A GPU-Based Adaptive Algorithm for Non-Rigid Surface Registration

Antonio C. S. Souza*
Federal University of Bahia, Brazil
Federal Institute of Bahia, Brazil

Márcio C. F. Macedo†
Federal University of Bahia, Brazil

Antônio L. Apolinário Jr.‡
Federal University of Bahia, Brazil

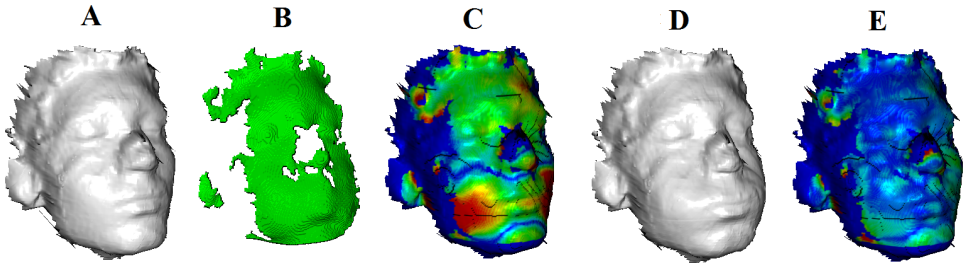

Figure 1: Given reference (A) and deformed (B) surfaces related by a deformation error (C), our GPU-based adaptive non-rigid registration algorithm successfully captures the main deformation present on the deformed surface (D) by minimizing the initial error estimated (E) also running at interactive frame rate.

ABSTRACT

Non-rigid surface registration is fundamental when accurate tracking or reconstruction of 3D deformable shapes is desirable. However, the majority of non-rigid registration methods are not as fast as the ones developed in the field of rigid registration. Fast methods for non-rigid surface registration are particularly interesting for markerless augmented reality applications, in which the object being used as marker can support non-rigid user interaction. In this paper, we present an adaptive algorithm for non-rigid surface registration. Taking advantage from this adaptivity and the parallelism of the GPU, we show that the proposed algorithm is capable to achieve near real-time performance with an approach as accurate as the ones proposed in the literature.

Index Terms: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities.

1 INTRODUCTION

In general, Augmented Reality (AR) applications deal with rigid objects because of the availability of real-time 3D rigid registration algorithms [2]. In this context, there is a set of objects that naturally demands a special attention to possible non-rigid user interaction, such as face (which can be deformed by facial expressions) or hand (which can be deformed by the movement of fingers). However, non-rigid surface registration still has some technical challenges (e.g. high execution time) which makes it unsuitable for use in AR applications.

The most common methods for fast non-rigid registration rely on the building of a deformation graph to represent the deformation for a given surface. In this representation, a graph is built from an uniform sampling on the source surface. Each node of this structure is associated with a 3D affine transformation which influences the nearby space. Therefore, this deformation graph allows a source surface to be deformed to a target surface based on the best affine transformation estimated for each node by using a non-linear optimization algorithm. We take advantage from this basis representation to present a fast adaptive algorithm for 3D non-rigid registration.

*e-mail:antoniocarlos@ifba.edu.br
†e-mail:marciocfmacedo@gmail.com
‡e-mail:apolinario@dcc.ufba.br

2 NON-RIGID REGISTRATION ALGORITHM

2.1 Surface Acquisition

The proposed algorithm requires a depth sensor and a computer equipped with GPU. In this paper, the Kinect sensor is used to capture object's depth data.

The main goal of the proposed algorithm is the fast non-rigid registration of two consecutive source (P_s) and target (P_t) point clouds (Figure 1, A and B respectively) to be used in a markerless AR application. Therefore, the input data in our algorithm must be given according to the markerless AR environment being used. To validate our approach, we have used the environment proposed in [1]. In this environment, markerless tracking is performed based on a 3D reference model generated from the object that will be tracked by the application. Rigid tracking is realized by a real-time variant of the ICP algorithm [2], which estimates the transformation that aligns the current depth frame (D_t) captured by the depth sensor with the previous depth frame (D_s) represented by the 3D reference model.

Given D_s and D_t, they are used to segment the object of interest in the scene. A 2D bounding box that contains both source and target objects is computed from both D_s and D_t. From the bounding box, it is discarded from the memory every position outside the xy-axis of the bounding box. Afterwards, all these data allocated on the GPU can be used by the proposed non-rigid registration algorithm.

2.2 Deformation Model

Our non-rigid registration algorithm is inspired in the Embedded Deformation (ED) algorithm proposed in [3]. However, to achieve near real-time performance, we have implemented it fully taking advantage from the parallelism provided by the GPU. Furthermore,

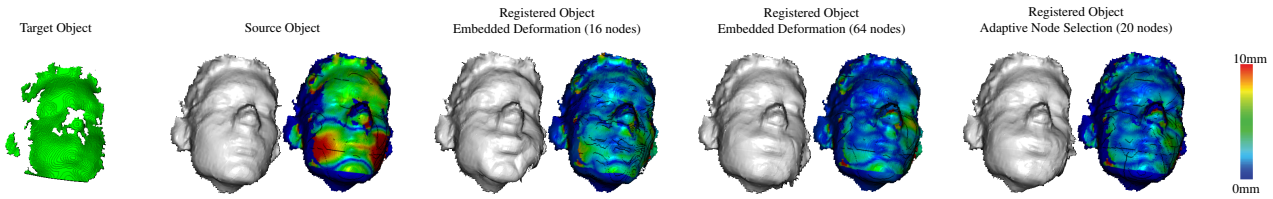

Figure 2: Accuracy comparison between ED algorithm and our adaptive approach with respect to the node selection.

to improve accuracy and to reduce the computational cost of the entire solution, we have adopted an adaptive approach.

For the selection of nodes in the deformation graph G, we use a quadtree-based algorithm which runs on the GPU. The algorithm for node selection can be divided in two steps: the building of the quadtree and the adaptive refinement/collapse of nodes in G.

We build the quadtree in the first iteration of our algorithm by associating each GPU thread with a position on the 2D projection of P_s to select a node iteratively for each level of the tree. For collapsing of nodes, given a region C around the 2D position of the node, the thread computes the average of an energy function (defined in [3]) for each point $\in C$. If the average error is below a certain threshold th_n, the children nodes in C must be collapsed and the region C is represented by the parent node. For refinement of nodes, the GPU thread computes the average error around a region C. If this error is above th_n, the node must be refined, creating four children nodes.

For the selection of constraints, instead of using all the points from P_s as constraints for the optimization, we employ an adaptive algorithm that performs the selection of constraints based on the residual error previously measured. Given a region on P_s, the higher the error, the higher the number of points selected as constraints for the optimization.

In the first iteration, where the residual error still was not measured, an uniform sampling is used to select the constraints. A $n \times n$ mask, with step n, is scanned through the 2D projection of P_s at the xy coordinates. The point at the center of this mask is selected to be a constraint if it exists in P_s (i.e. it is not in a hole). From empirical tests, $n = 4$ produced the best results.

In the remaining iterations, we use the same $n \times n$ mask to perform a scan on the 2D projection of P_s and its residual error. First, the algorithm evaluates the average residual error at the $n \times n$ region being scanned. Based on the average error E_{avg} and a pre-defined threshold th_c, the number of points selected at that region will be defined.

Therefore, we select more constraints in the regions where the deformation is high and must be minimized, but we still consider the regions where the deformation is small or none, by selecting a small number of constraints to represent them.

3 Results and Discussion

In this section we analyse the performance and accuracy of the proposed algorithm and describe the experimental setup used. For all tests we ran our algorithm on an Intel(R) Core(TM) i7-3770 CPU @3.50GHZ, 8GB RAM, NVIDIA GeForce GTX 660. We have tested our algorithm in synthetic and real datasets with different levels of noise and precision.

As each dataset has its own minimum and maximum errors, we set the thresholds for adaptive node (th_n) and constraint selections (th_c) to be half of the averaged root mean squared error measured.

The average accuracy results can be seen in Table 1. In comparison with the ED algorithm, we could improve over 30% of accuracy by using adaptivity. Moreover, our approach is comparable to ED algorithm using the triple number of nodes (Figure 2). In terms of using adaptivity for constraint selection instead of uniform sampling with fixed step size, non-rigid registration achieves results as accurate as the ones obtained by using all the points from source object as constraints. Moreover, from the tests conducted, we need only three levels for the quadtree building and refinement to register two objects with an accuracy as good as the one obtained by related work.

	Acc. (mm)		Std. Dev. (mm)		Perf. (FPS)	
Dataset	Adap.	ED	Adap.	ED	Adap.	ED
Synthetic	0.58	0.83	0.46	0.6	18	6
Real	2.58	3.9	2.59	3.2	17	5

Table 1: Average accuracy (Acc.), standard deviation (Std. Dev.) and performance (Perf.) results obtained by our adaptive algorithm (Adap.) and the Embedded Deformation (ED) implemented in GPU for real and synthetic datasets.

As can be seen in Table 1, our approach achieves interactive frame rate. Moreover, it is up 3 times faster than ED algorithm. The use of adaptivity for constraint selection greatly reduces the processing time originally demanded by the ED algorithm. Optimization is a common bottleneck in non-rigid registration algorithms [3]. The number of constraints selected is directly related to the time required by the optimization phase. Therefore, by reducing adaptively the number of constraints used, we can achieve good performance even for the optimization phase. Moreover, as long as the error is minimized over the surface, the number of nodes is dynamically decreased from G. With less parameters to be estimated, the optimization algorithm converges faster.

4 Conclusion and Future Work

Here we present a fast method for non-rigid registration which is able to register two noisy point clouds captured from the Kinect sensor with high accuracy. We have extended and improved the Embedded Deformation algorithm by applying an adaptive strategy for distribution of nodes and selection of constraints. From the tests conducted, we have shown that the proposed algorithm is faster and more accurate than the original ED. For future work, we intend to incorporate to our approach optimized implementations for GPU processing.

Acknowledgements

We are grateful to the PCL project for providing the open-source implementation of the KinectFusion algorithm. This research is financially supported by FAPESB and CAPES.

References

[1] M. Macedo, A. Apolinario, A. C. Souza, and G. A. Giraldi. A Semi-Automatic Markerless Augmented Reality Approach for On-Patient Volumetric Medical Data Visualization. In *SVR*, 2014.

[2] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3DIM*, June 2001.

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