

3D Motion Estimation By Evidence Gathering

Anas Abuzaina, Mark S. Nixon, John N. Carter

School of Electronics and Computer Science,

Faculty of Physical Sciences and Engineering,

University of Southampton, UK

Email: anas.abuzaina@gmail.com, msn, jnc@ecs.soton.ac.uk

Abstract—In this paper we introduce an algorithm for 3D motion estimation in point clouds that is based on Chasles' kinematic theorem. The proposed algorithm estimates 3D motion parameters directly from the data by exploiting the geometry of rigid transformation using an evidence gathering technique in a Hough-voting-like approach. The algorithm provides an alternative to the feature description and matching pipelines commonly used by numerous 3D object recognition and registration algorithms, as it does not involve keypoint detection and feature descriptor computation and matching. To the best of our knowledge, this is the first research to use kinematics theorems in an evidence gathering framework for motion estimation and surface matching without the use of any given correspondences. Moreover, we propose a method for voting for 3D motion parameters using a one-dimensional accumulator space, which enables voting for motion parameters more efficiently than other methods that use up to 7-dimensional accumulator spaces.

I. INTRODUCTION

The recent and considerable progress in 3D sensing technologies mandates the development of efficient algorithms to process the sensed data. Many of these algorithms are based on computing and matching of 3D feature descriptors in order to estimate point correspondences between 3D datasets. The dependency on 3D feature description and computation can be a significant limitation to many 3D perception tasks; the fact that there are a variety of criteria used to describe 3D features, such as surface normals and curvature, makes feature-based approaches sensitive to noise and occlusion. In many cases, such as smooth surfaces, computation of feature descriptors can be non-informative. Moreover, the process of computing and matching features requires more computational overhead than using points directly. On the other hand, there has not been much focus on employing evidence gathering frameworks to obtain solutions for 3D perception problems. Evidence gathering approaches, which use data directly, have proved to provide robust performance against noise and occlusion. More importantly, such approaches do not require initialisation or training, and avoid the need to solve the correspondence problem.

Accurate motion parameter estimation is based on invariant properties that can be inferred from the motion [1]. In this paper we investigate the Chasles' theorem [2] in the context of computer vision and develop evidence gathering methods employing rigid geometric constraints based on the analysis of corresponding points between 3D point clouds. More specifically, we are interested in estimating the rigid transformation between 3D datasets without the use of keypoint detection and

feature description and matching methods. We present a novel approach that estimates 3D motion parameters directly from the data by exploiting the geometry of rigid transformation using an evidence gathering technique in a Hough-voting-like approach. The developed algorithm is based on the observation that:

There is one and only one invariant axis that synthesises the rigid body transformation in 3D.

When the transformation is rotation only, this invariant axis is the rotation axis; when the transformation includes rotation and translation the invariant axis is the screw axis.

The algorithm tracks the motion between two frames (point clouds) and estimates a number of candidates for the screw axis using Chasles' theorem, and implements a voting algorithm to select the screw axis with the most likely parameters (3D position and orientation), then computes the rotation angle and linear displacement along the screw axis to estimate the full motion parameters. According to [3], kinematics theorems such as Chasles' theorem, have been widely used in kinematics and mechanics to analyse the nature and characteristics of rigid body motions. However, the implications of this theory for computer vision have not been properly considered.

It appears that this is the first research to use the kinematics theorems in an evidence gathering framework for motion estimation and surface matching without the use of any given correspondences and also without employing feature description and matching.

This paper is structured as follows: First we discuss related work, in terms of motion estimation using geometrical analysis, we also discuss methods that use a Hough voting scheme to vote for motion parameters. Second, we give a detailed overview of our algorithm with establishing the foundations for understanding its methodology. Then we validate and verify the theory and give experimental analysis. After that we show experimental results on real data and finally we evaluate the performance of the algorithm and conclude this paper.

II. RELATED WORK

A. Motion Estimation By Geometrical Analysis

A number of researchers have presented geometric constraint frameworks based on invariant properties of 3D rigid transformations to analyse transformation parameters. A key element of the geometric analysis is formulating algebraic constraints between a rotated and translated model and the observed features [4]. The constraints are aimed to define or refine the set of allowable positions for the object, consistent

with the observed features. In [5], the authors presented a method for kinematic registration that is based on specifications of line features. In this method, kinematic registration involves computation of the screw parameters of a motion from specified positions of geometric features of the moving body. The problem is formulated using a special complex of lines associated with kinematics namely the bisecting linear line complex. The methods presented in [6], [1], [3] are the first approaches to extend Chasles' screw motion concept to the estimation of motion parameters in computer vision. The authors formalise the relevant geometric properties of corresponding vectors by providing methods to construct the rotation point or axis and estimate a number of solutions for the point or axis. These methods analyse the geometric properties of vectors connecting corresponding feature points and their angular information, synthesised into a single coordinate frame, and define relevant constraints from the point of view of geometric invariants. These methods are superior to methods based on perspective and epipolar geometries because they are based on explicit expressions of distance between feature points and angle measurements rendering them appropriate to calibrate transformation parameters in vision applications.

Several geometric constraints have been identified [4] for estimating surface positions, given information about the shapes of planar surface patches and the relationships between the model and data patches. These constraints include position constraints, defined by algebraic relationships between vectors and data-to-model patch constraints which are based on aligning normals and on proximity and distance between points on the surface.

A method was introduced by [7] for motion estimation based on the analysis of rigid motion equations as expressed in geometric algebra framework. This method finds the correspondences of two 3D points sets by finding a certain 3D plane in a different space that satisfies certain geometric constraints using tensor voting. This method estimates the rigid transformation by producing a set of putative correspondences and uses them to populate joint spaces. If any rigid transformation exists in the set of correspondences, then four planes must appear in these spaces.

B. 3D Motion Estimation by Hough Voting

There has been a number of 3D motion estimation methods for 3D object recognition and registration that are based on voting schemes. More specifically, these methods employ a voting process to obtain an accurate transformation hypothesis from the matched features. According to [8], these methods estimate the 6 DoF pose between 3D data sets by first generating an empirical distribution of pose through the collation of a set of possible poses, or votes, which are often computed by matching local features, from a test object to those in a library with known pose. And then finding one or more best poses in the distribution which is the maximum (or peak) in a parameter space (also called Hough space). To summarise, Hough transform based methods for object recognition largely follow the same pipeline: features are detected and descriptors for the features are computed. The votes are then computed by matching features in the test data

Method	Parameters	Hough Space	Feature
Tsui & Chan [14]	R, t	$2 \times 3D$	Line segments
Hu [15]	t	3D	Edges
Khoshelham [13]	R, t, s	7D	Surface normals
Ashbrook et al.[9]	R, t	$2 \times 3D$	PGH
Knopp et al.[12]	R, s	4D	3D SURF
Wang et al.[16]	t	4D	HOG
Tombari & Stefano [11]	t	3D	SHOT
Our Method	R, t	1D	-

TABLE I: Comparison of Hough-based pose estimation methods. Second column shows pose parameters each method votes for (R : rotation, t : 3D position, s : scale), third column shows the dimensionality of the voting (Hough) space and fourth column shows the feature used for matching. (PGH \equiv Pairwise geometric histograms.)

with features from training data with ground truth scale and pose. These methods mainly differ in the type of features they employ, and also the metrics used to match the features. For example [9] use Bhattacharyya metric [10] to find the similarity between pairwise geometric histograms, while [11] use the SHOT features with Euclidean Distance metric. In the next step, pose hypotheses are generated from each matched feature pair, and votes are accumulated in a high dimensional Hough space. The dimensionality of the Hough space depends on the parameters being voted for; for example [12], votes for the 3D position parameters and scale; hence a 4D Hough space, while [13] votes for position, rotation and scale and hence a 7D Hough space. Finally a search for the highest votes (peaks) is implemented, and the peak parameters are considered to be the transformation hypothesis. Table I shows details of Hough based methods.

The algorithm presented in this paper has the advantage of having a one-dimensional space. This significantly reduces the complexity of allocating and searching the accumulator spaces.. This is achieved by representing the transformation as one axis and then voting for similarity between axes in terms of position and direction. and searching the high dimensional accumulator spaces that other methods suffer

III. MOTION ESTIMATION BY EVIDENCE GATHERING

In this section we introduce an algorithm which employs Chasles' theorem [2] to estimate the screw axis describing the rigid motion. The algorithm is divided into two stages, the evidence gathering algorithm and the voting algorithm. In the evidence gathering algorithm, Chasles's theorem is applied iteratively on two 3D point clouds, taking three correspondences at each iteration, and the estimated screw parameters are stored along with their three corresponding point pairs. The voting algorithm is applied on the stored axes to find the best estimation of the true screw axis representing the 3D motion.

A. Chasles' Theorem

Chasles' theorem states:

Any rigid body displacement in 3D can be produced by a rotation around a line followed by a translation along that line.

In other words, any motion in 3D can be represented by a rotation around a single axis and a translation parallel to that axis, this axis is referred to as the *screw axis* and as this

motion is reminiscent of the displacement of a screw, it is called a *screw motion*.

To define a screw 3D transformation between two frames, the screw axis \mathbf{h}_c , the rotation angle θ and the pitch η must be specified. The pitch is the ratio of the linear displacement to the rotation. For a pure translational motion, the screw axis is not unique; since the rotation is zero, the screw axis will have infinite pitch.

According to [1], given three 3D correspondences (p_i, p'_i) , $\{i = 1, 2, 3\}$, their correspondences vectors \mathbf{d}_i (Equation 2) can be evaluated. If the difference of correspondence vectors are not parallel, then the screw axis directional vector $\mathbf{h} = [h_x, h_y, h_z]^T$ can be uniquely determined as:

$$\mathbf{h} = \frac{(\mathbf{d}_2 - \mathbf{d}_1) \times (\mathbf{d}_3 - \mathbf{d}_1)}{\|(\mathbf{d}_2 - \mathbf{d}_1) \times (\mathbf{d}_3 - \mathbf{d}_1)\|} \quad (1)$$

This equation holds for all true correspondences, i.e. all vectors resulting from the cross product of vector difference between all true correspondences coincident at the same axis, which is the screw axis.

$$\mathbf{d}_i = p_i - p'_i = [x_i, y_i, z_i]^T - [x'_i, y'_i, z'_i]^T \quad (2)$$

Note that Equation 1 describes only the direction of the screw axis, and hence a point on the axis is required to complete its definition. Once \mathbf{h} is computed, any correspondences (p_i, p'_i) are projected on a plane perpendicular to \mathbf{h} , using the equation:

$$\underline{p} = (I - \mathbf{h}\mathbf{h}^T)p, \quad \underline{p}' = (I - \mathbf{h}\mathbf{h}^T)p' \quad (3)$$

where I is the identity matrix. Given two non-parallel projected correspondence vectors $(\underline{\mathbf{d}}_1, \underline{\mathbf{d}}_2)$ evaluated from two projected correspondences $(\underline{\mathbf{d}}_1 = \underline{p}_1 - \underline{p}'_1, \underline{\mathbf{d}}_2 = \underline{p}_2 - \underline{p}'_2)$, the critical point \mathbf{c} in 3D can be uniquely estimated by:

$$\mathbf{c} = \frac{p_2^T p_2 - p_2'^T p_2'}{2(\underline{\mathbf{d}}_2)^T H(\underline{\mathbf{d}}_1)} H(\underline{\mathbf{d}}_1) + \frac{p_1^T p_1 - p_1'^T p_1'}{2(\underline{\mathbf{d}}_1)^T H(\underline{\mathbf{d}}_2)} H(\underline{\mathbf{d}}_2) \quad (4)$$

Where the matrix H is given by:

$$H = \begin{bmatrix} 0 & -h_z & h_y \\ h_z & 0 & -h_x \\ -h_y & h_x & 0 \end{bmatrix} \quad (5)$$

Together the direction of the screw axis \mathbf{h} and the critical point \mathbf{c} define the screw axis \mathbf{h}_c in 3D space, the screw axis can be written as:

$$\mathbf{h}_c = \mathbf{c} + u\mathbf{h} \quad (6)$$

The screw axis \mathbf{h}_c has interesting geometric properties; according to [6], \mathbf{h}_c is equidistant to any 3D correspondences (p_i, p'_i) and the including angle between the lines passing through (p_i, p'_i) , perpendicular to and intersecting \mathbf{h}_c is equal to the rotation angle θ of the transformation, Figure 1.

Moreover, the critical point \mathbf{c} also has interesting geometric properties which are summarised by the following property: There is one and only one invariant point \mathbf{c} in 3D which is equidistant to the projected correspondences $(\underline{p}, \underline{p}')$ on the plane perpendicular to the screw axis $\mathbf{h} = (h_x, h_y, h_z)^T$ and the including angle between vectors $(\underline{p} - \mathbf{c})$ and $(\underline{p}' - \mathbf{c})$ is equal to the rotation angle θ around the screw axis. Moreover,

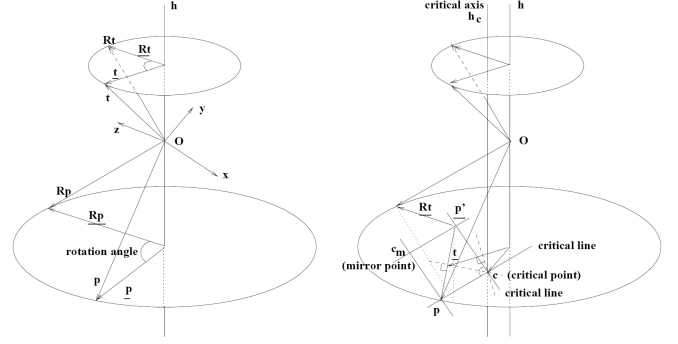


Fig. 1: The relationships among correspondence (p, p') , critical point \mathbf{c} , critical (screw) axis \mathbf{h}_c , and translation vector \mathbf{t} in 3D. Left, given a transformation (R, \mathbf{t}) the rotation axis (screw axis direction) \mathbf{h} and rotation angle θ are determined. Right, corresponding pairs (p, p') and the translation vector \mathbf{t} are projected on a plane perpendicular to \mathbf{h} . The critical point \mathbf{c} is determined at the intersection of the bisector lines $(\underline{p}, \underline{p}')$ and \mathbf{t} . The critical point determines the position of the critical axis \mathbf{h}_c where the vectors \underline{cp} and \underline{cp}' are of equal length with the including angle equal to the rotation angle θ of the transformation. Source [6].

the vertical bisector of the projected translation vector $\underline{\mathbf{t}}$ on this plane also intercepts at the same point, Figure 1. This property can be formalised by the following equations:

$$\frac{\|\underline{p} - \mathbf{c}\|}{\|\underline{p}' - \mathbf{c}\|} = 1 \quad (7)$$

$$\cos(\theta) = \frac{(\underline{p} - \mathbf{c})^T (\underline{p}' - \mathbf{c})}{\|\underline{p} - \mathbf{c}\| \|\underline{p}' - \mathbf{c}\|} \quad (8)$$

Given a homogeneous transformation matrix $(T = [R|\mathbf{t}])$, a conversion to screw axis representation ($\mathbf{h}_c = \mathbf{c} + u\mathbf{h}$) can be achieved by the following equations:

$$\mathbf{h} = \frac{\mathbf{1}}{2\theta \sin \theta}, \quad \mathbf{c} = \frac{(I - R)\mathbf{t}}{2(1 - \cos \theta)} \quad (9)$$

Where I is the identity 3×3 matrix, and $\mathbf{1}$, θ are given by the following equations:

$$\mathbf{1} = [(R_{32} - R_{23}) \quad (R_{13} - R_{31}) \quad (R_{21} - R_{12})] \quad (10)$$

$$\theta = \text{sign}(\mathbf{1} \times \mathbf{t}) \left| \cos^{-1} \left(\frac{R_{11} + R_{22} + R_{33} - 1}{2} \right) \right| \quad (11)$$

Proofs of the above equations can be found in [17].

B. Algorithm for 3D Motion Estimation

The algorithm is divided into two stages, the evidence gathering algorithm and the voting algorithm. In the evidence gathering algorithm, Chasles' theorem, summarised in Equation 6, is applied iteratively on the two point clouds, and the estimated parameters of the screw axis are stored along with their three corresponding point pairs. The voting algorithm is applied on the stored axes to find the best estimation of the true screw axis. The three displacements must be non-parallel ($\mathbf{d}_1 \nparallel \mathbf{d}_2 \nparallel \mathbf{d}_3$), and the absolute difference of distances between points on the same cloud should be below a certain threshold α . Since three points on each cloud are taken at each iteration, we set this difference to be between

```

for (i = 0; i < M - 1; i++) do
  for (k = i + 1; k < M - 1; k++) do
    for (o = k + 1; o < M - 1; o++) do
      for (j = 0; j < M' - 1; j++) do
        for (l = 0; l < M' - 1; l++) do
          for (w = 0; w < M' - 1; w++) do
            if (j ≠ l and l ≠ w and j ≠ w)
              then
                if ( (d1 ∥ d2 ∥ d3) and
                  (|dmin1 - dmin2| < α)) then
                  Compute hc
                  Save h, c in L
                  Save (i, j), (k, l), (o, w)
                end
              end
            end
          end
        end
      end
    end
  end
end

```

Algorithm 1: Evidence gathering for 3D motion.

the smallest distances between the three pairs on each cloud (d_{min1}, d_{min2}).

Due to a potentially large number of invalid correspondences, even with imposing geometrical constraints, it is necessary to use a voting algorithm to obtain accurate estimates for the rotation axis. The algorithm identifies the largest cluster of similar axes. Axes estimated from erroneous correspondences will be randomly distributed in space, whereas correct correspondences will be closely aligned. To calculate the clusters efficiently, a pair of thresholds are used.

The first threshold (t_{dir}), is the minimum value that the dot product between the two axes can take. Axes that are in approximately the same direction will have a dot product close to 1. As the axes become more divergent the dot product value will approach -1, this is usually referred to as cosine similarity. The axes might have similar orientations but opposite directions, hence the absolute value is taken. The second threshold (t_{pos}) is the maximum Euclidean distance between the axes. The optimal threshold values for the voting algorithm can be identified by calculating the most generous pair of thresholds that correctly identify the true axis in a training data set. Once Algorithm 2 is implemented, and the screw axis with the most consensus $\hat{\mathbf{h}}_c$ is determined, the rotation matrix R and the 3D translation vector \mathbf{t} are computed by the following equations:

$$R = I - H \sin \theta + (1 - \cos \theta) H^2 \quad (12)$$

$$\mathbf{t} = (I - R^T) \mathbf{c} + \mathbf{h} \mathbf{h}^T \dot{\mathbf{d}} \quad (13)$$

where I is the identity 3×3 matrix, θ is defined in Equation 8, and $\dot{\mathbf{d}}$ is the linear displacement ($\dot{\mathbf{p}} - \dot{\mathbf{p}}'$) computed from any pair of the three pairs of point correspondences indexed with $\hat{\mathbf{h}}_c$. The derivation of equations 13 and 12 can be found in [6].

```

for (i = 0; i < L; i++) do
  for (j = i + 1; j < L; j++) do
    if (|hi · hj| > tdir) then
      if (||ci - cj|| < tpos) then
        Accumulator[i]++
        Accumulator[j]++
      end
    end
  end
end
Search Accumulator for peak.

```

Algorithm 2: Voting for invariant axis.

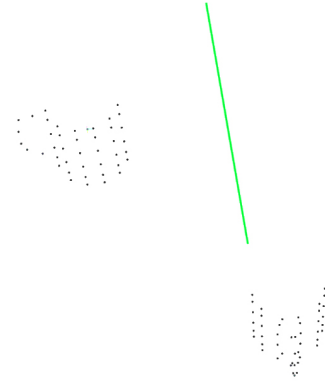


Fig. 2: 2.5D point cloud rotated and translated in 3D, ground truth screw axis representing motion in green.

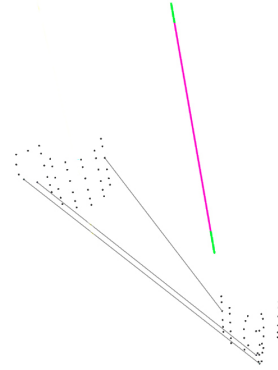


Fig. 3: Estimated screw axis with its associated three pairs of correspondences. The estimated axis (purple) perfectly coincides with ground truth screw axis (green).

IV. ANALYSIS

The algorithm is verified by implementing it on two 2.5D point clouds of the same object rotated and translated in 3D, note that the two point clouds have partial overlap. The corresponding ground truth screw axis \mathbf{h}_g , shown in 2, of this transformation is computed using Equation 9 is equal to:

$$\mathbf{h}_g = \begin{bmatrix} 175 \\ 0 \\ -25 \end{bmatrix} + u \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (14)$$

The ground truth screw axis has a rotation angle θ of 90°

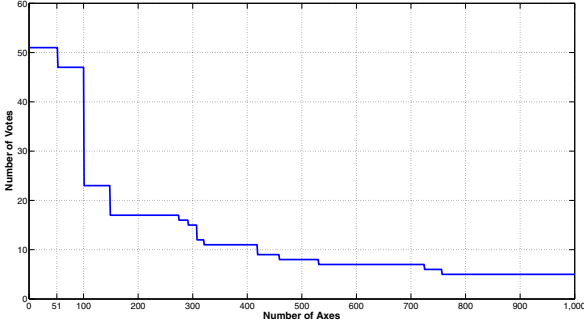


Fig. 4: Histogram of votes for axes generated by Algorithm 2.

and a pitch $\eta = 3.197$. Algorithm 2 is implemented to find the screw axis \mathbf{h}_c best describing the 3D transformation. Figure 3 shows \mathbf{h}_c and the three pairs of estimated corresponding points $((p_1, p_1'), (p_2, p_2'), (p_3, p_3'))$, it is clear in the figure that the estimated screw axis \mathbf{h}_c perfectly coincides with the ground truth screw axis. From \mathbf{h}_c and any of the associated three point pairs, the motion parameters are computed using equations 12 and 13.

Figure 4 shows the histogram of axes in terms of number of votes, note that the number of axes having the maximum number of votes is equal to the maximum number of votes, which confirms the methodology of the voting algorithm.

In the presented algorithm, the voting is not for the actual motion parameters, it is actually for the axis representing the motion. Hence its accumulator space is one-dimensional (counter), defined only by the number of points of the object; as opposed to other methods, such as those in Table I, which their accumulator spaces are defined by the resolution (bin size) and ranges of motion and rotation parameters, in addition to the number of points. Moreover, the application of the axes similarity thresholds (t_{dir} , t_{pos}) discards a large number of axes from the voting process; only axes passing the two thresholds cast votes in the 1D space. However, other methods will vote for motion parameters generated from *all* points (or features) in a high dimensional space. So not only the accumulator space dimensionality of the presented algorithm is smaller, but also the size of the accumulator itself, i.e. number of votes, is smaller. In other words, using axes to vote for motion parameters makes the accumulator array “dynamic” rather than a “large” predefined high-dimensional array. Hence the presented algorithm will have improved memory requirements.

V. RESULTS

Figure 5 shows the result of implementing the proposed algorithm on point clouds rotated and translated in 3D space. The performance of the algorithm can be seen from the figure as the screw axis is estimated with minimal errors. Once the screw axis, the rotation angle and the linear displacement are estimated, Equations 12 and 13 are applied to obtain the homogeneous transformation parameters. Details of the implementation are in Table II. Note that the point clouds are subsampled by applying a voxel grid and approximating the points within each voxel to one point.

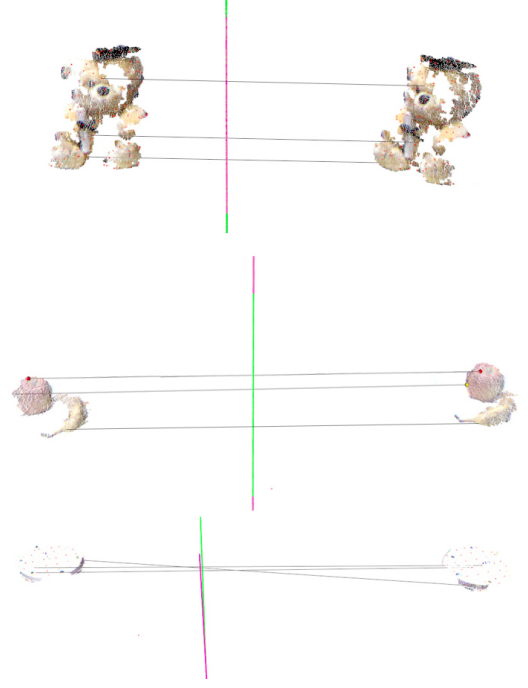


Fig. 5: Results of implementing general motion estimation algorithm on different point clouds. Ground truth screw axis in green, estimated in purple.

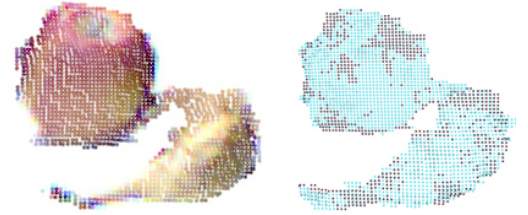


Fig. 6: Registration result using motion estimated by proposed algorithm.

VI. EVALUATION

Figures 7 to 9 show the positional, directional and angle offsets between the ground screw axis and the estimated screw axis at increasing voxel size (subsampling step). The performance of the algorithm degrades as the larger the voxel size used in subsampling, the larger the volume that will contain surface points that will be approximated to one point. As more points are approximated to one point, less points are included in the evidence gathering process. Also, the larger the voxel size is, the larger the Euclidean distance between the subsampled point and the original surface points. Hence the axis estimation becomes less accurate.

PC	dms (mm^3)	e_{pos}	e_{dir}	$e_{\theta} (^{\circ})$	D	Time(s)
Fruits	$175 \times 64 \times 217$	0.01	0.0	0.65	0.1	34.56
Teddy	$47 \times 22 \times 28$	0.04	0.0	0.01	5.3	14.87
Disc	$231 \times 29 \times 197$	2.68	0.03	2.53	1.9	57.76

TABLE II: Performance of motion estimation algorithm. Columns: point cloud name, dimensions, positional offset (mm), directional offset, angle offset, difference between the ground truth linear displacement along the screw axes and the measured displacement along estimated axis (mm) and processing time.

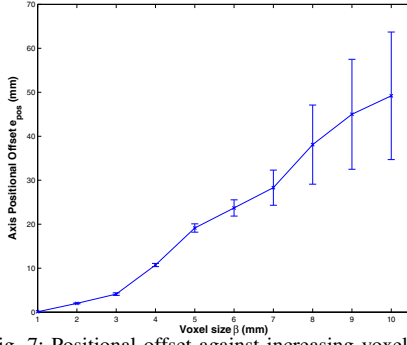


Fig. 7: Positional offset against increasing voxel size.

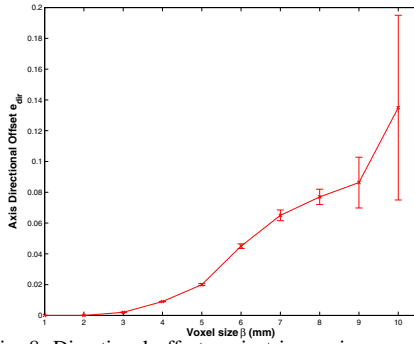


Fig. 8: Directional offset against increasing voxel size.

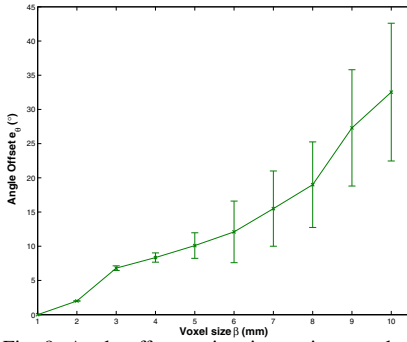


Fig. 9: Angle offset against increasing voxel size.

VII. CONCLUSION

In this paper, we presented a novel algorithm for 3D motion estimation in point clouds that is based on Chasles' kinematic theorem. The algorithm estimates 3D motion parameters directly from the data by exploiting the geometry of rigid transformation using an evidence gathering technique in a Hough-voting-like approach. It does not involve keypoint detection and feature descriptor computation and matching, and hence it can be used as an alternative to the feature description and matching pipelines commonly used by numerous 3D object recognition and registration algorithms. To the best of our knowledge, this is the first research to use the kinematics theorems in an evidence gathering framework for motion estimation and surface matching without the use of any given correspondences. We discussed similar related work, in terms of motion estimation using geometrical analysis and Hough voting, and showed that how voting for motion parameters can

be achieved in only one-dimensional space instead of a multi-dimensional one. Experimental analysis and evaluation have been provided, and results of implementing the algorithm on real point clouds have been shown and analysed.

REFERENCES

- [1] Y. Liu and M. A. Rodrigues, "Invariant geometric properties of image correspondence vectors as rigid constraints to motion estimation," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, no. 08, pp. 1165–1179, 1999.
- [2] W. B. Heard, *Rigid body mechanics: mathematics, physics and applications*. John Wiley & Sons, 2008.
- [3] Y. Liu and M. A. Rodrigues, "Geometric understanding of rigid body transformations," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1275–1280.
- [4] R. B. Fisher, "Geometric constraints from planar surface patch matching," *Image and Vision Computing*, vol. 8, no. 2, pp. 148–154, 1990.
- [5] J. Baroon and B. Ravani, "A three-dimensional generalization of reuleaux method based on line geometry," in *ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2006, pp. 1123–1130.
- [6] Y. Liu and M. A. Rodrigues, "Geometrical analysis of two sets of 3D correspondence data patterns for the registration of free-form shapes," *Journal of Intelligent and Robotic Systems*, vol. 33, no. 4, pp. 409–436, 2002.
- [7] L. Reyes, G. Medioni, and E. Bayro, "Registration of 3D points using geometric algebra and tensor voting," *International Journal of Computer Vision*, vol. 75, no. 3, pp. 351–369, 2007.
- [8] M.-T. Pham, O. J. Woodford, F. Perbet, A. Maki, B. Stenger, and R. Cipolla, "A new distance for scale-invariant 3d shape recognition and registration," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 145–152.
- [9] A. Ashbrook, R. Fisher, C. Robertson, and N. Werghi, "Finding surface correspondence for object recognition and registration using pairwise geometric histograms," in *Computer Vision ECCV98*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1998, vol. 1407, pp. 674–686.
- [10] F. J. Aherne, N. A. Thacker, and P. I. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 34, no. 4, pp. 363–368, 1998.
- [11] F. Tombari and L. D. Stefano, "Hough voting for 3D object recognition under occlusion and clutter," *IPSN Transactions on Computer Vision and Applications*, vol. 4, no. 0, pp. 20–29, 2012.
- [12] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, "Hough transform and 3D SURF for robust three dimensional classification," in *Computer Vision—ECCV 2010*. Springer, 2010, pp. 589–602.
- [13] K. Khoshelham, "Extending generalized Hough transform to detect 3D objects in laser range data," in *ISPRS Workshop on Laser Scanning, Proceedings, LS 2007*. Citeseer, 2007, pp. 206–210.
- [14] H.-T. Tsui and C.-K. Chan, "Hough technique for 3D object recognition," *Computers and Digital Techniques, IEE Proceedings E*, vol. 136, no. 6, pp. 565–568, 1989.
- [15] G. Hu, "3-D object matching in the Hough space," in *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, vol. 3. IEEE, 1995, pp. 2718–2723.
- [16] T. Wang, X. He, and N. Barnes, "Learning structured Hough voting for joint object detection and occlusion reasoning," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1790–1797.
- [17] B. Siciliano and O. Khatib, *Springer handbook of robotics*, 2008.