

# On Averaging Multiview Relations for 3D Scan Registration

Venu Madhav Govindu and Pooja A.

**Abstract**—In this paper, we present an extension of the iterative closest point (ICP) algorithm that simultaneously registers multiple 3D scans. While ICP fails to utilize the multiview constraints available, our method exploits the information redundancy in a set of 3D scans by using the averaging of relative motions. This averaging method utilizes the Lie group structure of motions, resulting in a 3D registration method that is both efficient and accurate. In addition, we present two variants of our approach, i.e., a method that solves for multiview 3D registration while obeying causality and a transitive correspondence variant that efficiently solves the correspondence problem across multiple scans. We present experimental results to characterize our method and explain its behavior as well as those of some other multiview registration methods in the literature. We establish the superior accuracy of our method in comparison to these multiview methods with registration results on a set of well-known real datasets of 3D scans.

**Index Terms**—3D scan registration, ICP, multiview geometry.

## I. INTRODUCTION

ADVANCES in techniques for 3D scanning and analysis have lead to the increased use of 3D representations of objects in diverse areas of application such as cataloguing and display of cultural artifacts, engineering modelling, virtual reality, prosthetic design etc. Since any scanning device can only view a part of an object from a given viewpoint, registration of partial scans is a necessary step in building a full 3D representation. The equivalent problem in the context of 2D images is variously known as image mosaicing, registration, stitching, building panoramas etc. and researchers have developed a variety of methods to solve the underlying problem [2], [31]. Due to the inherent differences in the properties of images and 3D scans, progress on the registration of 3D scans has been slower, but nevertheless there are some stable solutions. The most significant of such methods is the iterative closest point (ICP) algorithm that was introduced in [3], [4] and has acquired canonical status and is available in the literature in a variety of forms. ICP is a greedy, iterative method and provides good results when started from an initialisation that is close to the final solution. However,

since the ICP algorithm is applied to two scans at a time, there is an inherent limitation to its accuracy. This limitation becomes pronounced when multiple scans are to be registered into a single object representation as ICP fails to account for the multiple constraints available. For instance, due to its sequential nature, ICP registration of scans acquired using a turntable fails to use the constraint between the last and the first scan. While there are many algorithms for 3D registration in the literature, in this paper, we develop a truly multiview extension of the ICP approach that utilises all available multiview constraints in an efficient manner while achieving high accuracy by distributing the errors evenly across the scans. Our approach is both more accurate and faster when compared with similar methods in the literature.

In Section II we survey the relevant literature on two-view and multiview scan registration. The motion averaging scheme used in our paper is presented in Section III and compared with other multiview averaging schemes in Section IV. Our multiview extension of ICP is given in Section V and two faster variants are described in Section VI. In Section VII we present results on real datasets that characterise our method and also compare its performance with two related multiview methods in the literature.

## II. RELATED WORK ON 3D SCAN REGISTRATION

In this section we only survey the existing techniques that are relevant to our approach of multiview registration based on ICP. We first briefly describe the ICP method before surveying the literature on multiview registration of 3D scans. Throughout we shall use the terms point sets and scans interchangeably.

### A. ICP Algorithm and its Variants

It is a well-known result that if correspondences between two point sets are known, we can easily solve for the rotation and translation (i.e. rigid 3D motion) required to register them in a common coordinate system [5], [6]. The closed-form SVD-based solution of [5] is efficient and optimal. However, in many scenarios such correspondence information is unavailable and needs to be estimated from the scan points themselves. Conversely, if we know the motion parameters that relate two point sets, the pointwise correspondence between the two sets can be easily established. This chicken-and-egg problem is effectively addressed by the well-known ICP algorithm that solves for the correspondences between two point sets and also the 3D motion model to register them.

Manuscript received October 20, 2011; revised August 5, 2012; accepted December 13, 2012. Date of publication February 11, 2013; date of current version February 6, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wan-Chi Siu.

V. M. Govindu is with the Department of Electrical Engineering, Indian Institute of Science, Bengaluru 560012, India (e-mail: venu@ee.iisc.ernet.in).

Pooja A. is with Amazon India, Bengaluru 560005, India (e-mail: apooja.june@gmail.com).

Digital Object Identifier 10.1109/TIP.2013.2246517

The ICP algorithm that was introduced in [3], [4] consists of interleaving the two steps of solving for the point correspondences and 3D motion. Given an initial motion estimate, we can compute a correspondence between two points sets (correspondence step). Using the correspondence mapping thus established, we can update the 3D motion (motion step). In ICP, these two steps form a single iteration and it can be shown that this algorithm iteratively converges to a local minima. Although it is a greedy approach, given a good initial guess of the 3D motion estimate, ICP works reasonably well. The 3D scan representation can either be in the form of a point cloud, i.e. lacking any structure, or as a mesh. For point clouds, the most intuitive way of assigning a correspondence for a point in the first scan is to pick its closest neighbour in the second scan [3]. Throughout this paper we use this method in the point correspondence step. In the case when we are given meshes or surfaces, [4] proposed to compute the correspondences based on the distance of a point to tangent planes of the second surface. A significant issue in the implementation of ICP is that often the two scans to be registered do not perfectly overlap, i.e. only a subset of scan points have correspondences. This fact implies that a variety of considerations like robustness [7], filtering of point correspondences [8] and visibility constraints [9] need to be accounted for to make ICP work in practice. Many such issues are succinctly examined in [10].

### B. Multiview Registration Methods

As we have seen above, the ICP approach registers two scans at a time. In contrast, the problem of building a full 3D model often entails the registration of a set of 3D scans that cover different regions of an object or scene. As a result, the registration problem is one of finding a set of transformations that will align multiple scans in a single reference frame. The straightforward approach to solving this problem is to align two scans and merge them. Next a new scan is registered to the merged scan and this process of registering and merging is repeated till all the scans are used. This approach is used in [4] and suffers from the problem that errors in the individual steps accumulate resulting in poor estimates as the model grows. The remedy is to simultaneously consider all the scans and solve the multiview registration problem. This is the problem addressed in this paper and here we briefly survey the existing multiview registration methods of relevance.

In some methods, the point correspondences across multiple views are assumed to be known and the problem is reduced to that of solving for the 3D motion models of all the scans involved. [11] provides a comparison of some of these techniques. In [12], the objective of registering known correspondences across views is achieved by minimising a registration error-measure using Newton's method on the product manifold of 3D rotations. In [13], the authors assume that the individual scan pairs are registered using ICP. Given these registration results, an overall error measure is relaxed across all the views with an aim of distributing the errors as well as possible. While [12], [13] solve for the registration of multiple scans, they do not directly address the issue of

solving for correspondences between scan pairs during the process of registration. Our multiview 3D scan registration method simultaneously addresses the problem of determining correspondences between all relevant scan pairs and also solves for the 3D motions of all scans involved.

In [14] a multiview registration method is proposed whereby a set of pairwise motions are first estimated and then refined through incremental registration of scans. The scans are added one at a time to the registered scans by using the relevant correspondences and estimated pairwise motions which in turn are adjusted to distribute the errors across the motion estimates. In [15], multiview registration is carried out via a set of key points. Scan points are registered to key points and in turn key points are updated using the registered points. [16] optimises a robust multiview cost function via a conjugate gradient search approach.

Amongst the initial attempts at multiview registration are those of [9], [29]. In their method, the scan that overlaps the maximum number of scans is selected as the reference. The motions corresponding to other scans are updated one at a time by considering all the matches that are shared between the given scan and the other scans. This is achieved by transforming all matching scans into the coordinate system of the scan for which we need to compute the motion. The process of simultaneous multiview registration is carried out by repeatedly cycling through all scans and updating their motion followed by updating the correspondences. This method is modified in [17] where the updating of a motion model is done *in situ* instead of a single update at the end as is done in [9]. Any potential bias in the solution is avoided by randomizing the sequence of updates of individual scan motion models in a given iteration. As a result of carrying out the individual updates in place, the convergence of the algorithm is accelerated.

Another method of significance in our context is that of [18] which uses the graph representation of the pairwise motions as the basis for multiview registration. This method assumes that each edge in the graph is present in at least one graph cycle. The degree of inconsistency over a cycle as seen in the context of each edge is estimated and the edge is corrected for this discrepancy. Using a heuristic, this method iterates over all such edge discrepancies till the algorithm converges to a final solution. While this results in a consistent solution, the errors are not distributed optimally resulting in a solution of poor quality. In this paper, we will compare the performance of Benjemaa *et al.* [17] and Sharp *et al.* [18] with that of our proposed method. Finally, we remark that our survey of the 3D registration methods has been necessarily brief and we have not considered a variety of papers that adopt methods quite different from our approach. The interested reader can refer to the surveys presented in [10], [19]. In addition, the discussion on related work in [13] also provides a good contemporary summary of the various registration techniques in the literature.

Although, the work on multiview 3D registration is significant, we note that most methods have certain shortcomings or limitations. Global optimisation techniques tend to be slow and cumbersome. While the motion estimation methods using

fixed correspondences such as [12] are relatively fast due to the reduced number of variables to consider, they are evidently suboptimal as initially dense correspondences between 3D scans are only approximately correct. When methods use correspondences obtained from intermediate ICP results e.g., [13], they inherit the biases and errors of the individual ICP results. In particular, we remark here that a pairwise ICP solution is based on limited information and will be less accurate than its multiview counterpart where more constraints are available. In contrast, the methods of [17], [18] are truly multiview, as they iterate by updating both motion and correspondences. However, as we shall explain and demonstrate later in this paper, during the iterations for registering 3D scans, both these methods adopt suboptimal motion updates resulting in inferior performance. Given these considerations, we now present our approach to a truly multiview extension of the ICP algorithm that estimates both the motions and correspondences between a set of 3D scans. To motivate our approach, we first describe the idea of motion averaging of [20] that takes a set of relative motions and efficiently and accurately averages them to solve for a global motion model. In our approach we utilise this idea of motion averaging to simultaneously solve the motion step for all scans.

### III. MOTION AVERAGING

In this section, we develop the idea of motion averaging and also describe the equivalent motion update approaches of [17], [18]. We reiterate that the motion averaging step works only on the 3D motion models and does not use correspondences, i.e. in this section we will only consider the motion step independent of the correspondence step used in the overall registration method. We denote the 3D rotation or Euclidean motion of a scan with respect to a given frame of reference as  $\mathbf{M}$ . Without loss of generality, we can attach the frame of reference to the first scan, i.e.  $\mathbf{M}_1 = \mathbf{I}$  where  $\mathbf{I}$  is the corresponding identity matrix. The motions required for full registration of all scans are denoted as  $\mathbf{M}_{\text{global}} = \{\mathbf{I}, \mathbf{M}_2, \dots, \mathbf{M}_N\}$  where  $N$  is the number of scans to be registered, i.e.  $\mathbf{M}_{\text{global}}$  is the 3D registration solution required. Also  $\mathbf{M}_{ij}$  represents the relative motion that will align scans  $i$  and  $j$ . Given these representations, we can describe the relative motion  $\mathbf{M}_{ij}$  in terms of the global motion model  $\mathbf{M}_{\text{global}}$  as

$$\mathbf{M}_j \mathbf{M}_i^{-1} = \mathbf{M}_{ij} \quad (1)$$

$$\Rightarrow \underbrace{\left[ \dots \mathbf{M}_{ij} \dots -\mathbf{I} \dots \right]}_{\text{Relative Motions : } \mathbf{A}_{ij}} \underbrace{\begin{bmatrix} \vdots \\ \mathbf{M}_i \\ \vdots \\ \mathbf{M}_j \\ \vdots \end{bmatrix}}_{\mathbf{M}_{\text{global}}} = \mathbf{0}.$$

Consider a graph  $G = (V, E)$  as shown in Fig. 1(a) that represents the relationship between a set of motion models. While the averaging idea presented here applies in a general sense, to give it the context of multiview 3D registration, each

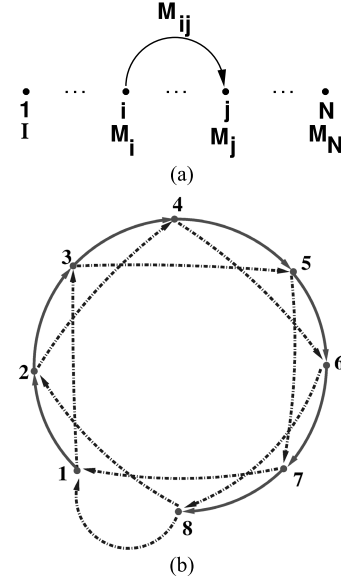


Fig. 1. (a) Viewgraph representing a set of scans to be registered.  $\mathbf{M}_{ij}$  represents the relative motion estimated from the scans  $i$  and  $j$ .  $\mathbf{M}_{\text{global}} = \{\mathbf{I}, \dots, \mathbf{M}_i, \dots, \mathbf{M}_j, \dots, \mathbf{M}_N\}$  represents the global camera motion to be estimated. For simplicity of viewing, only one edge of the viewgraph is shown here. (b) Viewgraph of a turntable sequence. The solid edges are relative motions used in sequential registration. Dashed edges are the additional relationships available for motion averaging.

vertex in  $V$  indexes an individual scan and represents the 3D motion required to globally align this scan with all other scans. The presence of an edge in  $E$  between vertices  $i$  and  $j$  implies that scans  $i$  and  $j$  overlap and further the motion model represented by this edge is the relative motion that will align the scans  $i$  and  $j$  in a common frame of reference. **Each edge in  $E$  represents a single relative motion between a pair of scans and provides a single constraint of the form in (1) on the global motion estimate  $\mathbf{M}_{\text{global}}$ .** If we stack all such constraints we obtain a system of equations  $\mathbf{A} \mathbf{M}_{\text{global}} = \mathbf{0}$  where  $\mathbf{A}$  is constructed by stacking  $\mathbf{A}_{ij}$ 's of (1) as rows. It is evident from the viewgraph representation  $G$ , that if  $E$  contains a spanning tree, we can construct a global motion estimate since each vertex (i.e. scan) is reachable from the frame of reference (i.e. the first scan). Further, if we have more than the minimal number of relationships represented by a spanning tree, by solving the overdetermined system of equations  $\mathbf{A} \mathbf{M}_{\text{global}} \simeq \mathbf{0}$ , we can effectively average the relative motions to get  $\mathbf{M}_{\text{global}}$ . In particular, we note that specifying  $\mathbf{M}_{\text{global}}$  requires  $N - 1$  motion models, whereas in a sequence we can have as many as  $(N(N - 1))/2$  edges (i.e. equations) available. In other words, given the redundancy of information in the individual relative motions, we can average them to obtain an accurate and globally consistent motion estimate. It may be noted here that any solution  $\mathbf{M}_{\text{global}}$  is by definition consistent and we seek  $\mathbf{M}_{\text{global}}$  that best explains the observations  $\mathbf{M}_{ij}$  according to (1).

#### A. Lie-Algebraic Averaging

In this subsection, we summarise the method used to average the relative motions as described by (1). The motion

representations used for registering 3D scans are either 3D rotations ( $\mathbf{R}$ ) or Euclidean motions ( $\mathbf{M}$ ) which are elements of the Special Orthogonal  $SO(3)$  or Special Euclidean groups  $SE(3)$  respectively. Specifically, Euclidean motions have the form

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \quad (2)$$

where  $\mathbf{R} \in SO(3)$  and  $\mathbf{t} \in \mathbb{R}^3$  are the 3D rotation and translation components of the Euclidean motion respectively. Here, we make the crucial observation that we cannot solve for  $\mathbf{M}_{\text{global}}$  by simply solving the linear system of equations  $\mathbf{A}\mathbf{M}_{\text{global}} \simeq \mathbf{0}$ . In the linear solution, the geometric constraints of each motion belonging to a Lie group are not enforced. Since the desired global motion should obey the underlying geometric constraints, our estimated  $\mathbf{M}_{\text{global}}$  should be confined to the product manifold of the corresponding Lie groups. While averaging on manifolds in general can be a difficult problem, we can derive an efficient solution in our case by exploiting the smooth differentiable structure of Lie groups. **The algorithm for averaging relative motions that we use in this paper was proposed in [20] and its simplicity and accuracy relies on the properties of Lie groups.** In the following we very briefly describe the properties of Lie groups of relevance to us and then describe the Lie algebraic averaging algorithms we will use. For further details, the reader may refer to [20]–[22].

In our context we consider matrix groups which are also finite-dimensional Lie groups. A Lie group is a group that also has a smooth differentiable structure. Specifically, a Lie group  $\mathcal{M}$  can be locally described by a vector space representation, i.e. the local neighbourhood of any group element  $\mathbf{M} \in \mathcal{M}$  can be fully described by the tangent space. This tangent space is represented by the corresponding Lie algebra  $\mathfrak{m}$ , which is a vector space equipped with a bilinear operation  $[\cdot, \cdot] : \mathfrak{m} \times \mathfrak{m} \rightarrow \mathfrak{m}$  known as the Lie bracket. Most important for our purposes, the Lie algebra and its corresponding Lie group are related by the exponential mapping, i.e.  $\mathbf{M} = \exp(\mathbf{m})$  and conversely by the logarithmic mapping, i.e.  $\mathbf{m} = \log(\mathbf{M})$ . In informal terms, this special property of Lie groups implies that the mapping between the Lie group and its tangent space is exact and does not involve any first order approximations. In the case of the specific Lie groups of interest to us, i.e.  $SO(3)$  and  $SE(3)$ , both the exponential and logarithm mappings have closed form solutions which leads to efficient averaging on these groups.

To develop the algorithm for averaging relative motions of the form  $\mathbf{M}_{ij}$  to estimate global motion  $\mathbf{M}_{\text{global}}$ , it will be useful to first consider an algorithm for averaging a set of absolute motions, i.e. of the form  $\mathbf{M}_i$ . For instance, given a set of rotation matrices  $\{\mathbf{R}_1, \dots, \mathbf{R}_N\}$ , their elementwise arithmetic mean  $1/N \sum_i \mathbf{R}_i$  is not necessarily a valid rotation as the orthonormality constraints of a rotation matrix (i.e.  $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ ) is not enforced. However, the desired rotation average can be obtained by solving the variational minimiser of the **Riemannian distance** on the  $SO(3)$  group while obeying the orthonormality constraints on the estimated rotation matrix. In other words, we seek  $\bar{\mathbf{R}} \in SO(3)$  that minimises  $\sum_{i=1}^N d^2(\mathbf{R}_i, \bar{\mathbf{R}})$ , where  $d(\cdot, \cdot)$  is the Riemannian

distance on the  $SO(3)$  group. For such a minimisation, an efficient solution is obtained by successively approximating the Riemannian distance on the tangent space of the Lie group, i.e. its corresponding Lie algebra.

Throughout we will illustrate our averaging procedure in a general setting that holds for all matrix Lie groups. Given  $N$  elements of a matrix group  $\mathcal{M}$ , i.e. a set of matrices  $\{\mathbf{M}_1, \dots, \mathbf{M}_N\}$  where  $\mathbf{M}_i \in \mathcal{M}$ , we seek their average  $\bar{\mathbf{M}} \in \mathcal{M}$  that minimises the variational measure of  $\sum_{i=1}^N d^2(\mathbf{M}_i, \bar{\mathbf{M}})$ , where  $d(\cdot, \cdot)$  is the Riemannian metric corresponding to the matrix group  $\mathcal{M}$ . The procedure for obtaining this average of absolute motions of the form  $\mathbf{M}_i$  is given as:

---

**Algorithm 1:** Algorithm for Intrinsic Average

---

Input :  $\{\mathbf{M}_1, \dots, \mathbf{M}_N\} \in \mathcal{M}$  (Matrix Group)

Output :  $\bar{\mathbf{M}} \in \mathcal{M}$  (Intrinsic Average)

Set  $\bar{\mathbf{M}}$  to an initial guess

Do

$$\Delta \mathbf{M}_i = \bar{\mathbf{M}}^{-1} \mathbf{M}_i$$

$$\Delta \mathbf{m}_i = \log(\Delta \mathbf{M}_i)$$

$$\Delta \bar{\mathbf{m}} = \frac{1}{N} \sum_{i=1}^N \Delta \mathbf{m}_i$$

$$\Delta \bar{\mathbf{M}} = \exp(\Delta \bar{\mathbf{m}})$$

$$\bar{\mathbf{M}} = \bar{\mathbf{M}} \Delta \bar{\mathbf{M}}$$

Repeat till  $\|\Delta \bar{\mathbf{m}}\| < \epsilon$

---

where  $\log(\cdot)$  and  $\exp(\cdot)$  are matrix operations and  $\mathfrak{m}$  is the corresponding Lie algebra of  $\mathcal{M}$ . This algorithm has been proved to be globally convergent if the individual matrices  $\mathbf{M}_i$  lie within a certain distance from each other, which in the case of  $SO(3)$  requires all  $\mathbf{R}_i$  matrices to lie within an open ball of radius  $\pi/2$ , see [30]. In the case of  $SO(3)$  and  $SE(3)$ , the  $\log(\cdot)$  and  $\exp(\cdot)$  operations (which relate the Lie group and its corresponding Lie algebra) have a **closed form solution** thus providing a computationally efficient procedure. For  $\mathbf{R} \in SO(3)$ , the corresponding Lie-algebra is the **skew-symmetric matrix**  $\boldsymbol{\Omega} \in \mathfrak{so}(3)$  given by the familiar Rodrigues formula for rotations [32]. In the case of  $\mathbf{M} \in SE(3)$ , the corresponding Lie algebra  $\mathbf{m} \in \mathfrak{se}(3)$  is given by

$$\mathbf{m} = \begin{bmatrix} \boldsymbol{\Omega} & \mathbf{u} \\ 0 & 0 \end{bmatrix}. \quad (3)$$

Here, given the  $\{\mathbf{R}, \mathbf{t}\}$  components of  $\mathbf{M}$ , we can derive the corresponding Lie-algebra forms  $\{\boldsymbol{\Omega}, \mathbf{u}\}$  using the following relationships

$$\begin{aligned} \mathbf{R} &= \exp(\boldsymbol{\Omega}) \\ \mathbf{t} &= \mathbf{P}\mathbf{u} \\ \text{for } \mathbf{P} &= \mathbf{I} + \frac{(1 - \cos \theta)}{\theta^2} \boldsymbol{\Omega} + \frac{(\theta - \sin \theta)}{\theta^3} \boldsymbol{\Omega}^2 \end{aligned} \quad (4)$$

where  $\theta = \sqrt{1/2 \text{tr}(\boldsymbol{\Omega}^T \boldsymbol{\Omega})}$ . In a manner analogous to such averaging via successive approximation using the Lie algebra, we can also solve for the global motions that best fit a given set of relative motions. For the relationship  $\mathbf{M}_{ij} = \mathbf{M}_j \mathbf{M}_i^{-1}$  we can write the equivalent relationship for the corresponding Lie algebras as  $\log(\mathbf{M}_{ij}) = \log(\mathbf{M}_j \mathbf{M}_i^{-1}) = BCH(\mathbf{m}_j, -\mathbf{m}_i)$

where  $BCH(.,.)$  is the well-known Baker-Campbell-Hausdorff form [22] that defines the Lie algebra for the product of two elements of a Lie group. Further, we utilise the first-order approximation  $BCH(\mathbf{M}_j \mathbf{M}_i^{-1}) = \log(\mathbf{M}_j \mathbf{M}_i^{-1}) \approx \log(\mathbf{M}_j) - \log(\mathbf{M}_i) = \mathbf{m}_j - \mathbf{m}_i$ , to derive

$$\log(\mathbf{M}_{ij}) = \log(\mathbf{M}_j \mathbf{M}_i^{-1}) = BCH(\mathbf{m}_j, -\mathbf{m}_i) \quad (5)$$

$$\Rightarrow \mathbf{m}_{ij} = BCH(\mathbf{m}_j, -\mathbf{m}_i) \approx \mathbf{m}_j - \mathbf{m}_i \quad (6)$$

$$\Rightarrow \mathbf{v}_{ij} = \mathbf{v}_j - \mathbf{v}_i \Rightarrow \mathbf{v}_{ij} = \underbrace{[\dots - \mathbf{I} \dots \mathbf{I} \dots]}_{\mathbf{D}_{ij}} \mathfrak{V}$$

where  $\mathbf{v}$  is a vector representing the independent variables that define the Lie algebra  $\mathfrak{m}$ .  $\mathfrak{V}$  contains all the variables  $\mathbf{v}_i$  stacked together into a single vector. By collecting all the relative motion constraints columnwise, we have  $\mathbb{V}_{ij} = [\mathbf{v}_{ij1}; \mathbf{v}_{ij2}; \dots]$  and  $\mathbf{D} = [\mathbf{D}_{ij1}; \mathbf{D}_{ij2}; \dots]$  which leads to the solution to the motion averaging in the tangent space as  $\mathfrak{V} = \mathbf{D}^\dagger \mathbb{V}_{ij}$ , where  $\mathbf{D}^\dagger$  is the pseudo-inverse of  $\mathbf{D}$ . This leads to the following Lie-algebraic averaging algorithm that solves for the global motion  $\mathbf{M}_{\text{global}}$  given enough relative motion estimates  $\mathbf{M}_{ij}$ :

---

**Algorithm 2:** Algorithm for Motion Averaging

---

Input :  $\{\mathbf{M}_{ij1}, \mathbf{M}_{ij2}, \dots, \mathbf{M}_{ijn}\} \in \mathcal{M}$  ( $n$  relative motions)

Output :  $\mathbf{M}_{\text{global}} : \{\mathbf{I}, \mathbf{M}_2, \dots, \mathbf{M}_N\} \in \mathcal{M}$  (global motion)

Set  $\mathbf{M}_{\text{global}}$  to an initial guess

Do

$$\Delta \mathbf{M}_{ij} = \mathbf{M}_j^{-1} \mathbf{M}_i \mathbf{M}_{ij}$$

$$\Delta \mathbf{m}_{ij} = \log(\Delta \mathbf{M}_{ij})$$

$$\Delta \mathbf{v}_{ij} = \text{vec}(\Delta \mathbf{m}_{ij})$$

$$\Delta \mathfrak{V} = \mathbf{D}^\dagger \Delta \mathbb{V}_{ij}$$

$$\forall k \in [2, N], \mathbf{M}_k = \mathbf{M}_k \exp(\Delta \mathbf{m}_k)$$

Repeat till  $\|\Delta \mathfrak{V}\| < \epsilon$

---

where  $\text{vec}(\cdot)$  extracts the parameters  $\mathbf{v}$  from  $\mathfrak{m}$ . In Algorithms A1 and A2, the procedure is terminated when the change of the motion parameters between consecutive iterations falls below a preset constant  $\epsilon$ . In our experiments we set  $\epsilon = 10^{-2}$ . It may be noted that the matrix  $\mathbf{D}^\dagger$  is fully specified by the viewgraph and can be precomputed for use in Algorithm 2. The reader should consult [20] for further details.

It may be recognised that Algorithm 2 is our solution for the system of equations specified in (1). In a manner analogous to Algorithm 1, the cost function minimised by Algorithm 2, is given by  $\sum_{(i_s, j_s) \in E} d^2(\mathbf{M}_{i_s j_s}, \mathbf{M}_{j_s} \mathbf{M}_{i_s}^{-1})$ . Here the summation is over all edges in  $E$ , i.e. the cost function accounts for the difference between the observed relative motions and the relative motions obtained by the fitting carried out by Algorithm 2. It will also be recognised that in the presence of outliers, i.e. when one or more of the relative motions  $\mathbf{M}_{ij}$  are grossly incorrect, Algorithm 2 can be suitably modified to provide a robust solution. The interested reader may refer to the robust motion averaging solutions provided in [25], [26].

As discussed earlier, we require that edges  $E$  contain a spanning tree to get a global motion estimate. On the other hand, a fully connected viewgraph will have as many as  $(N(N-1))/2$  relative motions available. Typically, since individual scans overlap only with adjacent ones in a sequence, we do not have all the possible relative motion estimates available. However, since there is a significant amount of redundancy of relative motions in any typical scan dataset, i.e. there are more than  $N-1$  relative motions available, we can average the motion information to our advantage. This idea can be illustrated by a pedagogical example. Consider an object on a turntable that moves by  $45^\circ$  between consecutive scans. The resulting viewgraph representing the scan relationships is shown in Fig. 1(b). The straightforward approach using ICP would register adjacent scans and use the estimated motions to solve for the global registration problem. The relative motions between adjacent scans used in this approach are shown in solid edges. In the case of ICP, if we consider scan 8, its registration with scan 1 will be defined by the composition of all the relative motions along the path from scan 1 to scan 8, i.e.  $(1 \rightarrow 2 \rightarrow \dots 7 \rightarrow 8)$ . In such a scenario, the individual errors along the path will be added up and the final motion estimate of scan 8 with respect to scan 1 can be significantly erroneous.

However, in most typical scenarios we can expect a substantial overlap between scans that are two steps away from each other, say between scans 1 and 3, 2 and 4 etc. Consequently, in addition to the motions between adjacent scans, we can also solve for the relative motions between scan pairs that are two steps apart (shown in dashed edges). Thus, in the example of Fig. 1(b) while the ICP-based global motion estimate uses a cascade of 7 relative motions, for our approach of motion averaging we have an additional 8 constraints. As a result, to solve for a global motion model, we can average all of these 15 relative motions, i.e. both solid and dashed ones. It may also be noted that the additional relative motions provided by the dashed edges act as constraints on the global solution and prevent the accumulation of error. In particular, a crucial factor that improves the averaged solution significantly is the availability of an additional constraint given as a dashed edge directly between scan 8 and scan 1. The relative motion obtained from this edge will act as an ‘‘anchor’’ and forces the solution to distribute the motion errors over the entire global motion. This prevents the drift of the solution as would happen in the case of the straightforward ICP approach. This idea of motion averaging can be applied to any situation where we can compute some additional relative motions beyond the minimal number defined by a spanning tree. Many instances of the application of this motion averaging idea in other domains are available. Some illustrative examples may be found in the context of image registration [23], [24] and structure-from-motion from image sequences [25], [26]. While in the pedagogical example of Fig. 1(b) we have used an ordered sequence of scans, the notion of motion averaging is independent of any such ordering. As long as we can establish the relative motions, the averaged global motion can be estimated.

#### IV. COMPARISON WITH OTHER MOTION AVERAGING METHODS

In Section V we present our solution that incorporates motion averaging into the multiview 3D registration routine. We reiterate that as in the case of ICP between two scans, the multiview 3D scan registration method presented in this paper also consists of a correspondence step and a motion step, i.e. ours is a multiview extension of the ICP algorithm. However, before describing our complete algorithm, it will be instructive to compare our motion averaging method with other solutions. Specifically, two other multiview registration methods that are similar in spirit to our approach are the work of Benjemaa *et al.* [17] and that of Sharp *et al.* [18]. In [17], the motion averaging step is computed directly on the correspondences. During each iteration, an individual scan is randomly selected to have its motion updated. All other scans that share corresponding points are first transformed into the frame of reference of the selected scan. Using all the transformed correspondences available, the motion model of the selected scan is updated. In this manner the motion estimates of all scans are updated in random order and subsequently the correspondences are also updated. In [18], the authors consider individual cycles in a viewgraph. If the individual relative motions  $\mathbf{M}_{ij}$  are exact, the net motion obtained by traversing a single cycle on the viewgraph should be zero, i.e. the net motion matrix should be an identity matrix. This is so because exact relative motions are consistent with each other and traversal of a cycle brings us back to the same vertex, i.e. there is no net motion. However, in the presence of noise, individual  $\mathbf{M}_{ij}$  estimates will not be consistent with each other and a complete traversal will result in a residual motion. The authors of [18] develop a heuristic strategy to distribute this residual error across the individual edges of a cycle and update each relative motion accordingly. A global motion estimate is obtained upon convergence.

In this section, we wish to develop our understanding of the performance of the motion averaging steps inherent in [17] and [18] and compare them with our Lie-algebraic averaging method of Algorithm 2. We emphasise that in [17], their algorithm is not presented in terms of the two steps of correspondence and motion estimation. However, we can derive a motion step that is in the spirit of the motion update of [17]. The same is available in the case of [18]. Such a comparison of our Lie-algebraic averaging method and the motion averaging methods in the spirit of [17], [18] serves two purposes. Firstly, we understand the behaviour of the motion averaging steps independent of the influence of the correspondence step, i.e. solely the quality and behaviour of the motion averaging approaches of each method. Secondly, in a general 3D registration method, the errors in both the motion and correspondence steps are mixed up. Establishing a baseline comparison of the motion averaging methods allows us to illuminate and interpret the performance of the overall 3D registration algorithms. As will be evident later, using the baseline performance comparison in this section, we can better explain the relative performance of our algorithm and that of [17], [18].

We now develop the motion averaging method inherent in the method of [17]. Consider the relative motion constraint of (1). During the motion step in a given iteration, we can randomly select a scan index  $j$  and hold all other scan motions to be fixed. In other words, except for  $\mathbf{M}_j$ , all other entries of  $\mathbf{M}_{\text{global}}$  are held fixed to their current estimates. As a result, there is only one unknown quantity  $\mathbf{M}_j$  and its estimate is updated to the average of individual estimates implied by (1), i.e. the average of  $\{\mathbf{M}_{ij}\mathbf{M}_i | \forall i : (i, j) \in E\}$  where  $\mathbf{M}_i$  is fixed to the current estimate. This solution is an average of the absolute motion estimates and can be carried out using the Algorithm 1 above. In other words, *our* interpretation of the motion step of [17] consists of individually updating each motion model of  $\mathbf{M}_{\text{global}}$ , i.e.  $\{\mathbf{M}_2, \mathbf{M}_3, \dots, \mathbf{M}_N\}$  in a randomised order. We denote this motion averaging approach as “Individual Update” method, thereby emphasising that such a motion update is our explanation and should not be confused as being given in [17]. In contrast to the “Individual Update” method, our Lie-algebraic averaging method carries out the update as a single combined estimation step.

Although in the case of [17] and [18], rotations are represented using quaternions, to establish uniformity in the comparison presented in the remainder of this section, we use the Lie group representation of  $SO(3)$  which can be readily seen to be optimal since the Riemannian metric on  $SO(3)$  exactly measures the magnitude of angular rotation between any two rotations represented on the Lie group. To compare the relative performance of Lie-algebraic averaging with the “Individual Update” derived from [17] and that of [18] we conduct the following experiment. Throughout this experiment, our focus is on the performance of the different rotation averaging methods and hence we only consider noisy relative rotation estimates, i.e. there are no 3D scans involved in our setup. We consider problems of different sizes, i.e. the viewgraph  $G$  has the number of vertices (equivalent to scans) varying as  $|V| \in \{10, 25, 50, 75, 100\}$ . To generate the ground truth absolute rotations in  $\mathbf{R}_{\text{global}}$  we randomly select an axis of rotation and rotate by an angle that is Gaussian distributed with a standard deviation of  $100^\circ$ . As a result, the individual rotations can vary a lot. Throughout we consider a complete graph, i.e. all vertices are connected to each other. Using the ground truth rotations, we generate all relative rotations  $\mathbf{R}_{ij}$  which in turn are perturbed by a random rotation of varying strength. Here the perturbation rotation has a random axis and its angle is drawn from a Gaussian distribution with varying  $\sigma \in \{5^\circ, 10^\circ, 15^\circ, 20^\circ, 25^\circ\}$ . Thus since we have 5 graph sizes and 5 noise levels, we have in all 25 combinations of settings. For each instance, we initialised the global motion using the estimate provided by a spanning tree and solve for the averaged global motion estimate using the three methods being compared. For each of the settings we carried out 100 trials of rotation estimation and averaged the results. We define the error for a given trial as the root mean squared error of rotation estimation. The average error over the trials is simply the arithmetic mean of the root mean squared error of each trial. Further for a given noise level  $\sigma$  we averaged the errors over all possible sizes of the graph  $G$ . In Fig. 2(a) we present these averaged errors as a function of the noise level whereas

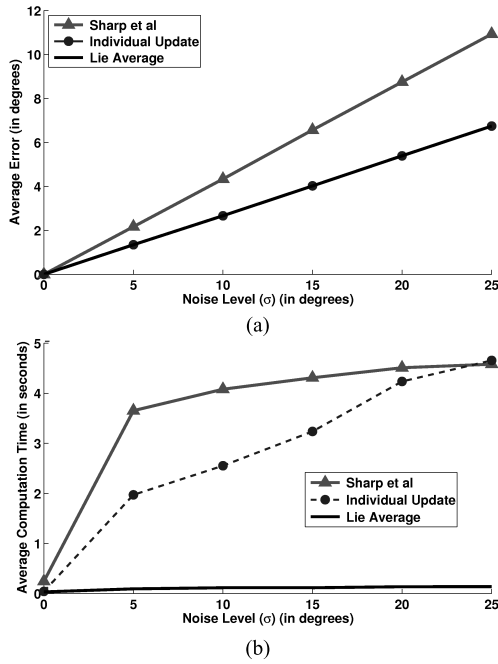


Fig. 2. Comparison of our rotation averaging with other methods. (a) The average fitting error for different noise levels averaged over different graph sizes. (b) The corresponding average computational time required for each method.

in Fig. 2(b) we plot the corresponding average computational time required for convergence.

As can be observed from Fig. 2(a), both our Lie-algebraic averaging method and the “Individual Update” method are identical in terms of the quality of the averaged motion model. This is so because both methods solve the same optimisation problem, i.e. the minimisation of the cost function as in Algorithm 2. In contrast, the averaging scheme of Sharp *et al.* [18] does poorly since it performs a suboptimal set of averages and as a result converges to a minimum that is different from the global minimum achieved by our method. However, it can be noted in Fig. 2(b) that although the “Individual Update” method achieves the same solution as our Lie-algebraic averaging it requires a significantly greater amount of time to achieve convergence. In the case of the method of [18], both the accuracy and speed are significantly poorer than our method. It may also be noted that for our method, the speed of convergence is almost independent of the amount of noise in the data. From these two observations, we can infer that our method of Lie-algebraic averaging outperforms the other two methods in terms of either accuracy or rate of convergence or both. As we will show in Section VII, the twin considerations of accuracy and rate of convergence have a significant impact on the performance of these averaging methods when used as the motion estimation step within a 3D registration iteration. Indeed, while the global minimum of the “Individual Update” method is the same as our Lie-algebraic averaging, during a single iteration the “Individual Update” method performs poorly when compared to our Lie-algebraic average update. Consequently, since after an individual update, the correspondences are also updated, the

overall result is that the performance of the algorithm proposed by [17] is poorer than our method. We also remark that in the experiment outlined in this section, we have used the optimal Riemannian distance metric as our intent was to compare the relative performance of the various averaging strategies. However, the motion estimation proposals of [17] and [18] use the quaternion representation and are therefore suboptimal methods.<sup>1</sup>

## V. MOTION AVERAGED ICP

As demonstrated in Section IV, our motion averaging method based on Lie-algebraic averaging carries out the optimal averaging of relative motions while being computationally efficient and having a fast rate of convergence. We now introduce our approach to solve for the global registration of scans by combining the basic ICP and the Lie-algebraic averaging method of [20]. Let  $\mathbf{M}_E = \{\mathbf{M}_{i_1 j_1}, \mathbf{M}_{i_2 j_2}, \dots, \mathbf{M}_{i_S j_S}\}$  denote the set of relative motions represented by the edges in  $E$  and  $S = |E|$ , where indices  $i_1 j_1, i_2 j_2$  etc. denote edges in  $E$ . Using this representation we can define the equivalent correspondence step and motion step for each iteration in our multiview extension. The reader may bear in mind that in our context of multiview registration, the motion estimation step outlined below consists of determining the individual pairwise motion steps followed by averaging them to obtain a motion step update that is consistent. Thus, each iteration of our motion averaged ICP consists of three steps: correspondence step, pairwise motion estimation step and motion averaging step.

**Correspondence Step:** Our correspondence step is a straightforward extension of that of the two-view ICP method. For every scan pair  $(i, j) \in E$ , we independently obtain correspondences by matching each point in scan  $i$  with its closest point in scan  $j$  (under the current relative transformation between the scan pair). In our implementation, we utilise the ideas summarised in [10]. At the end of the correspondence step we have correspondences determined for every scan pair in the viewgraph, i.e. for all  $S$  scan pairs.

**Motion Step:** In our multiview extension, for every scan pair  $(i, j) \in E$  we use the same estimation procedure as in the case of ICP, i.e. estimate the optimal motion between two point sets using the method of [5]. As a result, we have  $S$  estimates of the relative motions  $\mathbf{M}_{ij}$  which can be averaged using Algorithm 2 given above. While our methods of computing the rotations and translations are the same as that for ICP, in the context of motion averaging in our multiview extension we use the relative motions estimated in the following manner. In the two-view case, at the  $k$ -th iteration, we compute the Euclidean motion  $\Delta \mathbf{M}(k)$  that registers the second scan to the first one, i.e. in each iteration we transform the second scan. However, if we consider the second scan in its *original* frame of reference the net motion at the  $k$ -th iteration denoted as  $\mathbf{M}(k)$  is the

<sup>1</sup>If the Riemannian distance between two rotations  $\mathbf{R}_1$  and  $\mathbf{R}_2$  is  $\theta$ , then it can be shown that the distance measured in the quaternion representation is equal to  $2 \sin(\frac{\theta}{4})$ , i.e. the quaternion-based metric is suboptimal.



composition of all previous motion estimates, i.e.

$$\begin{aligned} \mathbf{M}(k) &= \prod_{i=1}^k \Delta \mathbf{M}(i) = \Delta \mathbf{M}(k) \mathbf{M}(k-1) \\ &= \Delta \mathbf{M}(k) \Delta \mathbf{M}(k-1) \cdots \Delta \mathbf{M}(2) \Delta \mathbf{M}(1) \end{aligned}$$

where  $\Delta \mathbf{M}(i)$  is the Euclidean motion computed at iteration  $i$ . In our notation here,  $\mathbf{M}(k)$  denotes any motion estimate at the  $k$ -th iteration and should not be confused with  $\mathbf{M}_k$  which denotes the motion of the  $k$ -th scan. In our method, during the  $k$ -th iteration, we average the relative motions  $\mathbf{M}_E(k) = \{\mathbf{M}_{i_1 j_1}(k), \mathbf{M}_{i_2 j_2}(k), \dots, \mathbf{M}_{i_s j_s}(k)\}$  using Algorithm 2 to get  $\mathbf{M}_{\text{global}}(k) = \{\mathbf{I}, \mathbf{M}_2(k), \dots, \mathbf{M}_N(k)\}$  which is our estimate for the global motion at the  $k$ -th iteration. Since we are interested in the averaged motion estimates to determine the scan motions for the next iteration (i.e. for the correspondence step of the next iteration), we replace each estimated relative motion in  $\mathbf{M}_E(k)$  with the global estimate, i.e.  $\mathbf{M}_j(k) \mathbf{M}_i^{-1}(k)$ . We can now state the complete motion step at the  $k$ -th iteration as:

- 1) compute incremental relative motions  $\Delta \mathbf{M}_E(k) = \{\Delta \mathbf{M}_{i_1 j_1}(k), \dots, \Delta \mathbf{M}_{i_s j_s}(k)\}$ ;
- 2) update relative motions in  $\mathbf{M}_E(k)$ :  $\mathbf{M}_{i_s j_s}(k) = \Delta \mathbf{M}_{i_s j_s}(k) \mathbf{M}_{i_s j_s}(k-1) \forall (i_s, j_s) \in E$ ;
- 3) compute  $\mathbf{M}_{\text{global}}(k) = \{\mathbf{I}, \mathbf{M}_2(k), \dots, \mathbf{M}_N(k)\}$  using Algorithm 2 on  $\mathbf{M}_E(k)$ ;
- 4) update relative motions  $\forall (i_s, j_s) \in E \mathbf{M}_{i_s j_s}(k) \leftarrow \mathbf{M}_{j_s}(k) \mathbf{M}_{i_s}^{-1}(k)$ .

Our algorithm can now be stated as a repeated iteration of the correspondence, motion and averaging steps till convergence. For notational convenience we denote this proposed method as MAICP for *motion averaged ICP*.

## VI. EFFICIENT VARIANTS OF MOTION AVERAGED ICP

In this section we present two variants of the MAICP method proposed in Section V, i.e. a causal version that only uses the relative motions of a scan with respect to previous ones in a sequence and a faster variant that utilises the transitivity of correspondence mappings.

### A. Causal Motion Averaging

While our MAICP approach effectively averages all available information, it is a batch process that assumes that all scans are available. In the scenarios where one builds 3D representations in an incremental fashion, e.g., during navigation using localisation and mapping (SLAM) or 3D modelling in real-time, the motion estimates will also have to be carried out in a causal fashion. Suppose, we have registered  $l-1$  scans and now need to register the  $l$ -th scan. If the registration process has to respect the causality principle, we are restricted to using the relative motions  $\mathbf{M}_{ij}$  where  $i \leq l$  and  $j \leq l$ , i.e. we cannot carry out a full motion averaging as outlined in Section V. However, as we shall describe now, even in such a restricted scenario it is possible to take advantage of the motion averaging principle while respecting causality.

The standard approach implied by ICP would compute  $\mathbf{M}_{l-1 l}$  and use it for registering the  $l$ -th scan. However, it is often the case that the  $l$ -th scan has an overlap with more than

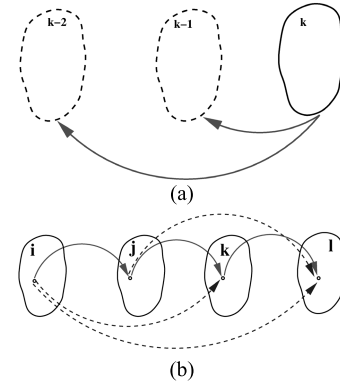


Fig. 3. (a) Causal motion averaging. Each scan is registered only to previous scans. (b) Transitive correspondences. Solid edges are correspondences in adjacent scans. Dashed edges represent other correspondence mappings. Transitivity implies that dashed correspondences can be established using the solid edges.

one of the previous scans, say with scans  $l-1$  and  $l-2$  (see Fig. 3(a) where the two relations are shown as solid edges). By using (1) as  $\mathbf{M}_{ij} \mathbf{M}_i = \mathbf{M}_j$ , we have two estimates of  $\mathbf{M}_l$ , i.e.  $\mathbf{M}_l = \mathbf{M}_{l-2} \mathbf{M}_{l-2}^{-1} \mathbf{M}_l$  and  $\mathbf{M}_l = \mathbf{M}_{l-1} \mathbf{M}_{l-1}^{-1} \mathbf{M}_l$  which are absolute motions that can be averaged by Algorithm 1. As a result, there is a reduction in the accumulation drift or error as the scan  $l$  is “tethered” to more than one previously registered scans. In fact we can obtain higher quality registration results if the current scan overlaps with multiple previous scans, but if the current scan has overlap with only one previous scan, our causal motion averaging method boils down to the conventional ICP method. In other words, our approach results in a motion averaging registration method that takes full advantage of all information available in a causal manner and is a generalisation of the conventional sequential registration procedure that uses ICP. We denote this method as MAICP (Causal).

### B. Using Transitivity for Correspondences

In comparison with using the standard ICP routine, in MAICP there are two additional computational steps. While for a sequence of  $N$  scans, the standard ICP method would need to estimate  $N-1$  sets of correspondences and motion estimates, in the case of MAICP we need to establish correspondences and motion estimates for  $S = |E|$  pairs of scans. Evidently, the quality of motion averaging is only improved to the degree that  $S > (N-1)$ , i.e. the amount of redundant information available. Now for each relative motion estimate obtained, we incur the computational expense of establishing a set of correspondences between the two scans involved. In other words, although a greater number of relative motions results in a higher quality motion estimate, the computational load is also correspondingly increased. In comparison with the additional computational load due to the need to solve for correspondences, the increased load due to motion averaging by Algorithm 2 is negligible. As a result, if we can reduce the computational demand due to an increased number of point correspondences to be established, we can improve the speed of our motion averaged ICP approach while retaining the



advantages of its accuracy. Indeed, precisely such an advantage is available by exploiting the transitivity of correspondence mappings as illustrated in Fig. 3(b).

Denote the 4 scans in Fig. 3(b) as  $\{i, j, k, l\}$  and consider a single point matched across all these scans and let it be denoted as  $\mathbf{p}_i$  in scan  $i$  etc. Further, we denote the correspondence mapping function that relates  $\mathbf{p}$  from scans  $i$  to  $j$  as  $\pi_i^j$ , i.e.  $\mathbf{p}_j = \pi_i^j(\mathbf{p}_i)$ . The standard ICP-based registration would require the correspondences between adjacent views, i.e. between pairs  $\{i, j\}$ ,  $\{j, k\}$  and  $\{k, l\}$  (shown in solid in Fig. 3(b)). However, to carry out our motion averaging estimate, we would also need additional correspondence mappings, i.e.  $\{i, k\}$ ,  $\{i, l\}$  and  $\{j, l\}$  denoted by dashed edges, which would appear to double the computational load. Here, we make the crucial observation that correspondence mappings are transitive, i.e. to establish the match in scan  $k$  for point  $\mathbf{p}_i$  we could use the path between  $\mathbf{p}_i$  and  $\mathbf{p}_k$  through  $\mathbf{p}_j$ . As a result,

$$\mathbf{p}_k = \pi_i^k(\mathbf{p}_i) = \pi_j^k(\pi_i^j(\mathbf{p}_i))$$

implying that  $\pi_i^k = \pi_j^k \circ \pi_i^j$ . Notice that both the mappings  $\pi_i^j$  and  $\pi_j^k$  are already computed for registering using ICP (i.e. the solid edges in Fig. 3(b)). Therefore, since the composition operation on two mappings is computationally trivial, establishing the correspondence map  $\pi_i^k$  does not incur any additional computational load.

To register all scans into a single frame of reference using the standard ICP routine, we only need  $N - 1$  correspondence mappings that form a spanning tree (ST) when considered as a viewgraph  $G$ . To reiterate, these are equivalent to the solid edges in Fig. 3(b). All other correspondence maps that we require (i.e. the dashed ones) are edges not on the spanning tree. Let us consider any such edge connecting vertices  $(r, s)$ . Since we have a spanning tree of correspondence mappings available, the correspondence mapping between scan  $r$  and scan  $s$  can be established as a composition of correspondence mappings along the edges on ST that connect vertices  $r$  and  $s$ . All such additional correspondence mappings belong to paths on ST of length greater than 1. As a result, the total computational load for establishing correspondences is the same as in the case of basic ICP, i.e.  $N - 1$  mappings since the additional load incurred in creating the composition maps using the transitive relationship is small. Also a point correspondence can be propagated along the ST only to the extent that it has correspondences along the edges of ST. We call our MAICP method using such transitive correspondences as MAICP (Transitive).

While the transitivity relationship as discussed above always holds, it must also be noted here that during the intermediate steps, each individual motion estimate is inaccurate. As a consequence, the correspondence mapping established by matching a point with its nearest neighbour on another scan may be inaccurate, i.e. the accuracy of using transitive correspondences relies on the correspondences between adjacent scans being accurate. In particular, if the error in the motion estimate is significant, it can lead to highly inaccurate point correspondences. Therefore, if we consider the direct correspondence between  $\mathbf{p}_i$  and  $\mathbf{p}_l$ , the error could be substantial

since the two views  $i$  and  $l$  are far apart to start with and during the intermediate steps the motion estimate  $\mathbf{M}_{il}$  is not very accurate. In contrast, each of the single steps from  $i$  to  $l$ , i.e.  $i \rightarrow j$ ,  $j \rightarrow k$  and  $k \rightarrow l$  are between adjacent scans that are closer to each other, implying that these correspondence mappings are expected to be more accurate. As a result each of these correspondence maps are expected to be reasonably accurate and their composition  $\pi_k^l \circ \pi_j^k \circ \pi_i^j$  can be expected to be closer to the true correspondence mapping than the directly computed correspondence mapping  $\pi_i^l$ . In other words, apart from the computational speed-up achieved by using the composition of correspondence mappings along a path on the spanning tree, the transitivity based correspondence method can be expected to be more accurate than the correspondence mapping established directly.<sup>2</sup> As we shall see later, in our experiments we use the output of our causal MAICP method to initialise the correspondence steps for multiview registration. As a result, it is always possible to reliably construct a spanning tree that ensures that the transitive approach is well initialised.

### C. Performance of Transitive Correspondence

We demonstrate the efficacy of using transitivity to solve for correspondences as discussed in Section VI-B via the following experiment. In the context of multiview scan registration, we need to consider the comparative performance of methods with respect to a) the amount of noise ( $\sigma$ ) and b) the (initial) amount of rotation prior to registration ( $\theta$ ). To generate a synthetic dataset, we randomly select 5000 points from the first scan of the well-known Stanford Bunny dataset [27]. Using these points, 4 additional point sets were generated by successively applying a rotation with a random axis. Here, scan  $l$  is obtained as a rotated version of the previous scan, i.e.  $l - 1$ . This ensures that each scan is rotated further away from scan 1. Finally, we add Gaussian noise to all scan points except to those from the first one. As a result, we now have 5 rotated and noisy point sets that need to be registered. Here the angle of rotation was uniformly sampled over the interval  $(0, \theta]$  where  $\theta \in \{2^\circ, 4^\circ, 6^\circ, 8^\circ, 10^\circ\}$ . In addition, we define the scale of the point set as the radius  $\mathbf{r}$  of its circumsphere. The  $\sigma$  of the Gaussian noise added to each dimension of a point is defined as a fraction of  $\mathbf{r}$ , i.e.  $\sigma \in \{1, 2, 3, 4, 5\}\%$  of  $\mathbf{r}$ . Such a parametrisation makes the noise level independent of the size of the object. For each  $\{\sigma, \theta\}$  pair, we perform 25 trials and average the results.

For each of the 5000 points from the first scan, we compute both the direct correspondence with points on scan 5 as well as the transitive correspondence estimated by the composition of the correspondence mappings between adjacent scans. Since this is a simulated scenario, we also know the correct correspondences. For both the direct and transitive correspondences we compute the root mean squared error (expressed in units of the mesh resolution for the Bunny scan). In Fig. 4(a) for each setting of  $\sigma$  the errors are averaged

<sup>2</sup>In the event that a correspondence exists but cannot be propagated along the spanning tree (e.g., between the last and the first scan in a turntable sequence), the correspondence is established in the usual direct manner.

TABLE I  
AVERAGE REGISTRATION ERRORS FOR REAL DATASETS (IN DEGREES). OUR MOTION-AVERAGED REGISTRATION METHODS SIGNIFICANTLY OUTPERFORM ICP AND THE OTHER MULTIVIEW REGISTRATION METHODS LISTED. SEE TEXT FOR DETAILS

	ICP	Other Multiview Methods		Multiview Motion Averaged ICP			
		Benjemaa <i>et al.</i> [17]	Sharp <i>et al.</i> [18]	Closure	Causal	MAICP	Transitive
Happy Buddha	5.78	1.96	1.53	2.62	3.67	0.85	0.91
Stanford Bunny	0.92	0.61	0.61	0.78	0.72	0.59	0.55
Stanford Dragon	7.69	3.75	5.54	5.51	5.72	2.74	3.39
Ohio Bunny	1.55	0.78	0.69	0.27	1.18	0.28	0.47
Ohio Pooh	13.33	1.55	1.00	0.68	4.52	0.5	0.36

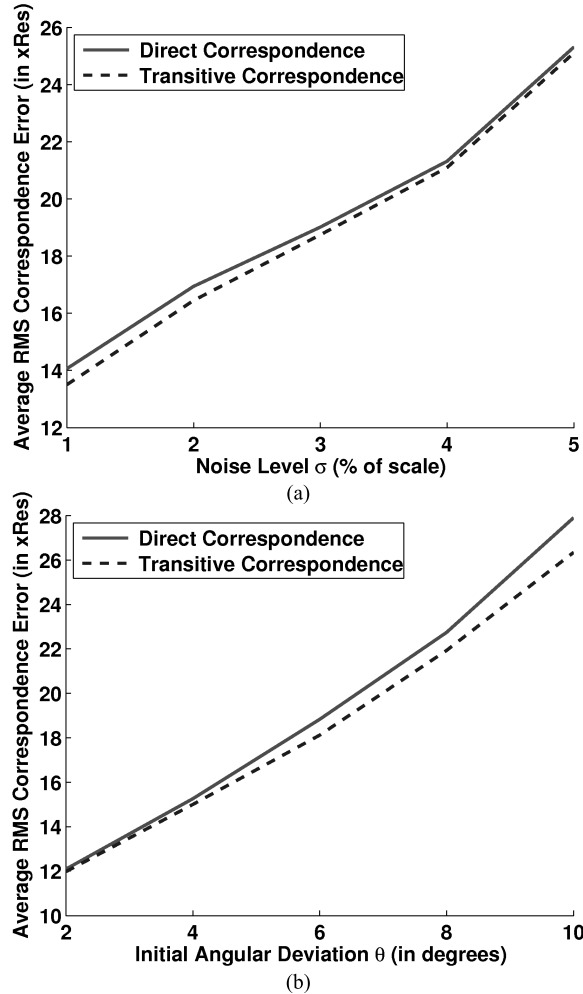


Fig. 4. Comparison of average correspondence error using direct and transitive correspondence mappings. The error shown is the average root mean squared error (in units of the mesh resolution) with respect to: (a) noise level  $\sigma$  and (b) rotation angle  $\theta$ .

over all values of  $\theta$  whereas in Fig. 4(b) for each setting of  $\theta$  the error is averaged over all values of  $\sigma$ . As is evident from both plots, the transitive correspondence mapping does as well or better than the direct method of estimating correspondence. It is particularly noteworthy that for large values of angular deviation  $\theta$ , the transitive method of estimating correspondences does significantly better than the direct method. The results of this experiment provide empirical justification for our use of the transitive correspondence mapping which is seen to be both efficient and accurate when compared to the direct method of using the nearest neighbour on a scan.

## VII. RESULTS

In this section we present our registration results for some well-known datasets and also compare with the methods of [17] and [18]. From the Stanford repository, we use the Bunny, Happy Buddha (standing) and Dragon datasets which have 10, 15 and 15 scans respectively [27]. Whereas the Bunny scan set has multiple viewpoints to cover the entire surface, the Happy Buddha and Dragon sets were acquired using a turntable rotated in steps of  $24^\circ$ . For all three datasets, the ground truth motion is available. In addition, we also used two datasets from the Ohio State repository, i.e. the Ohio Bunny and Pooh which consist of 18 turntable scans that are  $20^\circ$  apart [28]. In the case of these two datasets, the scans are of inferior quality and also the ground truth axes of rotations are not provided. Consequently, for measuring the registration error, for the Stanford repository datasets we compute the angular difference between the ground truth rotation and the estimated rotation matrix. However, for the two Ohio repository datasets, the angular error is simply the difference between the ground truth angle and the angle of the estimated rotation matrix, i.e. independent of the axes of rotations. For the experiments on the three Stanford datasets, we perturb the individual scans from the registered ground truth position by a rotation with a random axis and an angle randomly drawn from a uniform distribution over  $(0, 5^\circ]$ . In the case of the two datasets from the Ohio repository, since we do not have the ground truth motions, we simply initialised the registration with identity matrices as our motion estimates. All multiview averaging methods are initialised by the results of MAICP (Causal) which is itself initialised using the results of ICP. This ensures that we progressively register scans as they are acquired and at the end carry out a final multiview improvement. For estimating the relative motion  $\mathbf{R}_{ij}$  we randomly select 1000 points from scan  $i$  and find their correspondences on the full scan  $j$ . Finally, the results for the Stanford datasets are averaged over 25 trials whereas for the Ohio datasets, the error rate is for a single experiment. Apart from MAICP and its variants, we also introduce an additional method that demonstrates the power of consistent averaging. Since in a turntable sequence, the last scan often has a substantial area of overlap with the first one, it is possible to estimate their relative motion (see Fig. 1(b)). In a sequential registration method applied to a turntable sequence, the error accumulates over the sequence and the last scan might be significantly affected by this accumulated error. This increased error is remedied by the simplest possible motion averaging scheme, i.e. by “closing the loop”. Thus, along with the edges

TABLE II  
COMPUTATION TIME FOR REAL DATASETS (IN SECONDS). OUR MOTION-AVERAGED REGISTRATION METHOD IS FASTER THAN THE OTHER  
MULTIVIEW REGISTRATION METHODS LISTED. SEE TEXT FOR DETAILS

	ICP	Other Multiview Methods		Multiview Motion Averaged ICP			
		Benjemaa <i>et al.</i> [17]	Sharp <i>et al.</i> [18]	Closure	Causal	MAICP	Transitive
Happy Buddha	9.73	20.47	33.86	17.69	8.64	23.09	19.41
Stanford Bunny	4.8	3.32	12.7	1.39	2.56	2.61	2.5
Stanford Dragon	13.22	24.00	41.84	12.7	7.6	25.7	22.92
Ohio Bunny	30.17	4.56	68.28	5.19	3.54	3.96	3.12
Ohio Pooh	58.3	44.8	129.18	35.66	24.65	28.54	23.28

between adjacent scans (solid edges in Fig. 1(b)) we also consider the additional dashed edge between the last scan and the first scan which closes the loop along the sequence.<sup>3</sup> This method is denoted as MAICP (Closure). It may be noted that MAICP (Closure) uses only one additional constraint compared to ICP, i.e.  $N$  edges instead of  $(N - 1)$  edges of a spanning tree.

In Table I, we give the comparative performance of all of our methods and that of Benjemaa *et al.* [17] and Sharp *et al.* [18] on these datasets. It will be noticed that while MAICP (Causal) improves upon the results of the ICP method, it performs poorly compared to the other multiview approaches. This suggests that it is crucial to utilise all available motion constraints for effective averaging. This view is borne out by the superior performance of all other MAICP variants and by the fact that MAICP does better compared to the other multiview approaches since it utilises all available information. The importance of imposing consistency criteria on the relative motions is clearly demonstrated by the fact that the mere addition of a single additional relative motion in MAICP (Closure) results in a significant reduction of error. This is so because the single constraint between the last and the first scan forces the averaging algorithm to redistribute the errors uniformly over the sequence so as to achieve a reduction in the overall error thereby overcoming the problem of drift in ICP.

It is also of interest to note that MAICP (Transitive) which uses only the correspondence maps obtained by the spanning tree of the ICP-based correspondences is very close in performance to the solution of MAICP providing additional validation for the proposed use of transitivity of correspondences in Section VI-B. Interestingly, for the Ohio Pooh dataset, MAICP (Transitive) does better than that of MAICP. This can be explained by the fact that in the Ohio Pooh dataset, the viewgraph is densely connected, thus allowing for the transitive correspondences to be accurately computed. This suggests that when the viewgraph is densely connected, MAICP (Transitive) can be expected to do as well or better than MAICP itself. In Table II, we also give the computation time for the different methods for a MATLAB implementation run on a 2.67GHz machine. As can be observed, our methods of MAICP and MAICP (Transitive) are computationally efficient while providing accurate performance. In particular, MAICP (Transitive) is consistently faster than both [17]

and [18]. Also the method of [18] is significantly slower than our methods.

Apart from the accurate performance of the MAICP method and its variants, it will also be noted that the methods of [17] and [18] perform with less accuracy. The poor performance of [18] is attributable to the suboptimal nature of its motion averaging step. In contrast, at first glance, the fact that the method of Benjemaa *et al.* [17] performs poorly would seem to be an anomaly since in Section IV we have noted that the abstraction of their motion averaging routine performs as well as that of the Lie-algebraic averaging method of Algorithm 2 used in this paper. However, there is a crucial observation to be made that explains this apparent difference. In the experiments of averaging many relative rotation estimates described in Section IV, the abstracted motion averaging method of [17], i.e. the “Individual Update” method converged to the same estimate as the Lie-algebraic averaging method but took many more iterations to do so. In the algorithm for registering 3D scans due to [17], the motion estimates of each scan undergo a *single* “Individual Update” in an iteration. In other words, at any given iteration of their routine, [17] does not update the motion models repeatedly till convergence but merely carry out a single update. With reference to the experiment in Section IV this would amount to carrying out a single update of  $\mathbf{M}_{\text{global}}$ , i.e. not till convergence. As we have seen, since their motion averaging scheme is slow, each individual update only results in a marginal improvement of the global motion estimate. Thus when this marginally improved estimate is repeatedly used to establish scan point correspondences, the resulting correspondence set is poor in quality. When in the motion step these correspondences are used, we get a poor estimate of the relative motion  $\mathbf{M}_{ij}$ . As a consequence, the overall 3D registration routine of [17] has poor performance. In contrast, in our case, in the motion step of each iteration, the relative motions  $\mathbf{M}_{ij}$  are averaged to convergence. As a result, in the correspondence step of the next iteration, the scans are better aligned resulting in improved correspondences. The overall effect is to lead our algorithm to converge to an accurate solution.

In addition to Table I, we also illustrate the performance of different methods in Fig. 5 in the form of cross-sections of registered scans for a single trial. The corresponding regions are indicated on the registered 3D models. It can be seen that our method accurately registers all the five datasets as opposed to [17] and [18] that show misregistrations on different datasets. Due to its sequential nature, ICP exhibits poor registration and has significant misalignments. In the

<sup>3</sup>In the case of the Stanford Bunny dataset, which is not a turntable sequence, we create an ordering of scans by defining a spanning tree.


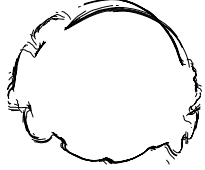
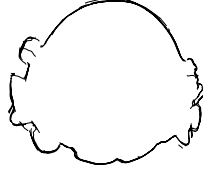
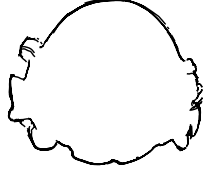




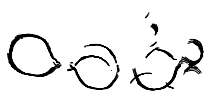


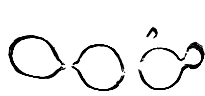
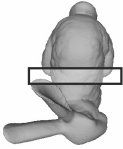


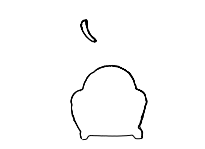
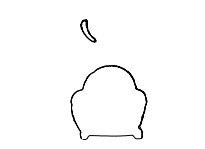


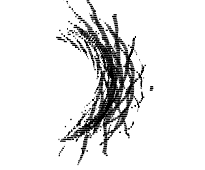


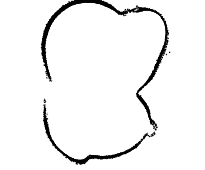
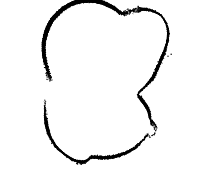






Happy Buddha 					
Dragon 					
Stanford Bunny 					
Ohio Pooh 					
Ohio Bunny 					
Model	Initial	ICP	Benjemaa[17]	Sharp[18]	MAICP

Fig. 5. Cross sections of registered scans for different methods.

Happy Buddha dataset, it will be noted that the method of [17] also does poorly as it is unable to distribute the errors well enough. Similarly in the case of the Dragon dataset, the method of [18] shows a significant error in registering some of the scans. However, in both the cases of the Happy Buddha and Dragon datasets, our motion averaging scheme works well since it uses all the relative motion constraints in a single motion averaging step thereby forcing the errors to be evenly distributed over the entire scan sequence. In the case of the Ohio datasets all the methods work well. While for the Pooh dataset, the quality of registration of the different multiview methods is virtually indistinguishable in visual terms, for the Ohio Bunny dataset, one will notice that the method of [18] has a somewhat poorer quality of registration as evidenced by the slight misregistration shown on the left part of the cross-sections shown. In any event, despite the visual similarity of quality of registration, our MAICP performs better than the other two methods for the Ohio datasets as can be clearly seen from the rows of Table I corresponding to these datasets. Similarly, over multiple trials the average performance of our algorithm on the Stanford datasets is substantially better than the other methods. This implies that under different initial

conditions the other algorithms fail to converge to a minima of the same quality resulting in larger average errors. Our motion averaging method does not suffer from this limitation and has a greater tolerance for the initial motion estimates used to start the iterations. Moreover, it is faster in convergence than the other multiview methods of [17] and [18].

While our averaging method was significantly faster than the other methods in Section IV, in Table II the relative difference is reduced. This is because in Table II, most of the computational time for all methods is spent on the correspondence step, hence the relative speed of our motion averaging method is not fully reflected in the total computation time. This is the case with the Stanford dataset experiments where we provide all methods a good initial guess using MAICP (Causal). However, the significance of the speed of our method is quite evident in the computation times of the Ohio Pooh dataset which is initialised with a global motion far from the ground truth. In this case, [17] takes much longer to converge compared to our MAICP approach.

As a result of all these observations, the inferences to be made from our experiments are that 1) it is important to use as many relative motion estimates as possible and 2) it is

TABLE III  
COMPARISON OF OUR MODIFICATION OF THE METHOD OF BENJEMAA *et al.* [17]

	Average Error (in Degrees)			Computation Time (in Seconds)		
	Benjemaa <i>et al.</i> [17]	Modified Benjemaa	MAICP	Benjemaa <i>et al.</i> [17]	Modified Benjemaa	MAICP
Happy Buddha	1.96	1.33	0.85	20.47	58.0	23.04
Stanford Bunny	0.61	0.58	0.59	3.32	5.6	2.61
Stanford Dragon	3.75	2.35	2.74	24.00	59.00	25.7
Ohio Bunny	0.78	0.46	0.28	4.56	123	3.96
Ohio Pooh	1.55	0.55	0.5	44.8	633	28.54

crucial that during each iteration, the motion averaging be carried out till convergence to obtain the greatest accuracy of registration possible. We provide further validation of the insight provided by our experimentation by demonstrating how the quality of the results of [17] can be improved by modifying it. As we have noted, in [17] during the motion step, the motion of each scan is updated in a random sequence. Since these individual updates are carried out only once, the best possible motion averaging is not available in [17]. However, we can modify the method of [17] in the following manner. During the motion step, instead of carrying out a single round of “Individual Update” of the motions, we repeatedly carry out the “Individual Update” steps till convergence. We denote this method as “modified Benjemaa” and also emphasise the point that this modification is possible due to the insight provided by our approach and is not available either in the original paper [17] or in the rest of the literature. In Table III, we compare the accuracy of the original method in [17], our method of MAICP and the “modified Benjemaa” method. An important observation is that allowing the motion updates to be averaged till convergence in “modified Benjemaa” leads to a greater accuracy than the original method of [17]. Indeed, in the case of the Stanford Dragon dataset, the average error over 25 trials of the “modified Benjemaa” method is lower than that of MAICP itself. However, when we compare the computation time required for each method, it will be immediately noted that although the “modified Benjemaa” method provides better accuracy, it comes at the cost of requiring much more computation time than either MAICP or the original formulation of [17]. This is so because, as we have shown in Section IV the “Individual Update” is far slower in achieving convergence when compared to Lie algebraic averaging. This difference in speed is particularly pronounced in the case of the two Ohio datasets where the individual scans are noisy, i.e. of low quality. A final point we make in this context is that while in Section IV, the accuracy of Lie algebraic averaging is identical to that of the “Individual Update” method, this is not so when we compare MAICP with the “modified Benjemaa” method. This difference arises since in the MAICP method, we extract individual pairwise relative motions and then average them on the Lie group. In contrast, in keeping with the approach of [17], using the current motion estimates we repeatedly transform the correspondences till convergence is achieved. In other words, in the case of MAICP the averaging is carried out on the relative motions, whereas in the case of “modified Benjemaa” the averaging is carried by directly using the transformed 3D

point correspondences as is done in the original formulation of [17]. Thus, we conclude that in terms of accuracy and computational speed, our MAICP approach outperforms the methods of Sharp *et al.* [18], Benjemaa *et al.* [17] and our modification of [17] that we call the “modified Benjemaa” method.

### VIII. CONCLUSION

In this paper we have introduced an extension of the basic ICP algorithm to the context of simultaneous registration of multiple scans. Our approach uses the method of Lie-algebraic averaging of relative motions to create an efficient and accurate multiview scan registration technique. In addition to demonstrating the superiority of our method when compared with similar algorithms in the literature, we have also introduced two variants of our method, i.e. one relying on causal motion averaging and the other exploiting the transitive nature of point correspondence mappings. While the causal variant improves upon ICP, the transitive method performs as well as our motion averaging technique while only incurring a marginal computational overhead. In addition, we have also demonstrated how the use of the insights provided by our experiments can be used to improve the accuracy of the method of [17].

### ACKNOWLEDGMENT

The authors would like to thank A. Chatterjee for the experiments in Section IV and the reviewers for their comments that have helped improve the presentation in this paper.

### REFERENCES

- [1] A. Pooja, and V. M. Govindu, “A Multi-view extension of the ICP algorithm,” in *Proc. 7th Indian Conf. Comput. Vis. Graph. Image Process.*, 2010, pp. 235–242.
- [2] R. Szeliski, “Image alignment and stitching: A tutorial,” *Found. Trends Comput. Graph. Comput. Vis.*, vol. 2, no. 1, pp. 1–104, Dec. 2006.
- [3] P. Besl and N. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [4] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, Apr. 1992.
- [5] S. Umeyama, “Least-squares estimation of transformation parameters between two point patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, Apr. 1991.
- [6] K. Kanatani, “Analysis of 3-D rotation fitting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 543–549, May 1994.
- [7] T. Masuda, K. Sakaue, and N. Yokoya, “Registration and integration of multiple range images for 3-D model construction,” in *Proc. Int. Conf. Pattern Recognit.*, vol. 1. Aug. 1996, pp. 879–883.

- [8] C. Dorai, G. Wang, A. K. Jain, and C. Mercer, "Registration and integration of multiple object views for 3D model construction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 83–89, Jan. 1998.
- [9] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau "Registration of multiple range views for automatic 3-D model building," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1994, pp. 581–586.
- [10] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. Int. Conf. 3D Digit. Image Model.*, Jun. 2001, pp. 145–152.
- [11] S. Cunningham and A. J. Stoddart, "N-view point set registration: A comparison," in *Proc. Brit. Mach. Vis. Conf.*, 1999, pp. 234–244.
- [12] S. Krishnan, P. Lee, J. Moore, and S. Venkatasubramanian, "Global registration of multiple 3D point sets via optimization-on-a-manifold," in *Proc. Eurograph. Symp. Geometry Process.*, 2005, pp. 187–196.
- [13] S.-W. Shih Y.-T. Chuang, and T.-Y. Yu, "An efficient and accurate method for the relaxation of multiview registration error," *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 968–981, Jun. 2008.
- [14] K. Pulli, "Multiview registration for large data sets," in *Proc. Int. Conf. 3-D Digit. Image Model.*, 1999, pp. 160–168.
- [15] T. Masuda, "Generation of geometric model by registration and integration of multiple range images," in *Proc. 3-D Digit. Image Model.*, 2001, pp. 254–261.
- [16] K. Nishino and K. Ikeuchi, "Robust simultaneous registration of multiple range images," in *Proc. Asian Conf. Comput. Vis.*, Jan. 2002, pp. 454–461.
- [17] R. Benjema and F. Schmitt, "Fast global registration of 3D sampled surfaces using a multi-z-buffer technique," in *Proc. Int. Conf. 3-D Dig. Image Model.*, May 1997, pp. 113–120.
- [18] G. Sharp, S. Lee, and D. Wehe, "Multiview registration of 3D scenes minimizing error between coordinate frames," *IEEE Trans. Pattern Anal. Mach. Int.*, vol. 26, no. 8, pp. 1037–1050, Aug. 2004.
- [19] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vis. Comput.*, vol. 25, no. 5, pp. 578–596, May 2007.
- [20] V. M. Govindu, "Lie-algebraic averaging for globally consistent motion estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jul. 2004, pp. 684–691.
- [21] K. Kanatani, *Group-Theoretical Methods in Image Understanding*, New York, USA: Springer-Verlag, 1990.
- [22] V.S. Varadarajan, *Lie Groups, Lie Algebras and Their Representations*, vol. 102, New York, USA: Springer-Verlag, 1984.
- [23] S. Farsiu, M. Elad, and P. Milanfar, "Constrained, globally optimal, multi-frame motion estimation," in *Proc. IEEE/SP 13th Workshop Stat. Signal Process.*, Jul. 2005, pp. 1396–1401.
- [24] P. Schroeder, A. Bartoli, P. Georgel, and N. Navab, "Closed-form solutions to multiple-view homography estimation," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2011, pp. 650–657.
- [25] V. M. Govindu, "Robustness in motion averaging," in *Proc. Asian Conf. Comput. Vis.*, 2006, pp. 457–466.
- [26] R. Hartley, K. Aftab, and J. Trumpf, "L1 rotation averaging using the weiszfeld algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3041–3048.
- [27] *Stanford University 3D Scan Repository*. (2011) [Online]. Available: <http://graphics.stanford.edu/data/3DScanRep>
- [28] *Ohio State University Range Image Database*. (2010) [Online]. Available: <http://www.sample.ece.ohio-state.edu/data/3DDb/RID/index.htm>
- [29] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Toward a general multi-view registration technique," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 5, pp. 540–547, May 1996.
- [30] J.H. Manton, "A globally convergent numerical algorithm for computing the centre of mass on compact lie groups," in *Proc. Control, Autom. Robot. Vis. Conf.*, vol. 3, Dec. 2004, pp. 2211–2216.
- [31] T. Vercauteren, A. Perchant, G. Malandain, X. Pennec, and N. Ayache, "Robust mosaicing with correction of motion distortions and tissue deformations for in vivo fibered microscopy," *Med. Image Anal.* vol. 10, no. 5, pp. 673–692, Oct. 2006.
- [32] M. Moakher, "Means and averaging in the group of rotations," *Soc. Indian Automobile Manuf. J. Matrix Anal. Appl.*, vol. 24, no. 1, pp. 1–16, 2002.



**Venu Madhav Govindu** received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA.

He is currently an Assistant Professor with the Department of Electrical Engineering, Indian Institute of Science, Bengaluru, India. His current research interests include geometric and statistical inference problems in computer vision.



**Pooja A.** received the M.Sc. (Engg.) degree from the Department of Electrical Engineering, Indian Institute of Science, Bengaluru, India.

She is currently a Machine Learning Analyst with Amazon India. Her current research interests include computer vision and machine learning.