On-line structure from motion

Martin Bujňák*
Faculty of Mathematics, Physics and Informatics
Comenius University
Bratislava / Slovakia

Abstract

In this paper we present on-line structure from motion reconstruction. Input to our algorithm is un-calibrated images from video sequence. We assume that images have zero-skew, principal point is at image center and aspect ratio is equal to 1. We are able to detect and correct small radial lens distortion of camera using fast iterative method. Camera focal length can vary across sequence. With these assumptions we are able to self-calibrate cameras using input image sequence and restrict space ambiguity from projective to metric one. In our approach we use linear self-calibration method.

Keywords: Visual modeling, Structure-from-Motion, Projective reconstruction, Self-calibration, Radial lens distortion

1 Introduction

In many nowadays applications such architectural visualization, cultural inheritance, medicine, movie and computer games industry it is required to acquire high detail photo-realistic 3D model/representation of some real objects. There exists several methods how to create virtual copy of existing real object – from modeling by artists to laser scanning.

Obtaining sparse scene reconstruction (structure) and camera positions (motion) is almost last step before dense 3D reconstruction. Sparse scene can be used as reference points for adding new 3D object to scene. With reprojecting new model to known cameras we can easily add virtual objects to existing video sequence.

We take a closer look to the one of the most accessible and cheapest way of obtaining motion and structure – using video sequences. With such methods user can freely move camera around an object or scene and record video. From this video we are able to reconstruct motion of the camera and sparse scene reconstruction. Neither camera position, nor camera setting has to be known a priori.

Our approach tracks point features across video sequence. From tracked features we create two-view geometry using robust algorithm. Then multiple view structure and motion is created. Every change of structure must kept sparse scene consistent. It means that all re-projected 3D points must be lying on its 2D corresponding feature (practically due to discrete space and noise we want to achieve minimal quadratic error – further in text we will call these "error aspects"). If mean error exceeds threshold we perform non-linear minimization. Using self-calibration we restrict space ambiguity from projective to metric.

In the past years many similar approaches has been proposed. The most similar - in way of how input and output are defined - are approaches of Marc Pollefeys[1], and Kanade et. al[2]. The main difference against these works is that we process input on-line and thus we can create model directly from camera stream. Sequential approach has been also proposed by [15], but unlike this algorithm, our does not require quasi-Euclidian initialization. Pollefeys uses stratified approach so as we do and Kanade uses perspective factorization method. There exist many other methods where more assumptions or space markers are required. Detailed description can be found in [3].

In this paper we describe our sequential stratified approach in three sections - feature tracking, structure and motion, self calibration. Each section contains overview and comparison of already proposed methods. In our work we target to feature tracking with our modification of guided matching, radial lens distortion removing and two-view to multiple-view merging. Other intermediate processes are described with smaller detail with references to complete description so that reader will get complete look to the problem. The paper is concluded with Results and Conclusion with future work.

Authors note : If we could effectively find global minima of following expression

$$\sum_{i=0}^{m}\sum_{j=0}^{n} d(m_{ij}, P_i X_j),$$

where m_{ij} is known 2D re-projection of unknown 3D point X_j by unknown camera P_i and $d(x, y)$ is distance of two 2D points, then all efforts behind reconstruction would be futile. Therefore all the effort is being put into finding initial condition for numerical minimization methods minimizing this formula.

* martinb@DataExpert.sk

2 Feature tracking

We define feature point as point that can be differentiated from its neighbouring points. Feature matching plays key role for most of the photo/video based modelling tools. We transformed this problem in computer vision to simpler problem of feature tracking with adding assumption on maximal motion between two input frames.
There already exists robust commercial feature tracking package like described in [4]. For our work we have selected free KTL feature tracking toolkit [5] and developed new feature tracker similar to KTL.

2.1 Harris based feature tracker

Ours feature tracker uses Harris point feature detector [6]. We detect feature points on two neighbouring images from sequence. Similarly as in KTL we select features that are good to track. As we assume small motion between neighbouring images, we can remove all features that can be miss-exchanged with its neighbouring feature points in the same image – we will refer to this as self-correlation (see Figure 1 left).

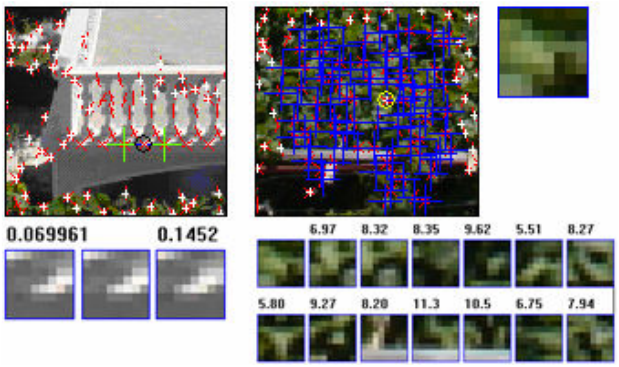

Figure 1 : left – self correlating feature, left bottom – correlation to two self correlating features, right – good feature to track, right bottom – neighbours correlations.

Each feature point is extended by its orientation. We defined orientation of feature point by two principal axes of covariance matrix formed from small region surrounding feature point. (Figure 1, top-left).
We also remove the features that have one principal component much bigger than second one. This is typical for edges. Due to noise in image and discrete sampling these features are not stable when we rotate image (see Figure 2).
Features left in two images are than matched using zero-mean normalized cross-correlation (ZNCC). We modified ZNCC to take in account feature orientation. This is done by changing coordinate system to polar coordinate system.

Figure 2 : Principal components marked by red. The only good feature is marked by green colour.

2.2 Removing outliers

From both KTL and our feature matching algorithms, we get well correlating feature pairs. In real images we noticed that these pairs sometimes do not point to the same object in the scene (see Figure 3). Such feature pairs (further in the text outliers, similarly good correspondences are called inliers) have to be removed as in next stages they may cause numerical errors.

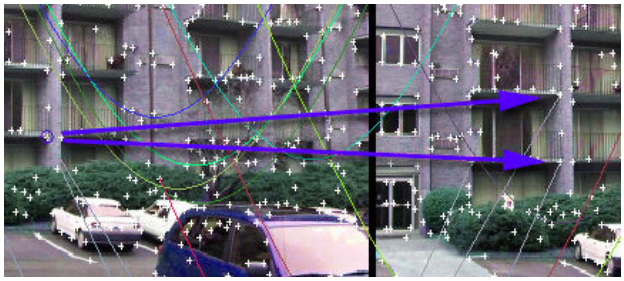

Figure 3 : Feature match perfectly to more features in second image.

In our approach we use RANSAC paradigm [7] to find two-view geometry. We use epipolar geometry described by fundamental matrix. All inliers will satisfy epipolar constrain

$$mFm' = 0, \qquad (1)$$

where m and m' are two corresponding feature points, and F is fundamental matrix.
Using this constraint we eliminate almost all outliners.

2.3 Finding more features

After two view geometry is obtained we can perform guided searching. Fundamental matrix restricts searching region for each point in first image to line in second image (see Figure 4).

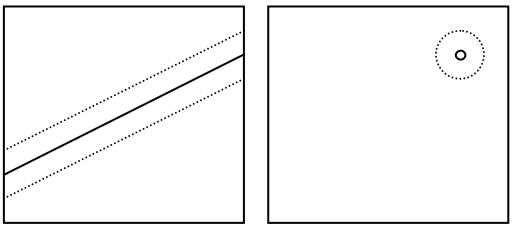

Figure 4 : Corresponding point to the point in right image must lay on (or due to noise lay near) the line in the left image.

Using this we can effectively find new features and matches.
In our approach we find epipolar lines for each feature. Guided matching is performed on these lines with taking ordering constraint into account. Two features are correlated only if length of epilolar line segment from previous feature point is similar to length on corresponding epiloar line (see Figure 5).

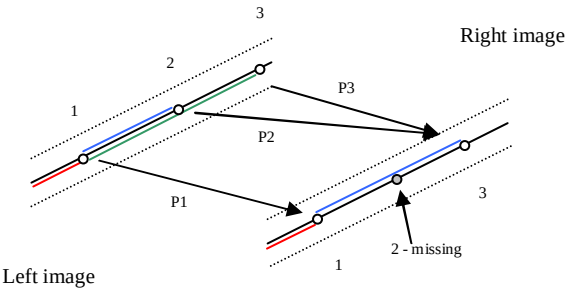

Figure 5 : Pair P1 1-1 will be added if correlation exceeds threshold. Pairs 1-3, 2-1 2-3 will not be tested because lengths of line segments are too different. If 2-2(If not missing) is used, 3-1 would not be tested due to ordering criterion.

From new matches we calculate new fundamental matrix using normalized 8point algorithm [8]. Normalized 8 point algorithm uses linear least-square error methods to find fundamental matrix. This method does not perfectly distribute error as pointed in [8]. Therefore we use result from 8point algorithm as an initial estimation for nonlinear numerical minimization of

$$\text{cost}(F) = \sum_i d(m_i, m_{it})^2 + d(m_i', m_{it}')^2,$$

where $m_{it} F m_{it}' = 0$, m_i and m'_i are i-th corresponding feature pair and $d(x, y)$ is distance of two 2D points.

We minimize cost function using sparse Levenberg-Marquard [9] algorithm.

3 Structure and Motion

From previous step we have pairs of matching features and also two-view geometry of last and new frame from image sequence. Note that we add frames sequentially. From relation between the views and feature correspondence we will create structure of the scene and motion of the camera.
Approach proposed here is similar to [1]. Unlike [1] we don't require to search for initial frame. All measurements are carried out in image space so that we can stay in projective space too. Structure and motion is built sequentially by merging camera pair with previous structure using common features. There must be enough – at least 4 – common feature points. Introducing image based measurements allows us to measure amount of motion parallax between image pairs. During merging step we take this motion to assign weight to common feature points.

3.1 Camera pair

3.1.1 Quasi-calibrated camera pair

Using epipolar geometry - known from previous step - we create projective camera pair. Canonical pair is defined as follows

$$P_1 = [I_{3x3} \mid 0_3],$$
$$P_2 = \left[[e_{12}]_x F_{12} + e_{12} a^T \mid o e_3 \right],$$

where F_{12} is fundamental matrix from image 1 to image 2, e_{12} is epipole. Note that o and a are free parameters and changing them will let epipolar geometry of camera pair unchanged [3]. o determines the global scale of reconstruction and a position of reference plane. Thus o can be simply set to one. In our approach we find a so that camera P_2 will hold calibration condition - zero skew, principal point at image centre and varying focal length. Note that we need at least 3 cameras to perform full calibration with our input assumptions. Therefore further in the text we will call this camera pair as quasi-calibrated.

3.1.2 Sparse scene from camera pair

Having projective matrices we are able to calculate 3D position for each feature pair. Usually, this is done using triangulation. Due to noise and discrete image space, sight lines may not intersect perfectly. We calculate 3D point so that distance between reprojected 3D point and matching 2D point is minimal:

$$d(m, P_1M)^2 + d(m', P_2M)^2,$$

where *m*, *m'* are corresponding 2D feature points, both corresponding to *M*, and P_1, P_2 are camera matrices pair and *d(x, y)* is distance of two 2D points.
Many methods how to obtain optimal 3D position are proposed in [3]. In our work we statistically find 3D position *M* by minimizing following formula:

$$\text{cost}(M) = \sum_i \left\| \overrightarrow{A_i m} M \right\|,$$

Sum members are distances of unknown point M and sight lines (A_i is camera centre, *m* point in 3D on projection plane). Such point can be computed using least squares method. If reprojected 3D point is to far from its 2D match, the feature is considered to be outlier.

3.1.3 Small motion – precision issues

Sometimes we are not able to calculate projection matrices. This occurs when no-motion was made, or virtual parallax occurred. In that case the epipolar geometry (fundamental matrix) is poorly estimated. To detect this in early phases of algorithm we perform 2D measurements and skip frame if median length of motion vectors is smaller than threshold. Virtual parallax caused by pure rotation around axis passing through focal point combined with pure zooming can be detected by thresholding fundamental matrix eigen values.
Even if fundamental matrix is well defined, discrete space and noise can cause too big freedom for placing 3D point (see Figure 6). The error can be enormous when camera motion is small. For that case we introduce image based measure (weight) for each 3D feature saying, how precisely we estimated 3D point (volume of intersection of sight lines). Similarly, if median weight is smaller than threshold we skip frame.

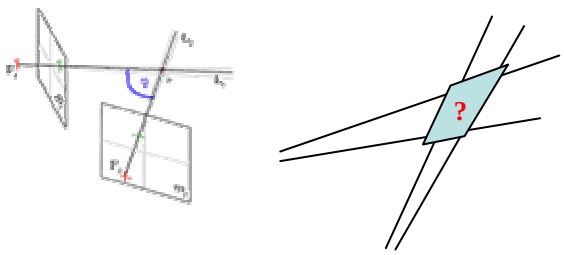

Figure 6 : Volume of intersection of sight lines grows with smaller angle between sight lines. Placing 3D point anywhere inside this volume will be perfect match.

Note that photo-consistent 3D point lays in intersection of all sight lines from all cameras from which the 3D point is visible.
Also note that skipped frames are hold in memory for feature tracking purposes.

3.1.4 Radial lens distortion

Optical distortion of camera lens can move 2D points from original position far away – even more than 10 pixels. In our work we take into account only radial lens distortion that can be approximated as

$$\lambda = (1 + \kappa(x^2 + y^2)),$$
$$x_1 = \lambda x,$$
$$y_1 = \lambda y,$$

where κ is unknown distortion factor.
Our algorithm is similar to [9]. In [9] authors modified distortion equation to $1/\lambda$ and this allowed them to modify linear algorithm for calculating fundamental matrix to calculate κ too. New equation for *F* matrix returns more roots (around 10) and all must be tested. Also authors comment that change of radial distortion equation creates many local minima around global minima.
Our radial lens distortion removal algorithm search for κ directly by minimizing

$$\sum_i d(m_i, Fm_i')^2 + d(F^T m_i, m_i')^2,$$

where m_i, m_i' are corresponding 2D feature points, *F* is fundamental matrix and *d(x, y)* is distance of line and 2D point. For each κ we un-distort features position and find new fundamental matrix. Features are scaled to fit window. κ is found using stimulate annealing [17].
We assume that κ is equal for two neighbouring cameras. κ for *i*-th camera is approximated by averaging κ obtained from both pairs (*i-1*th and *i+1*th cameras). For small motion this algorithm does not find good κ.

3.2 Updating the structure and motion

In this section we describe how camera pairs are merged together with existing reconstruction. Our algorithm merges new camera pair with existing structure and motion using their common camera. Due to noise, estimation errors, amount of outliners and so on, error will accumulate and after few camera are added the resulting scene will be inaccurate. In our approach we calculate re-projection error of 3D space on each camera. If mean error exceeds threshold value, than we perform nonlinear minimization - bundle adjustment [11]. Taking

in account sparsity of the problem, this problem can be solved effectively [12].

3.2.1 Merging camera pairs

For merging process we have new camera pair P_1, P_2 - in canonical form, and existing reconstruction. Let P be last camera in existing motion structure. P corresponds to P_1 camera in new pair. Merging pairs means to transform both P_1 and P_2, so that P_1 will be equal to P. After that P_2 will not be correctly placed as P_2 can differ in position of reference plane and scale factor (see Figure 7).

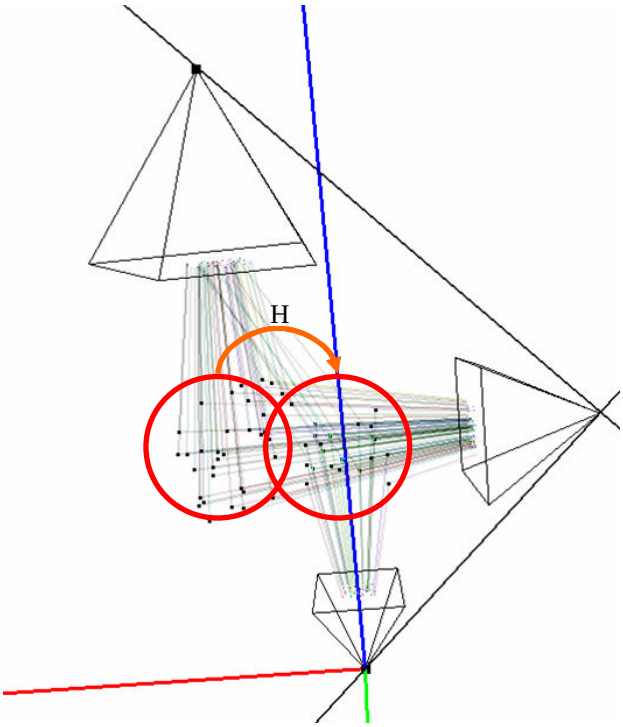

Figure 7 : Merging new pair and existing space using common features.

In ideal conditions we can express homography transformation that will fix P_2 camera from known 3D space and common correspondence as

$$Y = HX,$$

where X and Y are corresponding 3D points in new pair 3D structure and existing structure.
From four 3D points we are able to calculate H. Practically due to error aspects we need robust approach as not all 3D points are suitable for calculating such homography. In ours approach we select bundle of those points that have biggest weight (see Small motion – precision issues). From these 3D points we calculate homography using RANSAC paradigm [7]. Homography is calculated so that error is measured in 2D.

After transforming P_2 with H we merge 3D structure of new pair into existing structure. Already known 3D points are merged with new points and are recalculated as described in section 3.1.2.
To minimize accumulated error we calculate mean 3D to 2D re-projection error. If this error exceeds threshold value we perform bundle adjustment.

4 Self-calibration

Until now we did not care about camera parameters. Reconstructed space and camera poses are locked by photo consistency constraints (reprojection error is small) but such reconstruction is not unique. Now we will detect camera intrinsic parameters using only images - this is called *self-calibration*. Many technics how to perform self-calibration of cameras are described in [3].
Let X by any 3D point of reconstruction, P any camera and m corresponding 2D feature in camera P. For any homography H_{4x4} we get:

$$\begin{aligned} m &= PX, \\ m &= PHH^{-1}X. \end{aligned} \qquad (2)$$

It means that we can transform both cameras and 3D points so that reprojection error will stay unchanged. Without loss of generality consider that H does not rotate, translate and scale. These components are interesting only if we want to align reconstruction to some existing space. Now, the only component that we will care about is projective part of the homography - the only part that can transform plane at infinity. Such homography can be described as follows:

$$H = \begin{pmatrix} k^{-1} & 0 & 0 & 0 \\ 0 & k^{-1} & 0 & 0 \\ 0 & 0 & k^{-1} & 0 \\ ak^{-1} & bk^{-1} & ck^{-1} & k \end{pmatrix}, \qquad (3)$$

where a, b, c, k are unknowns.
Camera matrix can be factorized into upper triangular 3x3 calibration matrix (4), 3x3 rotation matrix and 3x1 translation matrix.

$$K = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad (4)$$

where α_x, α_y - focal length, $\alpha_x : \alpha_y$ aspect ration, s – skew, $[x_0, y_0]$ principal point.
In our approach we find homography H so that all cameras transformed with H will have calibration

matrices as assumed – skew equal zero, principal point at image centre, aspect ration equal 1 and varying focal length. Key of finding such homography lays in projection of *absolute conic*. Absolute conic can be represented using dual absolute quadric [13]. One of the most important properties of absolute quadrics Ω is that they are invariant to similarity transformations. Another property leads to direct key to how to find H: The projection of dual absolute quadric is directly related to the intrinsic camera parameters [1] :

$$KK^T = P\Omega P^T, \qquad (5)$$

where P is 3x4 projection matrix.
Since the images are independent to the projective basis of the reconstruction, equation (5) is always valid and constraints on the intrinsic can be translated to constraints of absolute quadric [1]. With our assumptions (5) can be rewritten into linear system with one cubic constraint [14].
Using H we transform whole structure and cameras as shown in (2). This will cause that re-projection error will stay unchanged and cameras hold 'real' conditions.

5 Results

In our work we aimed on feature tracking with our modification of guided matching, radial lens distortion removing and two-view to multiple-view merging.

Figure 8 : Our tracker – traceable features marked white.

Our feature detector does not differ from KLT detector a lot. Comparing with KLT we introduced feature orientation, perform feature correlation in colored images and have other criterion for selecting features that are good to track. Taking orientation into account caused that our feature tracker does not lose features when camera is rotated. For small camera rotation even KLT will work fine. Because we perform feature correlation on colored images we can track more features if these can be differentiated by color.
Our criterion for selecting traceable features keep main role in scenes like in Figure 8 (compare to Figure 9). Here KLT selects features that are not suitable for tracking and will cause many bad matches. For small motion both KTL and our feature trackers give the same results in comparable time.

Figure 9 : KLT good features – traceable features marked red. Bigger motion results in invalid matching.

Guided matching algorithm finds matches that satisfy epipolar constraint (1). If two matching pairs lay down on their epipolar lines, than epipolar constraint is satisfied even if these features are outliers.
Ordering constraint and length criterion dramatically eliminate number of outliers. Computational complexity stays in worst case $O(nm)$ - for n features in first and m features in second image. New criterions reduced number of expensive cross correlation tests (Figure 10).

Figure 10: Guided matching. Matching is processed on two corresponding epipolar lines.

We tested our radial distortion on grid patterns and real images. We did not aim to find perfect calibration, because global nonlinear minimization corrects radial distortion too. Numerical complexity depends on number of feature pairs because we recalculate fundamental matrix every iteration. Our experience show that linear estimation suffices. For 200 feature points we estimate radial distortion under 1 second on 2.8GHz machine.
For grid pattern the error against ground truth was under 2 pixels at image corners (measured on feature points).

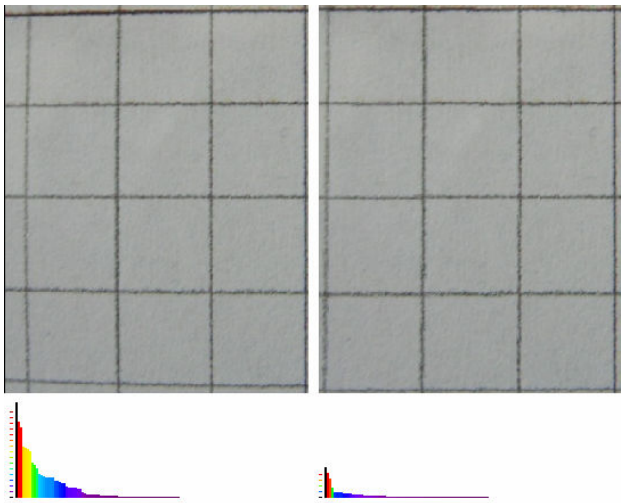

Figure 11: Radial distortion test. Left original, right undistorted. Under images are error graphs.

Ransac paradigm in both two-view and merging to n-view processes makes algorithm robust for presence of outliers. Tests on synthetically generated data showed that algorithm can also deal with 30% amount of outlier (for 100 feature correspondence). Radial lens distortion influence 3D scene, but 3D to 2D re-projection error stays under 1pixel. Adding noise with gaussin distribution to the images will cause problems for linear algorithms. Having more cameras will cause that 3D points are estimated from more 2D correspondences and thus noise is slightly suppressed so that structure does not change dramatically. Because camera projection matrix is calculated from only from using fundamental matrix, numerical minimization on fundamental matrix is in this case essential. For noise with radius 3pixels, cameras were estimated poorly even after numerical minimization. For such case it would be better to calculate projection matrix from 3, 4 or more view image constraints.
Tests on real data give good results even for small resolution camera. Figure 12 shows sparse reconstruction of scene captured by digital camera in resolution 320x240.
Although quasi-calibration of pairs is not required, our experiences show that it helps in merging processes.

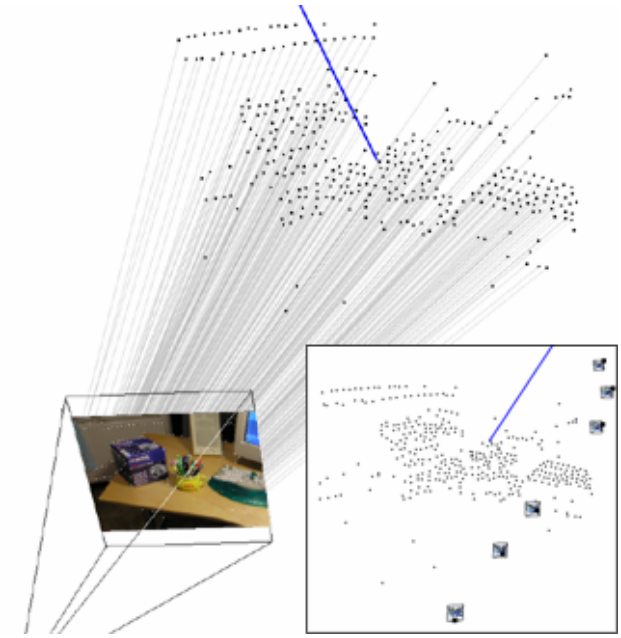

Figure 12 : Sparse reconstruction of the scene.

Merging quasi-calibrated pairs will cause that sparse 3D space is not so distorted by perspective. Such 3D points are near true position and space is more uniformly distributed. We explain better results by better uniformity of distribution of the space.

6 Conclusion and future work

We have presented a sequential stratified approach for creating calibrated motion and structure from un-calibrated video sequence. Sequential processing allows us to process input video directly from camera stream. Biggest advantage of processing from stream is that we can skip process of storing to disk and video compression which leads to better quality (due to uncompressed transfer).
Because there is always noise in the images it is not good to calculate camera from only two views. Therefore we want to improve camera projection matrix calculation, using more images. Our experiences also showed that if principal point is not in image centre, than scene stay skewed even after self-calibration. For cheap hand held cameras it is unexpected to have principal point at image center. Allowing principal point to be constant (non zero) or to be varying leads to non-linear self-calibration algorithm.
As future work we plan to extend algorithm to extract dense scene structure and reconstruct textured 3D mesh.

References

[1] M. Pollefeys - L. Van Gool - M. Vergauwen - F. Verbiest - K. Cornelis - J. Tops - R. Koch. Visual modeling with a hand-held camera, *International Journal of Computer Vision* 59(3), 207-232, 2004.

[2] M. Han - T. Kanade. *Creating 3D Models with Uncalibrated Cameras*, proceeding of IEEE Computer Society Workshop on the Application of Computer Vision (WACV2000), December 2000.

[3] R. Hartley - A. Zisserman. *Multiple View Geometry In Computer Vision.* Second Edition. Cambridge University press, UK. March 2004.

[4] A. W. Fitzgibbon – A. Zisserman. *Automatic Camera Tracking*. Robotics Research Group. Department of Engineering Science. University of Oxford, UK.

[5] Stan Birchfileld. *KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker*. Stanford University, *http://vision.stanford.edu/~birch*

[6] C. Harris -M. Stephens. A combined corner and edge detector, *Fourth Alvey Vision Conference* pp.147-151, 1988.

[7] M. Fischler – R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting. *Communications of the ACM*, 24 (6), 381-395. 1981.

[8] R. Hartley, In defense of the eight-point algorithm. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 19(6):580-593, June 1997.

[9] A. W. Fitzgibbon. *Simultaneous linear estimation of multiple view geometry and lens distortion.* Department of Engineering Science. University of Oxford, UK.

[10] W. Press – S. Teukolsky – W. Vetterling. *Numerical recipes in C : the art of scientific computing*, Cambridge university press, 1992

[11] B. Triggs – P. McLauchlan – R. Hartley – A. Fitzgibbon. Bundle Adjustment - Modern Synthesis, *Vision Algorithms: Theory and Practice*, Springer Verlag, 298-375, 2000.

[12] M.I.A. Lourakis - A.A. Argyros. *The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm*, Institute of Computer Science – FORTH, Heraklion, Crete, Greece, August 2004.

[13] B. Triggs. The absolute Quadric, *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society press, pp. 609-617, 1997.

[14] M. Pollefeys – R. Koch – L. V. Gool. Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Intrinsic Camera Parameters, *International Journal of Computer Vision, KLuwer Academic Publishers, Boston, 1998*

[15] P. Breadsley – A. Zisserman – D. Murray. Sequential Updating of Projective and Affine Structure from Motion. *International Journal of Computer Vision (23), No. 3, Jun-Jul 1997, pp. 235-259.*

[16] M. Pollefeys. *3D Photography*, comp290-89 Fall, University of North Carolina, 2004.

[17] V.Kvasnička – J.Pospíchal – P.Tiňo. *Evolučné algoritmy.* Slovak technical university, Bratislava, 2000