3D projective reconstruction with decomposed projective depths

Ferenc Tél[§] Béla Lantos[‡]

***Abstract*-This paper describes a bundle-adjustment based method that can be used to recover of 3D projective structure and camera matrices from multiple images taken about the scene. The difference from the previous methods are twofold. First, the minimization is based on the reprojection error using Euclidean distance on the image planes, unlike factorization based methods, that use algebraic (SVD) reprojection error. Iterative method is used to minimize the reprojection errors. Second, it directly addresses the computation of the *m+n* independent parameters of the projective depths instead of using *mn* dependent parameters. This reduces the number of parameters that should be calculated and automatically involves the computation only of the required parameters.**

***Index Terms*-3D projective reconstruction, bundle adjustment, reprojection error**

I. INTRODUCTION

More and more applications such as intelligent robot control algorithms (e.g. path planning, collision avoidance), object reconstruction methods, augmented virtual reality, etc. require the 3D description of the surround world. This paper describes the projective reconstruction method that was developed as the part of the intelligent stereo vision system for PUMA robot and dexterous hand at BUTE. Older stereo methods use calibrated cameras to recover the Euclidean structure. But it turned out, that the calibration based methods have several drawbacks. The calibration process usually cannot be made on-line and it supposes, that the camera parameters will not be changed later, therefore some types of cameras (e.g. auto zoom) cannot be applied. Many applications (e.g. invariant based object recognition) do not require the detailed Euclidean reconstruction at all. Therefore the reconstruction process can be divided into two independent phases. First recover the projective structure of the scene and motion of the cameras and apply Euclidean (or affine) constraints later, only if it is required.

Projective reconstruction algorithms use perspective images of uncalibrated cameras to extract information about the 3D scene structure. It have been proven that in the uncalibrated case the scene can only be reconstructed up to an unknown projective transformation (collineation), if no other constrains are involved.

Several methods have been developed in the last years to recover the 3D projective structure of the scene and the respective camera motion.

Most mature methods [1,2] use stereo image pairs to determine the epipolar geometry that describes the relationship between images. The epipolar relations are usually characterized by a 3x3, rank 2 homogeneous matrix, called fundamental matrix. Different methods exist to calculate the fundamental matrix. Linear algorithm [3] usually minimizes algebraic distances and do not include the rank 2 constraint of the fundamental matrix. Nonlinear methods minimize real Euclidean distances on the images and use special parameterizations [4] or iterative methods [5] to enforce the rank 2 condition. Nonlinear methods need an initial estimation which can be found by using linear methods.

It turned out that there are also strong relationships between more than two views. Shashua [6] describes a trilinear tensor involving three images into reconstruction process. Hartley [7] showed that this tensor can also be used to recover lines.

Faugeras et al. [8] and Triggs [9] proposed similar methods to recover the structure from any number of views. This factorization based method uses the fact, that the rank of the scaled measurement matrix must be 4. But this method requires the estimation of the projective depths to obtain a possible reconstruction. Han et. al. [10] propose an iterative method to calculate projective depths. Triggs [9] uses the set of fundamental matrices to achieve this task without iterations, but it requires the calculation of the epipolar relations between image pairs.

One of the drawback of the factorization based methods is the handling of the missing data. It is possible that some of the features cannot be seen on all of the views, mainly for longer image sequences, therefore the measurement matrix constrains “holes”. Jacobs [11] proposed a method to determine the missing elements. This method have been further improved by Martinec et el. [12].

Another drawback of the factorization algorithms that they minimize an algebraic entity, called SVD reprojection error. Unfortunately this lacks any physical meaning, therefore it gives only a sub optimal solution.

Another way to estimate the scene structure is to use bundle adjustment methods. One version of these type of methods was developed by Quan et al. [13] which directly minimizes the reprojection error. This method requires nonlinear least

[§] Budapest University of Technology and Economics
Dept. Control Engineering and Information Technology, H1117 Budapest, Magyar Tudósok krt 2, Hungary
email: tel@opsys.hu

[‡] Budapest University of Technology and Economics
Dept. Control Engineering and Information Technology, H1117 Budapest, Magyar Tudósok krt 2, Hungary
email: lantos@iit.bme.hu

squares optimizers. Application of these types of estimators, such as Levenberg-Marquardt method, can be slow in case of large number of views and/or features. Nonlinear methods require also an initial estimation, that can be calculated with a linear method, mentioned above. The advantage of the bundle adjustment based methods is that they can easily manage the handling of the missing data by simply ignoring the missing terms during the minimization. These algorithms can handle a common framework for different types of features (points, lines).

In this paper a bundle adjustment based algorithm is proposed, that decouples the calculation to the calculation of the structure (intersection), projection matrices (resection) and projective depths to eliminate the nonlinear optimization steps. Unlike previous similar methods [14], the proposed method estimate only the required (and independent) $m+n$ coefficients instead of calculating mn quatity separately. From these $m+n$ coefficient the projective depths can be calculated.

II. PROJECTIVE RECONSTRUCTION FROM VIEWS

This section describes the reconstruction of 3D features (currently points) from multiple image projections. Let M_j represent the homogeneous coordinate vector of *j*th 3D point, P_i be the 3x4 projection matrix for the *i*th camera and q_{ij} the homogeneous coordinate vector of the projection of the *j*th spatial feature on the *i*th image. Each entity is defined up to a nonzero scale factor. The number of views (image projections) are *m*, the number of 3D features are *n*.

The projection equation can be written into the following form:

$$\lambda_{ij} q_{ij} = P_i M_j$$

Using this equation it can be seen that in case of uncalibrated cameras the scene can be reconstructed up to a nonsingular projective transformation, *T*:

$$\lambda_{ij} q_{ij} = \left(P_i T^{-1}\right)\left(T M_j\right)$$

These λ_{ij} scale factors are called *projective depths*. There exist mn scale factors (one for each projection) but only $m+n$ are are independent among them.

III. DECOMPOSITION OF THE PROJECTIVE DEPTH

Each λ_{ij} depends on two quantities, the π_i factors are related to cameras and the γ_j factors are related to 3D features. Therefore each projective depth can be written as a product of these quantities:

$$\lambda_{ij} = \pi_i \gamma_j$$

Applying these facts, the joined projection equations can be written into the following matrix equation:

$$\Pi Q \Gamma = PM$$

where

$$\Pi_{3m\times 3m} = \langle \pi_1 \quad \pi_1 \quad \pi_1 \quad \pi_2 \quad \pi_2 \quad \pi_2 \quad \cdots \rangle$$

$$Q_{3m\times n} = \begin{bmatrix} q_{11} & q_{12} & \cdots \\ q_{21} & q_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

$$\Gamma_{n\times n} = \langle \gamma_1 \quad \gamma_2 \quad \cdots \rangle$$

$$P_{3m\times 4} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \end{bmatrix}, \quad M_{4\times n} = \begin{bmatrix} M_1 & M_2 & \cdots \end{bmatrix}$$

Here <.> denotes diagonal matrices. These equations are valid only for point projections. Initially only the elements of *Q* are known from image measurements (e.g. as output of feature detector). It can be seen that in an ideal (noise free, non degenerate) case, the rank of the $R = \Pi Q \Gamma$ measurement matrix must be 4 (as a product of two rank 4 matrices, *P* and *M*). If the λ_{ij} projection depths were known, the joint projection matrix *P* and the projective shape *M* could be determined by using a decomposition method (e.g. SVD). This is the base of the mentioned factorization method. But there are some drawbacks of the factorization method:

- It minimizes an algrebraic distance, called SVD reprojection error [15], that does not represent any physically meaningful quantity.
- The handling of the missing data requires special attention. The missing elements of *Q* should be estimated before factorisation.
- In this form, the projection equations represent the projections of points only. Higher level features (e.g. lines) could only be used as point sets.

IV. THE NEW RECONSTRUCTION ALGORITHM

In this paper we propose a method, that aims to solve the first two problems (but the extension to the line features will be mentioned, too), and estimates only the minimal required number of parameters. Therefore using the original projection equation a cost function can be defined as the difference in the position between the estimated and the real feature (point) projections:

$$E(\cdot) = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}^2 \left\| \hat{\pi}_i \hat{\gamma}_j q_{ij} - \hat{P}_i \hat{M}_j \right\|$$

where the elements denoted by '^' are the estimated values. The w_{ij}^2 values are weights, that can be used to make the algorithm more robust, e.g. features with large error can be classified as outliers and can be eliminated from the estimation process.

It can be seen, that function *E*(.) is nonlinear in the unknowns. Some algorithms [e.g. 13] use the Levenberg-Marquardt method and general initial values to directly minimize the cost function *E*(.). But fortunately the parameters to be estimated can be separated into different groups, because they are "independent" from each other. This is the well-known *resection-intersection* method, that holds every group of parameters fixed, except those, that are currently minimized. Therefore the minimization of *E*(.)

can be achieved by minimizing the values $\hat{P}_i, \hat{M}_j, \hat{\pi}_i, \hat{\gamma}_j$ separately.

A. Estimation of the parameter groups

1) Minimization in $\hat{M}_j$

During the minimization of projective shape $\hat{M}_j$, the values of the other parameters $\hat{P}_i, \hat{\pi}_i, \hat{\gamma}_j$ are treated as constants. The M_j's as 3D projective features are independent from each other, because they depend only on the objects in the scene and they are not influenced by the projections. Therefore the estimation for the *j*th feature can be calculated by making the derivative of *E*(.) by $\hat{M}_j$ to zero:

$$\hat{M}_j = \left(\sum_{i=1}^{m} w_{ij}^2 \hat{P}_i^T \hat{P}_i \right)^{-1} \left(\sum_{i=1}^{m} w_{ij}^2 \hat{P}_i \hat{\pi}_i \hat{\gamma}_j q_{ij} \right)$$

2) Minimization in $\hat{P}_i$

As for the shape values, the cameras are also independent from each other (theoretically the cameras can be placed anywhere around the scene). Therefore the projection matrices could be estimated separately. In order to solve for the values of $\hat{P}_i$, the elements are stored into a vector

$$\hat{p} = [\hat{p}_{11} \quad \hat{p}_{12} \quad \hat{p}_{13} \quad \hat{p}_{14} \quad \hat{p}_{21} \quad \cdots]^T$$

generating a matrix A from the elements of M_j

$$A = \begin{bmatrix} \hat{M}_j^T & 0_4^T & 0_4^T \\ 0_4^T & \hat{M}_j^T & 0_4^T \\ 0_4^T & 0_4^T & \hat{M}_j^T \end{bmatrix}.$$

The cost function becomes

$$E(\cdot) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij}^2 \left\| \hat{\pi}_i \hat{\gamma}_j q_{ij} - \hat{A}_j \hat{p}_i \right\|$$

Making the derivative of *E*(.) by $\hat{p}_i$ to zero yields the solution in closed form:

$$\hat{p}_i = \left(\sum_{j=1}^{n} w_{ij}^2 A_j^T A_j \right)^{-1} \left(\sum_{j=1}^{n} w_{ij}^2 A_j \hat{\pi}_i \hat{\gamma}_j q_{ij} \right)$$

3) Minimization in $\hat{\gamma}_j$

The shape dependent factors of the projective depths can be easily calculated from the derivative of E(.) by $\hat{\gamma}_j$ in closed form

$$\hat{\gamma}_j = \frac{\sum_{i=1}^{m} w_{ij}^2 q_{ij}^T \hat{P}_i \hat{M}_j}{\sum_{i=1}^{m} w_{ij}^2 \hat{\pi}_i q_{ij}^T q_{ij}}$$

4) Minimization in $\hat{\pi}_i$

The camera dependent factors of the projective depths can be easily determined from the derivative of E(.) by $\hat{\pi}_i$ in closed form

$$\hat{\pi}_i = \frac{\sum_{j=1}^{n} w_{ij}^2 q_{ij}^T \hat{P}_i \hat{M}_j}{\sum_{j=1}^{n} w_{ij}^2 \hat{\gamma}_i q_{ij}^T q_{ij}}$$

B. Handling of missing data

The handling of missing data during the minimization is easy. Skip those *i,j* entries in the error function, that do not have valid q_{ij} value (no projection of the given feature is detected on the image).

C. Steps of the minimization of E(.)

The parameters of the cost function are estimated using an iterative method, therefore an initial estimation for its values is required. This can be achieved as follows.

Choosing the subset of points that can be seen on all of the images, a rank 4 factorization method is achieved. This gives initial estimation for all of the required projection matrices and for those points, that are involved in the factorization. The remaining features can be initialized using backprojected points. This means the determination of a point which has minimal distance from the rays connecting the image points and the camera focal points in least squares sense. All of the π_i and γ_j values are initialized to 1.

The algorithm itself consists of the repeated steps of the minimization from 1) to 4). After every iteration the reevaluation of the w_{ij} weighting factors are achieved and the actual value of the cost function is calculated. If the cost is less than a desired threshold (or maximum allowed number of iterations is reached), the algorithm terminates.

V. RESULTS

We tested our method using simulated data in order to check the robustness and accuracy of the algorithm. The scenes consist of random point sets generated within the box having edges between [-1:1] unit along each axes. The cameras are placed randomly around the scene, the distances from the origin are approximately 5-8 units. The viewing directions are perturbed, the internal parameters of the cameras are also varied slightly but the overall projections yield the projected image features fall into the usual 512x512 image size.

In the first experiment Gaussian noise with different standard deviations was added to the projected points, where the standard deviations are varied between 0.0 and 2.5 pixels. The average reprojection errors for the trials using 20 points are depicted on Figure 1. It can be seen, that the relationship is almost linear between the pixel noise and the reprojection error.

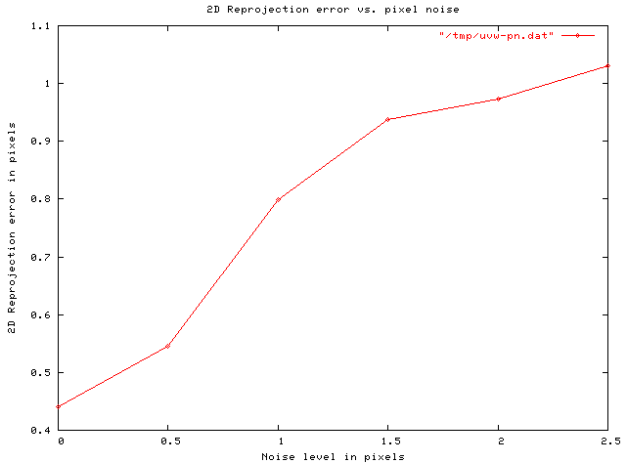

Figure 1: 2D reprojection error vs. pixel noise in case of 20 points

The second experiment examined the effect of the number of used image features. The number of points varied between 8 to 100. The results can be seen on Figure 2. We found that the volume of the reprojection error is almost constant with respect to the number of points above 40-60. In this case the noise was fixed with standard deviation 1.0.

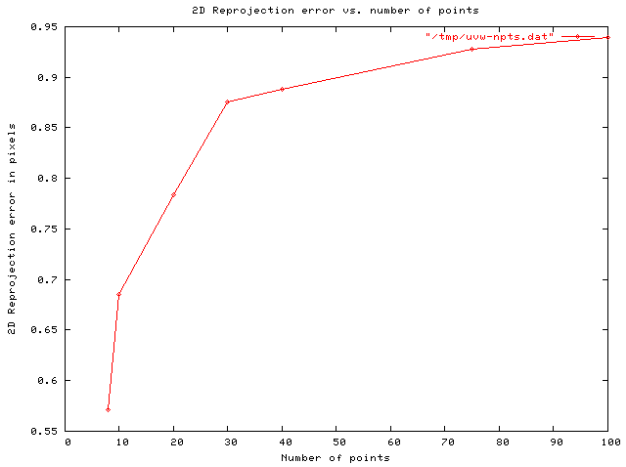

Figure 2: 2D reprojection error vs. number of points in case of pixel noise with standard deviation 1.0

Using the fact, that because of simulation the 3D Euclidean position of the original scene points are exactly known, we also tested the accuracy of the Euclidean reconstruction. To achieve this, we determined those transformation, that maps from the projective to Euclidean representation, using all of the projective-Euclidean point pairs. Applying this transformation to the projectively reconstructed features, the results for a sample scene can be seen on Figure 3. The numerical results of these trials can be seen in Figure 4.

Last experiment was to determine the behavior of the reconstruction algorithm with different number of cameras. The result was, that the reprojection errors are slightly increased using more cameras, see Figure 5. Therefore at first sight it seems useless to involve more cameras into the reconstruction process. But considering the accuracy of the 3D Euclidean reconstruction, it turned out that increasing the number of cameras the reconstruction errors become smaller.

We found, that this error term is also influenced by the spatial configuration of the cameras. Cameras differed only in distance from the scene but almost common optical axes gave unacceptable results, because the backprojected rays from the matched image points were nearly identical (there were no acceptable baseline information). Slightly more distributed cameras yield better results (smaller reconstruction errors).

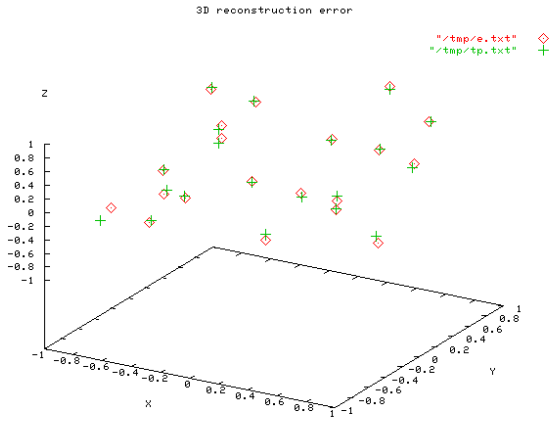

Figure 3: 3D reconstruction for a sample scene for 20 points

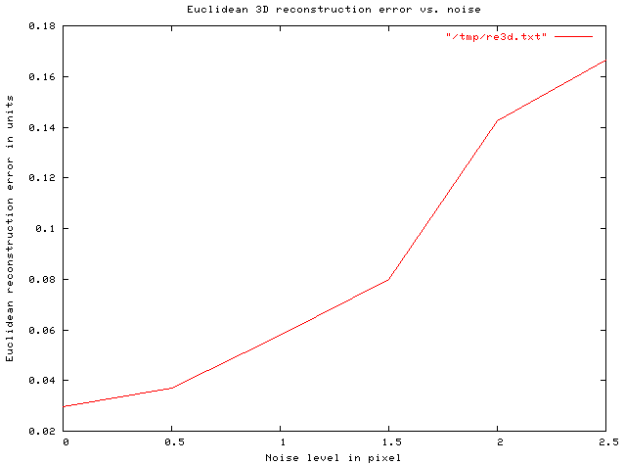

Figure 4: 3D reconstruction error vs. pixel noise.

During the tests the number of iterations required by the algorithm were between 5-20 to assure convergence for non-degenerate configurations.

VI. IMPLEMENTATION

Typical stereo vision methods use resolution from 256 up to 1024 pixels. Therefore for the average image features, the values contained in the homogeneous coordinate vector q_{ij} could have very different magnitudes, e.g. $u_{ij} \approx 50$, $v_{ij} \approx 500$, $w_{ij} = 1$. In the cost function these magnitudes are doubled (in logarithmic sense) because of squaring. The

magnitude differences can cause numerical problems, ill-conditioning during minimization. To avoid this problem, Hartley [16] proposed a normalization method to transform each image feature such that the center of the point set will be at the origin and the average length of homogeneous coordinate vectors will be approximately 1. Our method uses this standardization process, however this requires to transform back the resulted projection matrices after minimization process in order to get the real solution.

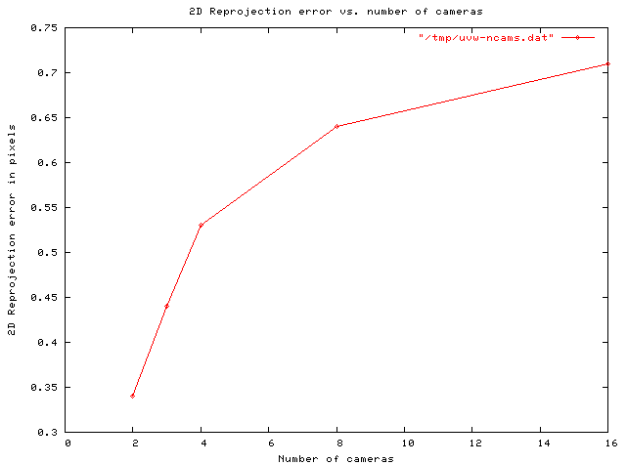

Figure 5: 2D reprojection error vs. number of cameras in case of noise with standard deviation 1.0

VII. CONCLUSION AND FUTURE WORK

This paper proposes a projective reconstruction algorithm that is capable to recover 3D shape and motion from point correspondences. The developed method calculates only the required *minimal* (therefore consequent) set of parameters decomposing the projective depths. The algorithm is also able to handle those cases, where some features cannot be seen in all of the images. Simulated scenes were used to measure the robustness and accuracy of the reconstruction process. It turned out that the algorithm behaves well and gives acceptable results for those scenarios (noise levels, number of features and cameras) that are commonly used in stereo vision.

A further possibility is to extend the algorithm to work with 3D line features, too. This can be helpful for the cases where the application of segment endpoints causes errors because of different occlusion relationships. Such a situation is shown in Figure 6. Point features *a* and *b* can be matched by a feature tracker in the images of camera *C1* and *C2*, respectively. But in reality, these represent different points on the same line (in 3D).

To manage lines as features a natural way is to add a separate term to the cost function:

$$E(\cdot)=\sum_{i=1}^{m}\sum_{j=1}^{n_p} w_{p,ij}^2 \left\| \hat{\pi}_i \hat{\gamma}_j q_{ij} - \hat{P}_i \hat{M}_j \right\| + \sum_{i=1}^{m}\sum_{j=1}^{n_l} w_{l,ij}^2 \left\| \hat{\pi}_i \hat{\omega}_j l_{ij} - \hat{R}_i \hat{L}_j \right\|$$

where l_{ij} is the *j*th line feature detected on *i*th image. $\hat{L}_j$ is the Plücker representation of the line feature by a six dimensional vector, which is the coordinate system independent representation of the line in 3D projective space. The matrix $\hat{R}_i = \begin{bmatrix} \hat{p}_2^T \wedge \hat{p}_3^T \\ \hat{p}_3^T \wedge \hat{p}_1^T \\ \hat{p}_1^T \wedge \hat{p}_2^T \end{bmatrix}_{3\times 6}$ is the line projection matrix composed from the rows of the respective point projection matrix. The "$\wedge$" denotes the meet operation resulting in a six dimensional row vector which can be calculated as the six 2x2 sub determinants of the matrix composed from the two vectors (*a* and *b*)

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \end{bmatrix}$$

using column pairs (1,2), (1,3), (1,4), (2,3), (2,4), (3,4).

Unfortunately this type of line error formulation has some drawbacks:

(i) the line related term does not represent Euclidean distance on image plane,

(ii) the minimization in the parameters of the projection matrix requires nonlinear optimization steps.

Improvements of handling lines in the cost function are in progress.

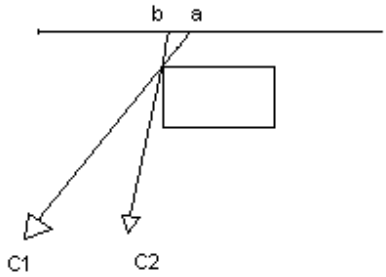

Figure 6: Line segment endpoint uncertainty

VIII. ACKNOWLEDGEMENT

This research was supported by the Hungarian National Research Program under grant No. OTKA T 042634 and by the Control System Research Group of the Hungarian Academy of Sciences.

IX. REFERENCES

[1] O. Faugeras, "**Stratification of 3-D vision: projective, affine, and metric representations**", In *Journal of the Optical Society of America A*, 12(3): pp. 465-484, March 1995

[2] Z. Zhang, "**Determining the Epipolar Geometry and Its Uncertainty: A Review**", In *Int. Journal of Computer Vision*, 27(2), pp. 161-198, Kluwer Academic Publishers, Boston, USA, 1998

[3] O. Faugeras, "**Three-Dimensional Computer Vision, A Geometric Viewpoint**", MIT Press, 1993

[4] G. Csurka, C. Zeller, Z. Zhang, O. Faugeras, "**Characterizing the Uncertainty of the Fundamental Matrix**", *INRIA Research Report 2650*, 1995

[5] R. Hartley and P. Sturm, "**Triangulation**", In *Computer Vision and Image Understanding, number 2 Vol. 68*, pp. 146-157, 1997

[6] A. Shashua. "**Trilinear Tensor: The Fundamental Construct of Multiple-view Geometry and its Applications"**, In *Int. Workshop on Algebraic Frames For The Perception Action Cycle* (AFPAC), Kiel Germany Sep. 8-9, 1997

[7] R.I. Hartley, "**A linear method for reconstruction from lines and points**", In *Fifth Int.. Conf. on Computer Vision*, Cambridge,Massachusetts, USA, pp. 882-887, IEEE, 1995

[8] O. Faugeras and B. Mourrain, "**On the geometry and algebra of the point and line correspondences between *n* images**", In *E. Grimson, editor, IEEE Int. Conf Computer Vision*, pp. 951-956, Cambridge, MA, USA, 1995

[9] B. Triggs, "**Factorization Methods for Projective Structure and Motion**", In *Proc. Of the Conf. On Computer Vision and Pattern Recognition*, pp. 845-851, San Francisco, California, USA, 1996

[10] M. Han and T. Kanade, "**Perspective Factorization Methods for Euclidean Reconstruction**", *tech. report, CMU-RI-TR-99-22*, Robotics Institute, Carnegie Mellon Univ. 1999. http://www.ri.cmu.edu/pubs/pub_3225.html

[11] D. Jacobs, "**Linear fitting with missing data: Applications to Structure-from-Motion and to Characterizing Intensity Images**", In *Proc. Of Conf. On Computer Vision and Pattern Recognition (CVPR'97)*, Puerto Rico, 1997

[12] D. Martinec and T. Pajdla, "**Outlier Detection for Factorization-Based Reconstruction from Perspective Images with Occlusions**", In *Proc. of the Photogrammetric Computer Vision*, pp. 161-164, Inst. f. Computer Graphics and Vision, TU-Graz, September 2002

[13] L. Quan and R. Mohr, "**Projective Reconstruction from Multiple Uncalibrated Images**", In *Modelling and Planning for Sensor Based Intelligent Robotic Systems Vol. 21*, pp. 236-256, 1995

[14] W. K. Tang and Y.S. Hung, "**A Factorization-based method for Projective Reconstruction with minimization of 2-D reprojection errors**", In *Proc. 24th DAGM Symposium*, Zurich, Switzerland, 2002

[15] B. Triggs, "**Some Notes on Factorization Methods for Projective Structure and Motion**", unpublished, http://www.inrialpes.fr/movi/people/Triggs/

[16] R. Hartley, "**In defense of the 8-point algorithm**",In *E. Grimson, editor, IEEE Int. Conf Computer Vision*, pp. 1064-1070, Cambridge, MA, USA, 1995