Joint Alignment and Stitching of Non Overlapping Meshes

Susana Brandão[1] and João P. Costeira[2] and Manuela Veloso[3]

***Abstract*— We contribute a novel algorithm for aligning and stitching non-overlapping 3D meshes. Mesh aligning and stitching are often combined to construct a complete 3D model of an object. There are standard algorithms that address this task for the case of overlapping meshes, i.e., meshes that share common parts of a given object. The overlap allows the registration and alignment of different meshes. The algorithm here proposed, Joint Alignment and Stitching of Non-Overlapping Meshes (JASNOM), addresses the case when the overlap is not possible due to acquisition or application constraints. JASNOM takes advantage that both meshes can only be connected by their boundary to reframe the alignment problem as a search of the best assignment between boundary vertices. Given an assignment, the boundaries can be aligned with a rigid transformation that aligns the two boundaries by minimizing the distance between corresponding vertices while preventing the meshes from intersecting. However, not all assignments are valid. The boundaries are oriented curves and assignments between the two boundaries must preserve the order of the vertices. JASNOM takes advantage of the reduced number of order preserving assignments to perform exhaustive search for the best assignment. The outcome of the search is not only the assignment between points in the boundaries, but also the rigid transformation that aligns the two meshes. After alignment, meshes can then be stitched by using the assignment between vertices. One of the applications we envision for JASNOM is the modeling of 3D objects retrieved from range cameras that cannot overlap such as the Kinect camera. Another possible use of JASNOM is to fill holes in a mesh using parts of other meshes.**

In this work, we propose a new algorithm to solve the problem of aligning and stitching two non-overlapping meshes of the same object. The problem of stitching meshes has long been introduced in 3D object modeling, e.g. in [1]. However, to the best of our knowledge, no work has yet been devoted to deal with meshes that share no common parts. Our objective, as depicted in Figure 1, is to align two meshes, M_1 and M_2, that belong to the same object but are complementary. After alignment, meshes should be glued together in order to create a single mesh, M that models the surface of the whole object.

A possible application for our algorithm, Joint Alignment and Stitching of Non Overlapping Meshes (JASNOM), is the construction of object models from meshes captured from range sensors in diametrically opposite positions with respect to the object. Placed in such a geometry, the sensors provide an almost complete view of the object in a single time frame. JASNOM is able to align and stitch meshes returned by each sensor. Because meshes from the different cameras would be acquired with little time lapse, our algorithm can model dynamic objects, such as humans or animals, as rigid objects. This application is of particular value in cases where the range sensors cannot be used in overlap with each other, as is the case of the Kinect cameras. Due to interference between the light patterns of different cameras, the Kinect cameras do not allow multiple acquisition of images from the same surface at the same time instant.

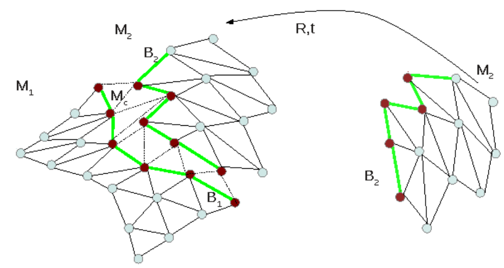


Fig. 1. Our objective is to construct a mesh M from two other meshes, M_1 and M_2. Both meshes have a boundary B_1 and B_2 that do not overlap. To construct M, we align both boundaries through a rotation R and a translation t and stitch the two meshes together by introducing a connection mesh M_c.

Another possible application of JASNOM is to fill holes in a mesh. In the case of interactive object modeling, our algorithm allows a user to select parts from a mesh or library of meshes and use them to fill holes in an incomplete 3D model. JASNOM automatically aligns meshes with boundaries and stitches the two meshes. Because it is designed for complementary meshes, the user would not need to select the exact part of the the mesh to fill the hole. Instead, the user would only need to select a part that would fit inside the hole. The capacity of filling holes from other meshes is of valuable use for the modeling of objects that have self similar surfaces such as planes or cylinders.

There are several algorithms stitch two meshes. To our knowledge, one of the first was presented in [1] and is available in mesh manipulation software programs such as Meshlab. However, the algorithm proposed in [1] assumes that meshes are initially aligned. The initial alignment requires the two meshes to have surfaces in common and thus the algorithm does not handle complementarity between shapes.

Furthermore, holes in 3d object models are so common that several algorithms have been proposed to remove them. Some algorithms, e.g. [2], [3], also use information from other meshes to fill in the holes. However, they again require overlap between meshes. There are also algorithms for closing holes that do not require overlap such as [4] and

[1] ECE departments at CMU and IST-UTL sbrandao@ece.cmu.edu
[2]ECE department at IST-UTL, jpc@isr.ist.utl.pt
[2] CS department at CMU, veloso@cmu.edu

[5]. However, those try to guess how the missing mesh should look based on its neighborhood.

Our proposed algorithm, JASNOM, differs from the previous works by not requiring previous alignment nor mesh overlap. JASNOM aligns meshes by iteratively assigning correspondences between vertices in the two meshes. Given an assignment between vertices, the algorithm computes the rigid transformation that minimizes a cost function that depends on the distance between corresponding vertices. All the possible assignments are tested and the best assignment is the one that has the minimum cost function after alignment. Because the search is exhaustive, JASNOM returns the global best assignment and respective rigid transformation. One of the contributions of this paper is on how to decrease the search space from the $N_{b1} \times N_{b2}$, the number of possible pairings between the boundary vertices of the two boundaries, to $\min(N_{b1}, N_{b2})$.

The reduction of the assignment search space is possible because the boundaries define a directed path. Correspondences between the two boundaries must respect the path topology and direction. In practice, given a single correspondence between one vertex in B_1 and another vertex in B_2, all correspondences between the remaining vertices are automatically defined. The search space is thus reduced to finding the first correspondence.

JASNOM introduces the order preserving constraint by parameterizing the boundaries. An example of a possible parameterization is the distance over the boundary to a given vertex. The parameterization also allows the use of sampling in the boundary vertices and further decreases the search space.

Given a set of correspondences, JASNOM aligns the boundaries by minimizing the distance between corresponding vertices, while preventing the two meshes from intersecting each other. The intersection is avoided by modeling the contact region between boundaries as a plane. There is no intersection if each mesh keeps to their own side of the plane. One of the contributions of JASNOM is the introduction, in the alignment cost function, of a term that penalizes the number of vertices that cross the plane to the *wrong* side.

Finally, we contribute an algorithm to stitch meshes after alignment. The stitching makes use of the assignments from the alignment stage and ensures that properties like mesh *manifoldness* are locally preserved.

The remaining of the paper is organized as follows: Section I overviews different approaches to 3D object modeling through stitching of different meshes; Section II motivates JASNOM by addressing the problem of mesh stitching from the point of view of topology; Section III introduces the JASNOM algorithm itself; Section IV presents experimental results and, finally, Section V concludes and presents future research.

I. Related Work

The use of range images for 3D object modeling motivates the use of mesh stitching to construct complete models. Due to their planar topology, range images induce an intrinsic mesh in point clouds, but do not represent the whole object. Thus, the objective becomes the construction of a complete model from meshes originated from different range images. The stitching is commonly used for either the construction of the whole object model or just to fill holes in a model.

In [1], the authors present an algorithm for stitching range images. The algorithm receives as input a set of meshes with overlap between them. The algorithm first step aligns meshes by means of an Iterative Closest Point (ICP) algorithm. The second step removes overlapping regions between two adjacent meshes, by deleting triangles. This step leaves only the triangles that do not overlap or that overlap only partially. The final step stitches meshes by the points where the partially overlapped triangles intersect. The stitching procedure adds vertices at the intersection and new triangles are built on top of the original ones. When there is no overlap, meshes cannot be aligned using ICP and the stitching cannot be built on top of existing triangles.

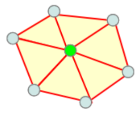
More recently, different authors, e.g. in [2] and [3], used the same technique of mesh stitching w with the purpose of filling holes in a model. In both algorithms an initial step for mesh alignment was required. However, while [2] used parts of the same object from different meshes to fill in the holes, [3] used other objects. Because the objects are different, instead of aligning the meshes with an ICP type of algorithm, [3] resorts to non rigid deformations. Both algorithms used the stitching algorithm proposed in [1] to combine different meshes.

A different approach to the stitching step itself is presented in [6]. This work proposes a closing gap algorithm, which stitches points from across narrow holes. Contrarily to the previously mention works, no alignment is needed in this case because the vertices are all from the same mesh. Also, in [6], stitching does not build on top of existing triangles, but introduces new edges and vertices. The algorithm minimizes the length of the introduced edges and does not constrain the edges to meet in pre existing vertices. Furthermore, it creates and deletes vertices in the boundary whenever required to prevent edges to cross each other. JASNOM, constrains edges to the pre-existing vertices while the distance is minimized in the alignment stage.

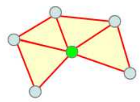
JASNOM adds to this list a stitching algorithm that aligns meshes with no overlap and connects the meshes without using existing triangles.

II. Definitions and Constraints

We now introduce some basic terminology on meshes and computational geometry to motivate and ease the discussion of our proposed algorithm. In particular, we address the problem of stitching from a topology point a view and explicit the constraints topology imposes onto stitching. The constraints will be used to design JASNOM. The terms here described can be found on computational geometry books, [7], or review papers, [8]. We start with the definition of manifold meshes and boundaries, concepts that until now have been used in loose terms. We follow with the

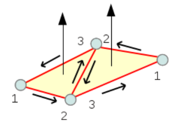


(a) Closed Fan

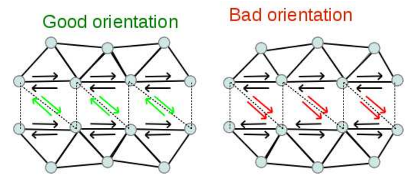


(b) Open Fan

Fig. 2. Examples of vertices in open or closed fans. In a closed fan the vertex is completely surrounded by mesh faces. In the open fan the vertex has a neighborhood that is not covered by the faces.



(a) Vertex Order



(b) Possible stitching orientations

Fig. 3. Orientability of the surface seen at the edge level. The figure on the left shows two adjacent faces with consistent normals. The order of the vertices shared by the two faces has to be different in both faces. The figure on the right shows the two possible orientations between boundaries.

introduction of some tools, such as the boundary orientation, that will be used throughout the algorithm discussion.

A. Elements of a Mesh

A mesh $M = (V, E, F)$ is composed by a set of vertices, V, edges, E, and faces, F. The vertices are defined by 3d coordinates: $V = \{x_1, x_2, ..., x_N\}, \quad x_i \in \mathbb{R}^3, \quad \forall_{i=1,N}$.

An edge in the mesh is the convex hull of two vertices and a face is the convex hull of N edges. In this work we are only interested in triangular meshes, so N=3.

A star of a vertex v, is defined as the union of all faces and edges that contain that vertex:
$S_v = \{(\cup_j E_j, \cup_k F_k) : E_j \cap v \neq \emptyset, F_j \cap v \neq \emptyset\}$.

JASNOM objective, as shown in Figure 1, is to connect two meshes, $M_1 = (V_1, E_1, F_1)$ and $M_2 = (V_2, E_2, F_2)$ using a stitching mesh: $M_c = (V_{s1} \cup V_{s2}, E_{s1} \cup E_{s2} \cup E_{stitch}, F_{stitch})$. (V_{s1}, E_{s1}) and (V_{s2}, E_{s2}) are a subset of (V_1, E_1) and (V_2, E_2) respectively. Furthermore, E_{stitch} is the set of edges incident to a vertex in mesh M_1 and another vertex in M_2: $\forall E_k \in E_{stitch} : E_k \cap V_1 \neq \emptyset \vee E_k \cap V_b \neq \emptyset$. Finally, each face in F_{stitch} is the convex hull of three edges from M_c.

The remaining of this section addresses the choice of subsets from $(V_{1,2}, E_{1,2})$ to use and how to define E_{stitch}.

B. Manifold Meshes and Boundaries

We are interested in meshes whose vertices lay on the surface of an object. To study the surface through the mesh, some properties from the former need to be preserved in the latter. One of the most important is the *manifoldness* of the object surface.

Not all meshes represent a manifold. A mesh is a 2-manifold if and only if the interior of the star of each vertex has the topology of $\mathbb{R}^2$, i.e. if it belongs to a closed fan as shown in Figure 2(a). If the mesh has a hole, it becomes a 2-manifold with boundary. In this case, the interior of the star of each vertex either has the topology of $\mathbb{R}^2$ or the topology of $\mathbb{R} \times \mathbb{R}_0^+$. The latter means that a vertex from the boundary belongs to an open fan as shown in Figure 2(b). From the edges perspective, if the mesh is a 2-manifold with no boundary, all the edges belong to 2 faces. If the mesh is a 2-manifold with boundary, the edges may belong to one or two faces.

The mesh boundary, $B = (V_b, E_b)$ is thus defined as the union of all edges that belong to just one face and all the vertices that belong to open fans. Since boundaries enclose holes, any boundary will be formed by the union of simple cycles, i.e. cycles that do not cross themselves. Even in the cases where the cycles share a vertex, they can be treated separately.

In our problem, both M_1 and M_2 are 2-manifold meshes with boundaries $B_1 = (V_{b1}, E_{b1})$ and $B_2 = (V_{b2}, E_{b2})$. By our choice, the edges connecting the two meshes, E_{stitch}, will each have a vertex in V_1 and another vertex in V_2. To preserve the manifoldness of M_1 and M_2, only the vertices from the boundary may belong to any of the edges in E_{stitch}. Any other vertex, $V_{1,i} \in V_1 \backslash V_{b1}$, is already part of a closed fan. Adding an edge to $V_{1,i}$ would break the $\mathbb{R}^2$ topology of the fan. Also, because we only consider adding edges between boundaries and not within boundaries, all the the boundary vertices need an edge in E_{stitch}. If no edge is added to one boundary vertex, it would maintain the open fan topology and there would still be a boundary on M. Finally, the edges in E_{stitch} must not cross each other. Furthermore, the boundaries that interest us, i.e., boundaries in object surface meshes, are cycles with an intrinsic direction that must be preserved by E_{stitch}. The intrinsic direction is a consequence of the orientability of the object surface mesh.

C. Orientability of Object Surface Meshes

A continuous manifold is orientable if and only if it has consistent normals to the surface in every point. Examples of orientable surfaces are the sphere or torus, while the Möbius Strip is an example of a non-orientable manifold.

For 2-manifold meshes, the definition of a normal to a face relates to a cyclic order of the face vertices. The normal to a face with vertices V_1, V_2 and V_3 can be estimated by the external product of the vertices coordinates $\hat{n}_F = (\bar{x}_2 - \bar{x}_1) \times (\bar{x}_3 - \bar{x}_2)$. If the order of the vertices changes, the direction of the normal vector will be the exact opposite. To ensure consistency on the orientation of two adjacent faces, the two vertices of the common edge must be in opposite order, as shown in Figure 3(a). The whole surface mesh is orientable if all adjacent faces are consistent. For edges that belong to the boundary, there is only one possible orientation. This orientation defines the intrinsic direction of the cycle.

To guarantee that the union of two meshes is orientable, their boundaries cannot have a random orientation with respect to each other. JASNOM stitches two meshes by assigning an edge from one boundary to the other. This situation, shown in Figure 3(b), requires the orientation of the boundaries to oppose each other.

D. Constraints

We have established the conditions to stitch two 2-manifold meshes with boundaries into a single manifold with no boundary, namely:

i) edges connecting the manifolds can only be assigned to vertices in the boundaries;
ii) all the vertices in each boundary must be assigned to at least one vertex in the other boundary;
iii) the assignment must preserve the order of the vertices in both boundaries;
iv) the order is defined by the normal to the surface of faces adjacent to the boundary.

III. Mesh Alignment and Stitching

The topology constraints from the previous section can be mapped to the geometry problem of mesh alignment and to the topology problem of stitching.

The first constraint allows solving the alignment problem just using boundary vertices. Considering that none of the other vertices can have more edges assigned to them, they will not participate on the stitching nor on the alignment. As previously stated, JASNOM only uses the boundary vertices.

The second constraint imposes that every boundary vertex should have an edge linking it to the other boundary. This is a topological constraint that does not effect the alignment between the boundaries, but affects the stitching algorithm. JASNOM contributes with two simple strategies to assign vertices to every edge and to close faces in M_c.

The third and forth constraint impose an order to the assignment and allow to define the ordering even without aligning the boundaries. JASNOM uses these constraints to parameterize the path of each boundary, and solves the correspondences problem by establishing an equivalence relation between both parameter. Because the boundaries are cycles, there is no predefined origin for the parameter. Furthermore, each choice of origins leads to a unique assignment that fulfills the previous topological constraints. Thus topology alone does not solve the correspondence problem. The geometry of both meshes then comes into play. After alignment, vertices assigned to each other should be close together. There for, each assignment between vertices induces a rigid transformation that, when applied to one boundary, minimizes the distance between assigned vertices while preventing the intersection between meshes. JASNOM evaluates how good is the alignment induced by each assignment chooses the global best.

In the remaining of the section, we first present the algorithm for creating order preserving assignments between the two boundaries; second we present our approach to alignment of the two curves given an assignment and finally we present a simple strategy to construct the faces in F_{stitch}.

A. Assignment Between Boundary Vertices

A boundary is a graph, $B = (V_b, E_b)$, with vertices defined by their coordinates $V_b = \left\{\{\bar{x}_1, \bar{x}_2, ..., \bar{x}_n\}, \quad x_i \in \mathbb{R}^3\right\}$. The edges, E_b, connect the vertices such that a simple cycle is formed, i.e., all the vertices are connected by a closed path that transverses all the vertices once and only once. By defining an origin to the boundary it is possible to embedded the boundary into a line. Given two boundaries, we address the problem of establishing correspondences between the two boundary vertices such that their order is preserved.

After choosing an origin for a boundary, B, any function defined over the boundary, can be written as a function of a single parameter, $l \in \mathbb{R}^+$. Each boundary has its own version of the parameter, l_1 and l_2. The local parameter should be monotonous in the direction of the respective path. Given two parameters, a map, $\varphi(x) : \mathbb{R}^+ \to \mathbb{R}^+$ such as $\varphi(l_2) = l_1$, defines an equivalence relation between the two boundaries. Given such a map, the pullback of a function in B_1, $f(l)$ is mapped to B_2 with the pullback of f by φ: $\varphi^* f(l_1) = f(\varphi(l_2))$.

JASNOM defines $l(x)$ as the distance over the boundary to the origin of any point x. The point x is either a vertex or just a point on a boundary edge. We use a subscript, l_k, whenever x corresponds to a vertex V_k.

The function that JASNOM needs to compare across boundaries, $f(l)$, is the ordering of points in their respective boundary. Because the order is only defined for the vertices and $f(l)$ over a continuous variable, the latter is in fact the linear interpolation of the vertices order as a function of the distance over the boundary:

$$f(l) = \frac{(l - l_k)}{(l_{k+1} - l_k)} + k, \quad l_k < l \leq l_{k+1}, \\ \forall k = 1, ..., K-1 \tag{1}$$

The order preserving assignment between boundaries depends only on the shift between the origins of both boundaries. This means that we can fix the origin of B_1 and only change the origin of B_2. Fixing the boundary of B_1 allows JASNOM to define f only once even when it searches for the best alignment, which requires consecutive shifts of the origin in B_2. Furthermore, the algorithm allocates the values of f in a look up table, which eases the retrieval during the search.

The map, φ, between the parameter l_2 and l_1 is a reflection and scaling: $\varphi(l_2(x)) = \alpha(L_2 - l_2(x))$ where L_2 is the total length of B_2 and α scales the two boundary parameters to the same length. The reflection accounts for the change of orientations from one boundary to the other.

In conclusion, JASNOM estimates the order preserving assignment of points from boundary B_1 to boundary B_2 by:

i) computing the ordering function $f(l_1)$ for B_1;
ii) computing the order i in B_1 of each vertex in B_2 by using the pullback of $f(l_1)$ by $\varphi(l_2)$: $i = \text{round}\left\{f(\varphi(l_2))\right\}$

The output of the assignment step is the assignment matrix $A \in \{0,1\}^{N \times M}$. A is a binary sparse matrix whose lines correspond to the ordered vertices in B_2 and columns to ordered vertices in B_1. $[A]_{i,j} = 1$ if and only if $i = round(f(\varphi(l_{2,j})))$. Because JASNOM assigns points from B_1 to B_2, each line in A has one and only one element equal to 1.

B. Alignment of Boundaries Curves

The problem of aligning two boundaries, B_1 and B_2 whose vertices are defined by their coordinates, $X \in R^{N\times 3}$ and $Y \in R^{M\times 3}$ respectively, is equivalent to the problem of finding a rotation, R^*, and a translation, t^*, that applied to Y minimizes some cost function of that depends on the distance from Y to AX and penalizes mesh intersection. The cost function assumes a fixed order preserving assignment matrix A, that defines the correspondences between vertex coordinates in Y and X.

JASNOM minimizes the cost function $J(R,t)$: $O(3) \times \mathbb{R}^3 \rightarrow \mathbb{R}$ defined in eq.2. Two independent terms contribute to the cost function. The first term, $J_1(R,t)$, minimizes the distance between corresponding points. The second term, $J_2(R,t)$ minimizes the intersection between meshes. To minimize the distance between points, we use the usual sum of the square of the 2-norm of the vectors $\bar{\xi}_k$ that connect each vertex k in boundary B_2 to boundary B_1. To prevent meshes from intersecting, JASNOM ensures that all the points of boundary B_2 are in the same side of the boundary B_1. In particular, they should be on the direction opposite to the normals $\bar{n}_v$ of vertices of M_1. The second term to the cost function, $J_2(R,T)$, is thus a logistic function that receives as argument the negative of projection of $\bar{\xi}_k$ on the mean normal to vertices in M_1, $\bar{n} =< \bar{n} >_v$. Negative values of the projection correspond to points that are on the wrong side of the boundary and thus are penalized by the logistic function. A slack variable, λ, combines the two cost functions. The value of λ depends on the object or application.

$$J(R,t) = J_1(R,t) + \lambda J_2(R,t) \tag{2}$$

$$J_1(R,t) = \sum_{k=1}^{N} \left\|\bar{\xi}_k\right\|_2^2 \tag{3}$$

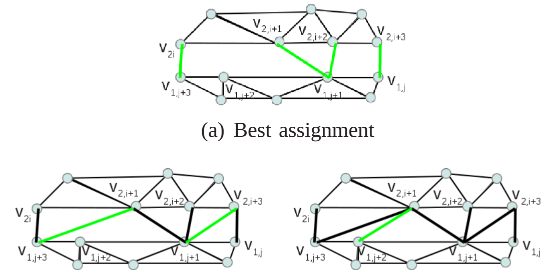
$$J_2(R,t) = \frac{1}{1+\exp\{\bar{\xi}_k \cdot \bar{n}/\|\bar{\xi}_k\|_2\}}\frac{1}{N} \tag{4}$$

$$\bar{\xi}_k = \sum_{j=1}^{M} A_{k,j}\bar{x}_j - \bar{t} - \bar{y}_j R \tag{5}$$

The cost function $J(R,t)$ is non-convex and a local solution can be found using a generic non-linear optimization algorithm, such as BFGS Quasi-Newton method [9]. To initialize the alignment algorithm, JASNOM first solves the relaxed problem obtained from eq. 2 by setting $\lambda = 0$. For $\lambda = 0$, the problem is known as an Orthogonal Procrustes problem, and has a closed form solution [10]

C. Reconstructing Triangles

After alignment, JASNOM uses the correspondences that induced the alignment to reconstruct the manifold M_c. In particular, the assignment ensures that each vertices in B_2 already has an edge connecting it to a vertex in B_1. However, not all the vertices in B_1 have an edge connecting to B_2 and some vertices in B_1 have more than one edge. Furthermore, just ensuring that there is an edge for all the vertices, does not ensure that the end result is a triangular mesh.



(a) Best assignment

(b) Second edge to B_2

(c) Missing edge in B_1

Fig. 5. Schematic for the stitching between the two meshes given the set of one to one correspondences that result from the alignment stage. See the text for comments.

To stitch the meshes together, we use two simple strategies that ensure that no edge crosses another edge from the stitching. First, we create a triangular mesh from the assignments already present. Then we assign the missing edges on B_1 so that they do not cross the edges already present.

For the first step, JASNOM runs through all the vertices $v_{2,i} \in V_{b2}$ and adds a second edge to each one. As shown in Figure 5(b), the target vertex, $v_{1,t} \in V_{b1}$ of the second edge of $v_{2,i}$ is the the first target of the next vertex, $v_{2,i+1}$ in boundary B_2,

To assign the missing edges in B_1, we run through all the vertices $v_{1,i} \in V_{b1}$ following the opposite of their order in the path. JASNOM assigns to each vertex with no edge the same target as the target of the previous vertex in line $v_{1,j-1}$, as stage is shown in Figure 5(c).

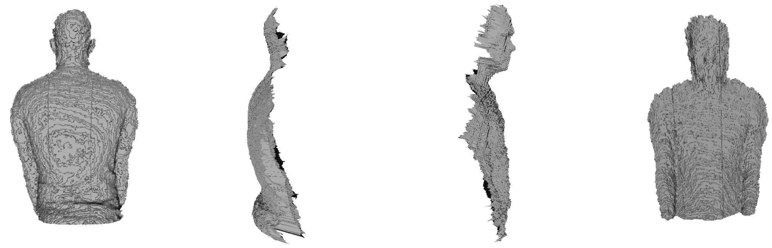
The strategies here presented are not the only way to create a mesh from the vertices and correspondences. However, these are simple criteria that result in triangles and ensure that the stitching edges, locally, do not cross each other.

IV. Experimental Results

We test our stitching algorithm with two experiments. In the first we use a range image with a hole obtained from a Kinect camera and patch the hole with a part of another mesh. In the second experiment, we construct a complete 3d model of the upper body of a human from range images obtained simultaneously with two Kinect cameras.

For the first experiment, we use a simple range image of an object with a hole, Fig. 6(a). From a second mesh a patch smaller than the hole was retrieved, Fig. 6(b). Finally JASNOM covered and stitched the hole, Fig. 6(c). Since the objective is for the patch to be inserted inside the other mesh, there was no need to penalize the intersections between meshes, i.e $\lambda = 0$;

For the second experiment, two range images of the top body of a human were retrieved simultaneously by two Kinect cameras. We then manually segmented the human was from each range image. The result from the segmentation is shown if Figure 4. The complete mesh obtained with JASNOM algorithm is shown in Fig. 7. This shape has several large holes across the boundary. In particular the hole created by the cut at the waist is large enough that, a simple attempt to minimize just the distance between points, would lead to mesh intersections. To prevent intersection,



(a) M_1, front (b) M_1, side (c) M_2, side (d) M_2, front

Fig. 4. Experiment II: construction of a mesh from a dynamic object: a human. We acquired the input meshes, M_1 and M_2, using two Kinect cameras in opposite sides of the human. We segmented the human manually.

(a) Mesh with a hole (b) Patch for the hole (c) Hole filled

Fig. 6. Results for the hole patching experiment using JASNOM

(a) Back (b) Front

(c) Top

Fig. 7. Human model completed using JASNOM.

we increased the value of the slack variable λ in the cost function. Furthermore, we note that in the two meshes used to construct the human, the boundaries are easily separated by a plane. Furthermore, because the shapes are homogeneous, the plane is reasonably well oriented according to the normal mean $< \bar{n} >$, which is in agreement with our estimate of *good* side of the mesh.

V. Conclusions and future work

In this work we have contributed an algorithm, JASNOM, that allows the joint alignment and stitching of complementary manifolds and provided evidence of its potential through simple experiment with data obtained with a Kinect camera.

With respect to existing stitching algorithms, JASNOM adds the capacity to create complete models without previous registration of individual meshes. The registration typically requires overlap between the two meshes which is not always available or convenient. JASNOM also does not require the calibration of one camera position with respect to the other. The registration and construction of models can be easily achieved with little effort and preparation.

The experiments showed that the alignment achieved at the connection region is quite satisfactory. JASNOM proved to be effective at reconstructing non rigid objects from partial views without calibrating the cameras positions. The success of JASNOM is due in part to the choice of the cost function. By using the mean of the normals as a measure of the plane that locally should divide the boundaries, the resulting alignment avoided severe intersections between the two meshes. If the object had severe in-homogeneous surfaces and the boundary was not properly explained by a plane, a different penalization would be required. In our future work we aim at defining local conditions for penalizing mesh intersections.

References

[1] Turk, G., Levoy, M., Zippered polygon meshes from range images. In: SIGGRAPH 94, New York, NY, USA, ACM (1994) 311318

[2] Marras, S., Ganovelli, F., Cignoni, P., Scateni, R., Scopigno, R., Controlled and Adaptive Mesh Zippering. In: International Conference on Computer Graphics Theory and Applications (2010) 104109

[3] Pauly, M., Mitra, N.J., Giesen, J., Gross, M., Guibas, L., Example-Based 3D Scan Completion. In: Eurographics Symposium on Geometry Processing. (2005) 2332

[4] Liepa, P., Filling Holes in Meshes. In Eurographics Symposium on Geometry Processing. (2003) 200205

[5] Clarenz, U., Diewald, U., Dziuk, G., Rumpf, M., Rusu, R., A Finite Element Method for Surface Restoration with Smooth Boundary Conditions. In: Comput. Aided Geom. Des. 21 (2004) 427445

[6] Borodin, P., Novotni, M., Klein, R.: Progressive Gap Closing for Mesh Repairing. In: Vince, J., Earnshaw, R., eds.: Advances in Modelling, Animation and Rendering. Springer Verlag (2002) 201213

[7] Edelsbrunner, H.: Geometry and Topology for Mesh Generation. Cambridge University Press, New York, NY, USA (2001)

[8] Gotsman, C., Gumhold, S., Kobbelt, L.: Simplification and Compression of 3D Meshes. In: Proceedings of the European Summer School on Principles of Multiresolution in Geometric Modelling (PRIMUS), Springer (1998) 319361

[9] Broyden, C. G.: The Convergence of a Class of Double-Rank Minimization Algorithms. In: Journal of the Institute of Mathematics and Its Applications 6: 7690 (1970)

[10] Schonemann, P.H.: A Generalized Solution of the Orthogonal Procrustes Problem. In: Psychometrika 31: 110 (1964)