Embedded Vision-based Nighttime Driver Assistance System


Journal of Convergence Information Technology, Volume 6, Number 2. February 2011

Abstract

This study presents an effective method for detecting vehicles in front of the camera-assisted car during nighttime driving and implements it on an embedded system. The proposed method detects vehicles based on detecting and locating vehicle headlights and taillights using techniques of image segmentation and pattern analysis. Firstly, to effectively extract bright objects of interest, a segmentation process based on automatic multilevel thresholding applied on the grabbed road-scene images. Then the extracted bright objects are processed by to identify the vehicles by locating and analyzing their vehicle light patterns and to estimate their distances to the camera-assisted car by a rule-based procedure. Finally, we also implement the above vision-based techniques on a real-time system mounted in the host car. The proposed vision-based techniques are integrated and implemented on an ARM-Linux embedded platform, as well as the peripheral devices, including image grabbing devices, voice reporting module, and other in-vehicle control devices, will be also integrated to accomplish an in-vehicle embedded vision-based nighttime driver assistance system.

Keywords: Embedded System, Vehicle Detection, Driver Assistance

1. Introduction

Detecting the road environment for driver assistance and autonomous vehicle guidance is an emerging research area. Accordingly, many studies have developed valuable techniques for recognizing interesting vehicles and obstacles from images of road environments outside the car [1][2], to facilitate applications on the camera-assisted system that assists drivers in understanding possible hazards on the road, and automatically controlling the apparatus of vehicles, such as headlights, windshield wipers, etc.

A vision-based vehicle and obstacle detection system is aiming at identifying of vehicles, obstacles, traffic signs and other patterns on the road from grabbed image sequences by means of image processing and pattern recognition techniques. By adopting different concepts and definitions of interesting objects on the road, different techniques are applied on the grabbed image sequences to detect them as vehicles or obstacles. For locating vehicles in an image sequence, the task can be carried out by searching for specific patterns on the images based on typical features of vehicles, such as shape, symmetrization, or their surrounding bounding boxes [6]-[8]. Until recently, most of these works focused on detecting vehicles under daytime road environments.

However, under bad-illuminated conditions in nighttime road environments, those obvious features of vehicles which are effective for detecting vehicles in daytime become invalid in nighttime road environments. At night, as well as under dark illuminated conditions in general, the only visual features of vehicles are their headlights and taillights. However, there are also many other illuminant sources coexisted with the vehicle lights in nighttime road environments, such as street lamps, traffic lights, and road reflector plates on ground. These non-vehicle illuminant sources cause many difficulties for detecting actual vehicles in nighttime road scenes.

In this study, we propose an embedded vision-based nighttime driver assistance system for vehicle detection and identify vehicles by locating and analyzing their headlights and taillights.
This proposed system comprises of the following processing stages. Firstly, a fast bright object segmentation process based on automatic multilevel histogram threshold is performed to extract pixels of bright objects from the grabbed nighttime road-scene images. The advantage of this automatic multilevel thresholding approach is robust and adaptable for dealing with various illuminated conditions at night. Then these bright components are grouped by a projection-based spatial clustering process to obtain potential pairing headlights of oncoming vehicles and taillights of preceding vehicles. Accordingly, a set of identification rules are applied on each group of bright objects to determine whether it represents an actual vehicle. The distance between each of the detected vehicles and the camera-assisted car is estimated and reported. We also implement the above vision-based techniques on an embedded real-time system mounted in the host car. The proposed embedded vision-based nighttime driver assistance system (EVNDAS) are integrated and implemented on an ARM-Linux embedded platform, as well as the peripheral devices, including image grabbing devices, mobile communication module, voice reporting module, and other in-vehicle control devices, will be also integrated to accomplish an in-vehicle embedded vision-based nighttime driver assistance system.

2. The Proposed Nighttime Vision System

In the proposed system, we adopt an effective nighttime vehicle detection method for identifying vehicles by locating and analyzing their headlights and taillights. This proposed method comprises of the following processing stages include of bright object segmentation, spatial clustering process, rule-based vehicle identification, vehicles distance estimation and traffic event warning and control signaling machinery. Figure 1 sketches the flow diagram of the proposed nighttime driver assistance system.

2.1. Bright Object Segmentation

The input image sequences grabbed from the vision system, these grabbed frames reflect nighttime road environments appeared in front of the car. Figure 2 shows a sample of nighttime road scene taken from the vision system. In this sample scene, there are two vehicles appeared on the road, where the left one is approaching in the opposite direction on the neighboring lane, and the right one is moving in the same direction with the camera-assisted car.

Hence, the first task is to extract these bright objects from the road scene image to facilitate further rule-based analysis. To save the computation cost on extracting bright objects, we firstly extracted the grayscale image, which is shown as Figure 3, i.e. the Y-channel, of the grabbed image by performing a RGB to Y transformation. For extracting these bright objects from a given transformed gray-intensity image, pixels of bright objects must be separated from other object pixels of different illuminations. Thus, an effective multilevel thresholding technique is needed for automatically determining the appropriate number of thresholds for segmenting bright object regions from the road-scene image. For this purpose, we have already proposed an automatic multilevel thresholding technique for image segmentation [9]. This technique can automatically decompose a grabbed road-scene image into a set of homogeneous thresholded images.

Figure 1. System block diagram
To preliminarily screen out non-vehicle illuminant objects such as street lamps and traffic lights located above the half of the vertical y-axis, i.e. the "horizon", and save the computation cost, the bright object extraction process will be only performed on the bright components located under the virtual horizon, as shown in Figure 4. Accordingly, as shown in Figure 5, after performing the bright object segmentation process, we can find that pixels of bright objects in Figure 2 are successfully separated into thresholded object planes under illumination conditions.

2.2. Spatial Clustering Process

To obtain vehicle-light-like components from the bright obtained object plane, a connected-component extraction process is then performed on the bright object plane to locate the connected-components [10] of the bright objects. We are interested in looking for the horizontal-aligned vehicle lights; hence a spatial clustering process is applied on the connected-components to cluster them into several meaningful groups. A resultant group comprises a set of connected-components, and it may consist of vehicle-lights, traffic lights, road signs, and some other illuminant objects which are commonly appeared in nighttime road environments. These groups are then processed by the vehicle light identification process to obtain the actual moving vehicles.

Firstly, the definitions used in the projection-based spatial clustering process are described as follows,

1). $C_i$ denotes one certain bright connected-component to be processed.

2). $CG_k$ denotes a group of bright components, $CG_k = \{C_i, i=0,1,2,...,p\}$, the total number of connected-components contained in $CG_k$ is denoted as $N_{CC}(CG_k)$.

3). The location of the bounding boxes of a certain component $C_i$ employed in the spatial clustering process are their top, bottom, left and right coordinates, and they are denoted as $t(C_i)$, $b(C_i)$, $l(C_i)$, and $r(C_i)$, respectively.

4). The width and height of a bright component $C_i$ are denoted as $W(C_i)$ and $H(C_i)$, respectively.

5). The horizontal distance $D_h$ and vertical distance $D_v$ between two bright components are defined as,

\[
D_h(C_i, C_j) = \text{max}[t(C_i), t(C_j)] - \text{min}[r(C_i), r(C_j)]
\]

\[
D_v(C_i, C_j) = \text{max}[l(C_i), l(C_j)] - \text{min}[b(C_i), b(C_j)]
\]

If the two bright components are overlapping in the horizontal or vertical direction, then the value of the $D_h(C_i, C_j)$ or $D_v(C_i, C_j)$ will be a negative value.
6). Hence the measure of overlapping between the vertical projections of the two bright components can be computed as,

\[ P_i(C_i, C_j) = -D_i(C_i, C_j) \sqrt{\min[H(C_i), H(C_j)]} \] (3)

To preliminarily screen out non-vehicle illuminant objects such as street lamps and traffic lights, we firstly filter out the bright components which are located above the one-third of the vertical y-axis, i.e. only the bright components located under the constraint line, the "virtual horizon" as shown in Figure 4, will be taken into account. This is because the vehicles which are located at the distant place on the road become very small light “points”, and will "converge" into a virtual horizon.

Secondly, to determine the moving directions of detected vehicles, it is necessary to distinguish the bright components into potential headlights and taillights for performing respective analysis. The distinguishable characteristics of taillights are red illuminated lights. However, when the preceding vehicles are near to the camera-assisted car, i.e. within 30 meters, their taillights cause “blooming effects” in CCD cameras, and are usually too bright to appear as white objects in the grabbed images. As a result, only the pixels that are located around the components of potential taillights have distinguishable red appearance. Hence the following red-light criterion is utilized to check if a bright component contains a potential taillight object, and this criterion is determined by,

\[ R_{\text{red}}(C_i) = T_{\text{red}} \] (4)

where \( T_{\text{red}} \) is a pre-determined threshold; \( R_{\text{red}}(C_i) \), \( G_{\text{red}}(C_i) \), and \( B_{\text{red}}(C_i) \) represent the average intensities of the R, G, and B color frames of the pixels which are located at the peripheral of a given bright component \( C_i \), respectively. Here the value of \( T_{\text{red}} \) is chosen as 10, to appropriately discriminate the potential taillights and other bright components. If a bright component \( C_i \) satisfies the red-light criterion, then \( C_i \) is tagged as a red-light component; otherwise, it is tagged as a non-red-light component.

Then the connected-components of bright objects are recursively merged and clustered into bright component groups \( CGs \) if they have the same light tags, are horizontally close to each other, vertically overlapped, and aligned. In other words, if a pair of neighboring bright components satisfies the following conditions, they are merged with each other and clustered as the same group \( CG \):

1). They have the same tags, i.e. both of them are red-light components, or both are non-red-light components.
2). They are horizontally close to each other, i.e.:

\[ D_i(C_i, C_j) < T_x \times \max[H(C_i), H(C_j)] \] (5)

3). They are highly overlapped in vertical projection profiles, i.e.:

\[ P_i(C_i, C_j) > T_y \] (6)

4). They have similar heights, i.e.:

\[ H(C_i)/H(C_j) > T_b \] (7)

where \( C_i \) is the one with the smaller height among the two bright components \( C_i \) and \( C_j \), while \( C_j \) is the larger one.
Figure 6. The spatial clustering process of the bright components of interest

Here $T_d$, $T_p$, and $T_h$ are the pre-determined thresholds to respect the pairing characteristics of vehicle lights, and the values of them are reasonably chosen as 3.0, 0.8, 0.7, respectively. Figure 6 illustrates the spatial clustering process of bright components. After performing the spatial clustering process, several groups of bright components are obtained, and they are called candidate vehicle light groups.

### 2.3. Rule-based Vehicle Identification

A rule-based identification process is then applied to determine whether candidate vehicle light groups (CGs) contain actual vehicle lights or other illuminated objects based on the statistical features of their contained bright components. If a certain candidate group $CG_k$ contains a set of actual vehicle lights to represent a vehicle, then the following heuristic determination rules must be satisfied,

1). The enclosing bounding box of the candidate vehicle light group must form a horizontal rectangular shape, i.e. the size-ratio feature of the enclosing bounding box of $CG_k$ must satisfy the following condition,

$$\frac{W(CG_k)}{H(CG_k)} \geq \tau_s$$  \hspace{1cm} (8)

where the threshold $\tau_s$ on the size-ratio condition is selected as 2.0 for suitably reflecting rectangular-shaped appearance of pairing vehicle lights.

2). The number of its contained bright components should also be within a reasonable range, because the vehicle lights are mostly appeared in symmetrical pairs, and some types of compound vehicular light set may comprise of at most four lights, so that the light number condition is defined as,

$$\tau_{n1} \leq N_{nc}(CG_k) \leq \tau_{n2}$$  \hspace{1cm} (9)

where the values of $\tau_{n1}$ and $\tau_{n2}$ are chosen as 2 and 4, respectively, to appropriately reflect the pairing characteristic of vehicle lights.

3). Moreover, its contained bright components should be well-aligned, and thus the number of these components should be in reasonable proportion to the size of the size-ratio feature of its enclosing bounding box, thus, the following alignment condition must be satisfied,

$$\tau_{a1}\left(\frac{W(CG_k)}{H(CG_k)}\right) \leq N_{ac}(CG_k) \leq \tau_{a2}\left(\frac{W(CG_k)}{H(CG_k)}\right)$$  \hspace{1cm} (10)

where the thresholds $\tau_{a1}$ and $\tau_{a2}$ are determined as 0.4 and 2.0, respectively, according to our analysis of typical visual characteristics of most vehicles during nighttime driving. These discriminating rules are obtained by analyzing many experimental results of videos on real nighttime road environments having vehicle lights appeared in different shapes, sizes, directions and distances.
2.4. Vehicle Distance Estimation

To estimate the distance between the camera-assisted car and detected vehicles, we apply the perspective range estimation model of the CCD camera as introduced in [11]. The origin of the virtual vehicle coordinate system is placed at the central point of the camera lens. The $X$ and $Y$-coordinate axes of the virtual vehicle coordinate are parallel to the $x$ and $y$-coordinates of the grabbed images, and the $Z$-axis is placed along the optical axis and perpendicular to the plane formed by the $X$ and $Y$ axes. A vehicle on the road at a distance $Z$ in front of the camera-assisted car will project to the image at a vertical coordinate $y$. Thus a perspective range estimation model as presented in [11] can be utilized for estimating the $Z$-distance in meters between the camera-assisted car and one detected vehicle by using the equation,

$$Z = k \cdot \left( \frac{f \cdot H}{y} \right)$$

(11)

where $k$ is a given factor for converting from pixels to millimeters for the CCD camera which is mounted on the car at the height $H$, and $f$ is focal length in meters.

2.5. Traffic Event Warning and Control Signaling Machinery

The traffic event warning and control signaling machinery is an automatic control process in the proposed system. This automatic control process is comprised of the vehicle headlight control process and the vehicle speed control process. When any oncoming vehicles being detected, the headlight control process switch the headlights to low beams, and then turn back to high beams after the detected vehicles leave the detection zone. The warning voice will be activated for noticing drivers to throttling slow down when the distance to the detected preceding vehicles being too close to collide. Figure 7 and Figure 8 illustrate the flowcharts of the headlight control process and the vehicle speed control process, respectively.

![Figure 7. Headlight control process work flow](image)

![Figure 8. Vehicle speed warning process work flow](image)
3. Implementations of EVNDAS

In this section, the implementation of EVNDAS will be presented in detail. Subsection A depicts the software implementation of the proposed system, including the integration of the ARM-based embedded platform with embedded Linux OS and Qt graphical user interface for user friendly control machinery. Then, we also apply a component-based framework to implement the software framework of the proposed system. Subsection B introduces the hardware architecture of the proposed system.

3.1. Software Development

In this section, a component-based framework is adopted for developing the software framework of the proposed system. The software framework of the proposed system includes the following modules: the bright object segmentation (BOS), the spatial clustering process (SCP), the rule-based vehicle identification (RVI), the vehicle distance estimation (VDE), and the traffic event warning and control signaling machinery (TEWCSM). The advantages of using component-based are easily to update new function modules and extend the system in any subsequent developments.

| BOS Subsystem | Input: get grayscale image
|               | Process: extracting bright objects
|               | Output: bright object plane
| SCP Subsystem | Input: bright object plane
|               | Process: connect-component groups
|               | Output: bright component sets
| RVI Subsystem | Input: bright component sets
|               | Process: rule-based identification process
|               | Output: vehicles locations
| VDE Subsystem | Input: vehicles locations
|               | Process: estimate vehicle distances process
|               | Output: estimated vehicle distances
| TEWCSM Subsystem | Input: estimated vehicle distance, vehicle lights type
|                | Process: headlight control process, vehicle speed control process
|                | Output: control signal

Figure 9. BOS module

Figure 10. SCP module

Figure 11. RVI module

Figure 12. VDE module

Figure 13. TEWCSM module
Based on the above-mentioned software modules, the proposed system includes the following five processing stages. As shown in Figure 9, in the first stage, the BOS module obtain the grayscale image sequences grabbed from the vision system, and then apply the segmentation process for extracting bright objects of interest, and resultantly export the obtained bright object plane to the following stage. In the second stage shown in Figure 10, the SCP module will receive the bright object plane from the BOS module, and apply a connect-component extraction process for grouping the bright objects into meaningful sets, then send out the bright component sets as a result. The third stage in Figure 11 is the RVI module, which is get the bright component sets from SCP module, and perform the rule-based identification process for detecting and verifying the oncoming and preceding vehicles, then the detected vehicles and their locations are sent out to the VDE module. As the fourth stage in Figure 12, the VDE module process the locations of the detected oncoming and preceding vehicles, and estimate the distances between them and the host car, and then the estimated vehicle distances will be provided for the following stage. The final stage in Figure 13 is the TEWCSM module, which perform and activate the headlight control process and the vehicle speed control process according to the information of detected vehicles and their related distances.

Figure 14. Proposed system software architecture

Figure 14 shows proposed system software framework, which comprises of three major layers, the first layer is the embedded Linux kernel ported on the development platform, the second layer consists of the Qt GUI SDK library ported on the embedded Linux kernel, and on the top layer, the processing modules of the proposed EVNDAS system are implemented based on Qt library and the Linux kernel. To control the hardware peripherals, the proposed system activate the corresponding control messages to the device drivers on the embedded Linux kernel, and then the Linux kernel will accordingly signaling the related hardware peripherals, such as GPIO signals, voice speakers, network devices. Thus, drivers can operate the proposed system by using the touch panel and hardware peripherals.

3.2. Hardware Architecture

The hardware architecture of the proposed system is shown in Figure 15 The proposed system comprise a 32-bit ARM processor with 520 MHz speed, as well as a 32MB Flash memory for storing the embedded Linux kernel and program files, and a 64MB SDRAM memory for executing the system programs. And the system bus is a breach between other device, capture device driver is use to connect vision system, LCD controller is use to control touch panel it can catch input signal when user use it.

Figure 15. Proposed system hardware architecture
4. Experimental Results

The proposed system was implemented on an ARM-based embedded system development platform which is set up on our experimental camera-assisted car. The frame rate of the vision system is 10 frames per second and the size of each frame of grabbed image sequences is 320 pixels by 240 pixels per frame. The proposed system has been tested on several videos of real nighttime road-scenes under various conditions. Figure 16 illustrates our embedded system development platform.

Figure 17 shows the main screen of proposed system, which consists of three buttons, including the functions of the system configuration, system starting and system stopping. The system configuration is for setting the system value such as the voice volume, control signaling, while system starting and system stopping buttons are applied for starting and stopping the proposed system, respectively.

Figure 18 shows the results of detecting the preceding vehicles, and the locations and estimated distances of the detected vehicles are illustrated on the LCD. Here the locations of the detected preceding vehicles are drawn by green rectangles. When detecting any oncoming vehicles, the traffic event warning and control signaling machinery process will transfer the head light status into low beams. Figure 19 is the situation of both oncoming and preceding vehicles being detected, and the detected oncoming vehicle is sketched by red rectangle. After the oncoming vehicle passing the host car, the traffic event warning and control signaling machinery process will change the headlight status into high beams. Figure 20 depicts a sample that the system determines the detected preceding vehicle being too close to the host car. In this situation, the traffic event warning and control signaling machinery process will display the warning message on the LCD and activate the warning message voice for notifying the driver to slow down to avoid collision dangers.
5. Acknowledgements

This paper was supported by the National Science Council of R.O.C. under Contract No. NSC-99-2221-E-027-100, and NSC-99-2622-E-027-019-CC3.

6. References