

Structure and Motion from Straight Line Segments

Abstract

A method to determine both the camera location and scene structure from image straight segment correspondences is presented. The proposed method considers the finite segment length in order to use stronger constraints than do those that use the infinite line that supports the image segment. The constraints between image segments involve a *weak* pairing between image segment midpoints. This allows deviations of the midpoint only in the segment direction. Experimental results are presented of structure and motion computations from the image straight line segment-matching using two real images.

Key words: structure and motion, straight segments.

1 Introduction

Structure and motion computing from correspondent straight features is a classical problem in computer vision. Usually, the straight image segments are modelled as their underlying infinite supporting line. This model can be easily represented with the available mathematical representations for lines. Its main pitfall, compared with the use of points, is that the constraints for lines are less restrictive. A review of structure and motion problems using points and lines is presented by [1]. In this, the usage of points is shown to be more restrictive, always.

Most man-made environments can be represented with line segments. This makes it possible to enforce the previous knowledge of the straightness of the contours. Normally, relevant image segments correspond to relevant 3D scene features. It should be noted that the semantically relevant feature is not the infinite supporting line of the segment but the finite length segment. Despite this, only a few works ([2,3]) have proposed models that include the finite length of the line segments in structure and motion computing. The consideration of the segment finite length has renewed interest in the usage of straight features in structure and motion problems.

Taylor and Kriegman [3] proposed an optimization procedure for image segments. The goal function is the image residue between the image infinite supporting line and the reconstructed 3D segment supporting line. The segment finite length is considered because the residue between the supporting lines is only computed in the region of the supporting line where the image segment is detected. They propose an algorithm to compute the structure and motion from straight segment correspondences in a sequence with no previous knowledge of camera location. The evaluation of the residues only in the detected image segments improves the solution but does not constrain the motion along the infinite supporting line. Consequently, at least three images are needed to compute the solution.

Zhang [2] proposed, for the first time, an algorithm to compute structure and motion from two images using only straight segment matching as the input. He proposed to maximize the overlap in the image between the image segments and the corresponding reconstruction by using epipolar geometry, and as a result, reconstruction computation was not required. Mathematically, the problem was reduced to a non-linear optimization problem. To compute the initial guess for non-linear optimization, he proposed to sample the solution space and use a few high overlapping solutions as starting points for further optimization.

The work presented in this paper is closely related to that of Zhang. Experimental results are presented of structure and motion computation using straight segment correspondences and two images. The solution is also presented as an optimization problem and the initial guess is also computed by sampling the parameter space. This paper differs in the image optimization by considering the image segment midpoints as correspondent. However, this constraint is considered only weakly and deviations are allowed for along the image segment direction. It is well known that image segment midpoints are not correspondent due to the unreliable extraction of the segment extreme points, as well as, the segment midpoint not being invariant under perspective projection. However, the detected direction for the image segment is reliable. The proposed weak pairing between image segment midpoints uniquely combines the line and point properties of the image segments. A reconstruction and reprojection scheme is proposed, which considers any number of images greater than or equal to two.

Section 2 presents the probabilistic model used to represent the imprecise location of the geometrical entities. Section 3 presents the equation that relates the image segment location, the camera location and the 3D segment location. In section 4 the solution of the structure and motion problem from straight correspondences is posed as a non-linear optimization. Section 5 presents the method in which initial guesses for the non-linear optimization with two images are computed. Experimental results with a pair of real images are pre-

sented in section 6. Finally section 7 is devoted to conclusions.

2 Modelling Geometric Information

A probabilistic model, named SPmodel [4], has been selected to represent the imprecise geometrical information. This is a general model for multi-sensor fusion mainly with the following qualities: homogeneous representation for every feature irrespective of the number of degrees of freedom (d.o.f for short in the rest of the paper), and a local representation for the error around the feature location estimate. The error is not represented additively, but as a transformation composition. These qualities for multi-sensor fusion are also recognized as important for computer vision in [5].

Moreover, a modification has been introduced in the original SPmodel that combines unprecise relations with deterministic relations. Sensorial information is imprecise due to measurement errors. However, some relations are known with a probability of one, e.g. a projection ray is known to pass through the optical centre of the camera.

The SPmodel is a probabilistic model that associates a reference G to locate each geometric element \mathcal{G} . The reference location is given by the transformation t_{WG} relative to a world reference W . To represent this transformation, a *location vector* \mathbf{x}_{WG} , composed of three Cartesian coordinates and three Roll-Pitch-Yaw angles, is used:

$$\begin{aligned}\mathbf{x}_{WG} &= (x, y, z, \psi, \theta, \phi)^T \\ t_{WG} &= \text{Trans}(x, y, z) \cdot \text{Rot}(z, \phi) \cdot \text{Rot}(y, \theta) \cdot \text{Rot}(x, \psi)\end{aligned}\quad (1)$$

The estimation of the location of an element is denoted by $\hat{\mathbf{x}}_{WG}$, and the estimation error is represented *locally by a differential location vector* \mathbf{d}_G relative to the reference attached to the element. Thus, the true location of the element is:

$$\mathbf{x}_{WG} = \hat{\mathbf{x}}_{WG} \oplus \mathbf{d}_G$$

where \oplus represents the composition of location vectors (the inversion is represented by \ominus). Note that the error is composed with the location estimate, instead of being added. The differential location error \mathbf{d}_G is a normally distributed random vector with a dimension of six. Some components of \mathbf{d}_G are set to zero in the following two cases:

symmetries .- Symmetries are the set of transformations that preserve the

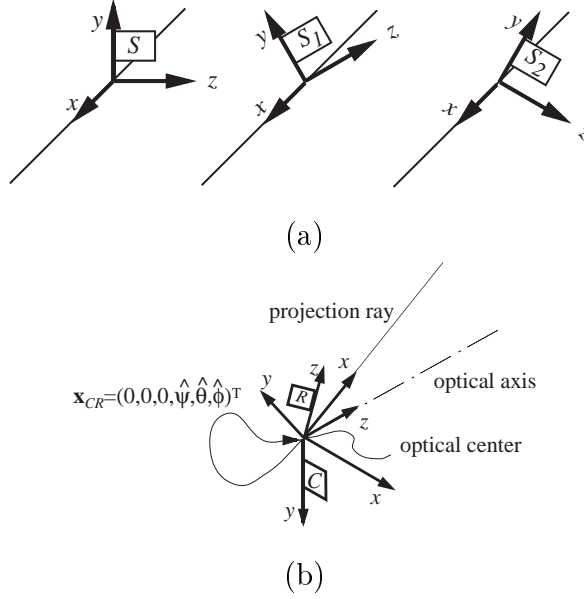


Fig. 1. (a) Several equivalent associated references, S , for a 3D segment. (b) A projecting ray R , located with respect to the optical centre C .

element. The location vector \mathbf{x}_{WG} represents the same element location irrespective of that \mathbf{d}_G component value. For example, consider the reference S , which locates a 3D segment (Fig. 1 (a)). Rotations around the X direction yield references that represent the same 3D segment. Theoretically those components could have any value, but a zero value is used.

deterministic components .- There are components of \mathbf{x}_{WG} known to have a probability of one. Among all the equivalent references for the element, the one with a null deterministic component is selected. The corresponding component is then forced to be zero. For example (Fig. 1 (b)) the reference, R , associated with a projection ray; R can be attached to the optical centre, expressing its location with respect to the frame of the optical centre C , \mathbf{x}_{CR} (and hence \mathbf{d}_G) always has null X , Y , and Z components.

To represent null components mathematically, \mathbf{d}_G is expressed as:

$$\mathbf{d}_G = B_G^T \mathbf{p}_G$$

where \mathbf{p}_G , the *perturbation vector*, is a vector containing only non-null components of \mathbf{d}_G . B_G , the *self-binding matrix*, is a row selection matrix that selects the non-null components of \mathbf{d}_G .

Based on these ideas, the information about the location of geometric element

\mathcal{G} is represented by a quadruple, the element *imprecise location*:

$$\mathbf{L}_{WG} = [\hat{\mathbf{x}}_{WG}, \hat{\mathbf{p}}_G, C_G, B_G]$$

Thus, the random vector defining the element location is expressed as:

$$\mathbf{x}_{WG} = \hat{\mathbf{x}}_{WG} \oplus \mathbf{B}_G^T \mathbf{p}_G \quad (2)$$

$$\hat{\mathbf{p}}_G = E[\mathbf{p}_G]; C_G = Cov(\mathbf{p}_G)$$

where \mathbf{p}_G is a normal random vector, with a mean of $\hat{\mathbf{p}}_G$ and a covariance matrix C_G . When $\hat{\mathbf{p}}_G = 0$, the estimation is regarded as centred.

There are geometrical elements whose location is used as input data for the problem, e.g. the image segments. The 3D segments and the cameras location are the output of the algorithm and are computed from the input data. The 3D segment is the geometrical element used to represent the scene structure. Moreover, we define an intermediate geometrical element, the *2D segment*, to define the constraint that relates an image segment to a 3D segment and the camera location. The 2D segment consists of the projection ray for the image segment midpoint, and the projection plane for the image segment supporting line. Next, the model of these four elements is presented in detail.

2.1 Camera imprecise location

We use the letter C to designate the camera reference, and model the camera as normalized, i.e. $f = 1$. The associated reference origin is joined to the optical centre of the camera. The z axis is parallel to the optical axis, and pointing towards the scene. The x axis points to the right. The y axis is defined to be a direct reference (see Fig.1(b)). Camera location has neither symmetry nor deterministic components in its differential location vector, so has no null components (its self-binding matrix is the identity):

$$B_C = I$$

Camera location estimate $\hat{\mathbf{x}}_{WC}$ is obtained from the structure and motion algorithm.

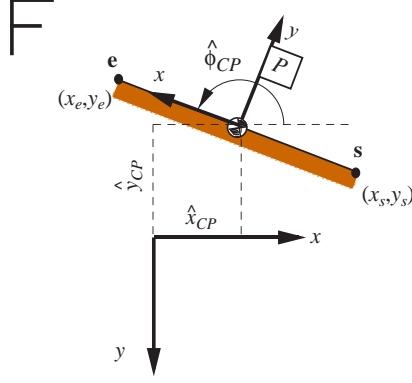


Fig. 2. Image segment in the normalized camera.

2.2 Image segment imprecise location

We use the letter P to designate references attached to image segments. The associated reference (Fig.2) is attached to the image segment midpoint. Its y axis is normal to the supporting line and pointing to the “light” side of the segment. Consequently, it codes the grey level gradient of the segment. Its z axis is parallel to the camera z axis. The x axis is defined to be a direct reference.

As an image segment belongs to the image plane, its z , ψ , and θ components are deterministic. Therefore, its self-binding matrix is:

$$B_P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The image segment location centred estimate is defined for the extreme points coordinates in the normalized image, with respect to the camera location, as (see Fig.2):

$$\hat{\mathbf{x}}_{CP} = (\hat{x}_{CP}, \hat{y}_{CP}, 1, 0, 0, \hat{\phi}_{CP})^T \quad (3)$$

$$\hat{\phi}_{CP} = \text{atan2}(y_e - y_s, x_e - x_s); \quad \hat{x}_{CP} = \frac{x_s + x_e}{2}; \quad \hat{y}_{CP} = \frac{y_s + y_e}{2}$$

The covariance assignment for an image segment is one of the central issues of this work. The covariance in the ϕ and y components is taken from the infinite supporting line. The standard deviation of the x component is defined to be proportional to the segment length. According to the proportionality constant κ , the allowed deviations of the midpoint along the segment supporting line

can be fixed. This covariance assignment mimics that proposed by [6], where it is used for 3D segments. In this paper, we use it for image segments. The same covariance assignment is also used by [7] for structure from camera location.

The quantitative expression for the covariance matrix is given by:

$$C_P = NC'_P N^T \quad (4)$$

where C'_P is the covariance for the image segment in pixels; N is the Jacobian matrix for the transformation that converts image segment in pixels to the image segment in the normalized camera. We have chosen the above expression in order to deal with the pixel aspect ratio. Appendix B gives a detailed expression for N as a function of the image segment location estimate (3), and the intrinsic camera parameters.

The form for C'_P is:

$$C'_P = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_\phi^2)$$

- σ_x is set proportional to the image segment length, n (in pixels).

$$\sigma_x = \kappa n \quad (5)$$

The experimental values for κ have been tuned to give $\kappa > 1$.

- σ_y^2 and σ_ϕ^2 are computed from the covariances of the image segment extreme points. Due to systematic errors, there is some correlation between the noise from the extreme points location. This effect is dealt with by splitting the extreme points covariance into two terms: σ_{cc}^2 completely correlated covariance (0 – 2px.), and σ_{nc}^2 non-correlated covariance (0.25 – 1px.). In [8] both the expression and tuning are given

$$\sigma_y^2 = \sigma_{cc}^2 + \frac{\sigma_{nc}^2}{2}, \quad \sigma_\phi^2 = \frac{2\sigma_{nc}^2}{n^2} \quad (6)$$

Figure 3 shows a comparison between the 95% acceptance regions for the origin of the reference that locates the image segment according to two different κ assignments.

It should be noted that the segment length is not considered as a geometrical parameter, but is used to define the element covariance. The segment is only located by its midpoint and its orientation. Intuitively, the image segment has been modelled as “a point with an orientation,” and its standard deviation along the segment supporting line is set proportional to its length.

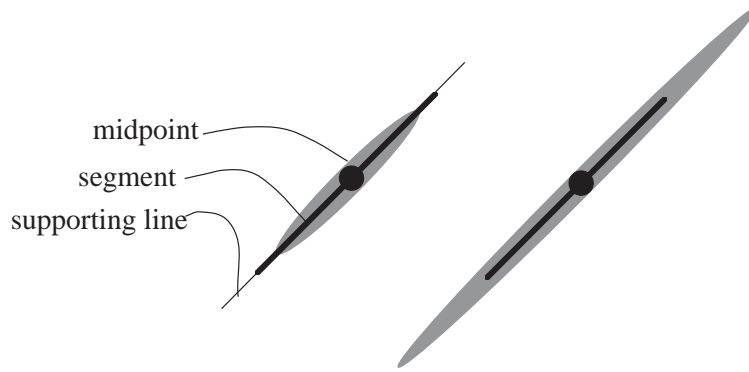


Fig. 3. 95% acceptance regions for the image segment reference origin. Left, $\kappa = 0.2$, right $\kappa = 0.5$

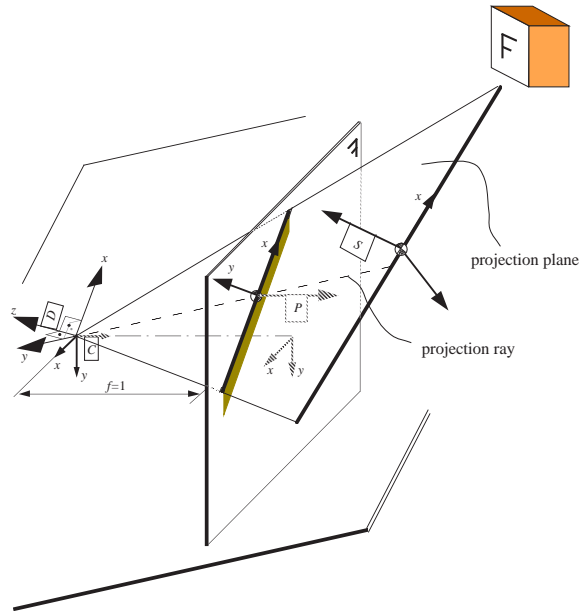


Fig. 4. The 2D segment (D) is an intermediate element used to relate the image segment (P) to the 3D segment (S). It includes both the projection ray for the midpoint and the projection plane for the supporting line.

2.3 2D segment

We use D to designate references attached to 2D segments. This geometrical element is used as an intermediate element to define the relation between an image segment and a 3D segment (Fig. 4). A 2D segment is composed of the projection elements of the corresponding image segment: the projection ray for the image segment midpoint, and the projection plane for the infinite supporting line. Its covariance is derived directly from that of the image segment.

The associated reference is attached to the optical centre of the camera, which

belongs to every projection element. Its $-y$ axis points towards the image segment midpoint. The z axis is normal to the supporting line projection plane. The x axis forms a direct reference. The z direction is defined to code the image segment grey level gradient. Since it is attached to the optical centre, the general location vector of a 2D segment, with respect to the camera frame, is:

$$\hat{\mathbf{x}}_{CD} = (0, 0, 0, \hat{\psi}_{CD}, \hat{\theta}_{CD}, \hat{\phi}_{CD})^T$$

See Appendix C for $\hat{\psi}_{CD}$, $\hat{\theta}_{CD}$ and $\hat{\phi}_{CD}$ expressions as function of image segment location (3).

As the translation components are deterministically null, the self-binding matrix only selects ψ , θ and ϕ components:

$$B_D = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The 2D segment covariance matrix is related to that of the image segment:

$$C_D = K_{DP} C_P K_{DP}^T$$

where C_P has been presented in (4). A detailed expression for the K_{DP} matrix as a function of the image segment location vector is given in Appendix C.

2.4 3D segment

The references associated with 3D segments are designated by S . The reference is attached (Fig. 1 (a) and Fig. 4) to a segment point, which approximately corresponds to the segment midpoint. The reference x axis is aligned with the segment direction. In this work, the 3D segment location estimate is computed from the integration of several 2D segments corresponding to different points of view. This integration also gives the 5×5 covariance matrix C_S .

The only symmetry for this element is the rotation around its direction. There-

fore, its self-binding matrix is:

$$B_S = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

3 Measurement Equation

This section is devoted to formalizing the pairing constraint between an image segment (P) and a 3D segment (S) (see Fig. 4). The camera detects the image segment (P). However, the proposed pairing constraint uses the 2D segment (D) and not the image segment (P). As mentioned in Section 2.3, the 2D segment imprecise location, is derived directly from that of the image segment.

The SPmodel method is used to define pairing constraints, which are defined in terms of the location vector \mathbf{x}_{DS} of the 3D segment with respect to the 2D segment. Let

$$\mathbf{x}_{DS} = (x_{DS}, y_{DS}, z_{DS}, \psi_{DS}, \theta_{DS}, \phi_{DS})^T$$

The pairing constraint is an implicit equation that indicates which \mathbf{x}_{DS} components should be zero; these null components are as follows.

- z_{DS} . Otherwise, the 3D segment would not belong to the projection plane.
- θ_{DS} , rotation around y axis. This should be zero, otherwise the 3D segment would not be in the projection plane.
- x_{DS} . Otherwise, the 3D segment midpoint would not belong to the image segment midpoint projection ray. Theoretically, *the segment midpoint is not invariant under perspective projection. However, we consider it as invariant, but it is shown later that this constraint is considered to have a low weighting.* Consequently, Fig. 4 the projection ray for the image segment midpoint does not contain the origin of S reference.

In summary, the nullity of z_{DS} and θ_{DS} takes into account the collinearity in the image between the image segment and the 3D segment. The nullity of x_{DS} implies that the image of the 3D segment midpoint is the midpoint of the image segment. This is equivalent to considering that the image segment midpoints are correspondent. Due to the assignment for image segment covariance along

the segment direction, the midpoint matching normally has lower weighting than the collinearity. The low weighting for this constraint is justified by the unreliable segment extreme points extraction, and by the approximation that the midpoint is invariant under perspective projection. Experimental results confirm the validity of this assignment.

The above ideas are formalized mathematically as follows:

$$\mathbf{f}(\mathbf{x}_{DS}) = B_{DS}\mathbf{x}_{DS} = 0 \quad (7)$$

$$B_{DS} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

considering that:

$$\mathbf{x}_{DS} = \ominus\mathbf{x}_{CD} \ominus \mathbf{x}_{WC} \oplus \mathbf{x}_{WS} \quad (8)$$

equation (7) establishes a relationship between the 3D segment location, the camera location and the image of the segment. The image segment is represented by the 2D segment (D). This equation is used to determine the structure and motion from the correspondences.

From equations (8) and (2), equation (7) can be expressed as:

$$\mathbf{f}(\mathbf{x}_{DS}) = \mathbf{f}(\mathbf{p}_D, \mathbf{p}_C, \mathbf{p}_S) = B_{DS} \left(\ominus B_D^T \mathbf{p}_D \ominus \hat{\mathbf{x}}_{CD} \ominus B_C^T \mathbf{p}_C \ominus \hat{\mathbf{x}}_{WC} \oplus \hat{\mathbf{x}}_{WS} \oplus B_S^T \mathbf{p}_S \right) = 0 \quad (9)$$

Consequently, we have an implicit function that relates three perturbation vectors, \mathbf{p}_D , \mathbf{p}_C , and \mathbf{p}_S corresponding to the 2D segment, the camera, and the 3D segment respectively, i.e. the normal random vectors involved in the problem.

4 Structure and Motion as a Weighted Minimization Problem

The structure and motion problem with m segments in n images can be stated as the solution to the non-linear system:

$$\mathbf{f}_i^{(k)} \left(\mathbf{x}_{D_i^{(k)} S_i} \right) = 0, \quad i = 1 \dots m, \quad k = 1 \dots n$$

where $\mathbf{f}_i^{(k)}$ is the measurement equation (7) between the 2D segment detected by camera k corresponding to the 3D segment i . Due to approximate midpoint matching and noise, the previous system is over-constrained. A minimization is proposed in order to solve it. The goal function is the summation of all the weighted residues in which the weighting matrix is the inverse of the measurement noise covariance:

$$\sum_{i=1}^m \sum_{k=1}^n \left[\mathbf{f}_i^{(k)} \left(\mathbf{x}_{D_i^{(k)} S_i} \right) \right]^T \left[\mathbf{R}_i^{(k)} \left(\mathbf{x}_{D_i^{(k)} S_i} \right) \right]^{-1} \left[\mathbf{f}_i^{(k)} \left(\mathbf{x}_{D_i^{(k)} S_i} \right) \right] \quad (10)$$

where $\mathbf{R}_i^{(k)} \left(\mathbf{x}_{D_i^{(k)} S_i} \right)$ is the measurement noise covariance. In general, this depends on $\mathbf{x}_{D_i^{(k)} S_i}$, which is the optimization variable.

Considering (8) and C_1 as the world frame, $\mathbf{x}_{D_i^{(k)} S_i}$ can be decomposed to:

$$\mathbf{x}_{D_i^{(k)} S_i} = \ominus \mathbf{x}_{C_k D_i^{(k)}} \ominus \mathbf{x}_{C_1 C_k} \oplus \mathbf{x}_{C_1 S_i}$$

then, the problem can be expressed as:

given: $\left\{ \mathbf{L}_{C_k D_i^{(k)}} \right\} \quad k = 1 \dots n \quad i = 1 \dots m$

determine: $\left\{ \mathbf{L}_{C_1 S_i} \right\}, \left\{ \mathbf{L}_{C_1 C_k} \right\} \quad k = 1 \dots n \quad i = 1 \dots m$, up to a scale factor, such that expression (10) is minimized.

where $\{ \}$ expresses a set of elements.

The previously stated problem optimizes both the structure and motion parameters. As a result, it needs to optimize $5 + 6(n - 2) + 5m$ parameters. These are: five parameters for the second camera location ($\mathbf{x}_{C_1 C_2}$), six for each of the remaining cameras $\left\{ \mathbf{x}_{C_1 C_k} \right\} \quad k = 3 \dots n$, and five for each scene 3D segment $\left\{ \mathbf{x}_{C_1 S_i} \right\} \quad i = 1 \dots m$. The result, however, can be computed by optimizing only the motion parameters as proposed in [9], in which a constraint between the structure and motion was used. The evaluation of the total residue is presented next only as a function of the motion. Given a value for the motion parameters $\left\{ \mathbf{x}_{C_1 C_k} \right\}$, the structure $\left\{ \mathbf{x}_{C_1 S_i} \right\}$ is computed using a structure from the camera location algorithm. The weighted residue is computed from the given motion and the computed structure (10). A diagram of the structure from the camera location algorithm is given in section 4.1. For normal and uncorrelated noise, the total residual (10) would follow a χ^2 distribution with

$$3mn - 5m - 6(n - 2) - 5 \quad \text{d.o.f.}$$

The rest of this section is devoted to defining the terms involved in (10).

The $\mathbf{R}_i^{(k)}$ matrix is computed from the linearization of equation (9):

$$\mathbf{R}_i^{(k)} = \mathbf{G}_i^{(k)} C_{D_i^{(k)}} \mathbf{G}_i^{(k)T}$$

$$\mathbf{G}_i^{(k)} = \left. \frac{\partial \mathbf{f}_i^{(k)}}{\partial \mathbf{p}_{D_i^{(k)}}} \right|_{\left(\mathbf{p}_{D_i^{(k)}}=0, \mathbf{p}_{C_k}=0, \mathbf{p}_{S_i}=0 \right)}$$

where:

$$\mathbf{G}_i^{(k)} = \begin{pmatrix} 0 & -\hat{z}_{DS} & \hat{y}_{DS} \\ -\hat{y}_{DS} & \hat{x}_{DS} & 0 \\ \sin \hat{\phi}_{DS} & -\cos \hat{\phi}_{DS} & 0 \end{pmatrix}$$

$C_{D_i^{(k)}}$ is the 2D segment location noise and is computed in section 2.3.

$\mathbf{f}_i^{(k)}$ is computed directly from $\hat{\mathbf{x}}_{D_i^{(k)} S_i}$ components. i and (k) scripts have been dropped for simplicity:

$$\mathbf{f}_i^{(k)} = \begin{pmatrix} \hat{x}_{CD} \\ \hat{z}_{CD} \\ \hat{\theta}_{CD} \end{pmatrix}$$

4.1 Structure from camera location

As previously mentioned, a structure from the camera location computation is proposed in order to evaluate the residue function (10).

The scene structure from the camera location and the correspondences are computed using LMSE solution for each 3D structure segment. It is computed from a linearized version of equation (9); the detailed algorithm for this structure computation is presented in [7]. It is given briefly by the following:

$$\mathbf{f} \left(\mathbf{p}_{D_i^{(k)}}, \mathbf{p}_{C_k}, \mathbf{p}_{S_i} \right) \approx \mathbf{f}_i^{(k)} + H_i^{(k)} \mathbf{p}_{S_i} + G_i^{(k)} \begin{pmatrix} \mathbf{p}_{D_i^{(k)}} \\ \mathbf{p}_{C_k} \end{pmatrix} = 0$$

expressed as the explicit linear measurement equation normally used in opti-

mal estimation:

$$\mathbf{z}_i^{(k)} = H_i^{(k)} \mathbf{p}_{S_i} + G_i^{(k)} \mathbf{v}, \quad \mathbf{v} \sim N(0, R_i^{(k)}) \quad (11)$$

The estimated value is \mathbf{p}_{S_i} , which is the correction for the 3D segment location. The camera and 2D segment location noise, \mathbf{p}_{C_k} and $\mathbf{p}_{D_i^{(k)}}$, play the role of measurement noise. Consequently, they are grouped in a measurement noise vector \mathbf{v} . The rest of the terms in (11) are:

$$\begin{aligned} \mathbf{z}_i^{(k)} &= -\mathbf{f}_i^{(k)} = -B_{DS} \hat{\mathbf{x}}_{D_i^{(k)} S_i} \\ H_i^{(k)} &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{p}_S} \right|_{\left(\mathbf{p}_{D_i^{(k)}}=0, \mathbf{p}_{C_k}=0, \mathbf{p}_{S_i}=0 \right)} \\ G_i^{(k)} &= \left(\left. \frac{\partial \mathbf{f}}{\partial \mathbf{p}_D} \right|_{\left(\mathbf{p}_{D_i^{(k)}}=0, \mathbf{p}_{C_k}=0, \mathbf{p}_{S_i}=0 \right)} \quad \left. \frac{\partial \mathbf{f}}{\partial \mathbf{q}_C} \right|_{\left(\mathbf{p}_{D_i^{(k)}}=0, \mathbf{p}_{C_k}=0, \mathbf{p}_{S_i}=0 \right)} \right) \\ \mathbf{v} &= \begin{pmatrix} \mathbf{p}_{D_i^{(k)}} \\ \mathbf{p}_{C_k} \end{pmatrix} \\ R_i^{(k)} &= \begin{pmatrix} C_{D_i^{(k)}} & 0 \\ 0 & C_{C_k} \end{pmatrix} \end{aligned}$$

$C_{D_i^{(k)}}$ is computed as shown in Sec.2.3, and C_{C_k} is tuned experimentally. Detailed expressions for the previous equations, as functions of the location estimates for the camera, 2D segment, and 3D segment, are available in Appendix D.

5 Initial Seed for the Two Images Problem

The main disadvantage of the non-linear optimization methods is that they need a good initial guess in order to converge to the absolute minimum of the goal function. This paper focuses on the two images problem. In order to find the minimum of equation (10), we compute the initial seed sampling the parameter space as proposed by Zhang in [2]. The whole minimization method can be summarized as follows.

- (1) Sample the parameter space. The sample points should approximately be uniformly distributed over the parameter space.
- (2) Evaluate the goal function (equation 10) on every point.

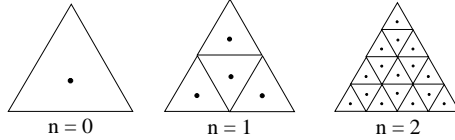


Fig. 5. Subdivision of an icosahedron face in order to obtain more direction samples

- (3) Keep the samples that produced the lowest residue (in our case we kept 30 samples).
- (4) Optimize the equation (10) with a classical optimization method (in our case Levenberg-Marquardt) using each of the kept samples as an initial guess. The minimum is then selected.

The rest of this section is devoted to the way in which the parameter space is sampled. Considering two images, the parameter to be optimized is the location of the second camera with respect to the first, i.e. $\mathbf{x}_{C_1C_2}$. Due to the translation scale factor, only the translation direction should be considered. Therefore, only five optimization parameters are considered:

- α_T, β_T , the two angles that define the translation direction.
- $\alpha_R, \beta_R, \gamma$, the first two angles define the rotation direction. γ defines the rotation magnitude.

from $\alpha_T, \beta_T, \alpha_R, \beta_R$ and γ the value for $\mathbf{x}_{C_1C_2}$ is straightforward. The rest of the section is devoted to the $\alpha_T, \beta_T, \alpha_R, \beta_R$ and γ sampling. Irrespective of whether the direction sampling α and β angles are for rotation or translation, direction sampling is carried out using the directions defined by the icosahedron faces. The centres of the faces produce 20 directions. Each triangular face can be subdivided n times, as shown in figure 5, producing 20×4^n directions.

The γ sampling can be carried out easily by uniformly sampling the rotation interval $[-a, a]$. We consider the $[-15 \text{ deg}, 15 \text{ deg}]$ interval and a sampling rate of 2 deg.

As Zhang proposed [2], only half of the sampled translation directions should be tested, since changing the translation sign leads to the solution changing the structure sign, which is geometrically equivalent. As a result, it is sufficient to try half of the translation samples and then test if the computed scene is behind or in front of the cameras. If it is behind, it has no physical meaning, and the real solution is achieved by changing the translation and scene sign.

6 Experimental Results

This section is devoted to showing the performance of the proposed pairing constraint to recover the structure and motion from two real images. We focus

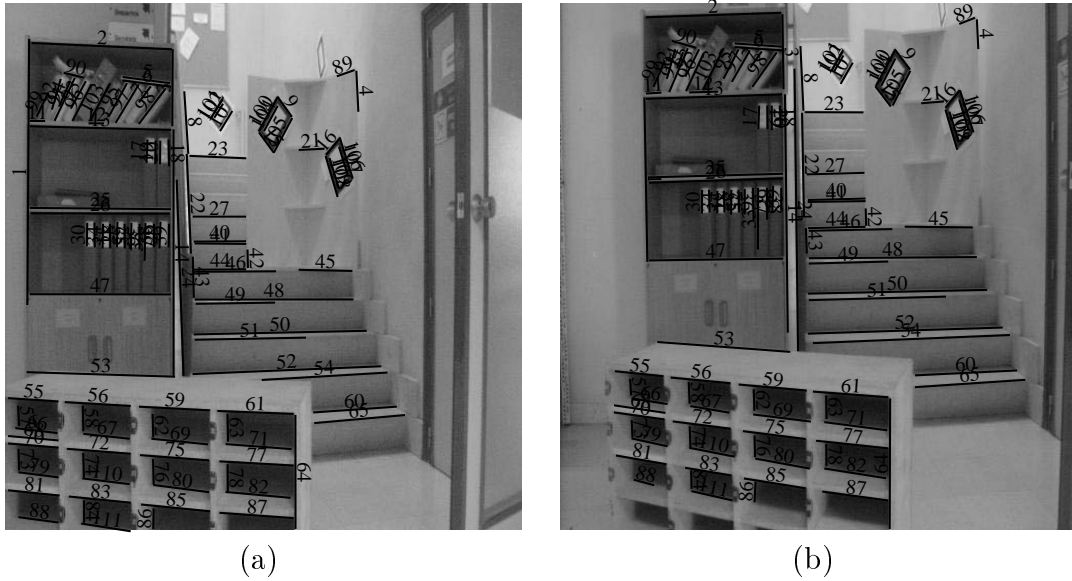


Fig. 6. Input image pair for structure and motion computation. Equal numbers label corresponding image segments.

on the ability to obtain a solution as a function of the κ value (see eq. 5). It is shown that weak pairing between the midpoints, i.e $\kappa > 0.1$ performed well, whereas strong midpoint pairing ($\kappa < 0.01$) performed badly. This confirms that tight pairing between the midpoints is not correct, as stated by several authors [2,3]. However, if deviations of the midpoints matching were allowed along the image segments directions, the correct result could be achieved.

Figure 6 shows the two images ($512 \times 500 \times 8$ B&W) that were used as the input. The corresponding image segments are labelled with the same numbers. These images came from a trinocular rig, and, as a result, it was possible to use the camera calibration as a ground true solution for the computed motion. Camera calibrations were computed using the Tsai method [10]. The camera focal length was 6mm., and the lens radial distortion was compensated for the extracted image segment location. Image segments were detected with the Burns algorithm [11]. Segments shorter than 15 pixels or those with a grey level gradient smaller than 20 grey levels per pixel, were removed. The matches were computed by a stereo program [7]; spurious matches were removed manually. A total of 111 matches were retained after removing three spurious ones.

The tuning parameters for the initial seed algorithm were: range for the rotation angles $[-15 \text{ deg.}, 15 \text{ deg.}]$, and rotation sampling step 2 deg.. Both initial orientation and translation directions were computed from the icosahedron faces with subdivision ($n = 1$). As a result. 48000 initial samples were used.

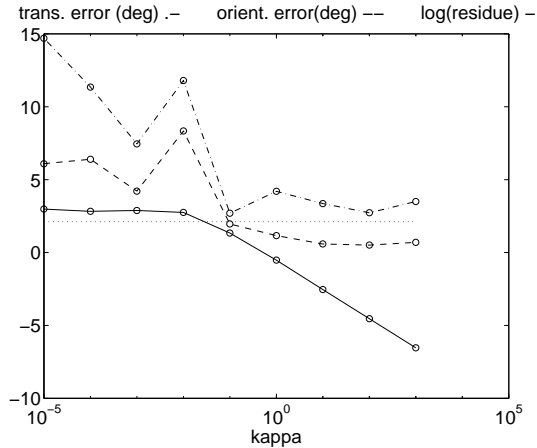


Fig. 7. Optimization residue (log scale), orientation error (deg.) and translation error (deg.) with respect to the κ value. The dot horizontal line represents the 95% χ^2 with 106 d.o.f

κ	estimated $\hat{\mathbf{x}}_{C_1C_2}$	error wrt. ground true	tr. error.	ori. error
ground true	$(0.82, -0.56, 0.15, -3.58, -8.86, 0.71)^T$	$(0, 0, 0, 0, 0, 0)^T$	0 deg.	0 deg.
1	$(0.81, -0.54, 0.22, -4.43, -10.02, 0.74)^T$	$(0.01, 0.02, 0.07, -0.84, -1.16, -0.04)^T$	4.3 deg.	1.2 deg.
10^3	$(0.82, -0.54, 0.21, -4.13, -9.56, 0.73)^T$	$(0.01, 0.02, 0.06, -0.54, -0.69, -0.03)^T$	3.7 deg.	0.7 deg.
10^{-3}	$(0.76, -0.59, 0.26, -0.90, -4.68, 1.33)^T$	$(-0.03, -0.04, 0.12, 2.79, 4.13, 0.87)^T$	7.5 deg.	4.2 deg.

Table 1

Computed camera location compared with the ground true solution. $\mathbf{x}_{C_1C_2}$ is the camera 2 location wrt. the camera 1 (translation normalized to unitary vector, angles in deg.). Error wrt. ground true is the location vector: $\ominus \hat{\mathbf{x}}_{C_1C_2}^{\text{ground}} \oplus \hat{\mathbf{x}}_{C_1C_2}^{\text{estimated}}$

The number of retained seeds was 30. The parameters for the covariances used in the optimization were: $\sigma_{cc} = 2$ pixels and $\sigma_{nc} = 1$ pixel used to tune the image segment location noise (6).

Figure 7 shows the optimization residue, the orientation error and translation error with respect to the κ value. A logarithmic scale is used for κ and the residue. The translation error is the angle (deg.) between the ground true and the computed translation. The orientation error is the angle (deg.), around some axis, required to align the ground true and computed camera frame. Table 1 shows a summary of various values.

Focusing on figure 7 it can be seen that there are two regions: weak midpoint pairing ($\kappa > 0.1$), and strong midpoint pairing ($\kappa < 0.1$). In the weak pairing region, the translation and rotation errors were small and the reconstructions were good. Figures 8 and 9 show the reconstruction and its backprojection in the images for the weak pairing case. Note that in both cases reconstruction

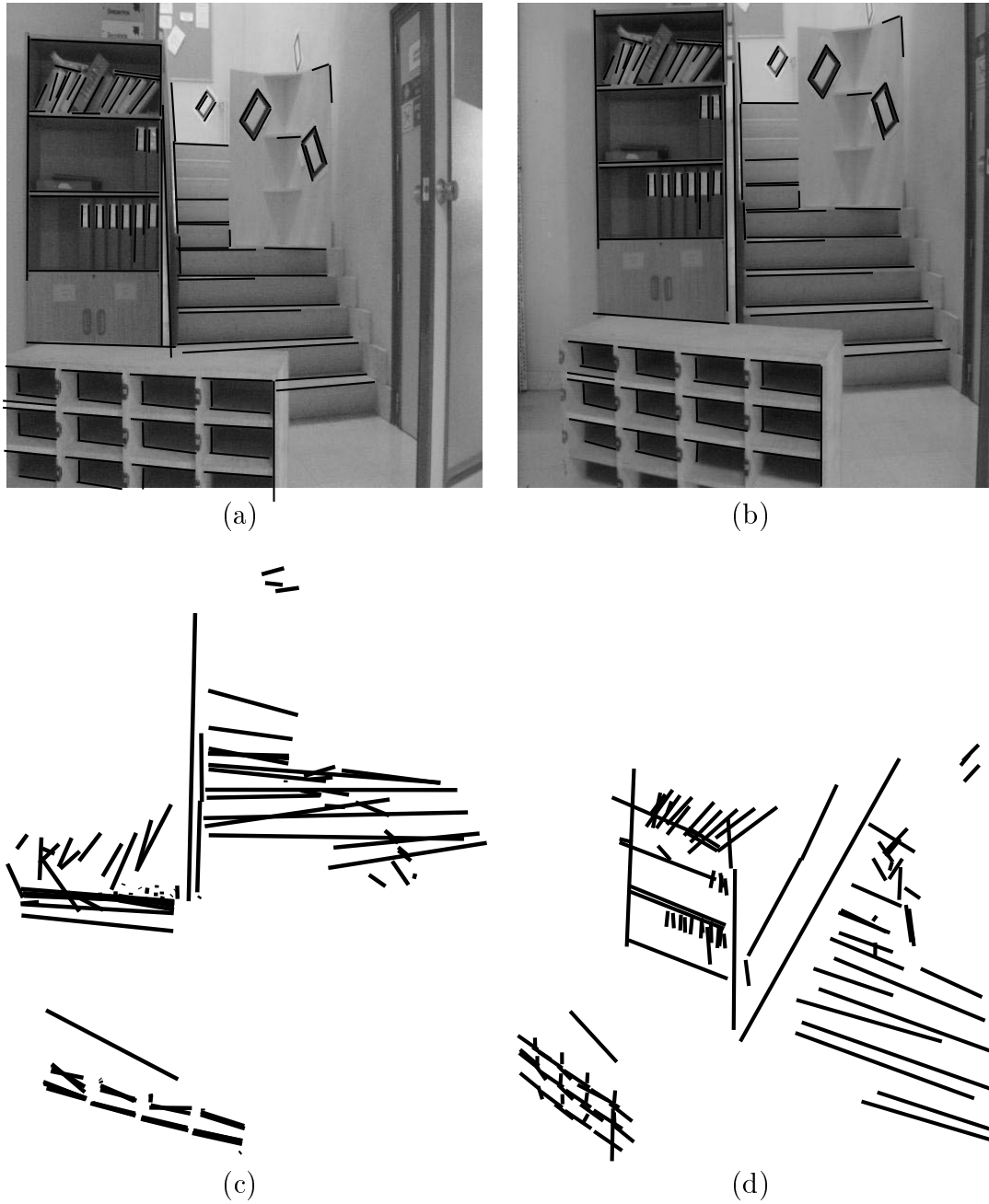


Fig. 8. Reconstruction with $\kappa = 1$. (a) and (b) show reconstruction reprojected on images 1 and 2. (c) and (d) show the reconstruction on top and general view

are very similar despite the difference in the κ value. The similarity for the backprojection is even greater and it can be seen how there nearly equal. In the strong pairing region, the orientation and translation errors were bigger and the reconstruction had no meaning at all.



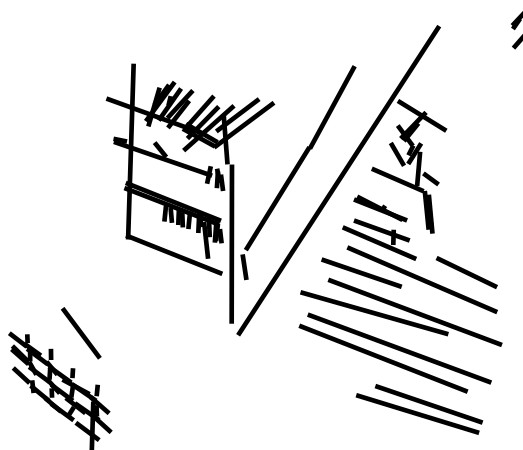
(a)



(b)



(c)



(d)

Fig. 9. Reconstruction with $\kappa = 10^3$. (a) and (b) show reconstruction reprojected on images 1 and 2. (c) and (d) show the reconstruction on top and general view

The residue after optimization was nearly constant in the strong pairing region. However, in the weak pairing region it was reduced as κ increased. With strong midpoint pairing, the results had no meaning because midpoints are not correspondent. The residue is dominated by the component orthogonal to the segment direction. With the weak pairing, however, deviations along the image segment were allowed. By increasing κ , it was found that the solution was approximately the same but the total residue was reduced because the weight of all the deviations was reduced by the same proportion. The reason that the solution could be obtained for any value $k > 0.1$ can be explained as follows. If only the infinite support lines (infinite κ) were used to solve the problem, every camera location would be possible and the residue would always be zero. Considering midpoint matching, we have found that the residue mainly came from the deviations of the midpoints. Once the midpoint matching becomes weak, only deviations along the segment directions are allowed. Consequently, if the κ is increased, all the weighted residues decrease and the total residue decreases, but the solution remains approximately the same.

As stated in section 4, the residue after optimization should follow, for 111 image segments in two cameras, a χ^2 with 106 d.o.f.. Fixing a 95% confidence region, the χ^2 value is 131 (note that figure 7 plots residue on a logarithmic scale so that 131 is represented as 2.11). It is shown in figure 7 that the transition between the strong and weak pairing is related to the intersection of the residue with the χ^2 value.

7 Conclusions

The finite length of the straight segments produces constraints stronger than if they were considered as their infinite supporting line. In this paper, it has been proposed that the image segments midpoints should be considered as correspondent but giving a lower weight in the image segment direction. The experimental results showed that the proposed constraint can be used to recover the structure and the camera motion from straight segment correspondences using only two images. This showed that the constraint is stronger than if only infinite lines were used. This result, together with [2] and [3], confirms the importance of considering the finite length of the segments.

The proposed image segment model and the weak midpoint correspondence has also been used to compute correspondent image segments and the structure when the camera location is known [7]. Consequently, the proposed model uniquely combines both the “point” and “line” properties of the straight segments. This model can be applied to sequences of images.

Acknowledgement

This work has been supported CICYT-TAP97-0992-C02-01. Software for simulation and visualization was developed by Jaime Segura and D. Berna Sanjuán

References

- [1] T.S. Huang and A.R. Netravali. Motion and structure correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, February 1994.
- [2] Z. Zhang. Estimating motion and structure from correspondences of line segments between two perspective images. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 17(12):1129–1139, December 1995.
- [3] C.J. Taylor and D.J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 17(11):1021–1032, November 1995.
- [4] J.D. Tardós. Representing partial and uncertain sensorial information using the theory of symmetries. In *IEEE Int. Conf. on Robotics and Automation*, pages 1799–1804, Nice, France, May 1992.
- [5] X. Pennec and Thirion J.P. Validation of 3-d registration methods based on points and frames. In *V Int. Conf. on Computer Vision*, pages 557–562, MIT, USA, 1995.
- [6] Z. Zhang and O. Faugeras. A 3d world model builder with a mobile robot. *Int. Journal of Robotics Research*, 11(4):269–284, August 1992.
- [7] J. M. M. Montiel and L. Montano. Probabilistic structure from camera location using straight segments. *Image and Vision Computing*, 17(3):263–279, March 1999.
- [8] J.M Martínez Montiel and L. Montano. The effect of the image imperfections of a segment on its orientation uncertainty. In *7th. Int. Conf. on Advanced Robotics*, pages 156–162, Spain, September 1995.
- [9] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences*. Springer-Verlag, Heidelberg, 1993.
- [10] 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*, RA-3(4):323–344, August 1987.
- [11] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(4):425–455, 1986.

Appendixes

A Transformations

The locations for the references are expressed as transformations. There are two mathematical representations for the transformation t_{WG} : a six component location vector \mathbf{x}_{WG} , and an homogeneous matrix \mathbf{H}_{WG} :

$$\mathbf{x}_{WG} = (x_{WG}, y_{WG}, z_{WG}, \psi_{WG}, \theta_{WG}, \phi_{WG})^T$$

$$\mathbf{H}_{WG} = \begin{pmatrix} n_{WG_x} & o_{WG_x} & a_{WG_x} & p_{WG_x} \\ n_{WG_y} & o_{WG_y} & a_{WG_y} & p_{WG_y} \\ n_{WG_z} & o_{WG_z} & a_{WG_z} & p_{WG_z} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Location vector form is well suited for theoretical discussion and for covariance assignment. However, the mathematical operations such as composition, inversion or derivation is better expressed using the homogeneous matrix. The conversion between them is given by:

$$\mathbf{H}_{WG} = \begin{pmatrix} C\phi_{WG}C\theta_{WG} & C\phi_{WG}S\theta_{WG}S\psi_{WG}- & C\phi_{WG}S\theta_{WG}C\psi_{WG}+ & x_{WG} \\ & S\phi_{WG}C\psi_{WG} & S\phi_{WG}S\psi_{WG} & \\ S\phi_{WG}C\theta_{WG} & S\phi_{WG}S\theta_{WG}S\psi_{WG}+ & S\phi_{WG}S\theta_{WG}C\psi_{WG}- & y_{WG} \\ & C\phi_{WG}C\psi_{WG} & C\phi_{WG}S\psi_{WG} & \\ -S\theta_{WG} & C\theta_{WG}S\psi_{WG} & C\theta_{WG}C\psi_{WG} & z_{WG} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A.1})$$

where C and S stands for $\cos()$ and $\sin()$ respectively.

$$\mathbf{x}_{WG} \begin{pmatrix} x_{WG} \\ y_{WG} \\ z_{WG} \\ \psi_{WG} \\ \theta_{WG} \\ \phi_{WG} \end{pmatrix} = \begin{pmatrix} p_{WG_x} \\ p_{WG_y} \\ p_{WG_z} \\ \text{atan2}(o_{WG_z}, a_{WG_z}) \\ \text{atan2}(-n_{WG_z}, +\sqrt{n_{WG_x}^2 + n_{WG_y}^2}) \\ \text{atan2}(n_{WG_y}, n_{WG_x}) \end{pmatrix} \quad (\text{A.2})$$

B Image Normalization Jacobian

$$N = \begin{pmatrix} \frac{C\hat{\phi}_{CP}C\hat{\phi}_{MP}}{\alpha_u} + \frac{S\hat{\phi}_{CP}S\hat{\phi}_{MP}}{\alpha_v} & -\frac{C\hat{\phi}_{CP}S\hat{\phi}_{MP}}{\alpha_u} + \frac{S\hat{\phi}_{CP}C\hat{\phi}_{MP}}{\alpha_v} & 0 \\ -\frac{S\hat{\phi}_{CP}C\hat{\phi}_{MP}}{\alpha_u} + \frac{C\hat{\phi}_{CP}S\hat{\phi}_{MP}}{\alpha_v} & \frac{S\hat{\phi}_{CP}S\hat{\phi}_{MP}}{\alpha_u} + \frac{C\hat{\phi}_{CP}C\hat{\phi}_{MP}}{\alpha_v} & 0 \\ 0 & 0 & \frac{\frac{\alpha_u}{\alpha_v}}{\left(\frac{\alpha_u}{\alpha_v}\right)^2 + \cos^2 \hat{\phi}_{MP} \left(1 - \left(\frac{\alpha_u}{\alpha_v}\right)^2\right)} \end{pmatrix}$$

where $\hat{\phi}_{CP}$ were defined in (3). $\frac{1}{\alpha_u}$ and $\frac{1}{\alpha_v}$ are the pixel sizes in the x and y directions, expressed in mm. S and C stands for the $\sin()$ and $\cos()$ functions. $\hat{\phi}_{MP}$ is defined as:

$$\hat{\phi}_{MP} = \text{atan2} \left(\alpha_v \sin \hat{\phi}_{CP}, \alpha_u \cos \hat{\phi}_{CP} \right)$$

C 2D Segment Definition

The 2D segment location with respect to the camera frame is expressed as:

$$\hat{\mathbf{x}}_{CD} = \left(0, 0, 0, \hat{\psi}_{CD}, \hat{\theta}_{CD}, \hat{\phi}_{CD} \right)^T$$

where:

$$\hat{\psi}_{CD} = \text{atan2} (o_z, a_z), \quad \hat{\theta}_{CD} = \text{atan2} \left(n'_z, \sqrt{n'^2_x + n'^2_y} \right), \quad \hat{\phi}_{CD} = \text{atan2} (n'_y, n'_x)$$

where:

$$n'_x = \cos \hat{\phi}_{CP} + \hat{y}_{CP}^2 \cos \hat{\phi}_{CP} - \hat{x}_{CP} \hat{y}_{CP} \sin \hat{\phi}_{CP} \tag{C.1}$$

$$n'_y = \sin \hat{\phi}_{CP} + \hat{x}_{CP}^2 \sin \hat{\phi}_{CP} - \hat{x}_{CP} \hat{y}_{CP} \cos \hat{\phi}_{CP}$$

$$n'_z = \hat{y}_{CP} \sin \hat{\phi}_{CP} + \hat{x}_{CP} \cos \hat{\phi}_{CP}$$

$$o_z = \frac{-1}{\sqrt{1 + \hat{x}_{CP}^2 + \hat{y}_{CP}^2}}$$

$$a_z = \frac{\hat{x}_{CP} \sin \hat{\phi}_{CP} - \hat{y}_{CP} \cos \hat{\phi}_{CP}}{\sqrt{1 + \left(\hat{x}_{CP} \sin \hat{\phi}_{CP} - \hat{y}_{CP} \cos \hat{\phi}_{CP} \right)^2}} \tag{C.2}$$

and the corresponding covariance is defined as a function of K_{DP} :

$$K_{DP} = \begin{pmatrix} 0 & -\frac{\sin \hat{\psi}_{DP}}{\hat{y}_{DP}} & 0 \\ 0 & -\frac{\sin \hat{\phi}_{DP} \sin \hat{\psi}_{DP}}{\hat{y}_{DP} \cos \hat{\phi}_{DP}} & \frac{\sin \hat{\psi}_{DP}}{\cos \hat{\phi}_{DP}} \\ \frac{\cos \hat{\phi}_{DP}}{\hat{y}_{DP}} & -\frac{\sin \hat{\phi}_{DP} \cos \hat{\psi}_{DP}}{\hat{y}_{DP}} & 0 \end{pmatrix} \quad (C.3)$$

where:

$$\hat{y}_{DP} = -\sqrt{\hat{x}_{CP}^2 + \hat{y}_{CP}^2 + 1}, \quad \hat{\psi}_{DP} = \text{atan2}(1, a'_z), \quad \hat{\phi}_{DP} = \text{atan2}(o_x, n_x)$$

and where:

$$\begin{aligned} a'_z &= (\hat{x}_{CP} \sin \hat{\phi}_{CP} - \hat{y}_{CP} \cos \hat{\phi}_{CP}) \\ n_x &= \frac{1}{\|\mathbf{n}'_D\|} \left(1 + (\hat{x}_{CP} \sin \hat{\phi}_{CP} - \hat{y}_{CP} \cos \hat{\phi}_{CP})^2 \right) \\ o_y &= \frac{-1}{\|\mathbf{o}'_D\|} (\hat{y}_{CP} \sin \hat{\phi}_{CP} + \hat{x}_{CP} \cos \hat{\phi}_{CP}) \\ \|\mathbf{o}'_D\| &= \sqrt{1 + \hat{x}_{CP}^2 + \hat{y}_{CP}^2} \\ \|\mathbf{n}'_D\| &= \sqrt{1 + \hat{x}_{CP}^2 + \hat{y}_{CP}^2 + \hat{x}_{CP}^2 \hat{y}_{CP}^2 + (\hat{y}_{CP}^4 + \hat{y}_{CP}^2) \cos^2 \hat{\phi}_{CP} +} \\ &\quad + (\hat{x}_{CP}^4 + \hat{x}_{CP}^2) \sin^2 \hat{\phi}_{CP} - (\hat{x}_{CP}^3 \hat{y}_{CP} + \hat{y}_{CP}^3 \hat{x}_{CP} + \hat{x}_{CP} \hat{y}_{CP}) 2 \sin \hat{\phi}_{CP} \cos \hat{\phi}_{CP} \end{aligned}$$

The values of \hat{x}_{CP} , \hat{y}_{CP} , and $\hat{\phi}_{CP}$ are taken from the image segment location with respect to the camera frame (3).

D Measurement Equation

The detailed expression for the matrices and vectors used in the linearizations are:

$$\mathbf{f} = \begin{pmatrix} \hat{x}_{DS} \\ \hat{z}_{DS} \\ \text{atan2}(-n_{DSz}, \sqrt{n_{DSx}^2 + n_{DSy}^2}) \end{pmatrix} \quad (D.1)$$

$$G = \begin{pmatrix} 0 & -\hat{z}_{DS} & \hat{y}_{DS} & n_{DSx} & o_{DSx} & a_{DSx} & 0 & 0 \\ -\hat{y}_{DS} & \hat{x}_{DS} & 0 & n_{DSz} & o_{DSz} & a_{DSz} & 0 & 0 \\ \sin \hat{\phi}_{DS} & -\cos \hat{\phi}_{DS} & 0 & 0 & 0 & 0 & \cos \hat{\psi}_{DS} & -\sin \hat{\psi}_{DS} \end{pmatrix}$$

$$H = \begin{pmatrix} -n_{CD_x} & -n_{CD_y} & -n_{CD_z} & n_{CD_y} \hat{z}_{CS} - & -n_{CD_x} \hat{z}_{CS} + & n_{CD_x} \hat{y}_{CS} - \\ & & & +n_{CD_z} \hat{y}_{CS} & +n_{CD_z} \hat{x}_{CS} & +n_{CD_y} \hat{x}_{CS} \\ -a_{CD_x} & -a_{CD_y} & -a_{CD_z} & a_{CD_y} \hat{z}_{CS} - & -a_{CD_x} \hat{z}_{CS} + & a_{CD_x} \hat{y}_{CS} - \\ & & & +a_{CD_z} \hat{y}_{CS} & +a_{CD_z} \hat{x}_{CS} & +a_{CD_y} \hat{x}_{CS} \\ 0 & 0 & 0 & -o_{CS_x} \cos \hat{\psi}_{DS} + & -o_{CS_y} \cos \hat{\psi}_{DS} + & -o_{CS_z} \cos \hat{\psi}_{DS} + \\ & & & +a_{CS_x} \sin \hat{\psi}_{DS} & +a_{CS_y} \sin \hat{\psi}_{DS} & +a_{CS_z} \sin \hat{\psi}_{DS} \end{pmatrix} \quad (D.2)$$

Previous expressions are given as functions of the homogeneous matrices \mathbf{H}_{CD} , \mathbf{H}_{DS} and \mathbf{H}_{CS} . These matrices can be computed directly from the location estimated for the 2D segment, \mathbf{H}_{CD} , the camera \mathbf{H}_{WC} , and the 3D segment location \mathbf{H}_{WS} .

About the Author – Dr. J.M. MARTÍNEZ MONTIEL received the PhD in Systems Engineering and Computer Science from the University of Zaragoza 1996. At present he is assistant lecturer at the University of Zaragoza. His current interests are computer vision based on discrete features, multisensor fusion, map building, stereo vision and robust methods.

About the Author – Dr. JUAN D. TARDÓS received the M.S. and Ph.D. degrees in Industrial-Electrical Engineering from the University of Zaragoza, Spain, in 1985 and 1991, respectively. He is Associate Professor in the Departamento de Informática e Ingeniería de Sistemas, University of Zaragoza, where he is in charge of courses in Real Time Systems and Computer Vision. His current research interests include sensor integration, robotics, and real time systems.

About the Author – Dr. LUIS MONTANO received the PhD in Systems Engineering and Computer Science from the University of Zaragoza 1987. At present he is the Head of the Department of Computer Science and Systems Engineering of the University of Zaragoza. His current interests are computer vision based on discrete features, robot programming and control, and mobile robot navigation.