ECCV'04 – In Proceedings of the European Conference on Computer Vision, Prague, Czech Republic, May 2004.

A Framework For Pencil-of-Points Structure-From-Motion

Adrien Bartoli[1,2], Mathieu Coquerelle[2], and Peter Sturm[2]

[1] Department of Engineering Science, University of Oxford, UK
[2] équipe MOVI, INRIA Rhône-Alpes, France
Bartoli@robots.ox.ac.uk, Coquerelle@inria.fr, Sturm@inria.fr

Abstract. Our goal is to match contour lines between images and to recover structure and motion from those. The main difficulty is that pairs of lines from two images do not induce direct geometric constraint on camera motion. Previous work uses geometric attributes — orientation, length, etc. — for single or groups of lines. Our approach is based on using Pencil-of-Points (points on line) or POPs for short. There are many advantages to using POPs for structure-from-motion. The most important one is that, contrarily to pairs of lines, pairs of POPs may constrain camera motion. We give a complete theoretical and practical framework for automatic structure-from-motion using POPs — detection, matching, robust motion estimation, triangulation and bundle adjustment. For wide baseline matching, it has been shown that cross-correlation scores computed on neighbouring patches to the lines gives reliable results, given 2D homographic transformations to compensate for the pose of the patches. When cameras are known, this transformation has a 1-dimensional ambiguity. We show that when cameras are unknown, using POPs lead to a 3-dimensional ambiguity, from which it is still possible to reliably compute cross-correlation. We propose linear and non-linear algorithms for estimating the fundamental matrix and for the multiple-view triangulation of POPs. Experimental results are provided for simulated and real data.

1 Introduction

Recovering structure and motion from images is one of the key goals in computer vision. A common approach is to detect and match image features while recovering camera motion. The goal of this paper is the automatic matching of lines and recovery of structure and motion. This problem is difficult for the reason that a pair of corresponding lines does not give direct geometric constraint on the camera motion. Hence, one has to work on a three-view basis or assume that camera motion is known a priori, e.g. [10].

In this paper, we attack directly the two view case by introducing a type of image primitive that we call *Pencil-of-Points* or POP for short. A POP is made of a *supporting line* and a set of *supporting points* lying on the supporting line. Physically, a POP corresponds to a set of interest points on a contour line. POPs can be built on the top of most contour lines. Contrarily to pairs of corresponding lines, pairs of corresponding POPs may give geometric constraints on camera motion, provided that what we call the *local geometry*, relating corresponding points along the supporting lines, has been computed. We exploit these geometric constraints for matching POPs and recovering structure and motion. Once camera motion has been recovered using POPs, it can be employed for a reliable guided-matching and reconstruction of other types of features.

The closest work to ours is [10]. The main difference is that the authors consider that the cameras are known and propose a wide-baseline guided-matching algorithm for lines. They show that reliable results are obtained based on cross-correlation scores, computed by warping the neighbouring textures of the lines using the 2D homography $\mathsf{H}(\mu) \sim [\mathbf{l}']_\times \mathsf{F} + \mu \mathbf{e}' \mathbf{l}^\mathsf{T}$, where $\mathbf{l} \leftrightarrow \mathbf{l}'$ are corresponding lines, F is the fundamental matrix and $\mathbf{e}'$ the second epipole. The projective parameter μ is computed by minimizing the cross-correlation score.

Before going into further details about our approach, we underline some of the advantages of using POPs for automatic structure and motion recovery. First, a POP has fewer degrees of freedom than the supporting line and the individual supporting points which implies that (*i*) its localization is often more accurate that those of the individual features, (*ii*) finding POPs in a set of interest points and contour lines increase their individual repeatability rate and (*iii*) structure and motion parameters estimated from POPs are more accurate than that recovered from points and/or lines. Second, matching or tracking POPs through images is more reliable than individual contour lines or interest points, since a pair of corresponding POPs defines a local geometry, used to score matching hypotheses based on geometric or photometric criteria. Third, the robust estimation of camera motion based on random sampling from putative correspondences, i.e. in a RANSAC-like manner [3], is more efficient using POPs than other standard features, since only three pairs of POPs define a fundamental matrix, versus seven pairs of points.

Contributions and paper organization. Using POPs for structure-from-motion is a new concept. We propose a comprehensive framework for multiple-view matching and recovery of structure and motion. Our framework is based on the following traditional steps, which also give the organization of this paper.

First, §2, we investigate the detection of POPs in images and their matching. We define and study the *local geometry* of a pair of POPs. We propose methods for its estimation, which allow to obtain putative POP correspondences, from which the epipolar geometry can be robustly estimated.

Second, §3, we propose techniques for estimating the epipolar geometry from POP correspondences. Minimal and redundent cases are studied.

Third, §4, we tackle the problem of triangulating POPs from multiple images. We derive and approximate the optimal (in the Maximum Likelihood sens) solution by an algorithm based on the triangulation of the supporting line, then the supporting points.

Finally, bundle adjustment is described in §5. We provide experimental results on simulated data and give our conclusions and further work in §§6 and 7 respectively. Experimental results on real data are provided throughout the paper. The following two paragraphs give our notation, some preliminaries and definitions.

Notation and preliminaries. We make no formal distinction between coordinate vectors and physical entities. Equality up to a non-null scale factor is denoted by $\sim$. Vectors are typeset using bold font ($\mathbf{q}$, $\mathbf{Q}$), matrices using sans-serif fonts (F, H) and scalars in italic (α). Transposition and transposed inverse are denoted by $^\mathsf{T}$ and $^{-\mathsf{T}}$. The (3×3) skew-symmetric cross-product matrix is written as in $[\mathbf{q}]_\times \mathbf{x} = \mathbf{q} \times \mathbf{x}$. Indices are used to indicate the size of a matrix or vector ($\mathsf{F}_{(3\times3)}$, $\mathbf{q}_{(3\times1)}$), to index a set of entities ($\mathbf{q}_i$) or to select coefficients of matrices or vectors (q_1, $q_{i,1}$). Index i is used for the n images, j for the m features and k for the p supporting points of a POP[3]. The supporting lines are written $\mathbf{l}_{ij}$ (the supporting line of the j-th POP in image i) and supporting points as $\mathbf{q}_{ijk}$ (the k-th supporting point of the j-th POP in image i). Indices are sometimes dropped for clarity. The identity matrix is written I and the null-vector as $\mathbf{0}$. We use the Euclidean distance between points, denoted d_e and an algebraic distance defined by:

$$d_a^2(\mathbf{q}, \mathbf{u}) = \|\mathsf{S}[\mathbf{q}]_\times \mathbf{u}\|^2 \quad \text{with} \quad \mathsf{S} = \left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right). \tag{1}$$

Definitions. A pencil of points is a set of p supporting points lying on a supporting line. If $p \geq 3$, the POP is said to be *complete*, otherwise, it is said to be *incomplete*. A *complete correspondence* is a correspondence of complete POPs. As shown in the next section, only complete correspondences may define a local geometry.

We distinguish two kinds of correspondences of POPs: *line-level* and *point-level* correspondences. A line-level correspondence means that only the supporting lines are known to match. A point-level correspondence is stronger and means that a point-to-point mapping along the supporting lines has been established.

2 Detecting and Matching Pencil-of-Points

2.1 Detecting

Detecting POPs in images is the first step of the structure-from-motion process. One of the most important properties of a detector is its ability to achieve repeatability rates[4] as high as possible, which reflects the fact that it can detect the same features in different images. In order to ensure high repeatability rates, we formulate our POP detector based on interest points and contour lines, for which there exist detectors achieving high repeatability rates, see [9] for interest points and [2] for contour lines.

In order to detect salient POPs, we merge nearby contour lines. Algorithms based on the Hough transform or RANSAC [3] can be used to detect POPs within a set of points and/or lines. We propose the following simple solution. First, an empty POP is instanciated for each line (which gives the supporting line). Second, each point is attached to the POPs whose supporting line is at a distance lower than a threshold, that we typically choose as a few pixels. Finally, incomplete POPs, i.e. those for which the number of supporting points is less than three, are

[3] To simplify the notation, we assume without loss of generality that all POPs have the same number of supporting points.

[4] The repeatability rate between two images is the number of corresponding features over the mean number of detected points [9].

eliminated. Note that we use a loose threshold for interest point and contour line detection, to get as many as possible POPs. The less significant interest points and contour lines are generally pruned as they are respectively not attached to any POP or form incomplete POPs. An example of POP detection is shown on figures 1 (a) & (b). It is observed that the repeatability rate of POPs is higher than each of the repeatability rates of points and lines.

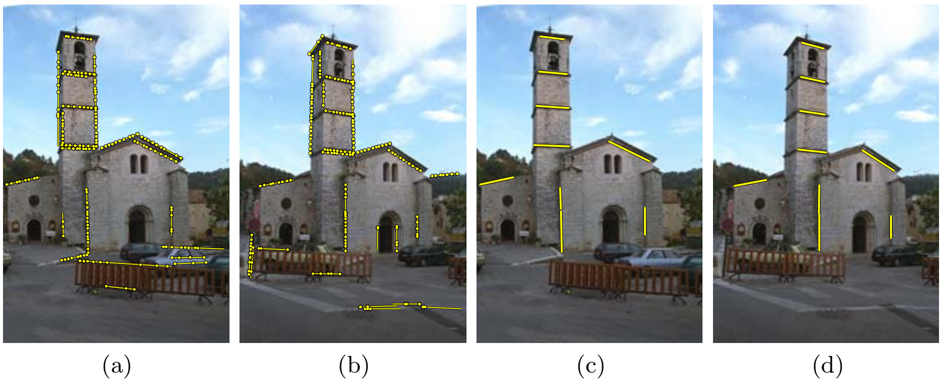


Fig. 1. (a) & (b) show the detected POPs. The repeatability rate is 51% while for points and lines it is lower, respectively 41% and 37%. (c) & (d) show the 9 putative matches obtained with our algorithm. On this example, all of them are correct, which shows the robustness of our local geometry based cross-correlation measure.

2.2 Matching

Traditional structure-from-motion algorithms using interest points usually rely on an initial matching, followed by the robust estimation of camera geometry and a guided-matching step, see e.g. [6]. The initial matching step is often based on similarity measures between points such as correlation or grey-value invariants. Guided-matching uses the estimated camera geometry to constrain the search-area. In the case of POPs, the initial matching step is based on the local geometry defined by a pair of POPs. This step is described below followed by the robust estimation of the epipolar geometry.

Matching Based on Local Geometry As mentioned above, the idea is to use the local geometry defined by a pair of POPs. We show that this local geometry is modeled by a 1D homography and allows to establish dense correspondences between the two supporting lines. Given a hypothesized line-level POP correspondence, we upgrade it to point-level by computing its local geometry. Given a point-level correspondence, a similarity score can be computed using cross-correlation, in a manner similar to [10]. For each POP in one image, the score is computed for all POPs in the other image and a 'winner takes all' scheme is employed to extract a set of putative POP matches. Putative matches obtained by our algorithm are shown on figures 1 (c) & (d).

Defining and computing the local geometry. We study the local geometry induced by a point-level correspondence, and propose an estimation method.

Proposition 1. *Corresponding supporting points are linked by a 1D homography, related to the* epipolar transformation*, relating corresponding epipolar lines.*

Proof: Corresponding supporting points lie on corresponding epipolar lines: there is a trivial one-to-one correspondence between supporting points and epipolar lines (provided the supporting lines do not contain the epipoles). The proof follows from the fact that the epipolar pencils are related by a 1D homography [12]. ■

First, we shall define a local $\mathbb{P}^1$ parameterization of the supporting points, using two Euclidean transformation matrices A and A' acting such that the supporting lines are rotated to be vertical and aligned with the y-axes of the images. The transformed supporting points are $\mathbf{x}_k \sim \mathsf{A}\mathbf{q}_k \sim (0 \;\; y_k \;\; 1)^\mathsf{T}$ and $\mathbf{x}'_k \sim \mathsf{A}'\mathbf{q}'_k \sim (0 \;\; y'_k \;\; 1)^\mathsf{T}$. Second, we introduce a 1D homography g as:

$$\begin{pmatrix} y'_k \\ 1 \end{pmatrix} \sim \mathsf{g} \begin{pmatrix} y_k \\ 1 \end{pmatrix} \quad \text{with} \quad \mathsf{g} \sim \begin{pmatrix} g_1 & g_2 \\ g_3 & 1 \end{pmatrix}, \tag{2}$$

which is equivalent to $\mathbf{x}' \sim \mathsf{G}(\boldsymbol{\mu})\mathbf{x}$ with $\mathsf{G}(\boldsymbol{\mu}) \sim \left(\begin{smallmatrix} \mu_1 & 0 & 0 \\ \mu_2 & g_1 & g_2 \\ \mu_3 & g_3 & 1 \end{smallmatrix}\right)$, where the 3-vector $\boldsymbol{\mu}^\mathsf{T} \sim (\mu_1 \;\; \mu_2 \;\; \mu_3)$ represents projective parameters which are significant only when $\mathsf{G}(\boldsymbol{\mu})$ is applied to points off the supporting line. The 2D homography mapping corresponding points along the supporting lines is $\mathsf{H}(\boldsymbol{\mu}) \sim \mathsf{A}'^{-1}\mathsf{G}(\boldsymbol{\mu})\mathsf{A}$.

The 1D homography g can be estimated from $p \geq 3$ pairs of supporting points using equation (2). This is the reason why complete POPs are defined as those which have at least 3 supporting points. Given g, $\mathsf{H}(\boldsymbol{\mu})$ can be formed.

Computing $\mathsf{H}(\boldsymbol{\mu})$*.* The above-described algorithm can not be applied directly since at this stage, we only have line-level POP correspondence hypotheses. We have to upgrade them to point-level to estimate $\mathsf{H}(\boldsymbol{\mu})$ with the previously-given algorithm and score them by computing cross-correlation. We propose the following algorithm:

- for all valid pairs of triplets of supporting points[5]:
  - compute the local geometry represented by $\mathsf{H}(\boldsymbol{\mu})$.
  - compute the cross-correlation score based on $\mathsf{H}(\boldsymbol{\mu})$, see below.
- return the $\mathsf{H}(\boldsymbol{\mu})$ corresponding to the highest cross-correlation score.

Computing cross-correlation. For a pair of POPs, the matching score is obtained by evaluating the cross-correlation using $\mathsf{H}(\boldsymbol{\mu})$ to associate corresponding points. The cross-correlation is evaluated within rectangular strips centered onto the supporting lines. The length of the strips are given by the overlap of the supporting lines in each image. The width of the strips must be sufficiently large for cross-correlation to be discriminative. During our experiments, we found that a width of

[5] Valid triplets satisfy an ordering constraint, namely middle points have to match.

3 to 7 pixels was appropriate. For pixels off the supporting lines, the $\boldsymbol{\mu}$ parameters are significant. The following solutions are possible: compute these parameters by minimizing the cross-correlation score, as in [10], or use the median luminance and chrominance of the regions adjacent to the supporting lines [1]. The first solution is computationally too expensive to be used in our inner loop, since 3 parameters have to be estimated, while the second solution is not discriminative enough. We propose to map pixels along lines perpendicular to the supporting lines. Hence, the method uses neighbouring texture while being independent of $\boldsymbol{\mu}$. In order to take into account a possible non-planarity surrounding the supporting lines, we weight the contribution of each pixel to cross-correlation proportionally to the inverse of its distance to the supporting line.

Robustly Computing the Epipolar Geometry At this stage, we are given a set of putative POP correspondences. We employ a robust estimator, allowing to estimate the epipolar geometry and to discriminate between inliers and outliers. We use a scheme based on RANSAC [3], which maximizes the number of inliers. In order to use RANSAC, one must provide a *minimal estimator*, i.e. an estimator which computes the epipolar geometry from the minimum number of correspondences, and a function to discriminate between inliers and outliers, given an hypothesized epipolar geometry. The number of trials required to ensure a good probability of success, say 0.99, depends on the minimal number of correspondences needed to compute the epipolar geometry. Our minimal estimator described in §3 needs 3 pairs of POPs. Applying a RANSAC procedure is therefore much more efficient with POPs than with points: with 50% of outliers, 35 trials are sufficient with POPs, while 588 trials are required for points (values taken from [6]).

Our inlier/outlier discriminating function is based on computing the cross-correlation score using [10]. Inliers are selected by thresholding this score. We use a threshold of few percents (2% — 5%) of the maximal grey value. Figures 2 (a-d) show an example of epipolar geometry computation, and the set of corresponding POPs obtained after guided-matching based on the method of [10].

3 Computing the Epipolar Geometry

Proposition 2. *The minimal number of pairs of* POP*s in general position*[6] *needed to define a unique fundamental matrix is 3.*

Proof: Due to lack of space, this proof is left for an extended version of the paper.

3.1 The 'Eight Corrected Point' Algorithm

This linear estimator is based on the constraints induced by the supporting points. Pairs of supporting points $\mathbf{q}_{jk} \leftrightarrow \mathbf{q}'_{jk}$ are obtained based on the previously estimated local geometries $\mathsf{H}(\boldsymbol{\mu})$. The first idea that comes to mind is to use the supporting points as input to the eight point algorithm [7]. This algorithm minimizes an algebraic distance between predicted epipolar lines and observed points. The

[6] General position means that the supporting lines are not coplanar and do not lie on an epipolar plane, i.e. the image lines do not contain the epipoles.

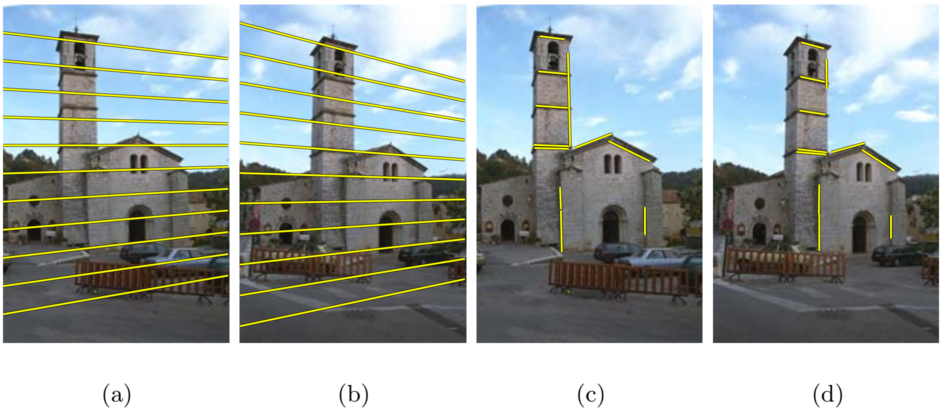

(a) (b) (c) (d)

Fig. 2. (a) & (b) show a representative set of corresponding epipolar lines while (c) & (d) show the 11 matched lines obtained after guided-matching using the algorithm of [10].

eight corrected point algorithm consists in correcting the position of the supporting points, i.e. to make them colinear, prior to applying the eight point algorithm. Using this procedure reduces the noise on the points positions, as we shall verify experimentally.

3.2 The 'Three POP' Algorithm

This linear algorithm compares observed points and predicted points. This algorithm is more statistically meaningful than the eight point algorithm, in the case of POPs, in that observed and predicted features are directly compared.

We wish to predict the supporting point positions. We intersect the predicted epipolar lines, i.e. $\mathsf{F}\mathbf{q}_{jk}$ in the second image, with the supporting lines $\mathbf{l}'_j$: the predicted point is given by $[\mathbf{l}'_j]_\times \mathsf{F}\mathbf{q}_{jk}$. Our cost function is given by summing the squared algebraic distances between observed and predicted points: $\sum_j d_a^2(\mathbf{q}'_{jk}, [\mathbf{l}'_j]_\times \mathsf{F}\mathbf{q}_{jk})$. In order to obtain a symmetric criterion, we consider predicted and observed points in the first image also, which yields:

$$\mathcal{C}_a = \sum_j \sum_k \left(d_a^2(\mathbf{q}_{jk}, [\mathbf{l}_j]_\times \mathsf{F}^\mathsf{T} \mathbf{q}'_{jk}) + d_a^2(\mathbf{q}'_{jk}, [\mathbf{l}'_j]_\times \mathsf{F}\mathbf{q}_{jk}) \right) . \tag{3}$$

After introducing explicitly d_a from equation (1) and minor algebraic manipulations, we obtain the matrix form $\mathcal{C}_a = \sum_j \sum_k (\|\mathsf{B}_{jk}\mathbf{f}\|^2 + \|\mathsf{B}'_{jk}\mathbf{f}\|^2)$ where $\mathbf{f} = \mathrm{vect}(\mathsf{F})$ is the row-wise vectorization of F and:

$$\mathsf{B}_{jk} = \mathsf{S}[\mathbf{q}_{jk}]_\times [\mathbf{l}_j]_\times \left(q'_{jk,1}\mathrm{I} \;\; q'_{jk,2}\mathrm{I} \;\; q'_{jk,3}\mathrm{I} \right) , \mathsf{B}'_{jk} = \mathsf{S}[\mathbf{q}'_{jk}]_\times [\mathbf{l}'_j]_\times \mathrm{diag}(\mathbf{q}_{jk}^\mathsf{T} \;\; \mathbf{q}_{jk}^\mathsf{T} \;\; \mathbf{q}_{jk}^\mathsf{T}) .$$

The cost function becomes $\mathcal{C}_a = \|\mathsf{B}\mathbf{f}\|^2$ with $\mathsf{B}^\mathsf{T} \sim \left(\mathsf{B}_{11}^\mathsf{T} \;\; {\mathsf{B}'_{11}}^\mathsf{T} \;\; \dots \;\; \mathsf{B}_{mp}^\mathsf{T} \;\; {\mathsf{B}'_{mp}}^\mathsf{T} \right)$. The singular vector associated to the smallest singular value of B gives the $\mathbf{f}$ that minimizes $\mathcal{C}_a$. Similarly to the eight point algorithm, the obtained fundamental matrix does not satisfy the rank-deficiency constraint in general, and has to be corrected by nullifying its smallest singular value, see e.g. [6].

3.3 Non-Linear 'Reduced' Estimation

The previously-described three POP estimator is statistically sound in the sense that observed and predicted points are compared in the linear cost function (3). However, the comparison is done using the algebraic distance d_a. This is the price to pay to get a linear estimator. In this section, we consider a cost function with a similar form, but using the Euclidean distance d_e to compare observed and predicted points:

$$\mathcal{C}_e = \sum_j \sum_k \left(d_e^2(\mathbf{q}_{jk}, [\mathbf{l}_j]_\times \mathsf{F}^\mathsf{T} \mathbf{q}'_{jk}) + d_e^2(\mathbf{q}'_{jk}, [\mathbf{l}'_j]_\times \mathsf{F} \mathbf{q}_{jk}) \right) . \tag{4}$$

We use the Levenberg-Marquart algorithm, see e.g. [6], with a suitable parameterization of the fundamental matrix [12] to minimize this cost function, based on the initial solution provided by the three POP algorithm.

4 Multiple-View Triangulation

We deal with the triangulation of POP seen in multiple views. Note that since the triangulation of a line is independent from the others, we drop the index j in this section.

4.1 Optimal Triangulation

The optimal 3D POP is the one which better explains the data, i.e. which minimizes the sum of squared Euclidean distances between predicted and observed supporting points. Assuming that 3D POPs are represented by two points $\mathbf{M}$ and $\mathbf{N}$ for the supporting line and p scalars α_k for the supporting points $\mathbf{Q}_k \sim \alpha_k \mathbf{M} + (1 - \alpha_k)\mathbf{N}$, the following non-linear problem is obtained:

$$\min_{\mathbf{M},\mathbf{N},\ldots,\alpha_k,\ldots} \mathcal{C}_{pop} \text{ with } \mathcal{C}_{pop} = \sum_{i=1}^{n} \sum_{k=1}^{p} d_e^2(\mathsf{P}_i(\alpha_k \mathbf{M} + (1 - \alpha_k)\mathbf{N}), \mathbf{q}_{ik}). \tag{5}$$

We use the Levenberg-Marquart algorithm, e.g. [6]. We examine the difficult problem of finding a reliable initial solution in the next section.

4.2 Initialization

Finding an initial solution which is close to the optimal one is of primary importance. The initialization method must minimize a cost function as close as possible to (5). We propose a two-step initialization algorithm consisting in triangulating the supporting line, then each supporting point. Our motivations for these steps are explained while reviewing line triangulation below.

Line Triangulation Line triangulation from multiple views is a standard structure-from-motion problem and has been widely studied, see e.g. [5]. The optimal line $< \mathbf{M}, \mathbf{N} >$ is given by minimizing the sum of squared Euclidean distances between the predicted lines $(\mathsf{P}_i\mathbf{M}) \times (\mathsf{P}_i\mathbf{N})$ and the observed points $\mathbf{q}_{ik}$ as $\min_{\mathbf{M},\mathbf{N}} \sum_{i=1}^{n} \sum_{k=1}^{p} d_e^2((\mathsf{P}_i\mathbf{M}) \times (\mathsf{P}_i\mathbf{N}), \mathbf{q}_{ik})$. To make the relationship with the cost function (5) appear, we introduce a set of points $\mathbf{Q}_{ik}$ on the 3D line. Using the fact that the Euclidean distance between a point and a line is equal to the Euclidean distance between the point and the projection of this point on the line, we rewrite the line triangulation problem as:

$$\min_{\mathbf{M},\mathbf{N},\ldots,\alpha_{ik},\ldots} \mathcal{C}_{line} \text{ with } \mathcal{C}_{line} = \sum_{i=1}^{n}\sum_{k=1}^{p} d_e^2(\mathsf{P}_i(\alpha_{ik}\mathbf{M} + (1-\alpha_{ik})\mathbf{N}), \mathbf{q}_{ik}). \qquad (6)$$

Compare this cost function (5): the difference is that for line triangulation, the points are not supposed to match between the different views. Hence, a 3D point on the line is reconstructed for each image point, while in the POP triangulation problem, a 3D point on the line is reconstructed for each image point correspondence. Now, the interesting point is to determine if, in practice, cost functions (5) and (6) yield close solutions for the reconstructed 3D line. Obviously, an experimental study is necessary, and we refer to §6. However, we intuitively expect that the results are close.

Point-on-Line Triangulation We study the problem of point-on-line optimal triangulation: given a 3D line, represented by two 3D points $\mathbf{M}$ and $\mathbf{N}$, a set of corresponding image points $\ldots, \mathbf{q}_{ik}, \ldots$, find a 3D point $\mathbf{Q}_k \sim \alpha_k\mathbf{M}+(1-\alpha_k)\mathbf{N}$ on the given 3D line, such that the squared Euclidean distances between the predicted and the observed points is minimized.

For point-on-line triangulation, we formalise the problem as $\min_{\alpha_k} \sum_{i=1}^{n} d_e^2(\mathsf{P}_i(\alpha_k\mathbf{M} + (1-\alpha_k)\mathbf{N}), \mathbf{q}_{ik})$ and by introducing $\mathbf{b}_i = \mathsf{P}_i(\mathbf{M}-\mathbf{N})$ and $\mathbf{d}_i = \mathsf{P}_i\mathbf{N}$, we obtain:

$$\min_{\alpha_k} \mathcal{C}_{pol} \text{ with } \mathcal{C}_{pol} = \sum_{i=1}^{n} d_e^2(\alpha_k\mathbf{b}_i + \mathbf{d}_i, \mathbf{q}_{ik}). \qquad (7)$$

Sub-optimal linear algorithm. We give a linear algorithm, based on approximating the optimal cost function (7) by replacing the Euclidean distance d_e by the algebraic distance d_a. The algebraic cost function is $\sum_{i=1}^{n} d_a^2(\alpha_k\mathbf{b}_i + \mathbf{d}_i, \mathbf{q}_{ik}) = \sum_{i=1}^{n} \|\alpha_k\mathsf{S}[\mathbf{q}_{ik}]_\times\mathbf{b}_i + \mathsf{S}[\mathbf{q}_i]_\times\mathbf{d}_i\|^2$. A closed-form solution giving the best α_k in the least-squares sens is $\alpha_k = -\frac{\sum_{i=1}^{n} \mathbf{b}_i{}^\mathsf{T}[\mathbf{q}_{ik}]_\times\tilde{\mathsf{I}}[\mathbf{q}_{ik}]_\times\mathbf{d}_i}{\sum_{i=1}^{n} \mathbf{b}_i{}^\mathsf{T}[\mathbf{q}_{ik}]_\times\tilde{\mathsf{I}}[\mathbf{q}_{ik}]_\times\mathbf{b}_i}$ with $\tilde{\mathsf{I}} \sim \mathsf{S}^\mathsf{T}\mathsf{S} \sim \begin{pmatrix} 1 & & \\ & 1 & \\ & & 0 \end{pmatrix}$.

Optimal polynomial algorithm. This algorithm consists in finding the roots of a degree-$(3n-2)$ polynomial in the parameter α_k, whose coefficients depend on the $\mathbf{b}_i$, the $\mathbf{d}_i$ and the $\mathbf{q}_{ik}$. Due to lack of space, details are left to an extended version of the paper.

5 Bundle Adjustment

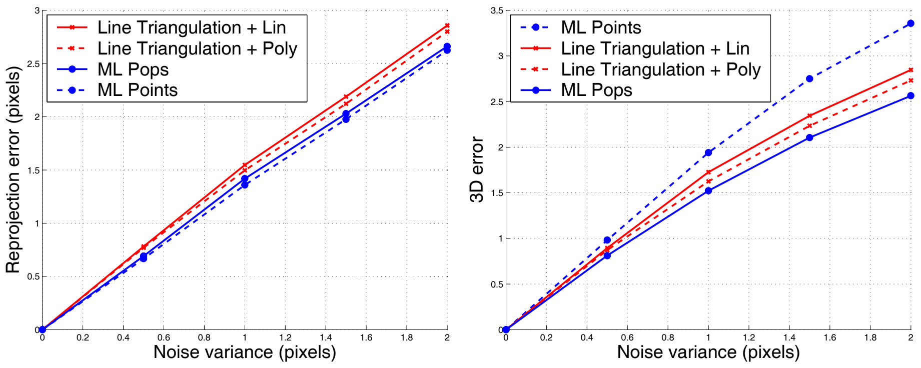
Bundle adjustment consists in minimizing the reprojection error over structure and motion parameters:

$$\min_{\mathsf{P}_1,\ldots,\mathsf{P}_n,\mathbf{M}_1,\mathbf{N}_1,\ldots,\mathbf{M}_m,\mathbf{N}_m,\ldots,\alpha_{jk},\ldots} \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p} d_e^2(\mathsf{P}_i(\alpha_{jk}\mathbf{M}_j + (1-\alpha_{jk})\mathbf{N}_j), \mathbf{q}_{ijk}),$$

where we consider without loss of generality that all points are visible in all views. We use the Levenberg-Marquardt algorithm to minimize this cost function, starting from an initial solution obtained by matching pairs of images and computing pair-wise fundamental matrices using the algorithms of §§2 and 3, from which the multiple-view geometry is extracted as in [11]. Multiple-view matches are formed, and the POPs are triangulated using the optimal method described in §4.

6 Experimental Results

We simulate a set of 3D POPs observed by two cameras, with focal length 1000 pixels. To simulate a realistic scenario, each POP is made of 5 supporting points. The supporting points are projected onto the images, and a Gaussian centered noise is added. The images of the supporting lines are determined as the best fit to the noisy supporting points. These data are used to compare quasi-metric reconstructions of the scene, obtained using different algorithms. We mesure the reprojection error and a 3D error, obtained as the minimum residual of $\min_{\mathsf{H}_u} \sum_j d^2(\underline{\mathbf{Q}}_j, \mathsf{H}_u \mathbf{Q}_j)$, where $\underline{\mathbf{Q}}_j$ are the groung truth 3D points, $\mathbf{Q}_j$ the reconstruction points and H_u an aligning 3D homography.



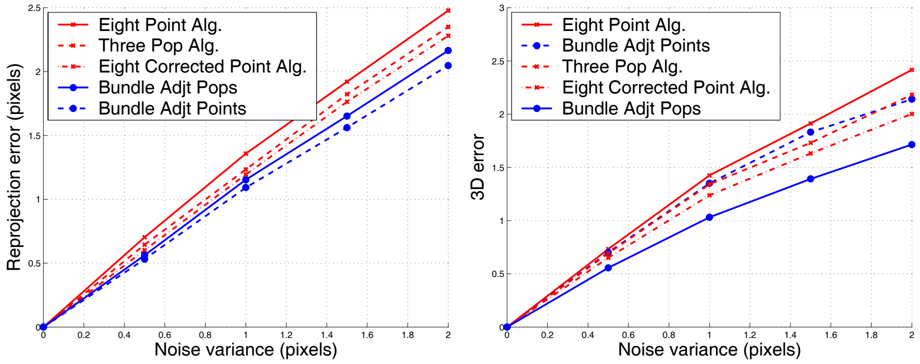
(a) (b)

Fig. 3. Reprojection and 3D error when varying the added image noise variance to compare structure and motion recovery methods.

Comparing triangulation algorithms. The two first methods are based on triangulating the supporting line, then each supporting point using the linear solution (method 'Line Triangulation + Lin') or using the optimal polynomial solution (method 'Line Triangulation + Poly'). The third method is Levenberg-Marquardt minimization of the reprojection error, for POPs (method 'ML Pops') or points (method 'ML Points'). We observe on figure 3 (a) that triangulating the supporting line followed by the supporting points on this line (methods 'Line Triangulation + *') produce results close to the non-linear minimization of the reprojection error of the reprojection error of the pop (method ‘ML Pops’). Minimizing the reprojection error individually for each point (method ‘ML Points’) produce lower reprojection errors.

Concerning the 3D error, shown on figure 3 (b), we also observe that methods ‘Line Triangulation + *’ produce results close to method ‘ML Pop’. However, we observe that method ‘ML Points’ gives results worse than all other methods. This is due to the fact that this method does not benefit from the structural constraints defining pops.

Comparing bundle adjustment algorithms. The two first methods are based on computing the epipolar geometry using the eight point algorithm (method ‘Eight Point Alg.’) or the three pop algorithm (method ‘Three Pop Alg.’), then triangulating the pops using the optimal triangulation method. The two other methods are bundle adjustment of pops and points respectively. We observe on figure 4 (a) that the eight point algorithm yields the worse reprojection error, followed by the three pop algorithm and the eight corrected point algorithm. Bundle adjustement of pops gives reprojection error slightly higher than with points. However, figure 4 (b) shows that bundle adjustment of pops gives a better 3D structure than point, due to the structural constraints. It also shows that the eight corrected point algorithm yields good results.



(a) (b)

Fig. 4. Reprojection and 3D error when varying the added image noise variance to compare triangulation methods.

7 Conclusions and Further Work

We addressed the problem of automatic structure and motion recovery from images containing lines. We introduced a feature that we call pop, for Pencil-of-Points. We demonstrated our matching algorithm on real images. This confirms that the repeatability rate of pops is higher than the repeatability rates of the points and lines from which they are detected. This also shows that using POPs, wide baseline matching and the epipolar geometry can be successfully computed in an automatic manner, using simple cross-correlation. Experimental results on simulated data show that due to the strong structural constraints, POPs yield structure and motion estimates more accurate than with points.

Advantages for using POPs are numerous. Briefly, localization, repeatability rate and structure and motion estimate are better with POPs than with points, and robust estimation is very efficient since only three pairs of POPs define an epipolar geometry. For this reason, we believe that this new feature could become standard for automatic structure-and-motion in man-made environment, i.e. based on lines.

Further work will consist in investigating the determination of parameters $\boldsymbol{\mu}$ needed to compute undistorted cross-correlation, since we believe that it could strongly improve the initial matching step, and studying methods for estimating the trifocal tensor from triplets of POPs.

Acknowledgements. The first author would like to thank Frederik Schaffalitzky from the University of Oxford for fruitful discussions. This paper benefited from suggestions from one of the anonymous reviewers. Images of the Valbonne church have been provided by INRIA Sophia-Antipolis.

References

1. F. Bignone, O. Henricsson, P. Fua, and M. Stricker. Automatic extraction of generic house roofs from high resolution aerial imagery. In *ECCV*, pp.85–96. April 1996.
2. J. Canny. A computational approach to edge detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
3. M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381 – 395, June 1981.
4. R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.
5. R.I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, 1997.
6. R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, June 2000.
7. H.C. Longuet-Higgins. A computer program for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.
8. G. Médioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics and Image Processing*, 31:2–18, 1985.
9. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, volume I, pages 128–142, May 2002.
10. C. Schmid and A. Zisserman. Automatic line matching across views. In *CVPR*, pages 666–671, 1997.
11. B. Triggs. Linear projective reconstruction from matching tensors. *Image and Vision Computing*, 15(8):617–625, August 1997.
12. Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, March 1998.