Computing Rigorous Bounds to the Accuracy of Calibrated Stereo Reconstruction

A. Fusiello, M. Farenzena, A. Busti,

Dipartimento di Informatica

Università degli Studi di Verona

I-37134 Verona, Italy

{fusiello,farenzen}@sci.univr.it

A. Benedetti

Division of Engineering and Applied Science

California Institute of Technology

Pasadena, CA 91125

arrigo@vision.caltech.edu

Currently affiliated with KLA-Tencor, One Technology Dr., MS 5-2017, Milpitas, CA 95035 (USA). Email: Arrigo.Benedetti@kla-tencor.com.

Abstract

We deal with the problem of computing rigorous bounds to the position of 3-D points obtained by stereo triangulation when both the camera matrix and the coordinates of image points are affected by measurement errors. By "rigorous bounds" we mean that the true unknown 3-D points are *guaranteed* to lie within the intervals computed by our method with mathematical certainty. 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 compute narrow intervals for the entries of the camera matrix by using numerical methods based on Interval Analysis. Finally, we apply triangulation and obtain cuboids that bound point coordinates. We employed two state-of-the-art methods for the solution of linear systems of interval equations, namely Rump's and Shary's methods. Our conclusion is that a careful selection of numerical techniques allows to use Interval Analysis as a tool for obtaining realistic bounds on the output error even in presence of significant errors in the input data.

I. Introduction

Being an empirical science, Computer Vision has to deal with measurements affected by errors. The problem of propagating errors from input data to results in Computer Vision is not new; in fact, it has been addressed in [1], [2] and in a landmark workshop [3]. Consolidated techniques are based on statistical analysis of error propagation: given an input error distribution, a closed form expression for the output error distribution is derived.

In this work we take a different approach, based on *Interval Analysis*. Data are represented by intervals containing unknown quantities, and error bounds are modeled by intervals. Arithmetic operations are then performed on these intervals, with the guarantee that the resulting interval contains the exact result.

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. We assume that in this process both camera matrices and the corresponding points in the image are affected by bounded errors. We also model error propagation in the calibration process by computing bounds to the entries of the camera matrix. We concentrate in particular on calibration and triangulation methods based on the solution of (over-determined) linear systems of equations.

Error propagation in the solution of linear systems have been widely studied (see [4]). Interval Analysis has the appeal of computing error bounds simultaneously with the solution, instead of estimating the error from perturbation analysis. Moreover, it does not make any assumption on the underlying statistical error distribution, apart from being bounded.

A. Related work

For the most part, error analysis in Computer Vision is based on statistical methods. Assuming that the input data has a certain error distribution, one desires to determine the output error distribution. In principle this should be done by propagating the input distribution through the steps 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. The theory of covariance propagation in Computer Vision have been developed in [1], [5], where 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. In [6] the authors criticize the use of covariance and propose an information theoretic criterion to estimate the probability distribution function of the parameter being estimated, rather than concentrating on certain moments only. A framework for performing statistical analysis of geometric algorithms has been introduced in [2].

Marik et al. [7] first suggested to use interval analysis for the study of error propagation in Computer Vision. These authors propose two error models, one based on covariance propagation, and another (called min/max value propagation) based on Interval Arithmetic. As the bounds obtained with direct application of the rules of Interval Arithmetic 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.” [7] As we shall see, however, a careful selection of suitable techniques allowed us to use Interval Analysis [8] as a tool for obtaining realistic bounds on the output error even in presence of a significant error on the input data.

In the specific case of calibrated stereo-based reconstruction, several authors have found expressions for the error probability distribution (under various assumptions) [9], [2], [1], [10], [11], while others performed empirical evaluations [12].In [13], [14] *confidence intervals* for 3-D reconstruction from stereo are derived. These are random intervals which capture the true value of the parameter being estimated with a given probability. In Interval Analysis, instead, intervals are guaranteed to contain the true values of the unknowns.

II. PROBLEM FORMULATION

This section briefly recalls the mathematical developments of stereo reconstruction that are relevant to our work. For more details see [1]. We concentrate on the linear formulation of calibration and reconstruction. It is well-known that optimal solutions to this problem, in the Maximum Likelihood sense, are obtained by nonlinear optimization of a geometrically-valid residual, whereas linear techniques only minimize an algebraic residual [15]. In our experience with Interval Analysis, however, the complexity of non-linear optimization causes excessive growth of intervals, thus giving rise to high code run times.

Let $w = (x, y, z)^\mathsf{T}$ 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 $m = (u, v)^\mathsf{T}$ of the projected image point are given by the *perspective projection* equation:

$$\begin{cases} u = \dfrac{p_1^\mathsf{T} w + p_{1,4}}{p_3^\mathsf{T} w + p_{3,4}} \\ v = \dfrac{p_2^\mathsf{T} w + p_{2,4}}{p_3^\mathsf{T} w + p_{3,4}}. \end{cases} \tag{1}$$

where $p_i \in \mathbb{R}^3$ for $i = 1 \dots 3$. The 3×4 full-rank matrix

$$P = \begin{pmatrix} p_1^\mathsf{T} & p_{1,4} \\ p_2^\mathsf{T} & p_{2,4} \\ p_3^\mathsf{T} & p_{3,4} \end{pmatrix} \tag{2}$$

models the pinhole camera, and it is called *camera matrix*.

A. Calibration

Calibration consists in estimating as accurately as possible the elements of the camera matrix P. If enough correspondences between world points and image points are available, it is possible to solve the perspective projection equation for the unknown entries of P.

Given N reference points, not coplanar, each correspondence between an image point $m_i = (u_i, v_i)^\mathsf{T}$, and a reference point w_i gives a pair of equations, derived from (1):

$$\begin{cases} (p_1 - u_i p_3)^\mathsf{T} w_i + p_{1,4} - u_i p_{3,4} = 0 \\ (p_2 - v_i p_3)^\mathsf{T} w_i + p_{2,4} - v_i p_{3,4} = 0 \end{cases} \tag{3}$$

The unknown camera matrix has 12 elements, however, being defined up to a scale factor, it has only 11 free parameters. We can choose $p_{34} = 1$, thus reducing the number of unknowns to 11, obtaining the following two equations:

$$\begin{pmatrix} w_i^\mathsf{T} & 1 & 0 & 0 & -u_i w_i^\mathsf{T} \\ 0 & 0 & w_i^\mathsf{T} & 1 & -v_i w_i^\mathsf{T} \end{pmatrix} \begin{pmatrix} p_1 \\ p_{1,4} \\ p_2 \\ p_{2,4} \\ p_3 \end{pmatrix} = \begin{pmatrix} u_i \\ v_i \end{pmatrix}. \tag{4}$$

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 pixel coordinate measurements m_i are affected by bounded errors (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 (4).

B. Triangulation

Given the matrices of the two cameras and the coordinates of the projections on the image planes of a 3-D point, its coordinates can be recovered by a simple linear algorithm. Geometrically, the process consists in intersecting the optical rays of the two image points, and this is the reason why it is known as *triangulation*.

Let P and P' be the two camera matrices, let w be the unknown coordinates of the 3-D point, and let $m = (u, v)^\mathsf{T}$ and $m' = (u', v')^\mathsf{T}$ be the image coordinates of a conjugate pair. From (1) we obtain a linear system of four equations in the unknown 3-vector w:

$$\begin{pmatrix} (p_1 - u p_3)^\mathsf{T} \\ (p_2 - v p_3)^\mathsf{T} \\ (p'_1 - u' p'_3)^\mathsf{T} \\ (p'_2 - v' p'_3)^\mathsf{T} \end{pmatrix} w = \begin{pmatrix} -p_{1,4} + u p_{3,4} \\ -p_{2,4} + v p_{3,4} \\ -p'_{1,4} + u' p'_{3,4} \\ -p'_{2,4} + v' p'_{3,4} \end{pmatrix}. \tag{5}$$

Again, a bounded error affecting P, m and m' translates into bounds to the entries of the coefficient matrix and the right hand side vector of (5).

In the next section we see how arithmetic operations can be defined on intervals, allowing to find rigorous bounds to the solution of linear systems of equations.

III. Interval Analysis

Interval Arithmetic is an arithmetic defined on *intervals*, rather than on real numbers. In the beginning, Interval Arithmetic was mainly employed for bounding the measurement errors of physical quantities for which no statistical distribution was known. Later on it was leveraged to a broad new field of applied mathematics, aptly named Interval Analysis [8], where rigorous proofs are the consequence of numerical computations.

In this paper we follow the notation used in [16], where intervals are denoted by boldface. Scalar quantities and vectors are denoted by lower case letters and matrices are denoted by upper case, brackets '$[\cdot]$' will delimit intervals, while parentheses '$(\cdot)$' will delimit vectors and matrices. Underscores and overscores will represent respectively lower and upper bounds of intervals. An interval $\boldsymbol{x}$ is called *degenerate* when $\underline{\boldsymbol{x}} = \overline{\boldsymbol{x}} = x$. $\mathbb{IR}$ and $\mathbb{IR}^n$ stand respectively for the set of real intervals and the set of interval vectors of dimension n. The midpoint of an interval $\boldsymbol{x}$ is denoted by $\mathrm{m}(\boldsymbol{x})$. The *width* of $\boldsymbol{x} \in \mathbb{IR}^n$ is defined as $\mathrm{w}(\boldsymbol{x}) = \max\{(\overline{\boldsymbol{x}}_i - \underline{\boldsymbol{x}}_i),\ i = 1, \ldots, n\}$. 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 $\boldsymbol{x} = [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$ and $\boldsymbol{y} = [\underline{\boldsymbol{y}}, \overline{\boldsymbol{y}}]$, a binary operation in the *ideal Interval Arithmetic* between $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as:

$$\boldsymbol{x} \,\mathrm{op}\, \boldsymbol{y} \triangleq \{x \,\mathrm{op}\, y \mid x \in \boldsymbol{y} \text{ and } y \in \boldsymbol{y}\}, \quad \text{for} \quad \mathrm{op} \in \{+, -, \times, \div\}. \tag{6}$$

Thus, the ranges of the four elementary interval operations are exactly the ranges of the corresponding real operations. The operational definitions for the four elementary Interval Arithmetic operations are

$$\boldsymbol{x} + \boldsymbol{y} \triangleq [\underline{\boldsymbol{x}} + \underline{\boldsymbol{y}}, \overline{\boldsymbol{x}} + \overline{\boldsymbol{y}}], \tag{7}$$

$$\boldsymbol{x} - \boldsymbol{y} \triangleq [\underline{\boldsymbol{x}} - \overline{\boldsymbol{y}}, \overline{\boldsymbol{x}} - \underline{\boldsymbol{y}}], \tag{8}$$

$$\boldsymbol{x} \times \boldsymbol{y} \triangleq [\min\{\underline{\boldsymbol{x}}\,\underline{\boldsymbol{y}}, \underline{\boldsymbol{x}}\,\overline{\boldsymbol{y}}, \overline{\boldsymbol{x}}\,\underline{\boldsymbol{y}}, \overline{\boldsymbol{x}}\,\overline{\boldsymbol{y}}\}, \max\{\underline{\boldsymbol{x}}\,\underline{\boldsymbol{y}}, \underline{\boldsymbol{x}}\,\overline{\boldsymbol{y}}, \overline{\boldsymbol{x}}\,\underline{\boldsymbol{y}}, \overline{\boldsymbol{x}}\,\overline{\boldsymbol{y}}\}], \tag{9}$$

$$\frac{1}{\boldsymbol{x}} \triangleq \begin{cases} [1/\overline{\boldsymbol{x}}, 1/\underline{\boldsymbol{x}}] & \text{if } \underline{\boldsymbol{x}} > 0 \\ [1/\underline{\boldsymbol{x}}, 1/\overline{\boldsymbol{x}}] & \text{if } \overline{\boldsymbol{x}} < 0 \end{cases} \quad (0 \notin [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]), \tag{10}$$

$$\boldsymbol{x} \div \boldsymbol{y} \triangleq \boldsymbol{x} \times 1/\boldsymbol{y}. \tag{11}$$

The above definitions imply the ability to perform the four elementary operations with arbitrary precision. When implemented in a digital computer, however, truncation errors occur, that may cause the result not to contain the result that would be obtained with ideal Interval Arithmetic. In order to avoid this effect, the result corresponding to the lower endpoint of the interval must be rounded down to the nearest machine number less than the mathematically correct result, and the upper endpoint must be rounded up to the nearest machine number greater than the mathematically correct result. This mode of operation, called *direct rounding*, is available on any machine supporting the IEEE floating point standard.

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 $x_1, x_2, \ldots, x_n$:

$$\boldsymbol{f}(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n). \tag{12}$$

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, $\boldsymbol{f}_1(\boldsymbol{x}) = \boldsymbol{x}^2 - \boldsymbol{x}$, $\boldsymbol{f}_2(\boldsymbol{x}) = \boldsymbol{x}(\boldsymbol{x} - 1)$, and $\boldsymbol{f}_3(\boldsymbol{x}) = (\boldsymbol{x} - 1/2)^2 - 1/4$ are all different interval extensions of $f(x) = x^2 - x = x(x-1) = (x - 1/2)^2 - 1/4$.

The following theorem is known as the fundamental theorem of Interval Analysis [8]:

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

$$\boldsymbol{f}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \subset \boldsymbol{f}(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n). \tag{13}$$

¿From this theorem it follows immediately that

$$\boldsymbol{f}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \;\supset\; \boldsymbol{f}^u(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n). \tag{14}$$

In the previous example, by setting $\boldsymbol{x} = [0, 1]$ we have

$$\boldsymbol{f}_2(\boldsymbol{x}) = [0,1]\,([0,1] - 1) = [0,1]\,[-1,0] = [-1,0]\,, \tag{15}$$

which necessarily includes the exact range $\boldsymbol{f}^u([0,1]) = [-1/4, 0]$.

A. Interval linear systems

Interval linear systems are useful to calculate rigorous bounds to the solutions of linear systems of equations. They have the form

$$\boldsymbol{A}x = \boldsymbol{b}, \tag{16}$$

where $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{IR}^{n}$. The *solution set* is defined as

$$\Sigma(\boldsymbol{A},\boldsymbol{b}) = \{x : \exists A\in\boldsymbol{A} \ \text{ and } \ \exists b\in\boldsymbol{b} \ \text{ t.c. } Ax=b\}. \tag{17}$$

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 [4]. Thus, we must accept to compute only 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 [17] 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 computational cost of $O(n^3)$.

The first practitioners of Interval Analysis realized very soon that the naive application of standard numerical algorithms like Gaussian elimination to interval data gives very poor results. This problem [4] has received a lot of attention during the last three decades, and today there are many algorithms specific to the solution of interval linear systems. Among these we have considered the method implemented by the `verifylss` function in the INTLAB toolbox for MATLAB [18], [19] by Rump, and the method introduced by Shary [20].

1) INTLAB method: The following description is based on [19]. The first stage of the algorithm implemented by the `verifylss` function is an iterative method introduced by Rump [21], based on the well-known Krawcyzk operator [22] (see also [4]).

Assuming that there is an interval vector $\boldsymbol{x}^i$ such that $\square\Sigma(\boldsymbol{A},\boldsymbol{b}) \subseteq \boldsymbol{x}^i$, then

$$\forall \tilde{A} \in \boldsymbol{A}, \tilde{b} \in \boldsymbol{b} : \tilde{A}^{-1}\tilde{b} = C\tilde{b} + (I - C\tilde{A})\tilde{A}^{-1}\tilde{b} \in C\boldsymbol{b} + (I - C\boldsymbol{A})\boldsymbol{x}^i. \tag{18}$$

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

$$\square\Sigma(\boldsymbol{A},\boldsymbol{b}) \subseteq \boldsymbol{x}^i \Rightarrow \square\Sigma(\boldsymbol{A},\boldsymbol{b}) \subseteq (C\boldsymbol{b} + (I - C\boldsymbol{A})\boldsymbol{x}^i) \cap \boldsymbol{x}^i, \tag{19}$$

and this gives the *Krawcyzk iteration*:

$$\boldsymbol{x}^{i+1} = (C\boldsymbol{b} + (I - C\boldsymbol{A})\boldsymbol{x}^i) \cap \boldsymbol{x}^i. \tag{20}$$

Rump's method, instead, proceeds by enclosing the error with respect to an approximate solution $\tilde{x} = C\,\mathrm{m}(\boldsymbol{b})$. By applying (19) to an enclosure $\boldsymbol{d}_i$ of $\square\Sigma(\boldsymbol{A}, \boldsymbol{b}-\boldsymbol{A}\tilde{x})$, gives the following iteration

$$\boldsymbol{d}^{i+1} = (C(\boldsymbol{b} - \boldsymbol{A}\tilde{x}) + (I - C\boldsymbol{A})\boldsymbol{d}^i) \cap \boldsymbol{d}^i. \tag{21}$$

The solution to the original problem is $\tilde{x} + \boldsymbol{d}^i$.

Starting from the Brower [23] fixed point theorem, the following implication holds:

$$C\boldsymbol{b} + (I - C\boldsymbol{A})\boldsymbol{x} \subseteq \mathrm{int}(\boldsymbol{x}) \Rightarrow \square\Sigma(\boldsymbol{A}, \boldsymbol{b}) \subseteq \boldsymbol{x} \tag{22}$$

, which is used to check convergence.

If there is no success after seven iterations, the algorithm described in [24] is applied.

2) Shary's method: Shary introduced an *algebraic approach* for enclosing the solution of a linear system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with interval coefficient matrix $\boldsymbol{A}$ and interval right hand side vector $\boldsymbol{b}$. 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* [25] $\mathbb{IR}_{ex}$, which is obtained by adding *improper* intervals $[\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$, $\underline{\boldsymbol{x}} > \overline{\boldsymbol{x}}$ to the set $\mathbb{IR} = \{[\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}] \mid \underline{\boldsymbol{x}}, \overline{\boldsymbol{x}} \in \mathbb{R},\ \underline{\boldsymbol{x}} \leq \overline{\boldsymbol{x}}\}$ of *proper* intervals. Shary's method is based on the fixed point equation

$$\boldsymbol{x} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{b}, \tag{23}$$

which resolves into finding an algebraic solution to the interval equation:

$$\boldsymbol{C}\boldsymbol{x} \ominus \boldsymbol{x} + \boldsymbol{b} = 0 \tag{24}$$

where $\boldsymbol{C} = I - \boldsymbol{A}$ and '$\ominus$' denote the *inner subtraction*, defined by $\boldsymbol{x} \ominus \boldsymbol{y} \triangleq [\underline{\boldsymbol{x}} - \underline{\boldsymbol{y}}, \overline{\boldsymbol{x}} - \overline{\boldsymbol{y}}]$. 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 $\mathbb{R}^{2n}$

$$\sigma(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \triangleq (\underline{\boldsymbol{x}}_1, \underline{\boldsymbol{x}}_2, \ldots, \underline{\boldsymbol{x}}_n, \overline{\boldsymbol{x}}_1, \overline{\boldsymbol{x}}_2, \ldots, \overline{\boldsymbol{x}}_n)^{\mathsf{T}}. \tag{25}$$

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

$$\Psi(x) = 0, \tag{26}$$

where $\Psi = \sigma \circ \psi \circ \sigma^{-1} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$, i.e.,

$$\Psi(x) = \sigma(\boldsymbol{C}\sigma^{-1}(x) \ominus \sigma^{-1}(x) + \boldsymbol{b}) = \sigma(\boldsymbol{C}\sigma^{-1}(x)) - x + \sigma(\boldsymbol{b}) \tag{27}$$

The existence of a solution to the fixed point equation (23) is related to the spectral radius of $\boldsymbol{C}$. In order to extend the applicability domain and to get a sharper enclosure, a suitable pre-conditioning step must be performed. Shary employs the fixed point equation (23), where

$$\boldsymbol{C} = G^{-1}(G - \boldsymbol{A}) \quad \text{and} \quad \boldsymbol{b} = G^{-1}\boldsymbol{b} \tag{28}$$

with $G \triangleq \mathrm{diag}(\mathrm{dev}(\boldsymbol{A}_{ii}))$ and

$$\mathrm{dev}\,\boldsymbol{x} \triangleq \begin{cases} \underline{\boldsymbol{x}} & \text{if } |\underline{\boldsymbol{x}}| \geq |\overline{\boldsymbol{x}}| \\ \overline{\boldsymbol{x}} & \text{otherwise} \end{cases} . \tag{29}$$

The search of the zeros of (26) is done using a variation of Newton's method, known as *sub-differential Newton's method*. All the details are in [20]. The convergence of the method and its properties are further studied in [26].

3) Overdetermined Interval Linear Systems: There are two ways of casting an overdetermined linear system of equations $Ax = b$ into a square one. The most common is the *normal equation*: $A^\mathsf{T} Ax = A^\mathsf{T} b$ which has the advantage of keeping the size small, but it worsens the conditioning of the system. The other way is to use the *augmented system*:

$$\begin{pmatrix} A & I \\ 0 & A^\mathsf{T} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix} \tag{30}$$

which keeps the conditioning unchanged at the price of doubling the size of the system. This solution is recommended by Rump [21] for the solution of overdetermined interval linear systems.

IV. Experimental Results

In this section we report some 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.

A. Calibration

Synthetic data consist of the 128 reference points of a calibration jig depicted in Fig. 2. 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 points 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 $\times$ 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.

In each trial, interval camera matrices $\boldsymbol{P}$ were computed by solving the linear system (4) with the two methods introduced in Sec. III-A. The projections of 3-D reference points onto the image plane with these camera matrices are not points but *rectangles*, as $\boldsymbol{P}$ 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 can be taken as a measure of the "effective width" of $\boldsymbol{P}$.

Experiments were also performed on the real calibration jig shown in Fig. 2. Six 640×480 images have been taken from a distance of approximately 1 m with a digital camera. Image points have been represented by square intervals 1 pixel wide, and interval calibration was performed on each image with both methods. Figure 3 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.

In both synthetic and real experiments, Shary's method performed significantly better.

B. 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 (5) with both methods. As in the previous case, the position of the points in the image was enclosed by intervals of width 0.1, 0.5, 1 and 1.5 pixel. The output (Fig. 4) was a set of cuboids that contains the true 3-D points and bounds the error. Fig. 5 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.

We performed a synthetic experiment where the distance separating the two cameras (*baseline*) was progressively decreased. If we assume a finite size pixel, the result of triangulation is a diamond-shaped solid, whose volume approaches infinity as the baseline vanishes. This behaviour was correctly reproduced by interval reconstruction, as the average volume of the interval reconstruction consistently increased.

Interval reconstruction was also tested on the same real images used for calibration (Fig. 2). Starting from the interval camera matrices previously obtained, triangulation was applied to each

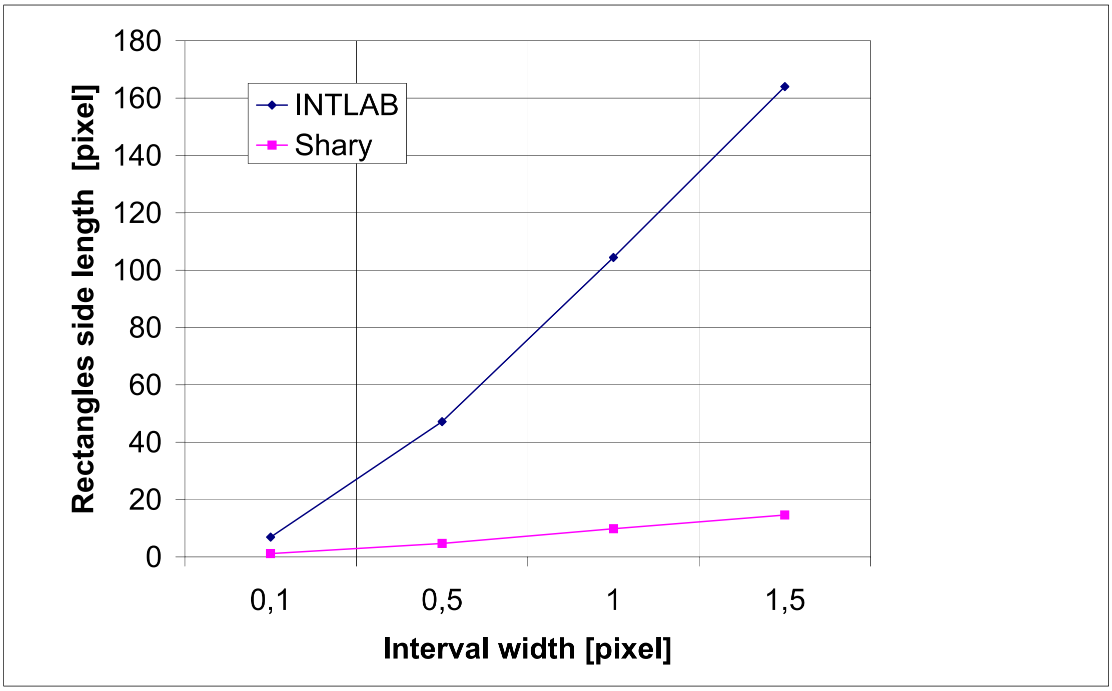

Fig. 1. Synthetic calibration experiment: average box width of the back-projected 3-D points vs. enclosure width of image points.

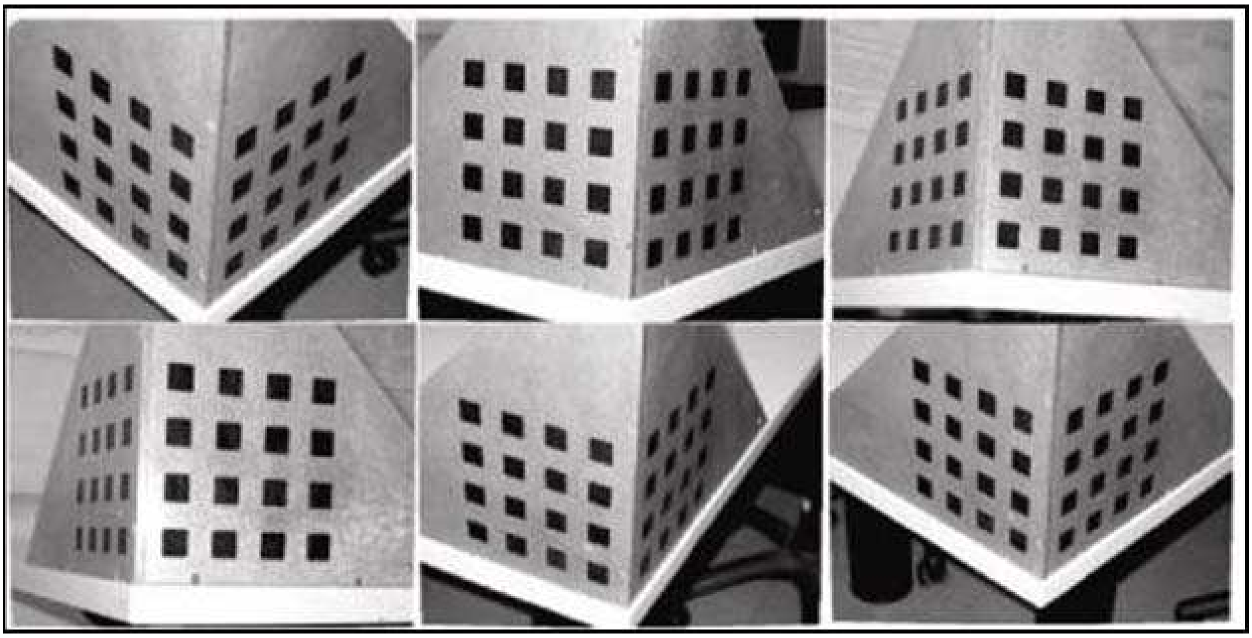

Fig. 2. Images of the calibration jig used in the real calibration experiment.

pair of images, assuming an error of 1 pixel in the localization of points.

Figure 6 shows an instance of interval reconstruction. 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.

In another experiment (Fig. 7) we applied the interval reconstruction technique to a view pair whose camera matrices had been obtained from *autocalibration* [27]. The procedure is much more complex than the calibration described here, but the results are again two interval camera matrices. Assuming that feature points in the images are contained in 2 pixel wide intervals, the average side length of the 3-D intervals enclosing the true solution was about 14 cm with INTLAB's, and 12 cm with Shary's method[1].

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

V. Conclusions

In this paper we showed how to obtain realistic bounds to the reconstruction error using numerical techniques based on Interval Analysis. This branch of numerical analysis has been strangely overlooked by the Computer Vision community, possibly because it 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 interval widths. However, as we show in this paper, a careful choice of numerical techniques allows to obtain meaningful and realistic error bounds. The advantages of Interval Analysis over traditional techniques are that i) no assumptions are made about the underlying error distribution – apart from being bounded, ii) the results *and* the error bounds are obtained simultaneously as the output of the same process and iii) error bounds 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 systems of interval equations, namely INTLAB's

[1]The global scale factor of the scene structure, which autocalibration cannot recover, was estimated by guessing the height of the columns of the *Tribuna*.

and Shary's methods. Empirical comparison shows that the latter provides sharper error bounds in this application. It is always recommendable, however, to take the intersection of the boxes yielded by the two methods in order to get the best out of both.

Results are encouraging: delimitations seem realistic, given the order of magnitude of the other parameters. We plan to pursue further the application of Interval Analysis 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] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA, 1993.

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

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

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

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

[6] A. K. Roy, A. Chowdhury, and R. Chellappa. An information theoretic criterion for evaluating the quality of 3D reconstructions from video from video. *IEEE Transactions on Image Processing*, 2004.

[7] 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.

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

[9] A. Verri and V. Torre. Absolute depth estimation in stereopsis. *Journal of the Optical Society of America A*, 3(3):297–299, March 1998.

[10] S. Das and N. Ahuja. Performance analysis of stereo, vergence and focus as depth cues for active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1213–1219, 1995.

[11] A.Harris and N.A.Thacker. Error modelling of stereo vision data. Tina Memo 1999-003, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, 1999.

[12] G.Florou and R.Mohr. What accuracy for 3D measurements with cameras? In *Proceedings of the International Conference on Pattern Recognition*, pages 354–358, 1996.

[13] R. Mandelbaum, G. Kamberova, and M. Mintz. Stereo depth estimation: A confidence interval approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 503–509, 1998.

[14] G. Kamberova and R. Bajcsy. Sensor errors and the uncertainties in stereo reconstruction. In K. W. Bowyer and P. J. Phillips, editors, *Proceedingsof the IEEE Workshop on Empirical Evaluation Techniques in Computer Vision*, 1998.

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

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

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

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

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

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

[21] S.M. 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.

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

[23] L.E.J.Brower. Uber abbildung von mannigfaltigkeiten. *Math.Ann.*, 1912.

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

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

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

[27] A. Benedetti, A. Busti, M. Farenzena, and A. Fusiello. Globally convergent autocalibration. In *Proceedings of the IEEE International Conference on Computer Vision*, 2003.

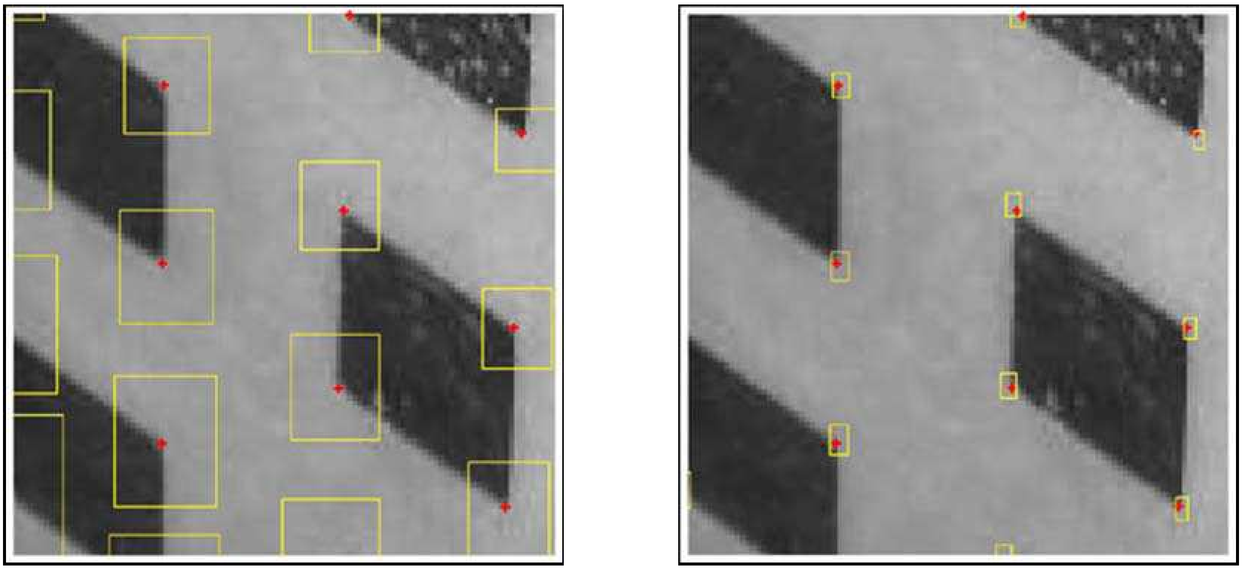

Fig. 3. Detail of the back-projected 3-D points for INTLAB's (left) and Shary's (right) methods. The red crosses are the back-projected 3-D points with the sharp (non-interval) camera matrix.

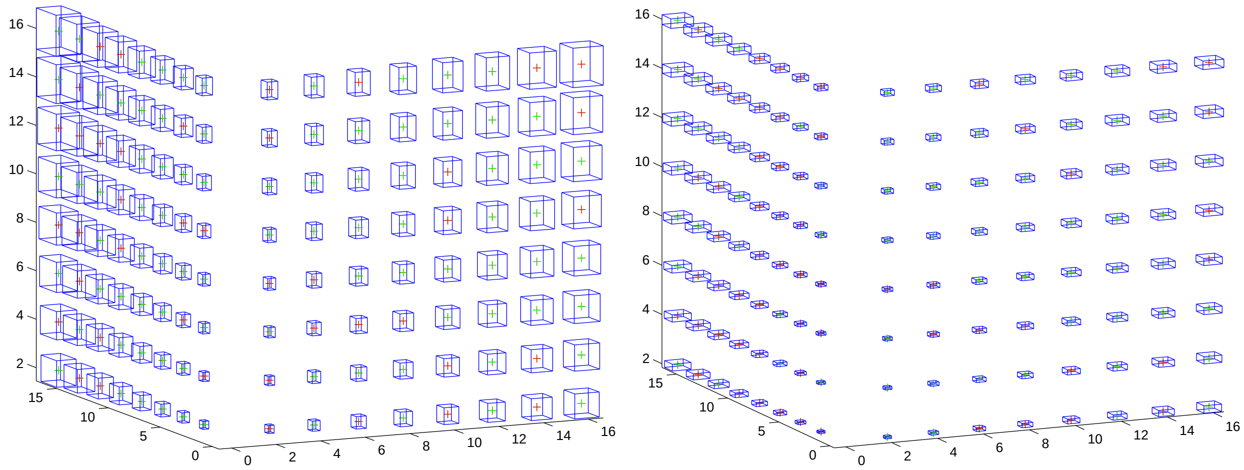

Fig. 4. Synthetic 3-D reconstruction with INTLAB's (left) and Shary's (right) methods.

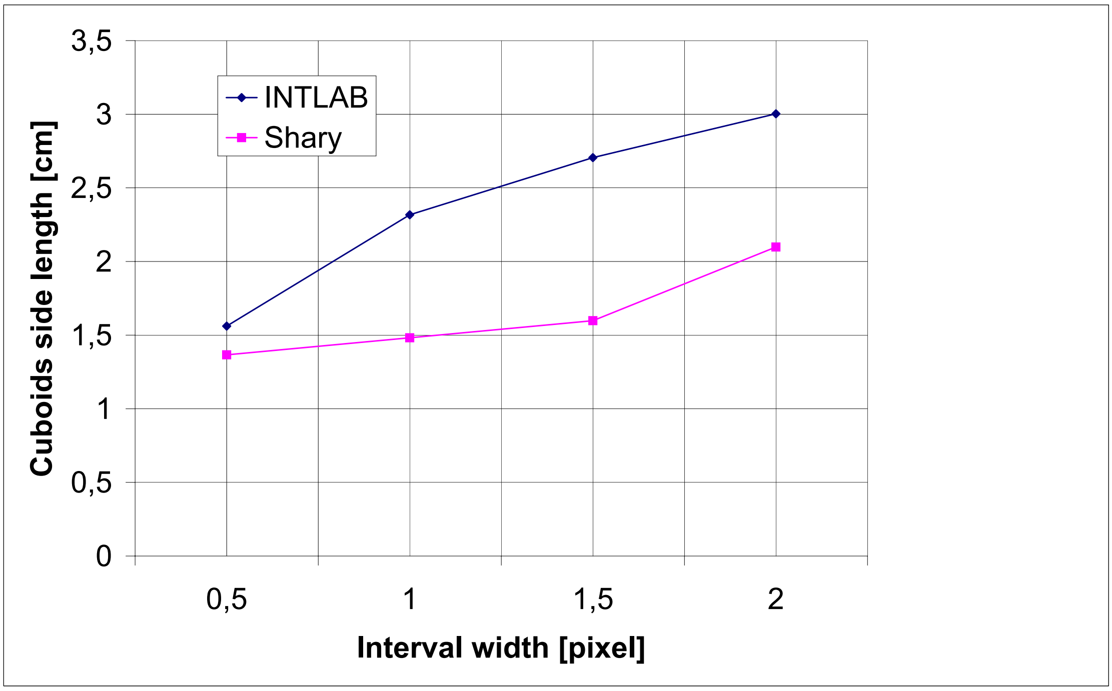

Fig. 5. Synthetic triangulation experiment: average side length of the reconstructed cuboids vs. enclosure width of image points.

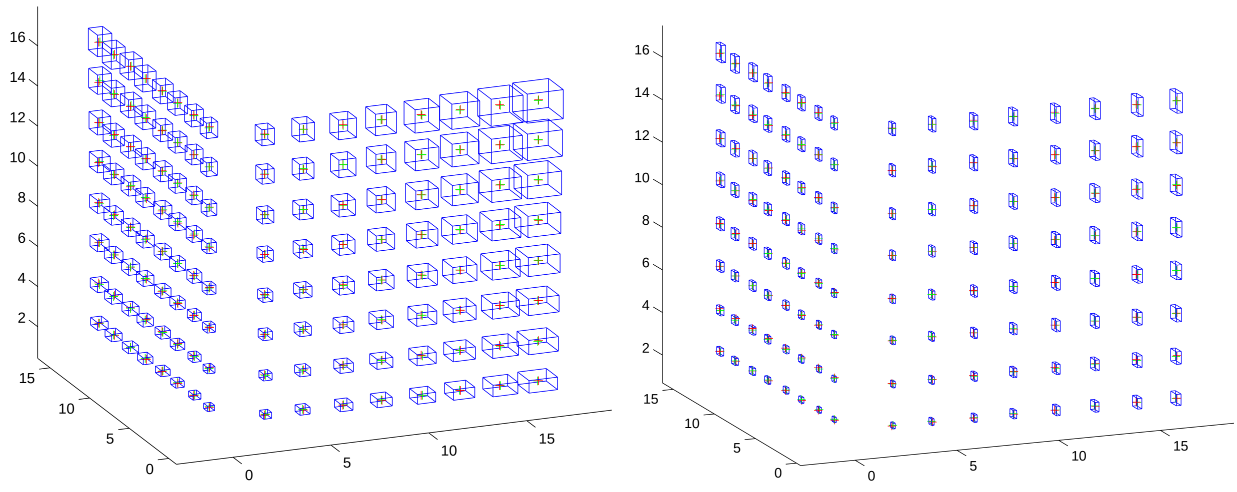

Fig. 6. Example of a reconstruction of the calibration grid obtained with INTLAB's (left) and Shary's technique (right). The true position of the 3-D points is marked with a cross.

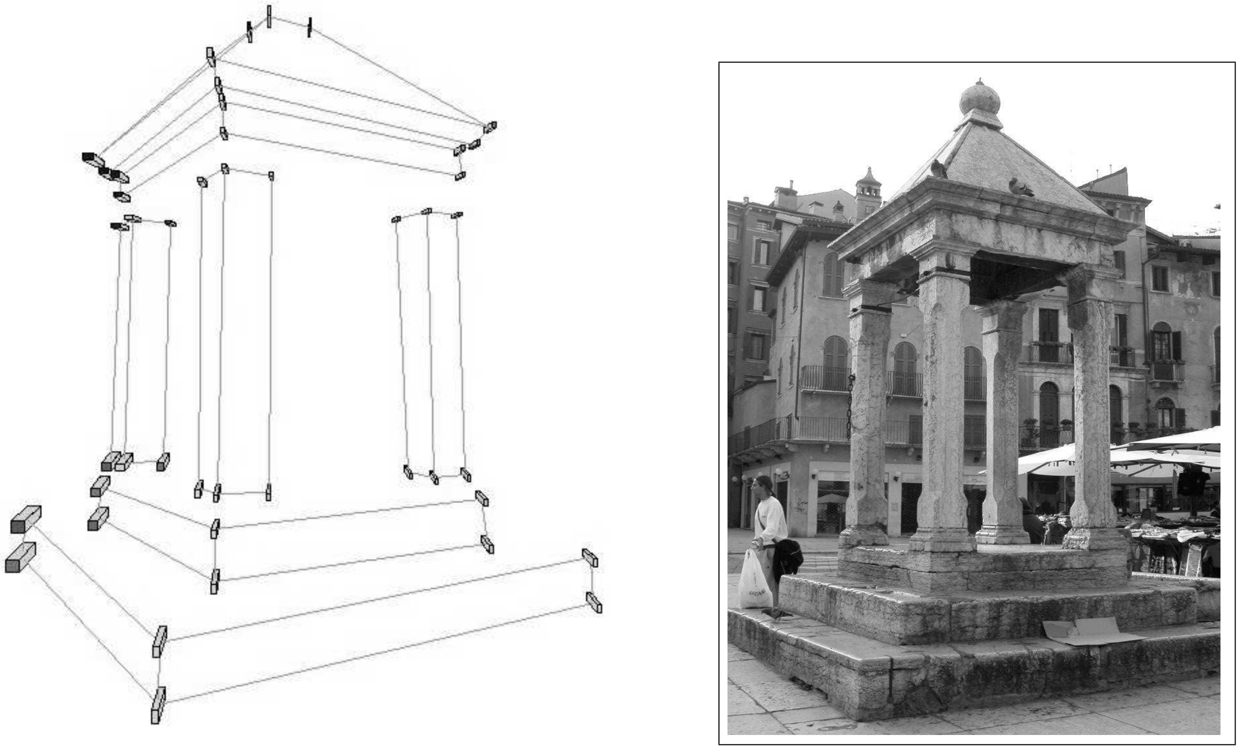

Fig. 7. Interval reconstruction of the *Tribuna* (left). The boxes are the intersection of the output of the two methods. Segments join the midpoints of the intervals. One frame of the sequence from which the reconstruction was obtained is shown to the right.