Global registration of multiple point clouds embedding the Generalized Procrustes Analysis into an ICP framework

Roberto Toldo
Dipartimento di Informatica
Università di Verona
Strada Le Grazie 15, 37134 Verona, Italy
roberto.toldo@univr.it

Alberto Beinat and Fabio Crosilla
Dipartimento di Georisorse e Territorio
Università di Udine
Via Cotonificio 114, 33100 Udine (Italy)
alberto.beinat@uniud.it fabio.crosilla@uniud.it

Abstract

In this paper we propose a novel approach to cope with the problem of global registration of multiple point clouds, where point correspondences and view order are unknown. The method iteratively minimizes a cost function considering all the views simultaneously. The proposed solution takes advantage of the well-known Generalized Procrustes Analysis, seamlessly embedding the mathematical theory in an Iterative Closest Point framework. A variant of the method, where the correspondances are non-uniformly weighted using a curvature-based similarity measure, is also presented. Several experiments show the robustness of both the proposed solutions.

1. Introduction

Extracting a model of a real object from tridimensional (3D) surface measurement is a common task with several applications in various contexts of computer vision, computer graphics, reverse engineering and digital photogrammetry. In most cases, a single view is not sufficient to describe the entire object and multiple acquisitions of the surface are necessary. Typically the views are obtained from multiple scanners or from a single scanner stationed at different locations and orientation. Each view is usually represented as a dense point cloud and the views are partially overlapping each other. When multiple views are present, a problem of surface registration arises, i.e. the different views have to be located in a common reference system. In other words, the objective is to determine a rigid transformation for each partial surface view, in order to find the optimal alignment among all of them. In real cases, the point-to-point correspondences for the overlapping regions are unknown and the view sequence, i.e. the neighborhood of each view, is not always available.

When only two-views are present, we are required to find the transformation that minimizes the distance between the two clouds. Two-views registration is a well studied problem in the literature. The Iterative Closest Point (ICP) algorithm is the most common solution [1]. It iteratively revises the transformation - usually composed of scale, rotation and translation (SRT) - needed to minimize the distance between the points of the two clouds. The point correspondances are extracted considering the closest point in the other view. Several variants have been proposed to improve either the speed or the robustness of the method, e.g. employing a kd-tree, thresholding the maximum point to point correspondence distance, or weighting every correspondence with a similarity measure. An excellent review about the existing variants can be found in [16].

Multiple view registration is a more complex problem. There are two strategies towards solving the problem, local (sequential) registration and global (simultaneous) registration. The sequential registration approach involves the sequential alignment of two overlapping views at a time. This commonly used approach is not optimal because errors can accumulate and propagate. Moreover the view sequence have to be known or manually specified. On the other hand, global registration attempts to align all scans at the same time by distributing the registration errors evenly over all the overlapping views.

Global multiple view registration schemes were considered by several researchers. One of the first method was proposed in [13] where the “mean rigid shape” of a set of possibly incomplete tuples in any dimension is computed and revised at each step. The method is simple and iterative, but does not include a matching algorithm and the correspondances were manually specified. A comparative study of similar multiple view registration schemes was performed by Cunnington and Stoddart [6]. In [15] is presented a method that first aligns the scans pairwise with each other and then uses the pairwise alignments as constraints in a multiview step. The aim is to evenly diffuse the pairwise registration errors, but the method itself is still based on pairwise alignments. In [20] an optimization in the transformations matrix space is derived. Point matches are found using a closest point relation for each subsequent view.

More recently, in [3] a novel global registration method, that distributes registration errors evenly across all views, has been proposed. It works in the transformation space, however the sequence of the views is required. In [12] a technique that involves a manifold optimization approach has been presented. It performs an explicit optimization on the manifold of rotations. The algorithm shows a fast convergence, but it does not include a matching strategy and point to point correspondances are assumed to be known. A technique that derives a bayesian formulation of the registration have been introduced in [11]. It aims to register the partial views to a final reconstructed watertight surface.

If the correspondances among the views are known, Generalized Procrustes Analysis (GPA) could be employed in order to estimate each SRT transformation in one step [5]. In this work we propose a novel multiple view registration technique that make use of the GPA and seamlessly integrate it in a ICP framework. A principle of *mutual correspondence* is employed to automatically define the matches at each step. The proposed solution reaps the benefits of most the aforementioned methods: it is completely automatic and neither requires the prior knowledge of the view sequence nor of the correspondances, it is simple, theoretically sound and computationally efficient.

This paper structured as follows: in sec. 2 the GPA is overviewed, in sec. 3 the method is presented, in sec. 4 a weighted variant is proposed while in sec. 5 several experiments are shown, comparing our approach with a classical sequential ICP.

2. Overview of the Generalized Procrustes Analysis

The Orthogonal Procrustes (OP) problem [17] is an optimization problem which solution gives the orthogonal transformation matrix R between two $p \times k$ dimensional matrices A and B that minimizes the least squares of the residuals:

$$E = AR - B \tag{1}$$

The problem has been extended in [18] where an unknown rotation R, an unknown translation t and an unknown scale factor c were introduced. This method is often referred as an Extended Orthogonal Procrustes (EOP). The residuals to minimize are the following:

$$E = cAR + jt^T - B \tag{2}$$

where j is a $(p \times 1)$ unit vector, t is a $(k \times 1)$ dimensional translation vector and c is a scale factor. Weight matrices, both for components and points, can be added to the EOP model, giving rise to the Weighted Extended Orthogonal Procrustes Analysis (WEOP). The residuals to minimize become:

$$tr((cAR + jt^T - B)^T W_P (cAR + jt^T - B) W_K) = e \tag{3}$$

where W_P and W_K are respectively $(p \times p)$ and $(k \times k)$ dimensional weight matrices. A direct solution can be inferred if $W_K = I$, otherwise an iterative solution must be applied. Generalized Procrustes Analysis (GPA) is a well-known technique that provides a least-squares solution when more than two model points matrices are present [9, 19, 8, 7, 2]. It minimize the following least squares objective function:

$$tr(\sum_{i=1}^{m} \sum_{j=i+1}^{m} ((c_i X_i R_i + jt_i^T) - (c_j X_j R_j + jt_j^T))^T \\ ((c_i X_i R_i + jt_i^T) - (c_j X_j R_j + jt_j^T))) = e \tag{4}$$

where $X_1, X_2, \ldots, X_m$ are m model points matrices, which contain the same set of p points in k dimensional m different coordinate systems. The GPA problem has an alternative formulation. Said $X_i^p = c_i X_i R_i + jt_i^T$, the following measures:

$$\sum_{i<j}^{m} ||X_i^p - X_j^p||^2 = \sum_{i<j}^{m} tr((X_i^p - X_j^p)^T (X_i^p - X_j^p)) \tag{5}$$

$$m \sum_{i<j}^{m} ||X_i^p - K||^2 = m \sum_{i<j}^{m} tr((X_i^p - K)^T (X_i^p - K)) \tag{6}$$

are perfectly equivalent [2], where K is the unknown geometrical centroid. Therefore Eq. 6, instead of Eq. 5, can be minimized in order to determine the unknowns $c, R, t_i (i = 1 \ldots m)$. The Matrix

$$K = \frac{1}{m} \sum_{i=1}^{m} X_i^p \tag{7}$$

corresponds to the least squares estimation of the centroid. As suggested in [5] the solution of the system can be found iteratively. First the centroid K is initialized. At each step a direct solution of the transformation parameters of each model points matrix A_i with respect to the centroid K is found by means of a WEOP solution. After the update, a new centroid can be estimated. The procedure continues until global convergence, i.e. the stabilization of the centroid K. In real applications all of the p points may not be visible in all of the model points matrices $X1, X2, , X_m$. In order to cope with the missing point case, Commandeur [4] proposed a method based on the association of a diagonal binary $(p \times p)$ matrix M_i, in which the diagonal elements are 1 or 0, according to the existence or absence of the point in the i-th model. This solution can be considered as zero weights for the missing points. Least squares objective function is defined as follows in the missing point case:

$$m\sum_{i<j}^{m}||X_i^p - K||^2 = m\sum_{i<j}^{m} tr((X_i^p - K)^T M_i (X_i^p - K)) \tag{8}$$

with:

$$K = (\sum_{i=1}^{m} M_i)^{-1}(\sum_{i=1}^{m} M_i(c_i X_i T_i + j t_i^T)) \tag{9}$$

In order to obtain a more general scheme, one should consider the combined weighted/missing point solution. The weight matrix P_i and the binary matrix M_i can be combined in a product matrix $D_i = M_i P_i = P_i M_i$. The corresponding least squares objective function becomes:

$$m\sum_{i<j}^{m}||X_i^p - K||^2 = m\sum_{i<j}^{m} tr((X_i^p - K)^T P_i M_i (X_i^p - K)) \tag{10}$$

with:

$$K = (\sum_{i=1}^{m} P_i M_i)^{-1}(\sum_{i=1}^{m} P_i M_i(c_i X_i T_i + j t_i^T)) \tag{11}$$

3. Generalized Iterative Closest Point algorithm

In this section we will explain how to take advantage of the previously described GPA to define an iterative closest point algorithm that can register multiple view in a simultaneous way. The theoretical result presented in [2], where eq. 5 and 6 were demonstrated to be perfectly equivalent, gives an important hint on the problem resolution. We do not need to use every combination of pairs for each view, thus avoiding ordering or computational issues. The only ambiguity to solve is the definition of the centroid K at every step of the algorithm, so that the transformation of each view can be updated according to the new centroid position. Unlike the algorithm introduced in [5] where the correspondances where manually specified, in an ICP framework they are not fixed and are assigned at every step by finding the closest neighbor of each point. In general, not every point will be visible in every view, so in the following eq. 10 and eq. 11 will be considered with uniform weights, i.e. $P_i = I$.

Our proposed solution stems from the consideration of regarding only at a subset of compatible matches, as also suggested in [16]. Instead of fixing a threshold on the distance for the compatible matches, we take into account only matches of points, belonging to different views, that are *mutually nearest neighbor*. We say that a point $x \in A$ is mutually nearest neighbor to another point $y \in B$ if the closest neighbor $\in B$ of x is y and the closest neighbor $\in A$ of y is x (Fig. 1(a)). This is a simple yet powerful approach to discard pairs of points that are very far each other without introducing any threshold. It is in fact more probable that the *mutual nearest neighbor relation* is verified for closer matches.

One could think to build an imaginary graph, considering every pair of points that are in a mutual nearest neighbor relation. The edge of the graph will be represented by the mutual nearest neighbor relation itself. In this way, the graph will be composed by completely independent sets as shown in Fig. 1(b). Each independent set will be represented by an unique point in K, computed as the centroid of the points of the independent set. Every point of an independent set is matched with the newly computed centroid in K (Fig. 1(c)). Using these correspondances, we can thus update simultaneously every transformation from the views to K. Points of different views belonging to the same independent set are matched with the same element in K and are thus driven into the same position. It may occurs that more than one point of the same view are present in the same independent set; this is a rare degenerate case and the set is simply discarded.

Algorithm 1 GPA-ICP

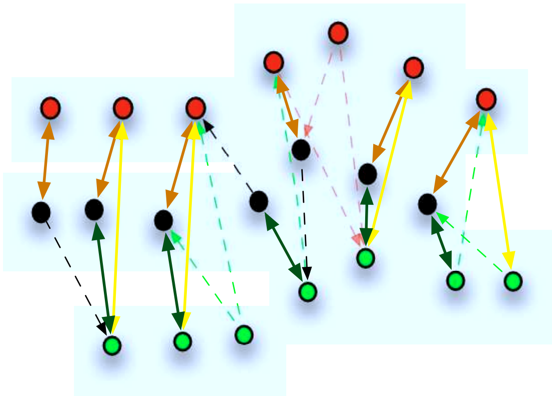
Input: m clouds of points $[X_i...X_m]$.
Output: m RST transformation $[T_i...T_m]$.

1. For every view, find the pairs of points that are *mutually nearest neighbor* of points in other views.
2. Define a new point in K as the centroid of each independent set of mutually nearest neighbors.
3. Estimate the RTS transformation parameters $[T_i...T_m]$ using 10.
4. Transform each view using the estimated parameters.
5. Iterate until global convergence or for a fixed number of steps.

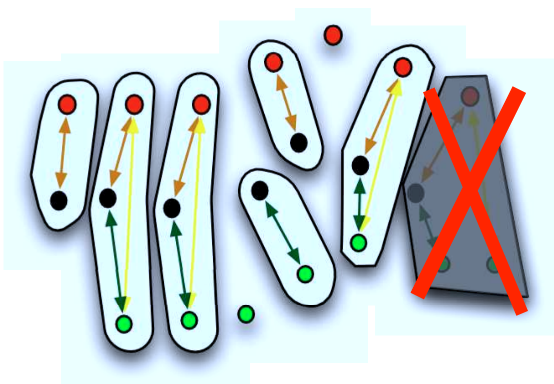
4. A weighted variant

In the previous section we considered only uniformly weighted correspondances, i.e. $P_i = I$. In the general case, however, it is possible to give a different weight for every match. At each step a subset of points in a view is in correspondence with a subset of elements of K, which are computed as the centroids of the independent sets. The natural solution is thus to define a different weight for every element of K, that can be derived from a compatibility criteria among the points of each independent set.

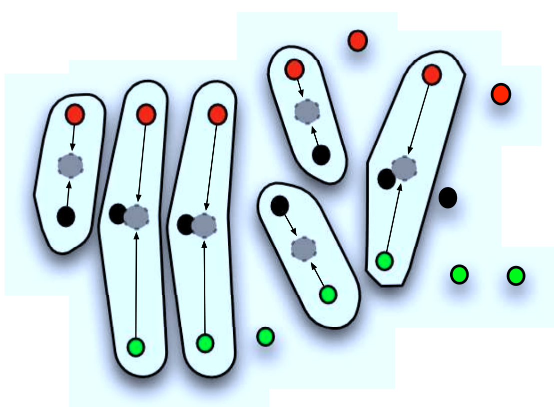
We suggest to employ the *Shape Index* (SI) that repre-



(a) Nearest Neighbor and Mutual Nearest Neighbor relation.



(b) Independent sets.



(c) Centroids correspondences.

Figure 1. Example with three different clouds of bi-dimensional points (red, green and black points). In the top figure an arrow depicts a closest neighbor relation and a double pointed arrow denotes a mutual closest neighbor relation. In the middle figure the resulting independent sets are shown. The rightmost independent set contains two points from the green set and is thus discarded. In the bottom figure the arisen correspondances with the centroid (grey points - elements of K) of each independent set are shown.

sents a concise way to express the type of shape of a region [14]:

$$s = -\frac{2}{\pi}\arctan\left(\frac{k_1+k_2}{k_1-k_2}\right) \quad k_1 > k_2 \qquad (12)$$

where k_1, k_2 are the principal curvatures of a generic vertex. The curvature for a vertex is computed by fitting a quadric surface on the neighboring vertices. The SI is bounded and varies in $[-1, 1]$: a negative value corresponds to concavities, whereas a positive value represents a convex surface. A weight is thus associated with each element of K and consequently with each correspondence. The weight is computed employing the Median of Absolute Deviations (MAD) among the points of each Independent Set (IndS):

$$MAD_{IndS} = median_i(|SI_i - median_j(SI_j)|) \qquad (13)$$
$$i, j \in IndS$$

The MAD varies in $[0, 1]$, the weight associated to each match is the inverse of the MAD, i.e. $w = 1 - MAD_{IndS}$.

5. Results

We tested the proposed approach (ICP-GPA) and its weighted variant (ICP-WGPA) with 8 global alignment experiments: 4 composed of 10 views and 10000 points per view and 4 obtained subsampling the views to 1000 points. The initial configurations, showed in figures, were misaligned to some extent. The views come from laser scanning and the subjects are a screw, a dog, a bunny and a dinosaur. We compared with the classical ICP algorithm applied sequentially for each pair of subsequent views (ICP), and a variant where distant matches are rejected according to the X84 rule (ICP-x84) [10]. The sequential order of the views is know, but, unlike for the sequential ICP, is not needed for our approach. The final alignments for the different methods, along with the initial configuration, are shown in Fig. 2, Fig. 3, Fig. 4, Fig. 5.

It can be noticed how both the plain and weighted variant of our method converge to a global minima in every example, whereas ICP and its variant ICP-x84 fail in some of them. In particular, ICP without threshold roughly fails with the simplified "screw" and with both "bunny" and "pitbull" experiments, whereas the x84 variant fails with both the "bunny" experiments and with the simplified "pitbull" experiment . As said before, besides the worse results, the sequential ICP algorithms require that the order of the view is *a-priori* known.

We further analyzed the results by computing the residual at each step of the algorithms (Fig. 6). The residuals are computed as the mean square distance between the points of every view and their closest neighbor considering every other view. The proposed approach and its weighted variant, perform better in every experiments, with the weighted variant showing, with the exception of the "Screw" experiments, a slightly faster convergence rate. In some experiments sequential ICP and its thresholded variant, converge to a solution that degeneratively increases the global residual. Only in a few experiments the global residual is comparable to out approach, but never lower. It should also be noticed that the number of points seems not to affect the effectiveness of the method, since it works well even with the subsampled cloud of points.

The algorithm has been written entirely in matlab language. The running times are comparable with the sequential ICP: it takes about 12 minutes to process 200 steps with 10 views and 10000 points per view, whereas the sequential ICP approach takes about 10 minutes. The code is available for download [1].

6. Discussion

In this work we proposed a novel approach to cope with the problem of simultaneous alignment of multiple views. The proposed approach is theoretically sound, basing on the well-known GPA theory and gives an efficient and elegant solution to automatically align views in a ICP framework. Several experiments have shown the effectiveness of our method compared with the classical sequential ICP algorithm. The algorithm is robust and the global minimum is reached even when the pre-alignment is roughly defined. Furthermore, the approach showed a superior accuracy in every experiment compared to the classical sequential ICP. The method has a wide range of applications, and can be applied in any case an alignment of multiple views, e.g. coming from laser scanning, is required to be automatically refined. Future improvements may be acquired by introducing more elaborated weight functions for the correspondances.

References

[1] P. Besl and H. McKay. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992. 1

[2] I. Borg and P. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Verlag, 2005. 2, 3

[3] U. Castellani, A. Fusiello, and V. Murino. Registration of multiple acoustic range views for underwater scene reconstruction. *Computer Vision and Image Understanding*, 87(1):78–89, 2002. 2

[4] J. Commandeur. *Matching configurations*. DSWO Press, Leiden University, Leiden, Netherlands, 1991. 2

[5] F. Crosilla and A. Beinat. Use of generalised Procrustes analysis for the photogrammetric block adjustment by independent models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 56(3):195–209, 2002. 2, 3

[6] S. Cunnington and A. Stoddart. N-view point set registration: A comparison. In *British Machine Vision Conference*, volume 2, 1999. 1

[7] I. Dryden and K. Mardia. *Statistical shape analysis*. Wiley New York, 1998. 2

[8] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 285–339, 1991. 2

[9] J. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 2

[10] F. Hampel, P. Rousseeuw, E. Ronchetti, and W. Stahel. *Robust Statistics: the Approach Based on Influence Functions*. Wiley Series in probability and mathematical statistics. John Wiley & Sons, 1986. 4

[11] Q. Huang, B. Adams, and M. Wand. Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, page 223, 2007. 2

[12] S. Krishnan, P. Lee, J. Moore, and S. Venkatasubramanian. Global registration of multiple 3D point sets via optimization-on-a-manifold. In *Proceedings of the third Eurographics symposium on Geometry processing*, page 187. Eurographics Association, 2005. 2

[13] X. Pennec. Multiple registration and mean rigid shapes: Application to the 3D case. In *Image Fusion and Shape Variability Techniques (16th Leeds Annual Statistical Workshop)*, pages 178–185, 1996. 1

[14] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys*, 34(2), 2002. 4

[15] K. Pulli. Multiview registration for large data sets. *Proc. 3DIM*, 1999. 1

[16] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001. 1, 3

[17] P. Schonemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 2

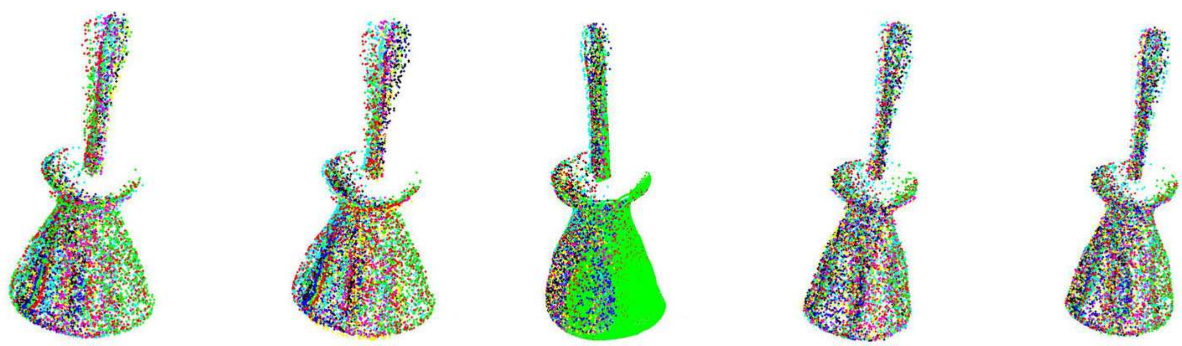
[18] P. Schonemann and R. Carroll. Fitting one matrix to another under choice of a central dilation and a rigid motion. *Psychometrika*, 35(2):245–255, 1970. 2

[19] J. Ten Berge. Orthogonal Procrustes rotation for two or more matrices. *Psychometrika*, 42(2):267–276, 1977. 2

[20] J. Williams and M. Bennamoun. Multiple view 3d registration using statistical error models. *Vision Modelling and Visualization*, 1999. 2

[1] http://profs.sci.univr.it/~toldo

(a) Initial Configuration. (b) Final Configuration ICP. (c) Final Configuration ICP-x84. (d) Final Configuration ICP-GPA. (e) Final Configuration ICP-WGPA.



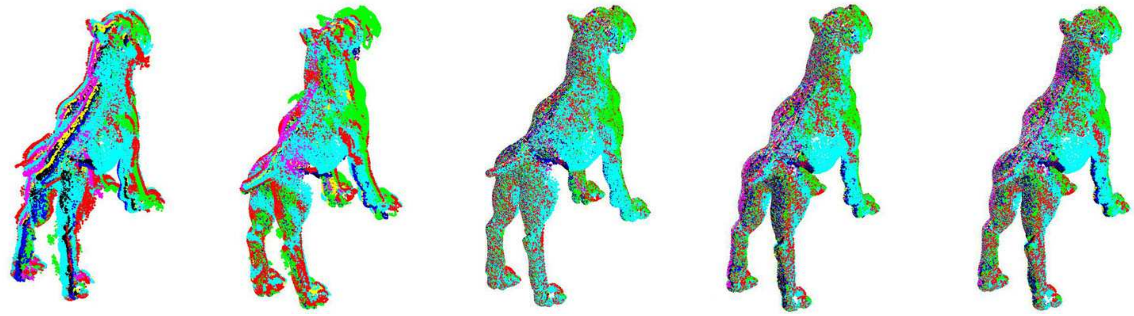
(f) Simpified mesh initial Configuration. (g) Final Configuration ICP. (h) Final Configuration ICP-x84. (i) Final Configuration ICP-GPA. (j) Final Configuration ICP-WGPA.

Figure 2. “Screw” example.

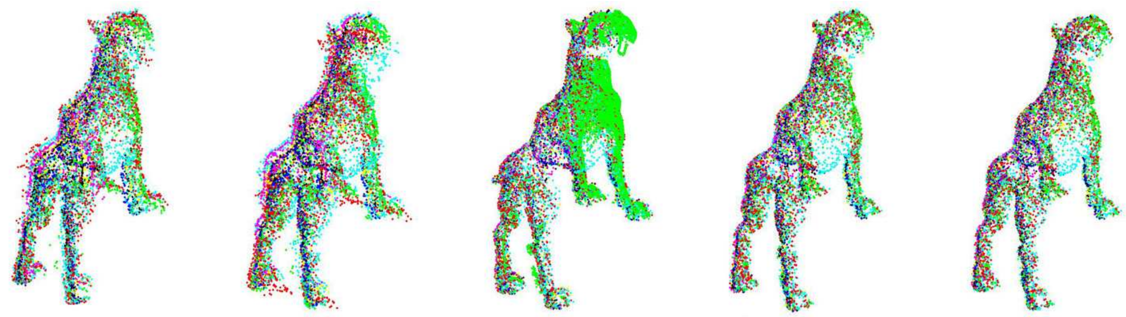
(a) Initial Configuration. (b) Final Configuration ICP. (c) Final Configuration ICP-x84. (d) Final Configuration ICP-GPA. (e) Final Configuration ICP-WGPA.

(f) Simpified mesh initial Configuration. (g) Final Configuration ICP. (h) Final Configuration ICP-x84. (i) Final Configuration ICP-GPA. (j) Final Configuration ICP-WGPA.

Figure 3. “Bunny” example.

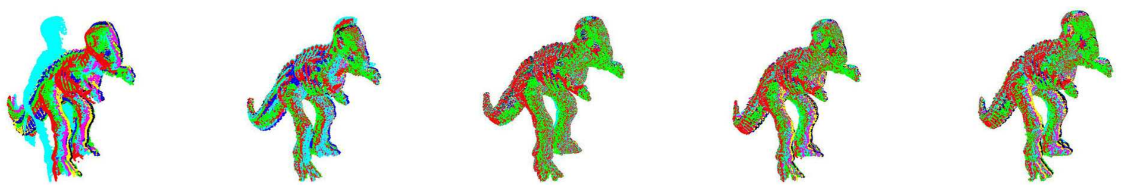


(a) Initial Configuration. (b) Final Configuration ICP. (c) Final Configuration ICP-x84. (d) Final Configuration ICP-GPA. (e) Final Configuration ICP-WGPA.

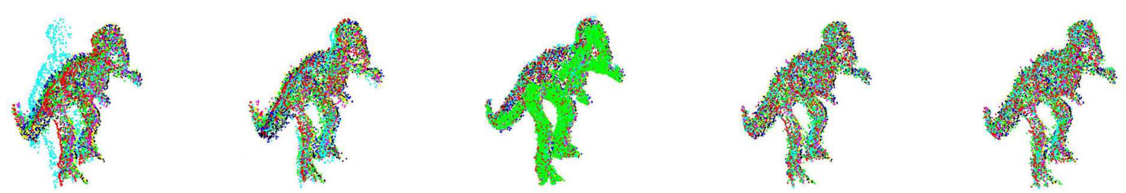


(f) Simpified mesh initial Configuration. (g) Final Configuration ICP. (h) Final Configuration ICP-x84. (i) Final Configuration ICP-GPA. (j) Final Configuration ICP-WGPA.

Figure 4. “Pitbull” example.

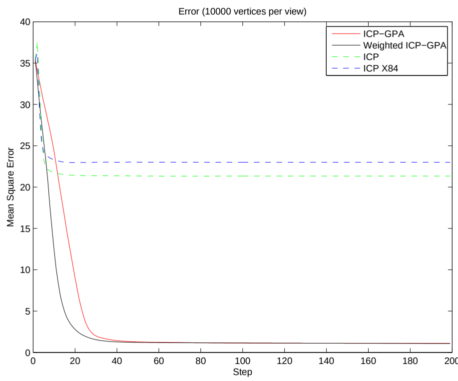


(a) Initial Configuration. (b) Final Configuration ICP. (c) Final Configuration ICP-x84. (d) Final Configuration ICP-GPA. (e) Final Configuration ICP-WGPA.

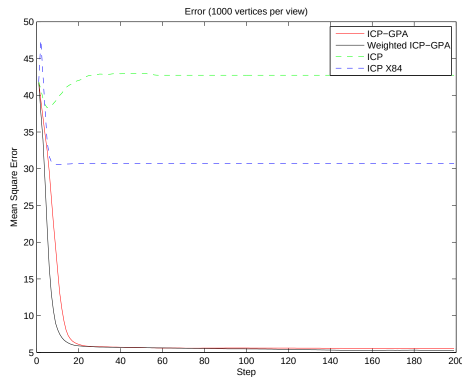


(f) Simpified mesh initial Configuration. (g) Final Configuration ICP. (h) Final Configuration ICP-x84. (i) Final Configuration ICP-GPA. (j) Final Configuration ICP-WGPA.

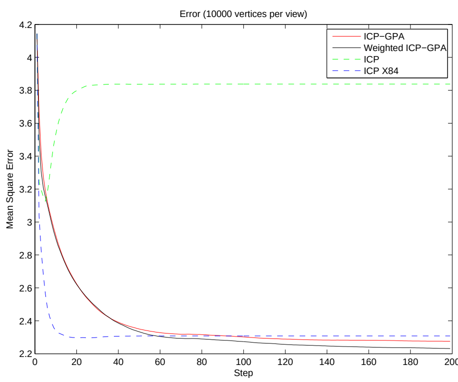
Figure 5. “Dino” example.



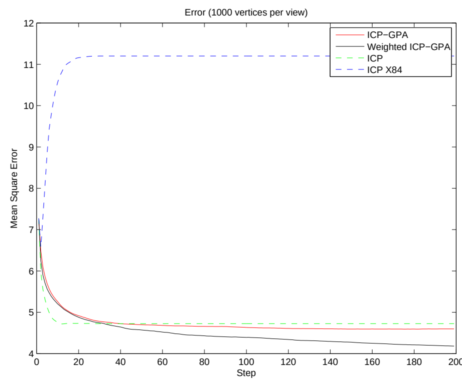
(a) “Bunny”.



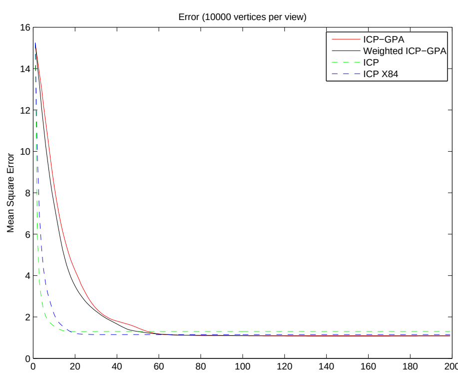
(b) Simplified “Bunny”.



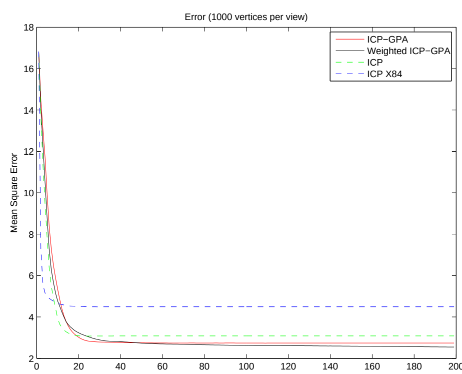
(c) “Pitbull”.



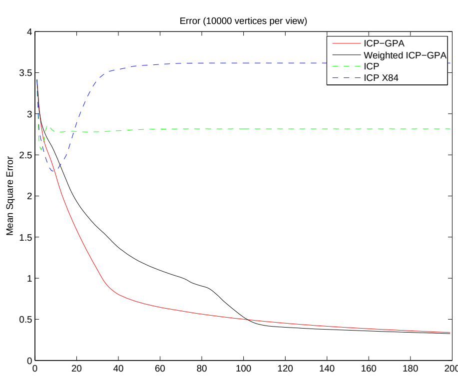
(d) Simplified “Pitbull”.



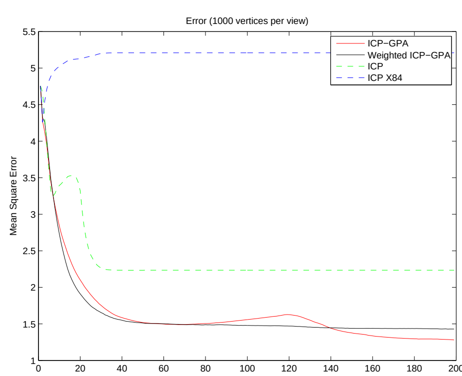
(e) “Dino”.



(f) Simplified “Dino”.



(g) “Screw”.



(h) Simplified “Screw”.

Figure 6. Residuals at each step of the algorithms. The residual error is computed as the mean square distance between every point of every view and its closest neighbor considering every other view.