

3D Reconstruction of conics

IACV Project

Veronica Cardigliano
Politecnico di Milano

veronica.cardigliano@mail.polimi.it

Thomas Jean Bernard Bonenfant
Politecnico di Milano

thomasjean.bonenfant@mail.polimi.it

Federico Caspani
Politecnico di Milano

federico1.caspani@mail.polimi.it

Abstract

Man-made environments often does not contain a high number of distinctive feature points, so here traditional Structure-from-Motion approaches may work not so well. A different approach to overcome this problem can be to implement a 3D reconstruction approach exploiting geometrical structures different from points, i.e. lines and curves. Analyzing the state of the art of the problem, the proposed approaches of 3D reconstruction starting from line segments, we tried to take a step towards the integration of conics in this pipeline. We have implemented an approach of conics reconstruction from two views by exploiting the geometrical properties of these elements in projective geometry. We have also tried to extended this method to a multi-view situation. We have obtained good results, which could be useful for continuing the integration of conics in the 3D reconstruction process.

1. Introduction

The academic project we worked on required us to look for a Structure from Motion that, differently from traditional pipelines working with sparse points, would use other geometric primitives such as lines. Our research started from the proposed work about Line-Based 3D Reconstruction of Wiry Objects [3] whose aim is to solve the possible failure of point-based Structure from Motion in man-made and urban environments with weakly textured surfaces, with a low number of distinctive feature points. Their pipeline consist in the generation of 3D line models in a novel and efficient way, further improved in their future works [1] [2] both in terms of performance and documentation. In fact Hofer, Maurer and Bischof [2] also published a well

documented implementation of their approach¹, which we had the opportunity to run and verify. Given the completeness of Line3D++ approach on lines, we decided to take up the challenge launched in their possible future implementations, that is to extend the method to curved edges as well. Elliptical arcs are also frequently occurring in man-made environments, so this could be of great use as much as the 3D reconstruction starting from lines. We reviewed the approach [4] suggested by Hofer [2], that however was quite far from Line3D++ implementation. We will give a brief summary of the two approaches in section 2. We decided then to focus on another fundamental geometric object in images that are conics. With respect to points and lines, conics are more global and compact features, they are projective transformations of points and lines and their mathematical properties have been studied in depth in algebraic projective geometry field, providing a strong mathematical support. Moreover, differently from points and lines, conics have sufficient information to impose correspondence conditions that will be explained in the "Proposed Approach" section. These motivations are what led also Long Quan [5] to focus his attention on these geometric elements. His work on conic reconstruction and correspondence from two views is a basic and general approach of which we have not found an available implementation. We therefore decided to start from here, analyzing the method (section 3) and proceeding with some experiments (section 4) providing our contribution by implementing this approach and trying to extend it on order to get closer to a possible wider usage.

2. Related work

The main works we focused on to take note of the state of the art of 3D scene reconstruction using geometric elements different from points are: "Efficient 3D scene abstraction

¹available at: <https://github.com/manhofer/Line3Dpp>

using line segments” [2] and ”A Parameterless Line Segment and Elliptical Arc Detector with Enhanced Ellipse Fitting” [4], suggested in the conclusion of the former work as a possible evolution. In fact, Hofer, Maurer and Bischof [2] have developed a method to reconstruct linear 3D structures based only on 2D line segments, and they would like to extend it to reconstruct also curved edges combining lines and elliptical arcs.

2.1. 3D scene reconstruction using line segments

Reconstruction from lines is particularly important for urban indoor and outdoor environments, characterized by linear structures. The phases of the proposed Line3D++ [2] approach are:

1. From an oriented image sequence as input, whose camera poses can be obtained by any SfM (Structure from Motion) pipeline, is extracted a sparse set of 3D points in order to define visual neighbors among images (clustering the ones that see the same thing). This is done considering the similarity among 3D (world) points contained in images, exploiting i.e. Dice similarity. Then, are established potential correspondences among line segments of different images, considering each pair of images for each particular visual neighbor and each pair of segments in them, and exploiting weak epipolar constraints.
2. Correspondences are evaluated through a scoring function, to distinguish correct and incorrect matches for each segment. It is based on 3D similarities and mutual support, analyzing the number of different views that capture each correspondence.
3. 3D locations are assigned to 2D segments, since each 2D segment can only be a projection of one specific 3D structure. The 3D location is defined as the 3D hypothesis with the highest confidence for each 2D segment.
4. Clustering 2D segments of the same 3D entity from different views, based on their spatial 3D proximity. This is done through a graph clustering procedure exploiting a global affinity matrix encoding pairwise similarities between correlated 2D segments. The output is a set of 3D lines clusters, each one composed of:
 - 2D residuals
 - 3D line obtained by 3D depth estimates of its 2D residuals (H)
 - One or more colinear 3D line segments (part of infinite line H)
5. An optional step is included in the paper [2], regarding the optimization of the SfM result and combining 3D lines with bundle adjustment.

The method scales almost linearly with the number of input images and has low memory requirements, so also hundreds of images can be exploited for reconstruction.

2.2. Parameterless line segment and elliptical arc detector with enhanced ellipse fitting

Parameterless line segment and elliptical arc detector with enhanced ellipse fitting is divided in 3 main steps:

1. **Candidate selection** through an heuristic: same of LSD (Line Segment Detection) for line segment part, is exploited the region growing process following the same gradient to find segments. For circular or elliptical arcs, instead, is used a curve growing process, with a convexity rule and a smoothness rule (it searches for a convex and roughly smooth contour), plus conic fitting for small or incomplete curves.
2. **Candidate validation:** in the end there are 3 candidates:
 - rectangle by region growing
 - circular ring
 - elliptical ring by curve growing (rings are composed of eight parameters: the five to fit the ellipse, plus three that are the delimiting angles and ring width)

These candidates are found through ”a contrario” framework, that has the advantage of being parameterless, non-accidental (rejects candidates found accidentally), good against false positives, with an automatic computation of detection thresholds.

Both these 2 points represent ELSD (Ellipse and Line Segment Detector) algorithm.

3. **Model selection** to choose the best geometric interpretation, that is the best candidate among the ones declared meaningful by the validation step.

We run the Line3D++ (result in fig. 1) implementation available on GitHub and it worked in the environment and with the tools described by the developers. These two methods have been already developed and work quite efficiently. However, Patraucean’s method is quite complex and has not been applied yet to Line3D++, so we decided to start exploring a possible application of a simpler conic reconstruction from two views. Conic Reconstruction and Correspondence From Two Views by Long Quan [5] resulted definitely pertinent to what we were looking for.

3. Proposed approach

Following Quan’s work [5], we derived algebraically two independent polynomial conditions that should be satisfied

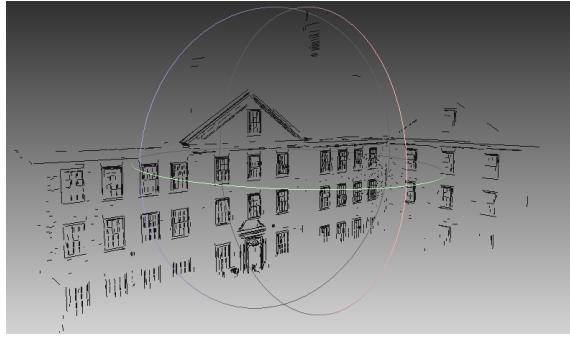


Figure 1: The result of the Line3D++ algorithm we ran with the example dataset in GitHub repository

for each pair of corresponding conics across two views, depending on the relative orientation of the views. A unified closed-form solution both for projective reconstruction of conics in space from uncalibrated camera views and metric (Euclidean) reconstruction from two calibrated camera views has been developed. The problem is first formulated using projective geometry based on projective properties of quadric surfaces. Then linear algebra is used to analyze the eigensystem of a matrix pencil of order four (pencils of quadric surfaces). The reconstruction procedure is linear.

3.1. Projection Matrices

Projection matrices can be decomposed into the form showed below (eq.1), where A (3x3 upper triangular) accounts for the five intrinsic parameters of the camera, R and t are a rotation matrix and a translation vector that account for the six extrinsic parameters.

$$P = A(I_3O_3) + \begin{pmatrix} R & t \\ O_3^T & 1 \end{pmatrix} \quad (1)$$

To realize the projection matrices (P and P') for two views, that can be captured from the same or different cameras, Quan [5] considered three frequent cases:

1. Two views from Calibration: the system is fully off-line calibrated, so there is a complete description of the views through the two projection matrices, with a resulting 3D reconstruction that will be fully metric.
2. Two views from Motion: if the two views are taken by a moving calibrated camera, equivalent to knowing only the essential matrix of the two views.
3. Two views from Epipolar Geometry: if the two views are taken from an uncalibrated moving camera, in this case can be estimated only the epipolar geometry between the two views. The resulting 3D reconstruction could be no more metric but only projective.

In our experiments, we explored the second case, starting from calibrating our cameras as done in standard Structure from Motion approaches, in order to obtain camera poses (extrinsic parameters). The reproduction of the experiments on synthetic data by Quan [5], instead, come from a calibrated stereo system (case 1).

3.2. Correspondence among conics by polynomial conditions

Given a corresponding pair of conics from two views (C in $view_1$, C' in $view_2$) so that:

$$C \equiv u^T C u = 0 \leftrightarrow C' \equiv u'^T C' u' = 0 \quad (2)$$

The goal is to find a conic in space that has been projected into C and C' . A conic is the intersection of a quadric surface and a plane, so it is found as the intersection of one of the cones associated with an image of the conic (C or C') and the plane that contains the conic in space. "PROPOSITION 1. Given the projection matrix P of a camera, the equation of the cone which joins the conic $u^T C u = 0$ in the image plane to the projection center of the camera is $x^T Q x = 0$, with $Q = P^T C P$." [5]

Considering the pencil of quadric surfaces given by:

$$Q + \lambda Q' = 0 \forall \lambda \quad (3)$$

each quadric belonging to it passes through all the common points of the two cones (obtained joining C and C' with the center of the camera for view 1 and 2 respectively). These intersections represent two curves called the base curve of the pencil. Given our constraints one of the components of the base curve is a conic in space. And if one of the components of the base curve is a conic, also the other one should be a conic. According to the results of projective geometry, also a pair of planes belongs to the pencil of quadrics, since it is the degenerate quadric of rank 2. And this particular degenerate quadric is examined. Now, the goal becomes to find a λ so that $A + \lambda B$ has rank 2 (degenerate quadric of the pencil). Then $x^T Ax = 0$ and $x^T Bx = 0$ are the cones corresponding to C and C' (the images of the conic). Rank 1 is proved to be unreachable. As stated and proved in the reference paper [5], "PROPOSITION 4. There exist only 2 independent polynomial conditions for a corresponding pair of conics." Then, the next step is to derive these conditions. Considering the characteristic polynomial of $A + \lambda B$ (eq.4), to have rank 2 it must have two distinct non zero eigenvalues and a double zero eigenvalue, that is $a_3 = 0$ and $a_4 = 0$. a_4 is by definition the determinant of $C(\lambda)$.

$$\begin{aligned} C(\lambda) &= A + \lambda B, \\ |C(\lambda) - \mu I| &= \mu^4 + a_1(\lambda)\mu^2 + a_3(\lambda) + a_4(\lambda) = 0 \end{aligned} \quad (4)$$

$$a_4(\lambda) = |C(\lambda)| = |A + \lambda B| = I_1 \lambda^4 + I_2 \lambda^3 + I_3 \lambda^2 + I_4 \lambda + I_5 \quad (5)$$

Then, I_1 and I_5 (5) are equal to the determinants of A and B that are zero (matrices of cones of rank 3, full rank). So a_4 can be factorized as:

$$a_4(\lambda) = \lambda(I_2\lambda^2 + I_3\lambda + I_4) = 0 \quad (6)$$

To have a rank 2 matrix in the pencil, we should at least have a generalized eigenvalue of multiplicity 2, so the equation must have two equal roots. From this we reach the first condition for correspondence:

$$\Delta \equiv I_3^2 - 4I_2I_4 = 0 \quad (7)$$

Then, a_3 is examined:

$$a_3(\lambda) = J_1\lambda^3 + J_2\lambda^2 + J_3\lambda + J_4 \quad (8)$$

The obtained equation 8 is a cubic polynomial in λ . Solving with respect to λ , using $\Delta = 0$, gives the second correspondence condition:

$$\Theta \equiv -J_1I_3^3 + 2J_2I_3^2I_2 - 4J_3I_3I_2^2 + 8J_4I_2^3 = 0 \quad (9)$$

$A + \lambda B$ pencils considered have a double generalized eigenvalue, so based on whether the algebraic multiplicity exceeds or not the geometric multiplicity, the matrix pencils can be defective or simple. In the case of simple ones, the condition $\Delta = 0$ is necessary and sufficient for C to have rank 2, while the second one becomes obsolete (implied by the first under the assumption of simple structure). In practice the matrix pencils have almost exclusively the simple structure, so it is assumed to be in this situation.

"PROPOSITION 5. The condition $\Delta = 0$ can be interpreted as that the absolute projective invariant I associated with the pair of cones is a constant" [5].

3.3. Closed form solution for reconstruction

The closed form solution to reconstruct the conic in space starting from its corresponding pair from two views can be developed extracting the degenerate quadric surface and then finding the planes from it. Solving the quadratic equation given by the second factor in eq. 6, knowing that we must have two equal roots for it, we obtain:

$$\lambda = -\frac{I_3}{2I_2} \quad (10)$$

obtaining the desired matrix $C = A + \lambda B$ of the degenerate quadric surface.

Then, the last step is to extract the two planes from 2 that is a rank 2 matrix. Given the conditions, the polynomial of the matrix C becomes:

$$\mu^2(\mu^2 + a_1(\lambda)\mu + a_2(\lambda)) = 0 \quad (11)$$

Solving the quadratic equation on the right we'll find the remaining nonzero eigenvalues. These can be found diagonalizing C, $T^TCT = diag(\mu_1, \mu_2, 0, 0)$, considering $x = Tx'$,

the quadric $x^T C x = 0$ becomes $x'^T diag(\mu_1, \mu_2, 0, 0)x' = \mu_1x'^2 + \mu_2y'^2 = 0$. Solving, the two planes become:

$$(\sqrt{\mu_1}, \pm\sqrt{-\mu_2}, 0, 0)^T x' = \sqrt{\mu_1}x' \pm \sqrt{-\mu_2}y' = 0 \quad (12)$$

To obtain real planes, $\mu_1\mu_2$ must be < 0 ,

$$(Tp_i')^T x = (\sqrt{\mu_1}v_1 \pm \sqrt{-\mu_2}v_2)^T x = 0 \quad (13)$$

considering v_1 and v_2 eigenvectors corresponding to μ_1 and μ_2 eigenvalues. So, the conic in space is the intersection of one of the two cones with one of the planes above. The final step is to choose the right plane that contains our conic. If we assume that the conic in space is a non transparent object, the visibility constraint may be used to get rid of the surplus solution. To be visible for a non transparent object from two different viewpoints, it is necessary that the two viewpoints are located on the same side of the plane (see fig. 2). This new constraint implies that the signed distance from our camera centers and the plane must have the same sign: if $(o^T p_i)(o'^T p_i) > 0$ then o and o' lie on the same side of the plane $p_i^T x = 0$ where $o = Ker(P)$ and $o' = Ker(P')$.

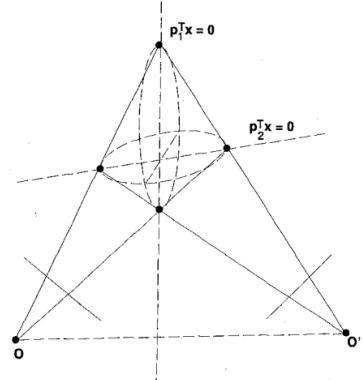


Figure 2

4. Experiments

Datasets. We used 4 different datasets.

A synthetic dataset providing camera projections matrices P_1 and P_2 from a real stereo system viewing two conics in space. The conics are described respectively by the intersection of the quadric surface $x^T Q_1 x = 0$ and the plane $p_1^T x = 0$ for the first conic and the intersection of $x^T Q_2 x = 0$ and $p_1^T x = 0$ for the second.

$$P_1 = \begin{pmatrix} 1.393757 & -0.244708 & -14.170794 & 368.0 \\ 10.624195 & 2.396275 & -0.433595 & 202.0 \\ 0.002859 & 0.011811 & -0.003481 & 1.0 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} 1.374060 & -0.612998 & -14.189693 & 371.0 \\ 10.979978 & -1.621189 & -0.469463 & 207.0 \\ 0.007648 & 0.010572 & -0.003449 & 1.0 \end{pmatrix}$$



(a) Image 1



(b) Image 2

Figure 3: Two images taken from the same moving camera



(a) Image 1



(b) Image 2

Figure 4: Two images containing three conics taken from the same moving camera

$$\left\{ \begin{array}{l} Q_1 = \begin{pmatrix} -0.0013 & 0.5 \cdot 10^{-5} & -0.00023 & 0.0058 \\ 0.5 \cdot 10^{-5} & -0.000078 & -0.00034 & 0.0033 \\ -0.00023 & -0.00034 & -0.0014 & 0.011 \\ 0.0058 & 0.0033 & 0.011 & -0.038 \end{pmatrix}, \\ p_1 = (-0.021 \ -0.16 \ -0.092 \ 1.0)^T \end{array} \right.$$

$$\left\{ \begin{array}{l} Q_2 = \begin{pmatrix} 1.0 & 0.0 & 0.0 & -9.0 \\ 0.0 & 1.0 & 0.0 & -2.0 \\ 0.0 & 0.0 & 1.0 & -10.0 \\ -9.0 & -2.0 & -10.0 & 85.0 \end{pmatrix}, \\ p_2 = (-0.196589 \ -0.812143 \ 0.239359 \ 1.0)^T \end{array} \right.$$

After validating the algorithm on this synthetic dataset we tried to apply it on real world images taken from our phone cameras. The first dataset with real images is shown in fig. 3 and contains only one conic that is the border of the coffee cup. The second dataset with real images contains three conics that come from parts of a coffee pot as shown in fig. 4. Lastly we used for the three view application of the algorithm a different dataset of three images taken from a different camera than the previous one.

Experiments setup. All the experiments were performed inside the MATLAB environment.

The camera used to take the images was calibrated using the Camera Calibrator App in the Computer Vision ToolBox

with images of a chessboard pattern in order to find the intrinsic parameters.

To estimate the extrinsic parameters, we exploited a basic approach from the Structure From Motion pipeline. Starting from the two images taken slightly moving the camera of which we have already computed the intrinsic parameters, these images are first undistorted. Then, the approach consists in matching a sparse set of points between the two images (i.e. detecting corners in the first one and tracking them in the second one). Finally, the essential matrix and the epipolar inliers are computed, finding from them the camera pose.

Using the synthetic data we added noise to the points sampled from the conics and next we used the least squares algorithm to fit new conics.

In real images each conic was retrieved selecting manually more than five points and applying least squares fitting.

Once we have retrieved the conics we apply the algorithm on each possible pair of conics (C_{1i}, C_{2j}) where C_{1i} is the i-th conic in image 1 and C_{2j} is the j-th conic in image j . Wrong correspondences are filtered using a threshold on value Δ (Eq. 7): if value Δ computed for an arbitrary pair of conics (C_{1i}, C_{2j}) is lower than threshold ϵ we can proceed with the reconstruction, otherwise the pair does not match.

The algorithm outputs a plane p . The intersection of this plane p with the two cones A and B generated from camera centers gives us two reconstructions of the same conic.

Lastly we tried to use the reconstruction algorithm in a three view setting coming from three images taken by the same camera in motion. The two view algorithm was applied in sequence on pairs (image 1, image2) and (image 2, image 3).

Results and discussion. In the synthetic dataset the algorithm achieved to reconstruct very well the given conics as shown in fig. 5. In the other experiments however we did not have data to represent the original conics so we can only see the reconstruction results. In fig. 6 we can see the reconstructed conic from the coffee cup images with the two cones A and B from the two cameras. Each cone will intersect the plane containing the conic so we have two reconstructions but in this example they match perfectly.

In the coffeepot images with three conics we had to match the correct conics between the two images. This was done tuning the parameter ϵ : if ϵ is too high the algorithm

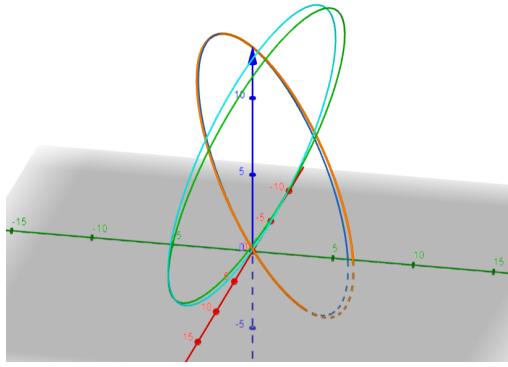


Figure 5: The orange and light blue conics are the target conics given by the synthetic dataset. The dark blue and green conics are the reconstructed conics obtained intersecting the obtained plane with one of the cones generated from camera centers. If noise is not added the reconstructed conics perfectly match the targets.

Δ	C'_1	C'_2	C'_3
C_1	$4.016 \cdot 10^{-9}$	0.026	$3.94 \cdot 10^{-4}$
C_2	0.021	$3.042 \cdot 10^{-6}$	$-1.501 \cdot 10^{-4}$
C_3	$3.699 \cdot 10^{-4}$	$-1.765 \cdot 10^{-4}$	$1.199 \cdot 10^{-9}$

Table 1: Different values of Δ for each pair of conics from the coffeepot images.

would have given rise to additional conics in space reconstructed from images of different conics. We can see the values of Δ for the different matches in Table 1: the correct matches are on the diagonal of the table and in fact Δ is the lowest for these pairs.

Lastly, on the three view experiment (which is a sequential application of our algorithm for a two views setting) we can see from fig.8 the different reconstructions for each conic. The reconstructed conics coming from the second application of the algorithm appear to be all moved along the axis of the cone with respect to the previous reconstructions. This may be caused by a not optimal calibration of the cameras.

5. Conclusion

With this work we had the opportunity to observe the state of the art regarding multi-view 3D reconstruction starting from geometric objects other than points. This is particularly useful in artificial environments, where lines and curves are very frequent and can be more easily reconstructed erroneously starting only from points as in the standard approach. We started (section 2) with the study and testing of a complete, efficient and well-implemented and

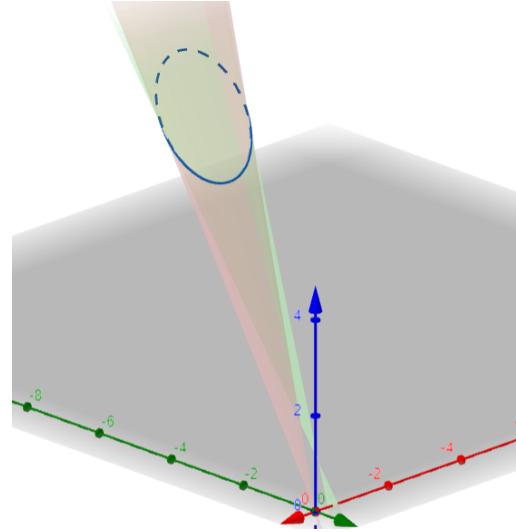


Figure 6: Coffee cup reconstruction with the two cones of the camera centers. The red cones starts from the center of camera 1 while the green ones start from the center of camera 2

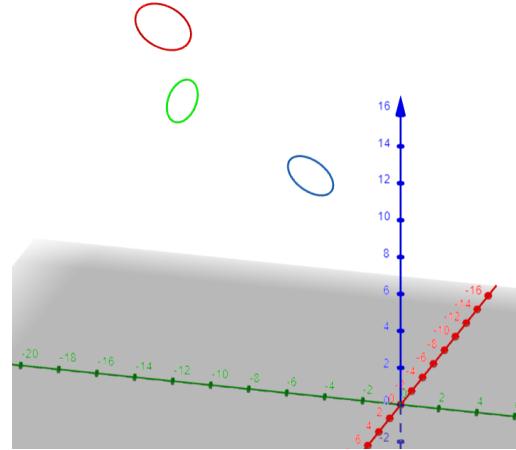


Figure 7: The camera of the first view is aligned with the z axis (blue in the image). The three conics are the reconstruction of the different parts from the coffee pot

documented approach of 3D reconstruction starting from straight lines, called Line3D++[2], to then consider the ELSD (Ellipse and Line Segment Detector) approach [4] proposed by Hofer, Maurer and Bischof [2] with the aim of integrating arcs into Line3D++. Given the complexity of the integration of the two methods, we started by reviewing (section 3) a general method of reconstruction of conics starting from two views [5] on the basis of pure geometrical considerations. We have implemented this method

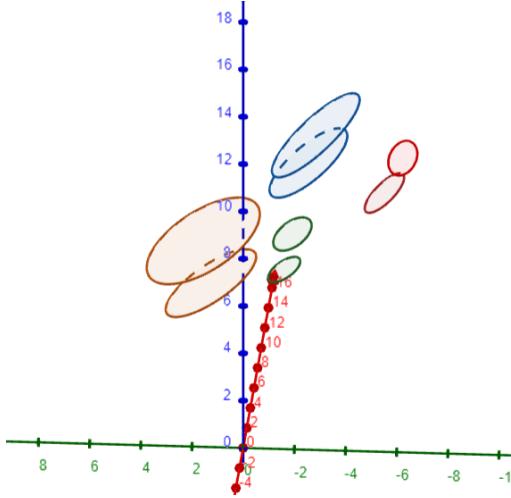


Figure 8: Three view reconstruction of four conics

in Matlab and we have tested it both on the synthetic data proposed by the author and on real data from our cameras (section 4). Our implementation is available on GitHub². The method proved to be valid and gave us good results, so we tried to extend it to a multi-view reconstruction applying the two view algorithm sequentially on image pairs, but we did not get so accurate results possibly due to a not optimal calibration of the extrinsic parameters. Possible future implementations may therefore be to include real world measurements in order to accurately compute the error on the reconstruction, extend effectively the algorithm to multi-view combining the different conic clusters into one unique 3D conic and finally try to integrate this approach in the Line3D++ pipeline as a possible extension.

References

- [1] M. Hofer, M. Maurer, and H. Bischof. Improving sparse 3d models for man-made environments using line-based 3d reconstruction. In *2014 2nd International Conference on 3D Vision*. IEEE, 12 2014. [1](#)
- [2] M. Hofer, M. Maurer, and H. Bischof. Efficient 3d scene abstraction using line segments. *Computer Vision and Image Understanding*, Mar. 2016. [1](#), [2](#), [6](#)
- [3] M. Hofer, A. Wendel, and H. Bischof. Line-based 3d reconstruction of wiry objects. In *Computer Vision Winter Workshop (CVWW)*, Feb. 2013. [1](#)
- [4] V. Pătrăucean, P. Gurdjos, and R. G. von Gioi. A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, pages 572–585. Springer Berlin Heidelberg, 2012. [1](#), [2](#), [6](#)
- [5] L. Quan. Conic reconstruction and correspondence from two views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):151–160, 1996. [1](#), [2](#), [3](#), [4](#), [6](#)

²<https://github.com/FedericoCaspani/ConicReconstructionIACV>