Relative pose from cylinder silhouettes

Anna Gummeson[0009−0003−3615−1679] and
Magnus Oskarsson[0000−0002−1789−8094]

CVML, Centre for Mathematical Sciences, Lund University, Lund, Sweden
{anna.gummeson, magnus.oskarsson}@math.lth.se

Abstract. In this paper we propose minimal solvers for relative pose estimation for two views of the projected silhouettes of two 3D cylinders. Using such line features instead of the standard point feature correspondences means more stable information (i.e. more stable to lighting condition, seasons, changes in environment etc.). Such features also lead to more compact and semantically interpretable representations in 3D as opposed to standard 3D point feature clouds. In this paper we show how it is possible to transform the problem into a simple parameterization where we can represent the problem as a set of six polynomials and provide solvers for their solutions. Through tests in synthetic and real settings we show that the solver is accurate and stable in the presence of added and inherent noise. Our code is publicly available[1].

Keywords: Relative pose · Cylinder geometry · Minimal solvers.

1 Introduction

In order to understand the world, we as humans make simplifications to interpret it. We typically use geometric primitives, to represent and build up the world, see e.g. the ideas of Marr [31], and the recognition-by-components (RBC) theory

[1] https://github.com/hamburgerlady/cylinder-SfM

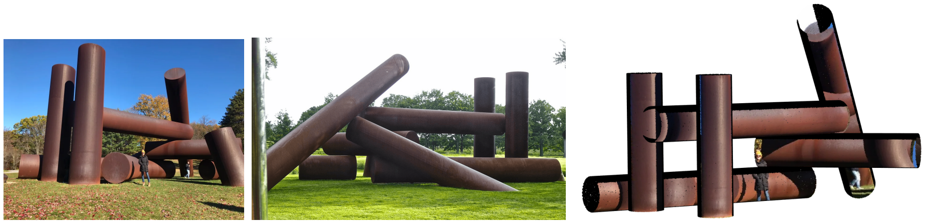

Fig. 1: An example of the target problem. Left and middle show two images of cylinder shapes from the Storm King Sculpture park in New York state. Note that there are *no* possible point feature correspondences between the two images, due to the view-point change. Traditional feature point based methods for SfM will hence fail. To the right is our reconstruction shown, based on cylinder geometry estimation.

of Biederman [7]. These ideas were put forward mostly in the context of object recognition, but are equally relevant for computer vision tasks such as 3D understanding and positioning and localization. Today, geometric estimation in Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM) applications is primarily built upon explicit representations (sparse 3D-points and camera matrices). Methods for extraction of semantic information from images are on the other hand typically highly data driven, and the representations are to a large extent learned and coded implicitly in neural network architectures. Many existing systems built to do SfM and SLAM work efficiently [35, 42, 43]. Sparse point clouds are well suited for matching, camera geometry estimation and optimization. However, there are significant problems with scaling to very large scenes, and with stability over time [41]. These methods are also not well suited for downstream tasks such as interpretation and recognition.

As an overall goal we would like to investigate the use of mid-level representations that carry more semantic meaning than simple points do, such as for example cylinder shapes. In this paper we will specifically study the problem of relative pose, given cylinder silhouette correspondences in two views. We will show that it is possible to estimate both the camera and cylinder geometry given only the silhouettes of two cylinders and two cameras. Note that we in this paper restrict image information to the silhouettes of the sides of cylinders, and not the apparent contours of the cross sections, i.e. we consider infinitely long cylinders (analogous to using lines as opposed to line segments for line geometry). Our focus has been on the geometric estimation, but line segment detectors have matured over the last years, [13, 37, 38, 48, 51] and we foresee that they will soon be applicable for many general computer vision tasks. An example of the problem setup is shown in Fig. 1. Here two images of a number of cylinders are shown. Note that since the images are taken from opposing views, there are no common point features seen in these images. This means that any point based method for SfM will fail. In general line correspondences will not give any constraints on the camera geometry in two views, but since we know that the silhouette lines lie on the cylinder outlines this will give constraints on the geometry. The 3D reconstruction to the right in the figure is based on our proposed near minimal solver given two cylinder silhouette correspondences. Given this initial solution, the other cylinders can be triangulated using the estimated camera matrices.

Our main contribution is a near minimal solver, that can be used for robust and accurate estimation of two view relative pose, given two or more cylinder silhouette correspondences. We show in our experiment that it works on a diverse set of scenarios, and on scenes that cannot be reconstructed using previous SOTA methods.

1.1 Related work

The previous work on geometric estimation involving quadrics and conics are primarily based on general projective models, [14, 30], and not using calibrated cameras. There are several results on pose or homography estimation and relative pose, e.g. in [17,18,23,33,34], and relative shape from two views was considered in [22,40] which is closely related to the theory for general silhouettes, [1,2].

A very relevant paper to us, is the work of Navab et al. [36], where they present algorithms for pose, 3D reconstruction and structure from motion from cylinders. Here there are however not given any explicit solvers for relative pose estimation. The problem of absolute camera pose from cylinder silhouettes was considered in [19]. The works from [50] and [45] address camera calibration, based on cylinders, but using the apparent contours of the ends of finite cylinders instead. The inverse problem, camera position from known cylinders, is covered in [19]. Triangulation methods for planar conics [21] and more complex surfaces based on line representations [24], have been proposed. More recently, work on cylinder triangulation has been given [20].

The most closely related work to ours is the work of Tegler et al. [46] where they give solutions to relative pose. There they, however, consider only the simplified case where all cylinders are parallel. This simplifies the geometry significantly.

Our proposed solver is targeted for use in robust estimation schemes such as RANSAC [16]. Following Fischler and Bolles, there have been many proposed improvements to the original algorithm, based on improved scoring [3,5,6], better sampling [11,39,47], early sample rejection [10] and early stopping criteria [32]. Many other variants have also recently been published [9,39,49]. Many modern RANSAC methods include local optimization to work with less accurate initial models, such as LO-RANSAC [12], LO$^+$-RANSAC [29], MAGSAC++ [5] and GC-RANSAC [4]. Recently it has been shown that in many cases it is more favourable to have models based on very few correspondences, even though these might yield less accurate models [15]. Our method fits very well into this modern scope of RANSAC.

2 Cylinder Geometry

A 3D cylinder can be defined using polynomial constraints based on its radius, direction and translation. When this cylinder is projected into a calibrated camera, the silhouette lines will additionally depend on the camera rotations and translations in 3D. This section will give a short explanation of these geometrical properties and its effect on the polynomial constraints. For more information on the geometry of cylinders and quadric surfaces the reader is referred to the excellent book by Semple and Kneebone [44].

Any quadric surface in 3D can be expressed with the quadratic equation

$$U^T C U = 0, \tag{1}$$

where $U \in \mathbb{P}^3$ is any point on the quadric surface expressed in homogeneous coordinates and C is a 4×4 symmetric matrix defining the *quadric locus*. Equivalently we may express the quadric in its dual form

$$\Pi^T D \Pi = 0, \tag{2}$$

where $\Pi \in \mathbb{P}^3$ is any tangent plane to the quadric and D is the inverse matrix to C (up to a non-zero scalar). It should be noted however that in the special case of cylinders C does not have full rank, but rank 3. The problem of finding an inverse is solved by an intermediate projection into 2D, onto any plane perpendicular to the direction of the cylinder. Here we may find an inverse to the reduced C, representing the circular cross-section of the cylinder. In addition to this we need to represent the direction v of the cylinder in 3D.

The back-projected cylinder silhouette lines should tangent the cylinder, and (2) can be expressed in the camera matrix P and projected silhouette lines l using

$$\Pi = P^T l \tag{3}$$

and the resulting equation is

$$l^T P D P^T l = 0. \tag{4}$$

In addition to this constraint, we should also have that back-projected planes ($P^T l$) should intersect in a line containing the point at infinity, corresponding to the cylinder direction v, i.e.

$$l^T P v = 0. \tag{5}$$

These are the two constraints that a cylinder silhouette line will pose on the geometry.

3 Relative Pose

The problem of relative pose is to find both camera parameters (rotations and translation in the calibrated case) and the 3D features. Normally these 3D features may be points, but in this paper we consider infinite cylinders with defining variables radius, direction and translation. Each cylinder has five parameters. A calibrated camera has six degrees of freedom. For each silhouette line in each view we get the two constraints given by (4) and (5). This means that if we use both silhouette lines for each cylinder, we get $4MN$ constraints in total for M cameras and N cylinders. If we want to know when the relative pose problem is well posed, we may count the unknowns and compute to the number of equations we have. Due to the gauge freedom in a relative pose problem we are able to choose the coordinate system with seven degrees of freedom (three translation-, three rotation- and one scale-parameter). In total we have $4MN-(6M+5N-7)$ excess constraints. This number is given in Table 1 for different combinations of M cameras and N cylinders. We see that we have an interesting near minimal case, for two views of two cylinders. This is the case that we will consider in this paper.

3.1 Problem parameterization

We now turn to the near minimal case of two cylinders and two cameras. We will here describe how we can simplify the problem, to ensure a fast and accurate

Table 1: Excess constraints $4MN - (6M + 5N - 7)$, for multiple view geometry of cylinders, with M cameras and N cylinders. In this paper we develop methods for solving the near minimal case of $M = 2, N = 2$.

		N cylinders				
		1	2	3	4	5
M cameras	1	0	-1	-2	-3	-4
	2	-2	$\boxed{1}$	4	7	10
	3	-4	3	10	17	24
	4	-6	5	16	27	38

solver. Denote the cylinders C_1 and C_2 and the cameras P_1 and P_2. To remove as many variables as possible before applying the solvers we may rectify the cameras in accordance with cylinder C_1, and also rotate P_1 as to center cylinder C_1 in view one. The result after rectification is that the cylinder in camera coordinates has a vanishing point at [0 1 0]. This is done by rotating the camera. The effect of rectification is that the projected silhouette lines is on the form $x = x_0$ in the camera. See figure 2. When rectifying, the intersection between the lines in the image is calculated and this is the vanishing point of the cylinder in camera coordinates. A rotation matrix for the rectification can then be constructed using

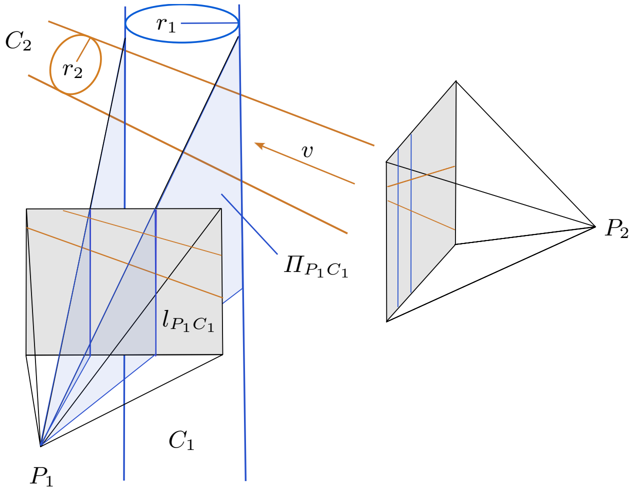

Fig. 2: Setup of cameras and cylinders after cameras P_1 and P_2 have been rectified to cylinder C_1. We denote the radii r_1 and r_2 respectively. $l_{P_1C_1 1}$ and $l_{P_1C_1 2}$ are the lines in camera P_1 from the silhouettes of C_1. The tangent plane to C_1 that give rise to one of these lines is $\Pi_{P_1C_1}$. v is the directional vector of C_2, or alternatively the vanishing point at which C_2 is pointing.

any orthogonal vectors to the infinity point (note that we still have freedom in rotating around the axis of the vanishing point, i.e. the y-axis). We select vectors such that the cylinder still is in-front of the camera. For details on image rectification see e.g. [8]. We may also choose the global coordinate system such that camera one is

$$P_1 = [I \;\; 0]. \tag{6}$$

These operations removes the x-component of the cylinder translation and both silhouettes of C_1 projected into P_1 will result in the same equation, so only one needs to be considered. The scale of the system can be set such that the shortest distance p_z from camera P_1 to center line of C_1 is equal to one. The radius of cylinder 1 follows directly from $p_z = 1$, using similarity and the position x of the straight line projected into camera 1. See figure 3 for the triangles in question. (Note that they are represented in different coordinate systems). The result is that $r_1 = |x|/\sqrt{x^2+1}$. All these changes can be done without loss of generality.

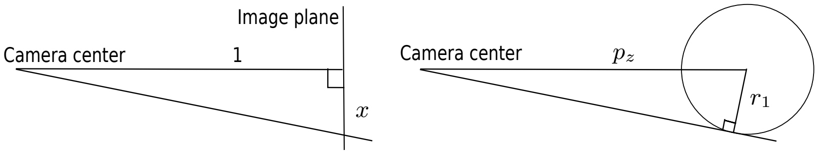

Fig. 3: The similar triangles used to calculate r_1

The inverses of the rectifying rotations can be applied after we have run our solver, to restore the cameras to the original coordinate system. The result of these modifications is that camera P_2 has four remaining degrees of freedom: the translation and one rotation angle around the y-axis. Cylinder C_2 has five, namely the direction, two translation parameters (the translation in direction of vanishing point, v, is superfluous) and radius r_2. The direction of C_1 is directly given, since we have set the coordinate system from the rectified P_1 at the origin. The direction for C_2 can be calculated knowing that the planes Π (equation 3) from camera P_1 defined by the projected silhouettes are tangents to the cylinder, and consequently their normals are orthogonal to the cylinder vanishing point. This vanishing point is calculated as the cross product of the two normals. In the next sections we show how to estimate the remaining variables.

3.2 Estimation of relative rotation

We will now show how the rotation of camera two can be estimated. Since we have a rectified system, the rotation R_{P_2} will be on the form

$$R_{P_2} = \begin{pmatrix} a \; 0 \; b \\ 0 \; 1 \; 0 \\ -b \; 0 \; a \end{pmatrix}, \quad a^2 + b^2 = 1 \tag{7}$$

Now, this rotation should move the direction of cylinder C_2 so that it coincides with the vanishing point of the corresponding silhouette lines of C_2 in view 2, i.e.

$$\Pi_{P_2C_2}v = P_2^T l_{P_2C_2}v = 0. \tag{8}$$

This gives an equation system with two equations which are linear in the unknowns a and b. In order to ensure a true rotation, the variables have to be normalised to enforce that $a^2 + b^2 = 1$. This additional constraint corresponds to the over-determinedness of the whole problem.

3.3 Solving for the remaining variables

In total we now have six unknowns remaining to be estimated, three translations in space for P_2, two translations for C_2 and the radius of C_2. We use (4) to get an equation system of six equations to solve for our variables, where the cylinder dual for C_1 is given by

$$D_1 = \begin{bmatrix} -p_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & (p_z^2(1+l_x^2) - p_z^2) & p_z(1+l_x^2) \\ 0 & 0 & p_z(1+l_x^2) & (1+l_x^2) \end{bmatrix}. \tag{9}$$

Here $p_z = 1$ is the distance from P_1 to midpoint of C_1 and l_x is the value defining the line $l_x x + 1 = 0$ when C_1 is projected into P_1. The other cylinder dual is

$$D_2 = AC_0^{-1}A^T \tag{10}$$

where

$$C_0^{-1} = \begin{bmatrix} -r_2^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -r_2^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

and

$$A = \begin{bmatrix} R_{C_2} & t_{C_2} \\ 0 & 1 \end{bmatrix}. \tag{12}$$

The equation system to be solved is:

$$\begin{cases} l_{P_2C_11}^T P_2 D_1 P_2^T l_{P_2C_11} = 0 \\ l_{P_2C_12}^T P_2 D_1 P_2^T l_{P_2C_12} = 0 \\ l_{P_1C_21}^T P_1 D_2 P_1^T l_{P_1C_21} = 0 \\ l_{P_1C_22}^T P_1 D_2 P_1^T l_{P_1C_22} = 0 \\ l_{P_2C_21}^T P_2 D_2 P_2^T l_{P_2C_21} = 0 \\ l_{P_2C_22}^T P_2 D_2 P_2^T l_{P_2C_22} = 0 \end{cases} \tag{13}$$

Due to our parameterization, the first two equations are only functions of the $x-$ and z-coordinates of the translation of P_2. This means that we can solve these equations independently of the other variables. This system has in general four solutions, and we have constructed a fast solver using the Grobner basis solver generator from [27, 28].

The radius r_2 of cylinder C_2 only appears in quadratic form and if we consider r_2^2 as our variable all four following equations are linear in r_2^2, so it is easily eliminated. Left is a small system three equations in three unknowns, namely the two translation parameters of C_2 and the $y-$coordinate of the camera translation of P_2. Here we have constructed a solver based on the Grobner basis solver generator from [27, 28]. This solver gives in general eight solutions. This special system can also be solved using the E3Q3 solver from Kukelova et al. [25], which we have also tested. In total we get maximum $4 \cdot 8 = 32$ solutions from the collective solvers. To find the correct solution one may use additional knowledge, for example the projection lines of an additional cylinder, and a procedure for this is described in Section 4.

3.4 Degenerate cases

The previous sections describe the case when we have cylinders in general forms. In some cases, special geometric setups will cause the solvers to fail. One degenerate case is when we have truly parallel cylinders. This gives an under-determined system along the translation in direction of the cylinders. This special case calls for a reduced system of equations with fewer variables (a reduction to a 2D problem) and was addressed by Tegler et al. [46]. However, we will show in Section 5.4 that we may still apply the solvers to almost parallel cylinders, and not much deviation is needed for a functional model. Another degenerative case would be if either of the cameras is placed within the infinite extension of one of the cylinders. Then no tangent planes will pass through the camera center. When the center-lines of the cylinders intersect we introduce ambiguity of where the cameras may be placed. One may construct other, less obvious, cases but we have not made sufficient analysis to be able to present this in the paper.

4 Consensus estimation from two views

Our solver will give a number of hypotheses, and we would like to estimate which solution is the correct one. Typically in a RANSAC loop one would estimate models on minimal or small subsets of data, and then check this model on the rest of the data, and in the end choose the model with largest support. In our problem there are a number of ways that we can do this. First of all, since our solver is not strictly minimal, we can check already in the rotation estimation stage if the chosen correspondence is possible. In Section 3.2 we estimate the rotation parameters a and b linearly. However, in order for this to be a valid rotation $a^2 + b^2 = 1$. So checking the norm of (a, b) gives a constraint that can rule out false correspondences. Note that there could be possible false correspondences that still fulfill this constraint.

For the different solutions, we can use cheirality to rule out a number of false or geometrically invalid solution candidates. This now gives us a set of up to 32 models to work with.

When we have estimated our multiple camera models, we want to check these models on the rest of the data. We could for instance triangulate the other cylinders using the estimated cameras, and then check the reprojection errors. But we have opted for a faster and simpler check. For a rectified system, the cylinder has three unknown parameters, the center point and the radius. For two given views we have four constraints on these parameters. We have tested to algebraically eliminate the three unknowns from the four equations. This gives a final equation that we can evaluate for each cylinder to check if it is consistent with our estimated cameras. We found however that this is quite numerically difficult to use, and set thresholds. Instead we have tried a more geometric approach. We relax the assumption of our cylinder to have circular cross-section, and estimate two (camera axis parallel) radii (and eliminate the center point) from the four equations. We then check the ratio of these radii. If it is close to one, then the cylinder is consistent with the estimated cameras. This gives and efficient and geometrically valid way of checking inliers in RANSAC.

5 Results

We will in this section present experimental results both on synthetic data and real images from a number of different scenery. We will show that our method works well as an initialization module for an intended cylinder based SfM system. Our focus has not primarily been on the system aspects, but rather on thorough testing of the properties of the solver in a number of scenarios.

5.1 Synthetic data

We construct and use our solvers in Matlab. To test the numerical stability of the solvers we generated a large number of random problem instances, and ran the solvers based on this input data. These problem instances were generated without any noise in the measurements. For the E3Q3-solver [25] we use the implementation in [26]. The resulting errors, intrinsic to the solvers can be seen in Fig. 4. One can see that all solvers get close to machine precision. It should be noted that the E3Q3 solver is almost 10 times faster than the Grobner basis solver. To test the dependence of noise of the solvers and our problem formulation, we again generate a large number of synthetic problem instances. We then add Gaussian noise to the lines with increasing standard deviation. The results can be seen in Fig. 5. The noisy lines are constructed by finding the point on the line closest to the center point in the image. The two points one length unit away from the center point are calculated and Gaussian noise with standard deviation σ is added. The line through these points are the new lines used for calculation. Since the error is dependant on not only noise level, but distance between points to define the line, scale set in 3D, angle between cameras etc. the

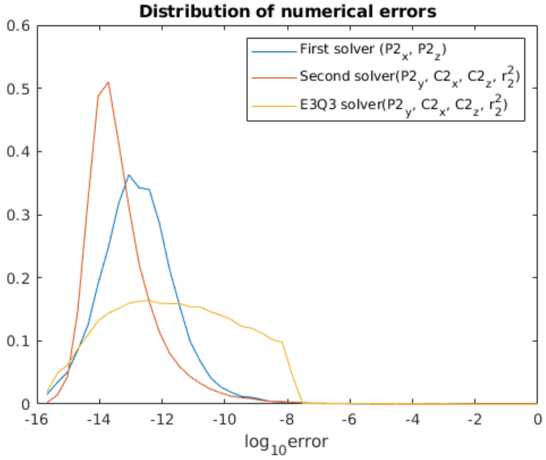

Fig. 4: Distribution of the log_{10}-error of our two solvers and the E3Q3-solver when applied to random synthetic data without noise.

most noteworthy in this experiment is the linear response to noise. The setup for this experiment is cylinders with diameter 1 meter, 10 meters from camera 1.

5.2 Roller coaster experiment

We now test our method on real data. We have two images with the same five cylinders visible. Here the cameras are calibrated. In order to get ground truth estimates for the cameras we ran COLMAP with standard settings [42, 43]. We use the estimated camera geometry as pseudo-ground truth for this experiment.

The silhouette lines for the two images were extracted manually. We run our algorithm for each combination of two different cylinders. Note that the procedure is not symmetric in the chosen cylinders so we get two different results for the same pair of cylinders. Example of re-projected result cylinders can be seen in Fig. 6. Results for all combinations can be found in Table 2. Here we can see that the performance can be sensitive to the chosen cylinder correspondences, but for a large amount of combination we get small errors.

5.3 Storm King experiment

We have in this experiment two images taken from the sculpture park Storm King, situated in New York, shown in Fig. 1. We manually extracted silhouette lines from the images. We then ran our minimal solver in a RANSAC manner on all possible silhouette correspondences. Here it is completely feasible to do exhaustive sampling due to the small number of line features. We used the consensus check described in Section 4. This enables us to get an accurate solution,

Table 2: Result table for roller coaster experiment. Radius is the estimated radius of cylinder two. The Relative error is the relative error of the radius of cylinder two. Rotation is the rotation error of camera P_2 in degrees and translation is the Euclidean distance between estimated and ground truth position of camera P_2. Cases cylinders $= [1 \quad 5]$ and cylinders $= [2 \quad 3]$ can be seen in figure 6. for comparison, the distance between cameras are 1.57m

Cylinder 1	Cylinder 2	Radius	Relative error	$\Delta\theta$ (deg)	Δxyz (m)
1	2	0.728	0.067	1.574	1.615
1	3	0.377	0.517	2.031	1.340
1	4	0.485	0.378	12.077	1.435
1	5	0.101	0.871	6.850	2.160
2	1	0.016	0.980	0.292	1.246
2	3	0.279	0.642	3.785	0.908
2	4	0.750	0.039	12.516	1.522
2	5	0.343	0.560	2.977	2.165
3	1	0.709	0.092	1.966	1.655
3	2	0.220	0.718	3.920	0.769
3	4	0.340	0.564	20.320	1.608
3	5	0.003	0.997	28.215	2.057
4	1	28.816	35.944	14.301	0.968
4	2	0.584	0.251	16.377	1.349
4	3	0.032	0.959	25.907	1.200
4	5	0.024	0.969	11.751	2.094
5	1	21.358	26.382	6.804	1.702
5	2	0.016	0.980	2.608	1.658
5	3	0.018	0.976	51.587	2.083
5	4	0.068	0.913	28.949	1.458

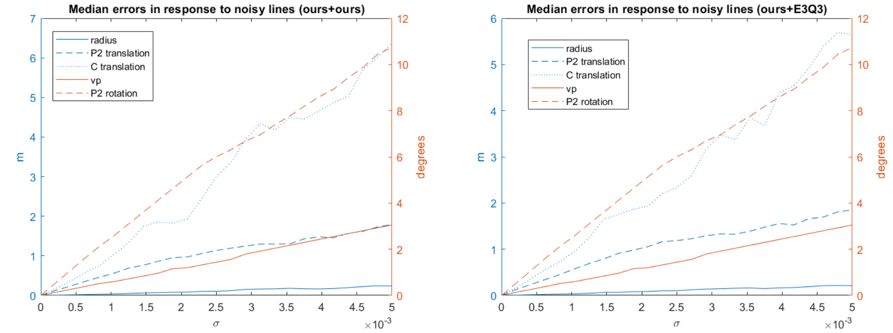

Fig. 5: Median errors for synthetic data in response to noise with increasing standard deviation. For the translations of P_2 and C_2 the error is the Euclidean distance between estimation and ground truth. Here vp stands for vanishing point and is the angular error between estimated and true v. The plot to the left is our solvers, right is our first solver followed by the E3Q3 solver.

that is consistent with all the cylinder silhouettes. In Fig. 7 the original images with reprojected lines are shown. The cyan lines correspond to the best chosen solution in the RANSAC loop. The other lines (in purple) are based on triangulated cylinders. The cylinders were triangulated using the estimated cameras using the method from [20]. The full 3D reconstruction is depicted to the right in Fig. 1. Note that all these results are without any additional bundle adjustment or non-linear refinement.

5.4 Forestry experiment

In this experiment we test how well our solver works in a challenging scenario. To the right in Fig. 8 two images of a forestry scene are shown. This data is from [19], with given ground truth cameras. Here we have non man-made structures (trees) that are not perfect cylinders, and they are close to parallel. We ran our solver on two example trees, and checked the results against the ground truth cameras. The reconstruction from a top view is shown to the left in Fig. 8. Also shown are the ground truth cameras. One can see that both position and camera direction are quite accurate in this case. Note that the results are only based on the minimal solver, without any additional non-linear refinement. Since we only use two cylinders here, we chose the solution from the minimal solver that was closest to the ground truth cameras. In Table 3 numerical results on the errors are given. We also compare with the solver by Tegler et al. [46] that assumes parallel cylinders. We get better rotation estimates, but slightly worse planar translation estimate. Note however that since their method assumes parallel cylinders, they do not get a camera height estimate at all.

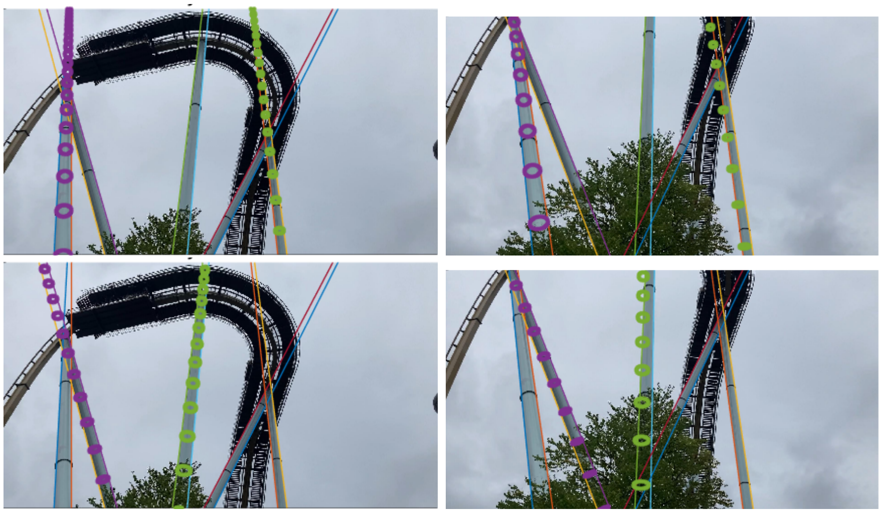

Fig. 6: Results of roller coaster experiment. The projections of the estimated cylinders C_1 and C_2, using the estimated cameras P_1 and P_2 are shown in green and purple. The original input lines are also plotted. Top and bottom show two successful runs based on two different cylinder inputs.

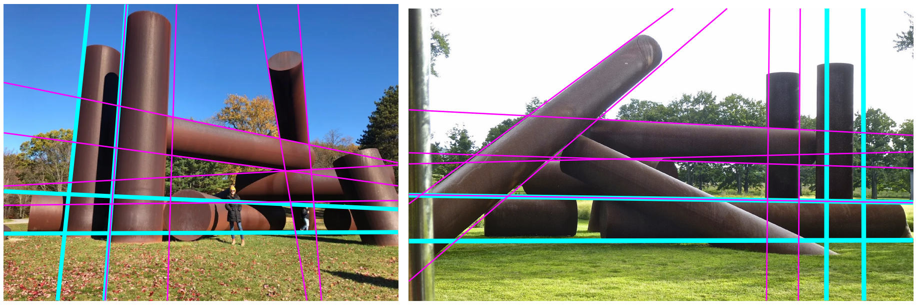

Fig. 7: The two images from the Storm King experiment. In cyan, the reprojected silhouettes from the estimated two cylinders and cameras are shown. The purple lines show reprojections from triangulated cylinders, based on the minimally estimated camera geometry.

Table 3: Errors for the estimates in the forestry experiment. We show viewing direction errors compared to the ground truth cameras, the planar position error and the height error. The scale in meters is given by the ground truth cameras. Results are given for the proposed method and compared to the method of Tegler et al. [46]. Note that since they assume parallel cylinders they do not get a height estimate of the cameras.

	$\Delta\theta$ (deg)	Δxy (m)	Δz (m)
Proposed	**6.0**	0.18	**0.48**
Parallel solver [46]	10.8	**0.085**	-

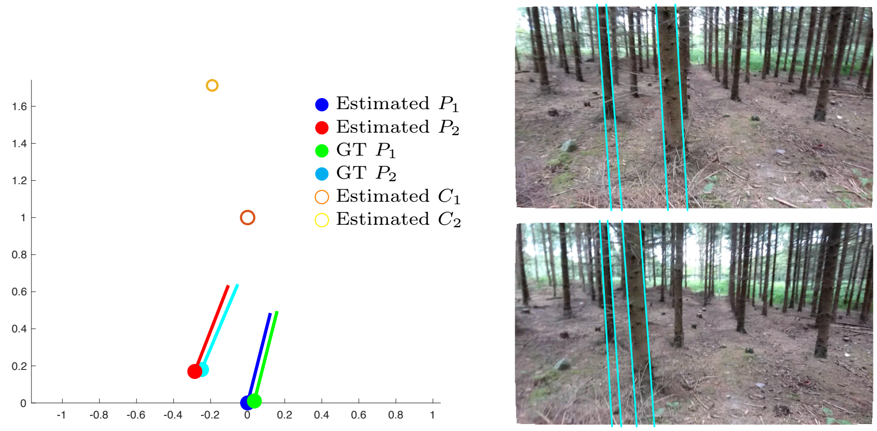

Fig. 8: To the left our reconstruction of the forestry scene, seen from above. The two estimated cylinder outlines are shown circles, in yellow and orange respectively. The two estimated cameras are shown in blue and red, where the lines indicate the viewing directions of the cameras. The corresponding ground truth cameras are shown in green and cyan. To the right are the two images, with reprojected silhouettes, given the estimated cameras and cylinders.

6 Conclusion

In this paper we have proposed minimal solvers for relative pose estimation based on projected silhouettes of 3D cylinders. We have shown how these solvers can be used to reconstruct cylinders in 3D, using only the projected silhouette lines in two views. Cylinder representations enable robust SfM, that is more stable to environment changes, and also enable further semantic processing in down-stream tasks. For some cases, the proposed solver targets problems that are infeasible using traditional point features, due to e.g. completely opposing views with no correspondences.

Future work will involve implementing the methods on a system level, incorporating e.g. cylinder line feature detectors and bundle adjustment. In this paper we have targeted the fully calibrated case, but it could be possible to look at cases where only partial camera calibration is known.

Acknowledgments. This work was partially supported by the strategic research project ELLIIT and the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg (KAW) Foundation.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Åström, K., Cipolla, R., Giblin, P.J.: Generalised epipolar constraints. International Journal of Computer Vision (1999), to be published
2. Åström, K., Kahl, F.: Motion estimation in image sequences using the deformation of apparent contours. IEEE Trans. Pattern Analysis and Machine Intelligence **21**(2), 114–127 (1999)
3. Barath, D., Cavalli, L., Pollefeys, M.: Learning to find good models in ransac. In: CVPR (2022)
4. Barath, D., Matas, J.: Graph-cut RANSAC. In: CVPR (2018), `https://github.com/danini/graph-cut-ransac`
5. Barath, D., Noskova, J., Ivashechkin, M., Matas, J.: Magsac++, a fast, reliable and accurate robust estimator. In: CVPR (2020)
6. Barath, D., Noskova, J., Matas, J.: Marginalizing sample consensus. IEEE TPAMI (2021)
7. Biederman, I.: Human image understanding: Recent research and a theory. Computer vision, graphics, and image processing **32**(1), 29–73 (1985)
8. Caprile, B., Torre, V.: Using vanishing points for camera calibration. Intl. J. of computer vision **4**(2), 127–139 (1990)
9. Cavalli, L., Barath, D., Pollefeys, M., Larsson, V.: Consensus-adaptive ransac. arXiv preprint arXiv:2307.14030 (2023)
10. Cavalli, L., Pollefeys, M., Barath, D.: Nefsac: Neurally filtered minimal samples. In: ECCV (2022)
11. Chum, O., Matas, J.: Matching with prosac-progressive sample consensus. In: CVPR (2005)
12. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25. pp. 236–243. Springer (2003)
13. Dai, X., Yuan, X., Gong, H., Ma, Y.: Fully convolutional line parsing. arXiv preprint arXiv:2104.11207 (2021)
14. De Ma, S.: Conics-based stereo, motion estimation, and pose determination. International journal of computer vision **10**(1), 7–25 (1993)
15. Ding, Y., Astermark, J., Oskarsson, M., Larsson, V.: Noisy one-point homographies are surprisingly good. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5125–5134 (2024)
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM (1981)
17. Frosio, I., Alzati, A., Bertolini, M., Turrini, C., Borghese, N.A.: Linear pose estimate from corresponding conics. Pattern recognition **45**(12), 4169–4181 (2012)
18. Frosio, I., Turrini, C., Alzati, A.: Camera re-calibration after zooming based on sets of conics. The Visual Computer **32**(5), 663–674 (2016)
19. Gummeson, A., Engman, J., Åström, K., Oskarsson, M.: Fast and efficient minimal solvers for quadric based camera pose estimation. In: Proceedings of the International Conference on Pattern Recognition (2022)
20. Gummeson, A., Oskarsson, M.: Robust and accurate cylinder triangulation. In: Scandinavian Conference on Image Analysis. pp. 451–466. Springer (2023)
21. Josephson, K., Kahl, F.: Triangulation of points, lines and conics. Journal of Mathematical Imaging and Vision **32**(2), 215–225 (2008)

22. Kahl, F., Heyden, A.: Using conic correspondences in two images to estimate the epipolar geometry. In: Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271). pp. 761–766. IEEE (1998)
23. Kaminski, J., Shashua, A.: Multiple view geometry of algebraic curves. International journal of computer vision (2003)
24. Keppel, E.: Approximating complex surfaces by triangulation of contour lines. IBM Journal of research and development **19**(1), 2–11 (1975)
25. Kukelova, Z., Heller, J., Fitzgibbon, A.: Efficient intersection of three quadrics and applications in computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1799–1808 (2016)
26. Larsson, V.: `https://github.com/vlarsson/re3q3`
27. Larsson, V., Åström, K., Oskarsson, M.: Efficient solvers for minimal problems by syzygy-based reduction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 820–829 (2017)
28. Larsson, V., Oskarsson, M., Åström, K., Wallis, A., Kukelova, Z., Pajdla, T.: Beyond gröbner bases: Basis selection for minimal solvers. In: CVPR (2018)
29. Lebeda, K., Matas, J., Chum, O.: Fixing the locally optimized RANSAC–full experimental evaluation. In: BMVC (2012)
30. Ma, S., Li, L.: Ellipsoid reconstruction from three perspective views. In: Proceedings of 13th International Conference on Pattern Recognition. vol. 1, pp. 344–348. IEEE (1996)
31. Marr, D., Nishihara, H.K.: Representation and recognition of the spatial organization of three-dimensional shapes. Proceedings of the Royal Society of London. Series B. Biological Sciences **200**(1140), 269–294 (1978)
32. Matas, J., Chum, O.: Randomized ransac with sequential probability ratio test. In: ICCV (2005)
33. Mei, J., Zhang, D., Ding, Y.: Monocular vision for pose estimation in space based on cone projection. Optical Engineering **56**(10), 103108 (2017)
34. Mudigonda, P.K., Jawahar, C., Narayanan, P.: Geometric structure computation from conics. In: ICVGIP. pp. 9–14. Citeseer (2004)
35. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
36. Navab, N., Appel, M.: Canonical representation and multi-view geometry of cylinders. International Journal of Computer Vision **70**(2), 133–149 (2006)
37. Pautrat, R., Barath, D., Larsson, V., Oswald, M.R., Pollefeys, M.: Deeplsd: Line segment detection and refinement with deep image gradients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17327–17336 (2023)
38. Pautrat, R., Lin, J.T., Larsson, V., Oswald, M.R., Pollefeys, M.: Sold2: Self-supervised occlusion-aware line description and detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11368–11378 (2021)
39. Piedade, V., Miraldo, P.: Bansac: A dynamic bayesian network for adaptive sample consensus. In: ICCV (2023)
40. Quan, L.: Conic reconstruction and correspondence from two views. IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(2), 151–160 (1996)
41. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al.: Benchmarking 6dof outdoor visual localization in changing conditions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8601–8610 (2018)

42. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
43. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016)
44. Semple, J.G., Kneebone, G.T.: Algebraic projective geometry. Clarendon Press (1979)
45. Sun, J., Cheng, X., Fan, Q.: Camera calibration based on two-cylinder target. Optics Express **27**(20), 29319–29331 (2019)
46. Tegler, E., Engman, J., Gillsjö, D., Flood, G., Larsson, V., Oskarsson, M., Åström, K.: The multi-view geometry of parallel cylinders. In: Scandinavian Conference on Image Analysis. pp. 482–499. Springer (2023)
47. Torr, P.H., Nasuto, S.J., Bishop, J.M.: Napsac: High noise, high dimensional robust estimation-it's in the bag. In: BMVC (2002)
48. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: a line segment detector. Image Processing On Line **2**, 35–55 (2012)
49. Wei, T., Patel, Y., Shekhovtsov, A., Matas, J., Barath, D.: Generalized differentiable ransac. In: ICCV (2023)
50. Winkler, A., Zagar, B.: A curve fitting method for extrinsic camera calibration from a single image of a cylindrical object. Measurement science and technology **24**(8), 084001 (2013)
51. Xu, Y., Xu, W., Cheung, D., Tu, Z.: Line segment detection using transformers without edges. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4257–4266 (2021)