Reconstruction of projective and metric cameras for image triplets

Andrey Khropov*, Anton Shokurov**, Victor Lempitskiy**, Denis Ivanov**
* Department of Mathematics and Mechanics, Moscow State University
** RL Labs Joint Stock Company, Moscow, Russia
{ akhropov, anton, vitya }@fit.com.ru; denis@rl-labs.com

Abstract

One of the most important parts of 3D computer vision systems is reconstruction of cameras. In this paper we describe our approach to the reconstruction of parameters of the three uncalibrated cameras using information from the three projections of the static scene. We match distinct features of the scene (such as corner points and straight line segments) and robustly sift out outlier matches using RANSAC techniques. Then the optimal trifocal tensor is built using an iterative algorithm which uses inlier matches. This trifocal tensor is used to reconstruct projective cameras. Finally these cameras may be transformed to metric if certain assumptions are presumed. The algorithm pipeline is fully automatic.

Keywords: *3D Reconstruction, Camera reconstruction, Trifocal tensor.*

1. INTRODUCTION

The problem of reconstruction of camera parameters from image sequences has been intensively studied during the last years. It was a considerable step forward from the previous *calibrated* approach when we had to calibrate our cameras explicitly (i.e. compute their intrinsic parameters from special kind of images, for example, chessboard-like black-and-white patterns). This approach is very limiting in real world 3D computer vision systems because it requires special non-automatic procedures and sometimes even particular hardware.

Reconstruction of the cameras is the first part of *Structure-from-Motion* algorithms. The second part is the reconstruction of 3D scene structure which is outside the scope of this paper. There are numerous approaches to this problem. There are a lot of papers on the topic, [17] (but deals with orthographic cameras) ,[2],[16], [11],[1] - to name a few. There is a good tutorial [10] too. In papers [4], [8], [9] and many others methods for camera reconstruction are addressed in particular. And the excellent textbook [7] incorporates the latest information and provides expert advice.

The algorithm pipeline described in this paper is for static scenes. We designed our algorithm with video sequences taken by a consumer-level camcoder in mind. This implies that frame-to-frame displacements are small and the quality of images is relatively poor.

1.1 Notation

Objects in the second image are marked with $'$ and in the third – with $''$. Tracked features are marked with $\bar{}$ and reprojected – with $\tilde{}$.

$\mathbf{x}_i = \begin{pmatrix} u_i^1 & u_i^2 & 1 \end{pmatrix}^T$ - 2D point in homogeneous coordinates

$\mathbf{X}_i = \begin{pmatrix} x_i^1 & x_i^2 & x_i^3 & 1 \end{pmatrix}^T$ - 3D point in homogeneous coordinates

$l_i = \begin{pmatrix} l_i^1 & l_i^2 & l_i^3 \end{pmatrix}$ - 2D line in homogeneous coordinates

$\langle \mathbf{s}_i^1, \mathbf{s}_i^2 \rangle$ - 2D line segment with endpoints $\mathbf{s}_i^1$ and $\mathbf{s}_i^2$.

$\mathbf{e}_i$ - epipole – projection of the i-th camera optical center

I - identity matrix

$A_{\mathbf{x}}$ - axiator matrix of **x** (multiplication by this matrix is equivalent to cross product)

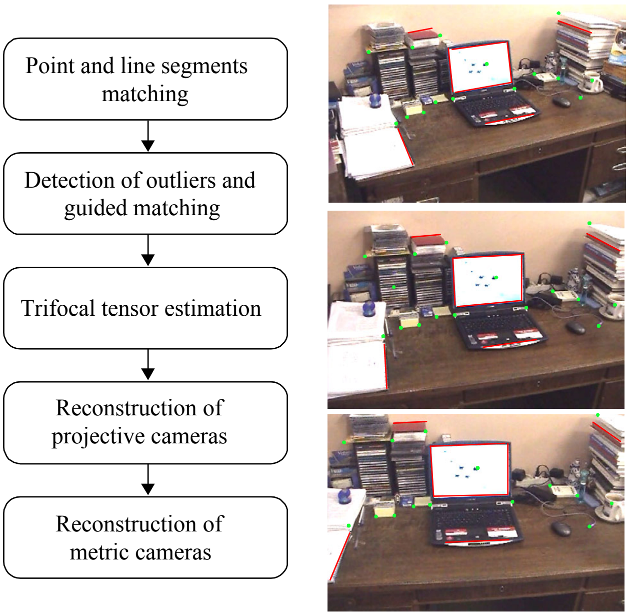

Point and line segments matching
Detection of outliers and guided matching
Trifocal tensor estimation
Reconstruction of projective cameras
Reconstruction of metric cameras

Figure 1. Overall pipeline.

2. FEATURE TRACKING

In this section algorithms for tracking corner points and straight line segments are briefly described. More information can be found in [14]. We use the term *tracking* because in practice we match features throughout the video sequence. We will also use the term *frame* instead of image.

2.1 Point tracking

Our feature point tracker is based on a cross-correlation approach to matching corner points in adjacent frames. We use this technique without multiscale strategy because we expect relatively small displacements between adjacent frames.

Corner points are being detected using the slightly modified Harris corner detector [6]. Modification assures that the points are not initially close to each other.

The tracking algorithm maintains desired number of points by detecting additional points through the sequence.

Possible mismatches are being detected with the help of geometrical constraints for two images (based on a fundamental matrix, see section 3) and three images (based on a trifocal tensor, see section 4) using robust RANSAC-based algorithms (see [5]).

We use guided matching for outlier points (using the estimated fundamental matrix or the estimated trifocal tensor) as described in [2].

2.2 Line tracking

The set of straight line segments is being detected in each frame during the tracking process. We apply the Canny edge detector [3] first to detect pixels that belong to edges. After that chains of connected pixels with a similar gradient direction are linked into segments.

A pair of segments on adjacent frames is considered as a match if:

- Average colors on one of the sides are similar
- Directions of normals to corresponding lines are similar
- Distance between corresponding segment ends is small

These seed matches are verified on the basis of geometrical constraint for projections of 3D line on three images (this constraint is formulated in terms of a trifocal tensor) using a robust RANSAC-based algorithm.

We can also use guided matching for segments without unambiguous correspondences on three frames using the estimated trifocal tensor (see lower).

3. FUNDAMENTAL MATRIX ESTIMATION

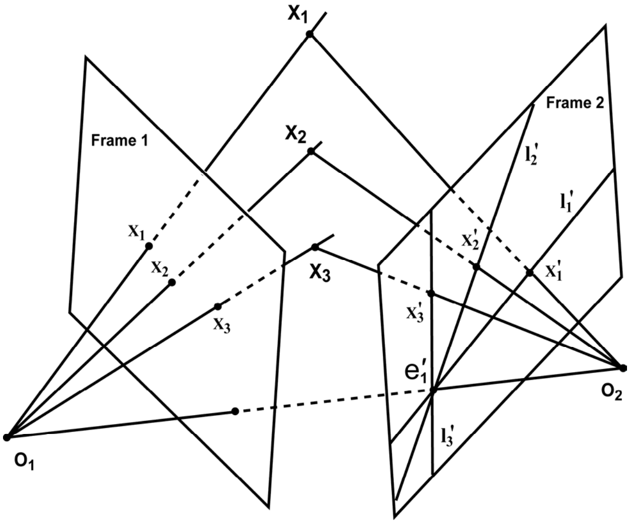


Figure 2.Epipolar constraint.

There is a geometric constraint called *epipolar* for projections of 3D point on two images. See [7] for detailed discussion. This constraint can be algebraically written in terms of *fundamental matrix F*:

$$\mathbf{x}_i^T \cdot F \cdot \mathbf{x}_i = 0 \qquad (1)$$

This constraint is useful for the detection of outliers. Not every 3x3 matrix can be fundamental. First it is defined up to scale, and the second constraint is $\det F = 0$. The algorithm that builds valid fundamental matrix must assure these constraints are satisfied.

Since constraint has the form (1), a point in one image specifies the corresponding *epipolar line* (see Fig.2) that must pass through the projection in the second image.

In the RANSAC scheme random sample of 7 points is being selected. The 7-point algorithm (see [14]) is used to reconstruct a putative matrix. Points that obey the constraint (distance from the corresponding epipolar line is less than a predefined threshold) are called *inliers*, others are called *outliers*. Many random probes are executed and point matches are finally classified using the matrix with the maximum number of inliers.

Without the loss of generality we assume that the first projection matrix is $\mathbf{P} = \begin{bmatrix} I & 0 \end{bmatrix}$ (since anyway they are defined up to projective transformation) and the second is $\mathbf{P}' = \begin{bmatrix} M & e_1' \end{bmatrix}$. Note that the last column of the second matrix is an *epipole* – the projection of the second camera optical center on the first image. All epipolar lines pass through the epipole (see Fig.2).

The optimal fundamental matrix is built using the Levenberg-Marquardt iterative algorithm (see [12]). The function being minimized is a reprojection error

and the parameters are coordinates of reconstructed 3D points and coefficients of matrix F. 3D points positions are reconstructed

$$\sum_i \left(d^2(\bar{\mathbf{x}}_i, \tilde{\mathbf{x}}_i) + d^2(\bar{\mathbf{x}}_i', \tilde{\mathbf{x}}_i') \right)$$

using a linear triangulation (see [7]). The algorithm is initialized with the matrix with the maximum number of inliers.

The idea of guided matching is to restrict search for a match in the second image to the neighbourhood of the corresponding epipolar line.

4. TRIFOCAL TENSOR ESTIMATION

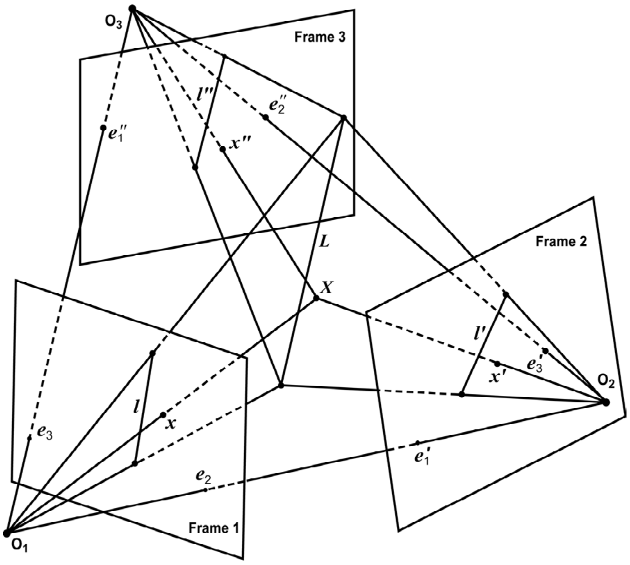


Figure 3. Line and point in three images.

A trifocal tensor is an analog of a fundamental matrix for three projection images [7]. It is more useful than a fundamental matrix because it defines constraints for both points and straight lines. And we also use point and line segment matches on three images to build it.

If lines on three images are presented in homogenous coordinates then a trifocal tensor T is defined as a bilinear operator $l_i = T_i^{jk} l'_j l''_k$ (we use tensor notation and Einstein's rule of summation). (For the derivation of this and the following formulae see [7] or [13]).

Again without the loss of generality we assume that the first projection matrix is $\mathbf{P} = \begin{bmatrix} I & 0 \end{bmatrix}$ (since anyway they are defined up to a projective transformation) and $\mathbf{P}' = \begin{bmatrix} A & e'_1 \end{bmatrix}$, $\mathbf{P}'' = \begin{bmatrix} B & e''_1 \end{bmatrix}$, then tensor components are expressed as

$$T_i^{jk} = \left(e'_1 \cdot (B^i)^T - A^i \cdot (e''_1)^T\right)^{jk} \tag{2}$$

where A^i, B^i are i-th columns of A and B respectively. A constraint for lines is derived from the trifocal tensor definition:

$$A_{\boldsymbol{l_i}} \cdot \left(\boldsymbol{l_i}' \cdot \begin{bmatrix} T_1 & T_2 & T_3 \end{bmatrix} \cdot (\boldsymbol{l_i}'')^T\right) = \mathbf{0}_3$$

Matrices $T_1 = T_1^{jk}, T_2 = T_2^{jk}, T_3 = T_3^{jk}$ are called *tensor slices*. A constraint for points (see [7] or [13]) is written as:

$$A_{\mathbf{x_i}'} \cdot [u_i^1 T_1 + u_i^2 T_2 + T_3] \cdot A_{\mathbf{x_i}''} = \mathbf{0}_{3\times 3}$$

The formula (2) shows that not every set of 27 numbers can be a valid trifocal tensor. In fact a tensor has 18 degrees of freedom (see [13]). So certain constraints or an appropriate parameterization should be chosen to assure validity of tensor. The most natural one is using elements of $\mathbf{P}'$ and $\mathbf{P}''$ (originally proposed by Hartley in [9]). Every choice of these 22 parameters provides a valid trifocal tensor through the formula (2) although this set of parameters is not minimal.

The simplest algorithm minimizes an algebraic error from constraints. Constraints are linear in elements of a tensor. Every point match provides 4 independent homogeneous linear equations and every line match – 2 equations. As a tensor is composed of 27 elements and defined up to scale we need at least 26 independent equations. So a minimum set for building a tensor is 7 points or 13 lines or some mixture of them. The system of equations $\mathcal{A}\,\mathrm{t} = 0$ is solved using SVD: $\mathcal{A} = U \cdot S \cdot V^T$ (see [12]). The last column of V is the solution (corresponding to the minimum eigenvalue of $\mathcal{A}^T\mathcal{A}$)

Again we use a RANSAC procedure similar to the one described in the previous section to build a seed tensor for an iterative estimation algorithm and detect outliers (in this case both points and lines). Use of a RANSAC approach with lines is difficult because on typical images we have significantly less lines (10-25) than points and the building algorithm requires 13 lines.

A criterion for a point to be an inlier is a small distance between the predicted (using trifocal tensor) and the detected (using tracking algorithm) positions in each of the three images.

A criterion for a line to be an inlier is a small distance between the predicted (using trifocal tensor) line and the detected (using tracking algorithm) segment end points in each of the three images.

Finally the best tensor is built using the Levenberg-Marquardt algorithm. The function being minimized is the sum of reprojection error

$$\sum_i \left(d^2(\bar{\mathbf{x}}_i, \tilde{\mathbf{x}}_i) + d^2(\bar{\mathbf{x}}'_i, \tilde{\mathbf{x}}'_i) + d^2(\bar{\mathbf{x}}''_i, \tilde{\mathbf{x}}''_i)\right)$$

for points and a reprojection error for line segments

$$\sum_i \Big(\max\left(d^2(\bar{\mathbf{s}}_i^1, \tilde{\boldsymbol{l}}_i), d^2(\bar{\mathbf{s}}_i^2, \tilde{\boldsymbol{l}}_i)\right) + \max\left(d^2(\bar{\mathbf{s}}_i^{1\prime}, \tilde{\boldsymbol{l}}'_i), d^2(\bar{\mathbf{s}}_i^{2\prime}, \tilde{\boldsymbol{l}}'_i)\right) + \max\left(d^2(\bar{\mathbf{s}}_i^{1\prime\prime}, \tilde{\boldsymbol{l}}''_i), d^2(\bar{\mathbf{s}}_i^{2\prime\prime}, \tilde{\boldsymbol{l}}''_i)\right)\Big)$$

with appropriate weight coefficients and the varying parameters are reconstructed 3D points & lines coordinates and coefficients of $\mathbf{P}'$ and $\mathbf{P}''$. 3D points positions are reconstructed using a linear triangulation (see [7]). 3D lines positions are reconstructed using an algebraic error minimization algorithm described in [7] too.

A tensor may be obtained from projection matrices by the formula (2). The Method for obtaining projection matrices $\mathbf{P}'$ and $\mathbf{P}''$ from the tensor coefficients is described in the next section. It is needed to initialize an iterative algorithm and to reconstruct 3D points and lines for a reprojection.

5. RECONSTRUCTION OF PROJECTIVE CAMERAS

Projection matrices are retrieved from a trifocal tensor using method described in [8]. As usual $\mathbf{P} = \begin{bmatrix} I & 0 \end{bmatrix}$ (since anyway they are defined up to a projective transformation) and $\mathbf{P}' = \begin{bmatrix} A & e'_1 \end{bmatrix}$, $\mathbf{P}'' = \begin{bmatrix} B & e''_1 \end{bmatrix}$.

First we compute epipoles $\mathbf{e}'_1$ and $\mathbf{e}''_1$ using the method presented in [15]. It is principally equivalent to the standard method (see [7]) but more robust because all tensor slices T_1, T_2, T_3 need not to be of rank 2.

Other coefficients (of A and B) satisfy the equation (2). But these equations have not a unique solution. There is a 4-parameter family of solutions. Hartley proposes the following solution:

$$A = \left(\mathrm{e}'_1 {\mathrm{e}'_1}^T - I\right) \cdot \begin{bmatrix} \mathrm{T}_1^T \mathrm{e}''_1 & \mathrm{T}_2^T \mathrm{e}''_1 & \mathrm{T}_3^T \mathrm{e}''_1 \end{bmatrix}$$
$$B = \begin{bmatrix} \mathrm{T}_1 \mathrm{e}'_1 & \mathrm{T}_2 \mathrm{e}'_1 & \mathrm{T}_3 \mathrm{e}'_1 \end{bmatrix}$$

An ambiguity here is solved by demanding that the columns of A are perpendicular to $\mathbf{e}''_1$ hence the second projection center lies in the plane at infinity. We also normalize $|\mathbf{e}'_1| = |\mathbf{e}''_1| = 1$ (in homogeneous coordinates).

6. RECONSTRUCTION OF METRIC CAMERAS

It is common knowledge that projective reconstruction from point projections is defined only up to an arbitrary projective transformation. Indeed, given arbitrary projective transformation T in $\mathbb{P}^3$, one may replace initial points and cameras and with new points and cameras using the following rules: $\hat{P} = PT$ and $\hat{\mathbf{X}} = T^{-1}\mathbf{X}$. This transformation preserves point projections. However, not all these reconstructions are equivalent from metric point of view. Consider arbitrary camera matrix $\mathbf{P} = \begin{bmatrix} M & t \end{bmatrix}$. Using QR decomposition, M can be factored into KR, where K is upper triangular and R is orthogonal. In metric case, R stands for camera orientation, whereas K defines camera intrinsic parameters. If something is known a priori about matrix K, then metric upgrade of projective reconstruction is possible, since metric reconstructions can be distinguished as having specific K-matrices for each camera. In our current implementation we assume that $K = s \cdot \mathrm{diag}\{\, f \;\; f \;\; 1 \,\}$. This corresponds to the case of a camera with square pixels, centered principal point and zero skew. Metric upgrade is performed as follows.

For our initial reconstruction we seek for the transformation T, that brings our cameras to metric state, having lower-tringular matrix (to reduce ambiguity). Assuming that T has been found, we write: $(\underline{PT_r})(\underline{PT_r})^T = \hat{P}\hat{P}^T = KRR^TK^T = KK^T = s^2 \cdot \mathrm{diag}\{\ f^2 \quad f^2 \quad 1\ \}$, where _ means taking left 3x3 matrix and T_r is T with zero right column. This gives us four linear homogeneous equations on the entries of matrix $T_rT_r^T$ per camera**.** Since $T_rT_r^T$ is symmetric it is represented with 10 variables and three cameras gives us 12 equations on 10 variables, allowing for obtaining entries of $T_rT_r^T$ via SVD-decomposition. One can also enforce $T_rT_r^T$ to have rank 3, using SVD. Matrix T_r then can be extracted from $T_rT_r^T$ via Cholesky decomposition. Since Cholesky decomposition is defined only for (semi)positively defined matrices, metric upgrade can fail if $T_rT_r^T$, obtained with our method, is not semi-positively defined due to tracking errors and noise.

Matrix T can be found from T_r by using any constant as T_{44} (this constant stands for the overall scale of reconstruction).

7. CONCLUSION

The method presented in this paper was implemented in program library. It is fully automatic non-realtime method and was tested on still image sets and video sequences both synthesized and captured by an ordinary camcorder.

We are working on an algorithm for camera and structure recovery for long sequences. We also plan to improve robustness in some degenerate cases. Some experiments with constrained tensor estimation (Gauss-Helmert model) are also planned.

8. ACKNOWLEDGMENTS

The research program presented in this paper is partially supported by the grant 03-07-90381 of Russian Foundation for Basic Research.

9. REFERENCES

[1] A. Bartoli, P. Sturm. *Multiple-View Structure and Motion From Line Correspondences*, In Proceedings of the IEEE International Conference on Computer Vision.

[2] P. Beardsley, P. H. S. Torr, and A. Zisserman. *3d model acquisition from extended image sequences*. In B. Buxton and Cipolla R., editors, Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge, pp. 683-695. Springer-Verlag, 1996.

[3] J.F. Canny. *Computational approach to edge detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679-698, 1986.

[4] O. Faugeras. *Camera self-calibration: Theory and experiments*. In European Conference on Computer Vision, Proceedings, Lecture Notes in Computer Science, pp. 321–334. Springer, 1992.

[5] M. Fischler and R. Bolles. *Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography*. Commun. Assoc. Comp. Mach., vol. 24:381-95, 1981.

[6] C.J. Harris and M. Stephens. *A combined corner and edge detector*. In Proc. 4th Alvey Vision Conference, Manchester, pp.147-151, 1988.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, 2nd Edition*. Cambridge, UK: Cambridge University Press, 2003.

[8] R. Hartley. *Projective reconstruction from line correspondences*. In Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'94), pp. 903–907.

[9] R. Hartley. *Lines and points in three views: A unified approach*. In ARPA Image Understanding Workshop, pp. 1009–1016.

[10] M. Pollefeys, *3D Modeling from Images, tutorial notes*, tutorial organized in conjunction with ECCV 2000, Dublin, Ireland, 26 June 2000. Newer online version: http://www.cs.unc.edu/~marc/tutorial.pdf

[11] M. Pollefeys, F. Verbiest and L.J. Van Gool. *Surviving dominant planes in uncalibrated structure and motion recovery*. In ECCV(2), pp. 837-851, 2002.

[12] W.Press, B.Flannery, S.Teukolsky and W. Vetterling. *Numerical recipes in C, 2nd Ed.* Cambridge, UK: Cambridge University Press, 1992. Updated online version: http://www.library.cornell.edu/nr/cbookcpdf.html

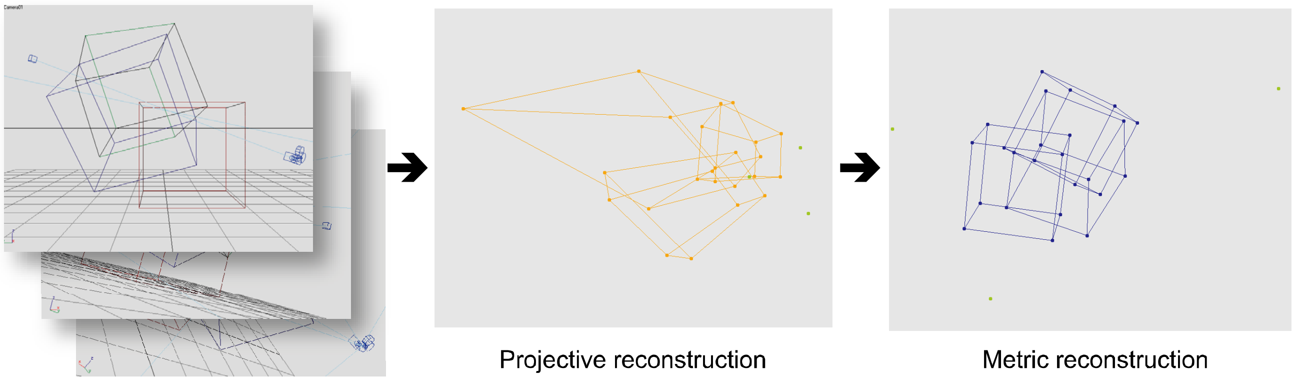
[13] C. Ressl. *Geometry, Constraints and Computation of the Trifocal Tensor.* PhD thesis, Technical University of Vienna, 2003. Online version:
http://www.ipf.tuwien.ac.at/phdtheses/car/diss_car.pdf

[14] A. Shokurov , A. Khropov, and D. Ivanov. *Feature Tracking in Images and Video*. GraphiCon-2003 Proc., pp. 177-179, 2003.

[15] M. E. Spetsakis and Y. Aloimonos. *A unified theory of structure from motion*. In Proceedings of a Workshop held in Pittsburgh, Pennsylvania, Sept.11-13,1990, pp. 271–283.

[16] P. Sturm and W. Triggs. *A factorization-based algorithm for multi-image projective structure and motion*. In Proc. 4th European Confernce on Computer Vision, Cambridge, pp. 709-720, 1996.

[17] C. Tomasi and T. Kanade. *Shape and motion from image streams under orthography: A factorization-based approach*. International Journal of Computer Vision, 9(2):137-154, 1992.



Projective reconstruction

Metric reconstruction