CAMERA-CAR VIDEO ANALYSIS FOR STEERING WHEEL'S TRACKING

R. Cucchiara[1], *C. Grana*[1], *A. Prati*[1], *F. Vigetti*[1], *M. Piccardi*[2]

[1]Dipartimento di Ingegneria dell'Informazione
Università di Modena e Reggio Emilia, Italy
[2]Department of Computer Systems, Faculty of IT
University of Technology, Sydney, Australia

ABSTRACT

Monitoring and controlling the driver's guidance by analyzing the rotation impressed to the steering-wheel can be a very important task in order to improve safety. 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 absolute angle is computed by means of the accumulation of inter-frame relative rotations and the error propagation is prevented with an alignment process. 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

Amongst vehicular technologies, those dedicated to safety play a more and more important role nowadays. In particular, monitoring and controlling the driver's guidance in real-time can be dramatically important. To this aim, real-time computation of the rotation angle of a car's steering-wheel must be provided in order to achieve relevant information about the driver's guidance.

Moreover, 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.

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.

In the rest of the paper, we make the following assumptions:

1. instrumentation of the wheel must be avoided, and thus no artificial reference points have been settled;

2. the environment is not structured and heavily cluttered; in particular, the driver's head often occludes the steering-wheel and shadows and light beams can suddenly change the luminance (Figg. 1(a) and 1(b))

3. the image quality is affected by burst noise, carried by vibrations and jumps.

The paper is structured as follows: section 2 reports the related works present in the literature, though they are few. Section 3 describes the solution proposed by detailing the wheel's motion model and the method used to compute the angle. Section 4 shows the experimental results obtained by comparing them with the telemetric data; eventually, conclusions are reported in section 5.

2. RELATED WORKS

One possible approach to measure the rotation angle in the case the object shape is a-priori known, is to generate every possible rotation and evaluate its matching with the current frame. However, in this application, the object shape cannot be considered known a priori, due to the large amount of different shapes to be taken into consideration. In addition, substantial parts of the target object might be occluded in the current frame and the matching consequently fail. A solution is offered by a feature tracking approach: what humans actually do to perceive rotation motion of a steering-wheel is basically to follow some recognizable points of the wheel or follow the motion of the driver's hands, which often move rigidly with the wheel itself. In the same way, a computer vision approach can perform automated feature detection and tracking to measure the rotation angle.

Many 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 [1]); 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 at detecting and tracking 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.

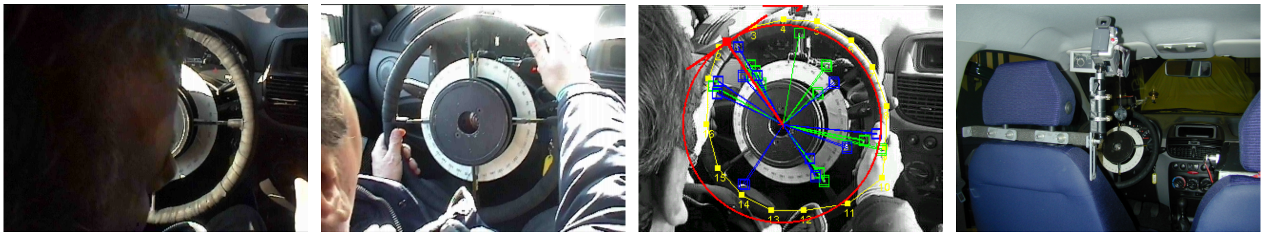
The first approach is applicable only if the object is a-priori known. Thus, for instance, it must be tuned and changed for each possible steering wheel model. The second approach, instead, requires a reliable techniques to detect motion or optical flow for each image pixel, or calls for a first robust stage for extracting significant points or features from each frame, in order to compute their motion.

3. 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 realign 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 be described briefly in the following.

We assume a circular motion around the wheel's axis. As a consequence, 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 [3], 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.

The goal of the application can be defined as the computation, frame by frame, of the absolute wheel angle ϑ^t with respect to a reference position (angle $\vartheta^t = 0$). We assume that the motion can be classified for each frame as belonging to two possible motion classes, according to the vehicle's trajectory. To this aim, we define the *motion type* m at time t as a variable in the domain m^t=$\{$`<rect>`,`<curve>`$\}$ that characterizes the steering-wheel motion typical of rectilinear driving or during a curve, respectively. In the first case, we can assume a very limited wheel's angle around the zero position, whereas in a curve it is reasonable to measure higher angle values. This assumption is confirmed by ground-truth measurements. Furthermore, we assume that the condition of angle $\vartheta^t = 0$ can be reliably detected (called *alignment* in the following) by pattern matching. To do this we rely on part of the steering wheel's model: we adopted a simplified model just enough to distinguish in the search area between the steering wheel and other distractors. Fig. 2 reports two examples taken from Formula 1 cars: in the first case (Ferrari racing team, Fig. 2(a)) we selected the upper rectilinear part of the steering wheel as model and detected it with the Correlated Hough Transform [4] that identifies parallel, near rectilinear lines; the second steering wheel (former McLaren racing team wheel, Fig. 2(b)) shows two rays that can be used (with HT) as a model.



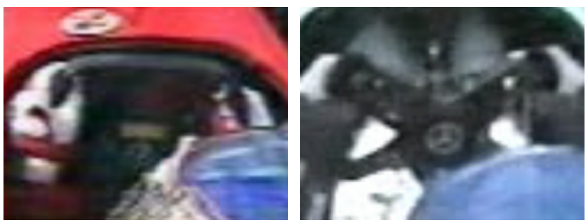
(a) Driver's head occlusion example

(b) Light beam that confuses the feature detection algorithm

(c) Example of feature detection and tracking

(d) View of the camera setup (with the courtesy of Centro Ricerche Fiat, Italy)

Fig. 1. The system proposed. (a) and (b) present two noisy situations, (c) the setup of the camera and (d) the feature detection and tracking of the system.



(a) Ferrari's steering wheel

(b) Former McLaren's steering wheel

Fig. 2. Two examples of steering wheel from Formula 1 car. (a) Ferrari car and (b) former McLaren car.

The wheel's motion can be only inferred by computing frame by frame the angular displacement α^t by means of syntactic feature detection and tracking. Due to the hypothesis of cluttered and hostile environment, two conditions of *invalid* or *valid* detection could occur. A condition is considered invalid when the signal-to-noise ratio or the lighting conditions prevent for an acceptable feature detection. Next subsection will describe the statistics used to reliably compute the angular displacement α^t in the case of valid condition. The use of statistics for improving feature tracking in cluttered situation is reported also in [3].

To compute the absolute angle we use an adaptive model described by the following equation:

$$\vartheta^t = \vartheta^{t-1} + k_1\alpha^t + k_2\alpha^{t-1} \tag{1}$$

where k_1, k_2 are two binary parameters. The condition of (k_1, k_2)=(1,0) represents the case of valid detection. On the other hand, the case of invalid detection requires further investigation. In fact, when there are no information available on the current frame, there are two possible decisions: we can assume that the absolute angle is not changed (i.e. the wheel has not been moved) or we can assume that, during this frame, the driver has continued to move the wheel in the same direction and with the same angle (i.e. we assume a constant derivative of the absolute angle, that is a constant relative angle α^t). The first condition (k_1, k_2)=(0,0) is true in the case of stationary motion condition (i.e., $m^t = m^{t-\Delta t}$), and the second (k_1, k_2)=(0,1) in a transient situation, in which $m^t \neq m^{t-\Delta t}$.

3.1. Angular displacement computation

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" or "good" if it is easy to track between two consecutive frames. To extract features we used the well-known algorithm of Tomasi-Kanade [5][6].

The definition of good feature is introduced in [5] and it is based on the selection of corners as "good" features. Fig. 1(c) reports an example of feature extraction (the bounding boxes). Therefore, given two consecutive frames I^t and I^{t-1} and the corresponding set of detected features $\Im^t$ and $\Im^{t-1}$, where $\Im^t = \{f_1^t, \ldots f_{n_t}^t\}$, we use a *feature tracking* method to match the features of $\Im^t$ with the features of $\Im^{t-1}$.

3.1.1. *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. 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 3, where the dotted line version is the case of pure translational motion.

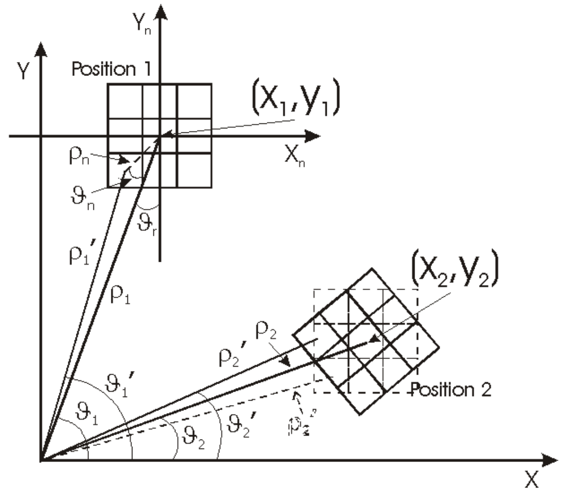


Fig. 3. The roto-translation of a 3x3 feature.

To track a feature we have implemented two methods. The first is directly derived from the Lucas-Kanade tracking method [1][6] 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:

$$\rho'_1 = \sqrt{\rho_n^2 + \rho_1^2 - 2\rho_n\rho_1 \cos\vartheta_n} \tag{2}$$

$$\vartheta'_1 = \vartheta_1 + \arccos\left(\frac{\rho_1 - \rho_n \cos\vartheta_n}{\sqrt{\rho_n^2 + \rho_1^2 - 2\rho_n\rho_1\cos\vartheta_n}}\right) \tag{3}$$

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:

$$\begin{aligned} I(\rho'_1\cos\vartheta'_1, \rho'_1\sin\vartheta'_1) \quad = \quad & J((\rho'_1 + \Delta\rho)\cos(\vartheta'_1 + \Delta\vartheta), \\ & (\rho'_1 + \Delta\rho)\sin(\vartheta'_1 + \Delta\vartheta)) \end{aligned} \tag{4}$$

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.

3.1.2. Feature Selection

The features' motion in the space is computed in an angular way with respect to a fixed center of the wheel (manually set). In this way, a feature match $\{f_i^t, f_j^{t-1}\}$ identifies a relative angle $\alpha^t(i,j)$. For the sake of brevity, in the following we will omit the subscripts. Therefore, for each frame in a valid condition we obtain a set $A^t = \{\alpha_1^t, ..., \alpha_k^t\}$ of measured angle displacements between pairs of features, where $k < min(n_t, n_{t-1})$.

Though this method is accurate, the set A^t of measured ancle displacements is typically full of outliers. The main reasons are three. 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, some features extracted and tracked could be background points (not belonging to the steering wheel) that satisfy the condition of "good" feature. Unfortunately, the feature pairs tracked will result in near-to-zero angle displacements that can affect the final absolute angle computation, and that can be confused with a condition of still wheel. This consideration hints that in the set A^t a systematic error due to near-to-zero displacements must be taken into account. Finally, 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.

In accordance with the above considerations, we can subdivide the set A^t into two subsets $A^t = A^t_+ \cup A^t_-$, where A^t_+ and A^t_- contain only strictly positive or negative displacements[1], respectively. Fig. 1(c) reports in green the features belonging to A^t_- and in blue the features belonging

[1]Positive angles are associated to left curves, whereas negative angles to right curves

$$\alpha^t = \begin{cases} \underset{i=1,\ldots,|A^t|}{arg\,min} \sum_{j=1}^{|A^t|} (d(\alpha_i^t, \alpha_j^t)),\ \alpha_i^t,\ \alpha_j^t \in A^t & if\ m^t = \texttt{<rect>} \\ \frac{1}{|\overline{A^t}|} \sum_{k=1}^{|\overline{A^t}|} \alpha_k^t,\ \alpha_k^t \in \overline{A^t} & if\ m^t = \texttt{<curve>} \end{cases} \quad (5)$$

$$m^t = \begin{cases} \texttt{<rect>} & if\ (m^{t-1} = \texttt{<rect>} \wedge |\vartheta^t| < T_\vartheta) \vee (m^{t-1} = \texttt{<curve>} \wedge |\vartheta^t| < T_\vartheta \wedge \Delta\vartheta^t < T_{\Delta\vartheta}) \\ \texttt{<curve>} & otherwise \end{cases} \quad (6)$$

to A_+^t. Note that we discard near-to-zero values in order to prevent the inclusion of the above-mentioned outliers due to still objects.

If we define $\overline{A^t}$ as the largest subset between A_+^t and A_-^t, the angle displacement is computed as defined in equation 5, where $d(.)$ is the metric used to compute the distance between angles. With the definition of equation 5, when the angles are near-to-zero, the outliers' distribution is supposed to be uniform around zero and the median function is a reliable statistic. Instead, in the curve situation, we discard as outliers all the displacements with the opposite sign of the majority of the measures. In other words, if the current state is "rectilinear" the median of all the computed relative angles is used, otherwise the mean of the more numerous set between positive and negative displacements is selected as relative angle for the current frame.

3.2. Motion type computation

Different statistics are used in dependence of the motion type, as reported in equation 5. In our case the motion type at time t is approximated by using the knowledge of the absolute angle at time t and of the motion type at time $t-1$, as stated in equation 6, where T_ϑ is a threshold on the absolute angle that fixes the maximum beyond which we consider a curve, $\Delta\vartheta^t$ is the derivative of the absolute angle computed at a distance in time of Δt (i.e. $\Delta\vartheta^t = \vartheta^t - \vartheta^{t-\Delta t}$), and $T_{\Delta\vartheta}$ is a threshold on the maximum derivative of the absolute angle.

3.3. Valid/Invalid condition assertion

Equation 1 decides whether to compute the absolute angle as $\vartheta^t = \vartheta^{t-1} + \alpha^t$ in the case of valid condition (where α^t can be obtained with equation 5) or to compute it as $\vartheta^t = \vartheta^{t-1}$ in the case of invalid condition and motion type unchanged with respect to Δt frames ago, or as $\vartheta^t = \vartheta^{t-1} + \alpha^{t-1}$ in the case of invalid condition and motion type changed.

In the system's diagram reported in Fig. 4 it is possible to note that there are some invalid conditions. A first possible condition is the presence of strong noise due to the vibrations and jumps. This type of noise is usually horizontally distributed in the image. For this reason, what we do to detect this class of noise is to compute the vertical gradient of the previous and current image. If the absolute derivative of the average gray level of the resulting images is greater than a threshold **T**, a sudden change in horizontal edges is present and, consequently, there is a high probability of interference. Another problem arises when the vehicle enters in a shadowed area (for example, due to trees at the roadside). The large variation in the brightness of the scene may result in the incorrect detection and tracking of the features.

4. EXPERIMENTAL RESULTS

To evaluate the proposed methods we collected data from a vehicle instrumented with sensors able to get synchronized telemetric data to compare our results with. In Fig. 1(d) an example of setup of a vehicle is reported (with the courtesy of Centro Ricerche Fiat, Italy).

In this paper we report the results achieve on 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 (as the one reported in Fig. 2(a)). We tested our motion-based method on this sequence and the comparison is reported in Fig. 5. This sequence was obtained with the courtesy of Ferrari Gestione Sportiva Spa.

The graph in Fig. 5 shows the absolute wheel angle ϑ^t during time, comparing the results achieved with the method proposed in this paper and with an electro-mechanical encoder (ground-truth). The continuous line (*Real angle*) represents the ground-truth and the dashed line represents the computed angle (*Computed angle*). In addition, horizontal lines are used to show where the motion model is classified as rectilinear (*Rectilinear*) and in curve (*Curve*). Results achieved with the proposed method seem really interesting since the computed curve is always very close to the ground-truth. The average error in this example is about 3%. In particular, when the driving is rectilinear, the two lines are almost identical. This result proves that computing the absolute angle by accumulating relative angles of tracked features does not lead to cumulative errors in time; in addition, frequent re-alignments of the computed angle

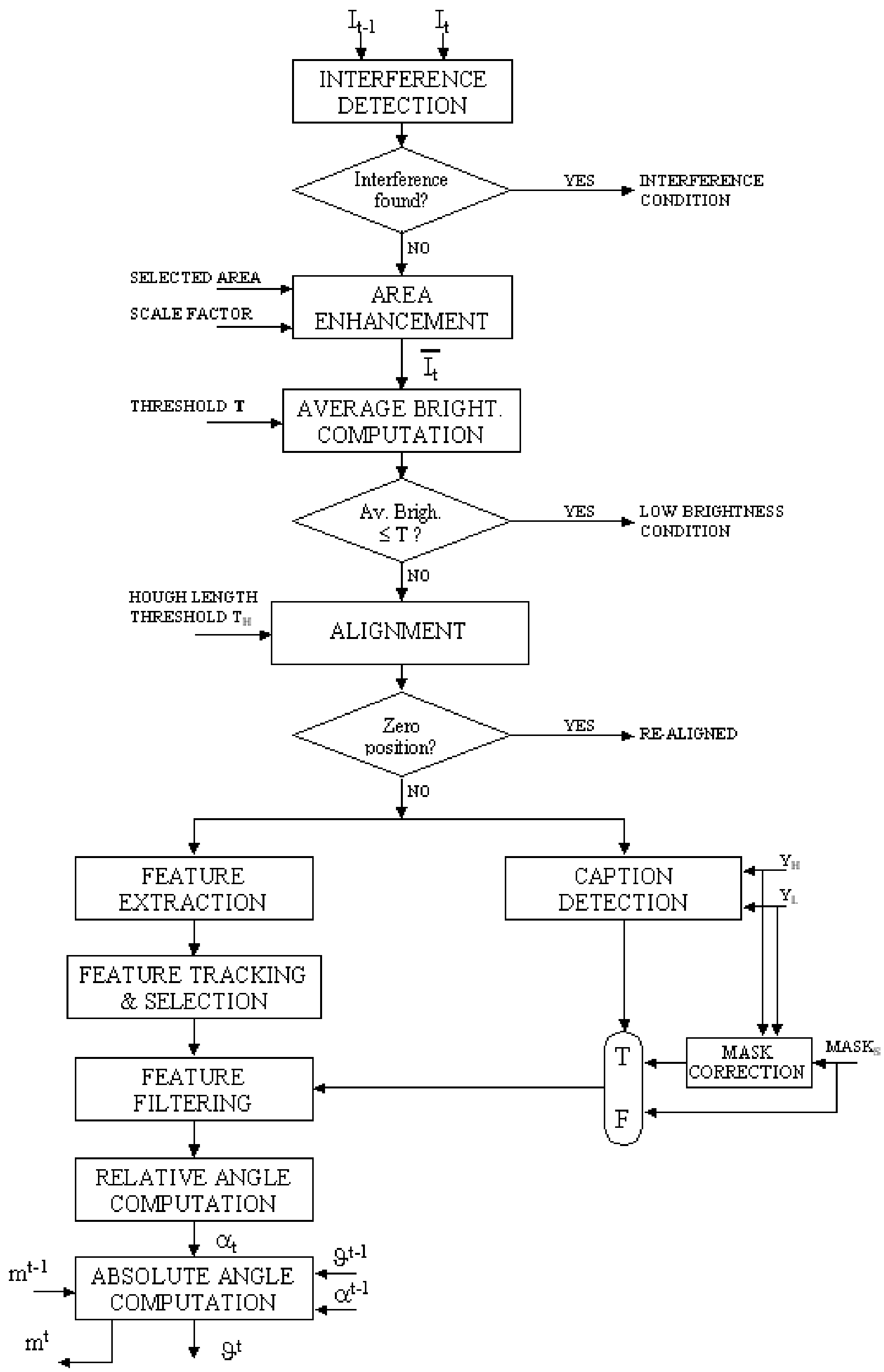


Fig. 4. System's flow diagram.

with the zero-degree position help to increase precision. During a curve, the computed and real angle are also very similar, but with some underestimation of the angle in the range of positive values. This is mainly due to the fact that a few static features visible in the scene tend to slightly lower the value of the computed angle; these errors accumulate in time, but, since the `<curve>` motion model generally holds only for short time intervals, the absolute error is limited.

The graph in Fig. 5 shows also the results for the computed angle achieved by computing the median angle over all the tracked features, without discriminating between the

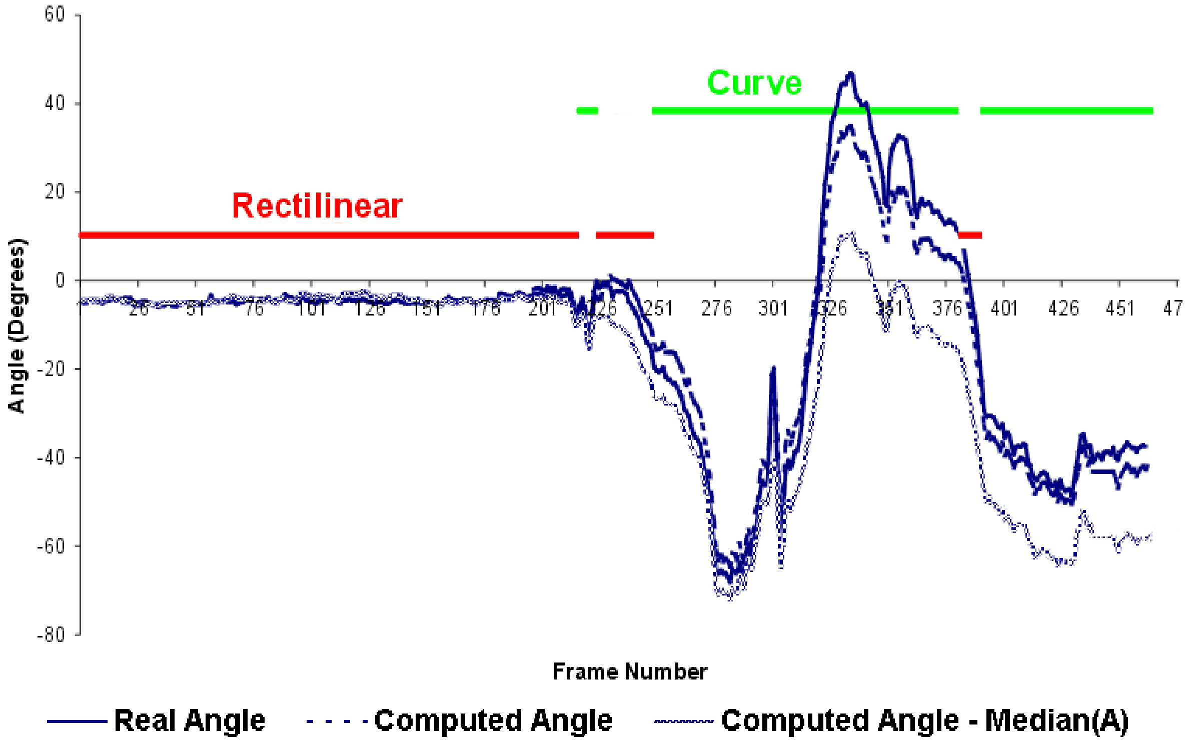


Fig. 5. Graph showing the comparison between the ground-truth and the results obtained by our approach

two different motion type models. Results are significantly worse than those achieved with the proposed method; in particular, during the curve phase the computed angle results in an under-estimation of the real angle of about 30 degrees on average. These larger errors are mainly due to the greater amount of outliers affecting the statistic computation, and prove the usefulness of the classification of the motion into motion models.

5. CONCLUSIONS

In this paper we have described our method for computing the absolute angle of a steering wheel by using only a single camera, and in presence of occlusions and cluttered situations. 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.

The computation both of the angle displacement and the absolute angle depends on the motion type and on the validity of the situation. 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. This approach has been tested in real and extreme conditions and compared with ground-truth obtained by telemetric data. Results are promising and prove that the approach based on motion model is effective.

Acknoledgments

Our research is supported by the Ferrari SpA. We wish to thank our collaborators from Ferrari for their support and interactions. We also wish to thank people from Centro Ricerche Fiat (CRF) - Orbassano for the help in collecting data and for the useful suggestions on how to handle the telemetric data.

6. REFERENCES

[1] 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*, 1981, pp. 593–600.

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

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

[4] R. Cucchiara, P. Mello, M. Piccardi, and F. Riguzzi, "An application of machine learning and statistics to defect detection," *Intelligent Data Analysis*, vol. 5, no. 2, pp. 151–164, 2001.

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

[6] C. Tomasi and T. Kanade, "Shape and motion from image streams: a factorization method - 3. detection and tracking of point features," Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, Apr. 1991.