Proceedings of ICAR 2003
The 11th International Conference on Advanced Robotics
Coimbra, Portugal, June 30 - July 3, 2003

Lane Extraction and Tracking for Robot Navigation in Agricultural Applications

Gabriel Aviña-Cervantes* Michel Devy Antonio Marín-Hernández

LAAS - CNRS
Laboratoire d'Analyse et d'Architecture des Systèmes
7, Avenue du Colonel Roche, 31077 Toulouse Cedex 4, FRANCE
e-mail: {gavina, michel, amarin}@laas.fr

Abstract

In this paper, we propose a method for extracting and tracking man-made roads. It could be used for robot navigation in agricultural environments or hazardous related areas, as it is able to detect and track roads from images provided by an on-board color camera. Road extraction is achieved using color segmentation and principal areas detection; road tracking is achieved by active contours. First of all, our approach segments color images in small areas, which will be characterized later by color and texture attributes. These features are classified using the K-NN rule or the Support Vector Machines (SVM) method. A global scene model is obtained where the road extraction is used to initialize the active contour tracking process. Some of the road features are also used, in a focalized gradient zone in order to attract active contour to the road boundary. Besides, using the scene model information we can correct the transient errors generated in the tracking procedure. This algorithm has been evaluated on a great number of images, acquired either on secondary tarred roads or on earthen roads, mainly in countryside scenes. Results obtained on image sequences show the robustness of the proposed approach.

keywords: *Color segmentation, classification, texture, tracking, snakes.*

1 Introduction

Navigation in natural environments requires a higher level of scene understanding and of vehicle control than for planetary or static urban environments. Autonomous guided vehicles have found many applications in many industries and the most of them have to navigate in unstructured environments [1]. Robotization of agricultural machines, like automatic harvesting of fruits, requires also an autonomous navigation, on a network of roads, to go for example from a farm to a given field; with respect to the technologies developed for intelligent vehicles, mainly on motorways, this application provides a new kind of problems in scene interpretation, due to the complexity of the environments.

In this context, modules that perform path planning or execution control of the trajectories, must analyse images captured by cameras mounted on the robots or vehicles. These images are interpreted to extract useful information such as positions, obstacles (trees, rocks…), road boundaries and the position and orientation (heading) of the vehicle on the road. The construction of a complete model of outdoor natural environment is one of the most difficult tasks in Computer Vision; the complexity lies on several factors such as the great variety of scenes, images of non-structured and non rigid objects (trees, clouds…), low control in the acquisition conditions (as illumination, temperature and sensor motion) and combined with this complexity, the requirement of fast algorithms to allow real time implementations. Only some of these factors can easily be overcome.

For visual navigation in natural environments [2], the color segmentation should be considered as a basic operation. Image segmentation, i.e. the extraction of homogeneous regions in an image, has been the subject of many researchers, mainly for gray scale images. However, the segmentation for color images, which convey much more information about the objects in the scene, has received lower attention of the scientific community.

In [3], a basic approach for image interpretation used in natural environments has been proposed. It consists in several steps: first, a color segmentation algorithm provides a description of the scene as a set of the most representative regions. These regions are characterized by several attributes (color and texture) [5, 6], and finally their nature is identified [8, 9]. The scene model is required for two functions: landmark-based modeling and landmark tracking. Murrieta[3] has evaluated this approach to recognize rocks, trees or grassy terrains from color images. In [4], a pre-classification step was used in order to select the database according to some global classification based on the images: this step allows to use the best knowledge database depending on the season (winter or sum-

*The author thanks the support of the CONACyT. This work has been partially supported by ECOS-Nord action No. M99M01, funded by the French and Mexican governments.

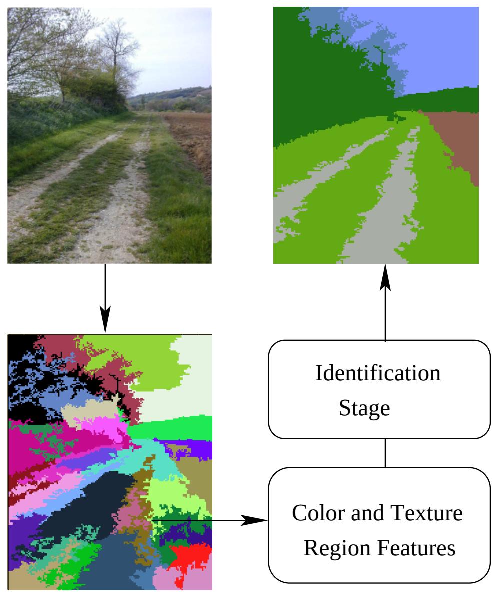

Figure 1: *Scene Modeling sequence*

mer), the weather (sunny or cloudy) or the kind of environment (countryside or urban).

In this paper, this approach is extended to focalize on the road extraction and tracking, either for tarred roads or earthen ones in order to initialize a road tracking function required to control the vehicle motion on the road. The *Road Extraction* function provides the relative road location on a single image, profiting of the global image analysis. The *Road Tracking* function determines the road location from local criteria in a sequence of consecutive images: the tracker initialization and control is performed using informations given by the Road Extraction function for the previous images of the sequence.

In the next section, a brief description of the segmentation algorithm is presented. In the section 3, the process to characterize and classify the regions obtained by the segmentation is described. The tracking procedure is detailed in the section 5. In the last section, some experimental results with road extraction to automatically initialize and to correct the tracking process on image sequence are presented.

2 Color Segmentation

Image segmentation is the process of extracting, the principal connected regions satisfying a uniformity criterion derived from its spectral components. These components are defined in a chosen color space model. The segmentation process could be improved by some additional knowledge about the objects in the scene such as geometrical, textural or optical properties. Therefore *the Color space* selection is a critical parameter in the implementation algorithm [10]. In color segmentation two goals are generally pursued: firstly, the selection of uncorrelated color features [6], and secondly this selection should be as independent as possible to *changes in illumination*. Color segmentation results have been obtained and compared using several color representations. It is possible to obtain good results with only chrominance attributes but they depend on the type of images. Chrominance effects are reduced in images with low saturation. It is for this reason, that the intensity component is kept in the segmentation stage. Over-segmentation errors can occur due to the presence of strong variations in illumination (i.e. shadows). However, over-segmentation is better than to merge non-uniform regions, because this effect can be detected and fixed. The best color segmentation was obtained using the $I_1 I_2 I_3$ representation[10], defined as:

$$\begin{aligned} I_1 &= \tfrac{R+G+B}{3} \\ I_2 &= (R-B) \\ I_3 &= \tfrac{2G-R-B}{2}. \end{aligned} \qquad (1)$$

The components of this space are uncorrelated, so statistically it is the best way for detecting color variations. Then, segmentation algorithm is a combination of two techniques: the thresholding or clustering, and region growing. The method does the grouping in the spatial domain of square cells. Those are associated with the same label defined in an attribute space (i.e. color space). The advantage of this hybrid method is that it allows to achieve growing process independently of the beginning point and the scanning order of the adjacent square cells. The division of the image into square cells provides a first arbitrary partition (an attribute vector is computed for each cell). Several classes are defined by the analysis of the *histograms*, which brings the partition into the attribute space. Thus each square cell in the image is associated with a class. The fusion of the square cells belonging to the same class is done by using an adjacency graph (8-adjacency). Finally, the regions which are smaller than a given threshold are merged to the nearest adjacent region using a color distance criterion.

In the histogram analysis, we have adapted the method suggested by Kittler [11], which assumes that the observations come from mixtures of Gaussian distributions and we use the Kullback information theory to estimate the thresholds. In its most simple way, the observations came from a mixture of two Gaussian distributions having respectively means and variances (μ_1, σ_1) and (μ_2, σ_2) in proportions q_1 and q_2.

Threshold(s) [11] resulting from $q_1, q_2, \mu_1, \sigma_1, \mu_2$ and σ_2 are selected so that they minimize the Kullback divergence $J = \sum_{i=1}^{n} P(i) log(\frac{P(i)}{f(i)})$ from the histogram *P(i)* to the unknown mixture distribution *f*. In our implementation this approach is generalized to detect *N* thresholds, and to obtain the optimal value for *N*.

3 Object characterization and classification

Texture is used to describe the surface of a given object: it is undoubtedly one of the more discriminant characteristics for image interpretation and pattern recognition. Texture is essentially a neighborhood property. Each region obtained in the previous segmentation stage is characterized by a color and texture vector. The texture operators are based on the sum and difference histograms [5]. This type of texture measure is an alternative to the usual co-occurrence matrices used for texture analysis.

This method requires less computation time and less memory storage than conventional spatial texture methods; it provides a probabilistic characterization of the spatial organization of the image, based on neighborhood analysis. Statistical information can be extracted from these histograms and seven texture features are computed from the sum and difference histograms. Since histograms change gradually in function of the view point -distance from the sensor to the scene and occlusions -, these features are rather reliable [6]. Additionally when the color information is available, the statistical means of I_1, I_2 and I_3 are added as attributes to characterize the color in each region. Statistical analysis of numerical values of the I_1 mean and of the mean sum and difference histograms shows an important correlation between the intensity and the texture features.

Once texture and color features have been computed for each region, they constitute an attribute vector in $\mathcal{R}^{10}$: 7 items for texture, 3 for color. These vectors are used to build the training sets required for the classification step. We have found that classification becomes more accurate when the region surfaces increase, which depends on the segmentation algorithm.

The class of a region extracted from the image is obtained by comparing a color/texture vector with a database composed by different classes such as *rock, sky, field, tree, grasp, woods, water, etc*. The Support Vector Machines has been efficiently evaluated with very good results but we have found that it is not very versatile during periodic database updates due to its slow learning convergence with our database. That is why K-Nearest Neighbor rule is the current classification method we use. It has been applied using a metric that takes into account the intrinsic relationships between variables or features. This metric was derived considering the data analysis techniques to reduce space features complexity in order to increase classification efficiency.

Such a reduction process is generally achieved with the principal component analysis, founded on the *Eckart-Young Theorem* which states that for any data matrix *X*, there exists a triple (U, Σ, V) satisfying: $X = U\Sigma V^t$, where X is the normalized data matrix, U, V are the rotation matrices and Σ is a diagonal matrix. If we project the attributes vectors into the new space generated by PCA, we have $x' = (V\Sigma)^t x$ and $y' = (V\Sigma)^t y$. Using this relations into the typical distance definition (Euclidean) we get the proposed metric,

$$\begin{aligned} D^2(x', y') &= (x-y)^t \cdot V \cdot \Sigma\Sigma^t \cdot V^t \cdot (x-y) \\ &= (x-y)^t \cdot R \cdot (x-y) \end{aligned} \quad (2)$$

If we work with normalized data (zero mean and unit variance), we realize that the matrix *R* becomes the covariance matrix and more particularly the *correlation coefficient* matrix . Equation 2 has the advantage that it depends linearly only on the measured data vectors and the covariance matrix of the normalized training data. We can easily see that equation 2 is equivalent to redefine a scalar product considering a weight matrix *R*, $< a, b >_R = a^t \cdot R \cdot b$. This formulation is a particular case of quadratic distances [7] like Mahalanobis distance. The use of the metric shown by the relation 2 removes some of the limitations of the Euclidean metric: (1) It corrects correlation between the different features and, (2) It can provide curved as well as linear decision boundaries.

The covariance (*R*) matrices can be hard to determine accurately, and memory and time requirements grow quadratically rather than linearly with the number of features. These problems may be neglected when only a few features are needed like in our implementation. Indeed we have no need to compute this matrix on-line.

4 Road extraction

Generally the color segmentation methods generate over-segmented images so a fusion phase is needed. In our methodology that phase must merge all the neighbor regions with the same texture and color characteristics (same nature). Its result gives a complete 2D Model of the outdoor natural scene.

a) b) c)

Figure 2: *Road extraction to initialize and update automatically road tracking process. a) Original image, b) 2-D Scene Model c) Road contours extraction*

The user must define a priori how many classes will be interesting for his application. Obviously, this class selection depends on the environment type. In our experiments we have selected seven classes, SKY, FIELD, GRASS, TREES, ROCKS, WATER and WOOD but we have considered the ROAD class as the primordial object to extract.

We should use the additional information to detect some obstacles as trees or rocks and control with this the robot velocity.

	Number of Regions	Correct Detection	Success
TREE	299	221	74 %
SKY	132	132	100%
GRASS	259	201	78 %
ROAD	276	247	89 %

Table 1: *Evaluation of the object classification*

In the table 1 are shown only the most frequently classes found in our experiments (i.e. statistically representatives). Our principal target is a correct road extraction to initialize the tracking phase during the control of robot trajectory. In the figure 2 we present some preliminary results obtained during the road extraction process. In the next section, we present some details on the tracking methodology.

5 Road Tracking

The task of road tracking in a sequence of images on natural environments can be seen as the tracking of a deformable object that grows or shrinks to unpredicted directions. That is the main reason why we have chosen to use snakes or active contours [12] for the tracking stage. This method is based on energy minimization along a curve subject to internal and external forces; these forces are defined by the desired shape and image properties, respectively.

Commonly, energy minimization process is restricted to perpendicular lines to the curve at each control point. However as shapes becomes more complex, after some iterations, the control points could be randomly grouped along the curve and then the final curve could be very different from the desired shape. We do not restrict this process and we solve the shape problem as in [13], where the contour is considered as an electric conductor which is charged by a constant electric charge Q. This charge generates a new force (repulsion) which redistributes the control points along a curve according to its curvature. Another common problem consists in the definition of external force, which is defined usually as the intensity image gradient. But, in complex environments either intensity images gradient nor color image gradients have enough information to attract active contour to the desired edges (figure 3a,b). As a consequence, we used a focalized color gradient (figure 3c,d) defined in normalized color space, with the characteristics found by the characterization stage.

The active contour is initialized by sampling the results of the road extraction described previously. As we used

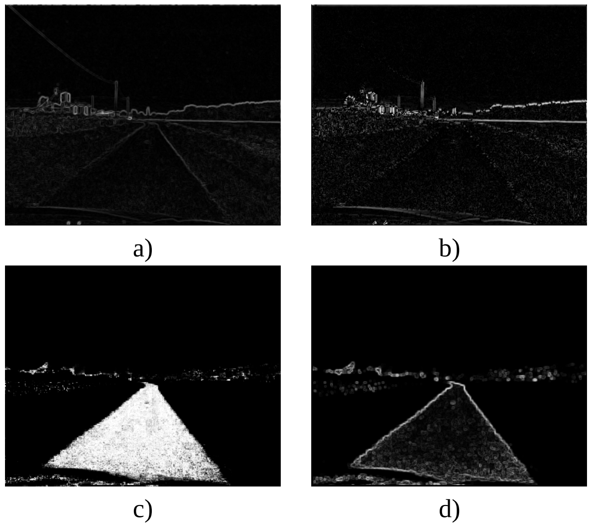

Figure 3: *External potential fields used for active contours. a) Intensity gradient, b) color gradient, c) normalized color selection, d) focalized color gradient*

dynamic programming to solve energy minimization [14], the sampling frequency needs to be a compromise between a few control points (to avoid method complexity) and enough number of control points (to get a good road estimation). The complexity for the dynamic programming method is $O(nm^{k+1})$, where n is the number of control points, m is the number of possible positions for a control point and k is the highest order differential for contour. We accelerate this stage applying it in a multi-scale space.

Snakes reinitialization process is needed because even with the focalized gradient it could be attached to some particular characteristics of the road, as objects in the sides of the roads or shadows. And also because snakes can be deformed in a not suitable way, making very hard the recovering of the desired characteristics. A shape test based in area moments, is performed on snakes over the last iterations in order to determine drastically changes of shapes.

6 Cooperation Extraction-Tracking

In our experiments, a standard 360$\times$288 pixels color image was processed by the color segmentation algorithm in about 100 ms on a Sparc Station 5. The segmented images after the modeling process are shown on the left of the figures 4. On the right we present the tracking evolutions. In some pairs of images we identify that only some images on the left are used as the initialization of the right ones. The red regions on the left images represent the outliers or non-classified elements.

In relation with the region recognition (roads), we have evaluated this phase using several images from an evaluation base. The recognition results using the K-NN method with the proposed distance metric are summarized in the Table 1. The test images were taken in spring scenes with

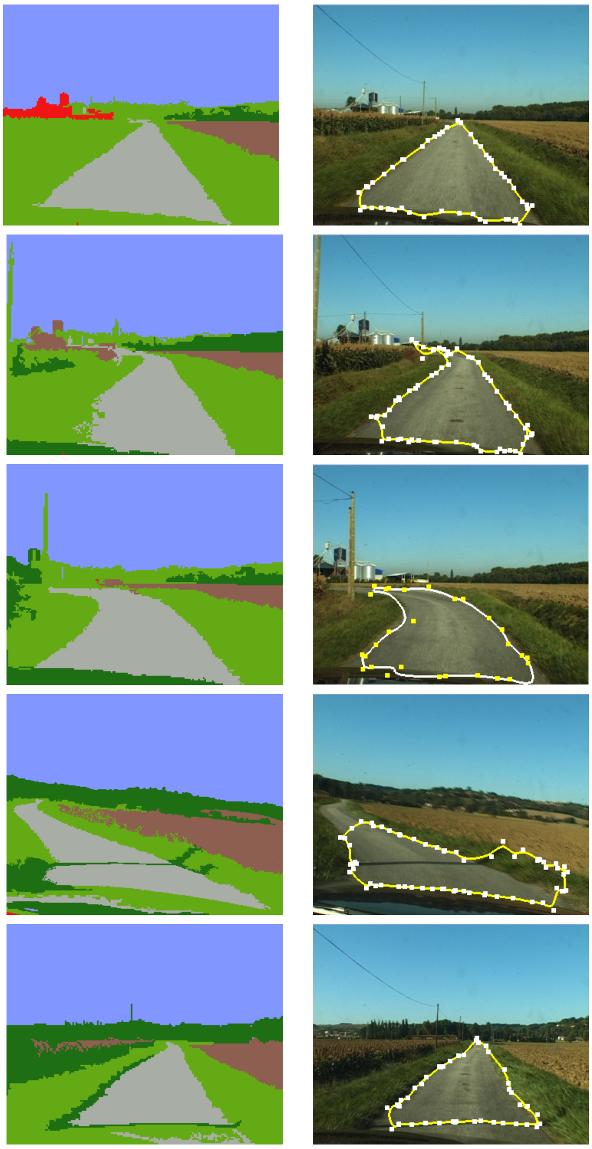

Figure 4: *Road Extraction and tracking for a video sequence.*

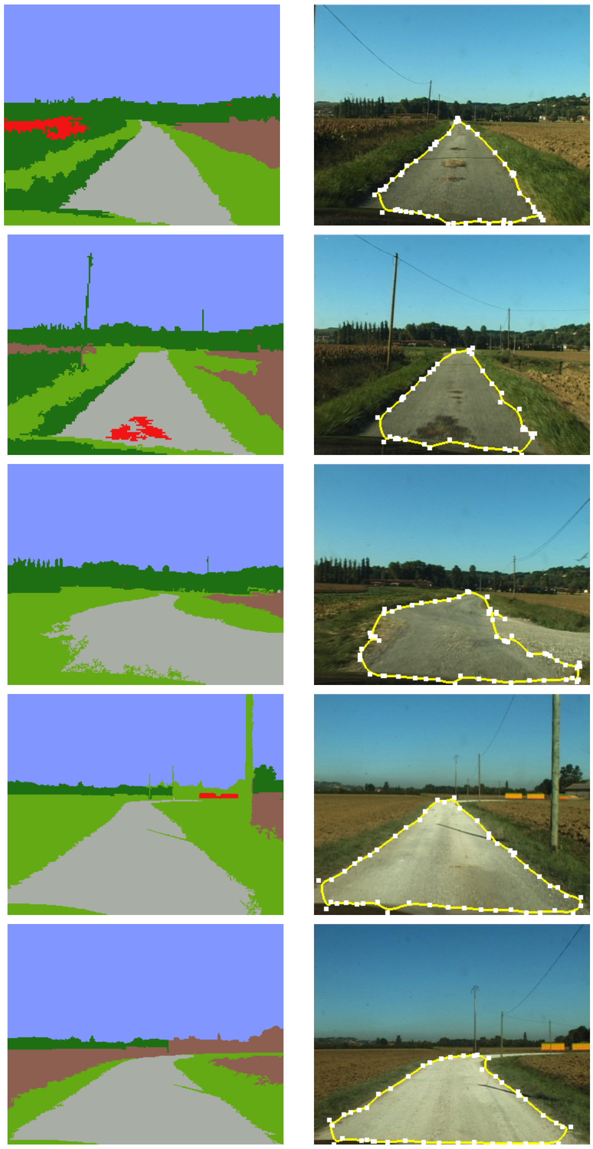

Figure 5: *Road Extraction and tracking for a video sequence.*

different illumination conditions. The total execution time including all the stages of the *Road Extraction* function, is less than 1.0 s per image. We have found a very good recognition rate particularly with the road zones that will be exploited in robot navigation.

The principal inconvenient in the *Road Extraction* function (noted R-LOC), concerns the processing time. When considering a visual-based control loop of a vehicle to navigate on a road, this period is not sufficient: the cooperation with the *Road Traking* function (noted R-TRACK) allows to overcome this problem (see figure 6). We have already analyzed such a cooperation for the navigation of indoor robots using only relative localizations [15], or for outdoor robots during a modeling task [3]. Let us summarize the different advantages of such a cooperation, assuming that R-TRACK can process 10 images when R-LOC can only analyse one:

- initially, on the image I_0, R-LOC finds the road position. This position is sent as the initial road boundary to R-TRACK; the color parameters are also transmitted, so that R-TRACK uses dedicated thresholds for the color gradient.
- every 0.1s, from I_{10}, for every image, R-TRACK provides a new road location, using only local motions of a snake linked to the road boundary. Because it is lo-

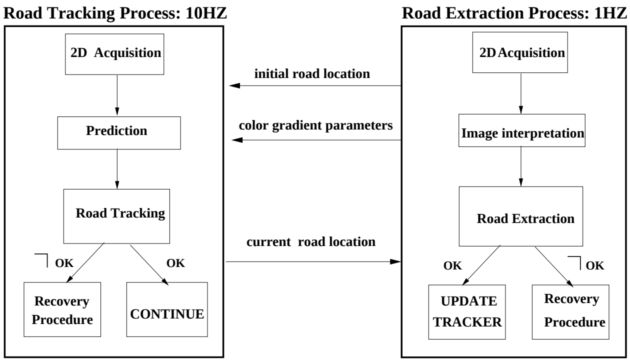

Figure 6: *Cooperation between tracking and extraction functions*

cal, cumulative drifts are possible.

- every 1s., from I_{10}, then for one image over ten, R-LOC gives a new road location, using a global scene interpretation. This global position allows (1) to detect the tracker errors and to reinitialize R-TRACK, and (2) to update the gradient parameters.

A predictive step in R-TRACK, is required in order to compensate the time shift between the two process. At this moment, This prediction requires the knowledge on the speed and the heading of the vehicle, and considers a planar world.

7 Conclusions

This article has presented a method designed to extract and track automatically roads starting from images provided by an on-board color camera. This technique represents some clear advantages such as automatic tracking initialization as well as periodic updates; it provides some additional useful information on the scenes. The correct identification of some other classes besides the roads is a relevant characteristic of our method. This advantage can be used to detect unforeseen objects (obstacles) on the road.

While our preliminary experiments show very promising results, several hurdles lay ahead and are the object of current research. For instance, object classification using its color and texture needs to be more robust to various environment conditions and more computationally efficient. A more efficient scheme is being developed for the fusion with data provided by some other 3D sensors when the texture and color fails. A color/texture classification must be reinforced with explicit shape reasoning in order to correctly characterize the objects on the image.

With our experiments, a rate of almost 90% of good road recognition have been reached, and scene interpretation from one image is generated in about 1 s. Obviously, using only road extraction is not enough to satisfy some real time constraints for some applications. However, the successful road tracking implementation permits us to work in real time. This algorithm generally runs faster than the road extraction algorithm, therefore an adequate synchronization on both tasks is crucial. We aim to apply this method to control the motion of robotized agricultural machines in a network of roads.

References

[1] Yue Wang, Dinggang Shen, Eam Khwang Teoh *Lane detection using spline model* Pattern Recognition Letters, 21:677-689, 2000.

[2] E. Celaya and C. Torras. *Visual Navigation Outdoors: The Argos Project* Intelligent Autonomous Systems 7, 63-67, March 25-27, 2002

[3] R. Murrieta, C. Parra and M. Devy *Visual navigation in natural environments: from range and color data to a landmark-based model.* Autonomous Robots, Vol.13, N2, pp.143-168, September 2002

[4] R. Murrieta, C. Parra, M. Devy, B. Tovar and C. Esteves *Building Multi-Level Models: From Landscapes to Landmarks.* ICRA 2002, Washington, D.C., May 2002.

[5] M. Unser. *Sum and difference histograms for texture classification.* I.E.E.E. on PAMI, 8(1):118-125,1986.

[6] T.S.C. Tan and J. Kittler. *Colour texture analysis using colour histogram.* I.E.E. Proceedings, 141(6):403-412, December 1994.

[7] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner and W. Niblack *Efficient Color Histogram Indexing for Quadratic Form Distance Functions* I.E.E.E. Transactions on PAMI, 17:729-736, 1995.

[8] R.O. Duda, P.E. Hart and David G. Stork. *Pattern Classification and Scene Analysis*. Wiley & sons, 1998.

[9] V. N. Vapnik, S. E. Golowich, and A. J. Smola. *Support vector method for function approximation, regression estimation, and signal processing* Advances in Neural Information Processing Systems, 9:281-287, 1996.

[10] Y. Ohta. *Knowledge-Based Interpretation of Outdoor Natural Color Scenes* Morgan Kaufman Publishers, Inc., Palo Alto, CA, 1985.

[11] J. Kittler and J. Illingworth. *Minimum Error Thresholding* Pattern Recognition, 19(1):41-47, 1986.

[12] M. Kass, A. Wirkin, and D. Terzopoulous, *Snakes: Active Contours Models.* Int'l J. Computer Vision, 1:321-331, 1988.

[13] A. Marin-Hernandez, H.V. Rios-Figueroa, *Eels: Electric Snakes* Computación y Sistemas, 2(2-3):87-94, 1999.

[14] A.A. Amini, T.E. Weymouth, R.C. Jain, *Using Dynamic Programming for Solving Variational Problems in Vision.* I.E.E.E. Transactions on PAMI, 12(9), Sep. 1990.

[15] V.Ayala-Ramirez, J.B.Hayet, F.Lerasle and M.Devy *Visual localization of a mobile robot in indoor environments using planar landmarks* 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2000), Takamatsu (Japon), November 2000.