frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 17 July 2023
DOI 10.3389/frobt.2023.1019579

OPEN ACCESS

EDITED BY
Arturo Gil Aparicio,
Miguel Hernández University of Elche, Spain

REVIEWED BY
David Valiente,
Miguel Hernández University of Elche, Spain
Luis Paya,
Miguel Hernández University of Elche, Spain

*CORRESPONDENCE
Alen Alempijevic,
Alen.Alempijevic@uts.edu.au

RECEIVED 15 August 2022
ACCEPTED 09 May 2023
PUBLISHED 17 July 2023

CITATION
Maleki B, Falque R, Vidal-Calleja T and Alempijevic A (2023), SPaM: soft patch matching for non-rigid pointcloud registration.
Front. Robot. AI 10:1019579.
doi: 10.3389/frobt.2023.1019579

SPaM: soft patch matching for non-rigid pointcloud registration

Behnam Maleki, Raphael Falque, Teresa Vidal-Calleja and Alen Alempijevic*

Robotics Institute, University of Technology Sydney, Ultimo, NSW, Australia

3d reconstruction of deformable objects in dynamic scenes forms the fundamental basis of many robotic applications. Existing mesh-based approaches compromise registration accuracy, and lose important details due to interpolation and smoothing. Additionally, existing non-rigid registration techniques struggle with unindexed points and disconnected manifolds. We propose a novel non-rigid registration framework for raw, unstructured, deformable point clouds purely based on geometric features. The global non-rigid deformation of an object is formulated as an aggregation of locally rigid transformations. The concept of locality is embodied in soft patches described by geometrical properties based on SHOT descriptor and its neighborhood. By considering the confidence score of pairwise association between soft patches of two scans (not necessarily consecutive), a computed similarity matrix serves as the seed to grow a correspondence graph which leverages rigidity terms defined in As-Rigid-As-Possible for pruning and optimization. Experiments on simulated and publicly available datasets demonstrate the capability of the proposed approach to cope with large deformations blended with numerous missing parts in the scan process.

KEYWORDS

deformable registration, non-rigid registration, soft patches, patch matching, pointcloud registration, as rigid as possible

1 Introduction

Given only point clouds, a common solution to the registration problem is to leverage mesh reconstruction, which is challenging for many depth sensors due to sensor noise, missing parts, and holes (stemming from occlusions). Mesh-based approaches [for example, leveraging Poisson surface reconstruction (Kazhdan et al., 2006)] are mainly suited to noise-free, water-tight surfaces. These methods generally involve interpolation and smoothing, consequently compromising the registration accuracy and potentially losing important details. Besides, other mesh reconstruction approaches (such as Ball Pivoting) leave some un-indexed points and disconnected manifolds, posing a challenge for state-of-the-art non-rigid registration techniques such as functional maps (Ovsjanikov et al., 2012).

If one were asked to manually determine the corresponding parts of two 3D point clouds (particularly of the challenging case of a large, featureless, semi-flat surface), one would start by taking the most conspicuous areas on one scan and look for correspondences on the other. By using these matched sections as the initialization step, the adjacent parts could be compared and evaluated in a transitive and progressive manner to grow a network of correspondences.

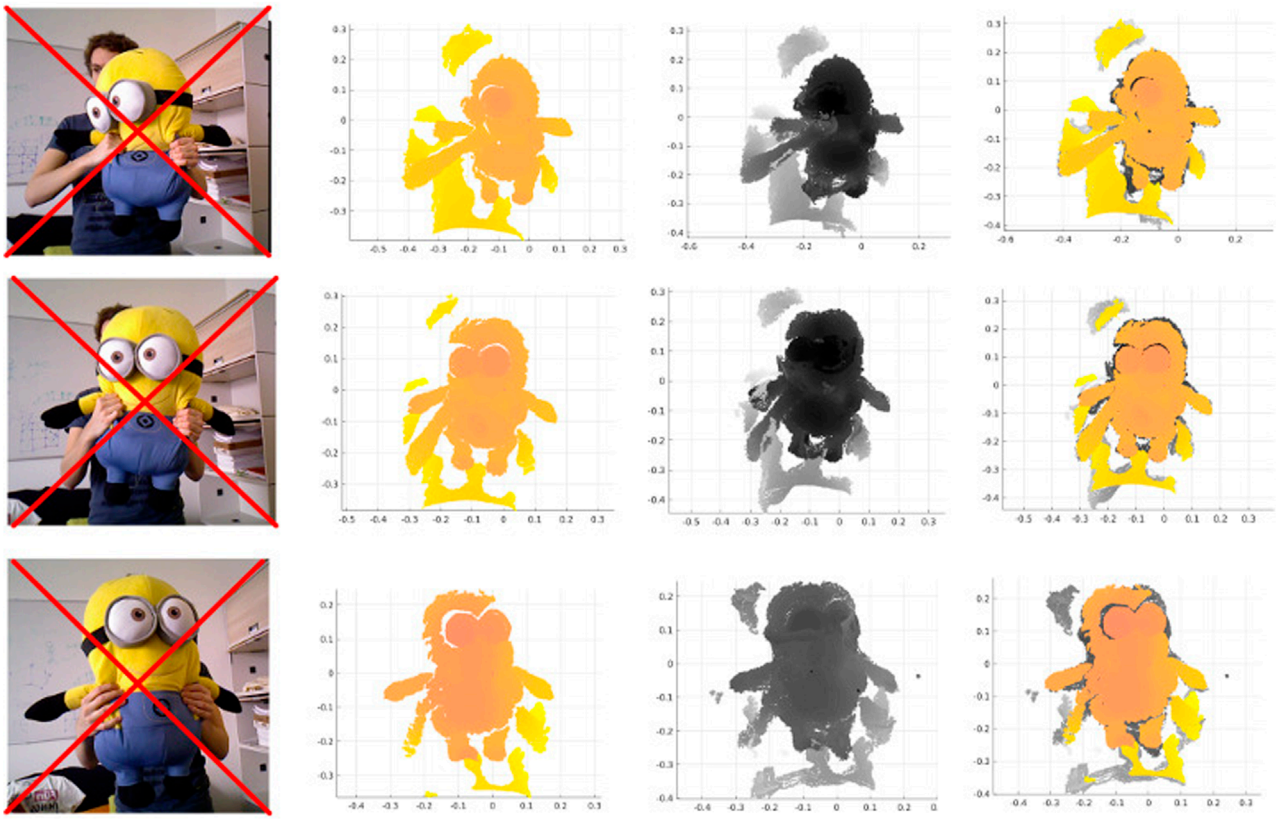

FIGURE 1
RGB image (not used), current frame, canonical frame and their registration for Minion dataset.

Following this idea, we propose a framework that is based on locally rigid patches and relies merely on geometrical features of 3D point clouds. Our proposed approachassumes that the aggregation of entire local rigidities accounts for the global non-rigidity of a deformable object throughout consecutive scans. Hence, we employ a concept of soft patches, whose primary benefit (compared with individual points), is a significant reduction of computational complexity and time.

The main contribution of our work is a meshless, visually-featureless, model-free, topology-aware, and geometry focused approach for 3D non-rigid registration that enables the 3D reconstruction of deformable objects. **Figure 1** shows an example of this registration on a public dataset. According to Cadena et al. (2016), our soft patch can be categorized as a low-level but flexible dense representation (with negligible computational overhead). In our pipeline, we obtain a sparse correspondence between (not necessarily consecutive) scans, which enables the approach to cope with small to large local deformations. We evaluate the proposed 3D registration method as well as the full reconstruction pipeline in simulated and publicly available datasets, demonstrating the validity of our approach.

2 Related work

The problem of registering successive belonging to deformable objects is often encountered in 3D mapping algorithms. Model or template-based methods for instance can handle severe deformation of an object quickly and effectively (Petit et al., 2017). The authors register a segmented point cloud in a rigid manner and then model elasticity by non-rigidly fitting a mesh based on the finite element method. In other approaches, an articulated motion model or additional proprioceptive sensors are used as priors to constrain and track the frame-to-frame non-rigid deformations to improve the reconstruction quality for fast body motion (Yu et al., 2017; 2018; Zheng et al., 2018). This reliance on other sensors or priors severely limits the applicability of these approaches.

Their surface representation based on Truncated Signed Distance Function (TSDF) is extracted by marching cubes and stored as a polygon mesh with point-normal pairs in the canonical frame. A single volume is registered to a single point in time (canonical frame), Dou et al. (2016) claim that the frame-to-frame motions in DynamicFusion are slow and carefully controlled due to the assumption on closest point correspondences between volume and frame. Thus, the approach cannot handle drastic deformation attributed to the challenges of fusing data back into a single model.

While DynamicFusion solely uses geometric correspondences, VolumeDeform (Innmann et al., 2016), upon generating a polygonal mesh, additionally takes advantage of RGB data via sparse globally consistent SIFT features to improve the alignment process. These features serve as global anchor points to mitigate drift and enable handling tangential motion. The methods relying on visual and colour features of the scene fail in the corresponding stage in the

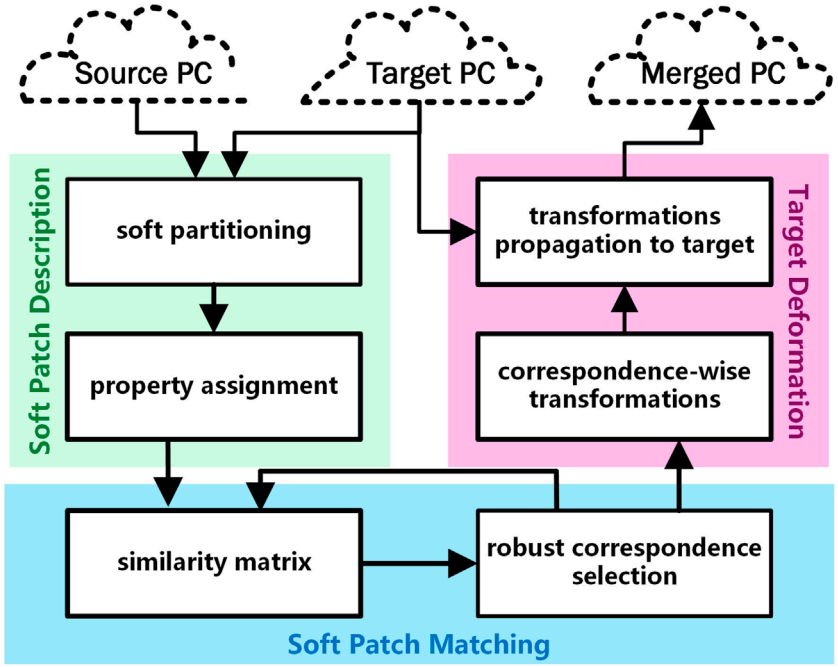

FIGURE 2
Flowchart of the devised framework.

presence of visually featureless objects, poorly-illuminated scenes, or drastic changes of view.

The noted approaches generally use a mesh for establishing correspondences and extracting features for inter-frame motion tracking. However, surface reconstruction in the case of sparse and noisy point clouds is a cumbersome task. Further, the method proposed by Newcombe et al. (2015) and Innmann et al. (2016) may fail under larger inter-frame motion due to the underlying mesh-based correspondence estimation (Slavcheva et al., 2017).

The recent works on rigid and non-rigid registrations make extensive use of the Signed Distance Function (SDF) and its different variants such as TSDF, probabilistic SDF (PSDF), and Euclidean SDF (ESDF) (Gupta et al., 2016; Slavcheva et al., 2017; Yu et al., 2017). This body of work has proven to be effective volumetric representations of a scene. To address the sensor noise, they smooth out errors in cumulative models. Overall, the performance of these algorithms highly depends on the scenario, and the capability to handle the range and speed of deformation is always a compromise against fidelity. In contrast to existing dense SLAM approaches, the recent work SurfelWarp (Gao and Tedrake, 2019) enhances DynamicFusion by replacing volumetric data structures with surfel-based representation of geometry and a deformation field. In their pipeline for aligning the reference frame with the current one, they first initialize the deformation field from the previous frame and then similarly to DynamicFusion (Newcombe et al., 2015) deploy an iterative closest point (ICP) based optimization to estimate the non-rigid warp field.

The majority of the non-rigid registration approaches are taking advantage of the transformation regularization term in their cost function. However, in motion boundaries (discontinuities of the ground-truth optical flow between 2 frames), it may create artifacts. Addressing this issue, Zampogiannis et al. (2019) used two notions of contact and separation to annotate the changing topology of a scene and then blended two forward and (inverted) backward warp fields locally, according to the type and proximity of detected events. The warp fields are computed separately by non-rigid ICP, and the correspondence association is improved by the frame image.

3 Overview

Given two *oriented* unorganized point clouds of a deforming object captured in different timestamps, one is used as the target point cloud, $\mathcal{P}$, and the second as the source, $\mathcal{P}$. The objective is then to find the local rigid transformations, to register the associated soft patches from source, $\mathbf{C}$, onto the corresponding soft patches of the target, $\mathbf{C}$, to transform and gradually register the entire source point cloud onto the target.

The proposed framework includes three key constituents as describing the patches, evaluating the metrics of associations to establish the correspondence, and deforming the source locally. A flowchart schematic of our framework is depicted in **Figure 2**.

4 Soft patch description

4.1 Soft partitioning

Thus the definition of locality plays a pivotal role. Given two pointclouds from the same surface labelled as target and source

which one has undertaken some deformation with respect to the other, the locality concept is then applied by independently subdividing these pointclouds into what we call partitions or patches. Let us assume that the corresponding partitions on the target and the source are rigidly transformed patches from one to the other. Then, it is possible to find the rigid transformation undergone by each patch individually, which consequently allows deforming the source non-rigidly to register with the target.

The aforementioned partitions have soft (in contrast to hard) boundaries, implying that each point in the pointcloud can belong to neighbouring patch (es) according to a measure called membership score. More specifically, the source and target are softly partitioned to create overlapping patches. Formally let us denote a pointcloud by $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}\}$, such that the coordinate of the ith point is $\mathbf{p}_i \in \mathbb{R}^3$ and the number of points in $\mathcal{P}$.

To differentiate between the target and the source properties, we make use of and superscripts, e.g., and denotes the number of points in target and source pointclouds, respectively. Note that multiple factors need to be considered while creating the partitions; the number of partitions, noted, should be proportional to the number of points in the pointcloud to generate meaningful patch descriptors. Also, the average number of points in patches, q, should be sufficient to create reliable local patch features as well as avoiding computational overhead. The other factor is the overlapping ratio of patches, so-called *softness* τ. Let $\mathcal{C}$ denote the set of overlapping patches $\mathcal{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}\}$, then by following the aforementioned factors, the number of partitions for each pointcloud is computed by $\coloneqq \lceil /(q \times (1-\tau)+1) \rceil$. The choice of the partitioning technique is irrelevant as long as the partitions are overlapping and the softness is controlled. In this work, we opt for k-medoids (Park and Jun 2009) with Euclidean distance, which is a variant of k-means with data points chosen as the centroids.

As a brief note in notation, throughout this work, the sequence of partition is denoted by centroids $\mathbf{M} = (\mathbf{m}_i | i = 1)$ and $\mathbf{M} = (\mathbf{m}_i | i = 1:)$, where and correspond to the number of partitions in the target and the source, respectively.

4.2 Property assignment

Given the set of soft patches, the centroids are accompanied by a distance matrix $\mathbf{B}_\times$ whose element d_{ij} is the Euclidean distance from $\mathbf{p}_i$ to the centroid of $\mathbf{C}_j$:

$$\mathbf{B}_\times \coloneqq \left[d_{ij} \middle| d_{ij} = \left\| \mathbf{p}_i - \mathbf{m}_j \right\|_2 ,\ i = 1:\ ,\ j = 1: \right] \tag{1}$$

where $\|.\|_2$ denotes L_2 norm. In order to determine whether a point p_i belongs to the soft patch C_j, the term d_{ij} should be smaller than a threshold. To avoid having patches with largely varying sizes, we consider a single cut-off threshold for the distance, $\bar{d}$. So, for each patch C_j, we find the distance with which q points fall inside its boundary, and then computing the average of all these distances yields the above mentioned cut-off threshold. Thus, the patches are given by:

$$\mathbf{C}_j = \left\{ \mathbf{p}_i \middle| \left\| \mathbf{p}_i - \mathbf{m}_j \right\|_2 \le \bar{d} ,\ i \in \{1:\} ,\ j \in \{1:\} \right\}. \tag{2}$$

To evaluate the similarity between 3D soft patches in different point clouds of the same surface, we use 3D SHOT descriptor (Salti et al., 2014), which produces a 352-tuple SHOT descriptor vector per point, $\mathbf{p}_i$. Let us define this descriptor vector of each point as *point SHOT*, $\mathbf{D}_{SH}(\mathbf{p}_i)$. Then for each (soft) patch, $\mathbf{C}_j$, we define a so-called descriptor called *patch SHOT*, $\mathbf{D}_{CSH}(\mathbf{C}_j)$, which is the normalised element-wise average of all including point SHOTS in $\mathbf{C}_j$:

$$\mathbf{D}_{CSH}(\mathbf{C}_j) \coloneqq \frac{\sum_{\mathbf{p}_i \in \mathbf{C}_j} \mathbf{D}_{SH}(\mathbf{p}_i)}{j} \times \left\| \frac{\sum_{\mathbf{p}_i \in \mathbf{C}_j} \mathbf{D}_{SH}(\mathbf{p}_i)}{j} \right\|_2^{-1}. \tag{3}$$

In this framework, the neighbourhood of every patch is a critical property that offers a reliable measure in the association stage. Hence, a complete neighbourhood map of every patch is generated, which consists of the index of adjacent patches, their related patch SHOTs, and the directional position of neighbours. The adjacent patches are defined by $\mathcal{N}(\mathbf{C}_j) \coloneqq \left\{ \mathbf{C}_k \middle| \ \|\mathbf{m}_k - \mathbf{m}_j\|_2 \le 2\bar{d} \ ,\ j \wedge k \in \{1:\} \right\}$,

To preserve the neighbourhood and increase the correspondence likelihood of adjacent patches of already matched patches, each patch is attributed with a topological constraint given its neighbourhood. We name this property *directional neighbourhood*, and define it as a set of 3D Euclidean unit vectors (illustrated in green in **Figure 3**) representing the direction of the lines connecting the centre of a query patch to the centres of its adjacent patches:

$$\mathbf{V}_{\mathbf{C}_j} \coloneqq \left\{ \vec{\mathbf{v}}_k \middle| \vec{\mathbf{v}}_k = \frac{\mathbf{m}_{\mathcal{N}^k(\mathbf{C}_j)} - \mathbf{m}_{\mathbf{C}_j}}{\|\mathbf{m}_{\mathcal{N}^k(\mathbf{C}_j)} - \mathbf{m}_{\mathbf{C}_j}\|_2} ,\ k = 1: \right\} \tag{4}$$

where $\mathcal{N}^k(C_j)$ is the kth element in the neighbors list of C_j.

5 Soft patch matching

5.1 Patch similarity

We propose two types of metrics to measure the pairwise similarity of the soft patches $\mathbf{C}_i$ and $\mathbf{C}_j$. These two metrics aim to reflect the confidence on the patch association. The definitions of the metrics are as follows.

5.1.1 SHOT vector distance

To measure the similarity of two patches SHOT descriptor vectors, $\mathbf{D}_{CSH}(\mathbf{C}_i)$ and $\mathbf{D}_{CSH}(\mathbf{C}_j)$, in addition to the original 352-D patch SHOT vector, we also use a 32-D Short SHOT (Seib and Paulus, 2018) vector. The Short SHOT is less dependent on the point normals and beneficial in the presence of noise. In our measure, the similarity obtained from the short SHOT variant, $\mathbf{D}^s_{CSH}(.)$ is combined with the one from the original Long SHOT, $\mathbf{D}^l_{CSH}(.)$. Then, the SHOT Vector distance d_{SV}, uses L_1 and L_2 norms of both variants:

$$\begin{aligned} d_{SV}(\mathbf{C}_i, \mathbf{C}_j) = \frac{1}{\beta_l + \beta_s} \sum_{y \in \{l,s\}} \frac{\beta_y}{\alpha_1 + \alpha_2} \times \big(& \alpha_1 \|\mathbf{D}^y_{CSH}(\mathbf{C}_i) \\ & - \mathbf{D}^y_{CSH}(\mathbf{C}_j)\|_1 + \alpha_2 \|\mathbf{D}^y_{CSH}(\mathbf{C}_i) \\ & - \mathbf{D}^y_{CSH}(\mathbf{C}_j)\|_2 \big) \end{aligned} \tag{5}$$

where α_1 and α_2 are the weights associated with L_1 and L_2 norms and also β_l and β_s denote the weights of Long and Short SHOT variants, respectively.

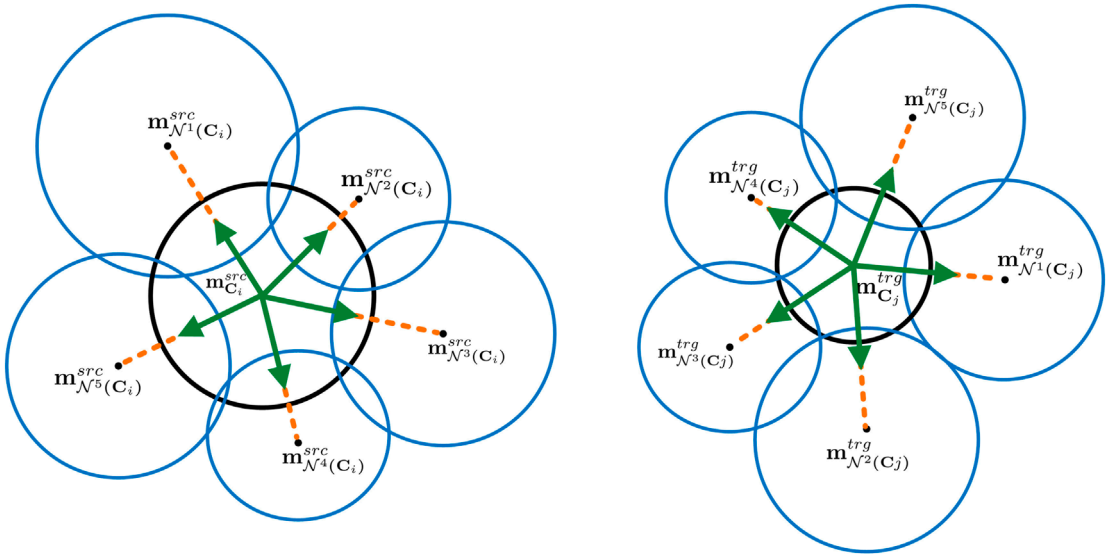

FIGURE 3
Schematic of a neighbourhood topological constraint of two patches (black circles) on source and target.

5.1.2 Neighborhood-topology-preserving SHOT distance

In this metric, the goal is to measure the similarity between two patches in terms of their neighbours' patch SHOT vector. Since even for two corresponding patches, the placements and sizes of adjacent patches are not necessarily similar, we need to evaluate the similarity of those neighbour patches, which are located in similar positions with respect to two query patches. For example, in Figure 3, the similarity measure between $\mathcal{N}^3(\mathbf{C}_i)$ and $\mathcal{N}^1(\mathbf{C}_j)$, and also $\mathcal{N}^5(\mathbf{C}_i)$ and $\mathcal{N}^3(\mathbf{C}_j)$ should contribute prominently in the overall similarity of $\mathbf{C}_i$ and $\mathbf{C}_j$.

First, the pairwise angles between the normalised vectors in $\mathbf{V}_{\mathbf{C}_i}$ and $\mathbf{V}_{\mathbf{C}_j}$ are computed and stored into a matrix $\Theta_\times$. Given a threshold v as the maximum allowed angular difference between these vectors, the matrix is updated by $\Theta \leftarrow v - \Theta$. Here, v depends on the number of generated patches and the partitioning softness. After setting the negative elements of this matrix to zero, Θ is normalized between 0 and 1. Thus, the rate of co-direction of the vectors associated with $\mathcal{N}^m(\mathbf{C}_i)$ and $\mathcal{N}^n(\mathbf{C}_j)$ is indicated by $0 \le \Theta(m,n) \le 1$. Then, the indices and values of the non-zero elements $(m,n,\Theta(m,n))$ of this matrix are used to compute the Neighborhood-Topology-Preserving SHOT distance, d_{NSV}:

$$I = \{(m,n) | \Theta(m,n) \neq 0, m \in \{1{:}\}, n \in \{1{:}\}\} \tag{6}$$

$$d_{NSV}(\mathbf{C}_i, \mathbf{C}_j) = \frac{\sum_{(m,n)\in I} d_{SV}(\mathbf{C}_i, \mathbf{C}_j)\Theta(m,n)}{\sum_{(m,n)\in I} \Theta(m,n)} \tag{7}$$

5.2 Similarity matrix

Using the distance functions defined in the previous section, we then build a similarity matrix, $\mathbf{S}$, storing the pairwise similarity between the target patches and the source patches. As the SHOT vectors are normalized, the maximum possible value of $d_{SV}(.,.)$ and $d_{NSV}(.,.)$ is $\sqrt{2}$. Therefore, the $(i,j)^{th}$ element of $\mathbf{S}$ is defined as a measure of the similarity between $\mathbf{C}_i$ and $\mathbf{C}_j$ as follow:

$$s_{ij} = \sqrt{2} - \frac{\kappa_1 d_{SV}(\mathbf{C}_i, \mathbf{C}_j) + \kappa_2 d_{NSV}(\mathbf{C}_i, \mathbf{C}_j)}{\kappa_1 + \kappa_2} \tag{8}$$

where κ_1 and κ_2 are the weights of the aforementioned distances. Thus: $\mathbf{S} = [s_{ij}]_\times$.

5.3 Robust correspondence selection

Given that the row and the column indices of $\mathbf{S}$ represent the index of the patches on the target and the source, respectively, in theory, the maximum element of a row (as the target patch index) should give the index of the corresponding patch on the source. However, due to the partial overlapping of pointclouds there are some target patches that lack correspondence, and also, such an approach would inevitably produce outliers. To obtain a more robust matching approach, we propose an assignment and rejection strategy that takes advantage of the rigidity terms defined in a reformulation of As-Rigid-As-Possible (ARAP) as the assignment error (Sorkine and Alexa, 2007). Our proposed strategy can be regarded as a discrete combinatorial search that attempts to minimize the piecewise rigidity between the target and the source.

Given two sets of the target and the source centroids as the representative of the soft patches, $\mathbf{M}$ and $\mathbf{M}$, and also n pairwise correspondences, denoted by $[x_i]_{n\times 1} \leftrightarrow [y_i]_{n\times 1}$ and alternatively $\mathbf{X} \leftrightarrow \mathbf{Y}$, representing $\mathbf{C}_{x_i} \leftrightarrow \mathbf{C}_{y_i}$ (i.e., $\mathbf{m}_{x_i} \leftrightarrow \mathbf{m}_{y_i}$), let us reformulate ARAP in this way: We first build a graph by n cells centred at a target centroid, $\mathbf{m}_{x_i}$, and k_A vectors (edges), $[\mathbf{E}_i]_{3\times k_A}$, connecting the centroid to its k_A neighbors using an adjacency list (obtained by a

k-d tree). Using this adjacency list and $\mathbf{m}_{y_i}$ a graph (formed by n cells) is then built on the source yielding $[\mathbf{E}_i]_{3\times k_A}$ (this process is illustrated in Figures 9C, D). We then compute the local rigidity term produced by the transformation of each cell from the target to the source. According to ARAP (Sorkine and Alexa, 2007), the optimal rotation $\mathbf{R}_i$ applied to the ith centroid $\mathbf{m}_i$ and its k_A neighbors $\mathcal{N}(\mathbf{m}_i)$ is obtained by using singular value decomposition (SVD):

$$\begin{aligned}\mathbf{U}_i\mathbf{\Sigma}_i\mathbf{V}_i^* &\leftarrow \mathrm{SVD}\left([\mathbf{E}_i]\,[\mathbf{E}_i]^\intercal\right)\\ \mathbf{R}_i &= \mathbf{V}_i\mathbf{U}_i^\intercal\end{aligned} \tag{9}$$

while enforcing that $\det(\mathbf{R}_i) > 0$ by changing the sign of the column of $\mathbf{U}$ related to the smallest singular value.

Finally, the associated local rigidity error is obtained as:

$$r_i = \sum_{j=1}^{k_A} \left\|\mathbf{E}(:,j) - \mathbf{R}_i\mathbf{E}(:,j)\right\|^2 \tag{10}$$

and the global rigidity error is: $\sum_{i=1}^{n} r_i$.

Thus, the selection can be formulated as a total rigidity minimization problem (implicitly local rigidity) induced by patch correspondence as:

$$\widehat{\mathbf{G}} \leftarrow \min_{X,Y,n} \frac{\mathrm{ARAP}(\mathbf{X},\mathbf{Y})}{n} \tag{11}$$

subject to: $x_i \in \{1{:}\}, y_i \in \{1{:}\}, n \in \{1{:}\}$ and if $i \neq j \rightarrow x_i \neq x_j \wedge y_i \neq y_j$, where $\widehat{\mathbf{G}} \coloneqq [\widehat{\mathbf{X}}\ \widehat{\mathbf{Y}}]_{\hat{n}\times 2}$ denotes the established correspondences. This optimization is a special case of integer programming with an undetermined number of variables, i.e., n.

Due to the constraints and the unknown number of variables, we propose an optimization scheme. First, a batch of potential correspondences is achieved by associating each row of $\mathbf{S}$ to the column with the maximum element of the row. This batch is then filtered by iteratively rejecting the correspondence with the largest rigidity computed by ARAP, recomputing the ARAP for the remained batch and repeating these two steps until all local rigidities fall below a dynamic local rigidity upper-bound u_l. After updating the similarity matrix (considering the established and the rejected correspondences), a new batch of potential correspondences is fed into the above filtering. This process terminates when the global rigidity of the established correspondences exceeds a dynamic total rigidity upper-bound, u_t. These two upper bounds are first estimated in an initialization stage and then updated if no correspondence is established in the filtering stage, by setting u_l to the mean of the local rigidities associated with the currently established correspondences and u_t to the number of target patches times the median of the current established local rigidities.

This scheme is exhibited in Algorithm 1 as the function CorrespondingPatchSelection with three main stages: 1) a bootstrapping stage to initialize and propel the optimization by estimating the two upper-bounds obtained from n_0 initial correspondences, 2) filtering the potential correspondences constrained by the current upper-bounds, 3) updating the two upper-bounds upon no new added correspondence.

Given n_0 as the initial number of correspondences in the bootstrapping stage, $\lceil \frac{n_0}{2} \rceil$ of them are acquired by the $\lceil \frac{n_0}{2} \rceil$ largest values of $\mathbf{S}$ as the most likely correspondences, and the remaining are associated with $n_0 - \lceil \frac{n_0}{2} \rceil$ lowest local rigidities given by ARAP

```
1: function CorrespondingPatchSelection(S)
2: [Δ]_{×1} ≔Max(Diff(Sort(row_1:(S))))
3: u_t ← inf; [r_i] ← ∅;
4: while u_t > Σ_{i=1}^{n̂} r_i do
5: if initialization = True then
6: (G'_a, S, u_t, u_l) ← OptInit S, k_A, n_0
7: S ← SimUpdater S, G'_a, Δ, "dis", "rew"
8: Ĝ ← G'_a;      initialization = False
9: end if
10: G' ← The row-column subscripts of the maximum
    elements of S per non-corresponded rows
11: ([G'_a]_{n'_a×2}, [R_1^{t2s}], [r_i], G'_r) ← CorrFilter G', Ĝ, k_A, u_l
12: S ← SimUpdater S, G'_r, Δ, "pen"
13: if n'_a ≠ 0 then
14: Ĝ ← Ĝ ∪ G'_a
15: S ← SimUpdater S, G'_a, Δ, "dis", "rew"
16: else
17: u_l ← max([r_i])
18: u_t ← Median([r_i])×
19: end if
20: end while
21: return [Ĝ]_{n̂×2} and [R_1^{t2s}]_{3×3×n̂}
22: end function
```

Algorithm 1. Scheme for corresponding patch selection.

function fed by the first batch of potential correspondences (potential corresponding patches). Then the two upper-bounds are initialized as mentioned before. In Algorithm 1 this stage is defined as the function OptInit.

The foundation of our scheme is the potential-correspondence filtering (CorrFilter in Algorithm 1) which iteratively filters the potential input correspondences $[\mathbf{G}']_{(-\hat{n})\times 2}$ achieved as the row-column subscripts associated with the maximum value of the non-associated rows and then as the output, set some as accepted, $\mathbf{G}'_a$, or rejected $\mathbf{G}'_r$ ($\coloneqq \mathbf{G}'\ \mathbf{G}'_a$). In this filtering process, given the already established correspondences, $\widehat{\mathbf{G}}$, discarding the potential correspondence related to the highest local rigidity (obtained from the embedded *ARAP* function with k_A neighbours) continues until no rigidity is higher than u_l.

Using a reward-penalty table, Δ, (acquired as the row-wise maximum value of the differentiated descendingly sorted rows in the original (un-updated) similarity matrix), $\mathbf{S}$ is updated by SimUpdater in Algorithm 1, in three ways denoted by three arguments *dis*, *rew* and *pen* as follows: 1) Disabling: in case of a newly added corresponding patch, the associated rows and columns are disabled by setting all elements to −inf. 2) Rewarding: when a correspondence is established, the elements of $\mathbf{S}$ associated with the co-directed neighbours are incremented by using Δ. 3) Penalizing: the rejected correspondence is penalized by decrementing their associated similarity score using Δ.

After updating $\mathbf{S}$, the newly established correspondences are appended to $\widehat{\mathbf{G}}$, and the adopted mechanism ensures no correct correspondences are falling below the current local upper bound.

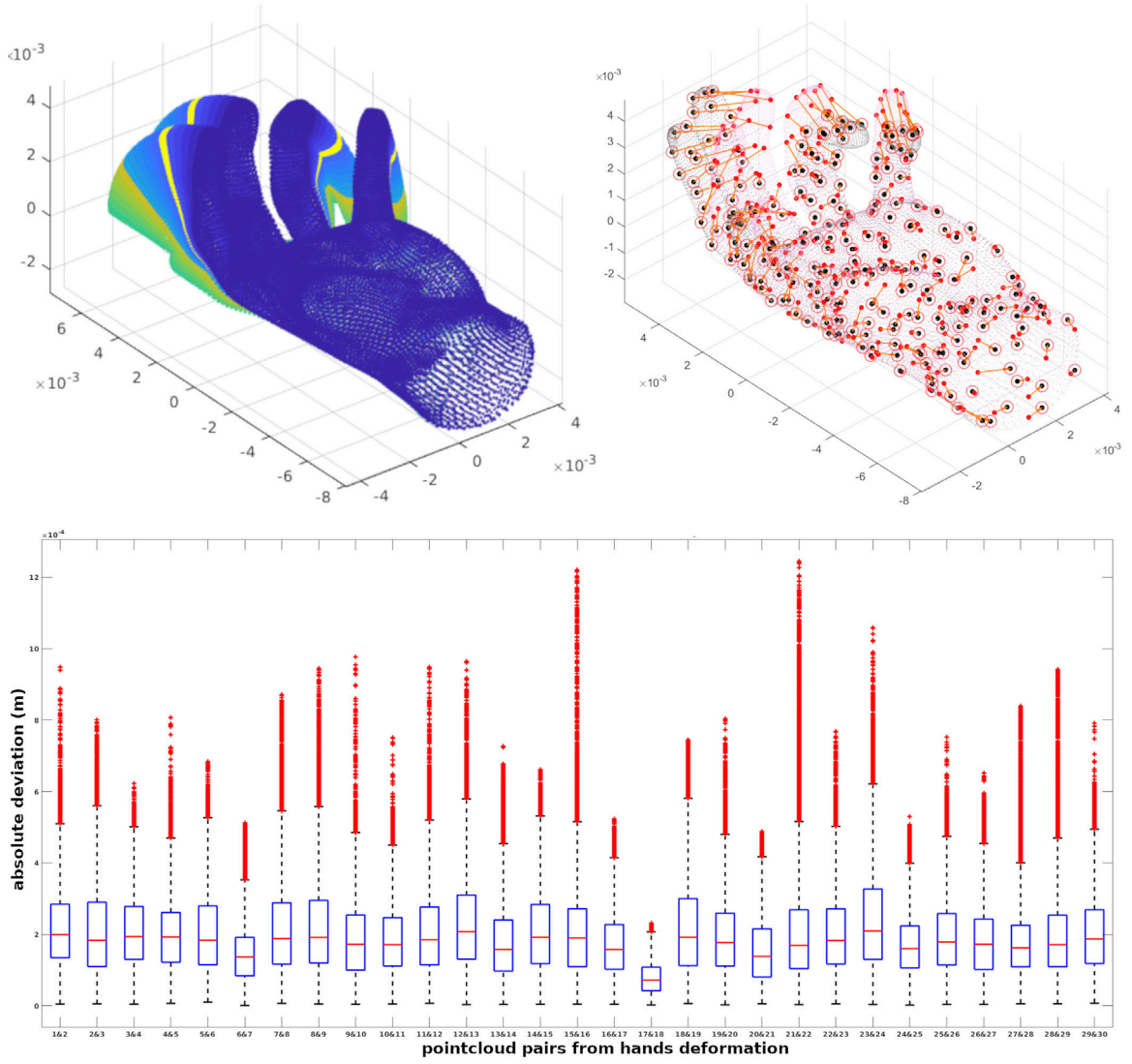

FIGURE 4
A quantitative evaluation is performed on 29 pairs of point clouds achieved from an animated 3D model. We show 30 different deformations of a hand, correspondences of articulation 1 and 7 depicted by centroids and the errors of points in registrations for 29 pairs of point clouds.

Then, upon no accepted correspondence, the two upper bounds are updated as there is a need to keep increasing the upper bounds to allow the correct but largely-deformed patches to be matched.

The optimization process terminates by reaching the global rigidity to the total upper bound.

6 Source warp

6.1 Correspondence-wise transformations

The useful by-product of the proposed optimization scheme (in **Algorithm 1** is the centroid-wise target to source rotation, $\mathbf{R}_i^{t2s}$, per correspondence i, whose inverse, $\mathbf{R}_i^{s2t} = (\mathbf{R}_i^{t2s})^{-1}$, is the rotation required to transform the source centroid to the position of the corresponding target one. Using the position of the matched centroids, the associated translation is:

$$\mathbf{t}_i^{s2t} = \mathbf{m}_{\mathbf{x}_i} - \mathbf{R}_i^{s2t}\mathbf{m}_{\mathbf{y}_i} \tag{12}$$

with which the transformation matrix, $\mathbf{T}_i^{s2t}$, to transform the ith matched source centroid to the corresponding target one is obtained by $\mathbf{T}_i^{s2t} = \begin{bmatrix} \mathbf{R}_i^{s2t} & \mathbf{t}_i^{s2t} \\ 0_{1\times3} & 1 \end{bmatrix}$.

6.2 Transformations propagation

By having these transformations, the points of the source pointcloud in the vicinity of a matched source centroid are impacted proportionally to their distance with respect to the centroid. Given the set of all matched source centroids, $\mathbf{M}(\hat{\mathbf{Y}}) = \{\mathbf{m}_{\hat{y}_i} | \hat{y}_i \in \hat{\mathbf{Y}}\}$, the indices of k_p matched source centroids adjacent to a query source point $\mathbf{p}_i$ are shown by $\mathbf{A}$ $(\mathbf{p}_i)$. So the k_p nearest centroids to point $\mathbf{p}_i$

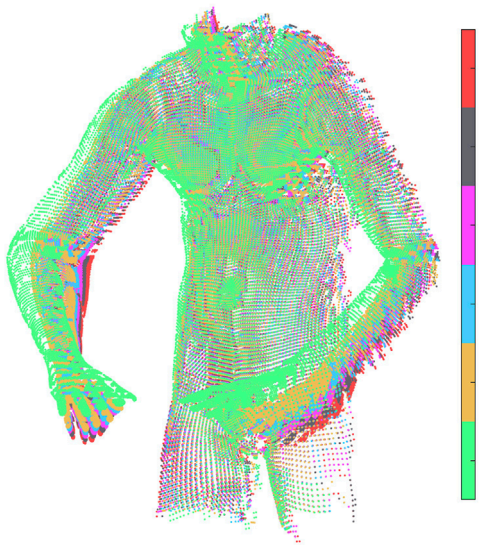

FIGURE 5
Six colour-coded animation frames of the human dataset captured by raytracing the associated original point clouds.

are:

$$\mathbf{M}(\mathbf{A}(\mathbf{p}_i)) = \left\{ \mathbf{m}_{a_j} | a_j \in \mathbf{A}(\mathbf{p}) \subset \mathbf{Y}, \quad j = 1{:}k_p \right\} \tag{13}$$

where, $\mathbf{M}_{\mathbf{A}(\mathbf{p}_i)} \subset \mathbf{M}_{\tilde{\mathbf{Y}}} \subset \mathbf{M}$. As each point of the pointcloud may have several already matched centroids in its vicinity, it should be transformed based on the weighted interpolation of all those adjacent transformations. The weight, w, reflects the amount of influence is received from the adjacent matched centroids. According to Eberly (2017), the interpolation (loosely speaking the average) function for two transformations in **SE(3)** is a curve that follows the shortest geodesic path between two transformations. By extending their work to a weighted multi-transformation averaging, the transformation imposed on $\mathbf{p}_i$ is computed by, $\bar{\mathbf{T}}_{\mathbf{p}_i} = \text{Ave}_{SE3}(\mathcal{T}_{\mathbf{p}_i}, \mathbf{W}_{\mathbf{p}_i})$, where $\mathcal{T}_{\mathbf{p}_i} = (\mathbf{T}_{a_j}^{s2t} | j = 1{:}k_p)$ and $\mathbf{W}_{\mathbf{p}_i} = (w_{a_j} | j = 1{:}k_p)$. For the weights, we opted for Embedded Deformation weighting scheme (Sumner et al., 2007), where the weights of the K adjacent centroids to the point i, are computed by: $w_{a_j} = 1 - \left\| \mathbf{p}_i - \mathbf{m}_{a_j} \right\|_2^2 / d_{\max}$ where $d_{\max}$ here is the distance to the $k_p + 1$ nearest centroid and then $w_{a_j} \leftarrow \frac{w_{a_j}}{\sum_{i=1}^{k_p} w_{a_i}}$. Then, given $\bar{\mathbf{T}}_{\mathbf{p}_i} = \begin{bmatrix} \bar{\mathbf{R}}_{\mathbf{p}_i} & \bar{\mathbf{t}}_{\mathbf{p}_i} \\ 0_{1\times3} & 1 \end{bmatrix}$, the new position of source point i is computed as $\tilde{\mathbf{p}}_i = \bar{\mathbf{R}}_{\mathbf{p}_i} \mathbf{p}_i + \bar{\mathbf{t}}_{\mathbf{p}_i}$ and the warped source to register on the target is $\tilde{\mathcal{P}} = \bigcup_{i=1} \tilde{\mathbf{p}}_i$.

7 Experimental results

In this section, we present the result of our experiments on two types of simulated and real datasets. To robustify the association, we used two concatenated SHOT descriptors (yielding a 704-D feature vector) per point, with big and small radii to account for the small and the large geometrical features. For the experiments in this section, the two SHOT radii, and n_0 are set to 0.02 m, 0.05 m, and 10, respectively. Also, we set both k_A and k_p to 10.

7.1 Simulated-data

The quantitative evaluation of the framework is performed on an animated 3D model whose points are associated with a specific identifier code. As a result, each point representing a specific part of the surface shares the same index The frame-to-frame animation of the 3D model, shown in **Figure 4**, is performed with bounded biharmonic weights (Jacobson et al., 2011), which produces a smooth deformation. **Figure 4A** shows 30 different deformations of a hand generated through this method. By running

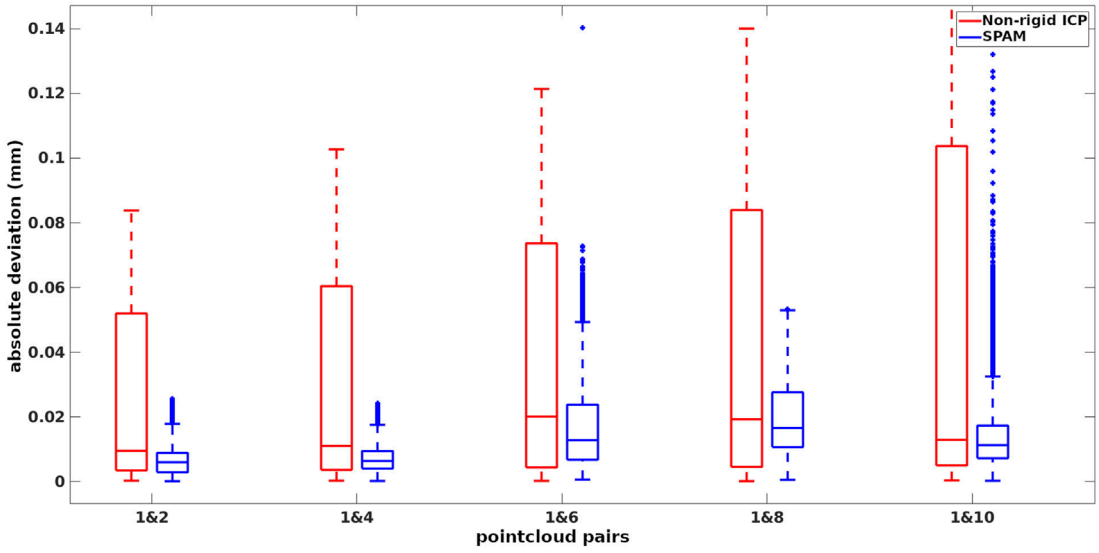

FIGURE 6
Deviations from ground-truth in 5 experiments from non-rigid ICP and SPaM on the selected pairs of deformation.

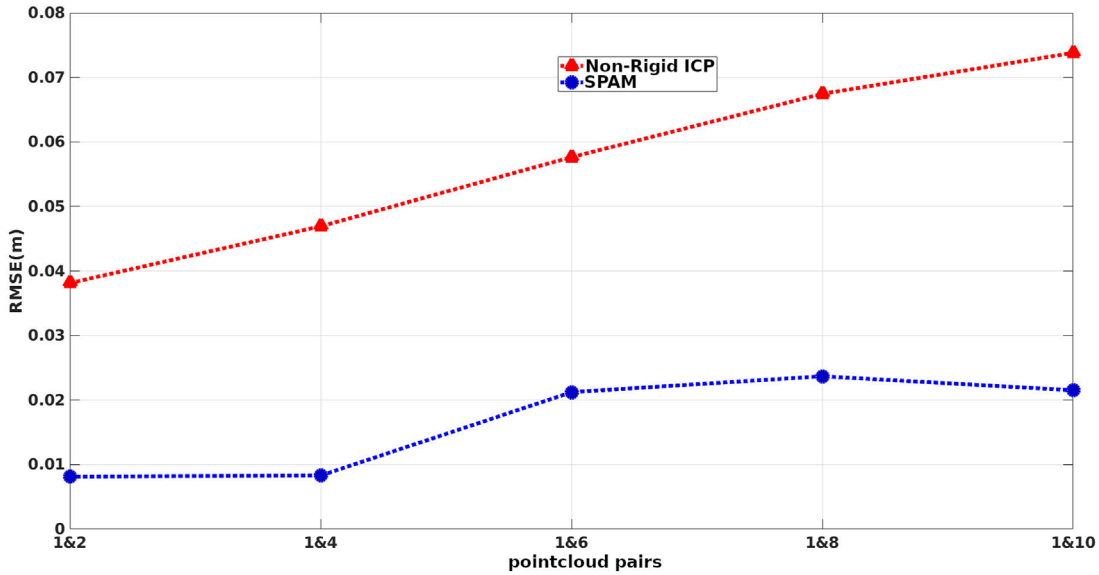

FIGURE 7
RMSE of distances between corresponding points of warped source and target on 5 sets of deformation.

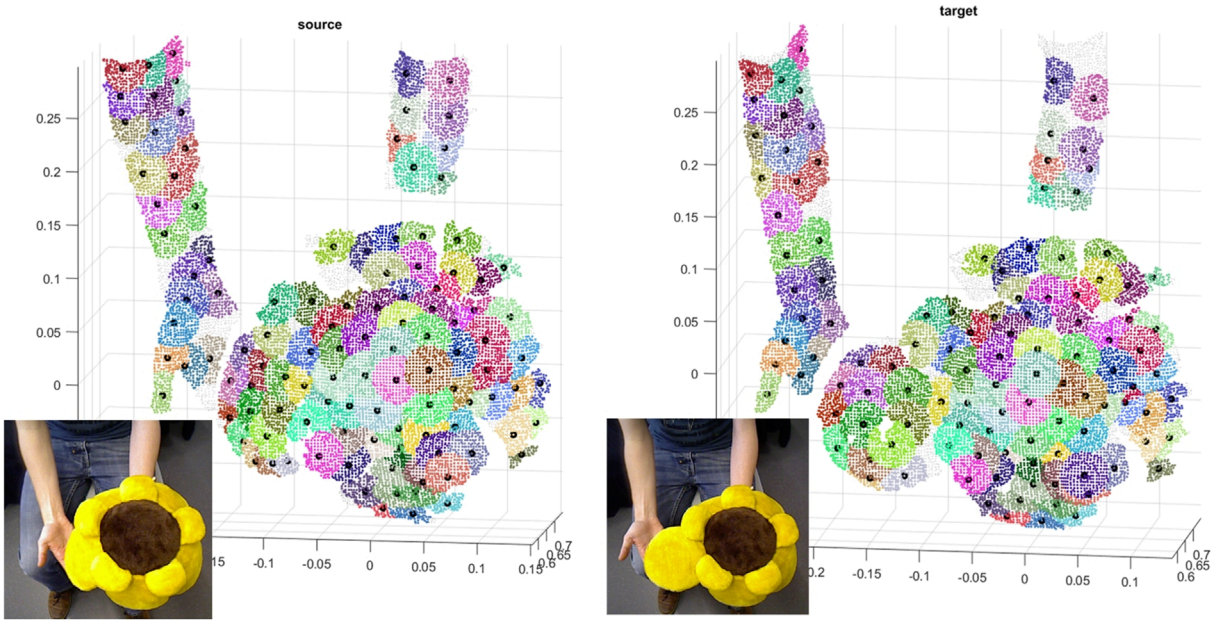

FIGURE 8
Color-coded patch-wise correspondences of soft patches on the source and the target including their centroids on Sunflower dataset.

our pipeline for 23894 points of this dataset, and choosing an average number of 100 points to be included in the partitions, in total 266 soft partitions for both source and target were generated. To show the robustness of the optimization, the detected correspondences of articulation 1, and articulation 7 of this dataset are depicted by centroids as representatives of patches in Figure 4B.

By warping all points of articulation i (source) to $i+1$ (target) using the correspondences, there would be some deviation with respect to the actual position of the target points. The box plots associated with the errors of points in registrations for 29 pairs of point clouds are shown in Figure 4C. In these 29 experiments, the average number of the established correspondences is 210 out of 280 soft patches achieved in 100 iterations on average.

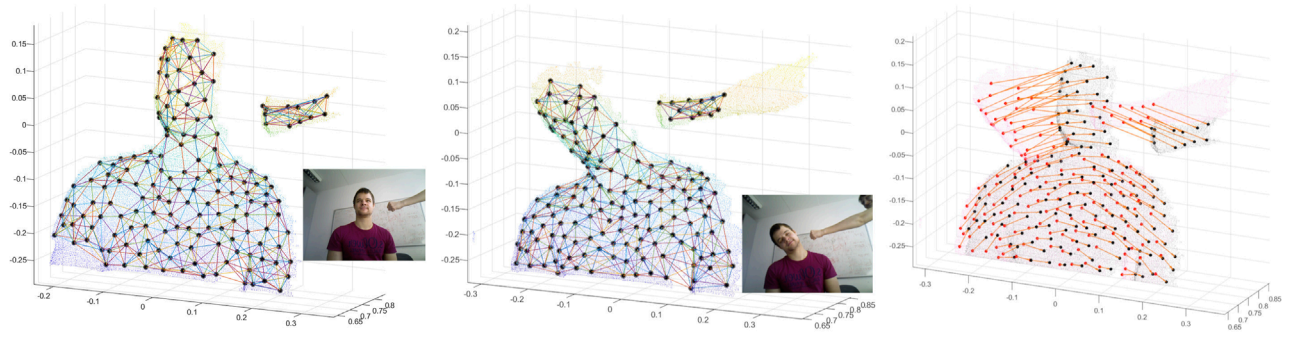

FIGURE 9
Registering two point clouds of the Boxing dataset, 40 frames apart.

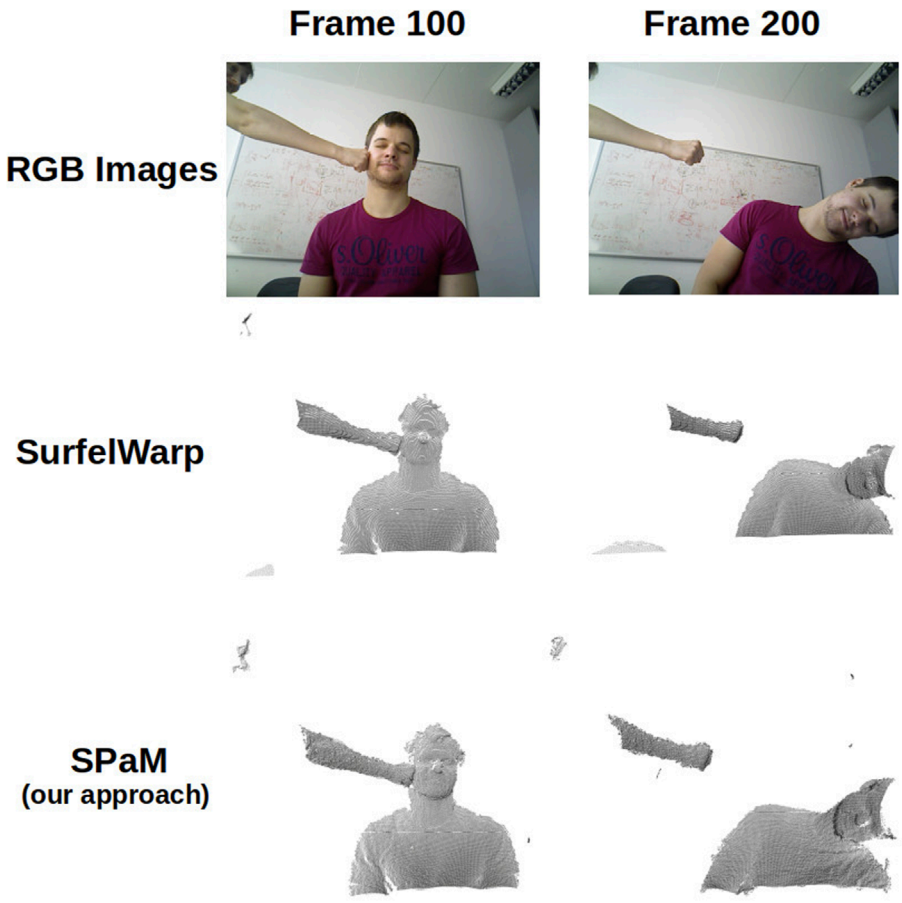

FIGURE 10
Qualitative comparison of our method (SPaM) compared to SurfelWrap (Gao and Tedrake, 2019) via the results of registration and reconstruction up to frames 100 and 200.

In another experiment with ground truth, we quantitatively compared the performance of our method against Zampogiannis et al. (2019), which is available as part of the Cilantro library (Zampogiannis et al., 2018). In this experiment, the goal is to create similar conditions as the capturing process with a single solid-state LIDAR camera (e.g., using a Realsense l515) in which the output point clouds are partially overlapping during deformation. For this experiment, we used the Deforming Human dataset with ground truth [generated again with bounded biharmonic weights (Jacobson et al., 2011)] in which each point of the surface has the same index in different animation frames. Simulating a fixed depth sensor by applying a raytracer algorithm (Skinner et al., 2014), the

point clouds with different deformation were placed in front of the sensor iteratively, and the result of raytraced frames 1, 2, 4, 6, 8, and 10 are displayed in **Figure 5** with the related colour-map for the frames. To make this experiment more realistic, sensor noise was simulated by applying zero-mean Gaussian noise with a standard deviation of 2 mm in the direction of the surface normal for each point.

During this process, the original indices (from the original articulation) of the raytraced points are saved. To evaluate the performance of methods in terms of handling different amounts of deformations, we paired the first (as the target) point cloud with the rest (as the source), giving 5 pairs of point clouds to be registered. Feeding these pairs into our pipeline and non-rigid ICP, we first found the corresponding points of the target and source (which share the same original indices) and then measured the absolute distance of those corresponding points in the warped source to the target. The box plot of deviations from ground truth associated with our pipeline (SPaM) and non-rigid ICP is illustrated in **Figure 6**, where the horizontal axis shows the experimental pairs, and the vertical axis represents the deviation from the ground truth in meters. In the box plots, the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively; the central mark shows the median; and the outliers are represented by "+" symbol. The Root Mean Square Error (RMSE) acquired from these experiments are displayed in **Figure 7**.

7.2 Real-data

We use the VolumeDeform dataset (Innmann et al., 2016), which contains a variety of deformable objects, to evaluate the performance of our method qualitatively. Our framework is capable of registering scans taken with a significant time difference, indicating the robustness of our optimization scheme for patch correspondence against large non-rigid local and global deformations. To demonstrate this capability, we used two pointclouds of the Sunflower dataset 30 frames apart. The scans are down-sampled and partitioned into soft patches (with 200 points on average), and then 95 corresponding patches were established (out of 125 and 139 soft patches of target and source). The soft partition concept is visualized in **Figure 8**.

We deployed our pipeline on boxing dataset as well; **Figures 9A, B** show the graph generated by the corresponding centroids and edges to 8 neighbour centroids (or rather patch centroids) for target and source. **Figure 9D** shows the target deformation onto the source by applying the average transformations of adjacent centroids to the points.

Although the focus of our proposed method is an offline CPU-based non-rigid registration (not reconstruction or fusion), we qualitatively compare the performance of SPaM on registering the consecutive frames with Surfelwarp (Gao and Tedrake, 2019). For this purpose, we use a simple iterative forward registration and fusion scheme. The current frame is regarded as the source and the merged model of all previous frames as targets (similar to a canonical frame). The objective in each iteration is, then, to register and warp the canonical frame towards the new frame. By using the boxing dataset, we merged the warped reconstructed model iteratively with the new scan. As there is not much deformation at the beginning of this dataset, we used frames from 40 to 200, and the result of reconstruction up to frames 100 and 200 acquired from two methods are compared in **Figure 10**, which shows the acceptable fidelity and alignment of the reconstructed model by SPaM to the current frame. It is worth mentioning that the average time for each CPU-based registration of this dataset is 120 s.

With the same conditions as above, we experimented on 170 consecutive scans of the minion dataset, and the results of three registrations are shown in **Figure 1**. The second column is the current frame, the third column is the reconstructed model, and the fourth column is the registration of the mentioned two pointclouds.

8 Conclusion

We have devised a framework (SPaM) to establish the correspondences of two non-rigidly deformed point clouds by using soft patches and an aggregation of locally rigid transformations. Our framework is evaluated on a challenging VolumeDeform dataset as capable of registering scans taken with a large time difference, indicating the robustness of our optimization scheme for patch correspondence against heavily non-rigid local and global deformations. However, on curved surfaces, the Euclidean distance is a suboptimal choice for propagating patch transformations to point elements. Re-organizing and optimizing our framework upon geodesic distance is a feasible option which we plan as an extension of our work. Also, by enhancing the optimization method, we plan to present this pipeline as a real-time approach.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://spamregistration.github.io/spamdataset/.

Author contributions

BM: Original author who designed the proposed algorithm and set of experiments. He was also responsible for writing **Sections 1**–**5** of the paper. RF: Theoretical contributions to **Section 5**, generated some of the datasets for the experiments, and helped with writing of **Sections 4**–**6** of the paper. TV-C and AA: Supervised the project and provided guidance, feedback on every aspect of the project and helped with the writing **Sections 1**, **2**, **8** of the paper. All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This paper is supported by funding from Meat and Livestock Australia (MLA) grant number B.GBP.0051. This work was possible due to the financial and in kid support and efforts of many individuals from NSW Department of Primary Industries,

University of Technology Sydney, Local Land Services, and Meat and Livestock Australia.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. robotics* 32, 1309–1332. doi:10.1109/tro.2016.2624754

Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S. R., Kowdle, A., et al. (2016). Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph. (TOG)* 35, 1–13. doi:10.1145/2897824.2925969

Eberly, D. (2017). "Interpolation of rigid motions in 3d," in *Geometric tools* (Redmond WA 98052), 1–10.

Gao, W., and Tedrake, R. (2019). *Surfelwarp: Efficient non-volumetric single view dynamic reconstruction. arXiv preprint arXiv:1904.13073.*

Gupta, T., Shin, D., Sivagnanadasan, N., and Hoiem, D. (2016). *3dfs: Deformable dense depth fusion and segmentation for object reconstruction from a handheld camera. arXiv preprint arXiv:1606.05002.*

Innmann, M., Zollhöfer, M., Nießner, M., Theobalt, C., and Stamminger, M. (2016). "Volumedeform: Real-time volumetric non-rigid reconstruction," in *European conference on computer vision* (Springer), 362–379.

Jacobson, A., Baran, I., Popovic, J., and Sorkine, O. (2011). Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 1–8. doi:10.1145/2010324.1964973

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. *Proc. fourth Eurogr. symposium Geometry Process.* 7, 1–10.

Newcombe, R. A., Fox, D., and Seitz, S. M. (2015). "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 343–352.

Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., and Guibas, L. (2012). Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph. (TOG)* 31, 1–11. doi:10.1145/2185520.2185526

Park, H.-S., and Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert Syst. Appl.* 36, 3336–3341. doi:10.1016/j.eswa.2008.01.039

Petit, A., Lippiello, V., Fontanelli, G. A., and Siciliano, B. (2017). Tracking elastic deformable objects with an rgb-d sensor for a pizza chef robot. *Robotics Aut. Syst.* 88, 187–201. doi:10.1016/j.robot.2016.08.023

Salti, S., Tombari, F., and Di Stefano, L. (2014). Shot: Unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst.* 125, 251–264. doi:10.1016/j.cviu.2014.04.011

Seib, V., and Paulus, D. (2018). "A low-dimensional feature transform for keypoint matching and classification of point clouds without normal computation," in *2018 25th IEEE international conference on image processing (ICIP)* (IEEE), 2949–2953.

Skinner, B., Vidal-Calleja, T., Miro, J. V., De Bruijn, F., and Falque, R. (2014). "3d point cloud upsampling for accurate reconstruction of dense 2.5 d thickness maps," in *Australasian conference on Robotics and automation* (ACRA).

Slavcheva, M., Baust, M., Cremers, D., and Ilic, S. (2017). "Killingfusion: Non-rigid 3d reconstruction without correspondences," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1386–1395.

Sorkine, O., and Alexa, M. (2007). As-rigid-as-possible surface modeling. *Symposium Geometry Process.* 4, 109–116.

Sumner, R. W., Schmid, J., and Pauly, M. (2007). Embedded deformation for shape manipulation. *ACM Trans. Graph. (TOG)* 26, 80. doi:10.1145/1276377.1276478

Yu, T., Guo, K., Xu, F., Dong, Y., Su, Z., Zhao, J., et al. (2017). "Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera," in *Proceedings of the IEEE international conference on computer vision*, 910–919.

Yu, T., Zheng, Z., Guo, K., Zhao, J., Dai, Q., Li, H., et al. (2018). "Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7287–7296.

Zampogiannis, K., Fermuller, C., and Aloimonos, Y. (2018). "Cilantro: A lean, versatile, and efficient library for point cloud data processing," in *Proceedings of the 26th ACM international conference on Multimedia*, 1364–1367.

Zampogiannis, K., Fermuller, C., and Aloimonos, Y. (2019). "Topology-aware non-rigid point cloud registration," in *IEEE transactions on pattern analysis and machine intelligence.*

Zheng, Z., Yu, T., Li, H., Guo, K., Dai, Q., Fang, L., et al. (2018). "Hybridfusion: Real-time performance capture using a single depth sensor and sparse imus," in *Computer Vision–ECCV 2018. ECCV 2018. Lecture Notes in Computer Science* (Springer, Cham) 11213, 384–400. doi:10.1007/978-3-030-01240-3_24