Uncertainty propagation: Avoiding the expensive sampling process for real-time image-based measurements

Leandro A.F. Fernandes*, Manuel M. Oliveira¹, Roberto da Silva¹

Instituto de Informática - PPGC, Universidade Federal do Rio Grande do Sul, CP 15064, CEP 91501-970, Porto Alegre, RS, Brazil

Received 20 July 2007; received in revised form 7 December 2007; accepted 8 December 2007
Available online 28 December 2007

Abstract

A common way of evaluating the quality of a measuring device is to use it to measure the properties of some test objects, thus obtaining a large number of samples whose values are then compared to a known ground truth. Such a process tends to be labor intensive and time-consuming. A more convenient and elegant alternative is to statistically propagate the uncertainty of the measurement process throughout the computation chain. A clear advantage of such an approach over the conventional sampling-based method is its practical nature: it allows the continuous changes in input parameter values and sampling conditions, which are common in real-time applications, to be instantly taken into account.

In order to demonstrate the benefits of using uncertainty propagation in real-time image metrology applications, we describe a method for automatic computation of box dimensions from single perspective projection images in real time. For this, we derive expressions for the uncertainty in the measurements based on the uncertainties present in all variables used in the computational flow. Our results show that these estimates are in accordance with the ones obtained using the conventional sampling process, thus safely replacing them. We also show that the uncertainty propagation approach is computationally efficient. This approach can be incorporated into applications that aim to make real-time measurements directly from images, and should also be useful in many other time-critical applications.

© 2007 Elsevier B.V. All rights reserved.

1. Introduction

Many applications, including analysis of forensic records, storage management, cost estimation, and surveillance, heavily rely on making measurements of three-dimensional objects. Even for the case of objects with very simple geometry, such as boxes, computing their dimensions can be an integral part of the day-by-day operations of many companies, such as couriers, airlines and warehouses, which use this information both for planning as well as for cost...
estimation. For these applications and companies, the ability to compute the dimensions of three-dimensional objects directly from images can greatly impact their performances. Unfortunately, unless some information relating distances measured in image space to distances measured in 3D is available, the problem of making measurements directly from images is not well defined. This results from the inherent ambiguity of perspective projection images caused by the loss of depth information (Hartley and Zisserman, 2000). Several authors eliminate the projective ambiguity and perform measurements or shape reconstruction from images by using active triangulation (Sanz, 1989; Proesmans and Van Gool, 1997; Bouguet, 1999), multiple views of the scene (Hartley and Zisserman, 2000; Longuet-Higgins, 1981; Liebowitz, 2001) or a single image (Criminisi et al., 1998; Criminisi, 1999; Criminisi et al., 1999, 2000; Lu, 2000) assuming previous knowledge of some 3D geometric information of the scene, like distances and angles. However, only a few of them are concerned with automatic methods (Lu, 2000).

In a recent work (Fernandes et al., 2006), we presented a completely automatic method for real-time computation of box dimensions from single perspective projection images. The approach uses information extracted from the silhouette of the target box and can be applied when at least two of its faces are visible, even when the box is partially hidden by other objects in the scene (Fig. 1). We eliminate the projective ambiguity by projecting two parallel laser beams onto one of the visible faces of the target box. We have demonstrated the effectiveness of this technique by building a scanner prototype (Fig. 1, left) for computing box dimensions and by using it to compute the dimensions of boxes in real time. In Fernandes et al. (2006), we analyzed the accuracy of the measurements obtained with the proposed approach using some relatively simple descriptive statistics.

This paper derives a statistical method for estimating the uncertainties in the measurements obtained with the technique described in Fernandes et al. (2006) and compares its results against the ones obtained with a sampling-based strategy. The approach is based on error propagation techniques (Parratt, 1961) and has some significant advantages over the conventional sampling-based method used for uncertainty estimation: (i) it allows the continuous changes in input parameter values and sampling conditions to be instantly reflected in the estimated uncertainty; (ii) it avoids the need of a labor-intensive sampling-based technique, since the computation can be performed from a single image (sample); and (iii) as we will demonstrate, it can be easily integrated with real-time measurement applications for instantly providing error estimates. We show that our results are equivalent to and can replace the ones obtained with the traditional sampling-based method with advantages. Moreover, the results of our experiments suggest that uncertainty propagation can be used by any application that aims to make real-time measurements directly from images, and should also be useful in many other time-critical scenarios.

The remainder of the paper is organized as follows: Section 2 reviews the method used to compute box dimensions (Fernandes et al., 2006). Its importance is twofold: (i) to describe the method used for computing box dimensions and all sources of uncertainties involved in the process, and (ii) to present the equations that will propagate these uncertainties throughout the computation chain. Section 3 shows how the inherent uncertainties are actually propagated into the final measurements. Section 4 discusses some of our results. Finally, Section 5 summarizes the paper.
2. Computing box dimensions

Our scanner prototype is comprised of a firewire color camera, two laser pointers and a software module (Fig. 1, left). The camera was mounted on a plastic support and the laser pointers were aligned with and glued to the sides of the support. In a typical measurement setup, the user points the laser beams to a target box and the system computes its dimensions and corresponding uncertainties from the images grabbed by the camera, in real time (Fig. 1, right). For each acquired image, the processing steps include:

1. identify the target box silhouette and the laser dots in the image;
2. recover the supporting lines for box silhouette edges;
3. compute the 2D coordinates of the imaged silhouette vertices;
4. eliminate the projective ambiguity;
5. reproject the box vertices back to 3D space; and
6. compute the dimensions of the target box using the estimated 3D coordinates of its vertices.

Fig. 2 illustrates the steps involved in the entire process. The identification of the target box silhouette and the positions of the laser dots (block 1), and the recovery of the supporting lines for the silhouette edges (block 2) are performed using standard image processing techniques. An in-depth description of these algorithms can be found in Fernandes et al. (2006). The remaining steps correspond to the computational stages that will propagate the uncertainties in the values of the extracted parameters (i.e., supporting silhouette lines and positions of the laser dots) toward the final measurements. They are described next for the case where the imaged boxes have three visible faces. The case involving only two visible faces is similar and is described in Section 3.2.

In the following derivations, we model boxes as parallelepipeds even though boxes can present many imperfections (e.g., bent edges and corners, asymmetries, etc.). Also, we assume that the images used for computing the dimensions were obtained through linear projection (i.e., using a pinhole camera, Fig. 4). We use simple warping
Fig. 3. Vanishing points ($\omega_i$) and vanishing lines ($\lambda_i$). $e_j$, $v_j$, $f_i$ and $m_0$ are the supporting lines for silhouette edges, the silhouette vertices, the faces of the box, and the inner vertex, respectively.

procedures (Hartley and Zisserman, 2000) to compensate, in real time, for the radial and tangential distortions introduced by the camera’s lens. Note that the origin of the image coordinate system is at the center of the image, with the $X$-axis growing to the right and the $Y$-axis growing down.

2.1. Computing the 2D coordinates of the imaged silhouette vertices

Once the supporting lines for silhouette edges $e_i$ have been identified in the image (solid lines in Fig. 1, right), the coordinates of the silhouette vertices $v_i$ (in homogeneous coordinates) are computed as the intersections of these silhouette lines as

$$\hat{v}_i = \begin{pmatrix} x_{\hat{v}_i} \\ y_{\hat{v}_i} \\ w_{\hat{v}_i} \end{pmatrix} = e_i \times e_{(i+1)\text{mod } 6}$$

where $0 \leq i \leq 5$, $e_i = (a_{e_i}, b_{e_i}, c_{e_i})^T$ are the coefficients of the general equation for the $i$-th silhouette line (see Fig. 3, left) and $\times$ is the cross product operator. Since $\hat{v}_i$ is represented in homogeneous coordinates, one must perform the division by $w_{\hat{v}_i}$ in order to get the actual coordinates of the vertex in the image:

$$v_i = \begin{pmatrix} x_{v_i} \\ y_{v_i} \\ 1 \end{pmatrix} = \frac{1}{w_{\hat{v}_i}} \begin{pmatrix} x_{\hat{v}_i} \\ y_{\hat{v}_i} \\ w_{\hat{v}_i} \end{pmatrix}.$$  \hspace{1cm} (2)

Given the intrinsic parameters of the camera (computed for radial and tangential distortion compensation (Hartley and Zisserman, 2000; Bouguet, 2005)) one can reproject vertices $v_i$ onto the image plane of a virtual camera in 3D, where such a plane is expressed in the virtual camera’s coordinate system as $Z = 1$ (i.e., making $f = 1$ in Fig. 4, right):

$$v'_i = \begin{pmatrix} x_{v'_i} \\ y_{v'_i} \\ 1 \end{pmatrix} = RK^{-1}v_i$$

where

$$K = \begin{pmatrix} \alpha_x & \gamma & \alpha_x \\ 0 & \alpha_y & \alpha_y \\ 0 & 0 & 1 \end{pmatrix}, \quad R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$  \hspace{1cm} (4)
Fig. 4. Pinhole camera model: (left) 3D view and (right) side view. Here, $O$ is the center of projection and $\Pi_I$ is the image plane. $P$ is a point in 3D projected to point $p'$ on the image plane. $o'$ is the principal point and the origin of the image reference system. $f$ is the focal length, defined by the distance between the image plane and the center of projection.

In Eq. (3), $K$ is the matrix that models the intrinsic camera parameters (i.e., the parameters that describe the specific camera independent on its position and orientation in space) and $R$ is a reflection matrix used to make the $Y$-axis of the image coordinate system grow in the upward direction. In Eq. (4), $\alpha_x = f/s_x$ and $\alpha_y = f/s_y$, where $f$ is the camera’s focal length, and $s_x$ and $s_y$ are the dimensions of the pixel (CCD) in centimeters. $\gamma$, $o_x$ and $o_y$ represent the skew and the coordinates of the principal point, respectively (Hartley and Zisserman, 2000).

2.2. Eliminating the projective ambiguity

We remove the projective ambiguity by projecting two parallel laser beams onto one of the visible faces of the target box and relating the distance of the laser dots measured on the image plane (where $Z = 1$) to the distance, in 3D, between the projections of the two laser dots on the box face.

The vector normal to a plane $\Pi_j$ in a given camera’s coordinate system can be obtained by multiplying the transpose of the camera’s intrinsic-parameter matrix (Eq. (4)) by the coefficients of the plane’s vanishing line (Hartley and Zisserman, 2000). Since the resulting vector is not necessarily a unit vector, it needs to be normalized:

$$N_{\Pi_j} = \begin{pmatrix} A_{\Pi_j} \\ B_{\Pi_j} \\ C_{\Pi_j} \end{pmatrix} = \frac{R K^T \lambda_j}{\| R K^T \lambda_j \|}. \tag{5}$$

In Eq. (5), $0 \leq j \leq 2$, $\lambda_j$ is the vanishing line defined by all vanishing points $\omega$ from all sets of parallel lines on the plane $\Pi_j$ and $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$ is the $L_2$ norm. $\omega_j$ and $\lambda_j$ (Fig. 3, right) can be computed from the supporting lines of the silhouette edges. Eqs. (6), (7) and Fig. 3 show the relationship among the vanishing points $\omega_j$, the vanishing lines $\lambda_j$ and the supporting lines $e_j$ for the edges that coincide with the imaged silhouette of a parallelepiped with three visible faces. The supporting lines are ordered clockwise.

$$\omega_j = \begin{pmatrix} x_{\omega_j} \\ y_{\omega_j} \\ w_{\omega_j} \end{pmatrix} = e_j \times e_{j+3} \tag{6}$$

$$\lambda_j = \begin{pmatrix} a_{\lambda_j} \\ b_{\lambda_j} \\ c_{\lambda_j} \end{pmatrix} = \omega_j \times \omega_{(j+1)\text{mod } 3}. \tag{7}$$

Given $N_{\Pi_j}$, finding $D_{\Pi_j}$ (the fourth coefficient of the plane equation) is equivalent to solving the projective ambiguity and will require the introduction of one more constraint. Thus, consider the situation depicted in Fig. 5, where two laser beams, parallel to each other, are projected onto one of the faces of the box. Let the 3D coordinates of the laser dots (on the box face) defined with respect to the camera’s coordinate system be $P_0 = (X_{P_0}, Y_{P_0}, Z_{P_0})^T$ and $P_1 = (X_{P_1}, Y_{P_1}, Z_{P_1})^T$, respectively (Fig. 5). Since $P_0$ and $P_1$ are on the same plane $\Pi$, one can write

$$A_{\Pi} X_{P_0} + B_{\Pi} Y_{P_0} + C_{\Pi} Z_{P_0} = A_{\Pi} X_{P_1} + B_{\Pi} Y_{P_1} + C_{\Pi} Z_{P_1} \tag{8}$$
Fig. 5. Top view of a scene. Two laser beams apart in 3D by $d_{lb}$ project onto one box face at points $P_0$ and $P_1$, whose distance in 3D is $d_{ld}$. $\alpha$ is the angle between $-L$ and $N_L$.

Using the linear projection model and given $p_l = (x_{pl}, y_{pl}, 1)^T$ (where $0 \leq l \leq 1$), the homogeneous coordinates of the pixel associated with the projection of point $P_l$, one can reproject $p_l$ on the plane $Z = 1$ using Eq. (3) and express the 3D coordinates of the laser dots on the face of the box as

$$p_l = \begin{pmatrix} X_{P_l} \\ Y_{P_l} \\ Z_{P_l} \end{pmatrix} = \begin{pmatrix} x_{p_l}^{'} Z_{P_l} \\ y_{p_l}^{'} Z_{P_l} \\ Z_{P_l} \end{pmatrix}.$$  \hspace{1cm} (9)

Substituting the expression for $X_{P0}, Y_{P0}, X_{P1}$ and $Y_{P1}$ (Eq. (9)) in Eq. (8) and solving for $Z_{P0}$, we obtain

$$Z_{P0} = k Z_{P1}.$$  \hspace{1cm} (10)

where

$$k = \frac{A\Pi x_{p1} + B\Pi y_{p1} + C\Pi}{A\Pi x_{p0} + B\Pi y_{p0} + C\Pi}.$$  \hspace{1cm} (11)

Now, let $d_{lb}$ and $d_{ld}$ be the distances, in 3D, between the two parallel laser beams and between the two laser dots projected onto one of the faces of the box, respectively (Fig. 5). $d_{ld}$ can be directly computed from $N_{\Pi}$, the normal vector of the face onto which the dots project, and the known distance $d_{lb}$:

$$d_{ld} = \frac{d_{lb}}{\cos(\alpha)} = \frac{d_{lb}}{-(N_L \cdot L)}.$$  \hspace{1cm} (12)

where $\alpha$ is the angle between $N_L$, the normalized projection of $N_{\Pi}$ onto the plane defined by the two laser beams, and $L$, the vector representing the laser beam direction. For now on, we will assume that the laser plane is parallel to the camera’s $XZ$ plane and $L = (0, 0, 1)^T$. Therefore, $N_L$ is obtained by dropping the $Y$ coordinate of $N_{\Pi}$ and normalizing the resulting vector. $d_{ld}$ can also be expressed as the Euclidean distance between the two laser dots in 3D:

$$d_{ld}^2 = (X_{P1} - X_{P0})^2 + (Y_{P1} - Y_{P0})^2 + (Z_{P1} - Z_{P0})^2.$$  \hspace{1cm} (13)

Substituting Eqs. (9), (10) and (12) into Eq. (13) and solving for $Z_{P1}$, one gets

$$Z_{P1} = \sqrt{\frac{d_{ld}^2}{ak^2 - 2bk + c}}.$$  \hspace{1cm} (14)
Fig. 6. Computing the projection of the normal vector \( N_{II} \) onto the laser plane \( II_{L} \). Here, \( N_{II} \) is the normal vector of the face that receives the laser dots. \( II_{L} \) is the plane defined by the laser beams. \( P_0, P_1 \) and \( P_2 \) are points on \( II_{L} \). The points \( P_3 \) and \( P_4 \) define a line perpendicular to \( II_{L} \) and that the laser dots. The intersection of such a line with the laser plane is the point \( Q \). \( Q \) and \( P_1 \) define the projection of \( N_{II} \) onto \( II_{L} \).

where \( a = (x_{p_0}^t)^2 + (y_{p_0}^t)^2 + 1, b = x_{p_0}^t x_{p_1}^t + y_{p_0}^t y_{p_1}^t + 1 \) and \( c = (x_{p_1}^t)^2 + (y_{p_1}^t)^2 + 1 \). Given \( Z_{P_1} \), the 3D coordinates of \( P_1 \) can be computed as

\[
P_1 = \left( \begin{array}{c} X_{P_1} \\ Y_{P_1} \\ Z_{P_1} \end{array} \right) = \left( \begin{array}{c} x_{p_1} Z_{P_1} \\ y_{p_1} Z_{P_1} \\ Z_{P_1} \end{array} \right).
\]

The projective ambiguity can be finally removed by computing the \( D_{II} \) coefficient for the plane equation of the face containing the two dots:

\[
D_{II} = - (A_{II} X_{P_1} + B_{II} Y_{P_1} + C_{II} Z_{P_1}).
\]

2.2.1. Estimating the laser plane

In practice, it is difficult to guarantee that the plane defined by the laser beams is parallel to the camera’s \( XZ \) plane, and that the \( L \) vector is aligned with the camera \( Z \)-axis. In our scanner prototype, we noticed that although the laser beams are parallel to each other, the plane they define \( (II_L) \) is not parallel to the camera’s \( XZ \) plane. Therefore, it is necessary to take into account the angle between these two planes before computing \( N_{L} \) and then \( d_{ld} \) (Eq. (12)).

Fig. 6 shows that the vector \( N_{L} \) can be computed from the laser dot \( P_1 \) and from the point \( Q \), the perpendicular projection of \( P_3 \) onto plane \( II_{L} \). \( P_3 \) is given by displacing \( P_1 \) along the \( N_{II} \) direction, the normal vector of the face onto which the dots project. Therefore, the problem can be solved as the intersection of plane \( II_{L} \) and the line perpendicular to \( II_{L} \) passing through \( P_3 \). The point of intersection between the plane defined by points \( P_0, P_1 \) and \( P_2 \) and the line defined by points \( P_3 \) and \( P_4 \) is given by Weisstein (2005):

\[
Q = P_3 + (P_3 - P_4) t
\]

where

\[
t = \begin{bmatrix} X_{P_0} & Y_{P_0} & Z_{P_0} & 1 \\ X_{P_1} & Y_{P_1} & Z_{P_1} & 1 \\ X_{P_2} & Y_{P_2} & Z_{P_2} & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = P_0 \begin{bmatrix} X_{P_0} \\ X_{P_1} \\ X_{P_2} \end{bmatrix} + P_2 \begin{bmatrix} Y_{P_0} \\ Y_{P_1} \\ Y_{P_2} \end{bmatrix} + P_4 \begin{bmatrix} Z_{P_0} \\ Z_{P_1} \\ Z_{P_2} \end{bmatrix} = \begin{bmatrix} X_{P_4} \\ Y_{P_4} \\ Z_{P_4} \end{bmatrix}.
\]

\( P_0 \) and \( P_1 \) are the 3D coordinates of the laser points, \( P_2 = P_1 - L, P_3 = P_1 + N_{II} \) and \( P_4 \) is

\[
P_4 = P_3 + \frac{(P_2 - P_0) \times (P_1 - P_0)}{\| (P_2 - P_0) \times (P_1 - P_0) \|}.
\]
Once computed \( Q \) (Eq. (17)), the normalized projection of \( N_{\Pi} \) onto the plane defined by the two laser beams is computed as

\[
N_L = \frac{Q - P_1}{\|Q - P_1\|}.
\] (18)

A first look at Eq. (17) and Eq. (18) suggests that the \( Z \) coordinates of points \( P_0, P_1, P_2, P_3 \) and \( P_4 \) are needed to compute \( Q \) and \( N_L \), respectively. However, this is not necessary, because \( N_L \) depends only on the orientation of the plane \( \Pi_L \). Given Eq. (9) and the relationship between \( Z_{P_0} \) and \( Z_{P_1} \) (Eq. (10)), Eq. (18) can be expressed as

\[
N_L = \frac{N_{\Pi} + mW}{\|N_{\Pi} + mW\|}
\] (19)

where

\[
m = \frac{\langle -W, N_{\Pi} \rangle}{\langle W, W \rangle}
\]

and

\[
W = (L \times p_1') + k (p_0' \times L)
\]

Here, \( \|\cdot\| \) is the \( L2 \) norm, \( \langle \cdot, \cdot \rangle \) is the dot product operator, \( \times \) is the cross product operator and \( k \) is given by Eq. (11).

The \( L \) vector is estimated using the Camera Calibration Toolbox for Matlab (Bouguet, 2005). By projecting the laser beams on a planar checkerboard calibration pattern, placed at varying distances from the scanner, we collect the coordinates of a set of 3D points (corresponding to these projections) along the laser lines and estimate \( L \)’s direction with respect to the camera’s coordinate system.

2.3. Reprojecting the box vertices back to 3D and computing the box dimensions

Given the coefficients of the equation of one of the planes (see Eqs. (5) and (16)), these can be used to recover the 3D coordinates of the vertices lying on this plane. For each vertex \( v \) in the image (Eq. (2)), its reprojection \( v' \) on the plane \( Z = 1 \) is obtained using the Eq. (3). So, the corresponding coordinate \( Z_V \) is calculated substituting Eq. (15) in the equation of the plane:

\[
Z_V = -\frac{D_{\Pi}}{A_{\Pi}x_v' + B_{\Pi}y_v' + C_{\Pi}}.
\] (20)

From that, given \( Z_V \), both coordinates \( X_V \) and \( Y_V \) are computed using Eq. (15). Since each face of a box shares two vertices with each of its adjacent faces, one can compute the \( D \) coefficients for these adjacent faces and, thus, allowing the recovery of the 3D coordinates of all visible silhouette vertices. From these, the dimensions of the box can be computed (given the assumption that no edge in the box silhouette is completely hidden).

3. Uncertainty propagation

Any given computation propagates the uncertainties associated with its input variables to its output. The uncertainty associated with a variable \( w \) whose value is computed from a set of experimental data can be estimated using an error propagation model (Parratt, 1961):

\[
A_w = \nabla f A_\theta \nabla f^T
\] (21)

where \( A_w \) is the covariance matrix that models the uncertainty in \( w \), \( \nabla f \) is the Jacobian matrix for the function \( f (\theta) \) that computes each term of \( w \) from the \( n \) input variables and \( A_\theta \) is the covariance matrix that models the uncertainty in the input variables. Using such a model, we can obtain confidence intervals for the computed lengths of each visible edge in a target box, from a single input image. To apply this error propagation model, one needs to estimate the uncertainty associated with each input variable (Section 3.1) and to compute the Jacobian matrix for the equation that calculates the edge lengths of the target box. However, expressing the entire computation chain as a single equation in terms of the input variables, and from it computing the Jacobian matrix, turns out to be impractical for two important reasons:
1. The resulting equation would be extremely big. According to our experience, symbolic manipulation programs, such as Mathematica (Wolfram Research Inc., 2005) and Matlab (MathWorks Inc., 2006), were unable to handle even the intermediate stages of the rewriting process of the final expression in terms of the input variables.

2. The intermediate variables can be combined in different ways, depending on which face receives the laser markers or the number of visible faces, for instance. As a result, the implementation must handle all possible computation flows.

Our solution to those problems is to solve the partial derivatives in matrix $\nabla f$ using the chain rule, step by step, until the final result is found. The details of this derivation are presented in Section 3.2.

3.1. Uncertainty in the input variables

Recalling Section 2, the input variables of our measurement process are:

- $e_i = (a_{e_i}, b_{e_i}, c_{e_i})^T$, $0 \leq i \leq 5$: the coefficients of the general line equations of the supporting lines for the silhouette edges;
- $p_j = (x_{p_j}, y_{p_j})^T$, $0 \leq j \leq 1$: the image coordinates of the two laser dots;
- $d_{lb}$: the distance between the two laser beams;
- $K$: the camera’s intrinsic-parameter matrix;
- $L = (X_L, Y_L, Z_L)^T$: the laser beam direction.

We must estimate the uncertainty in such variables in order to compute their impact in the resulting measurements.

3.1.1. Supporting lines for silhouette edges

We identify the supporting line of the edges on the box silhouette using a variation of the Hough transform. The Hough transform (Hough, 1962; Duda and Hart, 1972) explores the duality between points and lines and essentially maps each feature pixel (e.g., silhouette pixel) to a set of lines in a discretized parameter space potentially passing through that pixel. Each cell of the parameter space accumulates the number of lines rasterized over it. At the end, the cells with the largest accumulated numbers (votes) represent the lines that best fit the set on input pixels. So, the problem of identifying line patterns in images is turned into the simpler problem of identifying peaks in a voting map representing the discretized parameter space. Unfortunately, the conventional Hough transform does not perform in real time. We have then developed a more efficient voting scheme for the Hough transform that allows it to achieve real-time performance (Fernandes and Oliveira, 2008).

In Eq. (1), the supporting lines $e_i$ are represented using the coefficients of the general equation of the line. However, the Hough transform represents image lines using the normal equation of the line given by

$$\rho = x \cos \theta + y \sin \theta$$

(22)

Therefore, one must to convert from one representation to another:

$$e_i = \begin{pmatrix} a_{e_i} \\ b_{e_i} \\ c_{e_i} \end{pmatrix} = \begin{pmatrix} \cos \theta_i \\ \sin \theta_i \\ -\rho_i \end{pmatrix}.$$  

(23)

As a result, the uncertainty associated with each supporting line comes from the uncertainties in $\rho_i$ and $\theta_i$ ($\sigma_{\rho_i}$ and $\sigma_{\theta_i}$, respectively). $\sigma_{\rho_i}$ and $\sigma_{\theta_i}$ are estimated, respectively, from the discretization steps $\delta_\rho$ and $\delta_\theta$ used to create the Hough transform voting map. In such a case, the distribution should be assumed uniform since the parameters of the normal equation of the line are in the interval $h_i = (\rho_i, \theta_i)^T \pm (\delta_\rho/2, \delta_\theta/2)^T$. The covariance matrix of such parameters is

$$\Lambda_{h_i} = \begin{pmatrix} \sigma_{\rho_i}^2 & \text{cov}(\rho_i, \theta_i) \\ \text{cov}(\rho_i, \theta_i) & \sigma_{\theta_i}^2 \end{pmatrix} = \begin{pmatrix} \delta_\rho^2/12 & 0 \\ 0 & \delta_\theta^2/12 \end{pmatrix}.$$  

(24)
3.1.2. Image coordinates of the laser dots

The projection of a laser beam on a box face produces a spot comprising several pixels (Fig. 7). We approximate the actual position of a laser dot \( p_l \) by the centroid of the pixels in the laser spot, and compute its covariance matrix using

\[
\Lambda_{p_l} = \begin{pmatrix} \sigma_{x_{p_l}}^2 & \text{cov}(x_{p_l}, y_{p_l}) \\ \text{cov}(x_{p_l}, y_{p_l}) & \sigma_{y_{p_l}}^2 \end{pmatrix} = \frac{1}{n^2} PP^T
\]  

(25)

where \( n \) is the number of pixels in the laser spot and \( P \) is a \( 2 \times n \) matrix containing the coordinates of these pixels after translating \( p_l \) to the origin.

3.1.3. Distance between laser beams

The uncertainty in the distance between the two laser beams comes from the radius \( r_{lb} \) of each beam (Fig. 8). Therefore, \( d_{lb} \) is in the interval \( d_{lb} \pm 2r_{lb} \) (i.e., one \( r_{lb} \) per laser beam). So, its variance can be estimated as

\[
\sigma_{d_{lb}}^2 = (2r_{lb})^2.
\]  

(26)

3.1.4. Intrinsic parameters of the camera

We estimate the intrinsic parameters of the camera using the Camera Calibration Toolbox for Matlab (Bouguet, 2005). Fortunately, the confidence interval of each camera parameter is computed by the toolbox during the calibration process, and the resulting covariance matrix (Eq. (27)) can be retrieved from the variables declared by the system.

\[
\Lambda_K = \begin{pmatrix}
\sigma_{\alpha_x}^2 & \text{cov}(\alpha_x, \alpha_y) & \text{cov}(\alpha_x, \gamma) & \text{cov}(\alpha_x, \alpha_x) & \text{cov}(\alpha_x, \alpha_y) \\
\text{cov}(\alpha_x, \alpha_y) & \sigma_{\alpha_y}^2 & \text{cov}(\alpha_y, \gamma) & \text{cov}(\alpha_y, \alpha_x) & \text{cov}(\alpha_y, \alpha_y) \\
\text{cov}(\alpha_x, \gamma) & \text{cov}(\alpha_y, \gamma) & \sigma_{\gamma}^2 & \text{cov}(\gamma, \alpha_x) & \text{cov}(\gamma, \alpha_y) \\
\text{cov}(\alpha_x, \alpha_x) & \text{cov}(\alpha_y, \alpha_x) & \text{cov}(\gamma, \alpha_x) & \sigma_{\alpha_x}^2 & \text{cov}(\alpha_x, \alpha_y) \\
\text{cov}(\alpha_x, \alpha_y) & \text{cov}(\alpha_y, \alpha_y) & \text{cov}(\gamma, \alpha_y) & \text{cov}(\alpha_x, \alpha_y) & \sigma_{\alpha_y}^2
\end{pmatrix}
\]  

(27)

3.1.5. Orientation of the laser beams

Finally, the last source of uncertainty to be considered is the orientation of the laser beams. We estimate the laser orientation with respect to the camera by sampling the 3D position of points which belong to the lines defined by one of the laser beams. Given a calibrated camera (i.e., whose intrinsic parameters are known), we project the laser beams on a calibration pattern and take some pictures of such a pattern at different distances. Then, we use the Camera Calibration Toolbox for Matlab to compute the 3D positions of the projected beams in the camera reference system. Combining pairs of points of the same beam, we generate \( n \) sample vectors \( J_k \). After normalizing the \( J_k \) vectors,
Fig. 8. The distance between the laser beams ($d_{lb}$) is measured on the scanner assembly. The radius ($r_{lb}$) of the beams introduces some uncertainty in the measurements.

we compute the mean vector $\bar{J}$. The orientation $L$ of the laser beams is computed by normalizing $\bar{J}$. The covariance matrix for $L$ is computed as

$$
\Lambda_L = \begin{pmatrix}
\sigma^2_{X_L} & \text{cov}(X_L, Y_L) & \text{cov}(X_L, Z_L) \\
\text{cov}(X_L, Y_L) & \sigma^2_{Y_L} & \text{cov}(Y_L, Z_L) \\
\text{cov}(X_L, Z_L) & \text{cov}(Y_L, Z_L) & \sigma^2_{Z_L}
\end{pmatrix} = \frac{1}{n^2 \|\bar{J}\|^2} M M^T
$$

(28)

where $M$ is the $3 \times n$ matrix with the vectors $J_k - \bar{J}$.

3.1.6. Setting up the input covariance matrix

The final set of 25 input variables is

$$
\vartheta = \{\rho_0, \theta_0, \rho_1, \theta_1, \rho_2, \theta_2, \rho_3, \theta_3, \rho_4, \theta_4, \rho_5, \theta_5, x_{p_0}, y_{p_0}, x_{p_1}, y_{p_1}, d_{lb}, \alpha_x, \alpha_y, \gamma, \alpha_y, X_L, Y_L, Z_L\}
$$

(29)

and the covariance matrix that models their uncertainties ($\Lambda_\vartheta$ in Eq. (21)) is given by the $25 \times 25$ matrix:

$$
\Lambda_\vartheta = \text{diag}(\Lambda_{h_0}, \Lambda_{h_1}, \Lambda_{h_2}, \Lambda_{h_3}, \Lambda_{p_0}, \Lambda_{p_1}, \sigma^2_{d_{lb}}, \Lambda_K, \Lambda_L).
$$

(30)

Next, we derive the equations necessary for propagating the uncertainties in the input parameters into uncertainties in the computed box dimensions.

3.2. Uncertainty propagation chain

Section 3.1 described how to estimate the uncertainty in the input variables and set up the covariance matrix $\Lambda_\vartheta$ (Eq. (30)). In order to propagate such uncertainties to the measured length $w_k$ for the $k$-th box edge, one needs to compute the Jacobian matrix of the function that computes $w_k$ and use it in Eq. (21). This is the following $1 \times 25$ Jacobian matrix:

$$
\nabla f_k = \begin{pmatrix}
\frac{\partial w_k}{\partial \rho_0} & \frac{\partial w_k}{\partial \theta_0} & \frac{\partial w_k}{\partial \rho_1} & \frac{\partial w_k}{\partial \theta_1} & \cdots & \frac{\partial w_k}{\partial X_L} & \frac{\partial w_k}{\partial Y_L} & \frac{\partial w_k}{\partial Z_L}
\end{pmatrix}.
$$

(31)

The following subsections and Figs. 9 and 11 describe the steps required for solving the partial derivatives in Eq. (31). Fig. 9 serves as a guide for the operations performed in the 2D image space (Sections 3.2.1–3.2.5). The operations regarding the 3D space are described in Fig. 11 (Sections 3.2.6–3.2.12). Both flows illustrate how uncertainty propagates from input variables to the final measurements.

3.2.1. Conversion from normal equation to general line equation

According to Fig. 9 and Section 3.1, the first stage of the processing obtains the normal equations of the supporting lines $e_i$, identified by the Hough transform, and represents them according to Eq. (23). Since this only involves a change in representation, the only derivatives to be calculated are

$$
\frac{\partial a_{e_i}}{\partial \theta_i} = -\sin \theta_i, \quad \frac{\partial b_{e_i}}{\partial \theta_i} = \cos \theta_i, \quad \frac{\partial c_{e_i}}{\partial \rho_i} = -1
$$

(32)
where $0 \leq i \leq 5$. The partial derivatives of $e_i$ with respect to the other variables in the set $\vartheta$ are equal to zero.

### 3.2.2. Silhouette vertices and vanishing points

The silhouette vertices and the vanishing points are computed as the intersections of support line pairs (Eqs. (1) and (6), respectively). After the division by the $w$ coordinate (Eq. (2)), the intersection $r$ between the supporting lines $u$ and $s$ is given by

$$
\begin{align*}
    r &= \left( x_r, y_r \right) = \frac{1}{w_r} \left( x_r, y_r \right) = \frac{1}{a_u b_s - a_s b_u} \left( a_s c_s - b_s c_u \right) \\
    &= \frac{1}{a_u b_s - a_s b_u} \left( a_s c_s - b_s c_u \right) (33)
\end{align*}
$$

where $u$ and $s$ are represented by the coefficients $(a_u, b_u, c_u)$ and $(a_s, b_s, c_s)$ in the general form. For a silhouette vertex $v_i$, $u = e_i$ and $s = e_{(i+5)\mod 6}$, where $0 \leq i \leq 5$. When three faces of the box are visible, the vanishing point $\omega_j$ is computed using $u = e_j$ and $s = e_{j+3}$, where $0 \leq j \leq 2$. When only two faces of the box are visible, $\omega_k$ is computed using $u = e_{k+1}$ and $s = e_{5-k}$, where $0 \leq k \leq 1$.

The partial derivatives for the coordinates of the point $r$ (Eq. (33)) are calculated as

$$
\begin{align*}
    \frac{\partial x_r}{\partial \vartheta} &= \frac{1}{w_r^2} \left( x_r a_s \frac{\partial a_s}{\partial \vartheta} + w_r b_u \frac{\partial b_u}{\partial \vartheta} - x_r b_s \frac{\partial a_u}{\partial \vartheta} - w_r a_s \frac{\partial a_u}{\partial \vartheta} \right) \\
    \frac{\partial y_r}{\partial \vartheta} &= \frac{1}{w_r^2} \left( x_r a_s \frac{\partial a_u}{\partial \vartheta} + w_r b_u \frac{\partial b_u}{\partial \vartheta} - x_r b_s \frac{\partial a_s}{\partial \vartheta} - w_r a_s \frac{\partial a_s}{\partial \vartheta} \right) (34)
\end{align*}
$$

### 3.2.3. Vanishing lines and supporting lines for inner edges

The coefficients of the vanishing lines (Eq. (7)) and of the support lines for the inner edges (Fig. 3) can be computed as a cross product between pairs of vectors representing 2D points in homogeneous coordinates. Assuming the homogeneous coordinate $w = 1$, the operation is simplified to

$$
\begin{align*}
    g &= \begin{pmatrix}
        a_g \\
        b_g \\
        c_g
    \end{pmatrix} = \begin{pmatrix}
        y_q - y_t \\
        x_t - x_q \\
        x_q y_t - x_t y_q
    \end{pmatrix} (36)
\end{align*}
$$

where $g$ is the resulting straight line and $q = (x_q, y_q, 1)$ and $t = (x_t, y_t, 1)$ are points on the straight line. For vanishing lines $\lambda_j$, $q = \omega_j$ and $t = \omega_{(j+1)\mod 3}$, where $0 \leq j \leq 2$. When three faces of the box are visible, the
Fig. 10. A case involving only two visible faces of a box. In detail, an image illustrating this case. Here, \(e_i, v_i, e'_0\) and \(f_i\) are, respectively: the supporting lines for silhouette edges, the silhouette vertices, the supporting line for the inner edge and the faces of the box. Parallel lines in 3D intersect in the vanishing point \(\omega_i\). The vanishing line \(\lambda_i\) of each one of the faces passes through the respective vanishing point and is parallel to \(e'_0\).

supporting lines for inner edges \(e'_j\) are computed using \(q = \omega_j\) and \(t = v_k\), where \(v_k\) is the closest silhouette vertex to \(\omega_j\). For the case involving only two visible faces, \(e'_0\) (the only inner edge) is defined by \(q = v_2\) and \(t = v_5\).

The partial derivatives for the coefficients of the line \(g\) (Eq. (36)) are computed as

\[
\frac{\partial a_g}{\partial \vartheta} = \frac{\partial y_q}{\partial \vartheta} - \frac{\partial y_t}{\partial \vartheta} \quad (37)
\]

\[
\frac{\partial b_g}{\partial \vartheta} = \frac{\partial x_t}{\partial \vartheta} - \frac{\partial x_q}{\partial \vartheta} \quad (38)
\]

\[
\frac{\partial c_g}{\partial \vartheta} = y_t \frac{\partial x_q}{\partial \vartheta} + x_q \frac{\partial y_t}{\partial \vartheta} - y_q \frac{\partial x_t}{\partial \vartheta} - x_t \frac{\partial y_q}{\partial \vartheta} \quad (39)
\]

3.2.4. Special case for vanishing line

When only two faces of the box are visible, the vanishing line of a box face is parallel to the supporting line of the (single) inner edge and passes through the vanishing point defined by parallel edges in 3D. This is shown by Fig. 10 and Eq. (40):

\[
\lambda_k = \left( \begin{array}{c} a_{\lambda_k} \\ b_{\lambda_k} \\ c_{\lambda_k} \end{array} \right) = \left( \begin{array}{c} a_{e'_0} \\ b_{e'_0} \\ - (a_{e'_0} x_{\omega_k} + b_{e'_0} y_{\omega_k}) \end{array} \right) \quad (40)
\]

where \(0 \leq k \leq 1\) and \(\omega_k\) are (from Eq. (33))

\[
\omega_0 = \left( \begin{array}{c} x_{\omega_0} \\ y_{\omega_0} \end{array} \right) = \frac{1}{a_e b_e - a_e c_e - b_e c_e} \left( \begin{array}{c} b_e c_e - b_e e_e \\ a_e c_e - a_e e_e \end{array} \right)
\]

\[
\omega_1 = \left( \begin{array}{c} x_{\omega_1} \\ y_{\omega_1} \end{array} \right) = \frac{1}{a_e b_e - a_e c_e - b_e c_e} \left( \begin{array}{c} b_e c_e - b_e e_e \\ a_e c_e - a_e e_e \end{array} \right)
\]
Fig. 11. Second part of the method for computing box dimensions from images. It represents stages (4)–(6) in Fig. 2. The dashed rectangles on the left come from Fig. 9. The Normal vectors (box B1 in the upper right corner) has been replicated to improve the flow organization. The input data (rounded rectangles) and the 2D features computed by previous stages contain uncertainties, which propagates to the resulting edge lengths.

The supporting lines in Fig. 10 are ordered clockwise in such way that $e_0$ and $e_3$ are parallel. The partial derivatives for the coefficients of the vanishing lines are

\[
\frac{\partial a_{lk}}{\partial \vartheta} = \frac{\partial a_{e_0'}}{\partial \vartheta} \quad \frac{\partial b_{lk}}{\partial \vartheta} = \frac{\partial b_{e_0'}}{\partial \vartheta} \quad \frac{\partial c_{lk}}{\partial \vartheta} = -\left( x_{ok} \frac{\partial a_{e_o'}}{\partial \vartheta} + y_{ok} \frac{\partial b_{e_o'}}{\partial \vartheta} + a_{e_0} \frac{\partial x_{ok}}{\partial \vartheta} + b_{e_0} \frac{\partial y_{ok}}{\partial \vartheta} \right). \tag{43}
\]

### 3.2.5. Inner vertex

According to Fig. 9, the only element not yet defined is the inner vertex $m_0$. The coordinates of $m_0$ are obtained using least squares to compute the intersection among the three inner edges. Each of these edges is supported by a line defined by one of the vanishing points and by the silhouette vertex between the two silhouette edges of the box used to compute such a vanishing point (Fig. 3).

\[
m_0 = \left( \begin{array}{c} x_{m_0} \\ y_{m_0} \end{array} \right) = \frac{1}{\hat{w}_{m_0}} \left( \begin{array}{c} \hat{x}_{m_0} \\ \hat{y}_{m_0} \end{array} \right) = \frac{1}{S_{aa}S_{bb} - S_{ab}^2} \left( S_{ab}S_{bc} - S_{bb}S_{ac} \right) \left( S_{ab}S_{ac} - S_{aa}S_{bc} \right) \right).
\]
where
\[ S_{aa} = \sum_{i=0}^{2} a_{e_i}^2, \quad S_{bb} = \sum_{i=0}^{2} b_{e_i}^2, \quad S_{ab} = \sum_{i=0}^{2} a_{e_i} b_{e_i}, \]
\[ S_{ac} = \sum_{i=0}^{2} a_{e_i} c_{e_i}, \quad S_{bc} = \sum_{i=0}^{2} b_{e_i} c_{e_i}. \]

The partial derivatives of the coordinates of \( m_0 \) with respect to \( \theta \) are then given by
\[
\frac{\partial x_{m_0}}{\partial \theta} = \frac{\hat{w}_{m_0}}{\hat{w}_{m_0}^2} \hat{X} - \hat{x}_{m_0} \hat{W}, \tag{45}
\]
\[
\frac{\partial y_{m_0}}{\partial \theta} = \frac{\hat{w}_{m_0}}{\hat{w}_{m_0}^2} \hat{Y} - \hat{y}_{m_0} \hat{W}, \tag{46}
\]

where
\[
\frac{\partial S_{aa}}{\partial \theta} = 2 \sum_{i=0}^{2} \left( a_{e_i} \frac{\partial a_{e_i}}{\partial \theta} \right), \quad \frac{\partial S_{bb}}{\partial \theta} = 2 \sum_{i=0}^{2} \left( b_{e_i} \frac{\partial b_{e_i}}{\partial \theta} \right),
\]
\[
\frac{\partial S_{ab}}{\partial \theta} = \sum_{i=0}^{2} \left( a_{e_i} \frac{\partial b_{e_i}}{\partial \theta} + b_{e_i} \frac{\partial a_{e_i}}{\partial \theta} \right), \quad \frac{\partial S_{ac}}{\partial \theta} = \sum_{i=0}^{2} \left( a_{e_i} \frac{\partial c_{e_i}}{\partial \theta} + c_{e_i} \frac{\partial a_{e_i}}{\partial \theta} \right),
\]
\[
\frac{\partial S_{bc}}{\partial \theta} = \sum_{i=0}^{2} \left( b_{e_i} \frac{\partial c_{e_i}}{\partial \theta} + c_{e_i} \frac{\partial b_{e_i}}{\partial \theta} \right),
\]

and
\[
\hat{X} = S_{ab} \frac{\partial S_{bc}}{\partial \theta} + S_{bc} \frac{\partial S_{ab}}{\partial \theta} - S_{bb} \frac{\partial S_{ac}}{\partial \theta} - S_{ac} \frac{\partial S_{bb}}{\partial \theta},
\]
\[
\hat{Y} = S_{ab} \frac{\partial S_{ac}}{\partial \theta} + S_{ac} \frac{\partial S_{ab}}{\partial \theta} - S_{aa} \frac{\partial S_{bc}}{\partial \theta} - S_{bc} \frac{\partial S_{aa}}{\partial \theta},
\]
\[
\hat{W} = S_{aa} \frac{\partial S_{bb}}{\partial \theta} + S_{bb} \frac{\partial S_{aa}}{\partial \theta} - 2 S_{ab} \frac{\partial S_{ab}}{\partial \theta}.
\]

So far, all points and lines were computed in image space and the only source of uncertainty comes from the Hough transform used to identify the supporting lines for silhouette edges. Now, the calculations will incorporate the camera parameters and some laser-beam-related data in 3D space.

### 3.2.6. Normal vectors

According to Eq. (5), the vector normal to each face is derived from its supporting plane’s vanishing line (Eq. (7)) and the intrinsic parameters of the camera (Eq. (4)), resulting in
\[
N_{II_j} = \frac{1}{\sqrt{A_i^2 + B_i^2 + C_i^2}} \begin{pmatrix} A_i \\ B_i \\ C_i \end{pmatrix} \tag{47}
\]

where
\[
\begin{pmatrix} A_i \\ B_i \\ C_i \end{pmatrix} = \begin{pmatrix} \alpha_x a_{\lambda_j} \\ -\gamma a_{\lambda_j} - \alpha_x b_{\lambda_j} \\ \alpha_x a_{\lambda_j} + \alpha_y b_{\lambda_j} + c_{\lambda_j} \end{pmatrix} \tag{48}
\]
with \( 0 \leq j \leq 2 \). Its partial derivatives are given by

\[
\frac{\partial A_{II_j}}{\partial \vartheta} = \frac{(B_i^2 + C_i^2) \frac{\partial A_i}{\partial \vartheta} - \left( B_i \frac{\partial B_i}{\partial \vartheta} + C_i \frac{\partial C_i}{\partial \vartheta} \right) A_i}{\sqrt{(A_i^2 + B_i^2 + C_i^2)^3}}
\]

\[ (49) \]

\[
\frac{\partial B_{II_j}}{\partial \vartheta} = \frac{(A_i^2 + C_i^2) \frac{\partial B_i}{\partial \vartheta} - \left( A_i \frac{\partial A_i}{\partial \vartheta} + C_i \frac{\partial C_i}{\partial \vartheta} \right) B_i}{\sqrt{(A_i^2 + B_i^2 + C_i^2)^3}}
\]

\[ (50) \]

\[
\frac{\partial C_{II_j}}{\partial \vartheta} = \frac{(A_i^2 + B_i^2) \frac{\partial C_i}{\partial \vartheta} - \left( A_i \frac{\partial A_i}{\partial \vartheta} + B_i \frac{\partial B_i}{\partial \vartheta} \right) C_i}{\sqrt{(A_i^2 + B_i^2 + C_i^2)^3}}
\]

\[ (51) \]

with

\[
\frac{\partial A_i}{\partial \vartheta} = \alpha_x \frac{\partial a_{x_j}}{\partial \vartheta} + a_{x_j} \frac{\partial \alpha_x}{\partial \vartheta},
\]

\[
\frac{\partial B_i}{\partial \vartheta} = -\gamma \frac{\partial a_{x_j}}{\partial \vartheta} + a_{x_j} \frac{\partial \gamma}{\partial \vartheta} - \alpha_y \frac{\partial b_{x_j}}{\partial \vartheta} - b_{x_j} \frac{\partial \alpha_y}{\partial \vartheta},
\]

\[
\frac{\partial C_i}{\partial \vartheta} = \alpha_x \frac{\partial a_{x_j}}{\partial \vartheta} + a_{x_j} \frac{\partial \alpha_x}{\partial \vartheta} + \alpha_y \frac{\partial b_{x_j}}{\partial \vartheta} + b_{x_j} \frac{\partial \alpha_y}{\partial \vartheta} + \frac{\partial c_{x_j}}{\partial \vartheta}.
\]

Given the orientation of the box faces (i.e., their normal vectors), one can compute \( D_{II_j} \), the coefficients of their corresponding plane equations, which will eliminate the projective ambiguity. This process is represented by Eqs. (9)–(16) and the corresponding uncertainty propagation is modeled in the following subsections.

3.2.7. Reprojecting points from the image onto the plane \( Z = 1 \)

According to Eq. (3), we have

\[
v'_i = \begin{pmatrix} x'_i \\ y'_i \\ z'_i \end{pmatrix} = \begin{pmatrix} \gamma (o_y - y_v) + \alpha_y (x_v - o_x) \\ (o_y - y_v) / \alpha_y \\ 1 \end{pmatrix}
\]

\[ (52) \]

where \( 0 \leq i \leq 5 \), and so

\[
\frac{\partial x'_i}{\partial \vartheta} = \left( \frac{\partial o_y}{\partial \vartheta} - \frac{\partial y_v}{\partial \vartheta} \right) \gamma + (o_y - y_v) \frac{\partial \gamma}{\partial \vartheta} + \left( \frac{\partial x_v}{\partial \vartheta} - \frac{\partial o_x}{\partial \vartheta} \right) \alpha_y
\]

\[
+ \left( (y_v - o_y) \gamma + (o_x - x_v) \alpha_y \right) \alpha_x \frac{\partial \alpha_y}{\partial \vartheta} + \left( y_v - o_y \right) \gamma \alpha_x \frac{\partial \gamma}{\partial \vartheta}
\]

\[ \frac{\partial y'_i}{\partial \vartheta} = \left( \frac{\partial o_y}{\partial \vartheta} - \frac{\partial y_v}{\partial \vartheta} \right) \alpha_y + (y_v - o_y) \frac{\partial \gamma}{\partial \vartheta}
\]

\[ \frac{\partial z'_i}{\partial \vartheta} = 0.
\]

(53) \hspace{1cm} (54) \hspace{1cm} (55)

One should notice that Eqs. (52)–(55) are also used to reproject the laser dots and the inner vertex \( m_0 \), by just replacing \( v_i \) by \( p_l \) \((0 \leq l \leq 1)\) and by \( m_0 \), respectively.

3.2.8. Distance between the laser dots

The distance between the laser dots is computed from the normal vector of the face containing the two dots (\( N_{II} \), Eq. (47)), from the known distance between the laser beams (\( d_{IB} \)), from the beam orientation (\( L \)), and from the
coordinates of the two reprojected dots on the \( Z = 1 \) plane (Eq. (12)), where \( N_L \) is the projection of \( N_I \) on the plane defined by the two laser beams (Eq. (19)). \( d_{ld} \) and \( N_L \) can be rewritten as

\[
d_{ld} = -\frac{d_{lb} \sqrt{A_{II}^2 + B_{II}^2 + C_{II}^2}}{C_{II}} \tag{56}
\]

where

\[
N_L = \begin{pmatrix} A_{II} \\ B_{II} \\ C_{II} \end{pmatrix} = \begin{pmatrix} A_{II} + mX_W \\ B_{II} + mY_W \\ C_{II} + mZ_W \end{pmatrix} \tag{57}
\]

and

\[
\begin{pmatrix} X_W \\ Y_W \\ Z_W \end{pmatrix} = \begin{pmatrix} Y_L - x_{p_1}Z_L + k \left( y_{p_0}Z_L - Y_L \right) \\ x_{p_1}Z_L - X_L + k \left( X_L - x_{p_0}Z_L \right) \\ y_{p_1}X_L - x_{p_1}Y_L + k \left( x_{p_0}Y_L - y_{p_0}X_L \right) \end{pmatrix}
\]

with

\[
m = -\frac{X_W A_{II} + Y_W B_{II} + Z_W C_{II}}{x_W^2 + y_W^2 + z_W^2}
\]

\[
k = \frac{k_1}{k_2} = \frac{x_{p_1}A_{II} + y_{p_1}B_{II} + C_{II}}{x_{p_0}A_{II} + y_{p_0}B_{II} + C_{II}}.
\]

Therefore, we obtain the partial derivatives as

\[
\frac{\partial d_{ld}}{\partial \theta} = \frac{\partial A_{II}}{\partial \theta} + m \frac{\partial X_W}{\partial \theta} + X_W \frac{\partial m}{\partial \theta} \tag{59}
\]

\[
\frac{\partial B_{II}}{\partial \theta} = \frac{\partial B_{II}}{\partial \theta} + m \frac{\partial Y_W}{\partial \theta} + Y_W \frac{\partial m}{\partial \theta} \tag{60}
\]

\[
\frac{\partial C_{II}}{\partial \theta} = \frac{\partial C_{II}}{\partial \theta} + m \frac{\partial Z_W}{\partial \theta} + Z_W \frac{\partial m}{\partial \theta} \tag{61}
\]

where \( \frac{\partial X_W}{\partial \theta} \), \( \frac{\partial Y_W}{\partial \theta} \) and \( \frac{\partial Z_W}{\partial \theta} \) are given by

\[
\frac{\partial X_W}{\partial \theta} = \frac{\partial Y_L}{\partial \theta} - Z_L \frac{\partial y_{p_1}}{\partial \theta} - y_{p_1} \frac{\partial Z_L}{\partial \theta} + \left( y_{p_0}Z_L - Y_L \right) \frac{\partial k}{\partial \theta} + k \left( y_{p_0} \frac{\partial Z_L}{\partial \theta} + Z_L \frac{\partial y_{p_0}}{\partial \theta} - \frac{\partial Y_L}{\partial \theta} \right)
\]

\[
\frac{\partial Y_W}{\partial \theta} = Z_L \frac{\partial x_{p_1}}{\partial \theta} + x_{p_1} \frac{\partial Z_L}{\partial \theta} - \frac{\partial X_L}{\partial \theta} + \left( X_L - x_{p_0}Z_L \right) \frac{\partial k}{\partial \theta} + k \left( \frac{\partial X_L}{\partial \theta} - x_{p_0} \frac{\partial Z_L}{\partial \theta} - Z_L \frac{\partial x_{p_0}}{\partial \theta} \right)
\]

\[
\frac{\partial Z_W}{\partial \theta} = X_L \frac{\partial y_{p_1}}{\partial \theta} + y_{p_1} \frac{\partial X_L}{\partial \theta} - Y_L \frac{\partial x_{p_1}}{\partial \theta} - x_{p_1} \frac{\partial Y_L}{\partial \theta} + \left( x_{p_0}Y_L - y_{p_0}X_L \right) \frac{\partial k}{\partial \theta} + k \left( x_{p_0} \frac{\partial Y_L}{\partial \theta} + Y_L \frac{\partial x_{p_0}}{\partial \theta} - y_{p_0} \frac{\partial X_L}{\partial \theta} - X_L \frac{\partial y_{p_0}}{\partial \theta} \right)
\]
where

\[
\frac{\partial m}{\partial \vartheta} = 2 \left( X_W A_{\Pi} + Y_W B_{\Pi} + Z_W C_{\Pi} \right) \frac{X_W \frac{\partial X_W}{\partial \vartheta} + Y_W \frac{\partial Y_W}{\partial \vartheta} + Z_W \frac{\partial Z_W}{\partial \vartheta}}{\left( X_W^2 + Y_W^2 + Z_W^2 \right)^2} - X_W \frac{\partial A_{\Pi}}{\partial \vartheta} + A_{\Pi} \frac{\partial X_W}{\partial \vartheta} + Y_W \frac{\partial B_{\Pi}}{\partial \vartheta} + B_{\Pi} \frac{\partial Y_W}{\partial \vartheta} + Z_W \frac{\partial C_{\Pi}}{\partial \vartheta} + C_{\Pi} \frac{\partial Z_W}{\partial \vartheta}
\]

\[
\frac{\partial k}{\partial \vartheta} = \frac{B_{\Pi} \left( y_{p_0} x_{p_1}' - y_{p_1}' x_{p_0} \right) + C_{\Pi} \left( x_{p_1}' - x_{p_0} \right) \frac{\partial A_{\Pi}}{\partial \vartheta}}{k_2^2} + \frac{A_{\Pi} \left( x_{p_0}' y_{p_1} - x_{p_1} y_{p_0}' \right) + C_{\Pi} \left( y_{p_0} - y_{p_0}' \right) \frac{\partial B_{\Pi}}{\partial \vartheta}}{k_2^2} + \frac{A_{\Pi} \left( x_{p_0} - x_{p_1}' \right) + B_{\Pi} \left( y_{p_0} - y_{p_1}' \right) \frac{\partial C_{\Pi}}{\partial \vartheta}}{k_2^2}
\]

(63)

\[
3.2.9. \text{3D coordinates of one of the laser dots}
\]

According to Eqs. (14) and (15), the partial derivatives for the 3D coordinates of one of the laser dots are given by

\[
\frac{\partial x_{p_1}}{\partial \vartheta} = Z \frac{\partial x_{p_1}'}{\partial \vartheta} + x_{p_1} \frac{\partial Z}{\partial \vartheta}
\]

(64)

\[
\frac{\partial y_{p_1}}{\partial \vartheta} = Z \frac{\partial y_{p_1}'}{\partial \vartheta} + y_{p_1} \frac{\partial Z}{\partial \vartheta}
\]

(65)

\[
\frac{\partial Z_{p_1}}{\partial \vartheta} = \frac{\partial Z}{\partial \vartheta}
\]

(66)

where \(\frac{\partial x_{p_1}'}{\partial \vartheta}\) and \(\frac{\partial y_{p_1}'}{\partial \vartheta}\) are given by Eqs. (53) and (54); and

\[
\frac{\partial Z}{\partial \vartheta} = \frac{d_{id} \left( 2 \left( ak^2 - 2bk + c \right) \frac{\partial a}{\partial \vartheta} \right) - d_{id} \left( k^2 \frac{\partial a}{\partial \vartheta} - 2k \frac{\partial b}{\partial \vartheta} + \frac{\partial c}{\partial \vartheta} - 2 \left( b - ak \right) \frac{\partial k}{\partial \vartheta} \right)}{2, \sqrt{d_{id}^2 \left( ak^2 - 2bk + c \right)^3}}
\]

(67)

while

\[
\frac{\partial a}{\partial \vartheta} = 2 \left( x_{p_0}' \frac{\partial x_{p_0}'}{\partial \vartheta} + y_{p_0}' \frac{\partial y_{p_0}'}{\partial \vartheta} \right)
\]

\[
\frac{\partial b}{\partial \vartheta} = x_{p_0} ' \frac{\partial x_{p_0}'}{\partial \vartheta} + x_{p_1} ' \frac{\partial x_{p_1}'}{\partial \vartheta} + y_{p_0} ' \frac{\partial y_{p_0}'}{\partial \vartheta} + y_{p_1} ' \frac{\partial y_{p_1}'}{\partial \vartheta}
\]

\[
\frac{\partial c}{\partial \vartheta} = 2 \left( x_{p_1} ' \frac{\partial x_{p_1}'}{\partial \vartheta} + y_{p_1} ' \frac{\partial y_{p_1}'}{\partial \vartheta} \right)
\]

and \(\frac{\partial k}{\partial \vartheta}\) is calculated according to Eq. (63).

\[
3.2.10. \text{Planes containing the faces}
\]

From the normal vector of the face containing the laser dots (Eq. (47)) and the coordinates of one point onto such a face (Eq. (15)), finding the fourth coefficient of the plane equation is trivial. It is done just by substituting the
coordinates of the point into the plane equation and solving for $D_\Pi$:

$$
\Pi = \begin{pmatrix}
A_\Pi \\
B_\Pi \\
C_\Pi \\
D_\Pi
\end{pmatrix} = \begin{pmatrix}
A_\Pi \\
B_\Pi \\
C_\Pi \\
-(A_\Pi X_P + B_\Pi Y_P + C_\Pi Z_P)
\end{pmatrix}.
$$

(68)

The partial derivatives of $A_\Pi$, $B_\Pi$ and $C_\Pi$ are exactly the same of derivatives of the normal vector coordinates (Eqs. (49)–(51), respectively). The partial derivative of $D_\Pi$ is given by

$$
\frac{\partial D_\Pi}{\partial \vartheta} = - \left( A_\Pi \frac{\partial X_P}{\partial \vartheta} + X_P \frac{\partial A_\Pi}{\partial \vartheta} + B_\Pi \frac{\partial Y_P}{\partial \vartheta} + Y_P \frac{\partial B_\Pi}{\partial \vartheta} + C_\Pi \frac{\partial Z_P}{\partial \vartheta} + Z_P \frac{\partial C_\Pi}{\partial \vartheta} \right).
$$

(69)

The plane equations containing the other faces of the box are obtained in a similar way, by substituting a shared vertex into Eq. (68). Next, we present the partial derivatives of the 3D coordinates of the vertices of the box.

### 3.2.11. 3D coordinates of the box vertices

Once we have calculated the coefficients of the plane equation of a given box face, they can be used to recover the 3D coordinates of the vertices of that face:

$$
V = \begin{pmatrix}
X_V \\
Y_V \\
Z_V
\end{pmatrix} = Z \begin{pmatrix}
x' \\
y'\\
1
\end{pmatrix}
$$

(70)

where $v'$ is the projection of the vertex on the plane $Z = 1$ (Eq. (52)) and $Z$ is given by

$$
Z = - \frac{D_\Pi}{x' A_\Pi + y' B_\Pi + C_\Pi}.
$$

(71)

So, the derivatives of coordinates of $V$ are calculated:

$$
\frac{\partial X_V}{\partial \vartheta} = Z \frac{\partial x'}{\partial \vartheta} + x' \frac{\partial Z}{\partial \vartheta}
$$

(72)

$$
\frac{\partial Y_V}{\partial \vartheta} = Z \frac{\partial y'}{\partial \vartheta} + y' \frac{\partial Z}{\partial \vartheta}
$$

(73)

$$
\frac{\partial Z_V}{\partial \vartheta} = \frac{\partial Z}{\partial \vartheta}
$$

(74)

where

$$
\frac{\partial Z}{\partial \vartheta} = \frac{D_\Pi}{(x' A_\Pi + y' B_\Pi + C_\Pi)^2} \left( x' \frac{\partial A_\Pi}{\partial \vartheta} + y' \frac{\partial B_\Pi}{\partial \vartheta} + C_\Pi \frac{\partial x'}{\partial \vartheta} + A_\Pi \frac{\partial y'}{\partial \vartheta} + B_\Pi \frac{\partial C_\Pi}{\partial \vartheta} \right)
$$

$$
- \left( \frac{1}{x' A_\Pi + y' B_\Pi + C_\Pi} \right) \frac{\partial D_\Pi}{\partial \vartheta}.
$$

(75)

### 3.2.12. Box dimensions

After the 3D coordinates of all vertices have been recovered, the box dimensions can be estimated as the distances between the pairs $(V_i, V_j)$ of vertices defining the visible edges:

$$
w_k = \| V_j - V_i \| = \sqrt{(X_{V_j} - X_{V_i})^2 + (Y_{V_j} - Y_{V_i})^2 + (Z_{V_j} - Z_{V_i})^2}.
$$

(76)

Given the function for computing $w_k$ and the partial derivatives of $V_i$ and $V_j$ (Eqs. (72)–(75)), the Jacobian matrix $\nabla f$ (Eq. (21)) can be finally calculated:

$$
\nabla f = \frac{\partial w_k}{\partial \vartheta}.
$$
Given the covariance matrix $\Lambda_\theta$ (Section 3.1) and the Jacobian matrix $\nabla f$, we estimate the uncertainty in the dimensions of the box using Eq. (21).

In this equation, $\Lambda_\theta$ stores the uncertainty in the input data, while $\nabla f$ weights the influence of these uncertainties in the computed dimension of each edge $w_k$.

4. Results

In order to evaluate the effectiveness of the proposed approach, we carried out some experiments. First, using a sampling-based method, we have demonstrated that our approach for computing box dimensions from single images is accurate (Section 4.1). Then, we have compared the sampling-based results to the ones obtained using uncertainty propagation and have shown that they are equivalent (Sections 4.2–4.4).

For the experiments, we selected six boxes and classified them as good, average or bad ones, according to their quality (i.e., similarity to a parallelepiped). Pictures of these boxes are shown in Fig. 12. For each box, we chose 30 different positions and, for each position, we collected a set of at least 30 images, while trying to hold the scanner prototype still as much as possible (there was some hand shaking). As the system collected the images, it also computed the dimensions of the silhouette edges of the boxes ($w_{k,i}$ in Eq. (76)) and their associated uncertainties ($\sigma_{w_{k,i}} = \sqrt{\text{var}(w_{k,i})}$), where the variance $\text{var}(w_{k})$ comes from the $k$-th diagonal element of matrix $\Lambda_w$, Eq. (21)). All measurements were in centimeters. Note that $\sigma_{w_{k,i}}$ comes from the uncertainty propagation applied to edge $k$ over a single image $i$.

In order to compare the results obtained with the uncertainty propagation approach to a sampling-based method, for any given box position, we took the dimensions computed with our method (Section 2) and calculated the mean ($\bar{w}_k$) and standard deviation ($\sigma_{w_k}$) of each silhouette edge of the box.
4.1. Confidence intervals for the sampling-based approach

The confidence intervals for the sampling-based approach were computed as

$$CI = \left[ \bar{w}_k - t_{\gamma} \frac{s_{w_k}}{\sqrt{n}}, \bar{w}_k + t_{\gamma} \frac{s_{w_k}}{\sqrt{n}} \right]$$

(78)

where \(n\) is the size of sample and \(t_{\gamma}\) is a Student’s-t variable with \(n - 1\) degrees of freedom, such that the probability of a measure \(\bar{w}_k\) belongs to \(CI\) is \(\gamma\).

Fig. 13 shows the computed confidence intervals for \(\gamma = 99.5\%\). Note that the values of the actual dimensions of the boxes (horizontal line segments) fall inside most of these confidence intervals, indicating accurate measurements. Boxes (a) and especially (b) are the ones with tightest confidence intervals. Those are well constructed boxes (Fig. 12). Wider confidence intervals were obtained for boxes with bent faces and edges, like the bad box (e). The only case for which the actual dimensions did not fall inside their confidence interval where for two edges of the bad box (f). This box has a cardboard lid that deforms the box silhouette, shifting the computed mean values away from the true ones.

4.2. Ratio between the standard deviations \(s_{w_k}\) and \(\sigma_{w_k,i}\)

For any given box position, let \(r = s_{w_k}/\sigma_{w_k,i}\), where \(s_{w_k}\) and \(\sigma_{w_k,i}\) were computed for edge \(k\) using the entire set of images acquired for that position. \(r\) provides a measure of the relative conservativeness between the sample-based and the uncertainty propagation approaches. Fig. 14 shows a histogram of the ratio \(r\) computed for all grabbed images (98,310 edge samples in 16,385 images). The green, blue and red parts of the bins are the ratios computed for good, average and bad boxes, respectively. Notice that there is a well-defined peak centered around \(r = 1\), but slightly to the left. In fact, we have \(r \leq 1\) for 57.54\% of the measurements. This means that the uncertainty propagation approach is equivalent to and slightly more conservative than the sampling-based approach. Almost all the samples of good boxes are inside the \([0, 2]\) range.

4.3. Dispersion of the standard deviations \(s_{w_k}\) and \(\sigma_{w_k,i}\)

When using a measurement device, one would like the confidence interval computed by the system not to vary too much from one measurement to another. Here, we compare the dispersion (variance) of the computed \(s_{w_k}\) and \(\sigma_{w_k,i}\) values. Our results show that \(\text{var}(s_w) = 0.4019\) and \(\text{var}(\sigma_w) = 0.0925\). Therefore, the values obtained with the uncertainty propagation approach are subject to less variability.
4.4. Relative errors

Section 4.2 has shown that the uncertainty propagation method is more conservative than the conventional sampling-based approach. However, a too conservative technique would overestimate the confidence intervals, making the measurements of little use. Here, we show that the ratio between the propagated uncertainty ($\sigma_{w_k,i}$) and the estimated length ($w_{k,i}$) for any given silhouette edge is small. This is illustrated in Fig. 15, which presents a histogram of these ratios. Note that for good and average boxes (green and blue bins, respectively) the ratio is always smaller than 4.5%, with most values under 3%. For bad boxes (red bins), the ratio is under 8% (a few isolated bins at the right of Fig. 15). These results indicate that the estimated uncertainty is relatively small compared to the dimensions of the edges.
4.5. Impact of the uncertainty from each input variable

The relative impact of the uncertainty from each input variable on the final measurement can be estimated using the error propagation model of Eq. (21). The absolute contribution of any given input variable \( i \) (or group of input variables) is obtained using a covariance matrix \( \Lambda_\vartheta \) (in Eq. (21)) for which the only non-null elements are the ones related to \( i \) in Eq. (29). The relative impact of variable \( i \) is then obtained by dividing its absolute impact by the sum of the absolute impacts of all input variables.

Fig. 16 shows the relative impact of each group of input variables assuming that they all have the same uncertainty and are uncorrelated. Under these assumptions, the biggest sources of uncertainty in the final measurements are the uncertainties found in the \((\theta, \rho)\) parameters of the supporting lines of the silhouette edges. This can be explained by the fact that the supporting lines are calculated in the initial stages of the computational flow (Section 2) and their values are used in many subsequent stages of the algorithm.

Fig. 17 shows a more realistic scenario illustrating the results obtained using the estimated uncertainties for the input parameters of our scanner prototype. Note that in this case, the biggest sources of uncertainties are the parameters of the supporting lines and the distance between the two laser beams. Thus, one can reduce the uncertainty in the measurements produced by our scanner prototype by, for instance, increasing the resolution of the voting map used for the Hough transform, or by moving the parallel laser beams apart (this reduces the relative error in measuring the distance between the two laser beams).

The data shown in Figs. 16 and 17 were computed using images of the wooden box constructed especially for this project (Fig. 12, b). Such a box is a perfect parallelepiped and contains none of the imperfections commonly found in cardboard boxes. For these experiments, we used the following parameter values:

- The discretization steps of the Hough transform voting map (Eq. (24)) were \( \delta_\rho = \delta_\vartheta = 0.5 \).
- The laser beams were 15.8 cm apart (\( d_{lb} \) in Eq. (12)) and the radius of each beam (\( r_{lb} \) in Eq. (25)) was estimated as 0.1 cm.
- The intrinsic parameters of the camera (Eq. (4)) were \( \alpha_x = 2222.2974, \alpha_y = 2223.1648, \gamma = 0, \alpha_x = 60.4526 \) and \( o_y = 56.5677 \); and their covariance matrix (Eq. (27)) was

\[
\Lambda_K = \begin{pmatrix}
22.7536 & 22.1284 & -0.0039 & 1.1464 & -3.1960 \\
22.1284 & 22.6816 & -0.0073 & 2.0672 & -2.1253 \\
-0.0039 & -0.0073 & 0.0006 & -0.0002 & 0.0081 \\
1.1464 & 2.0672 & -0.0002 & 76.2808 & 2.8090 \\
-3.1960 & -2.1253 & 0.0081 & 2.8090 & 57.8346
\end{pmatrix}.
\]
• The lasers orientation was estimated as \( \mathbf{L} = (-0.0125, 0.0121, 0.9998)^T \), and its covariance matrix was

\[
\Lambda_L = \begin{pmatrix}
6.2189 \times 10^{-7} & 2.6162 \times 10^{-7} & 6.7550 \times 10^{-9} \\
2.6162 \times 10^{-7} & 5.7027 \times 10^{-7} & -3.8055 \times 10^{-9} \\
6.7550 \times 10^{-9} & -3.8055 \times 10^{-9} & 1.7543 \times 10^{-10}
\end{pmatrix}.
\]

It is important to note that the image coordinates of the laser dots and their uncertainty are computed on-the-fly for any given image (Section 3.1.2). Also, the resolution of the input images was 640 \times 480 pixels and they were acquired from distances varying from 1.7 to 3.0 m from the box.

5. Summary and conclusions

We have described an automatic method for computing box dimensions from single perspective projection images. Using a sampling-based approach, we have shown that such a method is accurate, with the actual box dimensions falling inside the computed confidence intervals, except for two edges of one box presenting some degeneracies (box (f) in Fig. 12). For this method of computing box dimensions, we presented an analytical derivation of how to propagate the uncertainties in the input parameters into the resulting dimensions. Comparing the propagated uncertainties with the variances obtained with the use of the sampling-based approach, our experiments show that the results of the analytical technique are in accordance with the ones obtained by the conventional sampling process. We have also shown that the uncertainty propagation approach is slightly more conservative than the sampling-based one. This means that if the actual dimensions are inside the confidence intervals defined by the sampling-based approach; they are also inside the intervals specified by the propagation-based one. On the other hand, such conservativeness does not tend to produce loose intervals. As we have shown, the ratio between the propagated standard deviation and the actual box dimensions is under 5% for essentially all the tested boxes.

Our experiments demonstrated that the uncertainty propagation approach not only can replace a sampling-based one when computing dimensions of box from single perspective images, but also does so with several considerable advantages. First, it avoids the need of collecting large amounts of samples for computing the statistics. This saves a considerable amount of time and work. Second, it instantly takes into account the changes in the input data and acquisition conditions. Third, it is computationally very efficient, allowing the uncertainty estimates to be obtained from a single image in real time.

More than having described a set of experiments and demonstrated the advantages of using uncertainty propagation in the context of our specific application, we hope to have convinced and inspired other researchers and practitioners
as regards the great potential of using uncertainty propagation in applications involving real-time measurements, in particular for the ones aiming to make measurements from images.

Acknowledgments

This work was partially sponsored by CNPq-Brazil (477344/2003-8) and Petrobras (502009/2003-9). Leandro A. F. Fernandes is sponsored by a CNPq-Brazil fellowship (142627/2007-0). Roberto da Silva has a CNPq-Brazil fellowship. The authors would like to thank Microsoft Brazil for additional support, and the anonymous reviewers for their comments and insightful suggestions.

References