A Framework for Feature-Based Motion Recovery in Ground Plane Vehicle Navigation

J.M. Sanchiz[†], F. Pla[*], J.A. Marchant[‡]
[†]*University Jaume I. Department of Computer Science. Castelló. Spain*
[‡]*Silsoe Research Institute. Silsoe. Beds. UK*

Abstract. This paper describes a feature point matching strategy and motion recovery applied to vehicle navigation. A transformation of the image plane is used that keeps the motion of the vehicle parallel to the transformed plane. These allows us to define linear tracking filters to estimate the real-world positions of the features. The correspondences between features are first selected by similarity, taking into account the smoothness of motion and rigidity of the scene. Further processing brings out the rest of correspondences. The method is applied to a real application consisting of an autonomous vehicle navigating in a crop field.

1 Introduction

The general problem of motion analysis from image sequences can be stated as to finding the motion parameters that the camera co-ordinate system has to undergo in two consecutive images in order to match the projection of the scene viewed in both images [1,2,3]. Recovering a general 3D motion is an ill-defined problem and, due to the speed-scale ambiguity, only the direction of the translation can be recovered if no a-priori knowledge is applied. The computation of the motion parameters is based on a former computation of the projected motion, that is, the motion of the projected scene on the image plane, which is approached by either the computation of the optical flow, basically using differences in the intensity of two consecutive images, or by the computation of feature correspondences [1,4]. This latter method seems more reliable for real-time applications since, once the features have been selected, the amount of data to process is significantly reduced.

In real-world applications some constrains are usually applied to the general problem: for example, if the motion is known, then a 3D map of the scene can be recovered; or there exist some landmarks on the scene whose real-world positions are known; or some of the motion parameters are fixed, a rotation angle or a component of the translation vector. The latest situation is usually the case in autonomous vehicle navigation, Fig. 1. The camera height, v, and tilt angle, φ, are fixed. Also the roll angle is fixed and set to zero, since the vehicle is assumed not to roll. It is often assumed that the features lie on the ground plane. This configuration can be found in applications like road-following and indoor or outdoor navigation. Feature tracking is a usual approach to motion estimation although the nature of the extracted features depends on the type of scenes we deal with. Feature point tracking is a common approach, where the points are usually extracted from grey-level images by some corner detector. This is the configuration that will be followed in the present work.

2 Problem Statement

The relation between the camera and the world co-ordinate systems can be seen in Fig. 2. At time instant 0 (first frame) the x_w axis is defined to be aligned with the x_c axis, in further frames this alignment will no

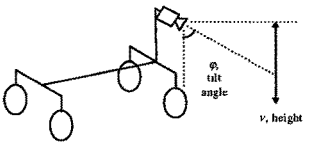

Fig. 1. Camera on a vehicle

longer exist (in general), given that the vehicle rotates and translates over the ground. With this camera configuration the vehicle 2D motion is expressed as a 3D motion in the camera co-ordinate system. To avoid this, and constrain the camera motion to be two-dimensional, we introduce the *virtual image plane*, Fig. 3, parallel to the plane of motion. The co-ordinate system of this virtual camera is defined by applying a rotation of angle φ around the x_c axis. The advantage of using the virtual image plane is that there will be no translation along the z_{vc} direction, this will be convenient when finding the feature correspondences and tracking the features through a sequence of images, since it allows us to define *linear* Kalman filters to estimate the real-world positions of the feature points. Features can be detected on the original image, then they are transferred to the virtual image plane, so one does not have to transfer the whole image, but just the selected features, saving processing time. Once the camera is calibrated [5] the transformation is fixed and it can be performed by a look-up table. The co-ordinates of the two planes are related by:

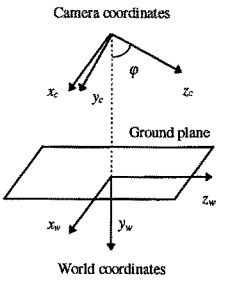

Fig. 2. Camera and world co-ordinates

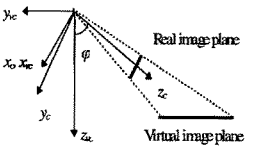

Fig. 3. Virtual camera

$$\begin{pmatrix} x_{vc} & y_{vc} & z_{vc} \end{pmatrix}^{\mathrm{T}} = \left(f\frac{x_c}{y_c \sin\varphi + f\cos\varphi} \quad f\frac{y_c\cos\varphi - f\sin\varphi}{y_c\sin\varphi + f\cos\varphi} \quad f \right)^{\mathrm{T}} \tag{1}$$

Let $\mathbf{r}_{k,k-1}$ and $\mathbf{t}_{k,k-1}$ express the real-world frame-to-frame motion, $\mathbf{r}_{k,k-1}$ is a 2D rotation matrix and $\mathbf{t}_{k,k-1}$ a 2D translation vector. These motion parameters relate the projections, in two consecutive frames, of a pair of matched features on the virtual image plane:

$$\begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_k = \mathbf{r}_{k,k-1}\begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_{k-1} + \frac{f}{z_{vc}}\mathbf{t}_{k,k-1} \tag{2}$$

$$\begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_{k-1} = \mathbf{r}_{k,k-1}^{-1}\left[\begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_k - \frac{f}{z_{vc}}\mathbf{t}_{k,k-1}\right] \tag{3}$$

where the superscript *proj* indicates co-ordinates of the projected features, and z_{vc} is the real depth of the feature, whose value is known from the calibration and equal to the camera height, since the features lie on the ground. The absolute motion of the vehicle can be computed recursively from the frame-to-frame motion. Let $\mathbf{R}_k$ (2D rotation matrix) and $\mathbf{T}_k$ (2D translation vector) be the absolute motion of the vehicle in world co-ordinates, then:

$$\begin{aligned} &\mathbf{R}_k = \mathbf{R}_{k-1}\mathbf{r}_{k,k-1}^{-1}\ ;\ \mathbf{R}_0 = \mathbf{I} \\ &\mathbf{T}_k = \mathbf{T}_{k-1} - \mathbf{R}_k\mathbf{t}_{k,k-1}\ ;\ \mathbf{T}_0 = \mathbf{0} \end{aligned} \tag{4}$$

The problem consists of finding $\mathbf{r}_{k,k-1}$ and $\mathbf{t}_{k,k-1}$ through a sequence of images. This implies to select the features and to track them estimating their real-world positions.

3 Feature Correspondence

Due to the special characteristics of the application to which this work is mainly directed, we have used points as features to be tracked. For the rest of the work we assume that a set of feature points is available for every image. The method to find the correspondence and motion recovery is independent of the way the feature points

were extracted, the only requirement is that they are stable and that some characteristics of the points can be provided. In our approach we assume a measurement of similarity on pairs of features is available, we combine it with the assumptions of smoothness of motion and rigidity of the scene, and we exploit the special configuration of the camera used in vehicle navigation to define the virtual image plane. The correspondence problem has to be solved as a first step to feature tracking, by which we mean to assign a new feature observation to a tracker (tracking filter) that estimates the real position of the feature, based on previous observations; and to the problem of estimating the camera motion from the set of correspondences (point correspondences in this case). Each feature has a tracking filter (we will use Kalman filters) assigned to it, that estimates its most likely position from all previous observations. The data to be estimated are the projection on the virtual image plane, $(x_{vc}^{proj}, y_{vc}^{proj})$. Features observed in image k-1 already have an associated tracker. Features appearing in image k have not been associated to a tracker yet, but they will be after the correspondence is found, then the new observation will be used to update the corresponding tracker. Features in image k for which no correspondence is found are assumed to be new, and a new tracker is initiated for them. The matching is found from features in frame k to all the estimated positions of the features that have been observed before, we will call these present tracks. The method can cope with poor feature extraction: a feature that is not detected during some frames will be assigned to its corresponding track when detected again. When a feature is not detected a change of co-ordinates is made to its estimated position to update it on the virtual plane. The tracks are filtered before computing the matching to reject those falling outside the present field of view.

The correspondence problem can be stated as follows:

Let n_k be the number of detected features in image k, and let n_{k-1} be the estimated positions from the trackers at time k-1 that survive the filtering (tracks in image k-1). Solving the correspondence consists of finding a backward mapping, Ψ: $i \in [1..\, n_k] \rightarrow j \in [0,1..\, n_{k-1}]$ and a forward mapping Θ: $j \in [1..\, n_{k-1}] \rightarrow i \in [0,1..\, n_k]$ following some criteria, and so that:

$\Psi(i)$ is the corresponding track in image k-1 to feature i in image k. $\Psi(i)=0$ means feature i has no corresponding in image k-1 (new appearing feature). $\Theta(j)$ is the corresponding feature in image k to track j in image k-1. $\Theta(j)=0$ means feature j has no corresponding in image k (it has disappeared from the field of view or has not been detected).

The criteria to find the mapping have to satisfy the following constraints:

3.1 Similarity between features

A general procedure to give a measure of similarity between features consists of computing a vector of characteristics for every feature, $(c_{1i}, c_{2i}, \ldots, c_{Ni})^T$. The meaning of these characteristics is highly dependent on the method used to detect them (details on the characteristics that have been used in our application will be given in the results section). Then a distance between features can be defined as:

$$d_{ij}^2 = \sum_{l=1}^{N} w_l^2 (c_{li} - c_{lj})^2 \tag{5}$$

where d_{ij} is the distance between feature i in image k and feature j in image k-1, and w_l is the weight associated to characteristic l (l=1...N).

3.2 Smoothness of motion

The previous estimated frame-to-frame motion, $\mathbf{r}_{k\text{-}1,k\text{-}2}$ and $\mathbf{t}_{k\text{-}1,k\text{-}2}$, is used to search for correspondences. Given a feature i in image k, the search area to find its corresponding in image k-1 is located by back-projecting (3) its co-ordinates on image k-1, but an estimation of the depth, z_{vc}, is needed in equation (3), so we make the assumption that the depth is equal to the camera height, i.e. the features lie on the ground. This is a quite reasonable approach for most autonomous navigation applications where the camera is pointed to the ground, the same idea was used by Liu et al. in [2] to recover motion from line and point correspondences assumed to be on the ground plane. Finally the search area is set as an ellipse of centres **a** and **b**, whose main axis is orientated along the motion epipolar line. Co-ordinates of point **a** are set by back-projecting the co-ordinates of the feature using a decreased value for z_{vc}, and point **b** is set by back-projecting it with an increased value of z_{vc} (30% of increasing/decreasing has been used in our application).

The distance used can be considered a modified Euclidean distance since points inside a fixed distance threshold do not fall inside a circle, but inside an ellipse orientated in the direction of the epipolar line, direction which has been found using the previous value of the motion parameters. By this approach we favour the searching for correspondences in the direction of the motion, which is mainly forward although can have some rotational or transversal translation component.

3.3 Rigidity of the scene

The rigidity of the scene constrains the correspondences which arise from the same motion of the camera, this means that, ideally, the values of $\mathbf{r}_{k,k\text{-}1}$ and $\mathbf{t}_{k,k\text{-}1}$ in (2) and (3) have to be the same for all features. Once some candidate correspondences have been found by selecting similar features in the search areas, a Hough Transform-like technique is applied to further select those features having very close values of $\mathbf{r}_{k,k\text{-}1}$ and $\mathbf{t}_{k,k\text{-}1}$.

4 Procedure to find the correspondence

Following the criteria explained above leads to obtaining an initial set of correspondences of present features to existing tracks, and a first guess for the frame-to-frame motion. After selecting the coherent correspondences a better value for the motion parameters can be obtained by minimisation, the rest of correspondences can then be computed by back-projecting the still non-matched features using the recovered motion, and by finding the most similar track in image k-1. The complete method to find all the correspondences can be expressed as follows:

4.1 Compute candidate matches

Build a distance matrix, **dm**. Each entry, **dm**[i,j], represents the distance (or dissimilarity) between feature i in image k and track j in image k-1. Then find candidate correspondences as those pairs (i,j) in which position **dm**[i,j] is at the same time minimum in its row and its in column. This means that track j is the most similar to feature i, and feature i is the most similar to track j.

4.2 Select the coherent matches by a Hough Transform-like technique

For each candidate correspondence, give values to ψ and compute the set of points in the 3D parameter space, (ψ, t_x, t_y), using equation (2). Apply a clustering [6] to the set of points (ψ, t_x, t_y) and find the biggest cluster, $(\psi_0, t_{x,0}, t_{y,0})$, which gives a first guess for the motion parameters, $\mathbf{r}_{0;k,k-1}$ and $\mathbf{t}_{0;k,k-1}$. Mark the correspondences that originated the points that support the biggest cluster as a coherent match and discard the others. A similar technique was used by Sanchiz et al. in [7] to find correspondences of blobs.

4.3 Compute the motion by minimisation

Find a best value of the frame-to-frame motion by minimisation [3], $\mathbf{r}_{1;k,k-1}$ and $\mathbf{t}_{1;k,k-1}$. The coherent correspondences are used to prepare two sets of 3D points in world co-ordinates. The projected co-ordinates of all the features are known, and the depths, z_{vc}, are set to the camera height. The real-world positions in virtual camera co-ordinates of a feature are then:

$$(x_{vc}, y_{vc}, z_{vc})^{\mathbf{T}} = (x_{vc}^{proj}\frac{z_{vc}}{f}, y_{vc}^{proj}\frac{z_{vc}}{f}, z_{vc})^{\mathbf{T}} \tag{6}$$

4.4 Apply a further filtering to the present matches

The matches that still represent a big variation from the motion parameters that were found by minimisation are rejected. Once the motion parameters are known, the new observed depth of a feature can be found by triangulation. From equation (2), and assuming that the module of the translation due to this correspondence is the same as the module of the translation found by minimisation, we can solve for z_{vc} ($z_{vc} = z_{vc\,(obs)}$) and $\mathbf{t}_{k,k-1}$ ($\mathbf{t}_{k,k-1} = \mathbf{t}_{k,k-1\,(obs)}$):

$$z_{vc\,(obs)} = f\frac{\left|\mathbf{t}_{1;k,k-1}\right|}{\left|\begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_k - \mathbf{r}_{1;k,k-1}\begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_{k-1}\right|} \tag{7}$$

$$\mathbf{t}_{k,k-1\,(obs)} = \frac{z_{vc\,(obs)}}{f}\left[\begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_k - \mathbf{r}_{1;k,k-1}\begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_{k-1}\right] \tag{8}$$

The filtering rejects the correspondences that produce a big variation in the depth or a big variation in the direction of the translation. In our application we have rejected variations in depth bigger than 30%, and variations of more than 5 degrees in the direction of the translation.

4.5 Find the final matching

The rest of the matches are found by computing $z_{vc\,(obs)}$ and $\mathbf{t}_{k,k-1\,(obs)}$ for all possible correspondences of still non-matched features. Computing the distance for those whose $z_{vc\,(obs)}$ and $\mathbf{t}_{k,k-1\,(obs)}$ values are inside the limits, and successively picking up the most likely correspondence.

4.6 Find the final value of the motion parameters

A new (and definitive) value of the frame-to-frame motion, $\mathbf{r}_{k,k-1}$ and $\mathbf{t}_{k,k-1}$, is found by minimisation [3] using all the correspondences.

5 Tracking Features

A tracking filter is initiated for every new feature appearing in the scene. Its function is to estimate the position of a feature from its set of observations, and from the estimated motion parameters. The data to estimate are the co-ordinates of the projection of a feature on the virtual image plane. Fixing the depth, z_{vc}, to the camera

height, the real-world position can be computed from equation (2). The Kalman filter [8] is used as a tracker, it estimates the best value, in a least-squares sense, of a state vector from a set of Gaussian noisy measurements in dynamic linear systems. Precisely the frame-to-frame motion can be expressed as a linear system if we use the co-ordinate axes of the virtual vertical camera. The Kalman filter equations are:

System: $\mathbf{x}_k = \mathbf{\Phi}_{k-1}\,\mathbf{x}_{k-1} + \Gamma_{k-1}\,\mathbf{u}_{k-1} + \mathbf{v}_k;\ \mathbf{v}_k \in N(\mathbf{0},\mathbf{R}_1)$ (9)

Measurement: $\mathbf{y}_k = \mathbf{C}_k\,\mathbf{x}_k + \mathbf{e}_k;\ \mathbf{e}_k \in N(0, \mathbf{R}_2)$ (10)

Initial state: $E[\mathbf{x}_0] = \mathbf{x}_{0|0};\ \mathrm{cov}[\mathbf{x}_0] = \mathbf{P}_0;\ E[\mathbf{v}_k\,\mathbf{e}_k^T] = \mathbf{0}$

Prediction at k-1: $\mathbf{x}_{k|k-1} = \mathbf{\Phi}_{k-1}\,\mathbf{x}_{k-1|k-1} + \Gamma_k\,\mathbf{u}_k$ (11)

$\mathbf{P}_{k|k-1} = \mathbf{\Phi}_{k-1}\,\mathbf{P}_{k-1}\,\mathbf{\Phi}_{k-1}^T + \mathbf{R}_1$ (12)

Prediction at k: $\mathbf{K}_k = \mathbf{P}_{k|k-1}\,\mathbf{C}_k^T\,[\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_2]^{-1}$ (13)

$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k\,[\,\mathbf{y}_k - \mathbf{C}_k\,\mathbf{x}_{k|k-1}\,]$ (14)

$\mathbf{P}_{k|k} = [\,\mathbf{I} - \mathbf{K}_k\,\mathbf{C}_k\,]\,\mathbf{P}_{k|k-1}$ (15)

The state vector is defined as: $x_k = \left(x_{vc}^{proj} \quad y_{vc}^{proj}\right)_k^T$ (16)

The transition matrix, $\mathbf{\Phi}_k$, is used to express the rotation, and the input part in equation (9) is used to model the translation:

$$\mathbf{\Phi}_k = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix}_k \quad (\text{or } \mathbf{\Phi}_k = \mathbf{r}_{k,k-1}) \qquad (17) \qquad \Gamma_k = \mathbf{I};\ \mathbf{u}_k = \begin{pmatrix} t_x / f \\ t_y / f \end{pmatrix}_k \qquad (18)$$

The transition from state k-1 to state k is:

$$\begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_k = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix}_{k-1} \begin{pmatrix} x_{vc}^{proj} \\ y_{vc}^{proj} \end{pmatrix}_{k-1} + \begin{pmatrix} t_x / f \\ t_x / f \end{pmatrix}_{k-1} \qquad (19)$$

The measurements from the visual information are the co-ordinates of the projected features, $(x_{vc}^{proj}, y_{vc}^{proj})$, so the measurement matrix is the 2x2 identity matrix, $\mathbf{C}_k$=**I**. The covariance matrices are initiated as: $\mathbf{P}_0 = \mathbf{R}_1 = \mathbf{R}_2 = \sigma^2\,\mathbf{I}$, where σ is set to a fraction of the field size.

6 Experimental results

In our application the context is an autonomous vehicle that navigates in an outdoor crop field. The scenes we deal with consist of a perspective view of a piece of crop field where only natural objects (plants) appear. The purpose of the application is to spray on the plants or weeds automatically, thus, the vehicle is equipped with a bar of nozzles to perform the spraying. Images are segmented [9] to divide the scene into three classes, regions of class "soil", "plant" and "weed". The same images are used to identify the plants and to compute the motion parameters, which are used to place the images on a map of the field, built up while the vehicle moves [10]. Exploring the map along the nozzle bar allows us to open those nozzles that are over a plant or weed. The motion estimation is intended to be passed to the vehicle control system, thus closing the loop and trying to perform an autonomous row following.

Features are detected as dominant points in the contours of the regions of class "plant". A contour following algorithm was applied to code the boundaries, and the dominant points were found by a neural network-based algorithm for dominant point detection [11]. The tracking method explained in this paper has been tested with several image sequences obtained from a camera mounted on a manually driven

vehicle, Fig. 1, undergoing a zigzag motion. The camera height was v=1200 mm., the tilt angle was φ=66 degrees and from a previous calibration [5] of the camera the lens focal length was f=40 mm.

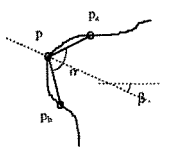

Fig. 4. Convexity, α, and orientation, β, at a dominant point.

Since we use dominant points in contours as features, two characteristics that give satisfactory results for similarity measurements are the convexity and the orientation of the contour in a small area around the point, so we fix N=2 in equation (5). Assuming that both characteristics have the same importance we fix w_1=1 and w_2=1 (5). From a dominant point, **p**, two points are found at either side of the contour, $\mathbf{p}_a$ and $\mathbf{p}_b$, so that the distance between **p** and $\mathbf{p}_a$, and between **p** and $\mathbf{p}_b$ is as close as possible to a given value (10% the contour length has been used). The angles of convexity and orientation are computed as shown in Fig. 4.

An example of the matching process can be seen in Fig. 5. The rate of successful correspondences was over 95% through a whole sequence of thirty images, this rate was determined by manually identifying the correct correspondences, the incorrect, and the missed ones, in every image of the sequence. From the frame-to-frame motion, $\mathbf{r}_{k,k-1}$ and $\mathbf{t}_{k,k-1}$, the absolute position and orientation of the vehicle, $\mathbf{R}_k$ and $\mathbf{T}_k$, are found from (4). In order to measure the accuracy of the motion estimation, $\mathbf{R}_k$ and $\mathbf{T}_k$ were used to place every image over the ground plane, thus building a map of the crop at a desired scale. Since the images overlap in a certain amount, a majority voting scheme was used to determine the classification of the pixels, counting the times that a pixel is assigned a certain class. As the plants are aligned in rows, every plant was manually assigned to a certain row, and straight lines were fitted to every row, using the centres of the blobs of class "plant" as the data for the fit. The root-mean-square (r.m.s.) error of the fit, the parallelism and the distance between neighbouring lines (and its comparison with the real-world distance) are measurements that indicate the accuracy of the map, and so of the estimated motion. Fig. 6 shows a map drawn in a 256x512 image at a scale of 15 mm. per pixel, the last field of view and position of the nozzle bar are outlined. The r.m.s. error of the fit was below 30 mm. for the three lines respectively, the angle between neighbouring lines was below 0.5 degrees and the distance (measured at the centre of the map) was 413.2 and 408.3 mm. (400 mm. is the approximate real-world distance between rows in the crop). It has also to be noted that the lines were fitted to the centres of the blobs of class "plant", which are not exactly over the lines that pass through the centre of the crop rows, nevertheless the results are quite satisfactory.

7 Conclusions

A strategy to solve the correspondence problem and the tracking of features has been presented. The method is intended for autonomous navigation applications, where a general constraint is that the motion is undergone on the ground plane. The similarity between features and the smoothness of motion are taken into account to provide an initial matching. The matches that are coherent with the rigidity of the scene are selected by a Hough Transform. The motion is computed by minimisation, and used, together with the similarity between features, to obtain the final correspondence. A Kalman filter is defined for each feature to estimate its real

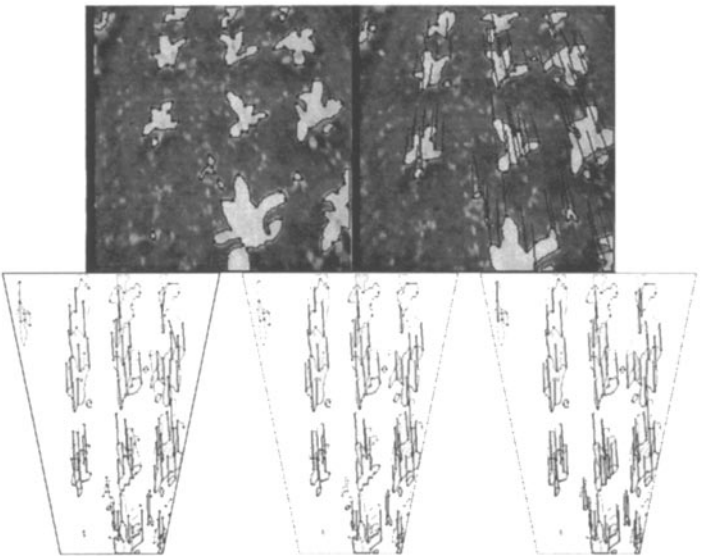

Fig. 5. From left to right and top to bottom: Two consecutive images (contours and dominant points outlined, correspondence superimposed on the second one). Both images transferred and overlapped on the virtual image plane, initial matching. Selected matches after applying the Hough Transform-like technique. Final correspondence.

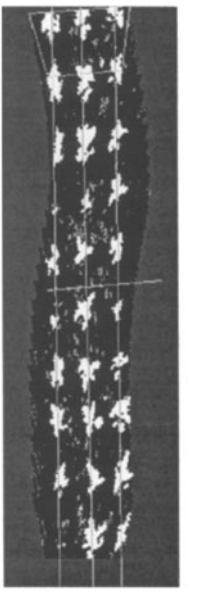

Fig. 6. Map built from the recovered motion, with lines fitted the rows.

position. The method has been applied to a real-world application.

(Work supported by the Spanish Ministry of Science, CICYT TIC95-0676-C02-01)

References

[1] Y.F. Wang, N. Karandikar, K. Aggarwal, "Analysis of video image sequences using point and line correspondences", *Pattern Recognition.*, vol. 24, pp. 1065-1084. 1993.

[2] Y. Liu, T.S. Huang, O.D. Faugeras, "Determination of camera location from 2D to 3D line and point correspondences". IEEE Transactions on PAMI, vol. 12, no. 1, pp. 28-37. 1990.

[3] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns", IEEE Transactions on PAMI, vol. 13, pp. 376-380, 1991.

[4] G.L. Scott, H.C. Longuet-Higgins, "An algorithm for associating the features of two images". Proceedings of the Royal Society London, B 224, pp. 21-26. 1991.

[5] R.Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision", Proceedings of the IEEE Conference CVPR'86, pp. 364-374. 1986.

[6] P. Trahanias, E. Skordalaskis, "An efficient sequential clustering method", Pattern Recognition, vol. 22, pp. 449-453, 1989.

[7] J.M. Sanchiz, F. Pla, J.A. Marchant, R. Brivot, "Structure from motion techniques applied to crop field mapping", Image & Vision Computing, vol. 14, no. 5, pp. 353-363. 1996.

[8] Y. Bar-Shalom, T.E. Fortmann, "Tracking and data association", edited by W.F. Armes. Academic Press, Inc. Math. & Science in Engineering, vol. 179. ISBN 0-12-079760-7. 1988.

[9] R. Brivot, J.A. Marchant, "Segmentation of plants and weeds using infrared images", IEE Proceedings, Vision, Image and Signal Processing, vol. 143, no. 2, pp. 118-124, 1996.

[10] J.M. Sanchiz, J.A. Marchant, F. Pla, A. Hague, "Real-Time visual sensing for task planning in a field navigation vehicle", Real-Time Imaging (in press). 1996.

[11] J.M. Sanchiz, J.M. Iñesta, F. Pla, "A neural network-based algorithm to detect dominant points from the chain-code of a contour", Proc. of the 13th ICPR, vol. IV, pp. 330-334. 1996.