A Buyer's Guide to Forward Intersection for Binocular Robot Vision

Stephan Grünfelder

AUSTRIAN AEROSPACE GmbH, Department of Electrical Design
Stachegasse 16, A-1120 Wien, Austria
phone: +43 1 80199-0, fax: +43 1 80199-5577, e-mail: stephan.gruenfelder@space.at

Reinhard Krickl

Vienna University of Technology, Institute of Flexible Automation
Gusshausstrasse 27-29/361, A-1040 Wien, Austria
phone: +43 1 504 14 46-12, fax: +43 1 505 59 83, e-mail: rk@flexaut.tuwien.ac.at

Abstract

Binocular Robot Vision, the extraction of 3d-information from images from two distinct digital cameras, is subject to intensive research for space and terrestrial applications.

In space applications we mainly find tracking tasks (e.g. [1]) and explorative vision tasks (e.g. [2, 3]). Our paper focuses on reconstruction which is indispensable for the latter type and might be used for tracking as well. The paper presents the results of a basic research exercise.

Four different algorithms are presented that allow the reconstruction of an object point given its projections in two distinct digital cameras. The algorithms are compared with respect to absolute accuracy, relative precision, and computational requirements by means of simulations. From the simulation results we derive a rule of thumb that tells which algorithm to take for a given problem. Furthermore, the implementation on a Digital Signal Processor (DSP) is discussed and results using experimental data are given.

key words: robot vision, stereo reconstruction, DSP application.

1 INTRODUCTION

Many 3d vision tasks with stereo cameras need image feature matching and 3d reconstruction. If time and computational complexity are a minor issue, photogrammetrists will rely on extensive camera calibration and "bundle adjustment", a process which is mostly semi-automatic. It yields the highest precision rating by simultaneously performing object reconstruction, image feature matching, and, if desired, the computation of selected calibration parameters [4]. In space applications often the computational capability is very restricted with respect to memory usage and processing power. Furthermore, many applications require real-time performance and full automatic processes (e.g. visual servoing, scene perception for autonomous vehicles).

The first step to accelerate 3d reconstruction from stereo images is the gradual decomposition of the process into calibration, image feature matching, and pure geometrical reconstruction itself. Ideally calibration is needed only once before the images of the object of interest are captured but can be performed to some extent on the basis of the captured images, as well [3, 6]. Image feature matching heavily depends on the actual task and will not be examined in this paper. Methods vary from image patch distortion approaches via hierarchical procedures to knowledge based versions.

The geometrical reconstruction of an object point, given the two 2d image locations of its projections, in two distinct cameras is known as *forward intersection*. This process does *not* change with the application; only timing and accuracy requirements may vary. Forward intersection is needed when no a priori object model is at hand or when object/model matching is performed in 3d space. In reality the thus reconstructed 3d point won't match the exact object point because of imperfections of perception of the projection of the object

Proc. Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space, 1-3 June 1999 (ESA SP-440)

point and imperfect calibration.

Section 2 presents the mathematical formulation for the projection – the camera model – that has been used in our investigations. Section 3 describes the four investigated algorithms that solve the inverse problem, the forward intersection.

The quality of a reconstruction can be judged in two different ways. One is the deviation of the reconstructed point to the real object point, referred to as *absolute deviation*. The second is the divergence of a reconstructed point cloud to the real object point cloud, the *consistency deviation*. In fact, an object is reconstructed as a point cloud and the point cloud is rotated and translated, the consistency deviation remains constant but the absolute deviation changes. The absolute deviation includes the relationship between the object coordinate system and the camera coordinate system – needed for example for autonomous navigation tasks.

The accuracy of the forward intersection is influenced by 2×10 different camera parameters – such as the angle of intersection of the optical axes – by the quality of image point perception (2×2 error parameters), and the quality of the calibration result of the 2×10 camera parameters. Because of this multi dimensional variety it is impossible to base the decision which algorithm is most suitable on a comprehensive error analysis. Instead it must be done with the help of realistic parameters of real world applications, where the parameters are slightly altered and the effects on the reconstruction quality are measured. This can be done by error propagation analysis, as has been done in [10], or by means of simulations. For our investigations we have chosen the latter method.

Section 4.1 depicts a subset of the simulations performed [11] to derive the results presented in section 4.2 and verified by experiments in section 5.

Section 6 discusses the computational requirements of the algorithms and the performance on a space qualified floating point DSP. A short summary is given in section 7.

2 CAMERA MODEL

The pin hole camera model we use does not contain distortion parameters. If needed, distortion parameters can be identified and measured image points corrected according to the distortion model. In this way the described forward intersection algorithms for the pin hole camera model can be used even if distortion needs to be taken into consideration [7].

Given a 3D point $A = (X, Y, Z)^T$ in a world coordinate system (WCS) the metric coordinates (x_{a1}, y_{a1}) and (x_{a2}, y_{a2}) of its two projections measured in pixels is obtained with the help of the so called collinearity equations (1, 2) and an equation that allows the transformation of the metric coordinates $(x_a, y_a)^T$ into pixel coordinates $(x_p, y_p)^T$.

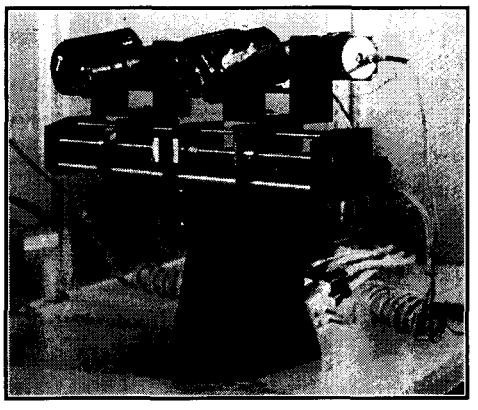

Figure 1: The KTH Robot Head.

$$x_a = -c\frac{r_{11}(X-X_0)+r_{21}(Y-Y_0)+r_{31}(Z-Z_0)}{r_{13}(X-X_0)+r_{23}(Y-Y_0)+r_{33}(Z-Z_0)} \quad (1)$$

$$y_a = -c\frac{r_{12}(X-X_0)+r_{22}(Y-Y_0)+r_{32}(Z-Z_0)}{r_{13}(X-X_0)+r_{23}(Y-Y_0)+r_{33}(Z-Z_0)} \quad (2)$$

$$x_p = x_a/s_x + x_0 \quad (3)$$

$$y_p = y_a/s_y + y_0 \quad (4)$$

The parameter c is the distance from the optical centre of the lens to the projection plane of the camera and s_x, s_y are the lengths of the pixels in the x and y directions. The coordinates of the principal point are (x_0, y_0). These intrinsic parameters do not change, if the cameras are moved and can be identified by calibration[1] [7, 8]. The other (extrinsic) parameters determine the translation (X_0, Y_0, Z_0) and the rotation to the WCS. The values r_{ij} are the matrix components of a rotation matrix that depends on three rotation parameters of the camera [5].

The extrinsic parameters can be identified by calibration but vary with camera movement. If the cameras are mounted on a robot head, see fig. 1, the kinematics of the robot head can be included in a more comprehensive model. Such a model reveals the extrinsic parameters with respect to the commanded poses of the cameras of the robot head [9].

3 THE FOUR ALGORITHMS

This section presents the four algorithms that are subject to our investigation. They are presented in the order of their computational complexity.

[1] The values of intrinsic parameters may vary if a zoom lens is used. In this case a lookup table can be established.

APP: The simplest algorithm expresses the X and Y coordinate of A out of eq. 1, 2 as follows [5].

$$X = X_0 + (Z - Z_0) \cdot \frac{r_{11}x_a + r_{12}y_a - r_{13}c}{r_{31}x_a + r_{32}y_a - r_{33}c} \quad (5)$$

$$Y = Y_0 + (Z - Z_0) \cdot \frac{r_{21}x_a + r_{22}y_a - r_{23}c}{r_{31}x_a + r_{32}y_a - r_{33}c} \quad (6)$$

These equations appear two times, once for the left and once for the right camera. From the two equations 5 we can express Z but from the two equations 6, as well. The mean value is taken and X and Y computed [5].

VECTOR: The computationally second cheapest algorithm reconstructs the beams of sight from the known camera parameters and the measured image points. These will not intersect – due to the imperfections mentioned. The object point A is assumed to lie in the middle of the shortest distance between the two beams [4], see fig. 2.

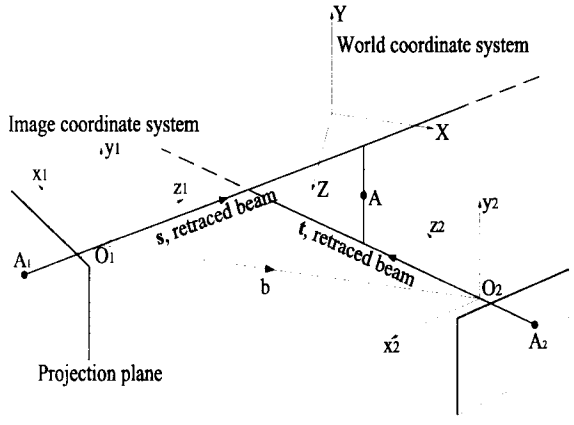

Figure 2: Point reconstruction with vector analysis.

LIN: Let's rewrite the 2 × 2 collinearity equations as shown for one camera [8, 11]

$$(r_{13}x_a + r_{11}) \cdot X + (r_{23}x_a + r_{21}) \cdot Y + (r_{33}x_a + r_{31}) \cdot Z = (r_{13}x_a + r_{11})X_0 + (r_{23}x_a + r_{21})Y_0 + (r_{33}x_a + r_{31})Z_0 \quad (7)$$

$$(r_{13}y_a + r_{12}) \cdot X + (r_{23}y_a + r_{22}) \cdot Y + (r_{33}y_a + r_{32}) \cdot Z = (r_{13}y_a + r_{12})X_0 + (r_{23}y_a + r_{22})Y_0 + (r_{33}y_a + r_{32})Z_0 \quad (8)$$

We interpret this as a linear system of equations in the variables X, Y and Z. The left hand side can be rewritten as a matrix S multiplied with the coordinates of A. The right hand side is a four dimensional column vector b. A least squares estimation for A is given by

$$A = (S^T S)^{-1} S^T b \quad (9)$$

NONLIN: Given a start estimation A_0 for the coordinates of A – based on one of the three algorithms described previously – we can compute the theoretical image points $(x_a^{mod}, y_a^{mod})^T$ in each camera by inserting A_0 into equations 1 and 2. In general, these theoretical values will deviate by some few pixels from the measured image points (x_a^{meas}, y_a^{meas}). We may tune the values for the coordinates of A_0, obtaining a new guess A_1 such that the deviation

$$((x_a^{mod}, y_a^{mod})^T - (x_a^{meas}, y_a^{meas})^T)^2 \quad (10)$$

becomes smaller. This process is reiterated until a threshold is reached. If we take several points simultaneously, we can also include camera parameters in this adjustment and overcome imperfections of the calibration. The resulting minimisation problem is best solved with a Gauss-Newton method. Photogrammetric literature contains the Jacobian Matrix needed for a numerically efficient implementation in symbolic form [5]. The computational costs of the process described increase with the square of the number of points used.

In photogrammetric applications it is common to add control points to the fields of view of the cameras. Control points are points with known coordinates in the WCS. This is not relevant for space applications and therefore omitted in this article.

4 SYNTHETIC DATA EXPERIMENTS

4.1 Description of the Simulations

In the simulations the camera set-up always consists of two identical cameras in differnet poses. The working space is defined as the area which is covered by both camera views. Within this working space a cloud of random object points is generated. The projection of every single object point onto the projection planes of both cameras is computed with the model given in section 2. To model erroneously evaluated image point locations, random pixel noise is added. The evaluation of the four forward intersection procedures has been performed on a basis of 10000 random points per camera set-up and has been performed for different scales and applications [11]. This section presents results from a possible mobile robot application where we place the WCS into the object centre, for simplicity, without loss of generality.

Let's warm up with simulations where we consider a camera set-up of two cameras facing the origin of the WCS with a constant distance to the origin of the WCS. The cameras are moved along a cricular course with its centre in the origin of the WCS. Their current pose is described by the angle between the cameras. For a mean absolute pixel noise set to 0.2 pixels and perfectly calibrated cameras (i.e. the exact intrinsic and extrinsic camera parameters are known) we notice similar deviations of the reconstruction results for all four algorithms. The accuracy of the reconstructions only differs in the range between 150° and 180°.

Large deviations are observed in the regions of very small and very large angles between the cameras (especially for APP), caused by the glancing intersection of the viewing beams at these angles. The *absoulte deviations* reach their minimum at an angle of 90° between the cameras, of course.

We will now discuss more realistic scenarios: noisy data and imperfectly calibrated cameras. The camera set-up complies with a typical set-up of a mobile robot. Both cameras are at a distance of 4 m to the origin of the WCS, the angle between the cameras is 10° and the working space is constrained by a cylinder with a diameter of 3 m, see fig. 3. To estimate the influence

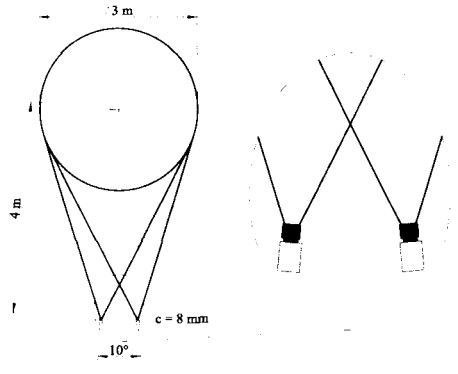

Figure 3: Camera set-up of a mobile robot.

of the calibration error of each of the three angles describing the rotation of the left camera, we set the mean absolute pixel noise to 0.5 pixels and vary the calibration error. Figures 4 to 6 demonstrate the simulation results. The unit of the ordinates are the *absolute deviations*, normalised to the range of the working space. Note the large influence of an erroneously calibrated vergence angle on the *absolute deviation* of the reconstruction. NONLIN may converge to a wrong minimum, because of the missing reference to the WCS.

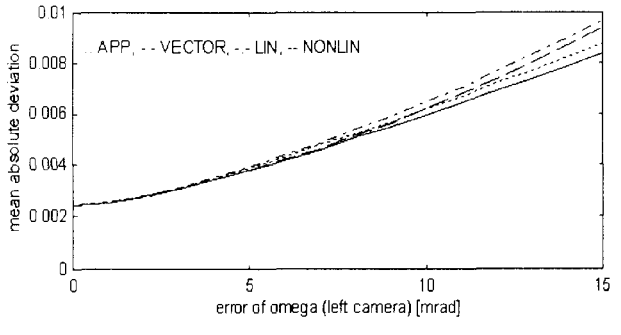

Figure 4: Imperfectly calibrated elevation.

Assuming that all 3 angles, that describe the rotation of one camera, are calibrated erroneously, we can

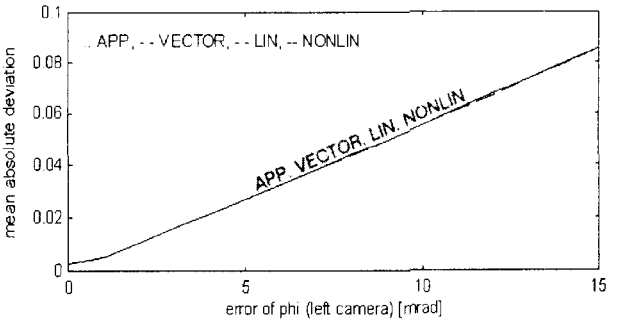

Figure 5: Imperfectly calibrated vergence.

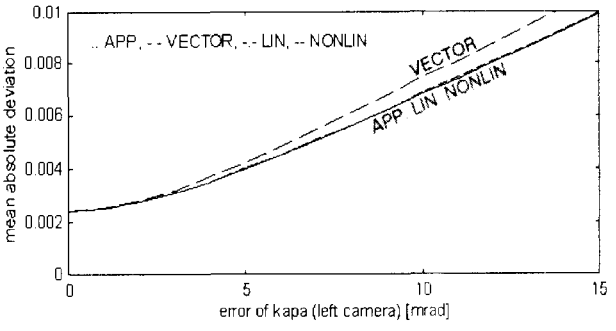

Figure 6: Imperfectly calibrated roll.

observe the *absolute deviations* of the reconstructions illustrated in fig. 7. A random noise of 0.5 pixels is added to the projection points. In the subsequent sim-

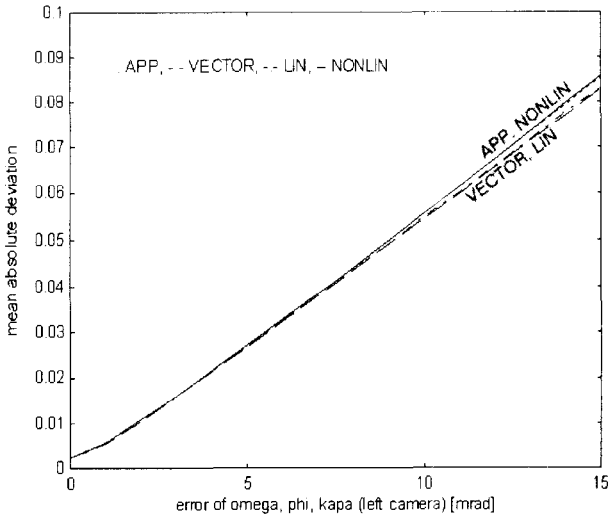

Figure 7: Imperfect calibration of one camera, *absolute deviation.*

ulation NONLIN regards the camera calibration parameters. That means the set of unknowns is extended with the erroneously calibrated angles of one camera. NONLIN uses 10 object points simultaneously in its minimisation procedure. Thus, the overdetermined system of equations consists of 40 equations derived from the 10 points to evaluate 33 unknowns (10 × 3

object coordinates, 3 angles). Fig. 8 presents the *consistency deviations* for the same set-up. We see that this is the domain of NONLIN, because it can correct the defective calibration parameters at the cost of computational complexity and the need for *several* points.

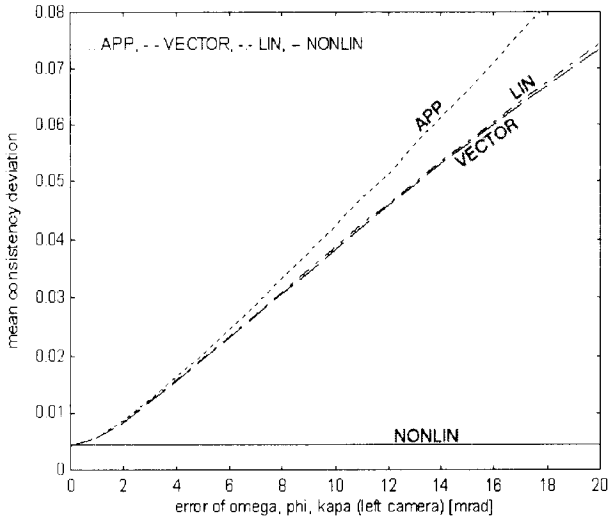

Figure 8: Imperfect calibration of one camera, *consistency deviation.*

4.2 Simulation Results

For the *absolute deviations* we can nominate a clear winner of the contest: VECTOR, followed closely by LIN. This was the case for all simulations we have performed. APP is third, NONLIN can not compete – it might converge to a wrong minimum, because its minimisation does not regard the WCS.

The *consistency deviations* are minimal when NONLIN is used. However, if the computational power is restricted VECTOR is the second best – with one exception. A badly calibrated roll error of one of the cameras can irritate VECTOR considerably and make LIN the winner. The roll of a camera is defined as a rotation about its optical axis. Robot heads mostly have a rigid construction that only allow pan and tilt movements, i.e. no rolling, and thus it is very unlikely that VECTOR is outperformed by LIN in most robot applications. The last – APP – lags more behind the others, when consistency deviations are regarded, compared to the results for absolute deviations.

5 REAL WORLD EXPERIMENTS

The implemented algorithms have been tested with real data provided by off-the-shelf cameras in a stereo set-up. Two JAI-235 industrial CCD cameras were mounted at a distance of 1000 mm to the object coordinate system, the angle between the cameras was fixed at 15° and thus the working space was constrained by a cylinder with a diameter of 377 mm. In the working space an ellipsoid solid was situated, see fig. 9 for the images of the left and right camera. The object coordinates of marked points on the ellipsoid have been determined previously with the help of a high precision bundle adjustment together with extensive camera calibration and the image coordinates of the respective projections have been identified by ellipse fitting.

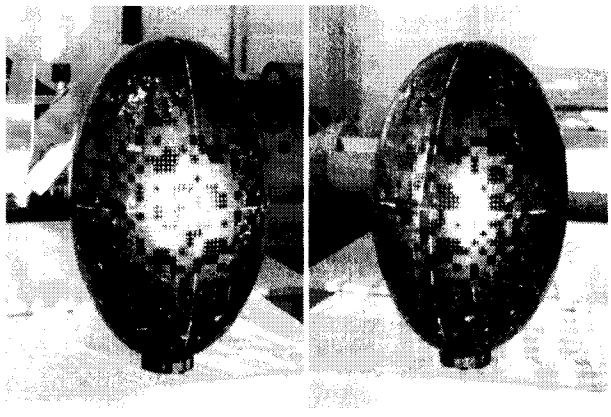

Figure 9: Images of the binocular camera set-up.

The following table lists the mean *absolute deviations* and the mean *consistency deviations* of the reconstructed object points normalised to the range of the working space. In this first approach NONLIN does not include calibration parameters in the minimisation procedure.

	deviation $[10^{-3}]$	
algorithm	*absolute*	*consistency*
APP	15.2	15.6
VECTOR	3.4	2.3
LIN	3.4	2.3
NONLIN	3.3	0.6

If NONLIN regards the camera calibration parameters of one camera and computes the object coordinates of 40 points simultaneously its mean *consistency deviation* reaches $0.2 \cdot 10^{-3}$, but note the quadratically increasing computational costs.

The results of these practical experiments underline the simulation results. Due to the comparatively large deviations of APP this algorithm is recommended as a start estimation, only.

6 DSP IMPLEMENTATION

The presented algorithms require the following number of floating point operations (FLOPS) for the reconstruction of one object point. The MATLAB im-

plementation used was without sophisticated optimisations and can be retreived from [11].

algorithm	FLOP count
APP	70
VECTOR	130
LIN	363
NONLIN	> 1600

For the given comparision, the input to each algorithm comprises the two image points and the two camera coordinate systems given as homogeneous matrices. For a DSP Implementation the FLOP count is an inadequate measurement to estimate processing requirements. Thus we present the cycle count for each algorithm on a typical floating point signal processor.

The TSC21020E is a radiation hardened version of the ADSP 21020. It exhibits an enhanced Harvard Architecture that allows loading or storing data in the program memory and data memory simultaneously, if the instruction is found in an instruction cache. Since forward intersection will normally be performed for more than a single point we assume that the respective algorithm is performed in a loop and makes use of a 2×32 LRU instruction cache. Furthermore this processor can perform a multiplication simultaneously with an addition or subtraction.

For optimised implementations the following table shows the execution times in processor cycles and – for convenience – in micro seconds on a DSP operating at 20Mhz. The division is implemented for the full 32 bit floating point resolution and requires 8 cycles.

algorithm	cycles	time (μs)
APP	56	2.8
VECTOR	152	7.6
LIN	213	10.65
NONLIN	> 800	> 40

As can be seen, the difference of computation costs between LIN and VECTOR gets smaller, but VECTOR remains the winner.

7 SUMMARY

In our paper we have addressed the problem of three dimensional point reconstruction given the projections of a point in two digital cameras with known extrinsic and intrinsic parameters. We have compared four different algorithms and have shown in which situation which algorithm is the best choice with respect to two different measures of precision. Finally we have given figures for processor loads for implementations. The MATLAB implementations of the algorithms can be obtained in electronic form [11].

References

[1] P. Wunsch, G. Hirzinger, "Real-Time Visual Tracking of 3-D Objects with Dynamic Handling of Occlusion". *Proc. Int. Conf. on Robotics and Automation*, Abuquerque, New Mexico - April 1997, pp. 2868 – 2873.

[2] G. Paar, W. Poelzleitner, "Stereovision and 3d Terrain Modeling for Planetary Exploration". *1st ESA Workshop on Comp. Vision and Image Processing for Spaceborne Applications*, Noordwijk, NL, June 1991.

[3] M. Ulm, G. Paar, "Relative Camera Calibration from Stereo Disparities". *Optical 3D-Measurement Techniques III* edited by Gruen/Kahmen, Wichmann, Heidelberg, 1995, pp. 526 – 533.

[4] K.B. Atkinson (Ed), *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, Caithness, 1996.

[5] K. Kraus: *Photogrammetry*. Dümmler Verlag Bonn, 1993.

[6] R.I. Hartley, "In defence of the 8-point algorithm". *IEEE Trans. PAMI*, vol. 19(1997), no. 6, pp. 580 – 593.

[7] J. Weng, P. Cohen, M. Herniou: "Camera Calibration with Distortion Models and Accuracy Evaluation." *IEEE Trans. PAMI*, vol. 14 (1992), no. 10, pp. 965 – 980.

[8] Mengxiang Li, "Camera Calibration of a Head-Eye System for Active Vision". *TR 147*, Computer Vision and Active Perception Lab, Royal Inst. of Technology (KTH), Stockholm, 1994.

[9] S. Spiess, M. Li, "Kinematic calibration of an active binocular head for online computation of the epipolar geometry". *TR 205*, Computer Vision and Active Perception Lab, Royal Inst. of Technology (KTH), Stockholm, 1996. See `http://www.bion.kth.se`.

[10] N. Georgis, M. Petrou, J. Kittler, "Error Guided Design of a 3D Vision System". *IEEE Trans. PAMI*, vol. 20 (1998), no. 4, pp. 366 – 379.

[11] R. Krickl, *A comparison of four forward intersection methods for binocular computer vision*. Master Thesis, Inst. of Flexible Automation, Vienna Univ. of Technology, 1998. For electronic copies contact authors.