Scalable Non-Rigid Registration for Multi-View Stereo Data

Gianpaolo Palma[a], Tamy Boubekeur[b], Fabio Ganovelli[a], Paolo Cignoni[a]

[a] *Visual Computing Lab - ISTI CNR, Pisa, Italy*
[b] *LTCI, Telecom ParisTech, Paris-Saclay University, Paris, France*

Abstract

We propose a new non-rigid registration method for large 3D meshes from Multi-View Stereo (MVS) reconstruction characterized by low-frequency shape deformations induced by several factors, such as low sensor quality and irregular sampling object coverage. Starting from a reference model to which we want to align a new 3D mesh, our method starts by decomposing it in patches using a Lloyd clustering before running an ICP local registration for each patch. Then, we improve the alignment using few geometric constraints and finally, we build a global deformation function that blends the estimated per-patch transformations. This function is structured on top of a deformation graph derived from the dual graph of the clustering. Our algorithm is iterated until convergence, increasing progressively the number of patches in the clustering to capture smaller deformations. The method comes with a scalable multicore implementation that enables, for the first time, the alignment of meshes made of tens of millions of triangles in few minutes. We report extensive experiments of our algorithm on several dense Multi-View Stereo models, using a 3D scan or another MVS reconstruction as reference. Beyond MVS data, we also applied our algorithm to different scenarios, exhibiting more complex and larger deformations, such as 3D motion capture dataset or 3D scans of dynamic objects. The good alignment results obtained for both datasets highlights the efficiency and the flexibility of our approach.

Keywords: Non-ridig registration, Multi-View-Stereo, Low-frequency deformation, Scalable implementation

1. Introduction

The generation of 3D surfaces from measured data is a very important task in the acquisition and computation of complete digital representations of real-world objects. Among the various 3D scanning technologies, Multi-View Stereo (MVS) reconstruction from images appears as a very cost-effective solution: the ubiquitous and wide availability of cameras gives everyone the possibility of harvesting, in a short time and with inexpensive hardware, many images to use for the 3D MVS reconstruction of the world around us. Furthermore, the rise and the consolidation of the community photo collections have increased considerably the amount of data available for this purpose ([1], [2], [3]). The 3D models obtained with these technologies can be used for different applications, among which multimodal capture is becoming more and more frequent. In this context, the MVS data can be used for two different purposes: the 3D model completion or the temporal environment monitoring. In the first case, the goal is to enrich an existing 3D model obtaining a complete sampling (for example to integrate and complete a high-quality 3D scan of a monument with the missing parts that are easier to acquire by photogrammetry with a drone). In the second case, the goal is to monitor the temporal shape evolution of an environment or an object automatically comparing a pair of 3D models acquired at different points in time. For both applications, a fundamental step resides in the registration of the computed MVS mesh to a reference model (e.g., a high-quality 3D model, a laser scan or another MVS reconstruction). Two separate issues hinder this goal. The first one is the estimation of the unknown scale factor of the new MVS mesh with respect to the reference model. A solution for this issue was proposed in [4] for general 3D models and in [5] when the input can be approximated with a height map. The second issue is the low

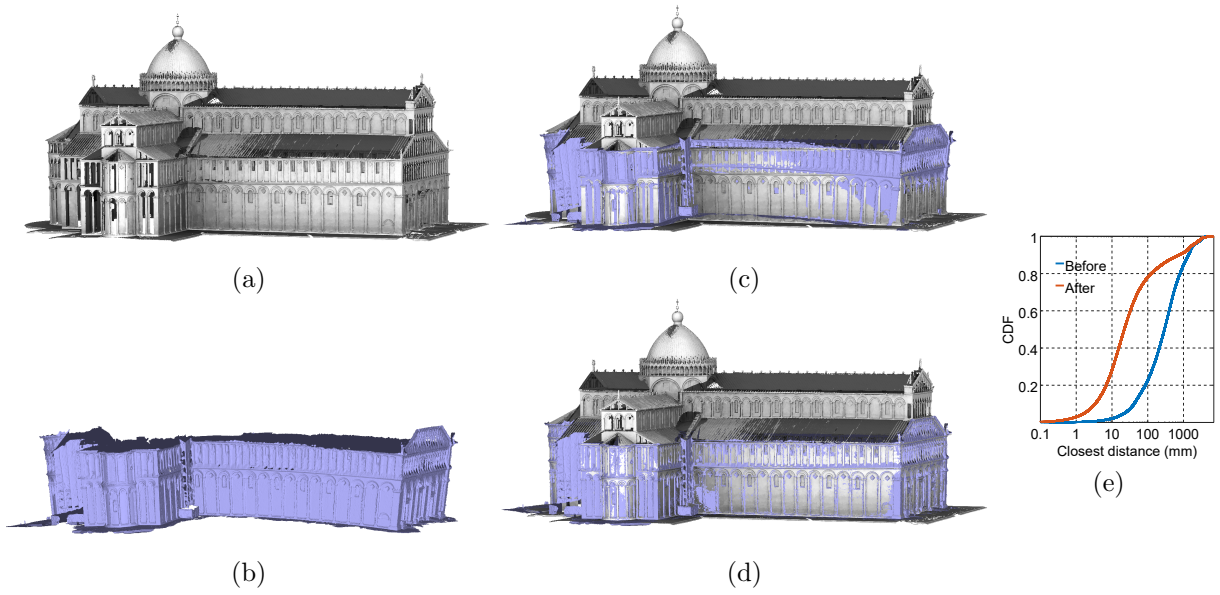


Figure 1: Our algorithm deforms a 3D model obtained with multi-view stereo (MVS) methods to align it over a reference mesh. (a) Input reference mesh by 3D scanning. (b) Input deformable mesh by MVS reconstruction. (c) Initial rigid alignment. (d) Final alignment computed by our algorithm. (e) Cumulative Distribution Function (CDF) of the vertex-to-mesh distance between the deformed mesh and the reference one (blue before and orange after the alignment). Our algorithm corrects the bending introduced by the MVS reconstruction along the nave of the cathedral, preserving the significant geometric changes like the scaffolding on the side of the building. This is also confirmed by the better CDF of the closest distance after the alignment.

frequency deformation introduced in the model by the Structure-From-Motion (SfM) step of the MVS reconstruction (see the input MVS model in Figure 1) and related to several factors: (i) the low sensor quality; (ii) the sampling acquisition coverage of the objects in term of overlap among the images; (iii) the spatial distribution of the views; (iv) the parameters of the involved algorithms that are not able to remove all the optical lens distortion or to detect a sufficient number of features to constrain the bundle adjustment step. Even if some of these problems can be controlled in the case of an ad-hoc photographic campaign, for example taking care of the sampling acquisition coverage or using expensive devices like a total station theodolite to acquire a number of Ground Control Points, the situation is worse in the case of community photo collections where the acquisition is incidental and the photos are acquired with different cameras.

In this paper, we propose an iterative and hierarchical non-rigid registration method to align a large MVS mesh to a reference 3D model while removing the deformation introduced by the MVS algorithm. The final goal is to correct the deformation in a relative manner since the reference model continues to have its own distortions. This relative alignment can be very useful for the applications that can take advantage of an acquisition with inexpensive hardware, like a camera, where the relative deformation of one model to the previously acquired one permits or to integrate the data of the two models or to detect the changed regions in a more robust way. Starting from a rough alignment to the reference mesh, at each iteration, our algorithm segments the model into patches using a Lloyd clustering. The resulting segmentation is used to build a deformation graph that allows transferring the affine transformation computed for each node of the graph onto the original model. The affine transformations are computed independently for each patch with an Iterative Closest Point (ICP) procedure, using the geometry of the adjacent patches to constrain the local stiffness. This local ICP procedure makes the algorithm easier to parallelize on multicore architectures, allowing, for the first time, the alignment of meshes with tens of millions of triangles in few minutes. When the ICP converges to a local minimum, we improve the estimation by regularizing the

local consistency of the transformation from the adjacent nodes. All along the iterations, we increase the number of patches of the Lloyd clustering to progressively fit smaller scale deformations.

Our main contribution is a non-rigid registration algorithm that allows processing very large and complex MVS meshes. We do not make any assumptions about the type of the input data and the amount of deformation between the models, as opposed to the state-of-the-art non-rigid registration algorithms that manage specific input, like range scan, taking advantage from their implicit 2D parameterization (for example [6] and [7]), or range video, taking advantage of the temporal coherence of the data (for example [8] and [9]). These assumptions prevent the trivial application of these solutions to our kind of input, and their adaptation to our input is not straightforward. Other state-of-the-art solutions are based on complex global non-linear energy minimization [10], which do not scale with the size of the input models. On the contrary, our solution is based on a localized ICP and local updates of the deformation model, making the algorithm easier to run in parallel and scalable to large 3D meshes. In particular, our algorithm is robust against all usual defects exhibited by a typical MVS mesh such as noise, missing geometry, irregular triangulation or irregular density. It can also handle multi-scale input with very different level-of-details in the same mesh and between the target and the deformed mesh. Finally, our method is also robust to the presence of geometric changes, like pieces of new geometry that did not exist in the reference mesh, preserving these changes (Figure 11). This property is key for the detection of changes in evolving 3D data where MVS techniques offer a practical cost-effective solution. We tested the algorithm with different real datasets using reference models of different quality, such as a high-quality 3D scanned model, a single raw LIDAR scan or another MVS model. Furthermore, we tested different scenarios that exhibit more complex and larger deformations, such as 3D motion capture datasets or dynamic object scans, showing that, although specifically designed for MVS data, our method achieves comparable results with a state-of-the-art algorithm on a broader set of application scenarios.

2. Related Work

The deformations in an MVS reconstruction are mainly due to the SfM step that introduces drifting effects due to different reasons (low sensor quality, not good view sampling of the scene, the parameters of the involved algorithms). Cohen et al. [11] propose a new SfM formulation to solve this problem in the case of architectural scenes with symmetric or repeated structures. Although this SfM method returns a more natural coordinate system, reducing the final deformation, the input requirements prevent its application to more general scenes. On the contrary, we formulate the problems as a non-rigid registration using another 3D mesh as reference.

The literature of the 3D registration problem is very wide and in this section, we analyze the approaches more closely related to the proposed method. Our starting point is the Iterative Closest Point [12] algorithm, which serves as a basis for many others. It estimates the rigid transformation between two shapes by minimizing the point-to-point [12] or the point-to-plane [13] distance of a set of correspondences created by closest search. The procedure iterates until convergence. To reach convergence to the optimal transformation, ICP and its variants [14] require a coarse alignment step between the two models. This initial rough alignment can be obtained by manually picking at least four correspondences between the two meshes or by using automatic methods based on the explicit extraction and matching of 3D key-points [15], the extraction of quadruple of congruent quasi-coplanar points ([16],[4]), or the combination of both [17]. For a complete overview of these methods we refer the reader to the work of Diez et al. [18].

When allowing deformation, the mapping is not encoded as rigid transformation, but as a generic function that should ideally mediate between an exact mapping of the shapes and a rigidity-preserving mapping. To correct low frequency warps on range scans, Ikemoto et al. [19] decompose the input scans using a coarse-to-fine hierarchy of locally rigid pieces that are aligned independently with a global registration procedure. They use a very simple splitting along the longest oriented bounding-box direction, keeping separated and independent the pieces without any final deformation of the original scans. A more efficient solution is presented in [20] where a locally weighted ICP provides the point correspondences among the range scans over which to define a Thin Plate Spline warping that corrects small misalignments.

Several solutions model the deformation between a pair of range scans as a weighted combination of

local rigid transformations. Li et al. [6] globally optimize the rigid transformations and the correspondences for a set of uniformly distributed vertices connected with a deformation graph. The optimization minimizes a global energy function that guarantees smoothness and rigidity among near nodes in the graph. Huang et al. [21] propose the matching of explicit 3D feature points to create correspondences that can be used to extract rigid clusters, points whose deformation can be described by a single rigid transformation assuming isometric deformation. Chang et al. [7] use a reduced deformable model to optimize simultaneously the rigid transformation and the influence weight to transfer the deformation using a grid-based representation. Many other techniques solve the registration using the temporal coherence of a dense stream of depth maps [22, 23, 8], or explicit markers [24] or a template model to fit the range scan data [25, 26, 27]. A new reconstruction method from range video was proposed by Zhou et al. [9], by partitioning the input stream into small fragments of few consecutive frames that are deformed using a volumetric registration formulation. Whelan et al. [28] developed a real-time dense SLAM using an RGB-D camera incremental acquisition to build consistent surfel-based maps of room scale environment. This is accomplished by using dense frame-to-model camera tracking and windowed surfel-based fusion with frequent model refinement through non-rigid surface deformations. Duo et al. [29] proposed a new pipeline for live multi-view performance capture using RGB-D cameras that combines the volumetric fusion with the estimation of a smooth deformation field across the RGB-D views. With respect to these solutions, our method works on general large meshes and not only on range scan and range video, without any assumption on the type of input.

Similar to our method, Cagniart et al. [10] propose a generic data-driven mesh deformation framework based on a subdivision of the original surface into patches with a hierarchical approach increasing the patch resolutions. Unlike from our method, which uses a local affine ICP, they propose a global energy minimization over all the vertices to compute the transformation of each patch guaranteeing the local rigidity of the mesh. A similar Laplacian-based deformation was proposed by Budd et al. [30] for the non-sequential alignment of multiple unstructured mesh sequences from non-rigid surface capture. Even if these papers do not present any performance analysis and data information, usually the global energy approach does not scale with the size of the input mesh, especially with tens of millions of triangles. On the contrary, our solution scales better with the size of the meshes thanks to the local ICP approach that naturally runs in parallel on multicore architectures. Rouhani et al. [31] propose the non-rigid registration of point clouds where the source is clustered in small patches and the target is approximated with an implicit representation. In this way, the association between the source and the target can be replaced by a continuous distance field that reduces the energy to minimize to a system of linear equations. Yang et al. [32] observe that the similarity among transformations is better captured by L^1 norm than by the widely used L^2 norm, which tends to overweight outliers, and they propose a sparse no-rigid ICP method. Both the methods use a hierarchical multiresolution approach to prevent the minimization from getting stuck in a local minimum.

In the context of the surface similarity, Bronstein et al. [33] address the problem of the partial matching between a template model and another surface with a multi-dimensional scaling algorithm that computes the mapping between the two surfaces minimizing the distortion. On the assumption of isometric deformation Bronstein et al. [34] introduce a diffusion distance robust to topological noise, Jain et al. [35] use correspondences in the spectral domain and Litman et al. [36] propose a scheme for learning optimal spectral descriptors. Chen et al. [37] showed how to model the minimization of this intrinsic distortion function with a Markov Random Field optimization, increasing the registration precision. Typically, all these solutions are less robust to noisy data, which is a typical defect of MVS reconstructions. Recently a very specialized solution was proposed by Wei et al. [38] to use deep learning methods for finding correspondences among 3D scans of humans.

3. Algorithm Overview

Given two triangular meshes A and B, the goal of our algorithm is to align B over A when B exhibits a deformation that cannot be modeled with a single affine transformation. Our strategy follows and extends the classic procedure presented in [6] which is based on the computation of a deformation graph of B, the estimation of an affine transformation for each node of the graph and the transfer of the deformation from the graph to the original mesh. We

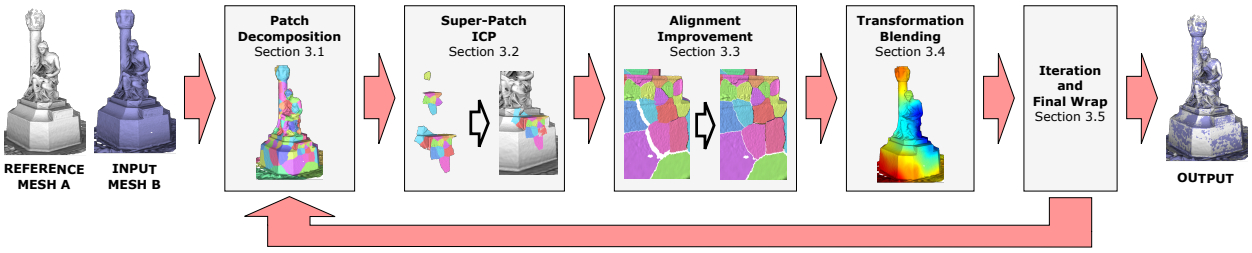


Figure 2: Algorithm overview.

propose a new scalable pipeline that is able for the first time to align meshes with tens of millions of triangles in few minutes.

Starting from an initial estimation of a rough alignment of the two meshes, our method can be decomposed into four steps (see Fig. 2):

1. the decomposition of B into patches with a Voronoi clustering to build the topology of the deformation graph (Section 3.1);
2. the registration of each patch with the mesh A using a local ICP algorithm (Section 3.2);
3. the collaborative improvement of the registration for all the patches with few geometric constraints (Section 3.3);
4. the smooth reconstruction of a transformation field from the finite set of estimated transformations at the nodes of the deformation graph, to compute the final per-vertex deformation of B (Section 3.4).

The initial rough alignment can be obtained automatically using [4] or [39] if the two meshes were created from multi-modal data and present large-scale differences. The procedure is iterated using as input the output of the previous iteration until convergence. After the first iteration, we increase the number of the nodes in the deformation graph in order to create more patches and capture smaller deformations. Last, when the procedure converges, we compute a final warp to recover the registration of the smallest features (Section 3.5).

3.1. Patch Decomposition

The patch decomposition procedure uses a simple Lloyd's algorithm to compute a Voronoi clustering $\mathcal{P}$ (see left image in Figure 3) of the mesh B which is adaptive to the mesh density: in the areas with more triangles we create more patches assuming that it indicates more geometric details. This adaptivity is particularly important for the meshes created from a multi-scale input with a very different level of details. Starting from a user-defined target number m of patches, we use an adaptive octree to distribute n initial seeds for the segmentation. Each leaf of the octree must contain at least v/m vertices with v the number of vertices of B. The seeds are initialized with the vertices of the mesh which are the closest to the center of the octree leaves. The procedure continues with the two classic steps of a Lloyd iteration:

1. the computation of the Voronoi diagram of the current seeds according to an appropriate metric L taking account of the mesh topology;
2. the optimization of the seed of each patch.

In the first step, we assign each vertex to the nearest seed according to the geodesic metric L computed using the mesh topology. In Section 4.3 we analyze the performance of the algorithm with seven different metrics L varying the number of Lloyd iteration. For the last steps, the idea is to assign as seed the centroid of the patch, that is the point that minimizes the sum of the square distances from the vertices of the patch. To approximate this point, we compute the quadric using all the vertices of the patch and we select the vertex that maximizes this quadric. To make the computation independent from the local triangulation, the contribution of each vertex to the quadric is weighted by its Voronoi region area defined by the local triangulation using the method proposed in [40].

Once the clustering is computed, we build the deformation graph $\mathcal{G}$ of the mesh B as its dual graph, where its vertices g_i are the nearest vertices of the mesh to the final centroid of each patch and the edges capture the topological adjacency among the patches using the topology of the mesh B (see right image in Figure 3). In order to guarantee the stiffness of the transformation between the different connected components of the mesh B, we force the construction of a connected deformation graph

by building a minimum spanning tree between the connected components of $\mathcal{G}$. Each new edge in the spanning tree simply connects the nearest nodes of two components of the graph.

3.2. Super-Patch ICP

Starting from the Voronoi segmentation $\mathcal{P}$ and the deformation graph $\mathcal{G}$, we compute an affine transformation for each node of the graph independently using an ICP approach. More precisely, we associate to each node a super-patch, composed at least by the patch of the node and its 1-ring adjacent patches. Then, we grow the super-patches radially until its geometry reaches a good distribution of normals in the Gauss Map. We approximate the Gauss Map with a spherical histogram of 40 seed normals generated with the Spherical Fibonacci point sets [41]. This growing procedure is stopped when the solid angle subtended by the biggest triangle defined by the most populated bin and two other non-empty bins of the histogram is greater than $\pi/2$. We discard the bins containing a number of elements less than 0.1% of the most populated bin. This growing procedure avoids the creation of too flat super-patches that could present too few geometric anchors for the convergence of the following ICP algorithm. Moreover, the redundancy of geometric data in the super-patches partially constrains the local stiffness between adjacent nodes.

Then, we compute the ICP between each super-patch and the mesh A. Our procedure extracts a subset of points p_i from the super-patch using a uniform distribution of normals n_{p_i} and seeks the nearest points q_i with normal n_{q_i} in the mesh A for each of them. We discard all the correspondences that are distant ($|p_i - q_i| < d$) and have inconsistent normal ($n_{p_i} \cdot n_{q_i} < \cos(\pi/4)$). These heuristics are the most used and robust for the ICP algorithm [14]. With the resulting k good correspondences, we estimate the affine transformation T by minimizing the classical point-to-plane error metric [13]:

$$E = \sum_{i=0}^{k} \| (T p_i - q_i) \cdot n_{q_i} \|^2 \tag{1}$$

The maximum distance d for the correspondence rejection is set equal to the minimum between the maximum distance of the super-patch from A and a user selected distance expressed or as a percentage of the bounding box of B or as a percentile of the vertices of B. The percentile is useful when the two meshes contains strong differences, such as no sampled areas or completely new geometry, to indicate the amount of change in B.

The estimated affine transformation, composed by rotation, translation and uniform scale, is stored at each node of the deformation graph (see top image in Figure 4). In particular, the uniform scale is required to correct small scale differences between the multi-view stereo geometry and the reference mesh A that the initial rough alignment, obtained also with the most robust existing solution [4], was not able to correct.

3.3. Alignment Improvement

To evaluate the quality of the transformations computed for each node of the deformation graph, we use the metric proposed by Gelfand et al. [42]. The goal is to detect when the ICP converges to a local minimum due to a set of correspondences not able to constrain unstable transformations, producing the sliding between the surfaces. With the points p_i of the correspondences used in the ICP procedure, we create the covariance matrix $C = FF^T$ with

$$F = \begin{bmatrix} p_1 \times n_1 & \cdots & p_k \times n_k \\ n_1 & \cdots & n_k \end{bmatrix} \tag{2}$$

and we compute its condition number $c = \lambda_{max}/\lambda_{min}$ as the ratio between the largest and the smallest eigenvalues of C. We consider as valid the transformations of the nodes with $c \leq t$. Based on our experiments, we set the value $t = 5$ for all the datasets.

For the nodes with an unstable transformation ($c > t$), or where the ICP failed due to few correspondences, our algorithm adopts a relative optimization strategy to improve the affine transformation quality. At this stage, the key idea is to propagate the transformation from the valid nodes with an advancing front approach following the topology of the deformation graph $\mathcal{G}$. The order depends on the geodesic distance of the invalid patches from the nearest valid patch in the graph (see the bottom image in Figure 4). Starting from the closest ones, all the patches with the same distance are processed in parallel at the same time. During the propagation, we minimize the registration error with respect to the valid transformation of the neighbors. In the first step, we take 100 uniform random vertices s of the patch $\mathcal{P}_i$ and we compute the weighted average position s' of each one using the transformations T_j

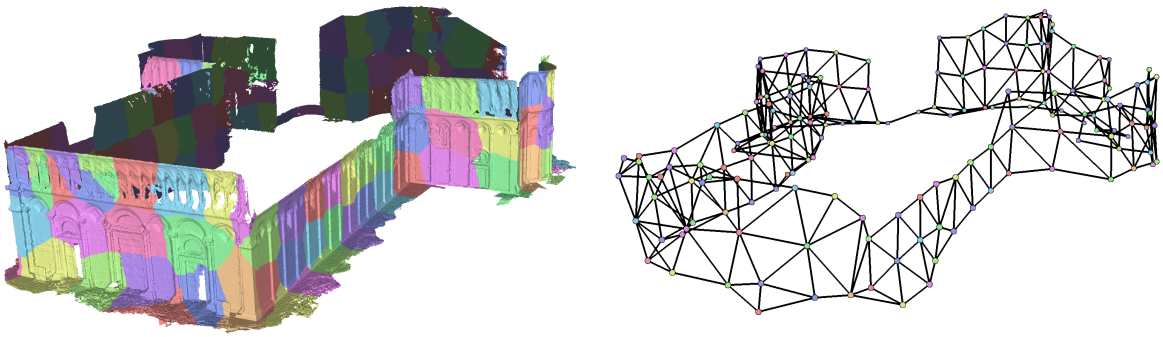


Figure 3: (Left) Voronoi clustering of the CATHEDRAL1 dataset. (Right) Deformation graph computed as dual graph of the Voronoi clustering.

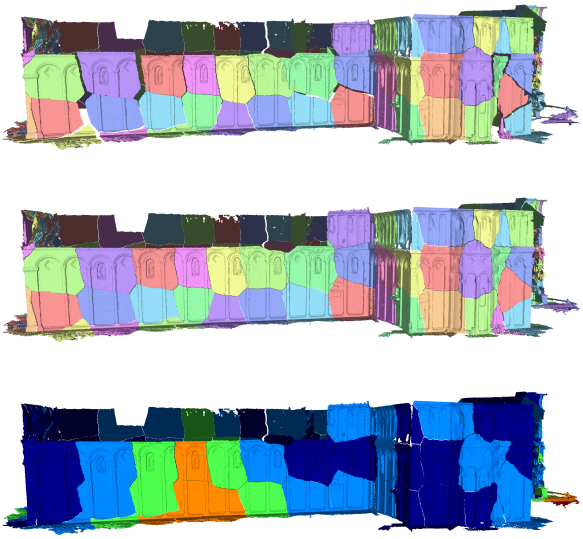


Figure 4: (Top) Patch-wise transformation computed by the procedure in Section 3.2. (Center) Improvement of the patch-wise transformation computed by the procedure in Section 3.3. (Bottom) Color coding of the geodesic distance of the invalid patch from the valid one (blue=valid patch, azure= distance 1, green=distance 2, yellow=distance 3). This distance is used to set the order of propagation of the valid transformations described in Section 3.3

of the 1-ring valid patches $\mathcal{N}(i)$:

$$s' = \frac{\sum_{j \in \mathcal{N}(i)} f_{ij}(T_j s)}{\sum_{j \in \mathcal{N}(i)} f_{ij}} \tag{3}$$

The weight function f_{ij} is defined as:

$$f_{ij} = b_{ij} \exp^{-c_j/2t} \tag{4}$$

where b_{ij} is the number of border edges between the patch $\mathcal{P}_i$ and the patch $\mathcal{P}_j$ and c_j is the condition number of the patch $\mathcal{P}_j$. This weight function gives more influence to the neighboring patches with a transformation computed with a higher quality (lower c values) and with a longer shared boundary with the patch $\mathcal{P}_i$. With the pairs of points $\langle s, s' \rangle$, we compute the new transformation T' for the patch $\mathcal{P}_i$ that minimizes the point-to-point error metric:

$$E = \sum_{i=0}^{100} \|Ts - s'\|^2 \tag{5}$$

This new transformation is refined with the ICP procedure described in Section 3.2 applied to the patch $\mathcal{P}_i$ and the mesh A. At the end of the ICP refinement, we recompute the quality c of the new estimated transformation and we mark the patch as valid to allow the propagation of its transformation to the nearest non-valid patches. This forward propagation from the valid patch is followed by a back propagation of the computed alignment to guarantee a smoother deformation in the critical areas, especially in the regions with strong changes. In this back propagation, we follow the reverse order used in the forward propagation starting with the farthest originally invalid patches. In particular, for each of these patches, we estimate the transformation that minimizes the metric in Equation 5 using the pairs of points $\langle s, s' \rangle$ computed with all the 1-ring patches. The middle image in Figure 4 shows the registration improvement obtained by applying this procedure.

3.4. Transformation Blending

To smoothly transfer the deformation from the graph $\mathcal{G}$ to the mesh B, we use the method proposed by Sumner et al. [43]. For each vertex b_i, we

compute its deformed position $\tilde{b}_i$ as the weighted mean of its positions after the application of the affine transformations T_j of the k-nearest nodes of the deformation graph to the patch to which the vertex belongs:

$$\tilde{b}_i = \frac{\sum_{j=0}^{k} w_j(b_i)[T_j b_i]}{\sum_{j=0}^{k} w_j(b_i)} \tag{6}$$

The k-nearest nodes are selected using the graph topology. Doing so, our approach is more robust in the case of close structures with opposite orientation and not topologically connected where the relative transformations are not influenced (for example the sides of a wall that must be moved in the opposite direction). We define the weights w_j as:

$$w_j(b_i) = (1 - \|b_i - g_j\| / d_{max})^2 \tag{7}$$

where d_{max} is the distance to the k+1-nearest node from the vertex. In all our experiments we use $k = 8$, the best value proposed by Sumner et al. [43] to avoid interpolation artifacts.

Using the same interpolation procedure, we compute two scalar fields to measure for each vertex the amount of displacement $\mathcal{D}$ and local flexibility $\mathcal{S}$ starting from the value computed for each node of the graph. The displacement field evaluates the amount of movement of each vertex of the mesh after the application of the computed deformation graph. The local flexibility gives a measure of how much locally similar are the final deformations computed for each vertex. For each node g_i of the graph, the displacement value $\mathcal{D}(g_i)$ is computed as the average distance between the vertices a_j of the bounding box of the relative patch before and after the estimated transformation T_i:

$$\mathcal{D}(g_i) = \sum_{j=0}^{7} \|T_i a_j - a_j\| / 8 \tag{8}$$

The use of the bounding box vertices allows a better estimation of the amount of rotation introduced by the computed transformations. The local flexibility $\mathcal{S}(g_i)$ is computed as:

$$\mathcal{S}(g_i) = \frac{\sum_{j \in \mathcal{N}(i)} \|T_j g_i - T_i g_i\|}{\#\mathcal{N}(i)} \tag{9}$$

where $\mathcal{N}(i)$ is the set of neighboring patches of $\mathcal{P}_i$. Figure 5 shows the scalar fields computed after the first iteration of the algorithm on a performance capture example.

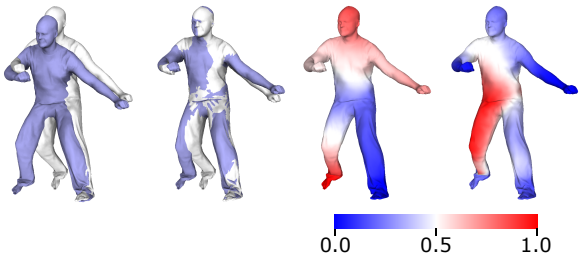


Figure 5: From left to right: initial alignment; alignment after the first iteration of the algorithm; the normalized displacement scalar field $\mathcal{D}$ computed with Equation 8; the normalized local flexibility scalar field $\mathcal{S}$ computed with Equation 9.

3.5. Iteration and Final Warp

Although the first execution of the proposed algorithm is able to correct most of the deformation between the two meshes, this procedure is iterated using as input the output of the previous iteration. After the first iteration, we evolve two parameters of the algorithms assuming that the mesh B has been moved to a closer position with respect the mesh A in order to capture smaller deformations. In particular, we double the target number m of nodes of the deformation graph to create smaller patches and we halve the maximum distance for the rejection of the point pairs to use in the ICP procedures. We detect convergence when, for all the nodes g_i of the deformation graph, the displacement value $\mathcal{D}(g_i) < 2e_i$, where e_i is the mean length edge of the patch of the node. Experimentally, we see that few iterations are sufficient to achieve good results.

Finally, after convergence, we compute a final warp to recover the registration of the smaller features. This warp is obtained by running a simplified version of the proposed algorithm without the step described in Section 3.3. In particular, we extract the Voronoi segmentation, we compute the ICP for each patch bounding the creation of super-patches to the only 1-ring adjacent patches and finally, we blend the estimated transformations. In this final warp step, we quadruple the target number m of nodes of the deformation graph used in the last iteration of the algorithm.

4. Results

We tested the method with two different datasets (Section 4.1): a set of MVS meshes using models of different quality as reference (high-quality 3D

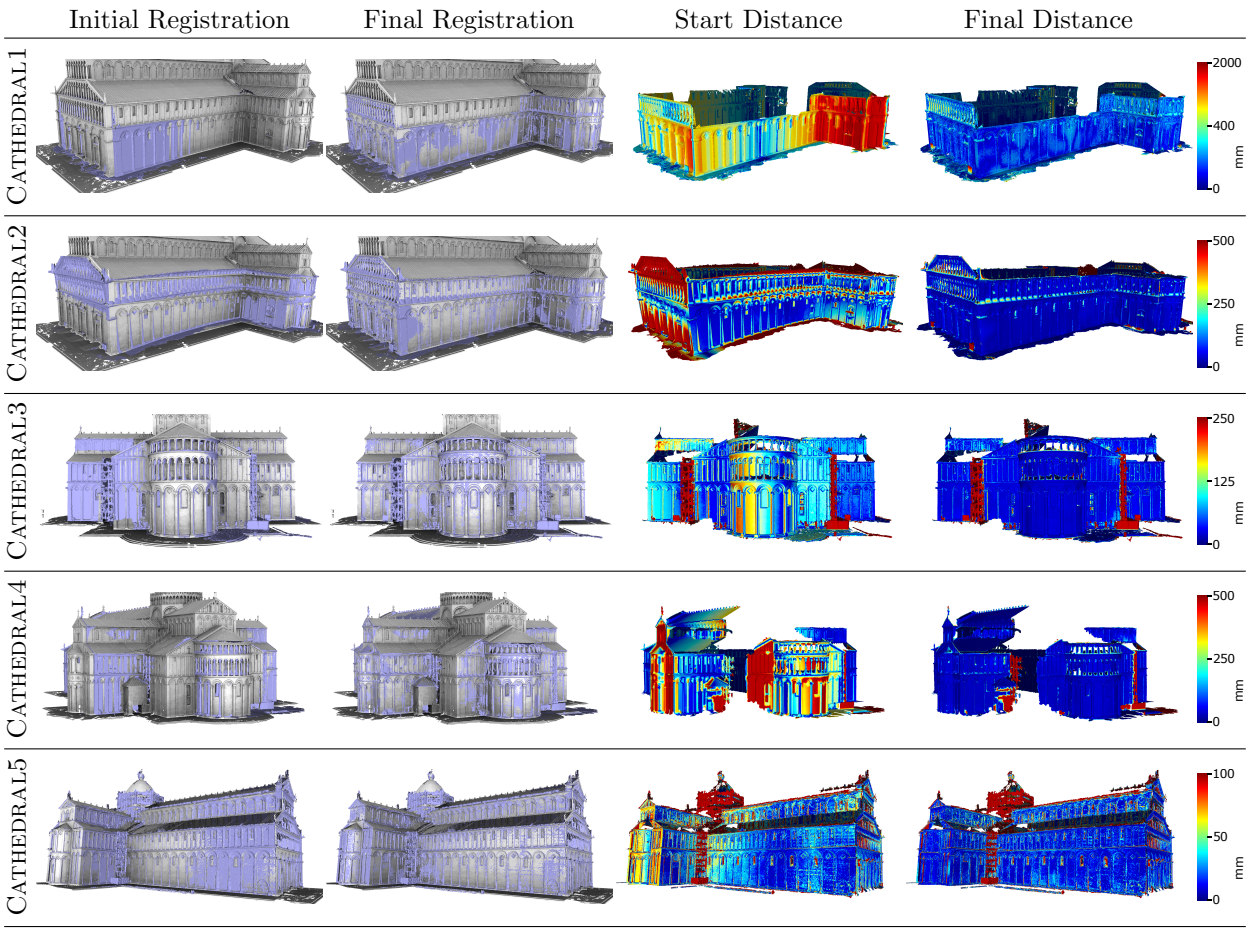


Figure 6: Registration results obtained for the dataset Cathedral using a high quality 3D model as reference mesh. For each test we show the starting alignment, the computed final registration and the color mapping of the distance from the reference mesh after and before our algorithm. The MVS models were created with Agisoft Photoscan (Cathedral1 and Cathedral3), COLMAP (Cathedral2 and Cathedral4) and MVE (Cathedral5).

scanned models in Figure 6, a single raw LIDAR scan in Figure 7, another MVS model in Figure 8); a set of more complex test cases made by 3D motion capture and 3D scanning of dynamic objects. Then we analyze the parallel performance of our multi-core implementation (Section 4.2) and the influence of the Voronoi clustering metrics to the registration results (Section 4.3), which is the most crucial step since it determines the amount and the type of deformation that we correct. For all the other input parameters of the algorithm we always used the same values, the best ones reported by the original papers.

4.1. Dataset

The MVS dataset is composed of four scenes (Cathedral, Arena, Excavation, Square), each one with several models reconstructed with three different MVS software: Agisoft Photoscan; Multi-View Environment (MVE) [44]; COLMAP [45] which is one of the best MVS reconstruction software according the recent benchmark in [46]. Cathedral is composed of 3 different photographic campaign acquired with different cameras. The first one was reconstructed from some frames of a video acquired with an action cam using both Photoscan (Cathedral1) and COLMAP (Cathedral2). The second one was acquired with a reflex camera and reconstructed with Photoscan (Cathedral3) and COLMAP (Cathe-

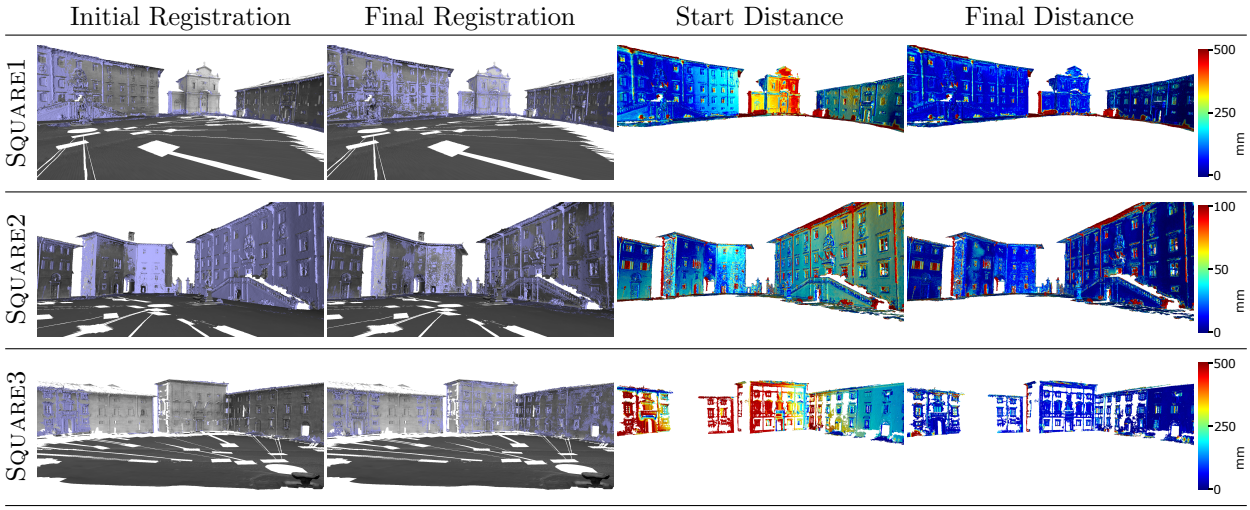


Figure 7: Registration results obtained for the dataset SQUARE using a single LIDAR scan as reference. For each test we show the starting alignment, the computed final registration and the color mapping of the distance from the reference mesh after and before our algorithm. The MVS models were created with Agisoft Photoscan (SQUARE1 and SQUARE2) and MVE (SQUARE3).

DRAL4). The third one was acquired with a reflex camera with a better view sampling and reconstructed with MVE (CATHEDRAL5). For this scene, we used a high-resolution Time-of-Flight 3D scanned model as reference for the registration. Similarly, SQUARE is composed of 3 MVS models: SQUARE1 acquired with an action cam and reconstructed with Photoscan; SQUARE2 acquired with a reflex camera and reconstructed with Photoscan; SQUARE3 acquired with a reflex camera and reconstructed with MVE. The reference mesh is a single and partial raw triangulated Time-of-Flight scan. ARENA is composed of 3 MVS models acquired by multiple people in an incidental way with different cameras and reconstructed with MVE. We used the first MVS model as reference for the registration of the others (ARENA1 and ARENA2). This dataset shows irregular density with very different level-of-details in the same mesh and among the models. EXCAVATION is composed of 3 MVS models of an archaeological excavation acquired with two different cameras at distance of years (2013, 2014, 2016) and reconstructed with COLMAP. The first MVS model is used as reference for the others. For each test, the initial position for the non-rigid registration was obtained with a rigid alignment procedure. In particular, we computed a coarse alignment by manually picking at least four correspondences between the models to compute the affine matrix that minimizes the least square error. Then the registration was refined by a rigid ICP. Only for the dataset ARENA, we computed the initial registration with the procedure proposed by Mellado et al. [4], to estimate automatically the scale difference between the models. All the tests present some deformations, with differences in the amount and the type of distortions, and show some geometric inconsistency in the areas where the MVS algorithm fails. All the tests contain some consistent temporal changes with the reference, like no sampled area or completely new geometry. Figures 6, 7 and 7 show the obtained results. Figures 13, 14 and 15 show closeups of the color mapping of the distance to the reference mesh before and after the non-rigid-registration. Table 1 contains the data of each test reporting the size of the input meshes, the number of nodes of the deformation graph used in the different steps, the number of iterations of the algorithm, the computation time, and the Root Mean Square error (RMS) computed on the distance of each vertex of the reference model from the closed point of the deformable mesh before and after our algorithm. We performed our test on a PC with an Intel Core i7-4790k 4.00GHz and 32 GB RAM.

In general, our method significantly improves the initial registration, as we can see in the color mapping of the distance from the reference mesh (Figures 6, 7 and 7). This is confirmed also by the RMS values in Table 1. The method corrects strong deformations, also of different type, due to the

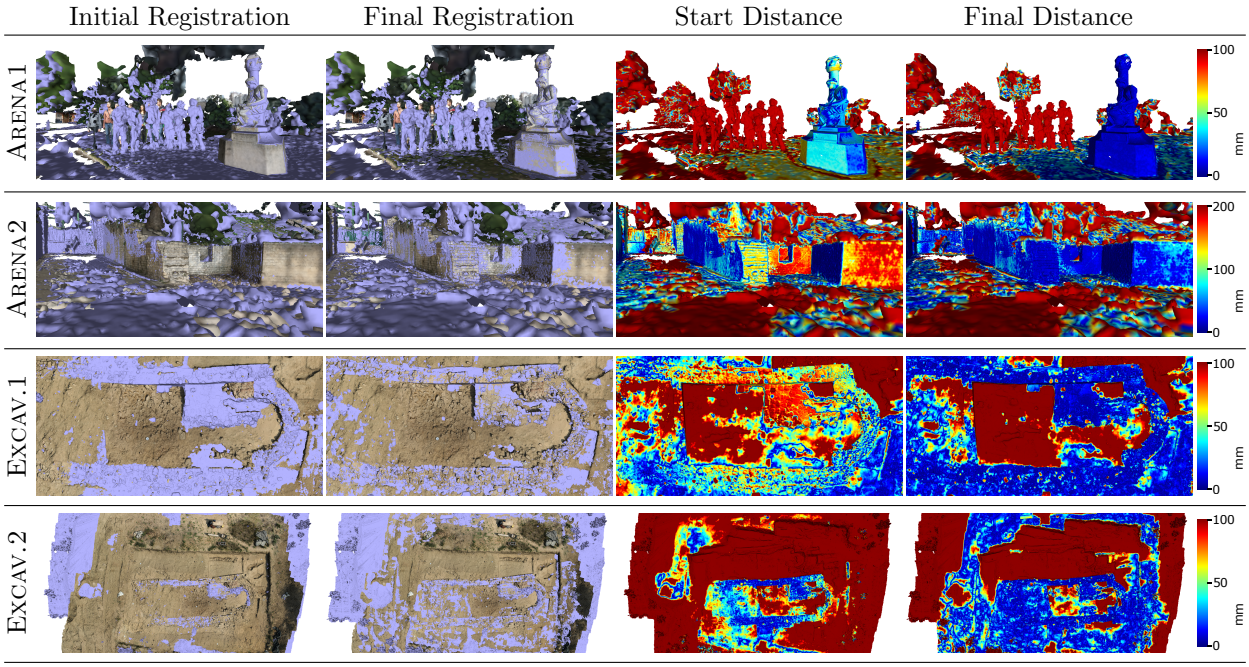


Figure 8: Registration results obtained for the datasets ARENA and EXCAVATION using a MVS model as reference. For each test we show the starting alignment, the computed final registration and the color mapping of the distance from the reference mesh after and before our algorithm. The ARENA models were created with MVE while the EXCAVATION models were created with COLMAP.

Structure-from-Motion step, like in CATHEDRAL1 and CATHEDRAL4, where some not aligned images do not permit to close the loop around the cathedral producing an incremental drifting, or in CATHEDRAL1, where the low quality of the input images generates a bending along the nave of the cathedral. It also manages input models with a very bad and partial triangulation like in SQUARE3. For all the tests, the method takes some minutes to compute the registration of huge models with several million of triangles. The higher time in the tests SQUARE3 and ARENA2 is due mainly to the Voronoi clustering biased by the irregular and partial triangulation of the models that present very long and narrow connected components.

We also tested the robustness of the method on different scenarios that exhibit more complex and larger deformation. In particular, we used some cases from the motion capture datasets published in [47] (DANCER), in [48] (FACE), and the TORSO dataset from [6]. To manage these larger deformations, we modified our algorithm for the dataset FACE and TORSO using a preprocessing step. In this step, we iterate our procedure until convergence using a low and fixed number of patches and with a less strict convergence criterion ($\mathcal{D}(g_i) < 10e_i$ for each node of the deformation graph). The mesh obtained at the end of the preprocessing is used as input to our standard procedure. This step permits to correct the low-frequency general deformation of the models, avoiding the estimation of wrong transformations. Although for this kind of data there exist more appropriate and specialized algorithms, the obtained results show the flexibility of our method (Figure 9 and Table 1). The method can correct the general deformation of the shape, but it does not recover very small deformations, like the high-frequency changes of the dress in the DANCER or the smallest features of the facial expression in the FACE dataset. Anyway, our method obtains comparable results with Li et al. [6], as shown in Figure 10, starting from the same initial conditions. This is confirmed also by the RMS values (our method RMS = 0.501, Li et al. [6] RMS = 0.498).

Furthermore, the method is robust to the presence of significant geometric changes in two different ways. First, it preserves new geometry appear-

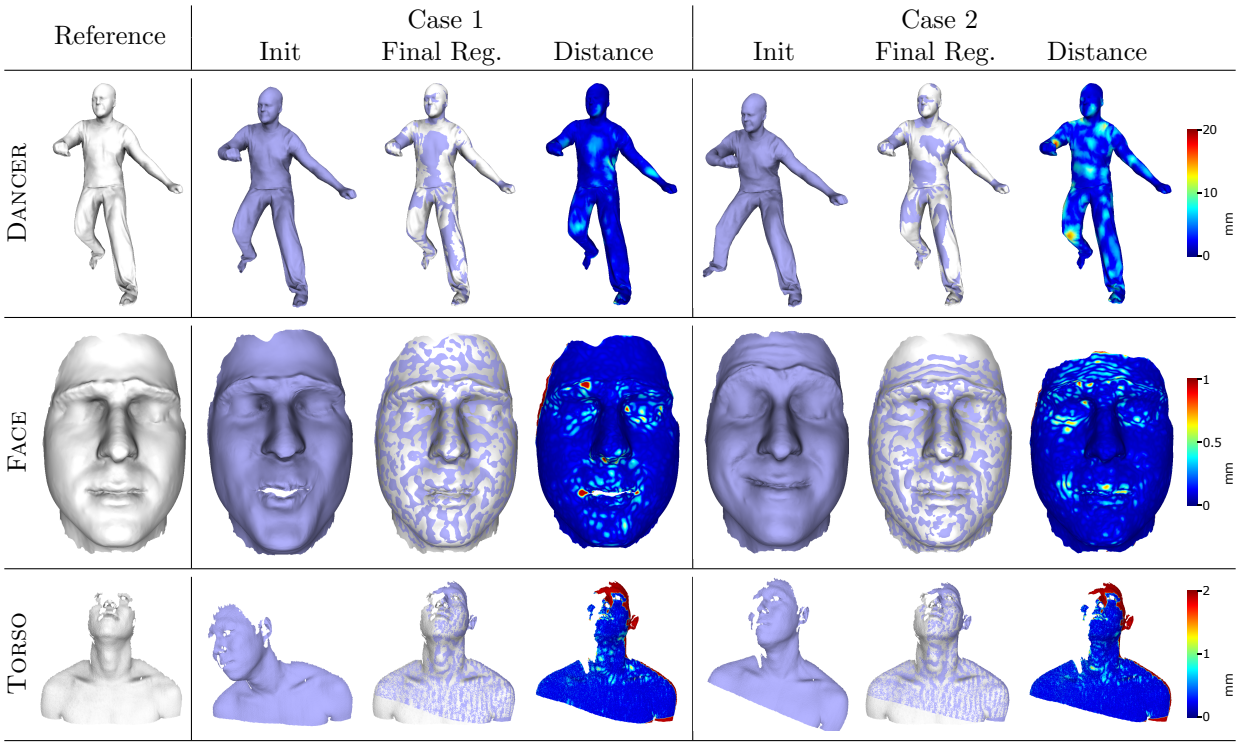


Figure 9: Dynamic dataset. For each dataset, we show the reference mesh and two test cases with the initial alignment of the input mesh B, the output registration and the color mapping of the distance from the reference model. For the Torso dataset, the starting alignment of the second test case is obtained after a rigid ICP on the input of the first case.

ing in the model while computing the deformation that registers the rest of the scene in a consistent way, like for the panel, the person and the scaffolding appearing in the Cathedral, shown in the first three columns of Figure 11. Similarly, in the Excavation dataset the changed regions in the middle due to the excavation process are well preserved (Figure 8). Secondly, it integrates the changes due to low-frequency deformation with small and locally rigid consistent movements like the bollard in the Square shown in the fourth column of Figure 11. This robustness is extremely important for the consistent monitoring of changes in 3D structure by 3D acquisition, an emerging field where MVS techniques offer a practical cost-effective solution.

4.2. Implementation

One of the main strengths of our algorithm is the use of a local ICP procedure to align each super-patch independently to the reference model. This naturally sets several components of the algorithm for parallel execution on a multicore architecture. In particular, our implementation uses

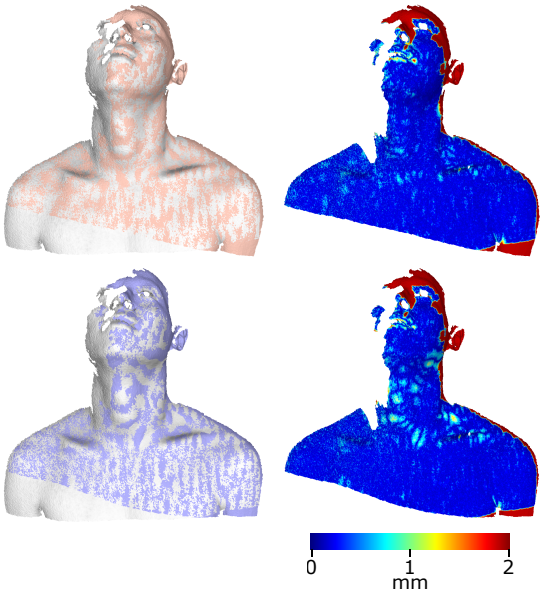


Figure 10: Comparison of the result obtained with our method (bottom) with Li et al. [6] (top) on the Torso dataset. The color mapping shows the distance from the reference mesh.

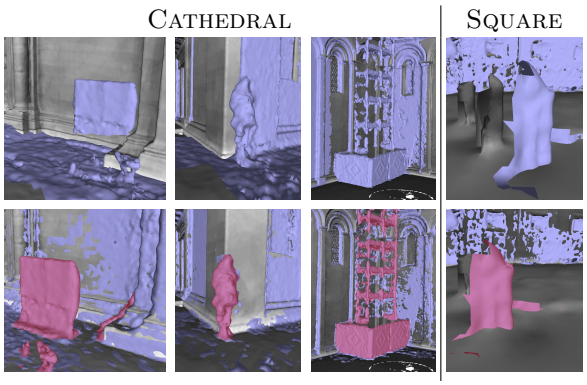
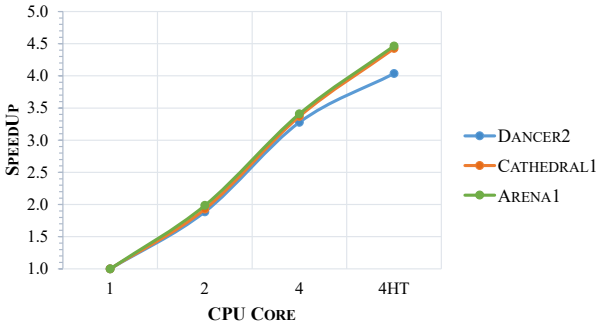


Figure 11: Examples of temporal changes that the registration can preserve (left) and can integrate (right). (Top) Initial registration. (Bottom) Final registration with the highlight of the change regions.



	CPU Core		
	2	4	4 (Hyper-Threading)
Dancer2	1.89 (0.94)	3.28 (0.82)	4.04 (0.51)
Cathedral1	1.93 (0.96)	3.37 (0.84)	4.43 (0.55)
Arena1	1.98 (0.99)	3.41 (0.85)	4.47 (0.56)

Figure 12: Speedup of the our algorithm by changing the number of CPU core (1, 2, 4 and 4 with hyper-threading). In parenthesis the parallel efficiency. The test has been done on three different datasets. (Dancer2, Cathedral1 and Arena1)

OpenMP to parallelize the Voronoi clustering (parallel flooding from the seed), the ICP procedure (one thread for each super-patch), the propagation of the valid transformation on the invalid patches (all the patches at the same distance in parallel) and the final interpolation (one thread for each vertex). Figure 12 shows the speedup and parallel efficiency obtained by changing the number of CPU cores for three different datasets (Dancer2, CathedralL1 and Arena1). We can observe in all the experiments that the parallel efficiency is above the 80% with the exception of the test with 4 cores with hyper-threading (8 virtual threads) where, due to the hyper-threading overhead, the efficiency is reduced to about 55%. Overall, linear scalability is mostly limited by the sequential overhead of the data structure creation required at the different steps of our algorithm. Note that for the all the experiments in the paper, we always report performance with 4 cores with hyper-threading.

4.3. Vononoi Metric Influence

In order to evaluate the best Voronoi Metric to use in the patch decomposition of the deformable mesh described in Section 3.1, we tested the registration performance of the algorithm using seven different metrics L varying the number of Lloyd iterations. The metrics are: the number of edges from the seed (Hop); the Euclidean geodesic distance (Euclid); the three metrics proposed in [49] (L_2, L_{21}, Sobolev); a metric based on the amount of displacement computed by the previous iteration of the algorithm using the scalar field computed with Equation 8 (Displace); a metric based on the amount of local flexibility of the deformation computed by the previous iteration of the algorithm using the scalar field computed with Equation 9 (Rigid). The metric Hop creates an adaptive partition to the local density of vertices of the mesh with patches with a similar number of triangles. On the contrary, the metric Euclid permits to create partitions with patches of similar area. The three metrics L_2, L_{21}, Sobolev allow the creation of patches as planar as possible with different degrees of anisotropy. Finally, the idea of the last two metrics is to guide the Voronoi clustering with the scalar field computed with Equation 8 and 9 (see Figure 5). This permits to create patches with points that underwent a similar transformation (Displace) or an as rigid as possible relative transformation (Rigid) in the previous iteration of the algorithm. In the first iteration of both metrics, we use the simple Hop measure. Table 2 shows the RMS value for the final alignment obtained on three different datasets testing the seven metrics increasing the number of Lloyd iterations. The values are very similar with a slight preference for the metrics Hop and Euclid. The only exceptions are the Variation style metrics (L_2, L_{21}, SOBOLEV) that work worse. This behavior comes from the very irregular dual graph generated by the three metrics and composed of patches of very different shape and size. This kind of graph is less effective when combined with the transformation blending described

in Section 3.4, which prefers a more regular decomposition. Considering that we only seek a partition of the deformable mesh which is adaptive to the local density and with patches of the same size in term of triangles, for the other tests in the paper, we chose the very simple HOP metric iterated only two times to retain a fast computation. To note that the additional error introduced by this metric with respect to the best one in the three tests in Table 2 is very limited.

5. Conclusion

We proposed a new non-rigid registration method to align a 3D model obtained by MVS methods to a reference mesh by removing the low deformations introduced by the MVS reconstruction process. Our algorithm uses an iterative hierarchical approach that segments the model in patches with a Voronoi clustering and runs for each of them an ICP procedure. During the ICP, we use the geometry of the surrounding patches in order to constrain the local stiffness of the deformation. Then, we recompute the transformation of the patches that underwent a sliding with respect to the reference surface. Finally, the algorithm blends the computed affine transformations using a global deformation function derived from the dual graph of the Voronoi clustering taking account of the graph topology. The algorithm is iterated until convergence by increasing the number of patches of the Voronoi clustering to capture smaller-scale deformations. Our method is designed for multi-core execution and scales with the number of CPU cores. This allows, for the first time, the alignment of large MVS meshes, made by tens of millions of polygons, in a couple of minutes. The final output is the relative alignment of the MVS mesh over the reference 3D model, also of different quality. The obtained registration can be useful in the context of the 3D model completion and temporal environment monitoring, where the use of acquisition technology based on cheap hardware, like a simple camera, is a practical cost-effective solution, allowing the integration of multimodal data and the detection of changed regions in a robust way.

Our experiments on MVS and dynamic datasets show that our approach succeeds in correcting significant deformations, while being robust to outliers, and it is fast enough to be practical for meshes of several millions of triangles. Additionally, our method turns out to perform at a similar level of quality with state-of-art, less scalable algorithms in the different application context of dynamic datasets. Our method is robust against the quality of the mesh when it is degraded by noise, missing geometry, irregular triangulation, and when it contains multi-scale input with a very different level of details. Finally, our approach is able to preserve geometric differences due to the presence of new geometry by integrating the low-frequency deformations with the locally rigid and consistent changes (Figure 11).

5.1. Limitations and Future Work

The robustness of our approach is related to the initial rough alignment between the two meshes and to the amount of their overlap. As for all local patch transform methods, if the starting position is very far from the optimal one, the method can converge to a wrong local minimum. However, the use of robust solutions for the rough global alignment before our method, such as [39] and [4], can reduce this problem significantly. In future work, we plan to investigate the introduction of a per-patch rough global alignment step. A challenging feature for the algorithm is the amount of overlap between the models. While the algorithm can manage a partial overlap, like for example the EXCAVATION dataset where the changed regions in the middle due to the excavation process are well preserved, a problem would occur if the overlapping has too few triangles and high geometric dissimilarity. In this case, a solution based on L1 minimization could solve the problem by reducing the contribution of the outliers. A further future research direction is the extension of the algorithm to the case of more than two meshes allowing the robust comparison of multi-temporal datasets. In this case, the independent alignment of each mesh with respect to a common reference model cannot guarantee a good registration among the other time steps. In future, it would be interesting to investigate how to obtain a good registration controlling the deformation consistency among all the input models with explicit change detection.

Acknowledgement

This work was partially financed by the SCI-ADRO project, co-funded by the Tuscany Region (Italy) under the Regional Implementation Programme for Underutilized Areas Fund (PAR

	Reference Mesh A	Input Mesh B	# nodes deformation graph	# iteration	time (sec)	Start RMS (mm)	Final RMS (mm)
Cathedral1	51.2M	10.9M	168/420/1382	4	320	748.8	46.1
Cathedral2	51.2M	17.6M	154/388/1218	6	465	487.5	39.9
Cathedral3	51.2M	15.4M	217/368/1692	4	397	66.3	34.6
Cathedral4	51.2M	20.9M	170/428/1612	7	587	352.3	27.8
Cathedral5	51.2M	15.5M	148/342/1241	3	381	40.3	27.7
Square1	4M	11.9M	119/247/1020	3	144	203.6	78.7
Square2	4M	8.7M	152/455/1222	3	221	78.4	64.1
Square3	4M	18.5M	91/247/836	8	1152	797.1	70.1
Arena1	24.2M	7M	336/607/1795	3	241	72.7	37.2
Arena2	24.2M	29.8M	326/604/1970	3	672	160.1	49.4
Excavation1	23M	12.5M	130/319/1134	4	498	96.7	21.4
Excavation2	23M	25.7M	121/249/1363	5	786	369.4	27.7
Dancer1	40k	40k	43/117/401	4	2.9	29.4	0.85
Dancer2	40k	40k	52/92/371	6	3.6	87.8	2.497
Face1	1M	1M	(46) 144/320/1103	(4) 4	(40) 51	3.72	0.174
Face2	1M	1M	(78) 158/352/1164	(12) 8	(91) 66	2.18	0.179
Torso1	287k	267k	(18) 127/215/673	(8) 7	(34) 43	31.6	0.501
Torso2	287k	267k	102/229/859	5	30	8.83	0.497

Table 1: Test Data. For each test, the table contains the number of triangles of each mesh, the number of nodes of the deformation graph at the different steps of the algorithm (first iteration/loop/final wrap), the number of iterations of the algorithm, the computation time, the RMS before and after our algorithm. For the dynamic dataset, the numbers in parenthesis are relative to the preprocessing step.

FAS 2007-2013) and the Research Facilitation Fund (FAR) of the Ministry of Education, University and Research (MIUR), the French National Research Agency (ANR), under grant ANR 16-LCV2-0009-01 ALLEGORI, and BPI France, under grant PAPAYA.

References

[1] M. Goesele, N. Snavely, B. Curless, H. Hoppe, S. M. Seitz, Multi-view stereo for community photo collections, in: 2007 IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8. doi:10.1109/ICCV.2007.4408933.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, R. Szeliski, Building rome in a day, in: 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 72–79. doi:10.1109/ICCV.2009.5459148.

[3] J. Heinly, J. L. Schnberger, E. Dunn, J. M. Frahm, Reconstructing the world* in six days, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3287–3295. doi:10.1109/CVPR.2015.7298949.

[4] N. Mellado, M. Dellepiane, R. Scopigno, Relative scale estimation and 3d registration of multi-modal geometry using growing least squares, IEEE Transactions on Visualization and Computer Graphics 22 (9) (2016) 2160–2173. doi:10.1109/TVCG.2015.2505287.

[5] R. A. Persad, C. Armenakis, Automatic co-registration of 3d multi-sensor point clouds, ISPRS Journal of Photogrammetry and Remote Sensing 130 (2017) 162 – 186. doi:10.1016/j.isprsjprs.2017.05.014.

[6] H. Li, R. W. Sumner, M. Pauly, Global correspondence optimization for non-rigid registration of depth scans, Computer Graphics Forum 27 (5) (2008) 1421–1430. doi:10.1111/j.1467-8659.2008.01282.x.

[7] W. Chang, M. Zwicker, Range scan registration using reduced deformable models, Computer Graphics Forum 28 (2) (2009) 447–456. doi:10.1111/j.1467-8659.2009.01384.x.

[8] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, G. Gusev, 3d self-portraits, ACM Trans. Graph. 32 (6) (2013) 187:1–187:9. doi:10.1145/2508363.2508407.

[9] Q. Y. Zhou, S. Miller, V. Koltun, Elastic fragments for dense scene reconstruction, in: 2013 IEEE International Conference on Computer Vision, 2013, pp. 473–480. doi:10.1109/ICCV.2013.65.

[10] C. Cagniart, E. Boyer, S. Ilic, Free-form mesh tracking: A patch-based approach, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1339–1346. doi:10.1109/CVPR.2010.5539814.

[11] A. Cohen, C. Zach, S. N. Sinha, M. Pollefeys, Discovering and exploiting 3d symmetries in structure from motion, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1514–1521. doi:10.1109/CVPR.2012.6247841.

[12] P. J. Besl, N. D. McKay, A method for registration of 3-d shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256. doi:10.1109/34.121791.

[13] Y. Chen, G. Medioni, Object modelling by registration of multiple range images, Image Vision Comput. 10 (3) (1992) 145–155. doi:10.1016/0262-8856(92)90066-C.

[14] S. Rusinkiewicz, M. Levoy, Efficient variants of the icp algorithm, in: Proceedings Third International Conference on 3-D Digital Imaging and Modeling, 2001, pp.

DANCER2

	Lloyd Iteration					
	2	4	8	16	32	128
HOP	2.497	2.338	2.476	2.595	2.647	2.635
EUCLID	**2.458**	**2.292**	**2.338**	**2.397**	**2.472**	**2.467**
L_2	3.203	3.318	3.169	3.097	3.130	2.910
L_{21}	3.571	3.381	3.511	3.565	3.370	3.186
SOBOLEV	3.501	3.231	3.206	3.095	3.078	2.901
DISPLACE	2.670	2.559	2.748	2.861	2.765	2.834
RIGID	2.622	2.422	2.415	2.530	2.616	2.843

CATHEDRAL1

	Lloyd Iteration					
	2	4	8	16	32	128
HOP	**46.08**	**45.76**	**45.96**	**45.85**	**46.41**	**46.42**
EUCLID	46.99	46.26	46.47	46.16	46.46	46.74
L_2	47.77	51.26	49.30	49.01	50.85	48.55
L_{21}	47.47	50.64	48.14	47.96	49.77	49.30
SOBOLEV	47.59	47.79	49.14	47.77	47.68	48.13
DISPLACE	46.88	46.64	46.68	46.75	47.00	47.26
RIGID	46.21	46.27	**45.96**	46.21	46.60	46.45

ARENA1

	Lloyd Iteration					
	2	4	8	16	32	128
HOP	37.34	37.34	37.32	37.35	**37.05**	**37.15**
EUCLID	**37.12**	37.32	37.41	37.60	37.28	38.05
L_2	41.13	42.26	38.84	41.61	41.81	41.51
L_{21}	37.39	37.41	37.60	38.64	39.16	38.31
SOBOLEV	37.40	37.67	38.00	38.35	37.84	39.16
DISPLACE	37.27	37.26	**37.18**	**37.01**	37.18	37.48
RIGID	37.49	**37.05**	37.30	37.38	37.40	37.35

Table 2: Evaluation of the alignment quality by computing the final RMS value of the three datasets DANCER2, CATHEDRAL1 and ARENA1 testing seven different metrics to minimize in the Voronoi segmentation. Each metric has been tested with a different number of Lloyd iterations. The bold values highlight the best results for the test with the same number of Lloyd iteration while the green values highlight the best absolute combination for each dataset.

145–152. doi:10.1109/IM.2001.924423.

[15] L. Malleus, T. Fisichella, D. Lingrand, F. Precioso, N. Gros, Y. Noutary, L. Robert, L. Samoun, Kppf: Keypoint-based point-pair-feature for scalable automatic global registration of large rgb-d scans, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 2017, pp. 2495–2502. doi:10.1109/ICCVW.2017.294.

[16] D. Aiger, N. J. Mitra, D. Cohen-Or, 4pointss congruent sets for robust pairwise surface registration, ACM Trans. Graph. 27 (3) (2008) 85:1–85:10. doi:10.1145/1360612.1360684.

[17] P. W. Theiler, J. D. Wegner, K. Schindler, Keypoint-based 4-points congruent sets automated marker-less registration of laser scans, ISPRS Journal of Photogrammetry and Remote Sensing 96 (2014) 149 – 163. doi:https://doi.org/10.1016/j.isprsjprs.2014.06.015.

[18] Y. Díez, F. Roure, X. Lladó, J. Salvi, A qualitative review on 3d coarse registration methods, ACM Comput. Surv. 47 (3) (2015) 45:1–45:36. doi:10.1145/2692160.

[19] L. Ikemoto, N. Gelfand, M. Levoy, A hierarchical method for aligning warped meshes, in: Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings., 2003, pp. 434–441. doi:10.1109/IM.2003.1240279.

[20] B. J. Brown, S. Rusinkiewicz, Global non-rigid alignment of 3-d scans, ACM Trans. Graph. 26 (3). doi:10.1145/1276377.1276404.

[21] Q.-X. Huang, B. Adams, M. Wicke, L. J. Guibas, Non-rigid registration under isometric deformations, Computer Graphics Forum 27 (5) (2008) 1449–1457. doi:10.1111/j.1467-8659.2008.01285.x.

[22] N. J. Mitra, S. Floery, M. Ovsjanikov, N. Gelfand, L. Guibas, H. Pottmann, , in: A. Belyaev, M. Garland (Eds.), Proceedings of the Fifth Eurographics Symposium on Geometry Processing, The Eurographics Association, 2007, pp. 173–182. doi:10.2312/SGP/SGP07/173-182.

[23] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, A. Schilling, Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data, ACM Trans. Graph. 28 (2) (2009) 15:1–15:15. doi:10.1145/1516522.1516526.

[24] B. Allen, B. Curless, Z. Popović, The space of human body shapes: Reconstruction and parameterization from range scans, ACM Trans. Graph. 22 (3) (2003) 587–594. doi:10.1145/882262.882311.

[25] M. Pauly, N. J. Mitra, J. Giesen, M. Gross, L. J. Guibas, Example-Based 3D Scan Completion, in: Eurographics Symposium on Geometry Processing 2005, The Eurographics Association, 2005, p. 23. doi:10.2312/SGP/SGP05/023-032.

[26] H. Li, B. Adams, L. J. Guibas, M. Pauly, Robust single-view geometry and motion reconstruction, ACM Trans. Graph. 28 (5) (2009) 175:1–175:10. doi:10.1145/1618452.1618521.

[27] B. Amberg, S. Romdhani, T. Vetter, Optimal step nonrigid icp algorithms for surface registration, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8. doi:10.1109/CVPR.2007.383165.

[28] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, S. Leutenegger, Elasticfusion: Real-time dense slam and light source estimation, The International Journal of Robotics Research 35 (14) (2016) 1697–1716. doi:10.1177/0278364916669237.

[29] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, S. Izadi, Fusion4d: Real-time performance capture of challenging scenes, ACM Trans. Graph. 35 (4) (2016) 114:1–114:13. doi:10.1145/2897824.2925969.

[30] C. Budd, P. Huang, M. Klaudiny, A. Hilton, Global non-rigid alignment of surface sequences, International Journal of Computer Vision 102 (1) (2013) 256–270. doi:10.1007/s11263-012-0553-4.

[31] M. Rouhani, E. Boyer, A. D. Sappa, Non-rigid registration meets surface reconstruction, in: 2014 2nd International Conference on 3D Vision, Vol. 1, 2014, pp. 617–624. doi:10.1109/3DV.2014.80.

[32] J. Yang, K. Li, K. Li, Y.-K. Lai, Sparse non-rigid registration of 3d shapes, Computer Graphics Forum 34 (5) (2015) 89–99. doi:10.1111/cgf.12699.

[33] A. M. Bronstein, M. M. Bronstein, R. Kimmel, Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching, Proceed-

ings of the National Academy of Sciences 103 (5) (2006) 1168–1172. doi:10.1073/pnas.0508601103.

[34] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, G. Sapiro, A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching, International Journal of Computer Vision 89 (2) (2010) 266–286. doi:10.1007/s11263-009-0301-6.

[35] V. Jain, H. Zhang, Robust 3d shape correspondence in the spectral domain, in: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006, SMI '06, 2006, pp. 19–. doi:10.1109/SMI.2006.31.

[36] R. Litman, A. M. Bronstein, Learning spectral descriptors for deformable shape correspondence, IEEE Trans. Pattern Anal. Mach. Intell. 36 (1) (2014) 171–180. doi:10.1109/TPAMI.2013.148.

[37] Q. Chen, V. Koltun, Robust nonrigid registration by convex optimization, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2039–2047. doi:10.1109/ICCV.2015.236.

[38] L. Wei, Q. Huang, D. Ceylan, E. Vouga, H. Li, Dense human body correspondences using convolutional networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1544–1553. doi:10.1109/CVPR.2016.171.

[39] N. Mellado, D. Aiger, N. J. Mitra, Super 4pcs fast global pointcloud registration via smart indexing, Comput. Graph. Forum 33 (5) (2014) 205–215. doi:10.1111/cgf.12446.

[40] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, in: Visualization and mathematics III, Springer, 2003, pp. 35–57.

[41] B. Keinert, M. Innmann, M. Sänger, M. Stamminger, Spherical fibonacci mapping, ACM Trans. Graph. 34 (6) (2015) 193:1–193:7. doi:10.1145/2816795.2818131.

[42] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, M. Levoy, Geometrically stable sampling for the icp algorithm, in: 3DIM 2003. Proceedings., 2003, pp. 260–267. doi:10.1109/IM.2003.1240258.

[43] R. W. Sumner, J. Schmid, M. Pauly, Embedded deformation for shape manipulation, ACM Trans. Graph. 26 (3). doi:10.1145/1276377.1276478.

[44] S. Fuhrmann, F. Langguth, M. Goesele, Mve: A multi-view reconstruction environment, in: Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage, GCH '14, Eurographics Association, 2014, pp. 11–18. doi:10.2312/gch.20141299.

[45] J. L. Schnberger, J. M. Frahm, Structure-from-motion revisited, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4104–4113. doi:10.1109/CVPR.2016.445.

[46] A. Knapitsch, J. Park, Q.-Y. Zhou, V. Koltun, Tanks and temples: Benchmarking large-scale scene reconstruction, ACM Trans. Graph. 36 (4) (2017) 78:1–78:13. doi:10.1145/3072959.3073599.

[47] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, S. Thrun, Performance capture from sparse multi-view video, ACM Trans. Graph. 27 (3) (2008) 98:1–98:10. doi:10.1145/1360612.1360697.

[48] D. Bradley, W. Heidrich, T. Popa, A. Sheffer, High resolution passive facial performance capture, ACM Trans. Graph. 29 (4) (2010) 41:1–41:10. doi:10.1145/1778765.1778778.

[49] D. Cohen-Steiner, P. Alliez, M. Desbrun, Variational shape approximation, ACM Trans. Graph. 23 (3) (2004) 905–914. doi:10.1145/1015706.1015817.

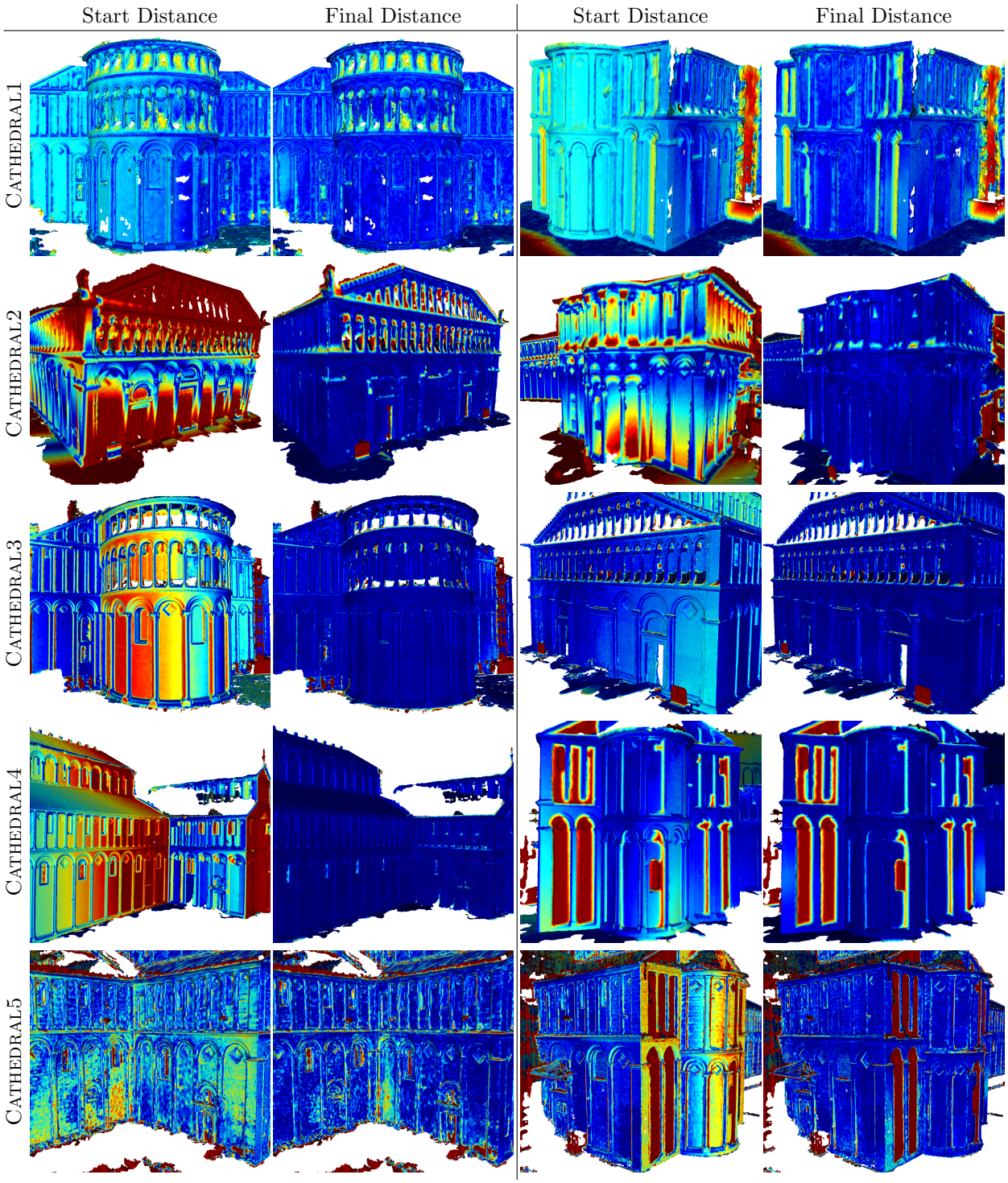


Figure 13: Closeups of some details for the dataset CATHEDRAL with the color mapping of the distance from the reference mesh after and before our algorithm. The renderings use the same color map of Figure 6.

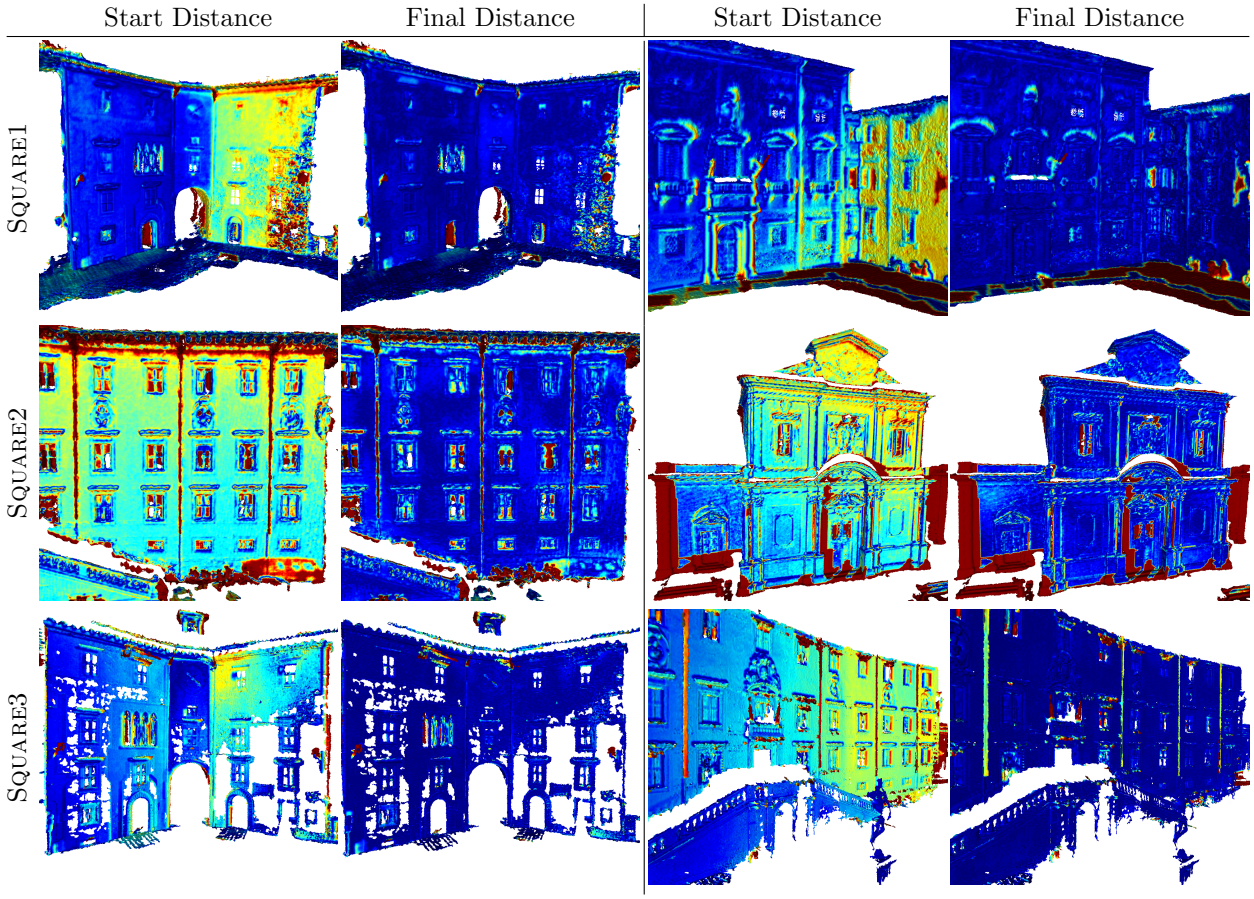


Figure 14: Closeups of some details for the dataset SQUARE with the color mapping of the distance from the reference mesh after and before our algorithm. The rendering use the same color map of Figure 7.

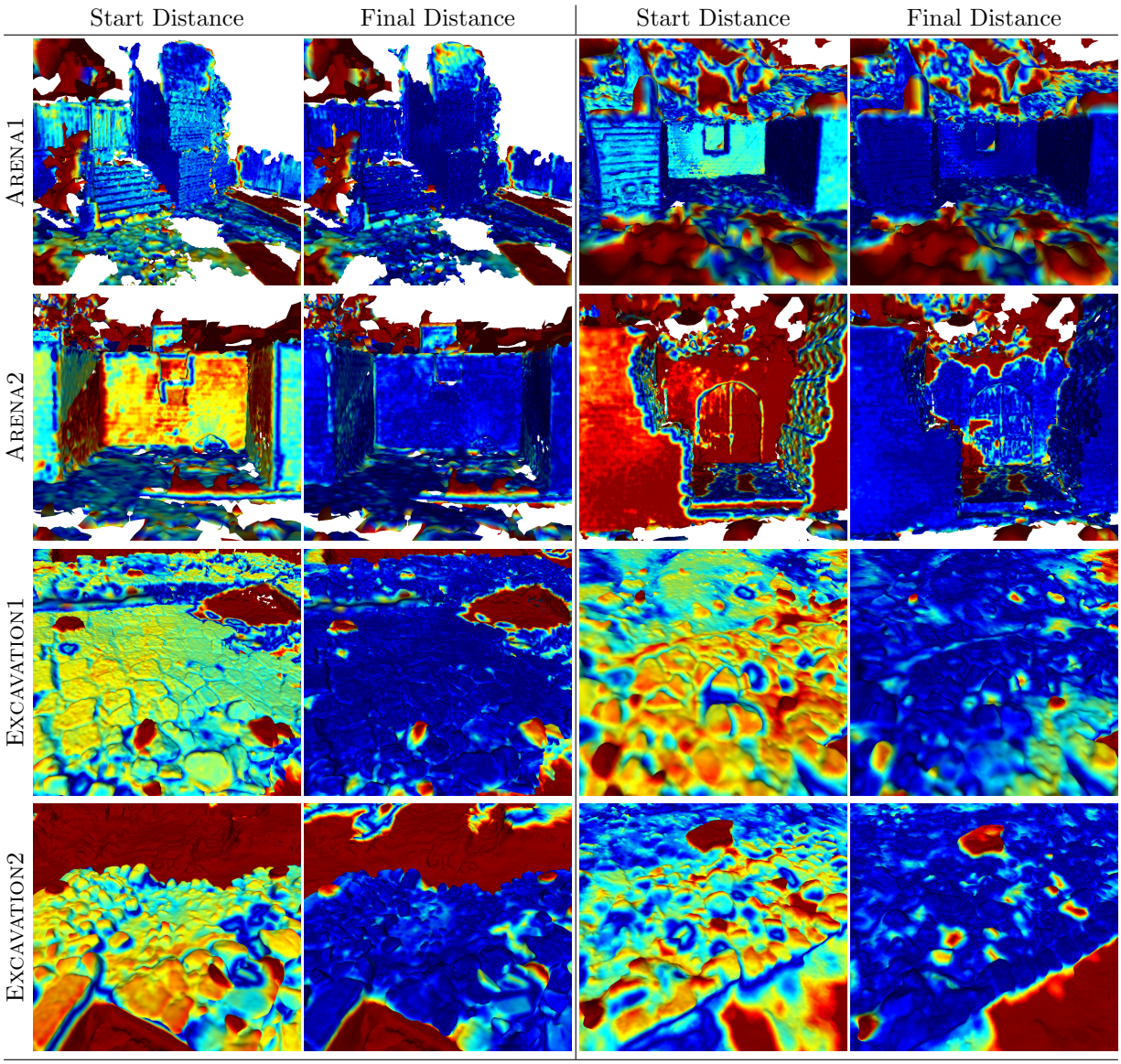


Figure 15: Closeups of some details for the dataset Arena and Excavation with the color mapping of the distance from the reference mesh after and before our algorithm. The rendering use the same color map of Figure 8.