grouped. Centres with a low displacement value form a group which corresponds to one blob. The low displacement value can be defined and altered.

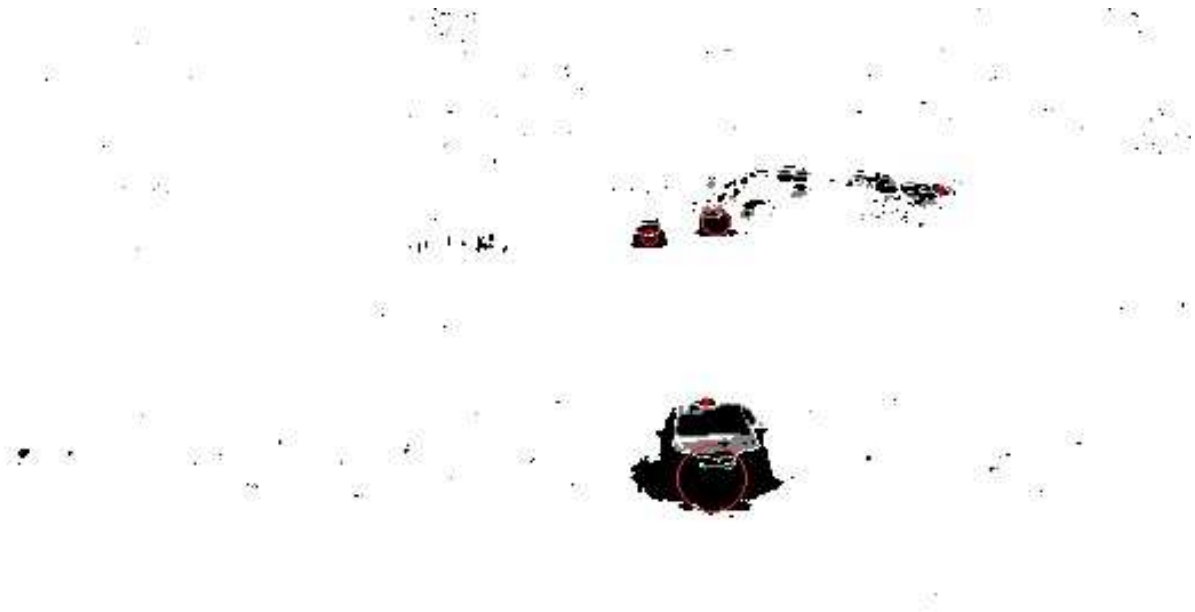


Figure 9. Blob features used to extract the locations of vehicles in individual frames. The regions of interest have been marked in red.

A blob feature is a cluster of pixels that differ from the surrounding regions on the basis of properties like brightness or colour. All points in a blob are similar to one another. In this experiment, the blob features are the vehicles, whose pixels have made black and differ from the surrounding white regions.

The blobs can be calculated on the basis on colour, area, circularity or convexity. [15]
Blob features are more suited to the needs of this experiment as the vehicles, or regions of interest, are clusters or blobs of pixels. The detection method provided reasonable results, and has thus been used in the experiment.

6.VEHICLE TRACKING

The detection mechanism places a bounding circle around every vehicle of interest. It also sends the pixel coordinates of every vehicle to a tracker. The tracker uses these coordinates to determine the object in the bounding polygon. It then tracks the object in every frame, iteratively. The trackers considered for the experiment have been described below.

6.1. *Optical Flow*

Optical Flow is defined as the 2D vector field where each vector is a displacement vector which shows the movement of the same points in consecutive frames. The method works on certain assumptions: (1) The pixel intensities of an object are constant in consecutive frames. (2)
Neighbouring pixels have similar motion.

The Lucas-Kanade method has been used to calculate the optical flow. The method solves the basic optical flow equations for all pixels using a weighted version of the least squares principle. It is less sensitive to image noise than point wise methods. However, being a local method, it does not provide flow information about the uniform interior regions of an image. [20]

OpenCV'scv2.calcOpticalFlowPyrLK() function calculates the sparse optical flow for moving objects in an image. An alternate version, cv2.calcOpticalFlowFarneback() calculates the dense optical flow. The sparse optical flow method tracks the centres of moving objects while dense optical flow is used to showcase the moving object itself. [15]

When using the Optical Flow method in the experiment, there was no need to perform any vehicle detection algorithms. This is because the method automatically detects and then tracks the moving objects in the consecutive frames of the experimental data. Figure 10 shows the performance of the Lucas-Kanade tracking method on the experimental data. While the method is fast and easy to implement, it is often erroneous in correctly detecting boundaries of moving objects, giving a large number of false positive detections.

(a) (b)

Figure 10. (a) The tracking results with the Lucas-Kanade tracking method. The blue dots encircle the moving vehicles. The dots have been masked over the original unprocessed frames. (b) The result with the processed version of a frame using which detection and tracking has been performed. The algorithm includes the shadow of the vehicle in the region on interest.

6.2. *CAMShift*

The Mean Shift algorithm is a widely used algorithm in the domain of tracking. It takes the histogram back projected image and initial target location as input. Movement in consecutive frames alters the histogram back projected image. The algorithm uses this change to move the target window to a new location. The window is iteratively moved across the picture until it converges on the region of maximum pixel density. This region contains the moving object. The process is then repeated for all frames of the video. [15]

The CAMShift or Continuously Adaptive Mean Shift algorithm is a modified version of the Mean Shift algorithm. It takes into account the size and rotation of the target object and alters the window accordingly. The algorithm applies Mean Shift and updates the window after convergence. [21]

The algorithm can be specified as follows. For each frame of the video,

(1) Convert the frame to HSV colour space
(2) Calculate the histogram back projection of the frame
(3) Apply mean shift on the frame, marking a bounding polygon around the ROI
(4) Modify the bounding polygon to adapt to the size and orientation of the ROI

With the algorithm being iterative, termination criteria must be specified: the first ensures the algorithm does not cross a specific number or iterations; the second ensures the algorithm terminates if it has already converged before reaching the iteration limit. This improves run-time performance.

CAMShift has certain limitations:

(1) The ROI must be manually given to the algorithm as input
(2) The algorithm can only track one object at a time
(3) As only the hue component of the HSV colour space is considered, only single shade objects are optimally tracked

The limitations have been overcome in the experiment. The methods defined in sections 4 and 5 automatically detect the region of interest and send their coordinates to the algorithm, solving the first limitation. To tackle the second limitation, CAMShift is individually called for each vehicle detected. [14] The third limitation does not hinder the system performance as the greyscale images used do not consist of multiple shades.

The Mean Shift algorithm has been used frequently for tracking and is a well-tested method. [22] CAMShift is also an efficient algorithm, which runs without any time lag in real-time. For these reasons, the CAMShift algorithm has been used in this experiment.

(a) (b)

Figure 11. (a) The final results of the proposed algorithm. The vehicles have been marked with red circles. To compare performance, (b) shows the results when using optical flow for tracking.

All the methods described in the previous sections, and all combinations of sequences of methods, were tested to determine the sequence with the best results was determined. The final algorithm to detect and track vehicles using images from an infrared thermographic camera is as follows:

For each frame in the video,
(1)Process the frame:
a. Convert to greyscale
b. Apply background subtraction
Apply thresholding
(2) Find the ROI coordinates using blob features
(3) Track individual vehicles using CAMShift (specified in section 6)

The algorithm proposed in this paper successfully detects vehicles from the infrared video of a thermographic camera. The performance of the final algorithm has been shown in Figure 11.

FUTURE SCOPE OF STUDY

The detection and tracking algorithm proposed in this paper, while successful, does not guarantee a very high level of accuracy. Further study could be performed to improve the performance of the system further.

Many countries and regions contain more categories of traffic than those on regular roads. Vehicle categories could include two and three wheel vehicles, trailers and non-motored vehicles like bicycles, wagons and carts. For example, India has more than fifteen differen categories of vehicles which are present on most roads of the country.
The detection mechanisms for such eccentric traffic shall vary and can be explored in the future. Due to unavailability of diverse data, these following proposals could not be tested, and can thus be considered for future study:

(1) Motored two and three wheeled vehicles: As the tyres shall be the hottest part of the vehicles, they can serve as detection targets.
(2) Two wheeled vehicles without motors (e.g. bicycles): The thermal energy radiated by the driver shall be greater than that from the surroundings. Thus, the driver can be used as a detection target. However, suitable methods must be devised to accurately detect from images with a lower contrast.
(3) Animal driven carts: Similar to the previous case, the animals driving the carts can be used as detection targets.

7.RESULTS AND CONCLUSTION

After testing all the methods in the previous sections, certain conclusions were noted:

(1) The double difference method did now work successfully for the experimental data
(2) Background subtraction (MOG2) was a suitable replacement for the double difference method
(3) Thresholding reduced noise by a very small amount

Acknowledgments

I would like to express my sincere gratitude to KuiljeitUppaal for her encouragement, unconditional support, valuable guidance and direction from the nascence to the completion of this research paper.

REFERENCES

[1] http://www.aboutcivil.org/traffic-volumestudy.html

[2] Iwasaki, Y., Kawata, S., &Nakamiya, T. (2011). Robust vehicle detection even in poor visibility conditions using infrared thermal images and its application to road traffic flow monitoring. *Measurement Science and Technology*, *22*(8), 085501.

[3] Ling, H., Bai, L., Blasch, E., & Mei, X. (2010, July). Robust infrared vehicle tracking across target pose change using L 1 regularization. In *Information Fusion (FUSION), 2010 13th Conference on* (pp. 1-8). IEEE.

[4] Iwasaki, Y., Misumi, M., &Nakamiya, T. (2015). Robust vehicle detection under various environments to realize road traffic flow surveillance using an infrared thermal camera. *The Scientific World Journal*, *2015*.

[5] Iwasaki, Y., Misumi, M., &Nakamiya, T. (2013). Robust vehicle detection under various environmental conditions using an infrared thermal camera and its application to road traffic flow monitoring. *Sensors*, *13*(6), 77567773.

[6] http://www.azosensors.com/Article.aspx?ArticleID=339

[7] http://www.encyclopedia.com/topic/Infrared_Detection_Devices.aspx

[8] Nakamiya, T. (2013). Road Traffic Flow Automatic Monitoring Robust for Various Environments Using Thermal Image Processing. Proceedings in ITS 2013-Inteligent Transportation Systems, (1).

[9] Sivaraman, S., & Trivedi, M. M. (2013). Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavioranalysis.Intelligent Transportation Systems, IEEE Transactions on, 14(4), 17731795.

[10] Lu, Y. J., Hsu, Y. H., &Maldague, X. (1992). Vehicle classification using infrared image analysis. Journal of transportation engineering, 118(2), 223-240.

[11] http://sofradir-ec.com/hownightvisionworks/

[12] http://www.flir.com/cvs/americas/en/view/?id=30 0 52

[13] Xia, J., Wu, J.,Zhai, H., & Cui, Z. (2009). Moving vehicle tracking based on double difference and camshift. In *Proceedings of the International Symposium on Information Processing* (Vol. 2).

[14] Xia, J., Rao, W., Huang, W., & Lu, Z. (2013). Automatic multi-vehicle tracking using video cameras: An improved CAMShift approach. *KSCE Journal of Civil Engineering*, *17*(6), 1462-1470.

[15] http://docs.opencv.org/3.0beta/doc/py_tutorials/py_tutorials.html

[16] Zivkovic, Z. (2004, August). Improved adaptive Gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (Vol. 2, pp. 28-31). IEEE.

[17] Zivkovic, Z., & van der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, *27*(7), 773-780.

[18] Bradski, G., &Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.".

[19] Shi, J., &Tomasi, C. (1994, June). Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on* (pp. 593-600). IEEE.

[20] Baker, S., & Matthews, I. (2004). Lucaskanade 20 years on: A unifying framework. *International journal of computer vision*, *56*(3), 221-255.

[21] Bradski, G. R. (1998). Computer vision face tracking for use in a perceptual user interface.