Linear Algorithms for Object Pose Estimation[†]

T. N. Tan, G. D. Sullivan and K. D. Baker
Intelligent Systems Group
Department of Computer Science
University of Reading, ENGLAND

Abstract

This paper concerns the estimation of object pose in scenes where objects are located on the ground plane which has known orientation and position w.r.t. the camera. Novel algorithms are described, based on the concept of interpretation planes and that of pencils of planes. The methods are linear, computationally simple, and give unique and closed-form solutions, thus eliminating many of the problems associated with the existing pose recovery algorithms. They require a minimum of two 2D-3D line correspondences. Experimental results are included which show that the proposed algorithms are robust to noise, and capable of accurate pose recovery using real images of outdoor scenes.

1 Introduction

Object pose recovery from monocular light intensity images is a major objective for computer vision. Most previous work in the area has used either 2D-3D point [1-4] or 2D-3D line [5-12] correspondences. Although some success has been reported in the literature (e.g., [5-6]), there are many problems associated with the existing pose estimation algorithms [16], most importantly: high or unknown sensitivity to sensory data noise, non-uniqueness of the solution, requirement of good initial guesses, unguaranteed convergence to correct solutions, and high computational complexity. These problems are mostly due to the inherent non-linearity of pose estimation and the fact that the existing pose recovery algorithms are (rightly) concerned with the general case of six degrees of freedom, and treat special cases as having marginal interest.

Nevertheless, special cases abound in the real-world. Their occurrence, if properly ascertained, can help to avoid the above problems, and dramatically simplify pose estimation. The particular special case considered here arises in traffic scenes where objects (e.g., cars) are located on a ground surface which is, at least locally, substantially planar. Other potential applications include model-based vision for industrial parts on conveyor belts or other robot working surfaces.

We approximate the ground surface by the X-Y plane of a world coordinate system (WCS) whose Z-axis points towards the sky. The pose of an object in this WCS is uniquely determined by three independent pose parameters (assuming the X-Y plane of

[†]. This work was carried out as part of the ESPRIT project P2152 (VIEWS).

BMVC 1992 doi:10.5244/C.6.62

the object-centred coordinate system coincides with that of the world coordinate system): the rotation angle θ about the (vertical) Z-axis, and the two translations T_x and T_y on the ground plane. The other three parameters are all zero:

$$\phi, \ \ \psi, \ \ T_z = 0 \tag{1}$$

where ϕ and ψ are the rotation angles around the X and Y axes, and T_z is the translation along the vertical axis. Equation (1) is called the *ground plane constraint* (GPC).

It is shown in this paper that by expressing object pose in the world coordinate system and by incorporating the GPC into the formulation of pose constraint equations, the pose estimation problem can be linearized. As a consequence, simple and robust pose recovery algorithms can be developed, and the problems associated with the existing algorithms listed at the beginning of this section are avoided. Moreover, closed-form solutions can be obtained. This choice of the WCS simplifies the analysis since it directly eliminates 3 of the 6 pose parameters; any other coordinate system could be used but would introduce additional parameters with their corresponding constraint equations computable from the GPC. The potential importance of the GPC in simplifying pose recovery has been discussed in several articles (e.g., [1, 11, 17]), but either no use has been made of the constraint [11], or it has not been used fully to avoid the problems listed at the beginning of this section [1, 17]. It should be pointed out that this paper concerns the use of the GPC in model-based object pose recovery. The application of the GPC in 3D structure and motion estimation is described in a companion paper [18].

In the following sections, we first describe the imaging model used in this paper, then discuss pose estimation under the GPC using 2D-3D line correspondences, and finally summarize the experimental results obtained.

2 Coordinate systems and imaging model

We assume a pinhole camera model with perspective projection and no lens distortion. This is sketched in Fig.1. It can be shown that under this imaging model and the GPC, the image coordinates (u, v) and the model or object coordinates (x_m, y_m, z_m) of point $\boldsymbol{P}$ are related to each other by [16]:

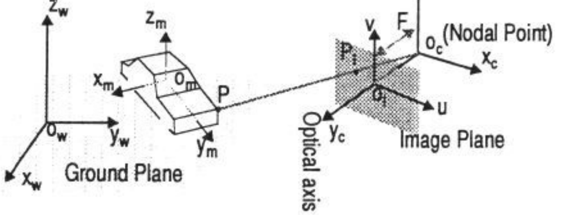


Figure 1. *Coordinate systems and the imaging model.*

$$\begin{bmatrix} su \\ sF \\ sv \\ s \end{bmatrix}^T = \begin{bmatrix} x_m \\ y_m \\ z_m \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & 0 & 1 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \tag{2}$$

where θ, T_x and T_y are the three pose parameters of the object in the WCS, F the camera focal length, s a non-zero scale, and m_{ij}, $i, j = 1, 2, 3, 4$, are the elements of the 4x4 homogeneous perspective transformation matrix which maps 3D world coordinates into 2D image coordinates, and which is assumed to be known in the subsequent discussions.

3 Pose recovery algorithms

A 3D model line L_i can be described either by two points $\vec{P}_{i1}$ and $\vec{P}_{i2}$:

$$L_i: \begin{cases} \vec{P}_{i1} = (x_{i1m}, y_{i1m}, z_{i1m}) \\ \vec{P}_{i2} = (x_{i2m}, y_{i2m}, z_{i2m}) \end{cases} \tag{3}$$

where subscript m indicates model coordinates, or by a unity directional vector $\vec{N}_i$ and a point $\vec{P}_i$:

$$L_i: \begin{cases} \vec{N}_i = (\alpha_i \ \ \beta_i \ \ \gamma_i)^T \\ \vec{P}_i = (x_{im}, y_{im}, z_{im}) \end{cases} \tag{4}$$

or by two intersecting planes Π_{i1} and Π_{i2}:

$$L_i: \ \{ \begin{matrix} \Pi_{i1}: & x_m + b_{i1}y_m + c_{i1}z_m = 0 \\ \Pi_{i2}: & x_m + b_{i2}y_m + c_{i2}z_m + d_{i2} = 0 \end{matrix} \tag{5}$$

Equation (5) implies that Π_{i1} and Π_{i2} are assumed not to be parallel to the X-axis, and Π_{i1} passes through the origin of the model (object-centred) coordinate system (MCS). Let the corresponding image line l_i of L_i be specified by

$$a_i u + b_i v + c_i = 0 \tag{6}$$

where a_i, b_i and c_i are known constants. Substituting u and v defined in (2) into (6) and writing the resultant equation in terms of model coordinates (x_m, y_m, z_m) yields [16]

$$\delta_i x_m + \varepsilon_i y_m + \zeta_i z_m + \eta_i = 0 \tag{7}$$

where

$$\begin{aligned}
\delta_i &= (\vec{r}_1 \bullet \vec{n}_i)\cos\theta + (\vec{r}_2 \bullet \vec{n}_i)\sin\theta \\
\varepsilon_i &= (\vec{r}_2 \bullet \vec{n}_i)\cos\theta - (\vec{r}_1 \bullet \vec{n}_i)\sin\theta \\
\zeta_i &= \vec{r}_3 \bullet \vec{n}_i \\
\eta_i &= (\vec{r}_1 \bullet \vec{n}_i)T_x + (\vec{r}_2 \bullet \vec{n}_i)T_y + \vec{r}_4 \bullet \vec{n}_i \\
\vec{n}_i &= (a_i \;\; b_i \;\; c_i)^T \\
\vec{r}_k &= (m_{k1} \;\; m_{k3} \;\; m_{k4})\,; \quad k = 1, 2, 3, 4
\end{aligned} \tag{8}$$

Equation (7) defines the plane Π_i in the MCS on which L_i must lie. We call this plane the *interpretation plane* of L_i. The normal vector of Π_i is $\vec{N}_{\Pi i} = (\delta_i \;\; \varepsilon_i \;\; \zeta_i)^T$.

In terms of the relationship between L_i and its interpretation plane Π_i, we have the following simple geometrical observations:

Observation I: *If L_i is described by (3), then $\vec{P}_{i1}$ and $\vec{P}_{i2}$ must lie on Π_i.*

Observation II: *If L_i is described by (4), then $\vec{N}_i$ must be normal to $\vec{N}_{\Pi i}$, and $\vec{P}_i$ must be on Π_i.*

Observation III: *If L_i is described by (5), then Π_{i1}, Π_{i2} and Π_i must all be members of the pencil of planes containing L_i.*

These three observations suggest three different algorithms for obtaining very similar pose constraint equations (see Equations (10), (12) and (16)). The algorithms differ only in the numerical stability of the calculation for deriving the coefficients of the equations.

From Observation I and for N 2D-3D line correspondences, $L_i \leftrightarrow l_i$, $i = 1, 2, \ldots, N$, we have

$$\begin{cases} \delta_i x_{i1m} + \varepsilon_i y_{i1m} + \zeta_i z_{i1m} + \eta_i = 0 \\ \\ \delta_i x_{i2m} + \varepsilon_i y_{i2m} + \zeta_i z_{i2m} + \eta_i = 0 \end{cases}, \quad \forall i \in \{1, 2, \ldots, N\} \tag{9}$$

which, by recalling the variables defined in (8) and by isolating the pose parameters, can be rewritten as

$$\begin{cases} A_{i1}\cos\theta + B_{i1}\sin\theta + C_{i1}T_x + D_{i1}T_y = E_{i1} \\ \\ A_{i2}\cos\theta + B_{i2}\sin\theta + C_{i2}T_x + D_{i2}T_y = E_{i2} \end{cases}, \quad \forall i \in \{1, 2, \ldots, N\} \tag{10}$$

where A_{ij}, B_{ij}, C_{ij}, D_{ij} and E_{ij} are terms computable from known constants [16].

From Observation II and for N 2D-3D line correspondences, $L_i \leftrightarrow l_i$, $i = 1, 2, \ldots, N$, we can write

$$\begin{cases} \alpha_i \delta_i + \beta_i \varepsilon_i + \gamma_i \zeta_i = 0 \\ \\ \delta_i x_{im} + \varepsilon_i y_{im} + \zeta_i z_{im} + \eta_i = 0 \end{cases}, \quad \forall i \in \{1, 2, \ldots, N\} \tag{11}$$

which, by using the variables defined in (8) and by isolating the pose parameters, can be equivalently written as

$$\begin{cases} F_i\cos\theta + G_i\sin\theta = H_i \\ A_i\cos\theta + B_i\sin\theta + C_iT_x + D_iT_y = E_i \end{cases}, \quad \forall i \in \{1, 2, \ldots, N\} \tag{12}$$

where A_i, B_i, C_i, D_i, E_i, F_i, G_i and H_i are terms computable from known variables [16].

To show how Observation III can be used to derive pose constraints, we represent planes Π_{i1}, Π_{i2} and Π_i by three 4x1 column vectors $\vec{\Pi}_{i1}$, $\vec{\Pi}_{i2}$ and $\vec{\Pi}_i$:

$$\begin{aligned} \Pi_{i1}:&\quad \vec{\Pi}_{i1} = (1 \;\; b_{i1} \;\; c_{i1} \;\; 0)^T \\ \Pi_{i2}:&\quad \vec{\Pi}_{i2} = (1 \;\; b_{i2} \;\; c_{i2} \;\; d_{i2})^T \\ \Pi_i:&\quad \vec{\Pi}_i = (\delta_i \;\; \varepsilon_i \;\; \zeta_i \;\; \eta_i)^T \end{aligned} \tag{13}$$

then Observation III implies that there exist real λ_1, λ_2 and λ_3 (not all equal to zero) such that [14]

$$\lambda_1\vec{\Pi}_{i1} + \lambda_2\vec{\Pi}_{i2} + \lambda_3\vec{\Pi}_i = 0 \tag{14}$$

from which the following two pose constraint equations can be derived (for details, see [16]):

$$\begin{aligned} \bar{F}_i\cos\theta + \bar{G}_i\sin\theta &= \bar{H}_i \\ \bar{A}_i\cos\theta + \bar{B}_i\sin\theta + \bar{C}_iT_x + \bar{D}_iT_y &= \bar{E}_i \end{aligned} \tag{15}$$

where $\bar{A}_i$, $\bar{B}_i$, $\bar{C}_i$, $\bar{D}_i$, $\bar{E}_i$, $\bar{F}_i$, $\bar{G}_i$ and $\bar{H}_i$ are terms computable from known constants such as the coefficients of the two intersecting planes [16]. Thus for N line correspondences, $L_i \leftrightarrow l_i$, $i = 1, 2, \ldots, N$, there are $2N$ equations of the form of (15):

$$\begin{cases} \bar{F}_i\cos\theta + \bar{G}_i\sin\theta = \bar{H}_i \\ \bar{A}_i\cos\theta + \bar{B}_i\sin\theta + \bar{C}_iT_x + \bar{D}_iT_y = \bar{E}_i \end{cases}, \quad \forall i \in \{1, 2, \ldots, N\} \tag{16}$$

It is clear from (10), (12) and (16) that, no matter what line representation method is used, we need $2N \geq 3$ or a minimum of two 2D-3D line correspondences to ensure that the number of equations is no less than the number of unknowns. This contrasts with the minimum of three lines in the general case of six degrees of freedom [6-8, 10-11].

3.1 The linear solution technique

If $\cos\theta$ and $\sin\theta$ are regarded as two independent unknowns, then each of (10), (12) and (16) becomes a set of $2N$ linear equations in four unknowns: $\cos\theta$, $\sin\theta$, T_x and T_y, and can be solved by the standard linear least squares technique [16]. We then compute the rotation angle by $\theta = \tan^{-1}(\sin\theta/\cos\theta)$. The correct quadrant of θ can be determined from the senses of $\cos\theta$ and $\sin\theta$.

3.2 The iterative solution technique

If $\cos\theta$ and $\sin\theta$ are not treated as two independent unknowns, then each of (10), (12) and (16) is a set of $2N$ non-linear equations in the three pose parameters θ, T_x and T_y. If an approximate value for θ is given, we can linearize $\cos\theta$ and $\sin\theta$, and thus transform each of (10), (12) and (16) into a set of $2N$ linear equations which can then be solved using an iterative linear least squares technique [16].

Several remarks can be made before concluding this section. The concept of pencil of planes has been used in structure from motion algorithms (e.g., [15]), but has not appeared in previous pose recovery algorithms to the best of our knowledge. (12) and (16) show that the recovery of the rotation angle and the translational parameters can be decoupled since the first N equations of (12) and (16) only involve the rotation angle.

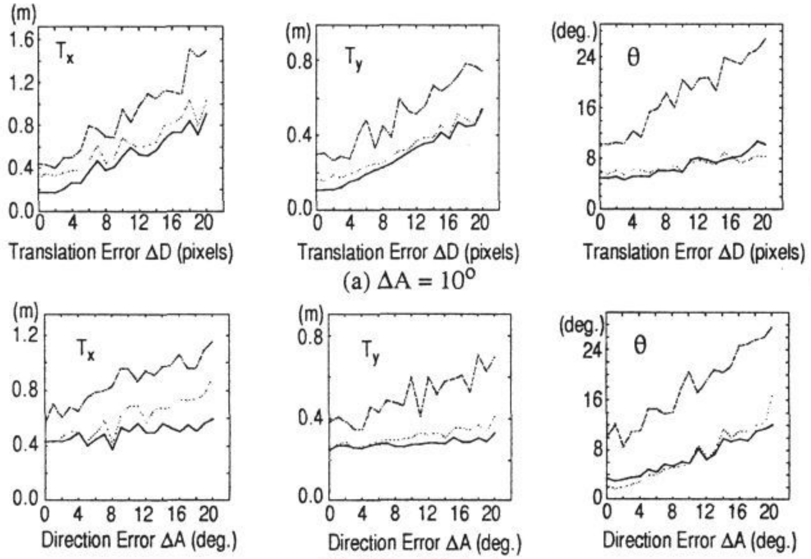
4 Summary of experimental results

The algorithms described in the above section have been tested under a variety of conditions. For convenience, we call the linear solution technique associated with the two points representation, the directional vector and point representation, and the intersecting planes representation of 3D lines, Algorithms A, B and C respectively. They are applied to solve (10), (12) and (16) respectively. Due to space limitation, full and detailed experimental results are not included here but can be found in [16].

4.1 Robustness against image measurement errors

The robustness of the algorithms against image measurement errors has been investigated using synthetic data. Lines were randomly generated from within a cuboid of dimension 8x4x2 m^3 (=length*width*height) located on the ground plane with a depth of about 20 meters from the camera. Images were of size 512x512 pixels. To generate a noisy 2D-3D line correspondence, each 2D noise-free line segment was first translated in its normal direction by Δd pixels, and then rotated about its middle point by $\Delta\alpha$ degrees, where Δd was uniformly distributed over $[-\Delta D, +\Delta D]$ (in pixels), and $\Delta\alpha$ uniformly distributed over $[-\Delta A, +\Delta A]$ (in degrees). The combination $(\Delta D, \Delta A)$ determines the noise level of the synthetic image data. Monte Carlo simulations were conducted at various noise levels with a fixed number (=10) of lines, and the absolute errors in the three pose parameters were recorded. As an example, Figure 2 shows the absolute error curves of the pose parameters vs. translational (Fig.2(a)) and directional (Fig.2(b)) errors of 2D image lines. (Tests carried out under a wider range of conditions are reported in [16]). Comprehensive Monte Carlo simulation results show that

- The performances of Algorithms A and B are very similar.
- Pose estimation by Algorithm C is much less accurate than that of Algorithms A and B. This may be due to the assumptions made in (5). If the randomly generated 3D lines during Monte Carlo simulation do not satisfy these assumptions, then the resultant representation of the two intersecting planes becomes very unstable, and may cause large errors in the recovered pose. Such *accidental* large errors can easily be avoided by adopting more stable representations for the intersecting planes.



(a) $\Delta A = 10^o$

(b) $\Delta D = 10$ pixels

Figure 2. *Absolute error curves of pose parameters in Algorithms A (solid), B (light), and C (dark).*

- As a whole, the proposed algorithms are robust to image measurement errors. Even at unrealistically high noise levels $(\Delta D = 20 \text{ pixels}, \Delta A = 20^o)$, the mean absolute errors of Algorithms A and B are below 1.20 meters in T_x, 0.60 meter in T_y, and 22^o in θ.

As an experiment, the iterative solution technique described in Section 3.2 was also applied to solve (10), using the rotation angle given by Algorithm A as starting conditions. Monte Carlo simulation results [16] show that under severe noise conditions, the iteration actually increased the error. This illustrates the effectiveness of the (simpler) linear solution technique. It also provides a clear case of Aggarwal's taunt that "Often it is better to keep [a] good initial guess and forget about the [non-linear constraint] equations!" [13].

4.2 Effectiveness of using more lines in controlling noise

Monte Carlo simulations were also conducted to investigate the effectiveness of using more line correspondences in combating noise. The noise level was fixed at $\Delta D = 5$ pixels and $\Delta A = 5^o$, and the number of lines was increased from the minimum of 2 to 40. The results are summarized in Fig.3. As expected, the accuracy of the estimated pose parameters is consistently improved by using more lines. The improvement is most dramatic when the number of line correspondences is increased from the minimum of 2 to 4 or 5, and beyond 8 there is little improvement. Therefore 5 to 8 line correspondences may be regarded as adequate for robust pose estimation using these algorithms.

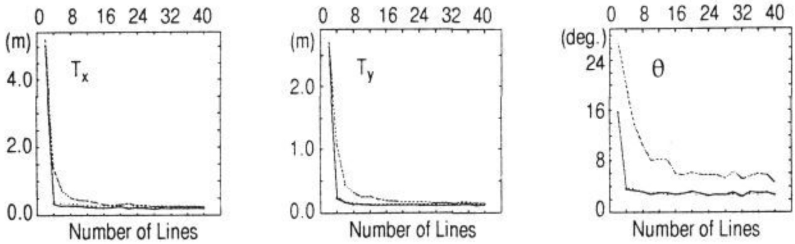


Figure 3. *Effectiveness of using more lines in improving the noise-robustness of Algorithms A (solid), B (light), and C (dark).*

4.3 Performance under real outdoor images

All three algorithms have also been applied to locate a saloon car in 9 frames taken at 0.48 Hz (i.e. every 12th video frame) of an outdoor image sequence. Because of space limitation, only results related to Algorithm B are included. Three out of the nine frames are shown in the left column of Fig.4, where the white car is the object to be located. The gradient images inset in the left column of Fig.4 are the outputs of the Canny operator applied to the small regions surrounding the car. Thresholding and a simple curve segmentation operation were then applied to the gradient images to obtain a set of straight line segments. A subset of these segments were retained which correspond to contour segments of the car, and are inset in the right column of Fig.4. The number of the retained line segments is between 5 and 10. No efforts were made to "tune" the various parameters (such as the scale of the Canny operator) involved in the iconic to symbolic data transformation. The correspondences between the selected 2D lines and the 3D

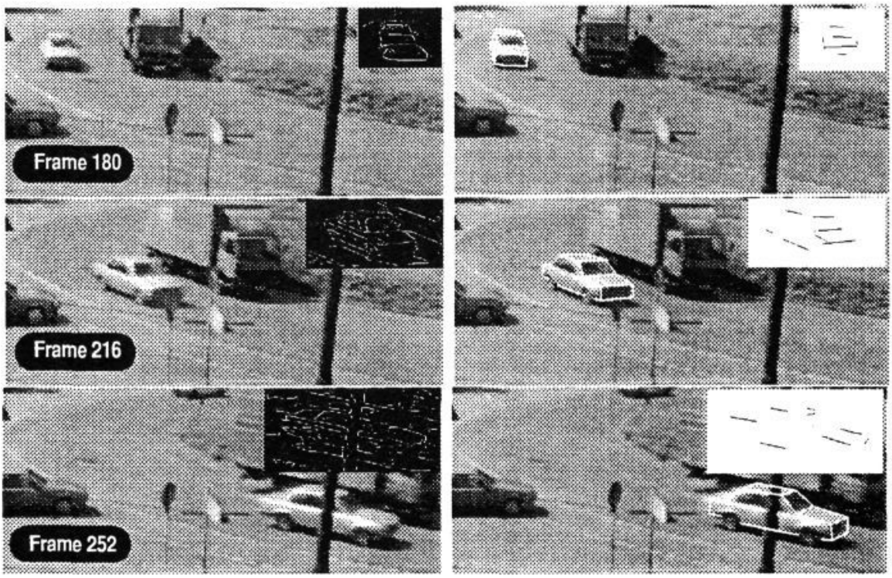


Figure 4. *Pose estimation in real outdoor images (see text for detailed captions).*

saloon model lines were established manually. These correspondences were used as the input to the pose recovery algorithms. The saloon model was then instantiated at the recovered poses, and superimposed on the original intensity images as shown in the right column of Fig.4. The accurate matching between the model and the image shown in Fig.4 indicates the high accuracy of the recovered poses. The performance of the proposed algorithms can be further appreciated from Fig.5 which shows the recovered X (i.e, T_x) and Y (i.e., T_y) coordinates of the car on the ground plane for all 9 frames. The recovered orientations of the car are also shown in Fig.5. The resultant path is very smooth and is physically plausible.

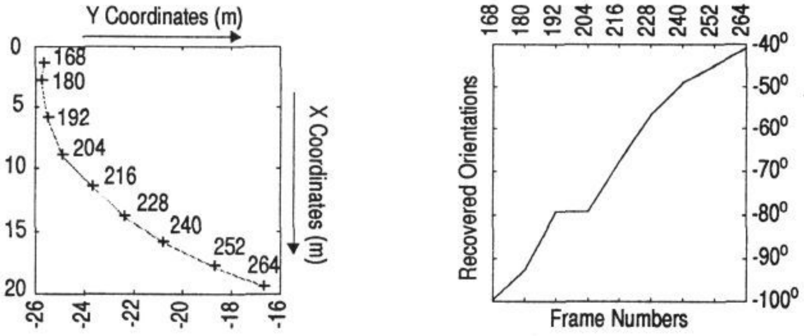


Figure 5. *Recovered car path on the ground plane. The numbers along the curve are the frame numbers, and the symbol "+" marks the recovered poses.*

5 Conclusions

This work has been concerned with object pose estimation under the ground plane constraint. A number of novel algorithms have been presented which make use of the constraint. The algorithms are linear, computationally simple, and give unique and closed-form solutions, thus eliminating many of the problems associated with the existing pose recovery algorithms. They require a minimum of two 2D-3D line correspondences, and are highly robust with 5 or more correspondences. The algorithms are also applicable to point correspondences.

Experimental results show that the proposed algorithms are robust to noise, and capable of accurate pose recovery in real images of outdoor scenes. The algorithms provide practical and efficient methods for pose recovery which can be applied in a wide range of industrial applications in which the objects move on a known plane.

References

[1] T. M. Silberberg, D. A. Harwood and L. S. Davis, Object Recognition Using Oriented Model Points, CVGIP, vol.35, 1986, pp.47-71.

[2] R. M. Haralick, et. al., Pose Estimation from Corresponding Point Data, IEEE Trans. Systems, Man and Cybern., vol.19, 1989, pp.1426-1446.

[3] S. Ullman and R. Basri, Recognition by Linear Combinations of Models, PAMI,

vol.13, 1991, pp.992-1006.

[4] S. T. Barnard, Interpreting Perspective Images, Artif. Intell., vol.21, 1983, pp.435-462.

[5] D. G. Lowe, Three-Dimensional Object Recognition from Single Two-Dimensional Images, Artif. Intell., vol.31, 1987, pp.355-395.

[6] D. G. Lowe, Fitting Parameterized Three-Dimensional Models to Images, PAMI, vol.13, 1991, pp.441-450.

[7] M. Dhome, et. al., Determination of the Attitude of 3-D Objects from a Single Perspective View, PAMI, vol.11, 1989, pp.1265-1278.

[8] S. Linnainmaa, D. Harwood and L. S. Davis, Pose Determination of a Three-Dimensional Object Using Triangle Pairs, PAMI, vol.10, 1988, pp.634-647.

[9] Y. C. Liu, T. S. Huang and O. D. Faugeras, Determination of Camera Location from 2-D to 3-D Line and Point Correspondences, PAMI, vol.12, 1990, pp.28-37.

[10] A. D. Worrall, K. D. Baker and G. D. Sullivan, Model Based Perspective Inversion, Image and Vision Comput., vol.7, 1989, pp.17-23.

[11] R. Horaud, New Methods for Matching 3-D Objects with Single Perspective Views, PAMI, vol.9, 1987, pp.401-412.

[12] C. Goad, Special Purpose Automatic Programming for 3D Model-Based Vision, in From Pixels to Predicates, A. Pentland, Ed., Norwood, NJ: Ablex, 1986, pp.371-391.

[13] J. K. Aggarwal and A. Mitiche, Structure and Motion from Images: Fact and Fiction, 3rd IEEE Workshop on Vision: Representation and Control, October 1985, pp.127-128.

[14] R. Hartshorne, Foundations of Projective Geometry, Benjamin, 1967.

[15] O. D. Faugeras, F. Lustman and G. Toscani, Motion and Structure from Motion from Point and Line Matches, Proc. of 1st Int. Conf. Comput. Vision, London, 1987, pp.25-34.

[16] T. N. Tan, Locating Objects on the Ground Plane Using 2D-3D Line Correspondences, ESPRIT II Project (P2152) Research Report, RU-03-WP.T3137-TNT-01, January 1992.

[17] J. L. Mundy and A. J. Heller, The Evolution and Testing of A Model-Based Object Recognition System, Proc. of ICCV90, December 4-7, 1990, Osaka, Japan, pp.268-282.

[18] T. N. Tan, K. D. Baker and G. D. Sullivan, 3D Structure and Motion Estimation from 2D Image Sequences, Proc. of British Machine Vision Conference 1992, Springer-Verlag, 1992.