

# Intelligent Headlight Control by Light Blob Detection

Amarjit Salam<sup>1</sup>, Ashutosh Kumar<sup>2</sup>, Mohammed Idris<sup>3</sup>, Naman Narain<sup>4</sup> and Sangappa S B<sup>5</sup>

<sup>1</sup>*Department of Electronics and Communication, K. S. Institute of Technology, India*

<sup>2</sup>*Department of Electronics and Communication, K. S. Institute of Technology, India*

<sup>3</sup>*Department of Electronics and Communication, K. S. Institute of Technology, India*

<sup>4</sup>*Department of Electronics and Communication, K. S. Institute of Technology, India*

<sup>5</sup>*Professor,PRO, Department of ECE,Visvesvaraya Technological University, India*

**Available online at: [www.ijcseonline.org](http://www.ijcseonline.org)**

**Abstract**— in this paper, we propose an enhanced method for detecting light blobs (LBs) for Intelligent Headlight Control (IHC) using Digital Image Processing Techniques. The main function of the IHC system is to automatically convert high-beam headlights to low beam when vehicles are found in the vicinity. Thus, to implement the IHC, it is necessary to detect preceding or oncoming vehicles. This process of detecting vehicles is done by detecting LBs in the images. Here the algorithm is developed to analyze a frame and the same will be applied to all frames in a video. The area of interest is enhanced by converting the image to binary and thus detecting the LBs. To detect tail lights, red component of the image is extracted. Threshold value can be set depending on the object to be detected. Morphological operations will be performed on the binary image to remove all unwanted objects that are present in the image. Area of the light blob is calculated in the binary image and based on the value of this area, high beam is converted to low beam.

**Keywords**— *Light Blobs (LBs), Intelligent Headlight Control (IHC), Digital Image Processing, headlight, tail light, threshold value, morphological operations, binary image.*

## I. INTRODUCTION

To drive safely at night, high-beam headlights must be kept turned on unless other vehicles are found within a certain distance from the ego-vehicle. However, if high beams are not converted to low beams at the appropriate moment, glare from these lights can temporarily blind the oncoming or preceding drivers, which could lead to serious traffic accidents. Glare is a serious problem for drivers during the night. This is caused due to sudden exposure of driver's eye to high intensity headlight. The high intensity headlight of vehicle in this case causes temporary blindness. This is called the Troxler Effect. [1] Eventually, this becomes the major reason for night accidents. The drivers are found rather negligent in actively switching between high-beam and low-beam. Hence, we developed a project titled "**Intelligent Headlight Control by Light Blob Detection**". Intelligent Headlight Control (IHC) allows the driver to use high beam headlight when required, and it automatically switches the high beam headlight to low beam, when it senses a vehicle approaching from the opposite side.

Our project not only avoids accidents, but also aims at societal concerns by providing safe and comfortable driving.

## II. ALGORITHM FOLLOWED

In this project, two distinct algorithms have been proposed. The first algorithm detects the headlight of the

oncoming vehicle. The second algorithm detects the taillight of the preceding vehicle.

A. There are 7 steps in the Headlight Detection Algorithm. They are:

1. Image Acquisition
2. Image Grayscale
3. Noise Filtering
4. Image Binarization
5. Morphological operations to the image
6. Light Blob area calculation
7. Conversion of High Beam to Low Beam

The above steps have been discussed in detail as follows.

### 1. Image Acquisition

The video is taken at a fixed point without changing the position of the camera. The video can be taken at a lower resolution for reducing the processing time. The video sequence is acquired from a web camera (VGA Camera) since it has low resolution. It can be mounted on the rear view mirror of the ego-vehicle. This results in coverage of a large field of view and minimizes the occlusions of the vehicles. A video sequence needs to be sampled into multiple frames before applying the image processing techniques and those set of frames are the inputs to the

subsequent stages of the system. Hence the built-in function VideoReader ( ) in Matlab was used to extract the frames. The built-in function imread ( ) in Matlab was used to read and convert the extracted frames into readable form. [2]

## 2. Image Grayscale

Converting color images into grayscale images is the first step of the pre-processing. Grayscale removes the color values of an image and simplifies computational time significantly compared to a color RGB image. There are several algorithms to convert a color RGB image to a grayscale image. The function called rgb2gray ( ), which is a built-in function in Matlab is one such method. [3]

## 3. Noise Filtering

After grayscaleing the image, it had to go through the filtering process to filter out any noise in the image. A wide variety of filtering algorithms are available to detect and remove noise, leaving us the required information. The filter used in this work is a median filter, which has a nonlinear operation and is often used to reduce the "salt and pepper" noise. Median filter is more effective than convolution when the goal is to simultaneously reduce noise as well as preserve the edges. The function medfilt2 ( ) is a built-in function in Matlab, which performs the median filtering operation. The output image from the previous operation is passed through the filter to obtain a noise free image. [2], [3]

## 4. Thresholding

After the filtering process, we have to determine the threshold to convert the grayscale image to binary image. The threshold is found by running the program for different distant images, our program has a range of 350m for detecting headlights. This maximum distance is the distance at which street lights and reflectors are removed by thresholding. For our captured video the threshold value is found to be 0.9 for a maximum distance of 350m; i.e., all the pixel values below 0.9 will be assigned to the value 0 (black), and all the pixel values above 0.9 will be assigned to the value 1 (white). The function im2bw (Th) is a built-in function in Matlab, which performs the thresholding operation and converts the image from grayscale to binary, where Th is the threshold value. [3]

## 5. Morphological Operations to the Image

To enhance the binary image, we apply different morphological operations which include:

**1. Filling the holes:** In the binary image, some areas of the headlight have black pixels which need to be filled to get more accurate results. The function imfill ( ) is a built-in function in Matlab, which performs the filling operation on the binary image.

**2. Dilate the image:** We then dilate the image twice the amount to make it clear. The function bwmorph ( ) is a built-in function in Matlab, which performs the dilation operation on the filled binary image.

**3. Cropping the image:** The upper portion of the image is cropped out to remove the glare from street lights. The function imcrop (XData, YData) is a built-in function in Matlab, which performs the cropping operation on the dilated binary image. [3]

## 6. Light Blob Area Calculation

After the morphological operations, we then calculate the number of white pixels present in the binary image. The function bwarea ( ) is a built-in function in Matlab, which returns the number of white pixels in the binary image. [3], [8]

## 7. Conversion of High Beam to Low Beam

After calculating the area of the number white pixels we then compare it to our threshold value of 250. This threshold value is calculated from a binary image in which the vehicle is at a distance of 3000m for headlight detection and at a distance of 150m for tail light detection. [3]

B. There are 7 steps in Tail light Detection Algorithm. They are:

1. Image Acquisition
2. Extraction of Red Component
3. Noise Filtering
4. Image Binarization
5. Morphological operations to the image
6. Light Blob area calculation
7. Conversion of High Beam to Low Beam

### 1. Image Acquisition

The video is taken at a fixed point without changing the position of the camera. The video can be taken at a lower resolution for reducing the processing time. The video sequence is acquired from a web camera (VGA Camera) since it has low resolution. It can be mounted on the rear view mirror of the ego-vehicle. This results in coverage of a large field of view and minimizes the occlusions of the vehicles. A video sequence needs to be sampled into multiple frames before applying the image processing techniques and those set of frames are the inputs to the subsequent stages of the system. Hence the built-in function VideoReader ( ) in Matlab was used to extract the frames. The built-in function imread ( ) in Matlab was used to read and convert the extracted frames into readable form. [2]

2. Extraction of Red Component

Our high beam can also affect the preceding vehicle, since the glare can fall on its side view and rear view mirrors. For detecting the tail light of the preceding vehicle, we extract the red component of the image since all the tail lights are red in color. We do this by acquiring the red, blue and green components pixel values and ANDing them. The function Vsize ( ) is a built-in function in Matlab, which returns the pixel values of the RGB Image. The red pixels are compared with the green and blue pixels. If the red pixels are significant in the image, then it is assumed that the tail light is present. [3]

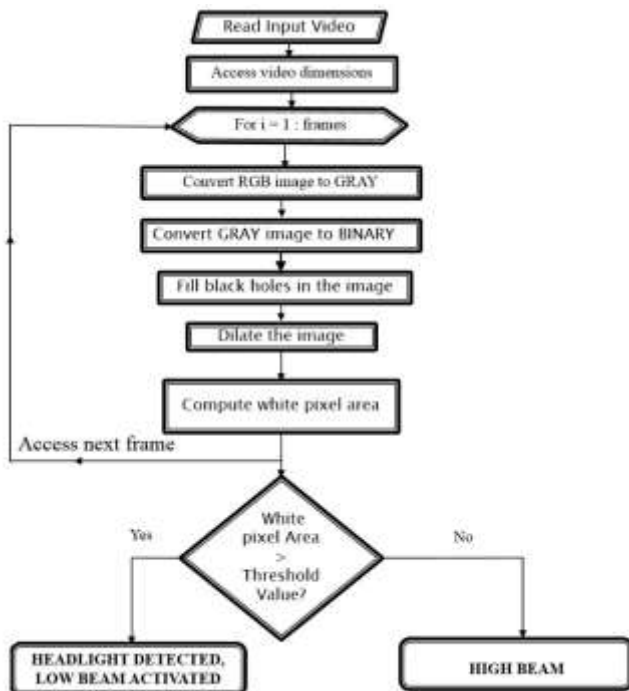
The remaining steps in the Tail light Detection Algorithm are already discussed in the previous section.

III. FLOWCHART FOR HEADLIGHT DETECTION

We can implement this project using Digital Image Processing Techniques. A camera can be fixed at the rear-view mirror or at the dashboard, which records video at 25-30 frames per second.

Each frame can be input as an image. Each image undergoes the below mentioned operations.

Finally, area of the white pixels is calculated and compared with the threshold value. If the area is greater than the threshold value, then High-beam headlight is converted to Low-beam; else High-beam is retained.

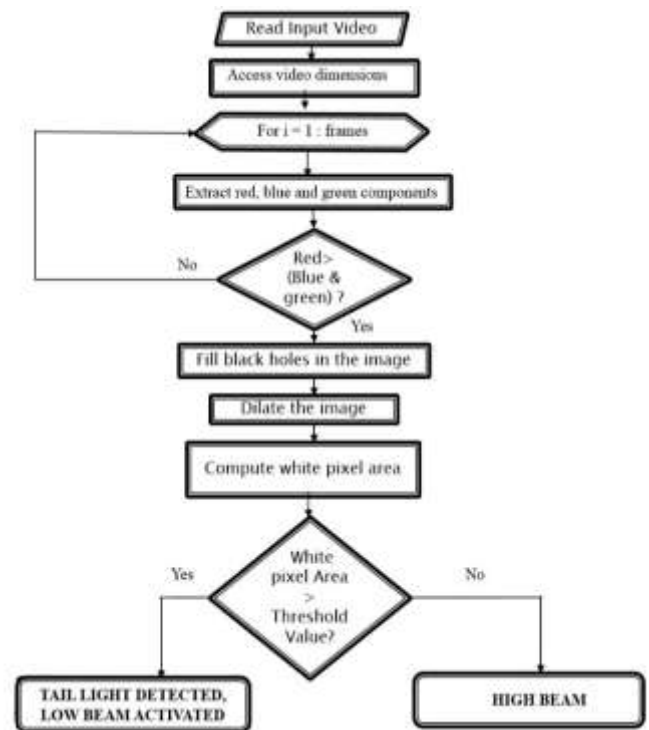


IV. FLOWCHART FOR TAIL LIGHT DETECTION

High-beam headlight from the ego-vehicle can cause temporary blindness to the driver of the preceding vehicle, since the glare can fall on the rear-view mirror of the preceding vehicle, thus affecting the driver’s eyes.

Hence, in order to detect the tail light of the preceding vehicle, red component of the image is extracted. Then the image undergoes the below mentioned operations. Finally, area of the white pixels is calculated and compared with the threshold value, then High-beam headlight is converted to Low-beam; else High-beam is retained.

MATLAB Tool can be used to simulate all the results on a computer. The simulation results performed using Matlab are shown in section V.



### V. SIMULATION RESULTS FOR HEADLIGHT DETECTION



Fig. 1: ORIGINAL IMAGE FOR HEAD LIGHT DETECTION



Fig. 2: IMAGE CONVERTED FROM RGB (COLOR) TO GRAY



Fig. 3: NOISE FILTERED IMAGE



Fig. 4: THRESHOLDED IMAGE

### VI. SIMULATION RESULTS FOR TAIL LIGHT DETECTION



Fig. 5: ORIGINAL IMAGE FOR TAIL LIGHT DETECTION



Fig. 6: RED COMPONENT EXTRACTED IMAGE



Fig. 7: HOLES FILLED BINARY IMAGE



Fig. 8: DILATED IMAGE

## VI. TIMING AND CONCLUSIONS

TABLE I: TIMING RESULTS

TYPE OF DETECTION	HEADLIGHT DETECTION OF AN IMAGE	TAIL LIGHT DETECTION OF AN IMAGE
PROCESSING TIME	0.5079s	0.2134s
MAXIMUM DISTANCE FOR DETECTION	250-300m	100-150m
MEMORY OCCUPIED	877B	897B

From the above table, it is seen that the processing time is reduced significantly when compared to other techniques such as using proximity sensors or photo-sensors. Also the proximity sensors have limited range of about 25m-40m. [4]

The memory required for the proposed method is also less since we are dealing with a single image at any instant of time.

A vehicle which is approaching at a speed of 150kmph towards us needs 9s to travel 375m. That means we have 9s to detect the vehicle's headlight and we are detecting it within 0.6s which is highly efficient. If we use Lane Detection Algorithm and Region of Interest false positive light blobs in the images can be eliminated and very high accuracy can be obtained. [5]

## VII. FUTURE WORK

This project can be merged with Lane Detection algorithm to make it more accurate. The method used here for detecting tail lights can also be used to prevent Forward Collision and issue a Forward Collision Warning (FCW) to the driver. [6]

Our project was implemented and simulated on MATLAB. This can be converted to C language and implemented on a digital signal processor for real time application of vehicle speed estimation and traffic law enforcement. [7]

We can also pair the headlight for four wheelers by using Pattern detection technique. By making use of the frames in which light blob is present we can even calculate the speed of the oncoming vehicle. [8]

## ACKNOWLEDGMENT

On the very outset of this paper, we would like to extend our sincere and heartfelt obligation towards all the personages who have helped us in this endeavor. Without their active guidance, help, co-operation and encouragement, we would not have made headway in this paper.

We are ineffably indebted to the Department of Electronics and Communication, K S Institute of Technology, Bangalore, for conscientious guidance and encouragement to accomplish this assignment.

We are extremely thankful and pay our gratitude to our faculty, Mr. Sangappa S. B., Associate Professor and PRO, K.S. Institute of Technology, for his valuable support and guidance in completion of this paper in its present form.

## REFERENCES

- [1] Automatic headlight dimmer, a prototype for vehicles; Muralikrishnan, B.E, Electrical and Electronics Engineering, Sri Venkateswara College of Engineering, Tamil Nadu, India. IJRET: International Journal of Research in Engineering and Technology. eISSN: 2319-1163 | pISSN: 2321-7308
- [2] "Digital Image Processing", by Rafael Gonzalez 2012.
- [3] MATLAB 2012b, by MathWorks, www.mathworks.com.
- [4] Night-time Vehicle Detection for Intelligent Headlight Control. Antonio L'opez, J'org Hilgenstock, Andreas Busse, Ram'on Baldrich, Felipe Lumbreras, and Joan Serrat, Computer Vision Centre and Computer Science Dept., Auton. Univ. of Barcelona, Volkswagen AG, Group Research, Carmeq GmbH, Business Team Surround Sensing, these authors are partially supported by Spanish MEC research projects Consolider Ingenio 2010: MIPRCV (CSD200700018) and TRA2007-62526/AUT. J. BlancTalon et al. (Eds.): ACIVS 2008, LNCS 5259, pp. 113–124, 2008. Springer-Verlag Berlin Heidelberg 2008
- [5] A Novel Traffic-Tracking System Using Morphological and Blob Analysis : Prabhakar Telagarapu, Department of ECE, GMR Institute of Technology, RAJAM- 532 127, AP, INDIA, email:prabhakar.t@gmrit.org
- [6] [https://www.trw.com/integrated\\_systems/driver\\_assist\\_systems/forward\\_collision\\_warning](https://www.trw.com/integrated_systems/driver_assist_systems/forward_collision_warning)
- [7] [https://www.researchgate.net/post/How\\_to\\_convert\\_MATLAB\\_code\\_to\\_C\\_program\\_in\\_Image\\_Processing](https://www.researchgate.net/post/How_to_convert_MATLAB_code_to_C_program_in_Image_Processing)
- [8] Enhancing Light Blob Detection for Intelligent Headlight Control Using Lane Detection: Sungmin Eum, Member, IEEE, and Ho Gi Jung, Senior Member, IEEE (IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 14, NO. 2, JUNE 2013).