Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 1998-V02

Tracking features with camera maneuvering for vision-based navigation[1]

Guerrero J.J., Sagüés C.

If you want to cite this report, please use the following reference instead:
Tracking features with camera maneuvering for vision-based navigation, Guerrero J.J., Sagüés C., *Journal of Robotic Systems*, Vol.15(4), pages 191-206, 1998.

[1]This work was supported by project TAP97-0992

Tracking features with camera maneuvering for vision-based navigation

J.J. Guerrero and C. Sagüés

Abstract

In this paper we present a method to obtain corresponding contour features in a sequence of images. Our final goal is to compute camera motion and structure, for navigation based on vision. We make a feature tracking in the image considering not only the contour geometric information, but also a brightness-based description. A kinematic model of brightness attributes has been proposed in order to obtain a brightness evolution constraint more reliable than the classical brightness constancy model. Camera maneuvers, like sudden orientation changes, are detected, estimated and used to obtain a better correspondence of features. Experimental results, using real images, are shown. Our main contributions are the proposed evolution model of brightness attributes and the detection of camera maneuvers.

Keywords

Mobile vision, straight edges, brightness parameters, tracking to match, camera maneuvering

I. Introduction

In vision, image contours correspond to local discontinuities of brightness. In many applications related to object recognition and motion analysis, these contours are used as key features. Image contours usually correspond to object boundaries. In industrial or urban environments, most of the contours are straight lines. Straight lines have a simple mathematical representation and involve more information than other features like points. Besides, lines are easier and more accurate to extract in a noisy image than points, and also easier to match than the latter [1]. In semi-structured environments, straight lines often appear and sometimes their tips can be identified.

Feature extraction is often followed by correspondence search to carry out higher level processes (object recognition, structure perception, navigation or motion analysis). In many works it is assumed that correspondence between features extracted from one image and those extracted from the next one, is available. However establishing and maintaining the correspondence is not an easy job [2].

In this paper a complete method to extract and match characteristic features is presented. We work with straight contours in the image, from which we extract not only the geometric representation of the straight line, but also a description associated with the observed intensity in its support region. The use of brightness makes the identification of the contour easier, also establishing its relevance. The contour extractor, explained in §II, is based in [3]. In §III a method to obtain identified points along the image contour is presented. This allows to get lines with identified points, which combine the advantages of both features in high level processes such as structure and motion computation.

In §IV we present a method to match features when structure and motion are unknown. It is based on the tracking in the image of both, geometric and brightness information. A kinematic model of the brightness parameters has been proposed and experimentally tested. As the camera may move unpredictably, a method to compute the camera maneuvers is proposed (§V), in order to improve the matching. This camera maneuver estimator is based on robust features and works because brightness attributes are used. In §VI some experiments are shown, emphasizing the advantages of using brightness attributes and their selection capacity. Finally §VII is devoted to presenting the conclusions of our work.

This work was partially supported by spanish project TAP-94-0390 of the Comisión Interministerial de Ciencia y Tecnología (CICYT).
The authors are with the Dpto. Informática e Ingeniería de Sistemas, Centro Politécnico Superior, University of Zaragoza, C) María de Luna 3, E-50015 ZARAGOZA, SPAIN, Phone 34-976-761940, Fax 34-976-762111, email: jguerrer@posta.unizar.es, csagues@posta.unizar.es

II. Extraction of straight edges

Extraction of edge lines from intensity images is a basic and powerful process to select the information contained in the image. Two main kinds of methods for straight segments detection can be used: Those based on the grouping of edge elements and those based on some brightness model.

In the first type of methods, edge elements are initially extracted. They are usually extracted by detecting the maximum of the gradient in the direction of the gradient, or by detecting zero crossing of the image convolution with the Laplacian [4], [5], [6]. Once edges are extracted, they must be grouped into line features. One of the most known method to join points into lines is the Hough transform. Other methods are based on graph theory, relaxation algorithms or fusing and eliminating points to create chains [7], [8].

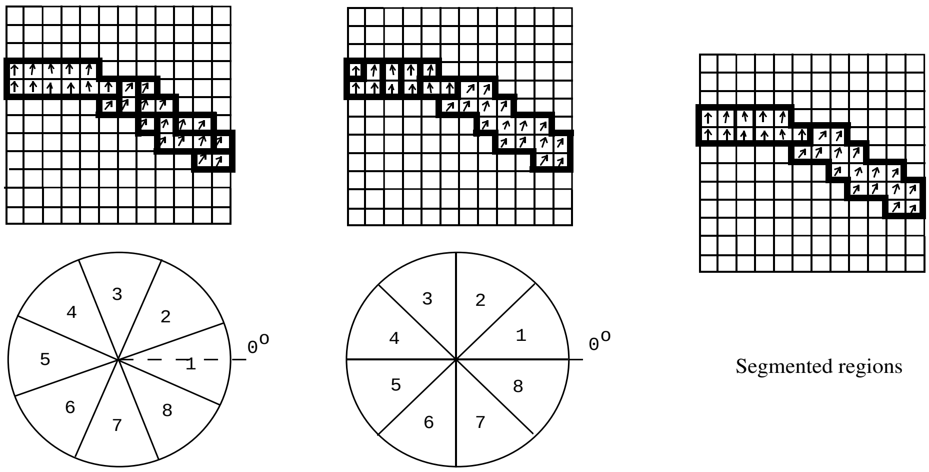

Fig. 1. Image segmentation into LSR by using fixed and overlapped partitions.

Straight contours can also be extracted considering a brightness model of the contour after a segmentation of the image into regions. The work of [3] is perhaps the most important to extract straight edges with this kind of methods. The key characteristic of this method is the global organization of brightness information into line support regions, before making additional considerations on the relevance of the contour. Two main advantages may be cited:

- It allows to obtain contours when the contrast is not high.
- It allows to extract not only geometric information but also information related with contour brightness.

Three steps can be considered in this method.

A. Segmentation

The first step in the procedure is the extraction of spatial gradients to segmentate the image. Pixels are grouped into regions of similar direction of brightness gradient, with gradient magnitude larger than a threshold. Segmentation is globally made using fixed partitions of gradient orientation. Two overlapping sets of partitions, with a post selecting process are used in order to avoid the problems related with the arbitrary boundary of fixed partitions (Fig. 1). From both segmentations, the support region giving a longer interpretation is selected in subsequent processes.

Thus, we have line-support regions (LSR) containing all information available of the straight contour.

B. Locating the line

To obtain the contour in the image, a planar brightness surface is fitted to the LSR by a least-squares approach, predicting the brightness (E) as a function of image coordinates. In this fitting, a weighting norm $N_w(x, y)$ proportional to gradient magnitude is considered, so that larger changes in brightness have a greater influence on the adjustment. The measurement minimized along the LSR in function of the parameters of the brightness surface (A_e, B_e, C_e) is expressed as:

$$\sum_{x,y}^{LSR} [A_e\, x + B_e\, y + C_e - E(x,y)]^2\, N_w(x,y)$$

The straight line is obtained, with subpixel accuracy, as intersection of this brightness plane and the horizontal plane of mean brightness E_m in the LSR (weighted with gradient magnitude).

$$E_m \;=\frac{\sum_{x,y}^{LSR} E(x,y)\; N_w(x,y)}{\sum_{x,y}^{LSR} N_w(x,y)}$$

Line orientation is given in range 2π, in such a way that its dark side remains on the right. This is quite useful to obtain better matches. The segment tips can be obtained as intersection of the line with the LSR boundary.

C. Brightness parameters

The LSR gives the opportunity to extract not only geometric information but also some parameters related to the observed intensity. These attributes are used to classify the edges and to identify them in successive images.

The extracted attributes are:

- Average gray level (agl)
- Contrast (c)
- Width (w)
- Steepness (s)
- Deviation from straightness (r)

The contrast can be defined as the change in the brightness through the line. It can be computed as the difference between maximum and minimum intensity in the LSR [3]. This measurement does not turn out to be robust. We propose to characterize the contrast as the standard deviation of the brightness σ_E in the LSR [9]. This is a global characterization which can be obtained simultaneously to the line extraction.

The width (w) is obtained by dividing the area of the LSR by the length of the segment. The steepness (s) is obtained by dividing the contrast by the width, $s = \frac{c}{w}$. In [3] these parameters are obtained in a different way, but we have observed by experiments how the contrast and the steepness obtained as we propose, have a more stable behavior.

As this method allows to extract lines with low contrast or length, it turns out quite interesting to filter the lines according to the obtained parameters. So, long lines or lines with a given orientation may be easily selected.

III. Extracting points on lines

Straight lines are considered as the basic features in perception based vision. In many works infinite lines have been used to compute motion [1], [10], [11]. In these works the tips of the lines are not needed to proceed.

On the other side, there are many works which explicitly consider points to compute motion. Corresponding points allow to solve motion and structure problem more easily than using lines, but extracting, matching points is normally more difficult.

We use points, but associated to lines. In this way, an agreement between the problems to extract or match points and the limitations of infinite lines to compute structure and motion is established [12].

Fig. 2. Image of the laboratory with the straight contours extracted (filtered in gradient and length).

Characteristic points are usually attached to some contour. So, focusing point search in the extracted contours turns out easier. In a first approach, tips obtained by the straight contours extractor could be used as identified points. However, they are not good enough because edge detectors do not work properly to obtain points [13]. Besides that, there are tips of contours that correspond with well-defined points, but in other cases tips stump unclearly.

As explained in [14], there are two kinds of methods to obtain points. In the first class, points with maximum curvature on an edge chain, are searched. In the second case, points are searched on the intensity image by using heuristic techniques such as the Moravec operator [7] or by measuring brightness variations [15]. Normally, these have higher computational cost [16], but higher precision.

We propose to search points along the line, avoiding isolated noise points and reducing the computational cost attached to the method. We consider as characteristic points those whose gradient multiplied by the curvature in the contour is a maximum [15]. So, deriving gradient orientation (E_x, E_y), along the normal direction to the gradient $n_\perp$, we have:

$$c = \frac{d\left(\arctan \frac{E_y}{E_x}\right)}{d\left(n_\perp\right)} = \frac{2\ E_{xy}\ E_x\ E_y - E_y^2\ E_{xx} - E_x^2\ E_{yy}}{(E_x^2 + E_y^2)^{\frac{3}{2}}}$$

The corner measurement of Kitchen-Rosenfeld is obtained multiplying the above expression by the modulus of the gradient:

$$K = \frac{2\ E_{xy}\ E_x\ E_y - E_y^2\ E_{xx} - E_x^2\ E_{yy}}{(E_x^2 + E_y^2)} \tag{1}$$

We search along the extracted lines for points with maximum K. In Fig. 2 and Fig. 3 a scene of the laboratory with the extracted segments and the points obtained, can be seen. The cornerness operator K supplies (with little computational cost) not only a qualitative measurement of tips of lines but also a quantitative measurement of point goodness.

In Fig. 4 examples of the corner measurement K along two lines in a real image are represented. K operator is null on straight lines, and it can be observed how some maximums appear in the K parameter, which are related to well-defined points. Often selected points are not coincident with tips obtained by the contour extractor.

IV. Matching features

In many vision algorithms feature extraction is followed by correspondence search over the image sequence. Our aim is to determine correspondences between linear segments in a monocular sequence

Fig. 3. Characteristic points selected along the contours of figure 2.

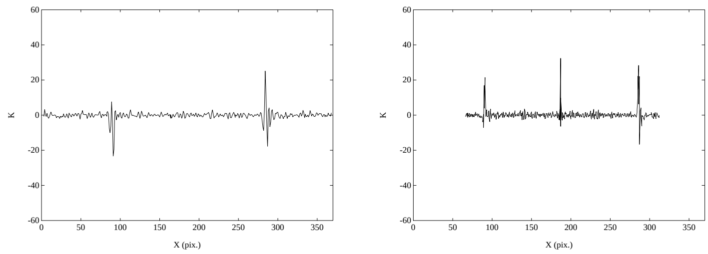

Fig. 4. Examples of the K measurement along two contours in a real image. In the first example two points are easily selected. In the second one, three characteristic points are obtained.

of images without knowledge about motion or scene structure. The camera is on a mobile robot which runs in a man made environment. Nothing about the camera motion is assumed except that we have an homogeneous sequence of images, nearly continuous in time. In this field correspondence determination by tracking geometric information in the image sequence has been proposed as a good solution [17], [18], [19]. The method uses a prediction-match-update loop, based on Kalman equations (Fig. 5).

However, we use not only geometric parameters but also brightness attributes supplied by the contour extractor. We propose also to consider a dynamic model of brightness information which is explicitly adapted, although some illumination change could appear [20]. The tracking and matching of brightness information can be made with very short computational overload, being crucial when motion or structure are not assumed. As camera motion is not known, geometric constraints are only locally valid and must be imposed in an heuristic way. For example epipolar constraint, used in stereo matching [21], can not be used here. Other works consider brightness parameters [22], [23], but they do not consider dynamics when dealing with them.

We particularize this presentation with straight segments, but the method can be used with other

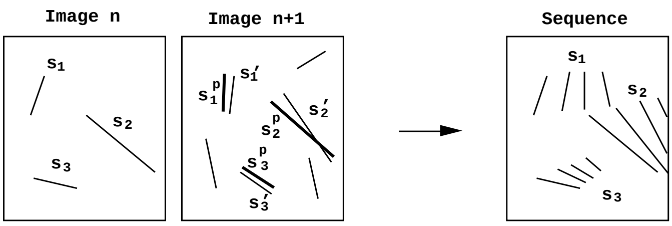

Fig. 5. Matching features by *tracking* in the image (s^p predicted, s' observed)

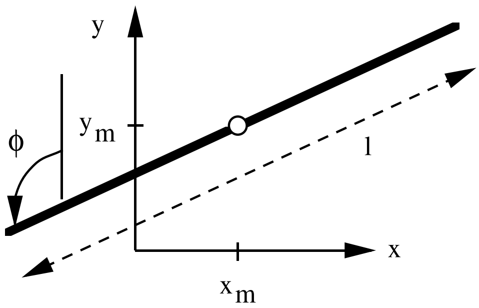

Fig. 6. Image segment representation.

features. As we have identified points along contours, the computation of matched segments gives indirectly corresponding points.

A. Brightness information

Most of the methods based on features do not use brightness information once the geometric features have been extracted. Therefore they through away useful information in subsequent processes (which are usually difficult). On the other hand, alternate methods to determine motion (optical flow methods) are based on the conservation of brightness information. Precision of constant brightness model has been questioned by some researchers [8]. Relaxation of the constant model has been proposed to deal with brightness variations not related with motion [24].

As said above we use brightness attributes to obtain better matching. We propose a relaxation of the constant model for brightness attributes, similar to kinematic models of geometric tracking, which allows slow variations of brightness information. Noise in state model allows the adaptation to brightness changes due to shadows, saturation of sensor, or other events. Besides that, our model includes the classical constant model when the variation predicted is null. We have analyzed three brightness attributes:

- Average gray level (agl).
- Contrast (c), characterized by the standard deviation of brightness.
- Steepness of brightness change (s).

We have made a set of experiments (resumed in §VI-B), and we have finally selected agl and c. The method is general to consider other kind of attributes.

B. Tracking in the image

As well as these brightness attributes, classical location image parameters of the segment are considered. We choose the image midpoint representation, which takes three location parameters x_m, y_m, ϕ and the length l of the image segment (Fig. 6). With this geometric representation we can assume no correlation between midpoint location and ϕ, l parameters [17]. Therefore, separate Kalman filters with a reduced dimension can be used to improve the efficiency.

Thus, image segment representation is composed of 6 parameters (4 geometric and 2 brightness parameters):

- x_m - X midpoint coordinate.
- y_m - Y midpoint coordinate.
- ϕ - Segment orientation.
- l - Segment length.
- agl - Average grey level in the LSR.
- c - Contrast of the contour.

The goal of the proposed tracking is to compute corresponding segments along an image sequence. The association between a target and its corresponding measurement is made by using a prediction-match-update loop based on Kalman Filter equations. The match is made with the nearest neighbor measurement, using a similarity function based on the Mahalanobis distance.

By using an extensive description of the target such as the proposed here, a correct measurement is selected without high computational cost. Better matches could be made postponing association decisions, examining information from several frames and/or considering all targets simultaneously. However these other tracking techniques like the track-splitting, joint-likelihood and multiple-hypothesis [25], all have exponential complexity, requiring large memory and computational resources, which make them useless in most computer vision applications. As the camera moves, new image areas appear and others disappear. Thus, the joint-probabilistic data association filter, which uses fixed computation resources, does not work out applicable because the number of targets must be known a priori.

B.1 System model

It is well known that the Kalman Filter equations are computed directly from the linear space state model of a stochastic system, that is:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{v}(k) & (2)\\ \mathbf{z}(k) &= \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k) & (3)\\ \mathbf{E}\left[\mathbf{v}(k)\mathbf{v}(j)'\right] &= \mathbf{Q}(k)\delta_{kj} & (4)\\ \mathbf{E}\left[\mathbf{w}(k)\mathbf{w}(j)'\right] &= \mathbf{R}(k)\delta_{kj} & (5) \end{aligned}$$

Equations (2) and (4) represent the evolution of each segment irrespective of how it is detected. As it was said, the segment is determined by $x_m, y_m, \phi, l, agl, c$. We assume that every segment parameter is decoupled from the others.

In order to pattern evolution of segment parameters along the sequence, different heuristic models can be used. We can consider a random walk model in which the state vector only includes the position and therefore $\mathbf{F}_i = 1$. There are some changes which can be modeled with a state noise. If we model the variation as white noise without mean and v_i covariance, then process noise covariance is $\mathbf{Q}_i = \Delta t^2 v_i$. In both cases i refers to each parameter. This model will be referred below as *position model*.

A second model can be considered. It includes variation of the parameters in the state, in such a way that:

$$\mathbf{x} = \left(x_m, \dot{x}_m, y_m, \dot{y}_m, \phi, \dot{\phi}, l, \dot{l}, agl, \dot{agl}, c, \dot{c},\right)^T$$

For each pair parameter-derivative we define its state equation by means of:

$$\mathbf{F}_i = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

However, the constant velocity can be considered valid only locally. There are some variations which can be modeled with the assumption of the target having a constant acceleration in each sample time. If we model the acceleration as a white noise without mean and q_i covariance, then process noise covariance is [26]:

$$\mathbf{Q}_i = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} q_i$$

To sum up, model equations when considering constant velocity are completely defined except in that referred to the state covariance of the parameters. These covariance values $(q_{x_m}, q_{y_m}, q_\phi, q_l, q_{agl}, q_c)$, must be tuned for each type of image sequence. The model just presented will be referred to as *velocity model* in §VI.

We have also made some experiments with a third model of constant acceleration which will be referred to as *acceleration model.* We do not give here more details about it because their equations can be easily deduced [26].

Regarding location parameters, the constant velocity model appears as a reasonable model when image sequence is nearly continuous in time, which normally happens when it comes from the motion of a mobile robot. With regard to the brightness attributes the stochastic position and velocity models appear more reliable than those of constant brightness assumed in other works, especially when brightness change due to non-motion aspects are important. We have observed experimentally a good fitting to the locally constant velocity model, because changes in most scene events that contribute to brightness variations in the image, are usually slow in time [24].

Therefore, we have selected the velocity kinematic model, being also useful to consider the acceleration model for location parameters and the position model for the brightness attributes.

Equations (3) and (5) are used to model how the camera detects the segment. The matrix $\mathbf{H}$ selects the values corresponding to the segment parameter because the derivatives are not measured.

In contrast to the state covariance $\mathbf{Q}$, there is some correlation between the different segment parameters in the measurement noise $\mathbf{R}$:

- As it was proposed in [17], measurement noise along the segment direction is bigger than noise in the orthogonal direction. Location uncertainties of segment tips in these directions are represented by $\sigma_\parallel^2$ and $\sigma_\perp^2$ respectively. It is well worth noting that x_m and y_m are coupled themselves in the measurement equation while they are decoupled in the dynamic equation. When a segment is detected, its orientation points out where the error can be expected, mainly along the segment because of partial occlusion or because of noisy segment extraction. However the dynamic equation represents the evolution of the ideal segment and therefore the orientation is irrelevant. This evolution is mainly determined by the unknown camera motion.
- Measurement noise corresponding to agl and c can be considered independent.

So, the $\mathbf{R}$ matrix is:

$$\begin{bmatrix} \sigma_\perp^2\mathrm{C}^2+\sigma_\parallel^2\mathrm{S}^2 & \sigma_\perp^2\mathrm{CS}-\sigma_\parallel^2\mathrm{CS} & 0 & 0 & 0 & 0 \\ \sigma_\perp^2\mathrm{CS}-\sigma_\parallel^2\mathrm{CS} & \sigma_\parallel^2\mathrm{C}^2+\sigma_\perp^2\mathrm{S}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\frac{\sigma_\perp^2}{l^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\sigma_\parallel^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{agl}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_c^2 \end{bmatrix}$$

where $\mathrm{C} = \cos\phi$, $\mathrm{S} = \sin\phi$.

Measurement noise covariance is not constant, but dependent on the measured values ϕ, l. Thus, the measurement model is completely defined except for $\sigma_\parallel^2, \sigma_\perp^2, \sigma_{agl}^2, \sigma_c^2$, which should be tuned.

B.2 Kalman filter equations

B.2.a State prediction.

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k) &= \mathbf{F}\hat{\mathbf{x}}(k|k) \\ \mathbf{P}(k+1|k) &= \mathbf{F}\mathbf{P}(k|k)\mathbf{F}^T + \mathbf{Q} \end{aligned}$$

It should be noticed that $\mathbf{F}$, $\mathbf{Q}$ are constant matrices and $\hat{\mathbf{x}}(0|0)$, $\mathbf{P}(0|0)$ must be initialized.

B.2.b State update.

$$\begin{aligned}
\mathbf{r}(k+1) &= \mathbf{z}(k+1) - \mathbf{H}\hat{\mathbf{x}}(k+1|k) \\
\mathbf{S}(k+1) &= \mathbf{H}\mathbf{P}(k+1|k)\mathbf{H}^T + \mathbf{R}(k+1) \\
\mathbf{K}(k+1) &= \mathbf{P}(k+1|k)\mathbf{H}^T\mathbf{S}^{-1}(k+1) \\
\hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)\mathbf{r}(k+1) \\
\mathbf{P}(k+1|k+1) &= \mathbf{P}(k+1|k) - \mathbf{K}(k+1)\mathbf{S}(k+1)\mathbf{K}^T(k+1)
\end{aligned}$$

where $\mathbf{S}(k+1)$ is the covariance of the innovation $\mathbf{r}(k+1)$.

B.3 Matching criteria

Due to the block structure of the system model, $\mathbf{S}$ is a block diagonal matrix where we can differentiate 4 blocks, and their correspondent subvectors in the innovation $\mathbf{r}$:
- A *location* 2×2 block, $\mathbf{S}_{\text{loc}}$, with its corresponding subvector $\mathbf{r}_{\text{loc}}$. It includes the innovation of the x_m and y_m parameters and its covariance.
- A *orientation* 1×1 block, $\mathbf{S}_{\text{ori}}$, with its corresponding subvector $\mathbf{r}_{\text{ori}}$. It includes the innovation of the ϕ parameter and its covariance.
- A *length* 1×1 block, $\mathbf{S}_{\text{len}}$, with its corresponding subvector $\mathbf{r}_{\text{len}}$. It includes the innovation of the l parameter and its covariance.
- A *brightness* 2×2 block, $\mathbf{S}_{\text{br}}$, with its corresponding subvector $\mathbf{r}_{\text{br}}$. It includes the innovation of the *agl* and c parameters and its covariance.

The proposed matching technique can be seen as a variation of the nearest neighbor standard filter using the localization and orientation parameters only. The Mahalanobis distance would be:

$$d_M = \mathbf{r}_{\text{loc}}^T\mathbf{S}_{\text{loc}}^{-1}\mathbf{r}_{\text{loc}} + \mathbf{r}_{\text{ori}}^T\mathbf{S}_{\text{ori}}^{-1}\mathbf{r}_{\text{ori}} \tag{6}$$

But some measurements are excluded from the acceptance region using length and brightness coherence. Thus, we propose to compare only the measurements which have passed two additional Mahalanobis tests, one for $\mathbf{r}_{\text{br}}$ and another for $\mathbf{r}_{\text{len}}$. The matching algorithm for each target can be summarized as:
1. Compute all the measurements compatible in location and orientation with prediction, and obtain the set $\mathcal{H}_{\text{loc+ori}}$:

$$\mathcal{H}_{\text{loc+ori}} = \left\{\mathbf{z}|(\mathbf{r}_{\text{loc}}^T\mathbf{S}_{\text{loc}}^{-1}\mathbf{r}_{\text{loc}} + \mathbf{r}_{\text{ori}}^T\mathbf{S}_{\text{ori}}^{-1}\mathbf{r}_{\text{ori}}) \leq \chi_3^2(95\%)\right\}$$

where χ_3^2 is the chi-square function with 3 degrees of freedom and a significance level of 5%. $\mathbf{R}$ depends on the measurement (ϕ and l), but the measurement has not been selected yet. So, to reduce the computational cost, we use an approximation to the innovation covariance, $\mathbf{S}^*$, based on the measurement prediction, $\hat{\phi}(k+1|k)$ and $\hat{l}(k+1|k)$, instead of the exact measured values. It could be obtained as:

$$\begin{aligned}
\mathbf{S}^*(k+1|k) &= \mathbf{H}\mathbf{P}(k+1|k)\mathbf{H}^T + \\
&+ \mathbf{R}(\hat{\phi}(k+1|k), \hat{l}(k+1|k))
\end{aligned}$$

2. Compute all the measurements compatible in brightness with the target prediction, and obtain the set $\mathcal{H}_{\text{br}}$:

$$\mathcal{H}_{\text{br}} = \left\{\mathbf{z}|\mathbf{r}_{\text{br}}^T\mathbf{S}_{\text{br}}^{-1}\mathbf{r}_{\text{br}} \leq \chi_2^2(95\%)\right\}$$

3. Compute all the measurements compatible in length with the target prediction, and obtain the set $\mathcal{H}_{\text{len}}$:

$$\mathcal{H}_{\text{len}} = \left\{ \mathbf{z} | \mathbf{r}_{\text{len}}^T \mathbf{S}_{\text{len}}^{-1} \mathbf{r}_{\text{len}} \leq \chi_1^2(95\%) \right\}$$

4. Compute the set of valid hypothesis, $\mathcal{H}$, as the intersection of all hypothesis sets:

$$\mathcal{H} = \mathcal{H}_{\text{loc+ori}} \cap \mathcal{H}_{\text{br}} \cap \mathcal{H}_{\text{len}}$$

The search could be only applied to those measurements which have passed the previous tests, being useful to start with the highest demanding.

5. Select, as the corresponding measurement, the nearest one in the hypothesis set $\mathcal{H}$ applying (6).

With this procedure, compensation between high precision in some parameter with high error in other parameter is avoided. A general Mahalanobis distance for the six parameters is not used because the correct weighting of different information as brightness based and location based in a sole distance is difficult and could easily lead to bad matches.

V. Camera maneuvering

The presented tracking method works correctly if there are not sudden changes in image velocity. Normally, the main cause of changes is the camera motion, which affects to every feature in the image. We propose to detect and to measure the maneuver by using the most relevant features with their brightness information. The maneuver estimate is used to correct predictions of the other features.

Maneuvering targets can be characterized [26] by a dynamic equation:

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) + \mathbf{v}(k) \tag{7}$$

where inputs $\mathbf{u}(k)$ or maneuvers are unknown.

We assume a nonrandom unknown input that will be estimated using the model without input (non-maneuvering model). Assuming which a maneuver appears essentially instantaneous (which is a reasonable assumption because smooth changes can be tracked as before), and using the innovations of the Kalman filter of the non-maneuvering model, the input can be detected, estimated and used to correct the state estimate of the targets. This can be made in a sole period, but observations of some previous periods are needed in order to have a good estimation of the state without maneuver.

Another method to obtain the maneuver could be to augment the state with the input which is reestimated within the augmented state, but in our case this solution gives too big state size, because it must include the inputs and the location of all targets. Besides that when the maneuver appears instantaneous is better to use the input estimation method, which only considers their detection and correction, ignoring other details of the maneuver [26].

In most of the applications, scene is far from the camera in such a way that camera rotations affect to the localization of the features in the camera much more than camera translations. The displacements in the image due to camera rotation can be predicted without knowledge of scene depth. With pure rotation, each point in 3D moves according to a circle and its image projection according to some quadratic curve (ellipses, hyperbolas, parabolas or circles). Rotations on a plane parallel to the image plane give hyperbolic trajectories for image points. This means that, if camera turns with W_x, W_y velocities, an image point (x, y) suffers a projected displacement with $(\dot{x}, \dot{y})$ velocity [27]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} xy & (-x^2 - 1) \\ y^2 + 1 & -xy \end{bmatrix} \begin{bmatrix} W_x \\ W_y \end{bmatrix}$$

As the camera field of view is normally quite short $(x, y \ll 1)$, the rotations being parallel to the image plane give displacement velocity in the image nearly constant $[\dot{x}, \dot{y}] \approx [-W_y, W_x]$. On the other side, when the camera is attached to a mobile robot, the component of rotation around the focal axis is nearly null.

We can consider that the main causes of maneuvers are the unpredictable camera rotations. Considering that these are around an axis being parallel to the image plane, its effect is nearly equal for every point in the image. This involves that the orientation of segments do not change, and therefore only changes of location in the image (x_m, y_m) are considered as maneuvers. In this way, camera calibration is not needed because the displacement is independent for both directions.

A. Detection, estimation and correction

It is known that innovations corresponding to the correct filter (maneuvering filter) $\mathbf{r}^m(k+1)$ are a zero-mean white noise sequence with covariance $\mathbf{S}^m(k+1)$. As shown in [26] the innovation of the non-maneuvering filter $\mathbf{r}^{nm}(k+1)$ is the same adding a term related to the inputs. When maneuver appears instantaneous, this relation is:

$$\mathbf{r}^{nm}(k+1) = \mathbf{r}^m(k+1) + \mathbf{HGu}(k) \tag{8}$$

Therefore the innovation $\mathbf{r}^{nm}$ of the non-maneuvering filter can be considered a linear measurement of the maneuver $\mathbf{u}$ in the presence of the additive white noise $\mathbf{r}^m$.

Assuming the maneuver exposed above, the input can be estimated using the least-squares method. In order to do that, we use the innovation $\mathbf{r}^{nm}(k+1)$ with covariance $\mathbf{S}(k+1)$ corresponding to the non maneuver filter, of some *robust* targets $(1...t)$. Therefore, naming:

$$\mathbf{Q} = diag(\mathbf{S}_1(k+1), \mathbf{S}_2(k+1), ...\mathbf{S}_t(k+1))$$

$$\mathbf{z} = \begin{bmatrix} \mathbf{r}_1^{nm}(k+1) \\ . \\ . \\ . \\ \mathbf{r}_t^{nm}(k+1) \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} \mathbf{HG}_1 \\ . \\ . \\ . \\ \mathbf{HG}_t \end{bmatrix}$$

the maneuver can be estimated as:

$$\hat{\mathbf{u}}(k) = (\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A})^{-1}\ \mathbf{A}^T\ \mathbf{Q}^{-1}\ \mathbf{z} \tag{9}$$

with a matrix covariance:

$$\mathbf{M}_{\mathbf{u}}(k) = (\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A})^{-1} \tag{10}$$

Considering the maneuver as a position change, we take $\mathbf{G} = [1, 0]^T$ for each localization parameter $(x_m,\ y_m)$. If it corresponds with velocity changes, we will take $\mathbf{G} = [\Delta t, 1]^T$.

Thus, the proposed algorithm for the detection, estimation and correction of a global localization maneuver is as follows:

1. Select a set of candidate targets to be initially matched allowing at the same time to obtain a hypothesis of the maneuver. This selection can be based on several criteria:

- *Isolation in the image.* Those features which are isolated in the image can be easily and more robustly matched. This can be estimated with the size of the set of hypothesis $(\mathcal{H}_{\text{loc}},\ \mathcal{H}_{\text{ori}},\ \mathcal{H}_{\text{len}},\ \mathcal{H}_{\text{br}})$ of each target in previous matches.

- *Outstand of the feature.* If a segment belongs to other non plentiful feature, the number of possible matches decrease.

- *Centering in the image.* Those features which are centered in the image have less probability of disappearance as the camera moves.
2. For each candidate target, select the set of compatible hypothesis according to the orientation, length and brightness:

$$\mathcal{H}^{nm} = \mathcal{H}_{\text{ori}} \cap \mathcal{H}_{\text{len}} \cap \mathcal{H}_{\text{br}}$$

3. From this set of hypothesis, compute the closest measures using four Mahalanobis distances:

$$\begin{aligned} d_1 &= \mathbf{r}_{\text{br}}^T \mathbf{S}_{\text{br}}^{-1} \mathbf{r}_{\text{br}} \\ d_2 &= \mathbf{r}_{\text{len}}^T \mathbf{S}_{\text{len}}^{-1} \mathbf{r}_{\text{len}} \\ d_3 &= \mathbf{r}_{\text{ori}}^T \mathbf{S}_{\text{ori}}^{-1} \mathbf{r}_{\text{ori}} \\ d_4 &= \mathbf{r}_{\text{loc}}^T \mathbf{S}_{\text{loc}}^{-1} \mathbf{r}_{\text{loc}} \end{aligned}$$

If the selected measure coincide in every test, the match is considered as *robust* to estimate the maneuver.
4. With the *robust* matches the maneuvers are estimated from (9) and (10).
5. An estimate is accepted, only if it is statistically significant. If the input were null, $\hat{\mathbf{u}}$ would be a normal random variable with zero mean and covariance $\mathbf{M}_{\mathbf{u}}$. Then $\hat{\mathbf{u}}^T(\mathbf{M}_{\mathbf{u}})^{-1}\hat{\mathbf{u}}$ is chi-square distributed with 1 degree of freedom. The maneuver is declared detected when the probability of false alarm is 0.01 or smaller and therefore when:

$$\hat{\mathbf{u}}^T(\mathbf{M}_{\mathbf{u}})^{-1}\hat{\mathbf{u}} \geq \chi_1^2(99\%)$$

6. If the maneuver is detected, the state prediction of all targets must be corrected by using the estimated maneuver.

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k) &= \hat{\mathbf{x}}^{nm}(k+1|k) + \mathbf{G}\hat{\mathbf{u}}(k) \\ \mathbf{P}(k+1|k) &= \mathbf{P}^{nm}(k+1|k) + \mathbf{G}\mathbf{M}_{\mathbf{u}}(k)\mathbf{G}^T \end{aligned}$$

7. The other targets are matched by considering this correction of the state prediction.

VI. Experimental results

In this paragraph we present experimental results with the proposed *tracker*. We emphasize the properties of brightness attributes and their capacity to select features.

A. Matching segments

To carry out the experiments we have used a set of eight images (Fig. 7). The motion is composed of a rotation around Y axis and a little translation in X direction. Scene illumination is natural and it saturates easily the sensor (the light comes from the windows). The focal length of the camera used is 12 *mm*. It works with the automatic gain control system (agc on). At the moment, only the contours detected in the first image are considered as targets for the *tracking* algorithm.

In a first experiment the algorithm of extraction has been tuned to select the segments in length and brightness in such a way that approximately 30 segments appear in each image. All the tracker matches were the same as the matches manually made, except a segment in a cluttered area (24 in Fig. 7) which has been lost by the *tracker*. We must say that the 98.5% of the matches would have been the same using as the main criterion the closest one according to brightness instead of the closest one according to location.

TABLE I

INNOVATION QUADRATIC ERROR USING POSITION FILTER, VELOCITY FILTER AND ACCELERATION FILTER WITH THE SAME MEASUREMENT NOISE AND SIMILAR INNOVATION COVARIANCE.

Model	*Innovation error* ($\overline{r^2}$)		
	$agl\ (glu^2)$	$c\ (glu^2)$	$s\ (\frac{glu^2}{pix^2})$
position	2.69	4.08	0.96
velocity	3.06	3.94	0.94
acceleration	3.33	4.05	0.97

A second sequence of segments obtained from the same images but with a less exigent length and gradient filtering has been tested (Fig. 8). The average number of segments extracted in each frame were close to 100. In this case there was a spurious match (segment 11 in Fig. 8) and some lost matches (17, 20, 77). The number of targets matched in all images was 40, which implies a similar percentage to the previous experiment. The 97.5% of matches (from 396) would have been the same by taking the closest one according to brightness criteria. In this experiment, the mean number of segments included in the brightness hypothesis set $\mathcal{H}_{\text{br}}$ was about the 5% of the mean number of segments extracted in each frame. Therefore the brightness information allows to eliminate a high number of possible matches. This selection capacity is treated more broadly in the following paragraph.

B. Brightness information

We have carried out some experiments to evaluate the selection capacity of the brightness parameter and its adjust to the proposed evolution model along a image sequence. The results are difficult to generalize but they can be considered relevant enough because we have used a complex sequence, not only in relation to geometric information but also in illumination conditions.

We have used the segments correctly matched in the sequence of the first experiment. So, we can observe in Fig. 9 the evolution of the three brightness parameters used for different contours.

Some significant examples of the evolution of the predictions and the measurements can be seen in Fig. 10. The filter has a quick convergence to the permanent. Thus an invariant filter as those presented in [26] can be used to improve the efficiency.

We have tested for the segments in the sequence the evolution of the brightness parameters to the dynamic models presented above (position model, velocity model and acceleration model). The compared filters have a similar size of search area. These models have been tested by comparing the mean value of quadratic prediction error.

As can be seen in table I, the three models have similar behavior. Thus, the position model turns out good enough in this sequence. Anyway, the velocity model works better in some cases, by example moving towards light sources or changing the illumination conditions of the scene.

Selection capacity of each attribute has been evaluated by comparing the mean innovation quadratic error normalized with the range of values of each attribute. We have tested that the most selective attribute is the average gray level. In the sequence presented its *selection capacity* is $\frac{1}{14}$, being $\frac{1}{5.5}$ for the contrast and $\frac{1}{5}$ for the steepness.

These measurements have also allowed to compare the method to extract the contrast and steepness attributes. We have compared the proposed in this work with the presented in [3] (table II). Referred to the contrast, the method presented here gives shorter innovation error in a ratio of $\frac{1}{1.5}$. Referred to the steepness of brightness, our method gives also shorter innovation error in a ratio of $\frac{1}{1.9}$.

TABLE II
INNOVATION QUADRATIC ERROR FOR THE CONTRAST AND THE STEEPNESS USING THE BURNS METHOD (FIRST AND SECOND COLUMNS) AND USING THE METHOD PROPOSED BY US.

c	s	$c = 2\sigma(E)$	$s = \frac{c}{width}$
0.0441	0.1054	0.0189	0.0285

VII. Conclusions

The main goal of our work is to compute robot motion and scene structure in a man made environment. To carry it out, extracting corresponding features is an essential step.

We have presented a method to obtain straight contours, which gives us geometric information and also parametric information related to the brightness of the region supporting the contour. We have also presented a point extractor to complement the information on straight contour with its tips, or some other characteristic points. This complementary information offers plenty of advantages to compute structure and motion.

A feature matcher to track segments in the image has been presented. The technique used is suitable to real time vision. It uses not only geometric description of the segment, but also the *robust intensity description* obtained with the extractor. The proposed variable brightness model makes the feature matcher work correctly although slight changes (shading, agc, ...) are present. This tracker can be used even if neither the camera motion nor the scene structure is known, provided a sequence of images moving nearly continuously in time is available. Our contribution here outlines the use of a brightness dynamic model, which makes it possible for the tracker to adapt to illumination changes. We have also proposed the detection and computation of camera maneuvers, which is very useful as the camera is fitted on a robot. All this was feasible because we took not only a geometric but also a complete description of the targets.

It has been observed from experimental results that the matching by using our model of brightness information works well and it allows disambiguate matches in cluttered scenes. A reduction of about 95% of the initial number of candidates to match can be obtained robustly by using the brightness information. We have also shown that the average gray level has a selection capacity higher than the contrast and the brightness steepness across the contour.

References

[1] Y. Liu and T.S.Huang, "Estimation of rigid body motion using straight line correspondences," *Computer Vision, Graphics and Image Processing*, , no. 43, pp. 37–52, 1988.

[2] J.K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images - a review," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 917–935, 1988.

[3] J.B. Burns, A.R. Hanson, and E.M. Riseman, "Extracting straight lines," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 4, pp. 425–455, 1986.

[4] D. Marr, *Vision*, Freeman, San Francisco CA, 1982.

[5] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[6] R. Deriche, "Using canny's criteria to derive an optimal edge detector recursively implemented," *International Journal of Computer Vision*, vol. 1, pp. 167–187, 1987.

[7] D.H. Ballard and C.M. Brown, *Computer Vision*, Prentice Hall, Englewood Cliffs, N.J., 1982.

[8] O. Faugeras, *Three-Dimensional Computer Vision. A Geometric Viewpoint*, The MIT Press, Massachusetts, 1993.

[9] J.J. Guerrero, *Percepción de Movimiento y Estructura con Visión basada en Contornos Rectos*, Ph.D. thesis, Dpto. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain, Mayo 1996.

[10] M.E. Spetsakis and Y. Aloimonos, "Structure from motion using line correspondences," *International Journal of Computer Vision*, , no. 4, pp. 171–183, 1990.

[11] X. Lebégue and J.K. Aggarwal, "Significant line segments for an indoor mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp. 801–815, 1993.

[12] J.J. Guerrero, C. Sagüés, and A. Lecha, "Motion and structure from straight edges with tip," in *IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, USA, Oct 1994, pp. 2459–2464.

[13] J. Cooper, S. Venkatesh, and L. Kitchen, "Early jump-out corner detectors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 8, pp. 823–828, 1993.

[14] R. Deriche and G. Giraudon, "A computational approach for corner and vertex detection," *International Journal of Computer Vision*, vol. 10, no. 2, pp. 101–124, 1993.

[15] L. Kitchen and A. Rosenfeld, "Gray level corner detection," *Pattern Recognition Lett.*, vol. 1, pp. 95–102, 1982.

[16] X. Xie, R. Sudhakar, and H. Zhuang, "Corner detection by a cost minimization approach," *Pattern Recognition*, vol. 26, no. 8, pp. 1235–1243, 1993.

[17] R. Deriche and O. Faugeras, "Tracking line segments," in *First European Conference on Computer Vision*, Antibes, France, 1990, pp. 259–268.

[18] J.L. Crowley, T. Skordas, and P. Puget, "Measurements and integration of 3-d structures by tracking edge lines," *International Journal of Computer Vision*, vol. 8, no. 1, pp. 29–52, 1992.

[19] Z. Zhang, "Strategies for tracking tokens in a cluttered scene," in *British Machine Vision Conference*, Surrey, September 1993.

[20] J.J. Guerrero and J.M. Martínez, "Determination of corresponding segments by tracking both geometrical and brightness information," in *International Conference on Advanced Robotics*, Barcelona, September 1995, pp. 703–709.

[21] N. Ayache, *Artificial Vision for Mobile Vision. Stereo Vision and Multisensory Perception*, MIT Press, London, 1991.

[22] B. Giai-Checa, R. Deriche, T. Viéville, and O. Faugeras, "Suivi de segments dans une séquence d'images monoculaire," Rapport de recherche RR-2113, I.N.R.I.A., Sophia-Antipolis, France, 1993.

[23] Y. Liu and T.S. Huang, "Determining straight line correspondences from intensity images," *Pattern Recognition*, vol. 24, no. 6, pp. 489–504, 1991.

[24] S. Negahdaripour and C.H. Yu, "A generalized brightness change model for computing optical flow," in *Fourth International Conference on Computer Vision*, Berlin, May 1993.

[25] I.J. Cox, "A review of statistical data association techniques for motion correspondence," *Int. Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[26] T. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press In., 1988.

[27] B.K.P. Horn, *Robot Vision*, MIT Press, Cambridge, Mass., 1986.

Fig. 7. Extracted contours in the first image and followed contours until the eighth image. Gradient ≥ 15 and length ≥ 40

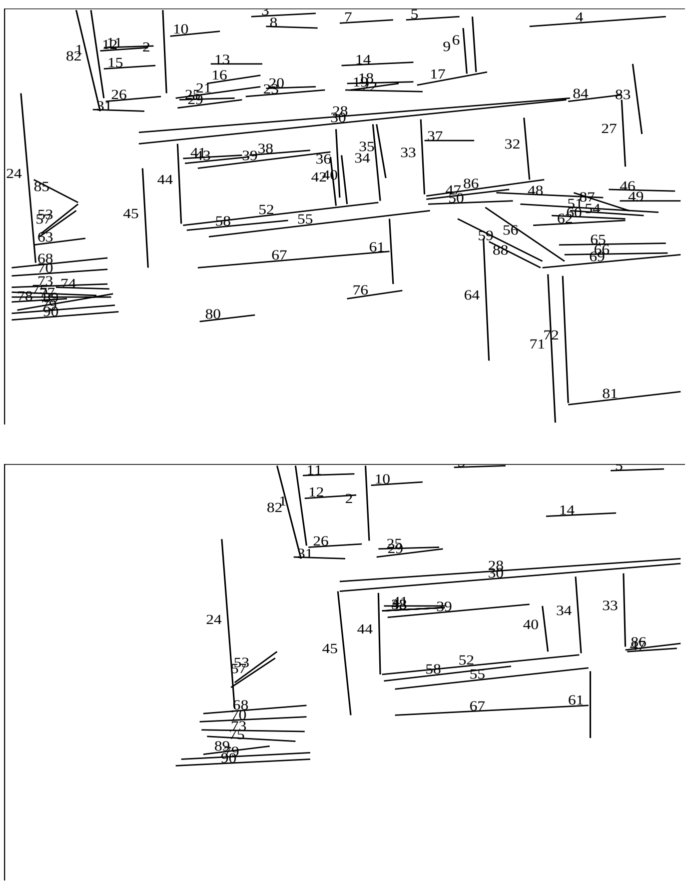

Fig. 8. Extracted contours in the first image and followed contours until the eighth image. Gradient ≥ 10 and length ≥ 25.

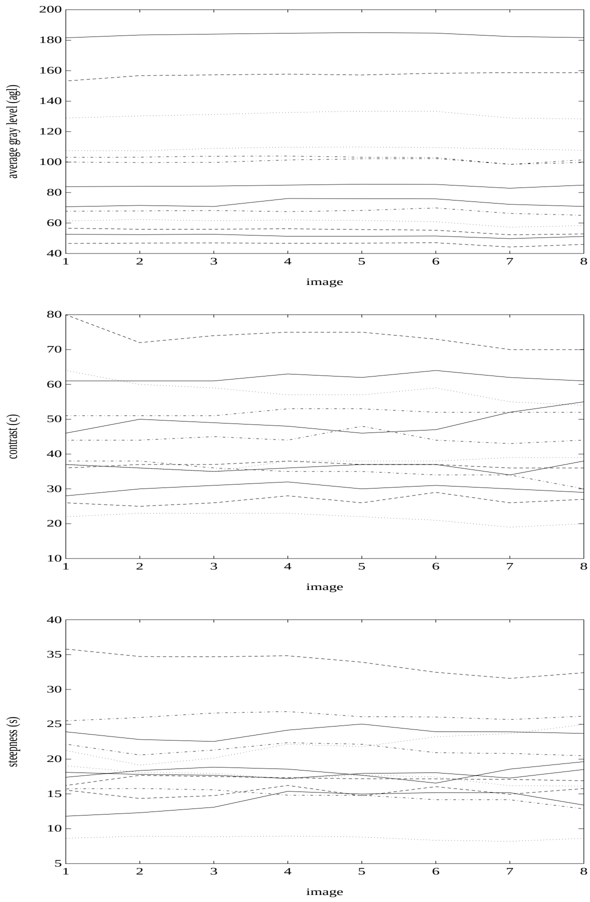

Fig. 9. Evolution of the brightness parameters in the sequence for different contours.

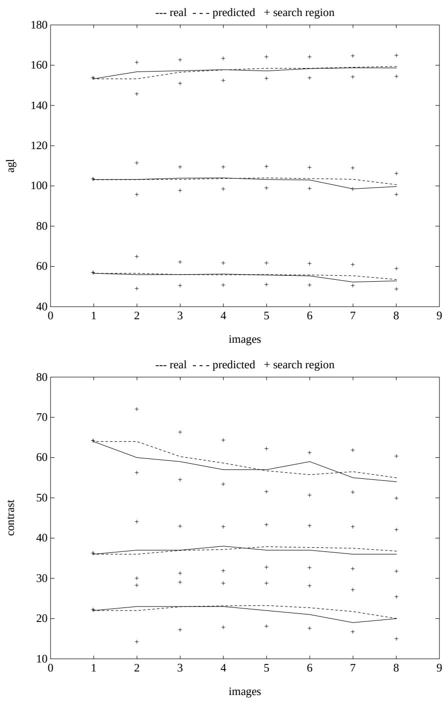

Fig. 10. Evolution of the average gray level and the contrast with their prediction and search area (95%) for some significant targets. They are expressed in gray level units glu.