SLAM Map Application for Tracking Lights on Car Dashboards

Francesco Carotenuto[1](✉), Ugo Erra[2], and Vittorio Scarano[1]

[1] Dipartimento di Informatica, Università di Salerno, Fisciano, Italy
francesco.carotenuto@gmail.com, vitsca@unisa.it
[2] Dipartimento di Matematica, Informatica ed Economia,
Università della Basilicata, Potenza, Italy
ugo.erra@unibas.it

Abstract. Recent studies conducted by some insurance companies highlighted that the most part of drivers do not know the meaning of the dashboard lights. This leads drivers to be dangerous for others and themselves. Hence, the need to provide drivers with tools that support them to always be aware of the state of their cars. This paper proposes a system for mobile devices that uses augmented reality to be able to give information on a particular dashboard lights. The system is implemented by combining the use of Simultaneous Localization and Mapping (SLAM) maps with central moment computation widely used in computer vision. Preliminary research results show that the proposed system achieves its goal enables a a real-time visual feedback.

1 Introduction

Recent studies carried out of some insurance agencies [6] [8] have highlighted that the most part of drivers do not know the car components, and, in particular, the meaning of the dashboard lights. This lack of knowledge of drivers is dangerous because unknown car malfunctioning can lead to incidents and, as a consequence, to be injured, or, in the worst cases, to dead. For this reason, car manufacturers provide drivers with a printed maintenance guide. However, it is tedious and time consuming to leaf through a long printed car maintenance guide and, as a consequence, several drivers give up keeping cars in an unsafe state. To avoid this dangerous behavior, some applications mobile augmented reality[4] have been implemented to replace printed guides and to allow drivers to obtain information in a simpler way. Usually in these applications the user points with the camera of the mobile device a component such as a car wheel and some information about it are shown, such as instructions to change a wheel.

However, a feature which lacks to these applications is the tracking of the lights on the dashboard. In order to take full advantage of this feature, it should be always performed even when the car stops in an area not served by any wireless connection. Moreover, systems providing this functionality should have two important requirements: working in any type of lighting condition and recognizing the lights even if partially occluded by steering wheel as illustrated in Figure 1.

L.T. De Paolis and A. Mongelli (Eds.): AVR 2014, LNCS 8853, pp. 413–420, 2014.
DOI: 10.1007/978-3-319-13969-2_31

(a) Nothing (b) Partial (c) Full

Fig. 1. Examples of possible occlusions of the dashboard caused by the position of the steering wheel

The aim of this paper is to enrich the state of the art with a system based on augmented reality capable of giving information about dashboard lights in a simple and fast way and, at the same, satisfying the above mentioned requirements.

The idea is to associate with each light a relative position with respect to a marker (such as a portion of the speedometer). On the basis of these positions are derived from the frame of the segments that triggered the tracking. Within these segments, of fixed size, there should be a particular indicator. To detect the presence of this indicator, we proceed to search for a centroid located in a position close to the center of the segment.

In particular, the proposed system combines the use of Simultaneous Localization and Mapping (SLAM) maps, implemented through Metaio framework, with central moment computation obtained through OpenCV library.

The paper is structured as follows. Section 2 presents the background knowledge including description of SLAM framework implemented by Metaio and study of the moments and, in particular, of the central moment, performed by OpenCV. Section 3 presents the general architecture of the system and the steps that implement the lights tracking. Before concluding in Section 5, system tests are presented in Section 4.

2 Background Information

This section presents the two principal methodologies on which proposed system is based: SLAM and moment computation. SLAM [7] allows to realize a mapping of a 3D environment and can support a fast and stable tracking [3]. The produced maps are typically used by robots to be able to move in an environment or simply to give them an idea of their position within the environment. There are multiple ways to be able to trace the maps SLAM, some of these methods make use of certain devices such as GPS [1], laser and sonar[2]. These last two ones are usually installed on self-propelled devices.

However, it is also possible to construct a SLAM map by using a mobile device equipped with a camera [11]. In this way, it is possible to recover a particular scene, identify a set of points that are placed on a 3D coordinate system. In this work, we exploits Metaio SDK. It provides a set of APIs for the development of augmented reality applications. In particular, it provides functionalities for displaying multimedia content, but especially features that allow to make the tracking using maps SLAM. For the creation of the maps, SLAM Metaio SDK [5] uses an application called Toolbox available for both Android and iOS.

The second principal methodology is the moment computation. The moment is a particular average intensity of pixels in an image. It is defined as:

$$M_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy \tag{1}$$

In particular, this work considers the study of the central moments, as they form a particular feature of an object represented in an image independently of its location within the image itself. In the case of a binary image the coordinates of the central moment is obtained with:

$$C = (\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}) \tag{2}$$

where M_{00}, M_{10}, M_{01} represent, respectively, the area of the object and the center of mass of the object on x and y axis. In this work, we OpenCV [9] as a library that provides functionalities to support applications that make use of Computer Vision. In particular, OpenCV is used to calculate the moments within the image that are the basis for the calculation of the central moment.

3 System Overview

This section presents the proposed system aimed at giving drivers information about dashboard lights. In order archive its aim, the system uses the slam maps, together with some pre-processed information. In particular, the system is composed of two modules: the first devoted to the acquisition of such information, called *Training*, and the second one devoted to the tracking, named *Detection*. The training module works to acquire the SLAM maps that make up the dashboard and some information that uniquely characterize the lights inside the dashboard. Instead, the detection module uses the SLAM maps and information produced by the training component in order to track dashboard lights.

The two components are used by different actors: training is used by car maker whereas detection is used by the driver. Hereafter, more details about system components are given.

3.1 Training

The training module of the system is illustrated in in Figure 2. From the diagram it is possible to note that before the training, there is a pre-training which

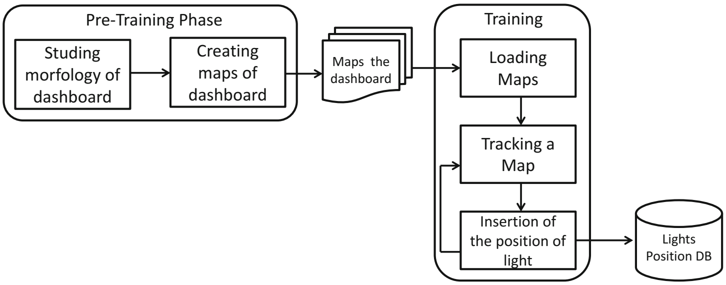

Fig. 2. Steps of training module

consists of two steps. The first step is devoted to the study of the morphology and arrangement of the lights on the dashboard. This step is useful to understand how many maps are needed to keep track of all lights. Using the information, we proceed to the creation of the SLAM maps using the application provided by Toolbox as shown in Figure 3.

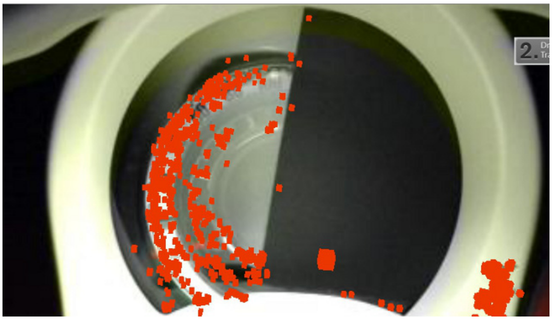

Fig. 3. SLAM map of a dashboard region

Subsequently, the generated maps are loaded into the training module as input. Using these maps, the user performs the tracking by pointing the camera of the device in the dashboard areas of interest. During the tracking an augmented object (AO) appears on the device screen. This augmented object can serve as a 3D placeholder to fix the position of the lights during the tracking as illustrated in Figure 4. Once the augmented object is placed on the light the user store its position in a database. Associated to the position of the light position further information can be inserted such as light name and more important a description of the meaning of the light. This process continues until all the 3D positions of all the dashboard lights are stored.

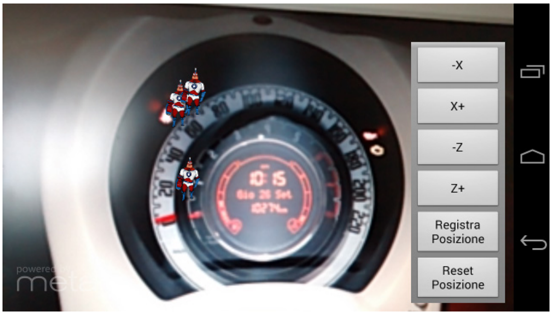

Fig. 4. Inserting of the positions of the light using training module

3.2 Detection

As illustrated in Figure 5, the detection module needs certain information such as those relating to the SLAM maps and the database with the positions of lights created in the previous phase.

In addition to this information, further data are required as the database of augmented objects(AO DB) that should be displayed in the case of tracking of one or more lights on. The AO associated with a single light is a tooltip which shows an icon and a brief description Figure 6.

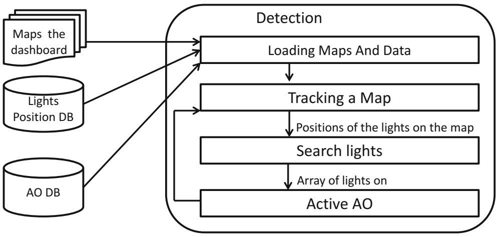

Fig. 5. Steps detection module

The first step performed by detection module is to load all the positions of the lights in the form of fictitious AOs. These objects are never displayed during the tracking of the portion of the dashboard where the user is pointing the camera for the light, but they only serve to obtain the position of the light in the scene. In detail, the detection module performs the tracking of a portion of the dashboard in order to obtain all the stored 3D position with the associated SLAM map. These positions, are mapped from 3D points in the world to 2D points using the camera device coordinates system.

Since the resolution and the aspect ratio of the display differs from those of the camera it is necessary to make the rescaling of the coordinate obtained in the previous step. After calculating all the coordinates of the lights in the area, we proceed with the capturing of the frame of the camera that triggered the tracking. Once obtained coordinates and the frame, it is possible to search the lights on in accordance with the steps described in Figure 7. The first step is to obtain

Fig. 6. AO visible on light on

segments of the image of a fixed size $n \times n$ whose the point at bottom-left is one of the obtained coordinates. Each one of these segments should contain a light. For each of these segments, we calculate the center of mass of the objects present in the segment. Ideally, the center of mass of the light should be in the coordinated $C = (n/2, n/2)$. Since the front shot can translate some centimeters, then it is necessary to define a range within which the system can obtain that there is a light on. This range is defined in an interval $\lfloor n/2 \rfloor - \delta \leq C_x \leq \lfloor n/2 \rfloor + \delta$ and $\lfloor n/2 \rfloor - \delta \leq C_y \leq \lfloor n/2 \rfloor + \delta$.

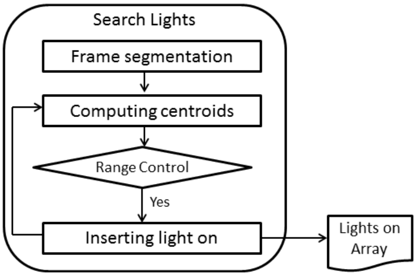

Fig. 7. Phases of search

The values of δ and n must be defined in order to avoid false positives and false negatives. A false positive occurs when the system detects the presence of a ligth on even if it is off, conversely as regards the false negative.

High values of δ and n can lead to increase the number of false positives, whereas, small values increase the number of false negatives. In this work, the values of δ and n are defined empirically, taking as reference the resolution of the camera 320×240.

Whenever it finds a light on in the dashboard is inserted a new element in an array given in input to the last step of the Detection component. At the end, this last step views AO associated with the name of the light in the array.

4 Experiments and Results

This section presents preliminary experiments performed to test the proposed system. The carried out experiments take into account different occlusions and light conditions.

The lighting condition is considered to be primarily related to the maximum illumination of the instrument panel lighting, obtained using the mobile devices flash while shooting the scene. To better evaluate the robustness of the approach made and taking account of the absence of the flash from some mobile devices, such as tablet it was thought to perform tests also considering the condition of natural light. For natural light means the only light coming from the windows and windshield of the vehicle.

In particular, we consider four combinations of occlusions and lighting conditions:

- Scenario 1 (S1): absence of occlusion with full light.
- Scenario 2 (S2): absence of occlusion with natural light.
- Scenario 3 (S3): presence of occlusion with full light.
- Scenario 4 (S4): presence of occlusion with natural light

Moreover, since the system should be able to detect one or more indicators site in the same are, we consider two cases: situations with one light and dual lights adjacent. By summarizing, combining the defined modes with the aforementioned combinations of lighting and occlusion, we perform 8 tests. Each performed test is repeated five times and with different mobile devices.

The hardware configuration was based on a Samsung Galaxy Tab 2 GT-P7100, dual core 1GHz Tegra 2, 1GB of RAM, camera 3Mpixel with flash and a Motorola Moto G, quad core 1.2Ghz Snapdragon 200, 1GB of RAM, camera 5Mpixel with flash.

The evaluation of the proposed system is carried out by using the following scale:

- *yes*: tracking is performed within 30 seconds of the first shot dashboard.
- *partial*: tracking is performed within 45 seconds.
- *none*: tracking fails.

Table 1. Summary table of the tests performed

Configuration	**S1**	**S2**	**S3**	**S4**
Single Light	yes	yes	yes	yes
Multiple Light	yes	partial	yes	partial

The Table 1 shows the result of the performed experiments. As shown, the proposed system yields good performance obtaining the highest score in the 75% of the performed tests. We point out that these tests are preliminary and we are going to perform further experiments considering several users and a full day usage.

5 Conclusions and Future Works

This work is aimed at creating a system for mobile devices that helps the driver to better understand the instrumentation and controls as an easily and more appealing alternative to the car guide. As shown by the preliminary experiments, the proposed system yields good performance.

However, the proposed system tends to generate false positives due to the presence of unexpected occlusion objects within the camera view frustum. As for instance, an object with a color similar to light on can be detected as an objects with a contour and consequently possess a center of mass.

In the future, to reduce false positives we think to add a pre-processing step on the frame before breaking it up into segments. For example, this step could consist in the application of techniques for color segmentation [10] useful to remove all objects that do not match the color of the lights. In this way, we think to minimize the number of calculated centroids, and as consequence to increase the accuracy of the proposed system.

References

1. Carlson, J.D.: Mapping Large, Urban Environments with GPS-Aided SLAM. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (August 2010)
2. Choi, J.W., Ahn, S., Chung, W.K.: Robust sonar feature detection for the slam of mobile robot. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 3415–3420 (August 2005)
3. Davison, A.J., Mayol, W.W., Murray, D.W.: Real-time localization and mapping with wearable active vision. In: Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 18–27 (October 2003)
4. Fiat. Ciao fiat. http://www.fiat.it/ciao-fiat
5. Metaio GmB. Metaio sdk. http://www.metaio.com
6. Horton, P.: Dashboard warning lights confuse drivers (July 2013). http://www.lv.com/about-us/press/article/dashboard-warning-lights-confuse-drivers
7. Leonard, J., Durrant-Whyte, H., Cox, I.J.: Dynamic map building for autonomous mobile robot. In: Proceedings of the IEEE International Workshop on Intelligent Robots and Systems, IROS 1990, Towards a New Frontier of Applications, vol.1, pp. 89–96 (July 1990)
8. Megna, M.: Warning! drivers' knowledge of dashboard lights is running on empty (November 2013). http://www.insurance.com/auto-insurance/safety/what-does-dashboard-light-mean.html
9. OpenCV. Opencv. http://www.opencv.org
10. Skarbek, W., Koschan, A.: Colour image segmentation - a survey (1994)
11. Ventura, J., Arth, C., Reitmayr, G., Schmalstieg, D.: Global localization from monocular slam on a mobile phone. In: IEEE Virtual Reality (March 2014)