A Computer Vision System for Autonomous Forklift Vehicles in Industrial Environments[1].

J. Pagès, X. Armangué, J. Salvi, J. Freixenet and J. Martí.

Abstract— **This paper describes a vision system to detect the 3D position of pallets for autonomous forklift vehicles. An accurated image segmentation method based on colour and geometric characteristics of the pallet is proposed. Moreover, the application computes the 3D position and orientation of the pallet and generates the vehicle trajectory to fork it. The system has been tested and experimental results are shown.**

Index Terms— **3D Image Reconstruction, Computer Vision, Feature Selection, Industrial Application and Segmentation.**

I. Introduction

This article describes a complete computer vision system designed to segment and locate the pallets in an industrial environment with the aim of automating forklift vehicles. Usually, autonomous navigation is based on using ultrasonic sensors and odometers. Computer vision has not been used in general applications due to its lack of robustness in the presence of illumination variations and also due to computer complexity. However, computer vision shows a great performance in specific applications. For instance, it has been widely used in industrial inspection [1]. Moreover, some efforts have been done using structured light in order to reduce computer complexity [2]. However, it remains difficult to apply computer vision in industrial navigation.

This paper addresses a specific application: the forklift vehicle (see Fig. 1) has to detect the pallet, which is situated on the floor, and fork it autonomously. The problem has been solved using computer vision. Although it is not the purpose of this article, the vehicle uses also ultrasonic sensor to avoid obstacles and odometers to keep the trajectory.

The article is divided as follows. First, section II deals with the pallet modeling and segmentation. Section III is based on detecting the pallet in the 2D image once it has been segmented. Then, section IV describes the methodology used to obtain the 3D position of the pallet with respect to the vehicle using a single camera attached on its top. Finally, section V describes the method used to generate the trajectory to fork the pallet. The article discusses the experimental results and ends with conclusions.

Fig. 1. The forklift that is being automated

II. Object modelling

A. Introduction

Models constitute representations of the real world, and thus, modelling implies the choice of a suitable set of parameters in order to build those representations. Obviously, the selection of the parameters will affect how well the models fit reality, and this becomes a central issue in any object recognition system. Here, a method for learning such models in training images is proposed.

Object modelling has been designed as a supervised task, where a teacher presents representative examples of objects in training images. Moreover, the presentation of which parts do not belong to the object of interest is also required. Afterwards, the application fulfil a stage of calculating or abstracting in order to find out which features can be used to identify whether an image pixel belongs to the target object. That is, the modelling process is planned as a features selection problem, in which the goal is to find the subset of features that best captures the representation of a given object. In this work, only colour features have been treated, however, the methodology here described is valid for whatever kind of features.

B. Feature selection

In this stage a combination of features which makes possible to identify an object must be found. The solution to this problem is entirely statistical due to the complexity and the amount of data to be treated. If all the combinations of features are tested, a problem of huge execution time has to be faced. Besides, there is another problem, more subjective, which is how to decide which set of features is the most appropriated. With the aim of showing the problem of feature evaluation, a simple example based only on two single features is proposed. Such a simplification has been considered in order to reduce the complexity of

Institute of Informatics and Applications. EIA-PII. University of Girona, Av. Lluis Santaló s/n, 17071 Girona (Spain). Jpages@eia.udg.es

[1] This work has been supported by the CeRTAP consortium of the local government of Catalonia.

graphical representation of spaces with multiple features. In Fig. 2 two couples of colour features of a set of pixel samples from two different objects have been calculated.

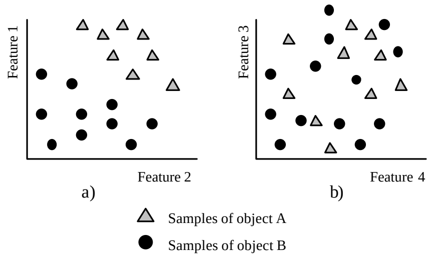

Fig. 2. Object samples distribution based on colour features couples.

The result is a 2D distribution of the given samples which allows us to see whether exists some relationship between them based on their feature values. The sample pixels corresponding to each object form two differentiated clusters, as shown on Fig. 2a, which means that these features can separate the pixels of both objects. Now, a procedure is needed to find out if these two features are more useful than the used in Fig. 2b, where there is no clear relationship between the features and the object samples.

The selection of an optimal subset of features will always rely on a certain evaluation function. Typically, an evaluation function tries to measure the discriminating ability of a feature or a subset of features, to distinguish the different objects [3]. Following the “divide and conquer” paradigm, the problem of feature evaluation has been solved by using a decision tree that separates the samples in a recursive way [4]. Fig. 3 shows such a decision tree operating in a two-dimensional feature space. The decision trees considered in our approach are binary trees with multivariate decision functions, where each node is a binary test represented by a linear function. The Fisher Linear Discriminant Function (LDF) has been demonstrated as a powerful classifier that maximizes the ratio between the inter-class covariance and the intra-class covariance [5]. Each node of the tree attempts to separate, in a set of known instances (the training set), a target (i.e., *pallet*) mapped as △, from non-target instances (*no-pallet*) mapped as ●. However, this is achieved only in a certain ratio, because realistic data is not always linearly separable. The resulting two subsets of samples are again subdivided into two parts using two new calculated linear functions. This process is extended along the binary tree structure, until an appropriate misclassification ratio is achieved. The result is a tree of hyperplane nodes that recursively try to divide the feature space into target and non-target samples. Every leaf has a group of samples with an associated misclassification. So that, all the subgroups of object samples can be found with multiple evaluation functions. To avoid an uncontrolled tree growing, the deep of its branches is usually limited.

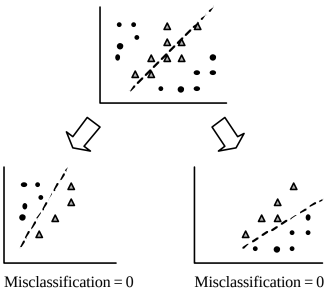

Fig. 3. Binary tree using Fisher recursive evaluation.

Once an evaluation function is chosen to compare the performance of different feature sets, a method to find the best set is required. As it was said before, all possible combinations of features should be tested. However other algorithms are usually used to avoid an exponential execution time. These methods are all based on reducing the search space, using for example AI algorithms like heuristics, genetics, etc. The technique chosen in this work is based on genetics [6]. In summary, it creates an initial population of chromosomes where every one of them represents a different combination of features. Each chromosome is a vector as long as the number of features chosen. Every position of the vector can be zero or one, indicating whether the feature associated to that position is present. Fitness is calculated for every chromosome, which is the misclassification produced for the activated features. A series of iterations are applied based on the initial population, trying to create new chromosomes by using cross techniques, mutating some parts of the existing and removing the worst of them. This process is repeated till a chromosome of certain fitness is generated. The performance of genetics, although its lack of determination, is quite good and the timing calculus is much better than an exhaustive combination method [6].

C. Real time image processing

One of the main traits of industrial applications is the capacity to work in real time. Therefore, the image processing, as part of our application, should be as fast as possible. The evaluation of a decision tree based on Fisher function for each pixel of an image requires a substantial processing time. To reduce it, a look up table is used. The procedure to create this table is simple: every possible combination of RGB colours are evaluated offline using the calculated Fisher tree. Thus, it is known *a priori*, which pixel have to be considered part of the target (pallet) and which must be filtered. All this information is stored in the look up table, that is a simple memory structure of three dimensions (one for each colour component). When an image has to be segmented, it is only necessary to calculate the RGB components of every pixel and consult the look up table to know whether is part of the target object. The use of a look up table is possible because only colour features have been used. If other features like textures are considered, a look up table is useless because of the required neighbourhood operation.

D. *Requirements for a good colour segmentation*

The study of the pallet colour has reflected that there are many grey-scale components in a large variety. This fact implies that a robust segmentation of the pallet is not easy to achieve if there are objects in the environment with important variety of grey components. As the aim of this work is for an industrial application in a controlled environment the adaptation of it to our requirements is truly possible. Therefore, the theoretical warehouse where this application will be used can be adapted so the colours of the walls and floor make easier the segmentation of the pallets.

III. Localization of the pallet

The goal of locating the pallet in a complex environment is not an easy task. Thus, the images are pre-processed to remove the most part of the scene except the pallet, which will appear as a non-uniform region. The pre-processing task consists on a colour-based segmentation. Once the pallet is the only remarkable blob in the image, some techniques are applied to get the positional information and orientation of the pallet. An example of pallet segmentation is shown in Fig. 4. Note the image contains some noise that is further removed. The global shape of the pallet can be observed.

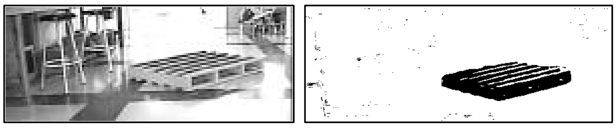

Fig. 4. Pallet colour segmentation sample.

Now, in order to fork the pallet with the forklifter, its 3D position and orientation have to be computed. Moreover, it is necessary to identify the nearest forking side of the pallet. Both tasks are explained in the following sections.

A. *Pallet vertex detection*

In order to find the pallet orientation, the Hough transform is proposed. The Hough transform can be applied to digital images, using the adapted algorithm introduced by Duda and Hart [7]. In summary, this method gathers information about the lines that can be found in the image. Besides, the pallet has a rectangular shape that, at maximum, two of its lateral sides can be observed in a 2D image depending on its orientation. Therefore, the lines of both sides that are in contact with the floor should be detected with Hough transform. Once these lines have been detected the position of three pallet vertexes can be easily found, as shown in Fig. 5. Images shown are in negative.

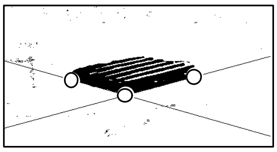

Fig. 5. Significative vertexes of a pallet

The way used to find the vertexes is here briefly described: both lines given by Duda-Hart are explored by using four cursors located at the four line ends. Each cursor explores the line going to the image centre until detects part of the pallet region, which determines one of the four vertexes and finishes its exploration. Obviously, a certain tolerance for this process is recommended in order to skip noise. Then, the exploration consists on using a 1x8 window centred in the line (instead of a 1x1 window) and moving it along the line. This is strongly recommended because of the inherent error of the discrete Hough transform, which can cause that the detected line differs a little from the right one.

It has been observed, during the study, that a single segment of the pallet is only required because the whole pallet can be reconstructed from it, as the pallet model is known.

The main constraint in the line detection process is the interpretation of the information given by Hough transform. The output of the Hough transform is a rectangular matrix in polar coordinates, where each cell represents one of the potential lines that might be present in the image. Every matrix cell contains a number expressing the amount of pixels that have been found in the source image that could be part of the same line. The cell with the highest number determines the largest line in the image. However, the main problem result from finding such a cell that represents the desired line among a non-uniform distribution with local maximums. The complexity of this problem has been described in some articles [8]. In this paper, an intermediate stage is applied to reduce the amount of local maximums. The process is described in the following section.

1) *Image filtering process*

The intermediate process is based on cleaning out the images before applying Hough by using filtering and morphologic operators. The source images in this process have been already segmented so they are binary coded.

The first step consists in a *close* morphological operation to grow the inner pallet region with the goal of producing a single blob. Secondly, an *open* operator is applied with the aim of removing the noise. After most noise is eliminated, a *sobel* filter is applied to enhance the edges of the image. Afterwards, a thining operation is executed to obtain the skeleton of every line. The result of the three steps can be observed in Fig. 6a. As it can be seen, most part of the noise has been removed. However, a lot of lines are still detected if Hough is now applied. As our aim is the detection of a single edge of the pallet that is in contact with floor, the rest of lines should be firstly removed. In order to achieve this goal, another step is applied: every column of the image is explored keeping only the pixel with the highest Y component and removing the rest. The effects of this filter can be seen in Fig. 6b.

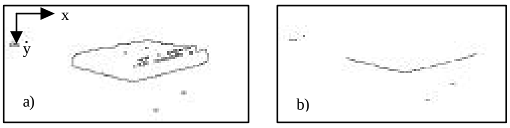

Fig. 6. a) Image filtering; b) Edge pallet detection.

As result of this procedure most part of pixels forming the two searched lines have overcome the filtering. Although some of them are not detected because of noise, the pallet edges still have an important pixel contribution so they are now easily detected by using Hough transform.

2) Maximum searching in Hough matrix

As it has been discussed in previous sections, a single pallet edge is required. It has been seen that, at least, one of the visible sides has a positive angle in polar coordinates, related to a coordinate system located at the left-bottom corner of the image. This could be used to restrict the scanning area of the hough matrix. However, both lines are searched in order to achive better results. This decission is due to the fact that some discrete lines of certain slopes have not a clear linear appearance. The scanning areas for maximums in the Hough matrix are determined by ρ = [1920, 3839], which represents the possible distances between the lines and the origin, and θ = ([159, 312], [316, 469] and [473,626]), representing the line angles. Fig. 7 shows the scanning areas and the correspondence between matrix indexes and angles. These ranges have been found with the analysis of a set of images where all boundary slopes of each side appeared. All this process could be easily automatized for any linear shape (not only rectangular) giving the number of sides and angles among them. A security thin blank space has been defined between each consecutive area with the aim of avoiding the discontinuities given by upright slopes. Finally, the maximum search starts.

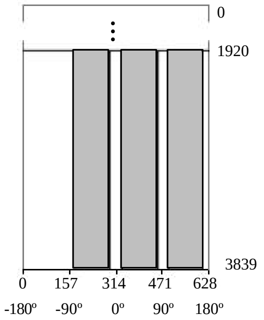

Fig. 7. Scanning areas of hough matrix of a 768x576 image.

An exhaustive scanning is done all over the area. Maximums do rarely appear as a single cell with a maximum value, but as a set of adjacent cells with the same value. Therefore, for each region of adjacent cells with the same maximum value, the gravity centre is calculated and stored as a local maximum. However, only the maximums that overcome a fixed threshold are considered. This minimum value has been fixed to 20 (what it means that lines with less than 20 pixels are not considered). Finally, the local maximum with the highest value is selected as the largest line in the image. Thus, a single edge of both pallet sides is presumed to be detected.

IV. 3D RECONSTRUCTION

A. Coplanar Tsai calibration

When the segment of one of the sides of the pallet, which are in contact with floor, has been detected in the image, its 3D real position in the world has to be computed in order to get the 3D position and orientation of the pallet. The method chosen is a transformation algorithm to get the 3D coordinates of 2D points from a single image. A camera calibration algorithm is required to calculate the relationship between the 2D points and the corresponding 3D optical rays. The Coplanar Tsai calibration method has been used [9]. This algorithm presumes that all the reconstructing points are placed in a plane, which coincides with the calibrating plane. Therefore, this method only gives accurate results under these conditions. The problem treated in this work can easily accomplish this requirement because the pallet is always on the floor and it can be used as the calibrating plane. In order to apply the calibration algorithm a 3D world coordinate system is required. Although it can be virtually situated everywhere, it is interesting to set it in the forklift adequately. Moreover, the camera is located as far as possible from the floor and on the forklift.

A set of 3D sample points (minimum 6, due to the number of variables involved) from the floor must be measured referred to the origin of the world coordinate system, trying to spread them to cover all the image scope, where the pallet can be located. The larger is the number of sample points the more accurate the transformation between 2D and 3D will be. Then, the 2D correspondences of every 3D point is measured in the image. Once the set of 3D points and their 2D correspondences are known, the camera can be calibrated by using the co-planar algorithm proposed by Tsai [9]. When the camera is calibrated, two main transformations can be applied: a) given a 3D point, its 2D projection on the image plane can be predicted; b) given a 2D point, its optical ray starting from the optical center of the image and passing through the 2D projection and the 3D point can be computed. The 3D position of every point lying on the floor can be computed intersecting the optical ray with such a floor. The, the position of the pallet related to the forklift (world coordinates) can be easily computed.

B. 3D Reconstruction of the pallet

As result of the 2D-edge localization process, both end points of the main segment in the image are known. Besides, it is known that both points lie on the floor, so the equation of the floor with respects to the world coordinate system is also known. This information is enough to calculate the rest of pallet points. Once we have the 3D coordinates of the detected points, the distance between them can be calculated. This distance represents the longitude of one of the pallet sides. The average dimensions of the pallets must be known and stored as part of the pallet model. The pallets used in our application have the largest side in the forking part, which is about 1 meter long. The shorter side is about 0.8 m. Therefore, analyzing computed distances, the forking side of the pallet can be detected. The rest of the pallet can be reconstructed by simple scaling and rotating adequately the edges of the pallet.

V. TRAJECTORY CALCULATION

Once the 3D information of the pallet relative to the vehicle is obtained, a trajectory strategy to fork the pallet has to be appointed. The simplest trajectory consists on dividing it into two linear segments. The goal is to fork the pallet from the forking side close to the vehicle (Note every pallet has two forking sides). In Fig. 8 a schema of this kind of trajectory is shown [10].

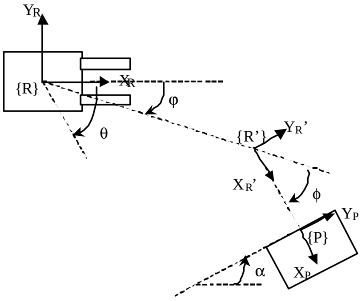

Fig. 8. Linear trajectory to fork the pallet.

In order to achieve such a trajectory, two couples of rotation-displacement operations are defined. The rotations are defined around de Z-axis, which is orthogonal to the floor, following the rules of a counter-clockwise system. The displacements are defined along the X-axis of the vehicle co-ordinate system. Then, three intermediate steps are considered to achieve the trajectory. The first step {R} is the position of the vehicle once the pallet has been detected. The second step {R'} represents the vehicle position when the first rotation and displacement and the second rotation have been made. Last step {P} represents the position and orientation of the vehicle when the pallet has been forked. Both rotations are identified by the angles j and f. The angle q represents the rotation that the vehicle might do in the initial step to get orthogonal to the target side of the pallet. The relationship of these angles is:

$$q = j + f$$

where θ can be calculated with the following rules:

$$a > 0 \to q = a - p/2$$
$$a < 0 \to q = a + p/2$$

where $\tan(a)$ is the slope of the target side referred to X_R. α is expressed in the range of [π/2, -π/2]. The intermediate point of the trajectory {R'} is computed so that X_R' is pointing to the target and located at a predefined distance with respect to it. Thus, the second segment displacement is always constant. In order to calculate the motion parameters, the transformation matrices between the three co-ordinate systems are used:

$$^{R}A_{R'} = {}^{R}A_P \cdot ({}^{R}A_P)^{-1}$$

Where $^{(*)}A_{(\#)}$ defines the co-ordinate system (#) referred with respect to (*). Observing Fig. 8 matrix $^{R}A_{R'}$ can be derived in the following way:

$$^{R}A_{R'} = \begin{pmatrix} \cos q & -\operatorname{sen} q & 0 & ^{R}Px \\ \operatorname{sen} q & \cos q & 0 & ^{R}Py \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} =$$

$$= \begin{pmatrix} \cos q & -\operatorname{sen} q & 0 & -d\cos q + {}^{R}P_X \\ \operatorname{sen} q & \cos q & 0 & -d \operatorname{sen} q + {}^{R}P_Y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Where $({}^{R}P_X, {}^{R}P_Y)$ are the co-ordinates of the central point of the target side of the pallet (origin of {P}) referred to {R}, d is the predefined distance between the origin of {P} and the origin of {R'}. Therefore, the co-ordinates of the intermediate point of the trajectory (origin of {R'}) are

$$(-d\cdot\cos q + {}^{R}Px,\ -d\cdot \operatorname{sen} q + {}^{R}Py)$$

Finally, the rest of parameters can be computed:

$$j = \text{ATAN2}(-d\cdot \operatorname{sen} q + {}^{R}Py\ /\ -d\cdot\cos q + {}^{R}Px) \qquad f = q - j$$

$$t1 = \sqrt{(-d\cdot\cos q + {}^{R}P_x)^2 + (-d\cdot \operatorname{sen} q + {}^{R}P_y)^2} \qquad t2 = d$$

where *t1* and *t2* are the two displacements along the X_R and $X_{R'}$ axis of the vehicle, respectively. The sequence of transformations to achieve the trajectory is: rotation φ, displacement *t1*, rotation ϕ and displacement *t2*.

VI. EXPERIMENTAL RESULTS

A Visual C++ application has been developed to study the performance of the whole process. All the different steps have been implemented from colour segmentation till trajectory calculation. With the aim of reducing the design time, the forklift has been implemented in a mobile robot model Pioneer2® of ActivMedia™, and a camera has been assembled on its top. This article describes only a part of a huge project financed by the local government, which integrates up to 7 research groups. Due to the difficulty to have the autonomous forklift all the time available in our university, as a first approach we have simulated its behavior in such a mobile robot.

A set of sample situations (varying the position and orientation of the pallet with respect to the vehicle) have been tested in the laboratory, where the environmental conditions have been adapted to simulate an industrial warehouse. Walls and floor colours have been chosen (white for the walls and green for the floor due to their easy segmentation), and the illumination system is light controlled. Such conditions have been discussed and accepted by the enterprises supporting the project and considered as really low-cost measures.

The study has addressed that the discrepancy between defined and real intermediate points depends on an inherent error produced by the control architecture of the robot, but also an error generated in the pallet localization step. The first component of the error has not been treated because it was not the aim of this work (other research groups are working in control architectures and sensor data fusion) and it is assumed to be small. The error due to the imprecision of the pallet localization method has been studied. An obvious result is that the larger is the distance between the pallet and the robot, the higher is the error between predicted and real pallet position.

Table I shows the real and the calculated world coordinates of both points of the detected side in a set of images, as well as the real and calculated longitude of the side, besides the errors produced by the system (all in mm).

TABLE I
PALLET 3D LOCALIZATION RESULTS

Real points				**Calculated points**			
X1	*Y1*	*X2*	*Y2*	*X1*	*Y1*	*X2*	*Y2*
1909	1105	2254	169	1916	1084	2335	180
2049	-40	1909	-804	2078	-33	1928	-801
2311	0	2533	-748	2373	-1	2560	-747
2408	703	2136	-28	2409	702	2115	-59
2522	166	2311	-804	2565	178	2353	-809
2713	402	2755	-595	2722	392	2766	-577
2713	402	3087	-502	2742	399	3135	-540
3153	180	3014	-804	3093	186	3033	-810
3416	378	3416	-402	3467	376	3460	-410
3571	763	3416	0	3677	765	3439	-34
4689	374	4220	-506	4333	250	4521	-538

Errors				**Side longitude**	
X1	*Y1*	*X2*	*Y2*	*Real*	*Calculated*
-7	21	-81	-11	1000	996
-29	-7	-19	-3	800	808
-62	1	-27	-1	800	793
-1	1	21	31	800	842
-43	-12	-42	5	1000	1009
-9	10	-11	-18	1000	969
-29	3	-48	38	1000	1018
60	-6	-19	6	1000	997
-51	2	-44	8	800	812
-106	-2	-23	34	800	860
356	124	-301	77	1000	810

This table is only a summary of a bigger test where 50 different pallet orientation and positions were proved. If there is not an important loss of the pallet geometry in segmentation, the discrepancy between real and calculated points has an average of 31.5mm along the x-axis and 12.5mm along the y-axis, with a standard deviation of 21.7mm and 10.1mm, respectively (see Fig. 8). The discrepancy has been computed when the pallet is close to the vehicle (up to 3 m). The error of the x component is larger because the range of distances along this axis is also bigger. Note that the difference between both pallet sides is 200 mm, so the system can keep an accurate identification of the forking side of the pallet. Besides, if the segmentation process erodes the pallet edges, the forking side identification decreases considerably. However, this occurs only when the pallet is far from the vehicle. Then, the strategy consists in approaching the vehicle to the pallet and, then, recalculating the trajectory when the pallet is close to 3 meters. The idea of this strategy is that the orientation of the pallet is not important until it is separated less than a certain distance from the vehicle, while if the distance is larger, the application only tries to calculate the mass center of the pallet. Before calculating the mass center, the statistical technique of the median is applied to remove possible remaining noise. The mass centre is used to calculate the approaching trajectory. Once the distance between pallet and vehicle is less than 3m, the detection process is used. If the dimensions of the detected pallet do not fit the pallet model, the processed image is rejected and a new one is grabbed, introducing a sort of feedback.

VII. CONCLUSIONS

This article describes a vision system useful for autonomous forklift vehicles in industrial environments. The application is based on accurate pallet segmentation based on its colour and geometric characteristics. A trajectory to fork the pallet is calculated based on its 3D position and orientation.

The pallet segmentation is one of the key points of the process. There are more powerful algorithms of feature selection and better discrimination functions than the linear ones that have been used. Although the results obtained with linear tools have been quite accurate, we are interested in testing other segmentation methods, like textures, that could improve the pallet segmentation when it is far from the vehicle. Another aspect related with the vision problem is the scene illumination. Moreover, the system was tested in light uncontrolled environments with the aim of observing its robustness, but the obtained results were really discouraging. Once more, a good illumination system is really required to obtain accurate results.

Related to 3D computation, other calibration algorithms should be tested to survey their accuracy. Moreover, the trajectory proposed is robust but quite simple so that some other more complex, like cubic splines, should be studied.

Finally, we are also thinking in a continuous vision feedback system to adapt the trajectory dynamically. Thus, erroneous pallet detection could be isolated. In summary, a first approximation to the problem has been developed. Hereafter, the aim is to develop a robust system based on the methods here described.

VIII. REFERENCES

[1] B.G. Batchelor, F.M. Waltz, *Machine Vision Systems Integration in Industry*, The International Society for Optical Engineering, 1991, vol. 1386.

[2] J.Salvi, E. Mouaddib and J. Batlle, “An overview of the advantages and constraints of coded pattern projection techniques for autonomous navigation,” *IEEE Int. Conf. on Intelligent Robots and Systems*, sept. 1997, vol. III, pp. 1264-1271.

[3] M. Dash and H. Liu, “Feature selection for classification”, *Intelligent data analysis*, vol. 1(3), 1997.

[4] J.Freixenet, J.Martí, X.Cufí and X.Lladó, “Use of decision trees in colour feature selection. Application to object recognition in outdoor scenes,” in *Proc. of the IEEE Int. Conference on Image Processing (ICIP)*, Sept. 2000, vol. 1 pp. 800-803.

[5] G.J. McLachlan, *Discriminant analysis and statistical pattern recognition*, Wiley-InterScience Publication, 1992.

[6] X.Lladó,J.Martí,J.Freixenet and X.Muñoz, “Features selection using genetic algorithms,” *3rd Catalan Conference on Artificial Intelligence*, Oct. 2000, pp. 152-156.

[7] W.K.Pratt, *Digital Image Processing*, 2on Edition. Wiley-InterScience Publication, 1991.

[8] R.C. Aggarwal, R.K. Shevgaonkar and S.C Sahasrabudhe, “A fresh look at the Hough transform,” *Pattern Recognition Letters*, vol. 17, pp 1065-1068, 1996.

[9] R.Y. Tsai. “A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses,” *IEEE Journal of Robotics and Automation*, aug. 1987, vol. 3, no.4, pp. 323-344.

[10] O. Faugeras, Three-Dimensional Computer Vision: a Geometric Viewpoint, The MIT Press, 1993.