SHAPE: LINEAR-TIME CAMERA POSE ESTIMATION WITH QUADRATIC ERROR-DECAY

Alireza Ghasemi *Adam Scholefield* *Martin Vetterli*

School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne

ABSTRACT

We propose a novel camera pose estimation or perspective-n-point (PnP) algorithm, based on the idea of consistency regions and half-space intersections. Our algorithm has linear time-complexity and a squared reconstruction error that decreases at least quadratically, as the number of feature point correspondences increase.

Inspired by ideas from triangulation and frame quantisation theory, we define consistent reconstruction and then present SHAPE, our proposed consistent pose estimation algorithm. We compare this algorithm with state-of-the-art pose estimation techniques in terms of accuracy and error decay rate. The experimental results verify our hypothesis on the optimal worst-case quadratic decay and demonstrate its promising performance compared to other approaches.

Index Terms— Perspective-n-point problem, camera pose estimation, multi-view geometry, triangulation, camera resectioning.

1. INTRODUCTION

Camera pose estimation, or the perspective-n-point (PnP) problem, aims to determine the pose (location and orientation) of a camera, given a set of correspondences between 3-D points in space and their projections on the camera sensor [1]. The problem has applications in robotics, odometry [2], and photogrammetry, where it is known as space resection [3].

In the simplest case, one can use an algebraic closed-form solution to derive the camera pose from a set of minimal 3D-to-2D correspondences. Usually, three correspondences are used and hence these algorithms are called perspective-3-point or P3P methods [4, 5].

When there is a redundant set of points available (more than three), the most straightforward solution is to use robust algorithms, such as RANSAC, which run P3P (or its variants) on minimal subsets of correspondences [6]. However, such algorithms suffer from low accuracy, instability and poor noise-robustness, due to the limited number of points.

An alternative approach is to directly estimate the camera pose, using an objective function, such as the ℓ_2-norm of the reprojection error, defined over all available point correspondences [7, 8].

Minimisation of the ℓ_2-norm leads to the maximum likelihood estimator, if we assume a Gaussian noise model. However, the main drawback of the ℓ^2-norm is that its resulting cost function is non-convex and usually has a lot of local minima [9]. This forces us to use iterative algorithms that are reliant on a good initialisation [10].

The shortcomings of the ℓ_2-norm have encouraged researchers to consider using other norms, such as the ℓ_∞-norm [11]. The main advantage of the ℓ_∞-norm is that its minimisation can be formulated as a quasi-convex problem and solved using Second-Order Cone Programming (SOCP) [9, 12]. This leads to a unique solution, however SOCP techniques are computationally demanding and rely on the correct tuning of extra parameters [13].

There is an interesting, well known, duality between pose estimation and triangulation, which allows common algorithms to be used for both problems [8, 14]. Triangulation estimates the location of a point given its projection in a number of calibrated cameras [15]. Various triangulation algorithms exist, which once again mostly relying on minimising the reprojection error [9]. To see the duality, notice that in both cases we have a set of projections and we want to estimate the location of an object of interest; i.e., the camera, in pose estimation, and the point, in triangulation.

In this paper, we propose a fundamentally novel approach to the camera pose estimation problem. Under certain assumptions, this leads to the optimal estimate for both the camera location and orientation, and a consistency region, where the true camera pose must lie. Our algorithm has a linear time-complexity, allowing it to be used efficiently with a large number of points. Moreover, exploiting the duality between camera pose estimations and triangulation, we use our earlier work [16, 17] to show that the expected error decays at least quadratically as we increase the number of available point correspondences.

In the rest of this paper, we formally define the problem and our imaging setup. For simplicity, we limit our discussion to 1-D vision systems, in which points in $\mathbb{R}^2$ are mapped to points on a 1-D image sensor. Although this special case is independently important, e.g. in developing autonomous guided vehicles or planar motion [14], the extension to 3-D vision is straightforward. In addition, we assume that the points' locations are known exactly.

After defining the problem setup, we propose our algo-

This work was supported by the ERC Advanced Grant—Support for Frontier Research—SPARSAM Nr: 247006.

A. Ghasemi is additionally supported by a Qualcomm Innovation Fellowship.

rithm, coined SHAPE (Sequential Half-Space Aggregation for Pose Estimation). We start with the simpler case of known camera orientation (i.e. estimating only the location of the camera) and then extend the algorithm to compute the orientation as well. Finally, we present experimental results comparing our algorithm to state-of-the-art techniques. The results verify our hypothesis that the worst-case error decay rate of our algorithm is quadratic and demonstrate its promising performance compared to other approaches.

2. PROBLEM SETUP

We now introduce the digital pinhole camera model which will be assumed throughout this paper. Our aim is to estimate the orientation θ and location $\mathbf{t} = (t_x, t_z)$ of a camera having a resolution of N pixels, given M 2D-to-1D correspondences between points $\mathbf{s}_i \in \mathbb{R}^2$, and their pixelised projections q_i on the camera.

As depicted in Fig. 1, we assume that the i-th point source $\mathbf{s}_i$, is projected to the position p_i on the camera's image plane before being quantised to q_i, the centre of the corresponding pixel. As well as modelling the finite pixel width, q_m also models the finite sensor width, by obtaining the value $\sim$ if the projected point lies outside the field of view. Later we will consider algorithms that take the quantised feature projections q_i and estimate the pose (t_x, t_z, θ) of the camera.

As shown in Fig. 1, the camera is centred at $\boldsymbol{t}$ and orientated θ radians anti-clockwise from the global coordinate system. With this notation, the projected point p_i is given by[1]

$$p_i = f\frac{(s_{i,x} - t_x)\cos\theta + (s_{i,z} - t_z)\sin\theta}{(s_{i,z} - t_z)\cos\theta - (s_{i,x} - t_x)\sin\theta}. \tag{1}$$

Here, f is the focal length of the camera and $(s_{i,x}, s_{i,z})$ and (t_x, t_z) are the coordinates of the i-th point and camera centre, respectively, with respect to a global coordinate system.

The quantised point, q_i, is given by $q_i = Q_\Lambda(p_i)$, where Q_Λ is the quantisation function defined as[2]

$$Q_\Lambda(y) = \begin{cases} \left\lfloor \frac{y}{w} \right\rfloor w + \frac{w}{2} & -\frac{\tau}{2} \le y \le \frac{\tau}{2}, \\ \sim & \text{otherwise.} \end{cases} \tag{2}$$

In (2), $\Lambda = \{\tau, w\}$ encapsulates the sensor width τ and the pixel width $w = \frac{\tau}{N}$, which define the quantisation error.

In the rest of the paper, we will be interested in PnP algorithms that take the pixelised projections q_i and the true locations of the M point sources $\boldsymbol{s}_i$, and produce an estimate of the camera pose ($\boldsymbol{t}$, θ).

[1]Here, and in the rest of this paper, we have assumed perspective projection, which is standard in most imaging applications. However, we can derive similar results for other camera models. In fact, orthogonal projection have been extensively studied in the quantisation literature [18, 19] and also in image processing [16].

[2]This definition is valid when N is even. For odd N, $Q_\Lambda(y) = \begin{cases} \text{sign}(y)\left\lfloor \frac{|y|}{w} + \frac{1}{2} \right\rfloor w & -\frac{\tau}{2} \le y \le \frac{\tau}{2}, \\ \sim & \text{otherwise.} \end{cases}$

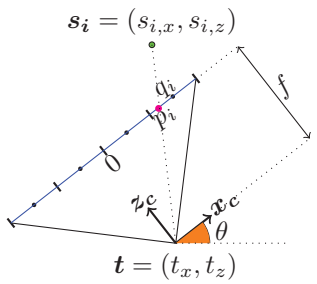

Fig. 1: The acquisition setup for a pinhole camera with a resolution of four pixels, acquiring a point source $\boldsymbol{s}$.

3. SHAPE: SEQUENTIAL HALF-SPACE AGGREGATION FOR POSE ESTIMATION

We now describe the proposed pose estimation algorithm, denoted SHAPE. We will first assume that we know the camera's orientation before considering the more general case.

3.1. Localising a Camera with Known Orientation

We would like to see how each point source constrains the location of the camera. Given a quantised projection q_i, we know that the true projected point p_i satisfies

$$q_i - \frac{w}{2} \le p_i \le q_i + \frac{w}{2}, \tag{3}$$

where w is the width of a pixel. Combining (1) with (3) and rearranging yields

$$a_i t_x + b_i t_z + c_i \ge 0, \tag{4}$$

and

$$a'_i t_x + b'_i t_z + c'_i \le 0. \tag{5}$$

Here a_i, a'_i, b_i, b'_i, c_i, and c'_i are defined as

$$\begin{aligned}
a_i &= f\cos\theta - (q_i + \tfrac{w}{2})\sin\theta, \\
b_i &= f\sin\theta - (q_i + \tfrac{w}{2})\cos\theta, \\
c_i &= (q_i + \tfrac{w}{2})(s_{i,z}\cos\theta + s_{i,x}\sin\theta) - f s_{i,x}\cos\theta - f s_{i,z}\sin\theta, \\
a'_i &= f\cos\theta - (q_i - \tfrac{w}{2})\sin\theta, \\
b'_i &= f\sin\theta - (q_i - \tfrac{w}{2})\cos\theta, \\
c'_i &= (q_i - \tfrac{w}{2})(s_{i,z}\cos\theta + s_{i,x}\sin\theta) - f s_{i,x}\cos\theta - f s_{i,z}\sin\theta.
\end{aligned}$$

Therefore, assuming we know the orientation of the camera, each point source constrains the camera location to lie between two half-spaces; i.e., within a semi-infinite triangle.

When there are multiple points, then there is a semi-infinite triangular region for each point and the camera must lie in their intersection. Therefore, we need to compute the intersection of all half-spaces which produces a polygon where the camera centre $\boldsymbol{t}$ must lie. Every point within this polygon is consistent with the projections and has an equal chance of

being the true camera location, as long as the feature detection accuracy is uniformly bounded within the pixel limits. SHAPE selects the point that leads to the minimum mean squared distance to all points inside the consistency polygon. This point is the centre of mass of the consistency region and can be computed very efficiently in constant time.

Figure 2 visualises one example of applying our proposed algorithm to a camera positioning problem with three known points and their projections in a 6-pixel camera.

3.2. Simultaneous Estimation of Camera Location and Orientation

When the camera orientation is unknown, we have an extra dimension which leads to a 3D solution space (t_x, t_z, θ). We previously analysed the t_x-t_z slice for the true camera orientation and saw that there was a polygon, which led to estimates of the camera location that were consistent with the measurements. Let us now consider slices for an arbitrary angle θ. As θ changes, each half space rotates around its point source. For many angles, there is no common intersection between the half spaces but there is a range of angles for which the intersection creates a polygon. Thus there is a 3D shape created by the union of all these slices, containing all estimates consistent with the measurements.

We would like to find the centre of mass of this 3D shape, leading to an estimate of the location and orientation. This 3D region is neither a polytope nor convex and calculating its centroid is not trivial. We can find an accurate approximation by taking the weighted average of a finite number of slices.

More precisely, suppose the camera orientations are discretised to $\Theta = \{0, \frac{2\pi}{k}, \frac{4\pi}{k}, \ldots, 2\pi\}$, and that, for every orientation $\alpha \in \Theta$, we have computed the location-consistency region $\mathcal{R}_\alpha$ and its centre of mass $\mathcal{C}(\mathcal{R}_\alpha)$. We approximate the centre of mass of the 3-D consistency shape as

$$(\hat{t}_x, \hat{t}_z, \hat{\theta}) = \frac{\sum_{\alpha\in\Theta} \mathcal{A}(\mathcal{R}_\alpha)\mathcal{C}(\mathcal{R}_\alpha)}{\sum_{\alpha\in\Theta} \mathcal{A}(\mathcal{R}_\alpha)}, \tag{6}$$

where $\mathcal{A}(\mathcal{R}_\alpha)$ is the area of the location-consistency region $\mathcal{R}_\alpha$. This is SHAPE's final estimate of the camera pose.

3.3. Time Complexity of the SHAPE Algorithm

The core part of the SHAPE algorithm is a series of $2M$ half-space intersection (M is number of points). Since the area between half-spaces is a triangle in the finite case, we can solve this problem using polygon intersection algorithms.

The worst-case time-complexity for polygon and half space intersection is $\mathcal{O}(M \log M)$, in the general case [20]. However, since the point-consistency regions form convex polygons, which can be intersected in $\mathcal{O}(V_1 + V_2)$ time, where V_1 and V_2 are the number of vertices of the two polygons [21]. Therefore, a series of intersections between convex polygons can be computed in $\mathcal{O}(MV_{max})$, where V_{max} is the maximum number of vertices of an intermediate polygon. In our case, it can be intuitively shown that the number of vertices of intermediate polygons do not grow with the number intersected polygons, for any practical number of points. Therefore, we can bound V_{max} and hereby reach a $\mathcal{O}(M)$, or linear time-complexity for our algorithm.

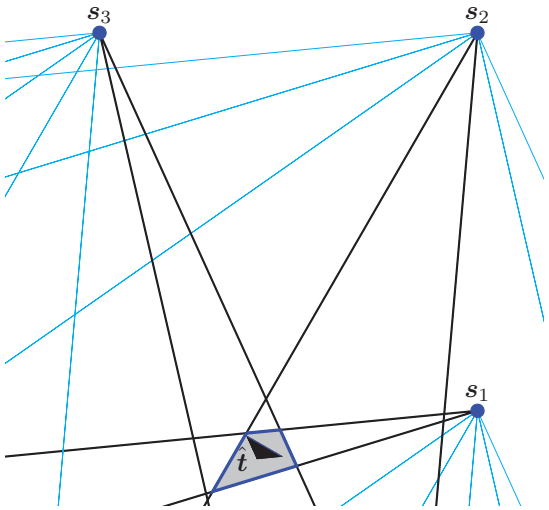

Fig. 2: An example of using polygon intersections for estimating the camera pose. Boundaries of half-spaces are depicted as black lines. The intersection of half-spaces is the light-grey polygon. The centroid of this polygon is the reconstructed centre $\hat{t}$ of the camera.

3.4. Error Decay Rate of the SHAPE Algorithm

We would like to know if our algorithm converges to the true latent value, as the number of point correspondences tend to infinity, and how fast the error decays.

By adapting results from frame quantisation theory [18], we have recently shown a quadratic error decay rate for the dual triangulation problem with circular and linear camera arrays [16, 17].

Triangulation with a linear camera array is equivalent to the PnP problem with collinear feature points and a known camera orientation. It follows that, if the shortest distance between the camera and the line where the points lie does not exceed $\frac{fb}{2w}$, where b is the largest distance between the points, the error of the SHAPE algorithm decays quadratically as the number of feature point correspondences increases.

In general, however, if the points are distributed randomly, the error decays much faster, since the large consistency regions of the linear case are unlikely to occur. This is why we obtain a much faster error decay rate in practice, as can be seen in Fig. 6. Moreover, the large consistency regions generated by collinear and coplanar point sets, explain the difficulties traditionally seen by PnP algorithms in these cases.

4. SIMULATION RESULTS

To assess our algorithm, we have randomly generated a set of camera poses and for each one randomly added points within

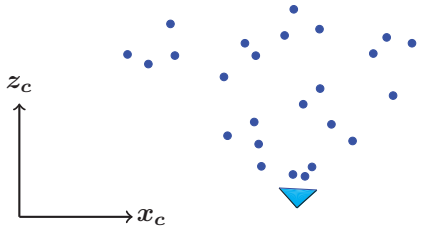

Fig. 3: An example of point sets used in the experiments, as well as the latent, true camera pose.

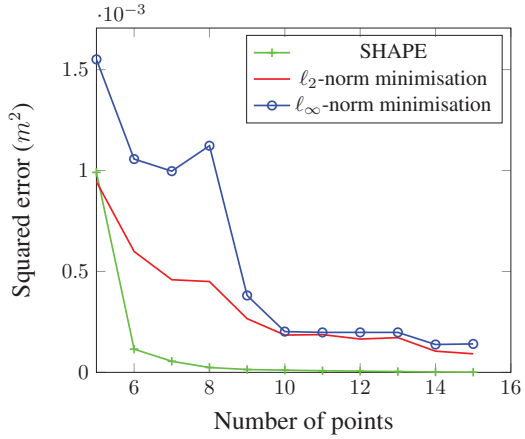

Fig. 4: Results of camera location estimation with fixed orientation.

the cameras field of view. Figure 3 depicts a sample configuration used in the experiments. The camera has a resolution of 320 pixels and a field of view of 90 degrees.

We have compared the result of SHAPE to the results of minimising the reprojection error measured with the ℓ_2 and ℓ_∞ norms. For the ℓ_2-norm, the cost function is non-convex, resulting in multiple minima. However, we have calculated the global minimum using a brute-force strategy, in order to justify the selected criteria and not the specific methods.

Figure 4 depicts the averaged result of camera pose estimation using the three approaches. Here we have assumed that the camera orientation is known and given as an input to the algorithms. Although initially, i.e. for a small number of correspondences, the results of SHARP are worse than the norm-based methods, our algorithm converges much faster and the difference in accuracy becomes more evident as the number of correspondences increases. Moreover, we can see that the error of SHARP converges to 0, which is not the case for norm-based approaches.

Figure 5 depicts the pose estimation results for the case of unknown camera orientation. We can see that similar results apply to this case as well.

To have a better visualisation of the convergence rate of the algorithms, we have additionally depicted a log-log plot of the error values, in Figure 6. In the log-log plot, convergence rates correspond to the gradients. Therefore, it is easily verified that the error decay rate of the SHAPE algorithm is faster than other approaches.

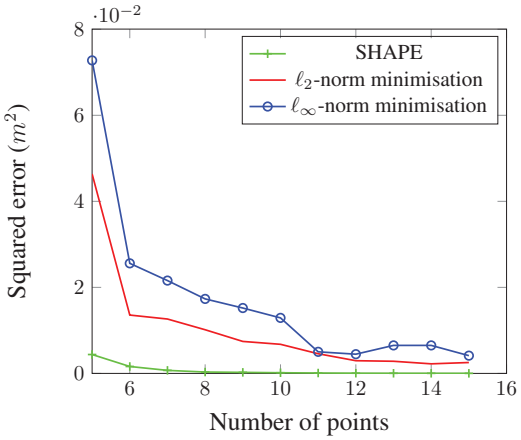

Fig. 5: Results of full camera pose estimation.

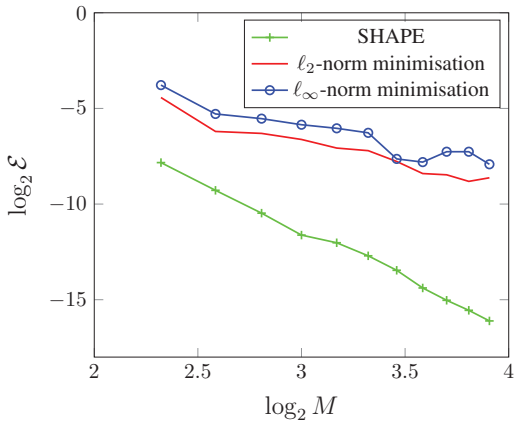

Fig. 6: Log-log plot visualising the convergence rate of different algorithms.

5. CONCLUSION

We have proposed the SHAPE algorithm, a fundamentally novel approach toward solving the PnP or camera pose estimation problem, using consistency regions and half-space intersections. We showed that SHAPE converges to zero error as the number of point correspondences tends to infinity. Moreover, we have shown that our algorithm benefits from a linear time complexity.

Further work needs to be done to handle incorrect point correspondances and uncertainty in the feature point locations. We can develop novel outlier-detection techniques to remove points with large error and then increase the pixel width for smaller error values. Another possible extensions would be to incorporate other camera models and also relax some known parameters of the problem, such as the focal length or projections of some points.

6. REFERENCES

[1] Y. Zheng, Y. Kuang, S. Sugimoto, K. Ästrom, and M. Okutomi, "Revisiting the pnp problem: A fast,

general and optimal solution," in *IEEE International Conference on Computer Vision*. 2013, pp. 2344–2351, IEEE.

[2] L. Quan and Z. Lan, "Linear n-point camera pose determination," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 8, pp. 774–780, 1999.

[3] B. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International Journal of computer vision*, vol. 13, no. 3, pp. 331–356, 1994.

[4] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 2969–2976, IEEE.

[5] L. Ferraz, X. Binefa, and F. Moreno-Noguer, "Very fast solution to the PnP problem with algebraic outlier rejection," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 501–508.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[7] F. Lu and R. Hartley, "A fast optimal algorithm for ℓ_2 triangulation," in *Asian Conference on Computer Vision (ACCV)*, pp. 279–288. Springer, 2007.

[8] Fredrik Kahl, Sameer Agarwal, Manmohan Krishna Chandraker, David Kriegman, and Serge Belongie, "Practical global optimization for multiview geometry," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 271–284, 2008.

[9] R. Hartley and F. Kahl, "Optimal algorithms in multiview geometry," in *Asian Conference on Computer Vision (ACCV)*, pp. 13–34. Springer, 2007.

[10] L. Kang, L. Wu, and Y.-H Yang, "Robust multi-view ℓ_2 triangulation via optimal inlier selection and 3-D structure refinement," *Pattern Recognition*, vol. 47, no. 9, pp. 2974–2992, 2014.

[11] F. Kahl and R. Hartley, "Multiple-view geometry under the ℓ_∞-norm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 9, pp. 1603–1617, 2008.

[12] H. D. Mittelmann, "An independent benchmarking of SDP and SOCP solvers," *Mathematical Programming*, vol. 95, no. 2, pp. 407–430, 2003.

[13] Y. Dai, H. Li, M. He, and C. Shen, "Smooth approximation of ℓ_∞-norm for multi-view geometry," in *Digital Image Computing: Techniques and Applications (DICTA)*., pp. 339–346. IEEE, 2009.

[14] Kalle Åström, Olof Enquist, Carl Olsson, Fredrik Kahl, and Richard Hartley, "An ℓ_∞ approach to structure and motion problems in 1D-vision," in *IEEE International Conference on Computer Vision, 2007*. IEEE, 2007, pp. 1–8.

[15] R. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[16] A. Ghasemi, A. Scholefield, and M. Vetterli, "On the accuracy of point localisation in a circular camera-array," in *IEEE International Conference on Image Processing (ICIP)*, 2015.

[17] A. Ghasemi, A. Scholefield, and M. Vetterli, "Consistent and optimal triangulation: Analysis and algorithms," *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[18] V. K. Goyal, M. Vetterli, and N. T. Nguyen, "Quantized overcomplete expansions in $\mathbb{R}^n$: analysis, synthesis, and algorithms," *Information Theory, IEEE Transactions on*, vol. 44, no. 1, 1998.

[19] Z. Cvetkovic, "Source coding with quantized redundant expansions: accuracy and reconstruction," in *Proceedings of DCC 99*. IEEE, 1999, vol. 1.

[20] F. P. Preparata and D. E. Muller, "Finding the intersection of n half-spaces in time $\mathcal{O}(n \log n)$," *Theoretical Computer Science*, vol. 8, no. 1, pp. 45–55, 1979.

[21] T. G. Toussaint, "A simple linear algorithm for intersecting convex polygons," *The Visual Computer*, vol. 1, no. 2, pp. 118–123, 1985.