Computer Algebra Algorithms Applied to Computer Vision in a Parking Management System.

R.J. López Sastre, P. Gil Jiménez, F.J. Acevedo, S. Maldonado Bascón
Department of Signal Theory and Communications
University of Alcalá
A-2 Km 33.600, Campus Universitario, 28805, Alcalá de Henares, Spain
Email: {robertoj.lopez, pedro.gil, javier.acevedo, saturnino.maldonado}@uah.es
www2.uah.es/teose

Abstract— From this paper, we propose a novel methodology to compute a 2D Homography applying some algorithms of computer algebra. We consider the classical problem of solving (exactly) a linear system of algebraic equations, and we suggest a new algorithm for computer vision, based on homomorphism methods over $\mathbb{Z}$, to solve a system of equations necessary to achieve a 3×3 matrix H which lets us to compute the projective transformation which translates coordinates between points in different planes. From this work, we want to show that it is possible to apply a symbolic approach to some crucial issues of computer vision, moreover of the numerical methodology, in order to reduce the complexity of some algorithms, and to eliminate the problems associated with loss of precision and normalization. We test our technique in a real situation: a parking management system, which creates a pseudo-top-view of a parking area to determine if there are free parking lots or not.

I. Introduction

The main objective of this paper is to link some computer algebra algorithms to computer vision, in order to reject some numerical approaches which are more complex and inefficient. Holt et al. presented some algebraic methods for image processing and computer vision in [1]: Bézout numbers, Gröbner bases and homotopy methods. From this paper we propose a new computer algebra resource: homomorphisms methods for solving systems of equations, which are widely described in [2], [3] and [4]. There are numerical and symbolic methods to compute all solutions of a given zero-dimensional polynomial system, and every approach has its advantages and disadvantages. An hybrid method could be a great solution for the problem, as [5] shows. To study the complexity of each type of approach, numerical or symbolic, as it is shown in [6], will be an important task if we want to decide which one that is better-conditioned.

Important problems in computer vision can be formulated as the solution of a system of equations. To be concrete, the problem of computing the 3×3 Homography matrix H, which lets us to compute the projective transformation which translates coordinates between points in different planes. In the literature reviewed ([7], [8] and [9]), all authors use numerical algorithms to determine the problem. Ideally, to solve this system we need four correspondences of points, but in these numerical approaches the more the number of pairs of points, the more exactly is the solution. We aim for a new approach

which uses a computer to perform a mathematical computation symbolically. This computation must solve a linear system of equations, and in our symbolic approach we only need four correspondences of points in order to achieve the exact solution of the system.

From this mathematical work, we propose a computer vision application for a parking management system in which we use the symbolic approach proposed to construct a pseudo-top-view of a parking area, computing the homography by means of the homomorphic method for solving a linear system of equations.

This paper is organized as follows. In section II we describe the 2D homography computation process, including a comparative analysis between the symbolic and the numerical approach. Section III shows a description of the developed parking management system, and in section IV we have described the texture feature extraction used in the parking management system. Finally, sections V and VI contain the experimental results and the conclusions, respectively.

II. 2D Homography Computation

A. Problem Under Consideration: 2D Homography

We can define a point $\mathbf{x}$ of an image I in homogeneous coordinates as follows $\mathbf{x} = (x, y, z)^T$. Given a set of points $\mathbf{x}_i$ in I, and the corresponding set of points $\mathbf{x}'_i$ in image J, we compute the projective transformation which transforms each $\mathbf{x}_i$ in $\mathbf{x}'_i$. Each image, I and J, is considered as a projective plane $\mathbb{P}^2$. A 2D Homography is a 2D projective transformation, which is defined by the equation

$$\mathbf{x}'_i = H\mathbf{x}_i \tag{1}$$

, where

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \tag{2}$$

The equation (1) may be expressed in terms of vector cross product as $\mathbf{x}'_i \times H\mathbf{x}_i = \mathbf{0}$. If the j-th row of the matrix H is denoted by $\mathbf{h}^{jT}$, then we can write

$$H\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1T}\mathbf{x}_i \\ \mathbf{h}^{2T}\mathbf{x}_i \\ \mathbf{h}^{3T}\mathbf{x}_i \end{pmatrix} \tag{3}$$

If $\mathbf{x}'_i = (x'_i, y'_i, z'_i)^T$, then the cross product may be given as

$$\mathbf{x}'_i \times H\mathbf{x}_i = \begin{pmatrix} y'_i\mathbf{h}^{3T}\mathbf{x}_i - z'_i\mathbf{h}^{2T}\mathbf{x}_i \\ z'_i\mathbf{h}^{1T}\mathbf{x}_i - x'_i\mathbf{h}^{3T}\mathbf{x}_i \\ x'_i\mathbf{h}^{2T}\mathbf{x}_i - y'_i\mathbf{h}^{1T}\mathbf{x}_i \end{pmatrix} = \mathbf{0} \quad (4)$$

The previous equation can be written as

$$\begin{pmatrix} \mathbf{0}^T & -z'_i\mathbf{x}_i^T & y'_i\mathbf{x}_i^T \\ z'_i\mathbf{x}_i^T & \mathbf{0}^T & x'_i\mathbf{x}_i^T \\ -y'_i\mathbf{x}_i^T & x'_i\mathbf{x}_i^T & \mathbf{0}^T \end{pmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0} \quad (5)$$

This last equation has the form $A_i\mathbf{h} = \mathbf{0}$. Although there are three equations in (5), only two of them are linearly independent, hence, this set of equations become

$$\begin{pmatrix} \mathbf{0}^T & -z'_i\mathbf{x}_i^T & y'_i\mathbf{x}_i^T \\ z'_i\mathbf{x}_i^T & \mathbf{0}^T & x'_i\mathbf{x}_i^T \end{pmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0} \quad (6)$$

B. Symbolic Approach

Our symbolic approach is based on a new computer algebra resource in computer vision: homomorphisms methods for solving systems of equations over $\mathbb{Z}$. The equation (6) describes a system of linear equations with coefficients $\in \mathbb{Q}$. From a theoretic point of view, the problem can be solved using symbolic computation, for example Cramer, which has complexity $n!$, where n is the number of equations. In order to reduce this complexity, we propose the homomorphism method over $\mathbb{Z}$, also called modular method.

Let P be a problem whose input and output are from domain $\mathbb{D}$. The basic idea of the modular method in computer algebra consists of applying homomorphisms $\varphi_1, ..., \varphi_n$ to $\mathbb{D}$, $\varphi_i : \mathbb{D} \mapsto \mathbb{D}_i$, such that the corresponding problem P_i can be more easily solved in $\mathbb{D}_i$, and the solutions of the problems in the image domains can be combined to yield the solution of the original problem P in $\mathbb{D}$. We can combine the partial solutions by the CRA (Chinese Remainder Algorithm), which is described in [2]. Is every homomorphism φ_i valid in this process?. How many homomorphism do we need?. To answer these questions we must complete a qualitative and quantitative analysis, respectively.

In our case, let being the problem $P \equiv$ "*Given a regular matrix $A = (a_{i,j})_{1 \le i,j \le n}$ over $\mathbb{D}$ and $\mathbf{b} = (b_{1,n+1}, ..., b_{n,n+1})^T \in \mathbb{Z}^n$, solve the system of equations defined by $A\mathbf{x} = \mathbf{b}$*". However, the solution to the 2D Homography problem is, in general, an element which belongs to $\mathbb{Q}^n$, and not to $\mathbb{Z}^n$. Under this situation, we propose the following method to achieve the solution. The problem P is equivalent to solve the system of equations $A\mathbf{y} = det(A)\mathbf{b}$, where now, the solution belongs to $\mathbb{Z}^n$, and then we only must divide $\mathbf{y}$ by $det(A)$. Therefore, we can reconsider the problem as follows: "*Given a regular matrix $A = (a_{i,j})_{1 \le i,j \le n}$ over $\mathbb{Z}$ and $\mathbf{b} = (b_{1,n+1}, ..., b_{n,n+1})^T \in \mathbb{Z}^n$, determine the vector $\mathbf{y}$, which solves the system $A\mathbf{y} = det(A)\mathbf{b}$*".

In the qualitative analysis we must conclude if every homomorphism is valid. For our concrete problem, the primes m, which do not solve the system, are those primes which make that $det(\varphi_m(A)) = 0$. In our case, these primes are the divisors of $det(A)$, and the number of divisors of $det(A)$ is lower than $log_2(n!) + nlog_2(n) + 2$. Then, the strategy will be to take a prime m, and if we discover that $det(\varphi_m(A)) = 0$, we will discard m and we will take another one.

In the quantitative analysis, we want to determine the number of homomorphisms that we need to find the solution. Let be $c \in \mathbb{N}$ an upper bound for the elements of the amplied matrix $|(A|\mathbf{b})|$. Then, each solution β_i, using Cramer, is the determinant of a submatrix of $|(A|\mathbf{b})|$, and we can use the bound of Hadamard, which determines that $|\beta_i| \le n^n c^{2n}$, where n is the number of rows of matrix A, to limit the number of primes to use.

From this point, we can present the steps of the proposed algorithm for solving a 2D Homography using the modular method:

- **1** Determine four correspondences between points $\mathbf{x}_i$ and $\mathbf{x}'_i$. Then, construct the system of equations defined by (6), to solve the system $A\mathbf{x} = \mathbf{b}$ with 8 unknowns.
- **2** Compute the bound given by the equation

$$CT = n! \cdot max \{|a_{i,j}| 1 \le i \le n, 1 \le j \le n+1\} \quad (7)$$

- **3** Initialization: $p = 1$; $P = 1$; $\beta_{1,p} = 0$; ...; $\beta_{n,p} = 0$; $\beta_{n+1,p} = 0$.
- **4** *while $P \le 2CT$ do*
  - **4.1** Take a prime m of simple precision.
  - **4.2** Reduction: $A^m = \varphi_m(A)$; $\mathbf{b}^m = \varphi_m(b)$; $A_a = (A^m|\mathbf{b}^m)$.
  - **4.3** Solve over $\mathbb{Z}_m$
    * **4.3.1** Apply Gaussian Elimination over $\mathbb{Z}_m$ to matrix A_a, to obtain matrix A_{ge}.
    * **4.3.2** $\beta_{n+1,p} = det(A_{ge})$. If $\beta_{n+1,p} = 0$ *return to step* **4.1**.
    * **4.3.3** Solve the system over $\mathbb{Z}_m$. Let be the solution of the system $(\beta_{1,p}, ..., \beta_{n,p})^T$
    * **4.3.4** Multiply this solution by the determinant $\beta_{n+1,p}$.
  - **4.4** *for i from 1 to $n+1$ do*

$$\beta_{i,p} = CRA([\beta_{i,p}, \beta i, m], [P, m]) \quad (8)$$

  - **4.5** $p = m; P = Pm;$.
- **5** Return the definitive solution $\left(\frac{\beta_{1,p}}{\beta_{n+1,p}}, ..., \frac{\beta_{n,p}}{\beta_{n+1,p}}\right)^T$

We have showed the symbolic algorithm used to solve a system of equations. For this work, we used the software *Maple* to achieve results.

C. Numerical Approach

This numerical approach is based on the algorithm called DLT (Direct Linear Transformation), which is widely developed in [7], and in this section we only present an outline of itself. At firs, we need to establish more than 4 correspondences between points, in order to reach a better solution, with 8 correspondences it is enough. In a second step, we have to make a normalization of $\mathbf{x}_i$ and $\mathbf{x}'_i$. We compute a transformation matrix T (which consists of a translation and

a scaling), which takes points $\mathbf{x}_i$ to a new set of points $\tilde{\mathbf{x}}_i$ such that the centroid of the new set is the coordinate origin $(0,0)^T$, and the average distance of all the points to the origin is equal to $\sqrt{2}$. Also, we compute a similar transformation T' between the set of points $\mathbf{x}'_i$ and $\tilde{\mathbf{x}}'_i$. T and T' are upper triangular matrices which are defined as follows

$$T = \begin{pmatrix} \frac{\sqrt{2}}{p} & 0 & -\frac{\sqrt{2}}{s_x}m_x \\ 0 & \frac{\sqrt{2}}{p} & -\frac{\sqrt{2}}{s_y}m_y \\ 0 & 0 & 1 \end{pmatrix} \quad (9)$$

, where: $\mathbf{m}$ is a vector of 2×1 defined as $\mathbf{m} = (m_x, m_y)^T$, which contains the mean value of the coordinates x and y of the vectors $\mathbf{x}_i$, $\mathbf{s}$ is a vector of 2×1 defined as $\mathbf{s} = (s_x, s_y)^T$, with the standard deviation of the coordinates x and y of a set of points $\mathbf{x}_i$, and p is the mean value of the components of vector $\mathbf{s}$. Then, we can compute the transformation between these two sets of points as follows

$$\tilde{\mathbf{x}}_i = T\mathbf{x}_i \quad (10)$$

From this point, we have two transformation matrix: T and T'. For each new correspondence between the set of points $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}'_i$ we compute the matrix A_i defined in (6). Then we assemble all the 2×9 matrices into a single matrix A. We seek the non-zero solution for $\mathbf{h}$. Matrix A has rank 8, hence it has a 1-dimensional null-space which provides the exact solution for $\mathbf{h}$. This would be the case if the measurement of image coordinates were exact, but in others situations the established correspondences between the sets of points are inexact. Instead of needing an exact solution (the null-space), we can find an approximate solution which minimizes a determined cost function. To solve the system of equations under this situation, we have to obtain the SVD (Singular Value Decomposition) of matrix A. The cost function to minimize is the quotient $||A\mathbf{h}||/||\mathbf{h}||$, that is the same that to minimize the norm $||A\mathbf{h}||$. The unit singular vector corresponding to the smallest singular value is the solution $\mathbf{h}$. Specifically, if $A = UDV^T$ with D diagonal and with positive diagonal entries, arranged in descending order down the diagonal, then $\mathbf{h}$ is the last column of V.

Let $\tilde{\mathbf{h}}$ be the obtained solution, and $\tilde{H}$ the compounded matrix. In order to obtain the matrix H we have to do the following denormalization

$$H = (T')^{-1}\tilde{H}T \quad (11)$$

$$H = \left(\frac{1}{h_9}\right)H \quad (12)$$

D. Experimental Results

We have presented an alternative approach to compute a 2D Homography based on computer algebra. A symbolic algorithm offers the exact solution of a problem, whereas the numerical one returns an approximation. There are advantages and disadvantages in each kind of approach. First of all, in the developed comparative analysis, we have to take into account that DLT algorithm needs transformations of normalization,

because its result, for computing homographies, depends on the coordinate frame in which points are expressed. This normalization, consisting of a translation and scaling of image coordinates, will make that the algorithm be invariant with respect to arbitrary choices of scale and coordinates origin. Whereas, the symbolic methodology proposed from this paper is invariant, without the necessity of any kind of normalization process which adds more complexity to the algorithm, and problems of loss of precision.

To compare both approaches, we have created a real-top-view of a figure over a plane. Then, we have matched 4 correspondences (8 for the numerical approach) between points, in order to compute the 2D Homography. Finally, we have measured the similarity between the real-top-view and the generated top-views, computing the SSD (Sum of Squared Differences). Figure 1 shows the picture generated and the real top-view, and in figure 2 we can see that both algorithms presents a quite similar solution, but the SSD obtained is of 9.876.292 for the symbolic approach, and for the numerical method is of 10.003.448.
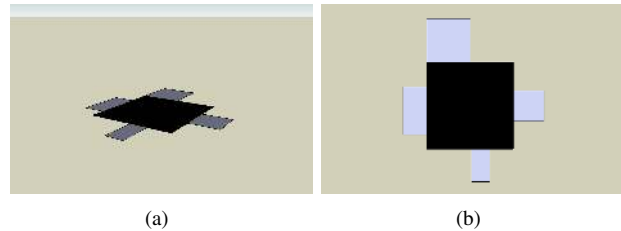

(a)   (b)

Fig. 1. This figure shows the created model: a) is the projective view, and b) is the real-top-view.
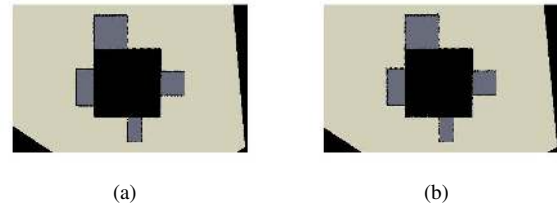

(a)   (b)

Fig. 2. This figure shows the pseudo-top-views created, a) with the numeric approach, and b) with the symbolic methodology.

III. APPLICATION: PARKING MANAGEMENT SYSTEM

To apply the mathematical work proposed from this paper, we have developed a parking management system. In this kind of systems, based on computer vision, there are some problems derived from the use of cameras. In [10] the authors present a work for determining car-park occupancy from single images, but a full visibility from the camera position to the whole car-park has to be ensured. Others works, as it is shown in [11], assume that it is possible to install a camera high enough to implement a parking condition discrimination system based on a real-top-view of the parking area. We understand that all of described situations impede the installation of these management systems in a lot of real places, as in underground

parkings. To solve these problems, our work compounds a pseudo-top-view of the parking area. It has to be ensured that our surveillance camera has a good visibility of the parking area, in order to manage more parking lots. The main objective of this image compounding task is to achieve a reference image where it can be done easily the identification and delimitation of each parking lot, and the extraction of the texture feature characteristics.

We present a solution based on a single camera. This first approach offers the following administration tasks: surveillance of the parking area, determination of free parking lots, determination of parked vehicles in forbidden zones and average time of parking. Every minute, we capture an image of a parking area from a surveillance camera. The view that offers the camera is not the optimal to the achievement of the proposed tasks. Our first goal is to compound a pseudo-top-view of the parking area. To reach this objective we implement an homography computation, like described in this paper, over the captured image of the parking area which has been bounded previously. Only with four identified correspondences between captured image and pseudo-top-view, we can compute the homography correctly. From this compounded image we can extract all the characteristics of interest easily. In a second stage, we realize a texture feature extraction of each parking lot using a Gabor filter bank. This kind of filter banks are used for the texture feature extraction in the literature [12], [13] and [14]. Finally, each feature vector is processed by an SVM (Support Vector Machine) trained to recognize parking conditions. Figure 3 shows the steps that our system follows to achieve these objectives.

IV. TEXTURE FEATURES EXTRACTION

The main goal of the proposed parking management system is the detection of free parking lots in a parking area. These kind of problems can be solved with techniques like motion detection and tracking, but due to given restrictions (single images, large distance, lens distortion, changes in lighting) these techniques are not applicable easily, and there are some works, [10] and [11], which deal with this difficulties. We suggest a solution to solve these problems: an SVM trained with texture feature vectors of every parking lot, which have been taken in different illumination conditions and with diverse type of shadows. From this paper, we propose Gabor filters for texture feature extraction of the background of the parking area, because they compound a multi-resolution representation of itself, as it is described in [12] and [13]. A Gabor filter lets us to extract various features related to the local power spectrum of an image. These features are obtained by filtering the image with a bank of two-dimensional Gabor filter. In [15], [16] and [17], there is a complete description of this kind of filters. A Gabor filter has a convolution kernel which is a product of a Gaussian and a cosine function, and we can establish a preferred orientation and a preferred spatial frequency. The use of Gabor filters, in extracting texture features of an image, is motivated on the following factors: Gabor representation is optimal in the sense of minimizing
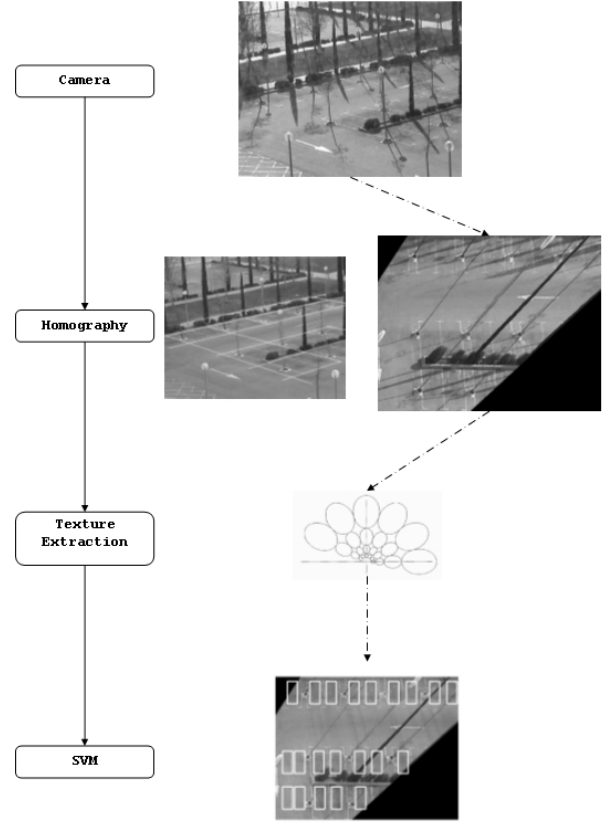

Fig. 3. Paring Management System Diagram.

the joint two-dimensional uncertainty in space and frequency, and a Gabor filters can be considered as an edge and line detectors, hence this kind of filters can be used to characterize the texture information. A two dimensional Gabor function $g(x,y)$ and its Fourier transform $G(u,v)$ can be written as:

$$g(x,y) = \frac{exp\left\{\frac{-1}{2}\left[\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2\right] + 2\pi jWx\right\}}{2\pi\sigma_x\sigma_y} \quad (13)$$

$$G(u,v) = exp\left\{-\frac{1}{2}\left[\left(\frac{(u-W)}{\sigma_u}\right)^2 + \left(\frac{v}{\sigma_v}\right)^2\right]\right\} \quad (14)$$

, where $\sigma_u = \frac{1}{2\pi\sigma_x}$ and $\sigma_v = \frac{1}{2\pi\sigma_y}$. The gaussian's major and minor axis widths are determined through σ_x and σ_y, respectively. Typically, an input image $I(x,y)$, where (x,y) ϵ Ω, and Ω represents the set of image points, is convolved with a 2-D Gabor function $g(x,y)$, to obtain a Gabor feature image $S(x,y)$ as follows

$$S(x,y) = \iint_\Omega I(\phi,\varphi)g(x-\phi,y-\varphi)d\phi d\varphi \quad (15)$$

For the texture feature extraction, we need to generate a Gabor filter dictionary scaling and changing the orientation of a Gabor function $g(x,y)$. Let $g(x,y)$ be the mother Gabor function, then the Gabor filter dictionary can be obtained

through the generating function $g_{mn}(x, y)$ as follows

$$g_{mn}(x, y) = a^{-m} g(x'y') \qquad (16)$$

, with

$$x' = a^{-m}(xcos(\theta) + ysin(\theta)) \qquad$$

$$y' = a^{-m}(-xsin(\theta) + ycos(\theta)) \qquad (17)$$

, where $a > 1$, m and n are integers, $\theta = \frac{n\pi}{K}$, and K is the number of orientations.

A Gabor filter family, generated with equations (16) and (17), forms a complete but non-orthogonal basis set, hence it implies that there is redundant information in the filtered images. We can establish some criteria to design a Gabor family depending on the application. For the texture feature extraction it is crucial to reduce the possibly redundancy derived from the non-orthogonality of the functions set. Let z_l and z_u be the lower and upper frequencies of interest, K the number of orientations and R the number of scales in the multi-resolution decomposition. The design strategy is to ensure that the redundant information is minimized, thus the half peak magnitude of the filter responses has to be tangent to its nearest half peak magnitude neighbours, as [12] shows. For our application we have employed the following values: $z_l = 0.05$, $z_u = 0.4$, $K = 6$ and $R = 4$. Figure 4 shows the contours which indicate the half peak magnitude of the filters.
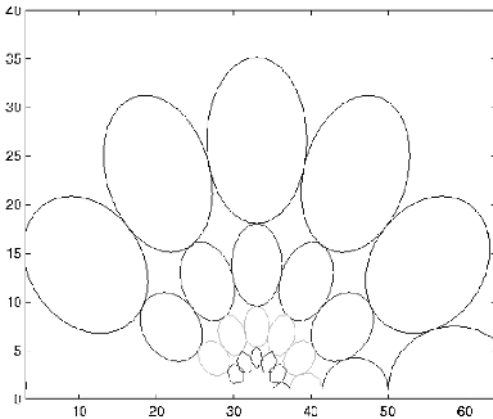

Fig. 4. Contour of the half-peak magnitude of the generated Gabor functions.

Then, given an image $I(x, y)$, its Gabor transform $S_{mn}(x, y)$ is obtained changing the Gabor function $g(x, y)$ in (15) by $g_{mn}(x, y)$. For each parking lot we compute its Gabor transform, then we use the mean μ_{mn} and the standard deviation σ_{mn} of the filtered image for the classification and recognition purposes. A feature vector $\bar{f}$ is created using μ_{mn} and σ_{mn} as follows

$$\bar{f} = [\mu_{00}\sigma_{00}\mu_{01}\sigma_{01} \ldots \mu_{(S-1)(K-1)}\sigma_{(S-1)(K-1)}] \qquad (18)$$

Length of vectors	Class Rate
48	99.73%
38	99.73%
28	99.71%
18	99.62%
8	98.17%

TABLE I
EXPERIMENTAL RESULTS OBTAINED WITH DIFFERENT LENGTHS OF THE TEXTURE FEATURE VECTORS.

For this work, each parking lot is filtered by the described Gabor filter bank and each feature vector of our application has 48 components, because we have employed the following values: $K = 6$ and $R = 4$.

V. EXPERIMENTAL RESULTS

The parking condition recognition task is done through a trained SVM. We need to use a supervised learning method because the surveillance camera takes images: with different illumination conditions, shadows and possibly strong sun-light spots. Thus, an approach to object detection based on techniques as motion detection or tracking is not easily applicable. For this work, we have created a texture database which consists of 2 different texture classes: one class represents the texture of the background of a parking lot, and the other one represents the textures of vehicles. We have trained the SVM with a database which contains 24300 different texture feature vectors, extracted from images which were taken in diverse illumination conditions, of which 22470 belong to the free parking lot class, and 1830 belong to the vehicle texture class. For the training step we have used 12150 feature vectors, the half of each class. With the SVM we can use some different types of kernels, but for this study we have used a Linear kernel only. We have made use of the implementation in C called SVM^{Light} described in [18].

Finally, the implementation of the parking management system offers the following results. Every minute the system captures an image of the parking zone of 720×576 pixels. For this work, we can manage a parking zone with 21 parking lots with a single camera. The first step is to compound the pseudo-top-view of the image. Then, the system extracts the texture feature vector of each lot. To end, these 21 texture vectors obtained are passed to the trained SVM for the recognition task. We have tested our trained SVM with 12150 feature vectors and the results obtained with different lengths of the feature vectors are in Table I.

From vectors with more than 38 components the class rate obtained is the same, then we can compute the recognition task with vectors of 38 components in order to reduce the average time of feature extraction and classification. These results show that the main information for class recognition is in low frequencies. The software developed for this work generates visual information over some images which indicates if the parking lot is free (white rectangle), if it is occupied (white rectangle with a cross inside) or if the car is parked

on a forbidden zone (white rectangle with a two white filled triangles inside). Figure 6 shows that our systems recognizes free parking lots even if there are shadows over the parking area, and it displays some results obtained with vehicles in forbidden zones and with sun-light spots.

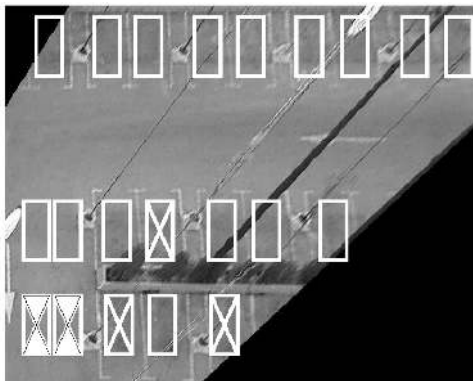Fig. 5.   Captured image with sun-light spots and shadows.

Fig. 6.   Experimental results over occupied parking lots.

VI. CONCLUSIONS

In this paper, we have proposed a novel methodology for computing a 2D homography applying a concrete algorithm of computer algebra: the homomorphic method for solving systems of equations over $\mathbb{Z}$. This algorithm is a new resource for the computer vision discipline, which eliminates the problems of normalization and of loss of precision which are associated with the numerical approach developed in the DLT algorithm. We have showed the results obtained with a comparative analysis between these two methodologies, and the symbolic approach presents the best solution: it solves the system without normalization process and with less number of correspondences between points. Also, we have presented

a Parking Management Systems based on the computer algebra algorithm proposed, which works well in a real scenario.

In future works, we will deal with: a deeper comparative analysis of complexity between these numerical and symbolic algorithms, and with a new search of computer algebra algorithms that could be appropriate in some computer vision problems. Related to the parking management system, future works will include tests with more than one cameras and with car tracking techniques. Then, we will try to compound a real-top-view of a parking zone, and we will be able to solve the problems of occluded regions in the image.

REFERENCES

[1] R. J. Holt, S. Huang, and A. N. Netravali, "Algebraic methods for image processing and computer vision," *IEEE Transactions on Image Processing*, vol. 5, pp. 976–986, June 1996.
[2] F. Winkler, *Polynomial Algorithms in Computer Algebra*. Springer, Wien New York, 1 ed., 1996.
[3] B. Buchberger, G. E. Collins, and R. Loos, *Computer Algebra, symbolic and algebraic computation*. Springer, Wien New York, 2 ed., 1983.
[4] K. O. Geddes, S. R. Czapor, and G. Labahn, *Algorithms for Computer Algebra*. Kluwer Academic Publishers, 6 ed., 1999.
[5] D. Manocha, "Solving systems of polynomial equations," *IEEE Computer Graphics and Applications*, vol. 14, pp. 46–55, March 1994.
[6] M. De-Leo, E. Dratman, and G. Matera, "Numeric vs. symbolic homotopy algorithms in polynomial system solving: a case study.," *Journal Of Complexity*, vol. 21, pp. 502–531, February 2005.
[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, pp. 25–65. Cambridge University Press, second ed., 2003.
[8] J. Chen, W. E. Dixon, D. M. Dawson, and M. McIntyre, "Homopraphy-based visual servo tracking control of a wheeled mobile robot," *IEEE Transactions on Robotics*, vol. 22, pp. 407–416, April 2006.
[9] M. I. A. Lourakis and A. A. Argyros, "Changing planar homographies for fast and reliable 3d plane tracking," in *18th Internacional Conference on Pattern Recognition*, vol. 1, pp. 582–586, aug 2006.
[10] S. Funck, N. Mohler, and W. Oertel, "Determining car-park occupancy from singles images.," in *IEEE Intelligent Vehicles Symposium*, (Parma, Italy), pp. 325–328, June 2004.
[11] N. Mizukoshi, I. Horiba, K. Ueda, and H. Onodera, "Parking condition discrimination system of image processing type using a neural network model.," in *IEEE Vehicle Navigation & Information Systems Conference Proceedings*, (Yokohama, Japan), pp. 69–74, Sept. 1994.
[12] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *Pattern analysis and machine intelligence*, vol. 18, pp. 837–842, Aug. 1996.
[13] S. E. Grigorecsu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on gabor filters," *Image Processing*, vol. 11, pp. 1160–1167, Oct. 2002.
[14] D. Dunn and W. E. Higgins, "Optimal gabor filters for texture segmentation," *Image Processing*, vol. 4, pp. 947–964, July 1995.
[15] D. Gabor, "Theory of communication," *Int. Elect. Eng.*, vol. 93, pp. 429–457, 1946.
[16] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A.*, vol. 2, pp. 1160–1169, July 1985.
[17] J. G. Daugman, "Complete discrete 2-d gabor transforms by neural networks for image analysis and compression," *Acoustics, Speech and Signal Processing*, vol. 36, pp. 1169–1179, July 1988.
[18] T. Joachims, "Making large-scale svm learning practical," *Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.)- MIT-Press, Cambridge, USA*, pp. 41–56, 1998.