Steering wheel's angle tracking from camera-car

Rita Cucchiara **Andrea Prati** **Francesca Vigetti**

Dipartimento di Ingegneria dell'Informazione
Università di Modena e Reggio Emilia
Modena, Italy

cucchiara.rita@unimore.it prati.andrea@unimore.it francesca.vigetti@libero.it

Abstract

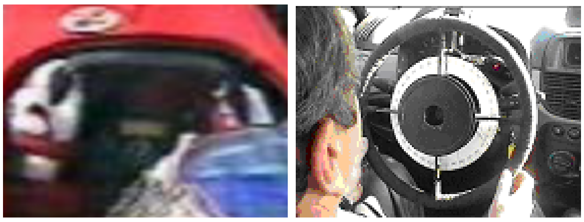
This paper proposes a general-purpose method to track the steering wheel's absolute angle by using a single camera vision system mounted inside the car. The approach is based on the modeling of the motion of the steering wheel, as it appears perspectively distorted by the point of view of the un-calibrated camera. We modified the Lucas-Kanade method for an approximatively rotational motion model in order to provide the detection and tracking of significant features on the wheel. The experimental results are compared with ground-truthed data obtained with different types of sensors.

1 Introduction

Real-time analysis of videos acquired from a camera mounted on a moving vehicle (namely camera-car) can be very attractive due to the large amount of visual information that can be extracted both inside the vehicle (to assess the driver conditions and control the environment to prevent from dangerous situations) and outside the vehicle (for automatic guidance purposes, as vehicle control and obstacle avoidance). In the first context, new research activities are devoted to the assessment of the driver's posture for smart air bag deployment, or to the acquisition of driving information. Another example is the use of cameras to detect potentially dangerous situations in which the driver is distracted (e.g., because he responds to a cell phone while is driving).

In this framework an interesting problem is the detection of the steering-wheel rotation angle. The possibility to compute this angle in real-time can be exploited to provide a feedback to the driver in terms of virtual (or augmented) reality, or to support an automatic guidance system, or to analyze the style of the driving by observing how the steering wheel's angle changes along time.

Theoretically, the same information could be obtained by other types of sensors, such as electro-mechanical sensors, potentiometers, and so on, applied to the steering wheel. The advantages of a vision-based system are basically three: first, the other types of sensors require a more invasive installation, and, moreover, cameras can be easily moved from one vehicle to another; second, electronic sensors can not work on pre-registered data, i.e. they can only obtain results on the moment, in real-time; third, the amount and the semantics of the information provided by a camera are more than any other type of "blind" sensor. As an example, refer to Fig. 1(b) where a special steering wheel equipped with potentiometers is used to acquire ground-truthed data.



(a) (b)

Figure 1: Examples of applications: (a) an application for automated telemetry from camera-car videos (with the courtesy of Ferrari Spa, Italy) and (b) a car testing facility (with the courtesy of Centro Ricerche Fiat - Orbassano, Italy).

According with these considerations, we propose a general-purpose approach to detect the rotation angle of the steering wheel in a reliable manner. The method can be used to track the trajectory of the car by tracking the rotation angle frame by frame.

research works address the problem of detecting the motion of rigid objects: often the motion is assumed as a translational model (for instance the Lucas-Kanade method and derived approaches [3]); more generally, an affine model in the 3D space is assumed: this approach is very general and very complex and therefore highly time consuming. Thus, is often used as a first qualitative step to detect motion in videos [2]. In this work, instead, we start from a constrained motion model since we aim to detect and track the rotation of a rigid object around a fixed point. A little effort in the research community has been done to study this type of problems. However, as for similar problems, we can subdivide the approaches in two main classes: based on the *object-model* and on the *motion-model*. The first class assumes that the model of the object to be tracked is known. Consequently, a shape recognition algorithm can be used to localize the object (and its orientation) in each frame. These methods are usually based on template or shape detection approaches such as the Hough transform (HT), as the basic HT for parametric curves or the generalized Hough Transform for general templates.

The second class is instead based on the model of the motion of the object or part of it.

The first approach is applicable only if the object is a-priori known. Thus, for instance, it must be tuned and changed for each possible steering wheel model.

The second approach, instead, requires a reliable techniques to detect motion or optical flow for each image pixel, or calls for a first robust stage for extracting significant points or features from each frame, in order to compute their motion.

2 The proposed approach

Our approach belongs to the class of motion-based techniques. Consequently, it can only compute the displacement of the points in a frame w.r.t. the points of the previous frame: thus, it suffers of the drawback of computing the absolute angle as a sum of relative angles. In particular, our method computes the absolute rotation angle of a steering-wheel (with respect to an initial zero-degree position) as an accumulation of inter-frame relative rotations. We can detect the passage for the zero-degree position in order to re-align the measured angle with the real (absolute) angle, thus avoiding error propagation. This is, indeed, the only process in which part of the model of the steering-wheel is required and will not be described due to lack of space.

Obviously, we can start from the assumption that the points of the steering wheel move with the same circular motion. However, due to the pinhole camera model [1], this is true in the image plane only if the focal plane of the camera is parallel to the steering wheel plane. In the other cases, because of the perspective, the points can be approximated as moving on to an ellipse. Since our scope was to provide a method that can work in as many situations as possible, we have used an elliptical model as reference.

Once the model of the motion is defined, the first phase is to detect "significant" features on the wheel. A feature must be considered as "significant" if it is easy to track between two consecutive frames. To extract features we used the well-known algorithm of Tomasi-Kanade [4, 5]. Eventually, once the features have been detected in two consecutive frames, we must match the same feature into the two frames. This phase is called *feature tracking* and will be detailed in the next subsection.

2.1 Feature Tracking

Let us consider a feature as a window of 3x3 points for the sake of simplicity. In the case of elliptical motion of the features, the actual motion of the features is not only translational, but roto-translational.

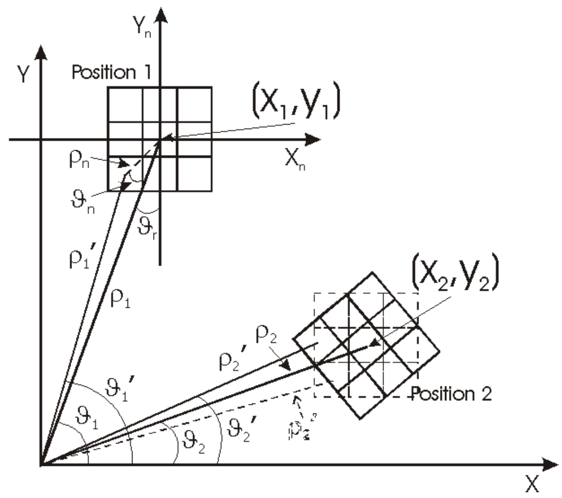


Figure 2: The roto-translation of a 3x3 feature

As a consequence, a feature does not only change its position inside the next frame, but it also rotates. This involves a non-correspondence between intensity of pixels with the same relative position, as shown in Figure 2, where the dotted line version is the case of pure translational motion.

To track a feature we have implemented two methods. The first is directly derived from the Lucas-Kanade tracking method [3, 5] is based on the hypothesis of only translational motion. The second method considers a roto-translational motion by modifying the Lucas-Kanade method to take also the rotation into account.

Considering a coordinate system $< O, X_n, Y_n >$ with the origin in the center of the 3x3 window in a reference frame, the polar coordinates of each point of the window in the reference frame is known and it is (ρ_n, ϑ_n). We can easily compute the polar coordinates (ρ'_1, ϑ'_1) in the main reference system:

$$\rho'_1 = \sqrt{\rho_n^2 + \rho_1^2 - 2\rho_n\rho_1 \cos\vartheta_n} \qquad (1)$$

$$\vartheta_1' = \vartheta_1 + \arccos\left(\frac{\rho_1 - \rho_n \cos\vartheta_n}{\sqrt{\rho_n^2 + \rho_1^2 - 2\rho_n\rho_1\cos\vartheta_n}}\right) \quad (2)$$

Once these coordinates are known, it is straightforward to obtain the coordinate (ρ_2', ϑ_2') of another point of the window in the new position by adding a displacement vector $\Delta\rho, \Delta\vartheta)$. Unfortunately, this is not true in the case of roto-translational motion and this introduces an additional error in the positioning of corresponding points of the window. As a consequence, the matching will be less precise.

Let us call I and J two consecutive frames. Thus, the correct equation in the case of roto-translational motion should be:

$$\begin{aligned} I(\rho_1' \cos\vartheta_1', \rho_1' \sin\vartheta_1') \quad = \quad & J((\rho_1' + \Delta\rho)\cos(\vartheta_1' + \Delta\vartheta), \\ & (\rho_1' + \Delta\rho)\sin(\vartheta_1' + \Delta\vartheta))(3) \end{aligned}$$

These considerations refer to the general case with elliptical motion and 3x3 window. Obviously, this method will work also with larger windows and circular motion.

It can be interesting to underline that, due to the rotation of the window, the new coordinates (ρ_2', ϑ_2') can correspond to not integer coordinates, thus an interpolation method is mandatory to determine the intensity of the real point. This interpolation will introduce further approximations to the matching. Please note that this interpolation is not necessary if we suppose the motion as pure translational. Moreover, the computational load is heavily affected by the more intensive computation and by the interpolation required by the second method. As a consequence, and taking into account that the improvement in the efficacy introduced by the second method is not so relevant, we have decided to suppose initially the model as translational, by approximating the search using the first method. Thus, a further refinement phase is necessary that verifies if the motion can be acceptable with an elliptic trajectory.

2.2 Feature Selection and Angle Computation

In the previous sub-section we described the methods used to track features between two consecutive frames. Though this method is accurate, the set of matched features is typically full of outliers. The reason is twofold: first, there is always a lot of noise in camera-car videos (in particular if acquired with radio technologies as in the Formula 1 races); second, the hands of the driver on the steering wheel can have their own motion that is, most of the times, opposite to (or in general different from) the steering wheel's motion. Consequently, we have to perform two further tasks to remove outliers from this set. The first task is to impose additional constraints to the matching rules in order to prevent motion of the features clearly not on an ellipse. By simple rules we prevent, for example, that a feature moves radially from one frame to the next.

The second task is devoted to the computation of the relative angle from the set of (reliable) features' motions. The rationale is that in an ideal case all the features of the steering wheel should have the same motion. Unfortunately, the ideal case is very rare, and noise and outliers create different motions that must be filtered.

To do this we have experimented and compared many methods, and found that the best one is the following. Let us call A the set of relative angles computed at this step and α the absolute angle computed at the previous step. From the value of α we can decide if the vehicle is in a rectilinear part or is approaching a curve. We use this information to select the statistical function to be used to compute the new relative angle value:

- if State = Rectilinear → Relative Angle = Median (A)
- if State = Curve → Relative Angle = Mean (A')

where A' is the set with the maximum cardinality between A+ (set of positive angles) and A- (set of negative angles).

The absolute angle is computed by adding the relative angle to α.

3 Experimental Results

To evaluate the proposed methods we collected data from an vehicle instrumented with sensors able to get synchronized telemetric data to compare our results with.

As a benchmark, we used a sequences acquired by normal television and reports part of a Formula 1 race shot by a camera-car mounted on the left top of the driver's cockpit (Fig. 1(a)). We tested our motion-based method on this sequence and the comparison is reported in Fig. 3. This sequence was obtained with the courtesy of Ferrari Gestione Sportiva Spa.

4 Conclusions

In this paper we have described our method for computing the absolute angle of a steering wheel by using

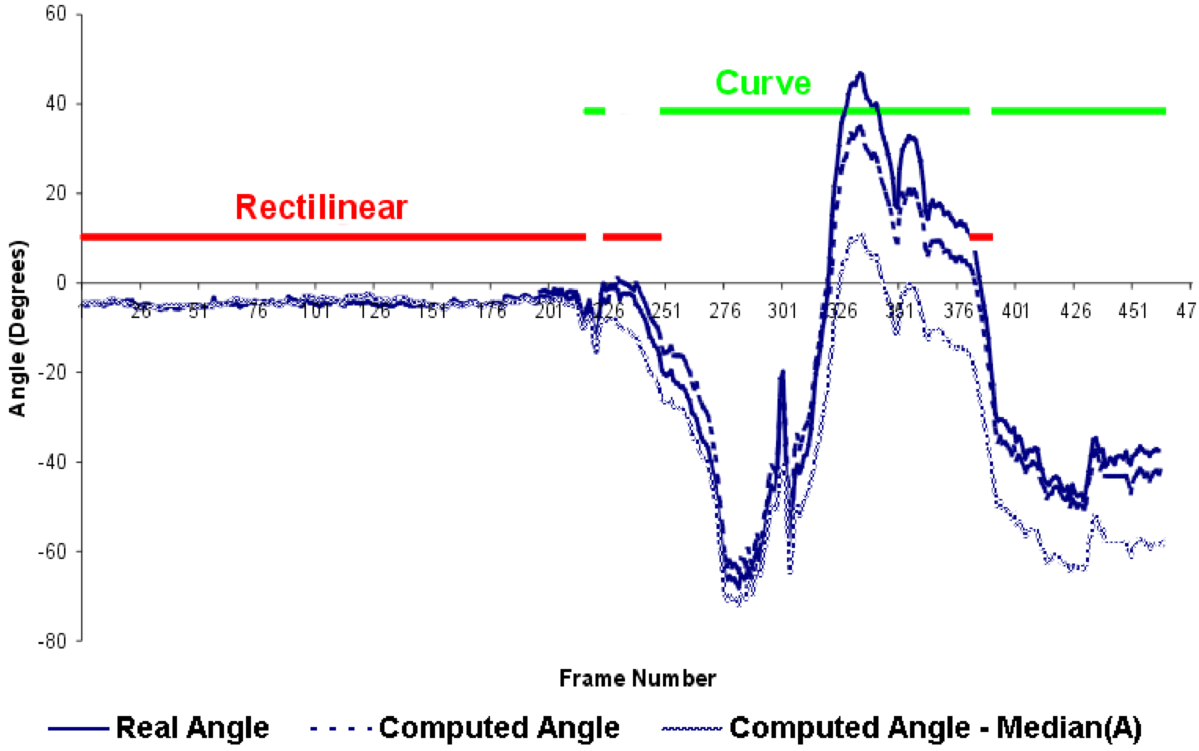


Figure 3: Graph showing the comparison between computed absolute angle and angle obtained by telemetric sensors

only a single camera. The proposed approach is based on the computation of significant features by means of Tomasi-Kanade algorithm and on the subsequent tracking of them by using a modified version of the Lucas-Kanade tracking algorithm. This modification takes into account the model of the motion of the points belonging to the steering wheel.

A feature selection method combined with a set of rules to compute the relative angle for each frame are then used to remove noise and outliers. We presented experimental results on a significant sequence to testify the correctness of the approach.

Much work can be still done to make to method reliable on different situations, but the preliminary results are very satisfactory and promising.

Acknowledgements

The authors wish to thank Piergiorgio Grossi and Marco Fainello from Ferrari Gestione Sportiva Spa for their valuable help in getting videos. The authors want also to acknowledge the people from CRF for the courtesy in giving us pictures of their equipments.

References

[1] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. *Pattern Analysis & Applications*, 2:312–320, 1999.

[2] M. Gelgon and P. Bouthemy. A region-level motion-based graph representation and labeling for tracking a spatial image partition. *Pattern Recognition*, 33:725–740, 2000.

[3] B.D. Lucas and T. Kanade. An iterative image restoration technique with an application to stereo vision. In *Proceedings of Int'l Joint Conference on Artificial Intelligence*, pages 593–600, 1981.

[4] J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Int'l Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[5] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method - 3. detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.