Non-rigid Registration for Deformable Objects

Van-Toan Cao, Van Tung Nguyen, Trung-Thien Tran, Sarah Ali and Denis Laurendeau
Computer Vision and Systems Laboratory, Department of Electrical and Computer Engineering, Laval University,
1065 Avenue de la Médecine, G1V 0A6, Québec, Canada

Keywords: Registration, Deformation, Correspondences, Optimization.

Abstract: We present an efficient algorithm for non-rigid registration of two partially overlapping 3D surfaces in which a target surface is a deformed instance of a source surface. The algorithm is implemented in two main phases. In the first phase, the robust algorithm that is used is based on a probability density estimation to find reliable correspondences between the two surfaces. Then, in the second phase, a deformation algorithm is applied for non-rigid registration where the displacement of each point is described by an affine transformation in relation with other points of the same surface and its corresponding point on the other surface. Combined with initial correspondences in the first phase, an effective strategy for optimization of a cost function is carried out to align the two surfaces without using any assumption and user-intervention on the algorithm. We test the robustness of our method by efficiently aligning pairs of surfaces of realistic scan data of human body models.

1 INTRODUCTION

Registration of two surfaces is a well-known problem and a key task in computer graphics where two surfaces are partial digital representations of the same object but are taken at different times or/and different locations. To align such surfaces into a common coordinate system, parameters must be computed for the movement of the source surface onto the target surface. For cases where the object is a rigid body, the problem of general registration reduces to a rigid registration problem and the movement between two surfaces is described globally by a rotation matrix, a translation vector (or/and a scaling coefficient). Many algorithms have been proposed to address rigid registration and it is now considered to be almost solved. Thus attention is shifting to dealing with non-rigid registration where the object undergoes deformations when being scanned and the movement between two surfaces cannot be described by a single rigid transformation.

Many algorithms are proposed to find the movement in a non-rigid registration paradigm with different strategies and assumptions. A common strategy of many of these algorithms is to associate a transformation for each respective point on the source surface and move each point to a corresponding point on the target surface while imposing coherence constraints with other points. For this step, the optimization of these methods uses some assumptions to obtain the final alignment.

In this paper, we present an algorithm to align two partially overlapping surfaces (in our case, they are two meshes but the framework of the algorithm can be applied to other surface representations) for addressing the challenges of non-rigid registration. Generally, each point of the source surface will have its own transformation to move it correctly to its corresponding point on the target surface. To achieve that, the optimization of the cost function requires that stable correspondences be found and the correct transformations to align two surfaces be estimated. Since convergence to a wrong minimum of the cost function is not desired, some methods put user-defined makers on the two surfaces to guide the optimization process. Other approaches assume the movement between the two surfaces is small, or that poses of the two surfaces are the same. However, our algorithm does not use any assumption for the optimization and automatically performs without any user-intervention. The algorithm is implemented in two phases. In the first phase, a robust method based on probability density estimation provides a set of candidates as correspondences between the two surfaces. In the second phase, these candidates are used as initial correspondences and combined with an effective optimization strategy to guide the algorithm to converge to a minimum.

Cao V., Nguyen V., Tran T., Ali S. and Laurendeau D..
Non-rigid Registration for Deformable Objects.
DOI: 10.5220/0004683100430052
In *Proceedings of the 9th International Conference on Computer Graphics Theory and Applications* (GRAPP-2014), pages 43-52
ISBN: 978-989-758-002-4

2 RELATED WORK

The goal of registration algorithms is to align two surfaces together to obtain a better representation of the object. If the object is rigid, one global transformation is sufficient to describe the movement of the source surface and to align it to the target surfaces. The Iterative Closest Point algorithm (ICP)(Besl and McKay, 1992; Chen and Medioni, 1991) and its variants (Rusinkiewicz and Levoy, 2001) are prominent among rigid registration algorithms due to their simplicity and robustness. ICP iteratively searches for correspondences based on a closest distance criterion for finding a rigid transformation until it reaches a local minimum. A limitation of ICP is that it is only effective when a good initial registration between the two surfaces is available or when the movement between the two surfaces is small.

Based on the ICP, a variety of techniques have been adapted to the problem of non-rigid registration (Brown and Rusinkiewicz, 2007; Allen et al., 2003; Amberg et al., 2007; Pauly et al., 2005; Huang et al., 2008). However, like ICP, these algorithms need to use prior knowledge to avoid converging to a wrong local minimum of the cost function. In (Allen et al., 2003; Amberg et al., 2007; Allen et al., 2002) a template model, which is the same pose as the scanned surface, provides a strong geometric prior for reconstructing high-quality models with automated hole-filling and noise removal. Moreover, user-defined markers placed on the template model and scanned surface are exploited to prevent the optimization algorithm to set stuck in a local minimum.

Some algorithms are proposed to take advantage of real-time 3D scanners. They do not use any template or user-defined markers for alignment. In (Mitra et al., 2007; Submuth et al., 2008), registration is conducted without computing correspondences explicitly. These algorithms aggregate all scans into a 4D space-time surface and estimate inter-frame motion from kinematic properties of this surface. In (Wand et al., 2009), a hierarchical registration approach is applied where the algorithm aligns every two frames, merges them, then aligns every two pairs of frames, merges them, and so on until it processes the entire sequence. These algorithms fail when the input data is noisy and the deformation between two surfaces is large.

Recent algorithms attempt to remove the dependency on template models or artificial markers. In (Huang et al., 2008), a set of geodesically consistent correspondences are extracted and the algorithm solves for transformations at many sampled locations on the surface and then interpolates the remaining surfaces using prescribed influence weights. This method can solve for large deformation where the assumption of geodesic consistency between two surfaces is warranted. Chang et al. (Chang and Zwicker, 2008) determine the correspondences by spin image (Johnson, 1997) and find transformation for each pair of matches. Based on non-linear mean-shift framework, they then cluster transformations between two surfaces into many groups before applying a graph-cut technique to associate each transformation for each point on the two surfaces.With this algorithm, some transformations do not exist in the motion sampling process, this causes some parts on the source surface to lack a precise transformation to align correctly with the respective parts on the target surface. Li et al. (Li et al., 2008) applied an embedded deformation algorithm to solve for a reduced set of transformations. During optimization of the cost function, their algorithm exploits a 2D parameterization of the range scans to search for closest points on the target scan as corresponding points and guide the optimization to converge to a local minimum. Therefore, their algorithm is limited to 3D scans. Moreover, if the movement between two scans is large, this algorithm can be trapped in a wrong local minimum. Our algorithm does not need a parameterization of the range scan and can deal with other surface representations. Moreover, we use initial correspondences as an optimization constraint to achieve a correct alignment for large deformations.

In (Anguelov et al., 2004), the method finds a good correspondence assignment with a given template shape by formulating a Markov Random Field optimization that best fits the observed data and preserves the shape of the template. The method is able to recover significant movements of articulated parts and non-rigid deformation. This method assumes that preservation of geodesic distance is warranted and one of the input shapes is a subset of the others. Myronenko et al. (Myronenko and Song, 2010) consider the alignment of two point sets as a probability density estimation problem. Their algorithm fits Gaussian Mixture Model (GMM) centroids (source point set) to the data (target point set) by maximizing the likelihood. The GMM centroids are moved coherently as a group to preserve topological structure of the point sets. Experimental results show that this method outperforms another well-known method (Chui and Rangarajan, 2003). In a similar work, algorithm of Jian et al. (Jian and Vemuri, 2011) minimizes a statistical discrepancy measure between two mixture models in which each model is a Gaussian mixture model and presents an input point set. However, a bottleneck of these three methods is that they require significant computation as well as large memory space to describe the relationship between each point of the first point set and each point of the second point set at each iteration of the optimization process (for more detail, we refer the reader to (Myronenko and Song, 2010)). Therefore, this approach is mostly applied to 2D data or small 3D data sets.

3 PROPOSED METHOD

3.1 Deformation Model

Because we need to move each point on the source surface to its corresponding point on the target surface, computation becomes too cumbersome to search for a transformation for each point of the source surface. A deformation model is necessary to reduce the computation time. The chosen model also has to be general enough to apply to any object and express complex deformations. Given these goals, we adopted the deformation model proposed by Sumner et al. (Sumner et al., 2007) for our algorithm. For this model, a space deformation is defined by a collection of affine transformations. The graph of the model consists of nodes and undirected edges. While nodes are chosen by uniformly sampling the source surface and one transformation is associated to each node, undirected edges connect nodes of overlapping influence to indicate local dependencies. Given the node position $\mathbf{g}_j \in \mathbb{R}^3, j \in 1 \dots m$, the affine transformation for this node is specified by a 3x3 matrix $\mathbf{R}_j$ and a 3x1 translation vector $\mathbf{t}_j$. The influence of the transformation is centered at the node's position so that it maps any point $\mathbf{p}$ in $\mathbb{R}^3$ to position $\widetilde{\mathbf{p}}$ according to:

$$\widetilde{\mathbf{p}} = \mathbf{R}_j(\mathbf{p} - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j \tag{1}$$

The influence of individual nodes is smoothly blended so that the deformed position $\widetilde{\mathbf{v}}_i$ of each point on the source surface $\mathbf{v}_i$ is a weighted sum of its position after application of the deformation graph affine transformations.

$$\widetilde{\mathbf{v}}_i = \sum_{j=1}^{m} w_j(\mathbf{v}_i)\left[\mathbf{R}_j(\mathbf{v}_i - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j\right] \tag{2}$$

The weights for each point are precomputed according to:

$$w_j(\mathbf{v}_i) = (1 - \|\mathbf{v}_i - \mathbf{g}_j\| / d_{\max})^2 \tag{3}$$

and then normalized to sum to one. Here, $d_{\max}$ is the distance to the $k+1$ nearest node. Once the deformation graph has been specified, the optimization will treat per-node affine transformations as unknowns for deformation. Constraints are needed so that the deformation is natural and preserves topology of the source surface. The first energy term, E_{rot}, penalizes deviation of each transformation from a pure rigid motion:

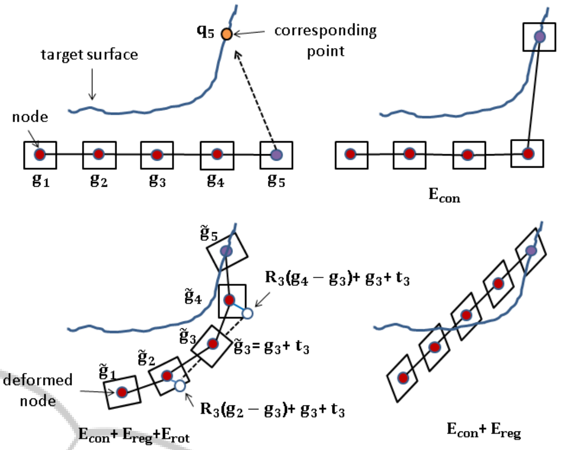


Figure 1: Effect of the three energy terms in a simple deformation graph. The quadrilaterals at each node illustrate the deformation induced by the respective affine transformation.

$$\mathrm{E}_{\text{rot}} = \sum_{j=1}^{m} \mathrm{Rot}(\mathbf{R}_j) \tag{4}$$

where

$$\begin{aligned}\mathrm{Rot}(\mathbf{R}) =& (\mathbf{c}_1 \cdot \mathbf{c}_2)^2 + (\mathbf{c}_1 \cdot \mathbf{c}_3)^2 + (\mathbf{c}_2 \cdot \mathbf{c}_3)^2 + \\ & (\mathbf{c}_1 \cdot \mathbf{c}_1 - 1)^2 + (\mathbf{c}_2 \cdot \mathbf{c}_2 - 1)^2 + (\mathbf{c}_3 \cdot \mathbf{c}_3 - 1)^2\end{aligned} \tag{5}$$

and $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ are column vectors of $\mathbf{R}_j$.

The second energy term, E_{reg}, serves as a regularizer for the deformation by indicating that the affine transformations of adjacent graph nodes should agree with one another:

$$\mathrm{E}_{\text{reg}} = \sum_{j=1}^{m} \sum_{k \in \mathcal{N}(j)} \left\| \mathbf{R}_j(\mathbf{g}_k - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j - (\mathbf{g}_k + \mathbf{t}_k) \right\|_2^2 \tag{6}$$

where $\mathcal{N}(j)$ is a set of neighbor nodes of node j.

The third energy term, E_{con}, penalizes the deviation of each deformed node and its corresponding point:

$$\mathrm{E}_{\text{con}} = \sum_{j=1}^{m} \left\| \widetilde{\mathbf{g}}_j - \mathbf{q}_j \right\|_2^2 \tag{7}$$

Here, $\widetilde{\mathbf{g}}_j$ is deformed position of node j. A description for deformation model (Sumner et al., 2007) is shown in Figure. 1.

In our case, $\mathbf{q}_j$ is the corresponding point of $\mathbf{g}_j$, not defined by the user. In the best case, we hope that each node has its corresponding point to guide the deformation leading to the target surface perfectly. However, due to partial overlap between the two surfaces, some nodes on the source surface will not have a corresponding point on the target surface.

Let's define a cost function:

$$E = w_{\mathrm{rot}}\mathrm{E}_{\mathrm{rot}} + w_{\mathrm{reg}}\mathrm{E}_{\mathrm{reg}} + w_{\mathrm{con}}\mathrm{E}_{\mathrm{con}} \quad (8)$$

where $w_{\mathrm{rot}}, w_{\mathrm{reg}}, w_{\mathrm{con}}$ refer to the contribution weight of each energy term.

Given m nodes in the deformation graph, the optimization for the cost function E has to find optimal values of $15m$ unknowns. Here, each node has 12 unknowns for its associated affine transformation and 3 unknowns for the 3D coordinates of its corresponding point. All unknowns are stacked into a column vector $\mathbf{z}$ and the optimal value of $\mathbf{z}$ will be obtained by our strategy when combined with the Levenberg-Marquardt algorithm (Madsen et al., 2004). The optimization is separated into two steps, called movement optimization and alignment optimization.

3.2 Finding Reliable Correspondences

A corresponding point $\mathbf{q}_j$ must be determined for each node to apply Eq. 7 to start the optimization problem (Eq. 8). Since we do not exploit user-defined markers, an automatic method is thus desired to obtain correspondences. For rigid registration, 3D shape descriptors (Johnson, 1997; Frome et al., 2004; Mian et al., 2004) can provide correspondences. However, it becomes difficult to apply these descriptors for non-rigid registration because changes between the two surfaces are not only due to a global movement but also to local deformations such as bulging, stretching, bending or flapping.

Instead of using geometry-based descriptors to find correspondences, we use a probability-based method for this purpose. The Coherent Point Drift (CPD) algorithm proposed by Myronenko et al. (Myronenko and Song, 2010) satisfies this requirement. Their algorithm uses a probability approach to align two point sets robustly. Moreover, it also provides the correspondences between the two point sets. The algorithm defines the correspondence probability between a point $\mathbf{y}_i$ (centroid) on the source point set and a point $\mathbf{x}_j$ (data point) on the target point set as the posterior probability of the GMM centroid given the data point $p(\mathbf{y}_i \mid \mathbf{x}_j) = p(\mathbf{y}_i)p(\mathbf{x}_j \mid \mathbf{y}_i)/p(\mathbf{x}_j)$. Because the number of points between the two sets are different, we cannot achieve one-to-one correspondences and thus one point in the source set can have more than one corresponding point in the target set. Given nodes of the deformation graph, we need to inspect which points are actual correspondences for these nodes.

As mentioned above, the CPD approach is not efficient when applied to large data sets due to its computation burden. So, given the data in the source surface $\mathbf{V}$, uniform sampling is used to reduce the size of the data. Then, the reduced data set is concatenated with the data of the deformation graph $\mathbf{G}$ to create another source data set $\mathbf{Y}$. This process is also applied to the target surface $\mathbf{Q}$ to create another target data set $\mathbf{X}$. Then, the correspondences between the new data sets $\mathbf{Y}$, $\mathbf{X}$ are computed with the CPD. Two additional steps are used for evaluation of the correspondences.

First, for each pair of correspondences, the Euclidean distance between deformed point $\tilde{\mathbf{y}}_i$ and its corresponding point $\mathbf{x}_j$ is computed. These distances are then used to build a distance distribution. In this distribution, there are two cases for which some distance values differ significantly. In the first case, because the CPD can give incorrect alignment of some patches between the two data sets, some correspondences belonging to these patches will be erroneous and thus result in large distances. In the second case, because the two data sets overlap partially, some patches of the source data do not exist on the target data, and vice-versa. However, the CPD always provides a corresponding point for each point on these patches; the correspondences are also erroneous and result in large distances. In both cases, these large distances are treated as outliers in the distance distribution. To detect these outliers, a standard statistical procedure is to determine the *"fourth spread"* of the distance distribution (DeVore, 2008).

$$f_s = m_u - m_l \quad (9)$$

where m_u is median of largest L/2 measurements, m_l is median of smallest L/2 measurements, L is number of distance values.

Statistically moderate outliers are $1.5f_s$ units above (below) the upper (lower) fourth, and extreme outliers are $3f_s$ units above (below) the upper (lower) fourth. In our experiment, we choose the moderate threshold $T_d = 1.5f_s + m_u$ to detect outliers. Given threshold T_d, the corresponding point to each node will be removed if the distance is larger than this threshold.

In the second step of the correspondence evaluation, we check if the corresponding points of the remaining nodes are reliable or not. At this point, a spin image-based method (Johnson, 1997) is used. The spin-image is not robust enough to find correspondences, especially for deformable objects, when each spin image of a point on the source surface is compared with all spin images of points on the target surface. There are many similar spin images of the points on the target surface, thus this comparison yields many erroneous correspondences. However,

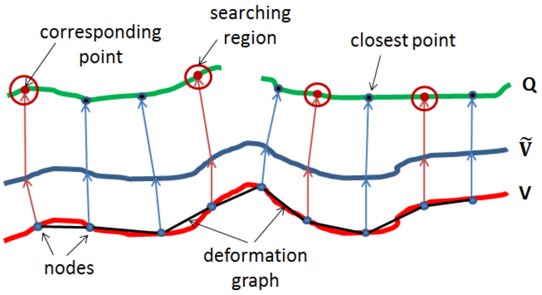


Figure 2: Based on reliable correspondences and an effective strategy for the optimization, our algorithm finds correct alignment between two deformable surfaces.

in our case, the corresponding points of nodes in the graph are known by the CPD and spin images become useful for checking whether these correspondences are sufficient or should be removed. So, given two spin images $\mathbf{I}_g$ and $\mathbf{I}_q$ of a node and its corresponding point (Note that during spin image computation for each node and its corresponding point, the points on the original surfaces $\mathbf{V},\mathbf{Q}$ are used), we measure the similarity between these two images using Eq. 10:

$$\mathrm{C}(\mathbf{I}_g,\mathbf{I}_q) = [atanh(\mathrm{R}(\mathbf{I}_g,\mathbf{I}_q))]^2 - \lambda(\frac{1}{M-3}) \qquad (10)$$

where

$$\mathrm{R}(\mathbf{I}_g,\mathbf{I}_q) = \frac{M\sum u_k t_k - \sum u_k \sum t_k}{\sqrt{[M\sum u_k^2 - (\sum u_k)^2][M\sum t_k^2 - (\sum t_k)^2]}} \qquad (11)$$

is a linear correlation coefficient; u_k, t_k are spin image values at pixels; M is number of overlaping pixels between two spin images; and λ weighs the importance of the correlation coefficient against the confidence in this coefficient.

With a predefined threshold T_c, the remaining nodes will have reliable corresponding points if similarity measures Cs between these nodes and respective corresponding points are greater than T_c.

3.3 Optimization

3.3.1 Movement Optimization

We set $w_{\mathrm{rot}} = 1000, w_{\mathrm{reg}} = 100$ and $w_{\mathrm{con}} = 10$. It means that rigid movement and regularization constraints are favored at the beginning of the optimization while the position constraint is kept small but guarantees that the deformation graph does not drift far away from the target surface. This setup ensures strong stiffness for the deformation and thus the nodes do not move directly towards their corresponding points, but move parallel to the target surface.

We now need to determine how the optimization identifies the corresponding points. To do this, we divide the nodes of the deformation graph into two subsets. The first subset, G_{cor}, consists of nodes whose reliable corresponding points are obtained according to the procedure described in the previous section (phase 1). The second subset, G_{clo}, consists of nodes which do not have corresponding points before the optimization. For the nodes in G_{cor}, we do not want their corresponding points to be fixed during the optimization, but rather give the chance for each node to find a new one surrounding its corresponding point. The deformation graph is thus more flexible when it approaches the target surface and it is still warranted to follow the right direction. Therefore, at each corresponding point on the target surface, we define a radius r to create a region in which the node can search for a closest point as a new corresponding point at the current iteration (Figure. 2). For remaining nodes in G_{clo}, their corresponding points are identified by a closest point method on the entire target surface based on a kd-tree algorithm.

While the $\mathrm{E}_{\mathrm{rot}}$ and $\mathrm{E}_{\mathrm{reg}}$ terms prevent the topology of the deformation graph from being destroyed during the optimization, the $\mathrm{E}_{\mathrm{con}}$ term plays an essential role in pulling the source surface toward the target surface. If we define group A consisting of the nodes in G_{cor} and their respective corresponding points; group B consisting of the nodes in G_{clo} and their respective corresponding points, then elements in group A have stronger confidence than those in group B for this movement. Consequently, we want the contribution of group A to $\mathrm{E}_{\mathrm{con}}$ to be greater than that of group B.

For this purpose, for group A, the contribution of each correspondence is weighed by w^A_{con} and $w^A_{\mathrm{con}} = w_{\mathrm{con}}$. For group B, we first compute the Euclidean distances of the correspondences to make a distance distribution and the median value m_b of this distribution is computed. Then, the contribution of each correspondence of the first half of the correspondences whose Euclidean distances are less than m_b is weighed by $w^{B1}_{\mathrm{con}} = w^A_{\mathrm{con}} = w_{\mathrm{con}}$. However, the contribution of the second half of correspondences whose Euclidean distances are greater than m_b is different.

Let D_{B2} be the total distance of the second half of correspondences, and d_{B2} be the distance of each correspondence of this half. Then, the contribution of each correspondence to the $\mathrm{E}_{\mathrm{con}}$ term is calculated by:

$$w^{B2}_{\mathrm{con}} = (1 - \frac{d_{B2}}{D_{B2}})w_{\mathrm{con}} \qquad (12)$$

The weight w^{B2}_{con} indicates that, for correspondences of the second half obtained by the closest point method, the greater the distance, the less useful the

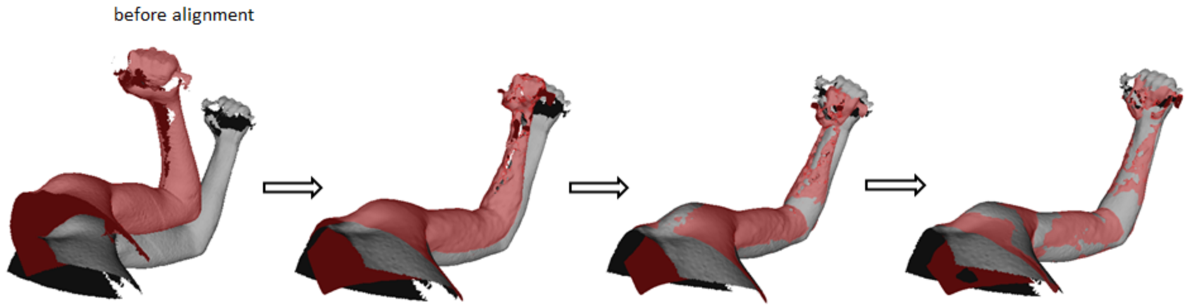


Figure 3: Alignment at different cycles in movement optimization.

correspondences are in the cost function during movement optimization.

After determining the various contributions to the cost function E, the optimization is started. When the deformation graph is far away from the target surface, the prominent weights for the rigid energy term and the regularization energy term make the nodes of the deformation graph move parallel to the target surface. After that, we want to reduce the importannce of these terms when the deformation graph approaches the target surface. To achieve this, an automatic method is employed to update the weights. First, the optimization process runs with a weight set $(w_{\text{rot}}, w_{\text{reg}}, w_{\text{con}}) = (1000, 100, 10)$ until the cost function meets one of two following convergent critera at iteration l:

$$\begin{aligned} |E_l - E_{l-1}| &< 10^{-4} \\ \|\mathbf{z}_l - \mathbf{z}_{l-1}\| &< 10^{-6} \end{aligned} \tag{13}$$

Then, this optimization process restarts a new cycle in which w_{rot} and w_{reg} are reduced by half while w_{con} remains constant. This stratergy is repeated until the $w_{\text{rot}} < 5$ and $w_{\text{reg}} < 0.5$. This whole process is called movement optimization. A sequence of alignments at different cycles of the movement optimization is shown in Figure. 3.

3.3.2 Alignment Optimization

When the movement optimization is completed, it provides a value $\mathbf{z}_o$ which pulls the deformation graph close to the target surface. However, $\mathbf{z}_o$ is not an optimal value when the deformation graph cannot align correctly with the target surface. Consequently, it is necessary to improve this value of $\mathbf{z}_o$ to achieve convergence toward minimum of the cost function E. An alignment optimization is thus carried out.

For the alignment optimization, the corresponding points of nodes in G_{cor} and G_{clo} are redefined. In this optimization, we do not distinguish whether a corresponding point is found by using solely the closest point computation or being based on the node's reliable corresponding point in the first phase. All corresponding points of the nodes are found by the closest point method. Because convergence of the cost function can be influented by poor correspondences, two tests are used to dectect these correspondences. In the first test, a contribution of a correspondence to E$_{\text{con}}$ will be ignored if the corresponding point of the respective node is located on the boundary of the target surface. In the second test, the contribution is removed if the angle between the normal of the corresponding point and the normal of the respective deformed node is greater than a fixed threshold T_n, where the normal of a point is transformed by:

$$\widetilde{\mathbf{n}}_i = \sum_{j=1}^{m} w_j(\mathbf{v}_i)\mathbf{R}_j^{-1T}\mathbf{n}_i \tag{14}$$

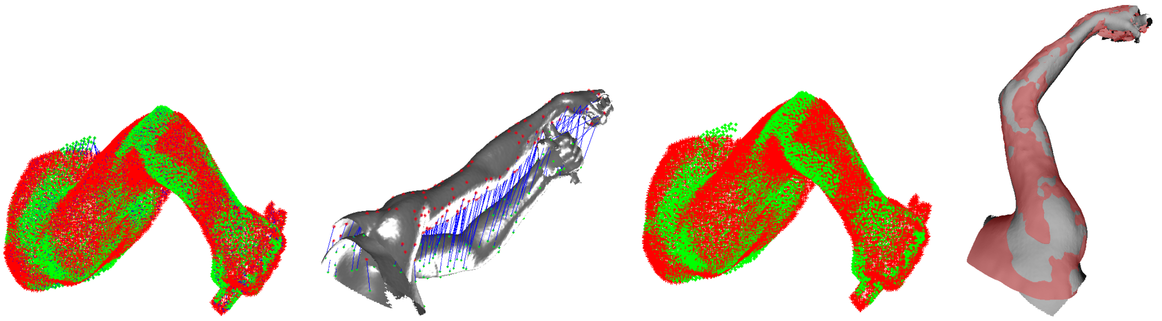
The weights for E$_{\text{rot}}$, E$_{\text{reg}}$ and E$_{\text{con}}$ and other initial values in the alignment optimization are inherited from the ones at the last iteration of the final cycle of the movement optimization. We obtain the optimal value $\mathbf{z}_*$ if one of two following criteria is satisfied:

$$\begin{aligned} |E_l - E_{l-1}| &< 10^{-7} \\ \|\mathbf{z}_l - \mathbf{z}_{l-1}\| &< 10^{-7} \end{aligned} \tag{15}$$

Once the affine transformations are found by optimization, the deformation of each point on the source surface is computed using Eq. 2. Due to the graph structure, transformations that are close to one another will be the most similar. Thus, for consistency and efficiency, we limit the influence of the deformation graph on a particular point to the k-nearest nodes. In our experiment, we use $k = 4$ or $k = 8$.

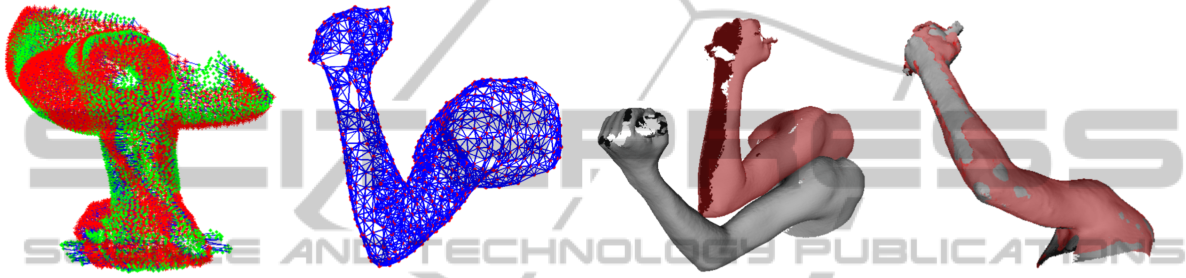
4 RESULTS

In this section, we evaluate our algorithm with real data sets which consist of meshes built from scanned parts of the human body. These data sets present



(a) CPD result (b) Correspondences (c) Alignment for reduced original data (d) Result of our algorithm

Figure 4: Non-rigid registration for two surfaces where the deformation between two surfaces is large.



(a) CPD result (b) Deformation graph (c) Before alignment (d) Result of our algorithm

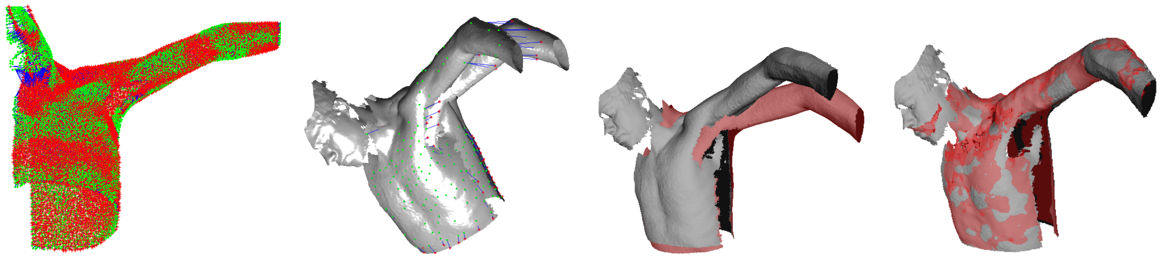
Figure 5: Our result for non-rigid registration although the CPD provides misalignment.

scanned objects which undergo a large deformation between the two scans.

In Figure. 4, we show the steps of our method applied to real data representing a human arm at two different poses. The number of data points on the original source surface $\mathbf{V}_{a1}$ and target surface $\mathbf{Q}_{a1}$ are $N_{\mathbf{V}_{a1}} = 49427$ and $N_{\mathbf{Q}_{a1}} = 47735$ respectively. After reducing the original data and creating a deformation graph, the number of points in the two point sets $\mathbf{Y}_{a1}$ and $\mathbf{X}_{a1}$ are $N_{\mathbf{Y}_{a1}} = 8504$ and $N_{\mathbf{X}_{a1}} = 7978$ respectively where the number of nodes in $\mathbf{Y}_{a1}$ is $m_{\mathbf{Y}_{a1}} = 536$. The two surfaces are very far away from each other, deformation between them is large, and data in the hand area of the source surface is missing on the target surface. Without prior knowledge, the optimization implemented by many algorithms (Allen et al., 2003; Amberg et al., 2007; Mitra et al., 2007; Chang and Zwicker, 2008; Li et al., 2008) will provide a non-optimal value and result in poor alignment. However, using the CPD (Figure. 4(a)) combined with phase 1 of our method, reliable correspondences (Figure. 4(b)) are obtained. These correspondences initialize the optimization process and guide the algorithm toward a good alignment (Figure. 4(d)). Figure. 4(c) shows our result for reduced data. When compared with the alignment achieved by CPD, alignment results are almost similar between the two methods. However, while the CPD is used for the reduced data set, our method can be applied to the original data (large data) set based on the deformation model (Eq. 2).

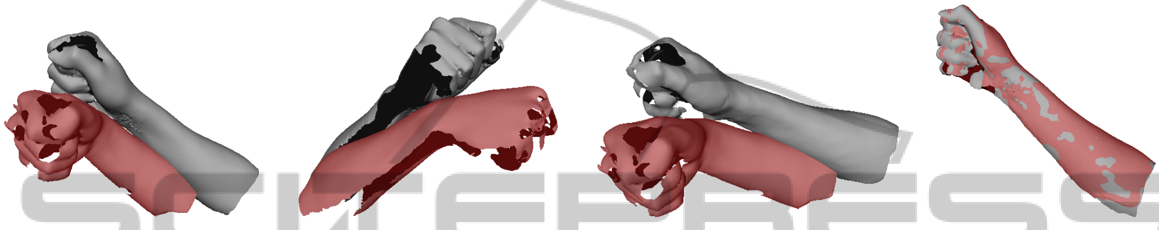
The next experiment is performed on two surfaces of the arm with the number of data points $(N_{\mathbf{V}_{a2}}, N_{\mathbf{Q}_{a2}}, N_{\mathbf{Y}_{a2}}, N_{\mathbf{X}_{a2}}, m_{\mathbf{Y}_{a2}}) = (40905, 47534, 8703, 7987, 629)$. A question that is considered is whether or not the deformation result of the CPD on reduced data can be used to continue the optimization on the original data. This means that we can sample the deformed data in the CPD result uniformly to create a deformation graph. Then, the optimization for the original data is initiated with this deformation graph.

This approach can help to accelerate the optimization. However, in our experiment, we are aware of a potential danger because this procedure can destroy the final alignment of original surfaces if the deformed data of the CPD result does not align well with the reduced target data. Consequently, the deformation for the original data is executed on a wrong deformation graph. In Figure. 5(a), the CPD provides an incorrect alignment where the deformed reduced data (red data) is not fitted well with the reduced target data (green data) near the wrist. Therefore, we always create the deformation graph from the original source surface (Figure. 5(b)). Our result for this experiment is shown in Figure. 5(d) where the alignment near the wrist is correct although there is missing data



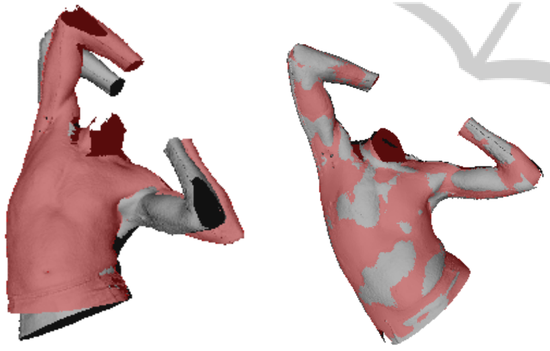
(a) CPD result (b) Correspondences (c) Before alignment (d) Result of our algorithm

Figure 6: When the CPD provides erroneous correspondences, our method can detect these outliers before it carries out the optimization.



(a) Before alighment (view 1) (b) Before alighment (view 2) (c) Lower arm before and after deformation (d) Result of our algorithm

Figure 8: Our algorithm applied for a complex data.



(a) Before alignment (b) Result of our algorithm

Figure 7: Our algorithm simultaneously solves local deformation, global movement and missing data between two surfaces.

on the target surface and the deformation provided by the CPD result is not completely correct.

In Figure. 6, we evaluate our method on a larger data set $(N_{\mathbf{V}_s}, N_{\mathbf{Q}_s}, N_{\mathbf{Y}_s}, N_{\mathbf{X}_s}, m_{\mathbf{Y}_s}) = (71382, 80762, 8540, 8000, 553)$, an upper part of the human body in which the deformation between two surfaces is prominant in the area around the left shoulder. Moreover, near the neck, data on the source surface does not exist on the target surfce. The alignment of the CPD (Figure. 6(a)) pulls the deformed data (red data) and align it poorly with the target data (green data) because it matches the points around the neck on the source data with the points around the head on the target data. Consequently, it provides wrong correspondences in this area. However Figure. 6(b) shows that our method correctly detects these outliers and removes them from the final set of reliable correspondences. Then, the proposed optimization strategy achieves a better result for the final alignment (Figure. 6(d)) compared to CPD.

In the next experiment, we apply our algorithm on two surfaces with $(N_{\mathbf{V}_t}, N_{\mathbf{Q}_t}, N_{\mathbf{Y}_t}, N_{\mathbf{X}_t}, m_{\mathbf{Y}_t}) = (153425, 149845, 9990, 9335, 731)$. The deformation occurs on the entire surfaces and there is a lot of missing data between the two surfaces (Figure. 7(a)). The algorithm deforms the source surface for final alignment (Figure. 7(b)) and solves the local deformation and global movement efficiently. For the missing data on the target surface (near the left elbow), the deformed surface correctly aligns with the target surface although there is a slight misalignment at the top of the left elbow. This phenomenon occurs when there are some nodes which cannot find a corresponding point in this patch, and the optimization is thus based on the rigid energy term and the regularization energy term for the deformation of this patch. Under this condition, the deformation graph cannot change rapidly enough to adapt to the abrupt change of surface curvature at the elbow.

In order to demonstrate the efficiency of the proposed method, a last experiment is carried out for two surfaces where $(N_{\mathbf{V}_{a3}}, N_{\mathbf{Q}_{a3}}, N_{\mathbf{Y}_{a3}}, N_{\mathbf{X}_{a3}}, m_{\mathbf{Y}_{a3}}) = (48626, 45397, 7930, 7981, 833)$. The data of the source surface (red color) and the target surface (grey color) of the lower arm is acquired by a Creaform 3D scanner in our lab (Figure. 8(a), 8(b)). This data is challenging because of missing data on both two surfaces, a largely global transformation between them, local deformation on the entire surface as well as sensor noise (around the hand area). Figure. 8(c) shows the source surface before and after deformation and illustrates that our method overcomes all the challenges of this data to deform correctly the source surface toward the target surface. Final alignment between the deformed surface and the target surface is shown in Figure. 8(d).

5 CONCLUSIONS

We have presented an efficient non-rigid registration algorithm to align two partially overlapping surfaces. Contrarily to other algorithms which normally require prior knowledge to obtain final alignment, our method is implemented automatically without any user-intervention for constraining the deformation and without making other assumptions. To achieve this, the algorithm is divided into two phases. While the first phase provides initial correspondences to constrain the optimization, the strategy of the second phase is performed in two steps in which the first step aims to move the source surface close to the target surface and the second step forces the two surfaces to coincide accurately. The experimental results prove that our algorithm can be applied to data sets where the deformation between the two surfaces is severe and prior knowledge is not available.

We are also aware of some limitations of our algorithm. Currently, all initial correspondences are treated equally in phase 1 without considering how accurate they are. We need to consider the contribution of each correspondence by using contribution weights before using it in phase 2. This change may increase convergence speed of the optimization process in phase 2. Another limitation is related to the deformation model. Because this model determines the influence area of affine transformations based on Euclidean distance, this property can create strange deformations when two nodes are close with respect to Euclidean distance but far away with respect to geodesic distance (two nodes of two close fingers, for example). This limitation should be considered in future work. Once these limitations are resolved, we plan to develop the algorithm so it can be applied for global registration including several surfaces in order to reconstruct a completely deformable object.

ACKNOWLEDGEMENTS

We are grateful to Myronenko et al for providing the CPD implementation and the GRAIL laboratory-University of Washington for providing the 3D data. This research was supported by the NSERC-Creaform Industrial Research Chair on 3D Scanning.

REFERENCES

Allen, B., Curless, B., and Popović, Z. (2002). Articulated body deformation from range scan data. *ACM Trans. Graph.*, 21(3):612–619.

Allen, B., Curless, B., and Popović, Z. (2003). The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594.

Amberg, B., Romdhani, S., and Vetter, T. (2007). Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8.

Anguelov, D., Srinivasan, P., Pang, H.-C., Koller, D., Thrun, S., and Davis, J. (2004). The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS'04*, pages –1–1.

Besl, P. and McKay, N. D. (1992). A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256.

Brown, B. J. and Rusinkiewicz, S. (2007). Global non-rigid alignment of 3-d scans. *ACM Trans. Graph.*, 26(3).

Chang, W. and Zwicker, M. (2008). Automatic registration for articulated shapes. In *Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1459–1468, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2724–2729 vol.3.

Chui, H. and Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.*, 89(2-3):114–141.

DeVore, J. (2008). *Probability and Statistics for Engineering and the Sciences: Enhanced [With Glossary of Symbols Booklet]*. Available 2010 Titles Enhanced Web Assign Series. Brooks/Cole, Cengage Learning.

Frome, A., Huber, D., Kolluri, R., Blow, T., and Malik, J. (2004). Recognizing objects in range data using regional point descriptors. In *EUROPEAN CONFERENCE ON COMPUTER VISION*, pages 224–237.

Huang, Q.-X., Adams, B., Wicke, M., and Guibas, L. J. (2008). Non-rigid registration under isometric deformations. In *Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1449–1457, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Jian, B. and Vemuri, B. C. (2011). Robust point set registration using gaussian mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1633–1645.

Johnson, A. E. (1997). Spin-images: A representation for 3-d surface matching. Technical report.

Li, H., Sumner, R. W., and Pauly, M. (2008). Global correspondence optimization for non-rigid registration of depth scans. In *Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1421–1430, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Madsen, K., Nielsen, H. B., and Tingleff, O. (2004). Methods for non-linear least squares problems (2nd ed.).

Mian, A., Bennamoun, M., and Owens, R. (2004). From unordered range images to 3d models: a fully automatic multiview correspondence algorithm. In *Theory and Practice of Computer Graphics, 2004. Proceedings*, pages 162–166.

Mitra, N. J., Flöry, S., Ovsjanikov, M., Gelfand, N., Guibas, L., and Pottmann, H. (2007). Dynamic geometry registration. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, SGP '07, pages 173–182, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2262–2275.

Pauly, M., Mitra, N. J., Giesen, J., Gross, M., and Guibas, L. J. (2005). Example-based 3d scan completion. In *Proceedings of the third Eurographics symposium on Geometry processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152.

Submuth, J., Winter, M., and Greiner, G. (2008). Reconstructing animated meshes from time-varying point clouds. In *Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1469–1476, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Sumner, R. W., Schmid, J., and Pauly, M. (2007). Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 26(3).

Wand, M., Adams, B., Ovsjanikov, M., Berner, A., Bokeloh, M., Jenke, P., Guibas, L., Seidel, H.-P., and Schilling, A. (2009). Efficient reconstruction of non-rigid shape and motion from real-time 3d scanner data. *ACM Trans. Graph.*, 28(2):15:1–15:15.