Rigorous accuracy bounds for calibrated stereo reconstruction

M. Farenzena, A. Busti, A. Fusiello, A. Benedetti
Dipartimento di Informatica - Università di Verona

Abstract

We deal with the problem of obtaining rigorous bounds to the position of 3-D points computed by stereo triangulation when both the camera matrix and the image points are affected by uncertainty. By "rigorous bounds" we mean that the true unknown 3-D points are guaranteed *to lie within the given intervals. To this end we first model the calibration process by assuming a bounded error in the localization of the reference points in the image, then we narrow the entries of the camera matrix. Finally, we apply triangulation and obtain cuboids that bound points coordinates. We concentrated two state-of-the art methods for the solution of linear system of equations, namely INTLAB's and Shary's methods. Empirical comparison shows that the latter always provides sharper error bounds, in this application.*

1 Introduction

Being an empirical science, Computer Vision has to deal with errors affecting measurements. The problem of propagating errors from input data to results in Computer Vision has been addressed in two well-known books [2, 6], and in a landmark workshop [1].

Established techniques are based on statistical analysis of error propagation: given an input error distribution, a characterization of the output error distribution is produced. In principle this should be done by propagating the input distribution through the various stages of the algorithm, but this approach is not practical except for simple algorithms. An approximation of this method is usually employed, assuming that the distribution is characterized by its first and second moments only. In [2, 15] the authors address the problem of propagating the covariance of $\mathbf{x}$ through $f(\mathbf{x})$ when f is known explicitly (by linearisation) and when f is specified implicitly as the minimiser of a scalar cost function. Statistical analysis of geometric algorithms is also developed in [6].

In this work we take a different approach, based on *Interval Arithmetic*. Data are represented by intervals containing the real value, and the width of the interval represent a bound to the error. Arithmetic operations are then performed on these intervals, with the guarantee that the resulting interval contains the exact result.

Marik et al. [9] first suggested to use Interval Arithmetic for the study of error propagation in Computer Vision. These authors propose two error models, one based on covariance propagation, and one (called min/max value propagation) based on Interval Arithmetic. As the bounds obtained with direct application of interval arithmetic rules are usually too pessimistic, they conclude that "the min/max model is very appropriate for studying the effect of the machine precision on some computation." [9] As we shall see, however, a careful selection of suitable techniques allowed us to use Interval Arithmetic as a tool for obtaining realistic bounds on the output error even in presence of a significant error on the input data.

In this paper we deal with error propagation in three-dimensional (3-D) reconstruction from stereo, i.e we seek a bound on the accuracy of the position of 3-D points obtained by triangulation. In this process we assume that both the camera matrices and the corresponding points in the image are affected by a bounded error.

2 Problem formulation

This section briefly recalls the mathematical developments of stereo reconstruction that are relevant to our work. For more details see [2].

Let $\mathbf{w} = (x, y, z)^\mathsf{T}$ be the coordinates of a 3-D point in the world reference frame. A *pinhole camera* projects the point onto the image plane. The coordinates $\mathbf{m} = (u, v)^\mathsf{T}$ of the projected image point are given by the *perspective projection* equation:

$$u = \frac{\mathbf{p}_1^\mathsf{T}\mathbf{w} + p_{1,4}}{\mathbf{p}_3^\mathsf{T}\mathbf{w} + p_{3,4}}, \quad v = \frac{\mathbf{p}_2^\mathsf{T}\mathbf{w} + p_{2,4}}{\mathbf{p}_3^\mathsf{T}\mathbf{w} + p_{3,4}}. \tag{1}$$

where

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_1^\mathsf{T} & p_{1,4} \\ \mathbf{p}_2^\mathsf{T} & p_{2,4} \\ \mathbf{p}_3^\mathsf{T} & p_{3,4} \end{pmatrix} \tag{2}$$

is a full-rank 3 × 4 matrix that models the pinhole camera, and it is called *camera matrix*.

2.1 Calibration

Calibration consists in computing the elements of the camera matrix $\mathbf{P}$. The unknown camera matrix has 12 elements, however, being defined up to a scale factor it depends on 11 parameters only. We can choose $p_{34} = 1$, thus reducing the number of unknowns to 11. Given N reference points, not coplanar, each correspondence between an image point $\mathbf{m}_i = (u_i, v_i)^\mathsf{T}$, and a reference point $\mathbf{w}_i$ gives a pair of equations, derived from (1):

$$\begin{pmatrix} \mathbf{w}_i^\mathsf{T} & 1 & \mathbf{0} & 0 & -u_i\mathbf{w}_i^\mathsf{T} \\ \mathbf{0} & 0 & \mathbf{w}_i^\mathsf{T} & 1 & -v_i\mathbf{w}_i^\mathsf{T} \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ p_{1,4} \\ \mathbf{p}_2 \\ p_{2,4} \\ \mathbf{p}_3 \end{pmatrix} = \begin{pmatrix} u_i \\ v_i \end{pmatrix}. \quad (3)$$

Therefore for N points we obtain a linear system of $2N$ equations in 11 unknowns: 6 non coplanar points are sufficient. In practice more points are available, and one has to solve a linear least-squares problem.
If we assume that the measurement of pixel coordinates $\mathbf{m}_i$ are affected by a bounded error (e.g., ± 0.5 pixel) this translates in bounding some of the entries of the coefficients matrix and the right hand side vector of the linear system (3).

2.2 Triangulation

Given the matrices of the two cameras and the coordinates of the projections on the image planes of a 3-D point M, coordinates of M can be recovered by a simple linear algorithm, called the *Linear-LS* method [4].
Let $\mathbf{P}$ and $\mathbf{P}'$ be the two camera matrices, let $\mathbf{w}$ be the unknown coordinates of the 3-D point, and let $\mathbf{m} = (u, v)^\mathsf{T}$ and $\mathbf{m}' = (u', v')^\mathsf{T}$ be the image coordinates of a conjugate pair. From (1) we obtain a linear system of four equations in the unknown 3-vector $\mathbf{w}$:

$$\begin{pmatrix} (\mathbf{p}_1 - u\mathbf{p}_3)^\mathsf{T} \\ (\mathbf{p}_2 - v\mathbf{p}_3)^\mathsf{T} \\ (\mathbf{p}'_1 - u'\mathbf{p}'_3)^\mathsf{T} \\ (\mathbf{p}'_2 - v'\mathbf{p}'_3)^\mathsf{T} \end{pmatrix} \mathbf{w} = \begin{pmatrix} -p_{1,4} + up_{3,4} \\ -p_{2,4} + vp_{3,4} \\ -p'_{1,4} + u'p'_{3,4} \\ -p'_{2,4} + v'p'_{3,4} \end{pmatrix}. \quad (4)$$

Again, a bounded error affecting $\mathbf{P}$, $\mathbf{m}$ and $\mathbf{m}'$ translates into bounds to the entries of the coefficient matrix and the right hand side vector of (4).
In the next section we see how arithmetic operations can be defined on bounded intervals, in order to find rigorous bounds to the solution of numerical problems.

3 Interval Arithmetic

Interval Arithmetic [10] is an arithmetic defined on intervals, rather than on real numbers. It was specifically introduced for bounding the measurement errors of physical quantities for which no statistical distribution was known.

In the sequel we shall follow the notation used in [14], where intervals are denoted by boldface. Underscores and overscores will represent respectively lower and upper bounds of intervals. The midpoint of an interval $\boldsymbol{x}$ is denoted by $\mathrm{m}(\boldsymbol{x})$. $\mathbb{IR}$ and $\mathbb{IR}^n$ stand respectively for the set of real intervals and the set of interval vectors of dimension n. If $f(x)$ is a function defined over an interval $\boldsymbol{x}$ then $\boldsymbol{f}^u(\boldsymbol{x})$ denotes the range of $f(x)$ over $\boldsymbol{x}$. Finally, the topological interior of a set S is denoted by $\mathrm{int}(S)$.
If $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, a binary operation in the *ideal interval arithmetic* between $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as:

$$\boldsymbol{x} \,\mathrm{op}\, \boldsymbol{y} \triangleq \{x \,\mathrm{op}\, y \mid x \in \boldsymbol{y} \wedge y \in \boldsymbol{y}\}, \forall\, \mathrm{op} \in \{+, -, \times, \div\}.$$

Thus, the ranges of the four elementary interval operations are exactly the ranges of the corresponding real operations.

Definition 1 *An* interval extension*, denoted by $\boldsymbol{f}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, of a real function $f(x_1, x_2, \ldots, x_n)$ is defined as any function of the n intervals $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ that evaluates to the value of f when its arguments are the degenerate intervals $x_1, x_2, \ldots, x_n$:*

$$\boldsymbol{f}(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n). \quad (5)$$

The *natural* interval extension of a function is obtained by replacing variables with intervals and executing all operations according to the rules above.
For instance, $\boldsymbol{f}_1(\boldsymbol{x}) = \boldsymbol{x}^2 - \boldsymbol{x}$, and $\boldsymbol{f}_2(\boldsymbol{x}) = \boldsymbol{x}(\boldsymbol{x} - 1)$ are all interval extensions of $f(x) = x^2 - x = x(x-1)$.
With these two definitions we can state the following theorem, known as the fundamental theorem of Interval Arithmetic [10]:

Theorem 1 *Let $\boldsymbol{f}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ be the natural interval extension of a real function $f(x_1, \ldots, x_n)$. If $\boldsymbol{x}_i \subset \boldsymbol{y}_i, (i = 1, \ldots, n)$ then*

$$\boldsymbol{f}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \subset \boldsymbol{f}(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n). \quad (6)$$

From this theorem it follows immediately that

$$\boldsymbol{f}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \supset \boldsymbol{f}^u(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n). \quad (7)$$

In the previous example, by setting $\boldsymbol{x} = [0, 1]$ we have$\boldsymbol{f}_2(\boldsymbol{x}) = [0, 1]\,([0, 1] - 1) = [0, 1]\,[-1, 0] = [-1, 0]$, which includes the exact range $\boldsymbol{f}^u([0, 1]) = [-1/4, 0]$.

3.1 Interval linear systems

Interval linear systems are useful to calculate rigorous bounds to the solutions of linear systems of equations. They have the form $\boldsymbol{A}x = \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{IR}^{n \times n}$ and $\boldsymbol{b} \in \mathbb{IR}^n$. The *solution set* is defined as

$$\Sigma(\boldsymbol{A}, \boldsymbol{b}) = \{x : \exists A \in \boldsymbol{A} \text{ and } \exists b \in \boldsymbol{b} \text{ s.t. } Ax = b\}. \quad (8)$$

In general $\Sigma(\boldsymbol{A}, \boldsymbol{b})$ is a star-shaped polygonal set, with up to 2^n spikes for a system of dimension n [13]. Thus, we must content ourselves with computing the *interval hull* $\square\Sigma(\boldsymbol{A}, \boldsymbol{b})$ of the solution, i.e., the smallest hyperrectangle containing the solution set. It has been shown [5] that the calculation of the *interval hull* is an NP-complete problem. However, practical methods give a reasonable inclusion of the solution set with a cost of $O(n^3)$. Among these we have considered the method implemented by the `verifylss` function in the INTLAB toolbox for MATLAB [17, 3], and the method introduced by S.P. Shary [18].

INTLAB method. The first stage of the algorithm implemented by the `verifylss` function [3] is an iterative method introduced by Rump [16], based on the well-known Krawcyzk operator [8] (see also [13]).
Assuming that there is an interval vector $\boldsymbol{x}^i$ such that $\square\Sigma(\boldsymbol{A}, \boldsymbol{b}) \subseteq \boldsymbol{x}^i$, then $\forall \tilde{A} \in \boldsymbol{A}, \tilde{b} \in \boldsymbol{b}$:

$$\tilde{A}^{-1}\tilde{b} = C\tilde{b} + (I - C\tilde{A})\tilde{A}^{-1}\tilde{b} \in C\boldsymbol{b} + (I - C\boldsymbol{A})\boldsymbol{x}^i, \quad (9)$$

where $C = \mathrm{m}(\boldsymbol{A})^{-1}$ is a preconditioner. Hence,

$$\begin{array}{l} \square\Sigma(\boldsymbol{A}, \boldsymbol{b}) \subseteq \boldsymbol{x}^i \Rightarrow \\ \square\Sigma(\boldsymbol{A}, \boldsymbol{b}) \subseteq (C\boldsymbol{b} + (I - C\boldsymbol{A})\boldsymbol{x}^i) \cap \boldsymbol{x}^i, \end{array} \quad (10)$$

and this gives the *Krawcyzk iteration*:

$$\boldsymbol{x}^{i+1} = (C\boldsymbol{b} + (I - C\boldsymbol{A})\boldsymbol{x}^i) \cap \boldsymbol{x}^i. \quad (11)$$

Rump's method, instead, proceed by enclosing the error with respect to an approximate solution $\tilde{x} = C\,\mathrm{m}(\boldsymbol{b})$. Applying (10) to an enclosure $\boldsymbol{d}^i$ of $\square\Sigma(\boldsymbol{A}, \boldsymbol{b} - \boldsymbol{A}\tilde{x})$, gives the following iteration

$$\boldsymbol{d}^{i+1} = (C(\boldsymbol{b} - \boldsymbol{A}\tilde{x}) + (I - C\boldsymbol{A})\boldsymbol{d}^i) \cap \boldsymbol{d}^i. \quad (12)$$

The solution to the original problem is $\tilde{x} + \boldsymbol{d}^i$.
If there is no success after seven iterations, the algorithm described in [11] is applied.

Shary's method Shary introduced the *algebraic approach* for enclosing the solution of a linear system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. This methods finds an *algebraic solution*, which is an interval vector $\boldsymbol{x}$ satisfying the system whenever all the operations are performed according to the rules of the extended interval arithmetic [7] $\mathbb{IR}_{ex}$, which is obtained by adding *improper* intervals $[\underline{x}, \overline{x}]$, $\underline{x} > \overline{x}$ to the set $\mathbb{IR} = \{[\underline{x}, \overline{x}] \mid \underline{x}, \overline{x} \in \mathbb{R},\ \underline{x} \leq \overline{x}\}$ of *proper* intervals. Shary's method is based on the fixed point equation

$$\boldsymbol{x} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{b}, \quad (13)$$

which resolves into finding an algebraic solution to the interval equation: $\boldsymbol{C}\boldsymbol{x} \ominus \boldsymbol{x} + \boldsymbol{b} = 0$ where $\boldsymbol{C} = I - \boldsymbol{A}$ and '$\ominus$' denotes the *inner subtraction*, defined by $\boldsymbol{x} \ominus \boldsymbol{y} \triangleq [\underline{x} - \underline{y}, \overline{x} - \overline{y}]$. Unfortunately, most of the existing computational approaches are not directly applicable to this problem, because $\mathbb{IR}_{ex}$ is not a linear space. Shary defines then an *immersion* map that identifies an interval vector of $\mathbb{IR}_{ex}$ with a real vector of $\mathbb{R}^{2n}$

$$\sigma(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \triangleq (\underline{x}_1, \dots, \underline{x}_n, \overline{x}_1, \dots, \overline{x}_n)^\mathsf{T}. \quad (14)$$

Thus, the original problem of finding the zeroes of the function $\boldsymbol{\psi}(\boldsymbol{x}) = \boldsymbol{C}\boldsymbol{x} \ominus \boldsymbol{x} + \boldsymbol{b}$, is transformed into the equation $\Psi(x) = 0$ where $\Psi = \sigma \circ \psi \circ \sigma^{-1} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$. The search of the zeros of $\Psi(x) = 0$ is done using a variation of Newton's method, known as *sub-differential Newton's method.* All the details are in [18]. The convergence of the method and its properties are further studied in [12].

Solving overdetermined systems. The trick used to transform an overdetermined interval system into a square one is due to Rump [16]. It is well known that the least squares solution of the overdetermined non-interval linear system $Ax = b$ is the solution of the square linear system $A^\mathsf{T}Ax = A^\mathsf{T}b$, when A has full rank. In the case of interval matrices, however, $A^\mathsf{T}A$ is in general ill-conditioned, and Rump prescribes to use the following equivalent linear systems instead:

$$\begin{pmatrix} \boldsymbol{A} & I \\ 0 & \boldsymbol{A}^\mathsf{T} \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b} \\ 0 \end{pmatrix}. \quad (15)$$

4 Results

In this section we report experimental results obtained by applying interval analysis techniques to the calibration and triangulation problems. The performances of Shary's and INTLAB's methods have been compared on both synthetic and real data.

	Err. 0.1	Err. 0.5	Err. 1.0	Err. 1.5
INTLAB	6.89	47.2	104	164
SHARY	1.16	4.69	9.83	14.6

Table 1. Synthetic calibration experiment: average box width of the back-projected 3-D points vs. enclosure width of image points.

Calibration. Synthetic data consisted of the coordinates of 128 reference points of calibration jig (depicted in Fig. 1). On each of the two orthogonal faces there are 64 points organised in a regular grid with a spacing of 2 cm. Views were generated by placing cameras at random positions, at a mean distance from the centre of 1 m with a standard deviation of 0.25 m. The orientations of the cameras were chosen randomly with the constraint that the optical axis should point towards the origin. The intrinsic parameters were given the values $\alpha_u = \alpha_v = 1600$, $u_0 = 320$, $v_0 = 240$ e $\gamma = 0$. Image points were (roughly) contained in a 640 × 480 pixels image. It was assumed that the position of the points in the image were affected by errors bounded in intervals of width 0.1, 0.5, 1 and 1.5 pixel.

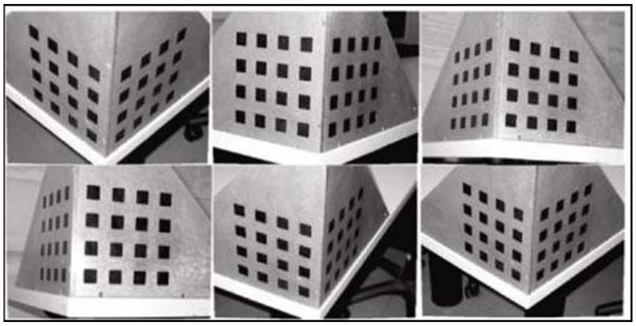

Figure 1. Images of the calibration jig used in the real calibration experiment.

In each trial, camera matrices $\boldsymbol{P}$ were computed by solving the linear system (3) with the two methods introduced in Sec. 3.1. The projections of 3-D reference points onto the image plane with these camera matrices are not points but *rectangles* since $\boldsymbol{P}$ is an interval matrix. Fig. 1 reports the square root of the average area of the rectangles over the image and over 100 independent trials versus the width of the intervals representing the position of the points in the image plane. The value on the y-axis is a sort of average box width of the intervals obtained by back-projecting the 3-D points, and can be taken as a measure of the error affecting the camera matrix $\boldsymbol{P}$.

Experiments were also performed on the real calibration jig shown in Fig. 1. Six 640 × 480 images have been taken from a distance of approximately 1 m with a digital camera. Image points have been bounded by square intervals 1 pixel wide, and interval calibration was performed on each image with both methods. Figure 2 compares a detail of the back-projected 3-D points for Shary's and INTLAB's methods. The average (over the image and over 6 trials) box width of the back-projected 3-D points was 5.9 pixel for Shary and 27.6 pixel for INTLAB.

Triangulation. Interval triangulation was tested in the same conditions used in calibration. In each trial, two random views were selected and both were calibrated assuming a 1 pixel wide enclosure for the position of the points in the image. The resulting interval camera matrices and the corresponding image points were used to perform interval triangulation, solving Eq. (4) with both methods. As in the previous case, the position of the points in the image was enclosed by intervals of width 0.5, 1.0, 1.5, and 2.0 pixel. The output was a set of cuboids that contain the true 3-D points and bounds the error.

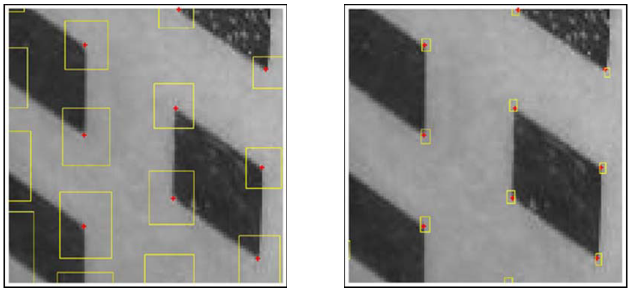

Figure 2. Detail of the back-projected 3-D points for INTLAB's (left) and Shary's (right) methods. The red dots are the projection of 3-D points with the non-interval camera matrix.

	Err. 0.5	Err. 1.0	Err. 1.5	Err. 2
INTLAB	1.56	2.32	2.71	3.00
SHARY	1.37	1.48	1.60	2.10

Table 2. Synthetic triangulation experiment: average side length of the reconstructed cuboids vs. enclosure width of image points.

Table 2 reports the cubic root of the average (over the image and over 100 independent trials) volume of the reconstructed cuboids versus the width of the intervals representing the position of the points in the image.

Interval reconstruction was tested on the same real images used for calibration (Fig. 1). Starting from the interval camera matrices previously obtained, triangulation was applied to each pair of images, assuming an error of 1 pixel in the localization of points. Figure 3 shows an instance of interval reconstructions. The true points (whose position is known) are contained in the cuboids. The average side length of the cuboids enclosing the true solution was 0.3 cm for Shary's and 1 cm for INTLAB's technique. The former is a reasonable bound, given that the grid points have a 2 cm spacing.

Although in this case INTLAB's accuracy is comparable to Shary's, the latter method produced significantly sharper inclusions in all the other synthetic and real experiments, confirming the outcome of the calibration experiments.

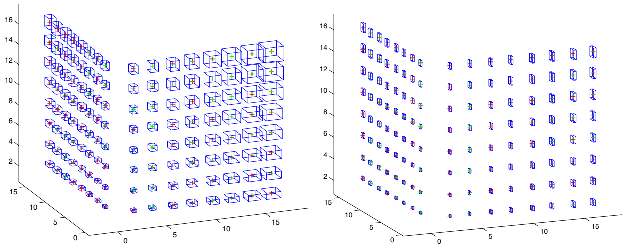

Figure 3. Example of a reconstruction of the calibration grid obtained with INTLAB's (left) and Shary's technique (right). The true position of the 3-D points is also shown.

5 Conclusion

In this paper we showed how to obtain realistic bounds on the reconstruction error using numerical techniques based on Interval Arithmetic. This branch of numerical analysis has been criticized, in the past, for providing too pessimistic bounds. Indeed, the straightforward application of interval arithmetic rules leads to an excessive growth of intervals' widths. However, as we show in this paper, a careful choice of suitable numerical techniques allows to obtain meaningful and realistic bounds on the output error even in presence of a significant error on the input data.

The advantages of Interval Arithmetic over traditional techniques are that i) no assumptions are made about the underlying error distribution, ii) the results *and* the error bounds are obtained simultaneously as the output of the same process and iii) error delimitation are guaranteed with mathematical certainty.

We concentrated on linear calibration and triangulation techniques and selected two state-of-the art methods for the solution of linear system of equations, namely INTLAB's and Shary's methods. Empirical comparison shown that the latter provides sharper error bounds in this application.

Results encourage us to pursue further the application of Interval Arithmetic to Computer Vision problems.

Acknowledgments. Thanks to S.P. Shary for providing the implementation of his linear systems solver. The use of the INTLAB toolbox for MATLAB by S.M. Rump is here acknowledged.

References

[1] *Workshop on Performance Characteristics of Vision Algorithms*, Cambridge, UK, 1996.

[2] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA, 1993.

[3] G. I. Hargreaves. Interval analysis in MATLAB. Technical report, Manchester Centre for Computational Mathematics, 2002.

[4] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.

[5] J.Rohn. Np-hardness results for linear algebraic problems with interval data. Preprint, Faculty of Math. and Physics, Charles University, Czech Republic, 1994.

[6] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993.

[7] F. Kaucher. Interval analysis in the extended interval space $\mathbb{IR}_{ex}$. *Computing Suppl.*, 2:33–49, 1980.

[8] R. Krawczyk. Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken. *Computing*, 4:187–201, 1969.

[9] R. Marik, J. Kittler, and M. Petrou. Error sensitivity assessment ov vision algorithm based on direct error-propagation. In *Workshop on Performance Characteristics of Vision Algorithms*, Cambridge, UK, 1996.

[10] R. E. Moore. *Interval Analysis*. Prentice-Hall, 1966.

[11] A. Neumaier. A simple derivation of the hansen-bliek-rohnning-kearfott enclosure for linear interval equations. *Reliable Computing*, 5:131–136, 1999. Erratum, Reliable Computing 6 (2000), 227.

[12] A. Neumaier. On shary's algebraic approach for linear interval equations. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1156–1162, 2000.

[13] A. Neumaier. *Introduction to Numerical Analysis*. Cambridge University Press, Cambridge, UK, 2001.

[14] R.B.Kearfott. *Rigorous Global Search: Continuos Problems*. Kluwer, 1996.

[15] R.M.Haralick. Propagating covariance in computer vision. In *Workshop on Performance Characteristics of Vision Algorithms*, pages 1–12, Cambridge, UK, 1996.

[16] S. Rump. Solving algebraic systems with high accuracy. In U. Kulisch and W. Miranker, editors, *A New Approach to Scientific Computation*, pages 51–120. Academic Press, 1983.

[17] S.M.Rump. INTLAB-INTerval LABoratory. In T.Csendes, editor, *Development in Reliable Computing*. Kluwer, 1999.

[18] S.P.Shary. Algebraic approach in the "outer problem" for interval linear equations. *Reliable Computing*, 3(2):103–135, 1997.