

# Reconstructing a 3D Model from Range Images Using Radial Flow Model

ROSANA MARQUES DA SILVA<sup>1,2</sup>, WU, SHIN-TING<sup>1</sup>

<sup>1</sup>DCA - FEEC - UNICAMP  
13081-970 Campinas - SP, Brasil, CP.6101  
{rosana,ting}@dca.fee.unicamp.br  
<sup>2</sup>DME - CCT - UFPB  
Campina Grande - PB, Brasil, CP. 10044

**Abstract.** The reconstruction of a 3D model from range images can be conveniently split into two stages. The first stage consists basically in the extraction of geometrical information, e.g. the depth and the orientation of image points, and the second stage concentrates on the transformation of these data to a 3D model.

In this paper we focus on the second stage. Our approach is based on a deformable model, which offers more flexibility to mold a 3D model according to several geometrical constraints. Moreover, instead of looking for a global functional that carries out the deformation, we opted for establishing discrete local correspondences between 3D surface point data extracted from images and 3D model points.

**Keywords:** deformable models, range image, 3D model reconstruction.

## 1 Introduction

The reconstruction of a 3D model from images can be conveniently split into two stages. The first stage consists basically in the extraction of geometrical 3D information from image points such as their depth and intrinsic geometrical properties. And the second stage concentrates on building a 3D model from the extracted data, assigning a topology to these data. Obtained a 3D model, the knowledge of geometrical properties of the object of interest is certainly enhanced, allowing further complex geometric processings.

There is a variety of methods for reconstructing 3D objects from images. Basically two approaches are distinguishable: *patchwork model* and *deformable model*. In patchwork approach, one reconstructs 3D objects without making any previous assumption about its shape [7, 8, 12]. Distinguished faces are segmented and “sewed”. In deformable approach, 3D object shapes are obtained from the dynamic deformation of a simple, topologically equivalent one [3, 4, 15, 17].

Due to the data inaccuracy and the numerical imprecision during segmentation, it is still a hard problem to integrate separately segmented faces into a surface without gaps. The deformable approach has the advantage that it avoids this problem, since we start with a simple surface without gaps and assume that no “topological surgery” is performed during the deformation process.

The deformation process emulates indeed a phys-

ical behavior. An initial simple elastic object is subject to internal and external forces, estimated from images [5]. Governed by the continuous mechanical laws the object shape is dynamically deformed, until it reaches the equilibrium state. There are two research directions. One direction is to reduce the problem to a functional *global* minimization one, whose objective-function is to fit the image points with a minimal elastic energy surface [3, 4, 15]. Another direction is based on *local* information to reshape dynamically the initial model [13, 17], in which no minimization technique is explicitly employed. The last technique has an advantage over the previous one, in the sense that no preliminary global data analysis is required to guarantee the correct solution convergence.

An *inflating balloon model* is an example of locally deformable models. It was proposed by Chen and Medioni [17]. On the basis of a set of registered range image<sup>1</sup>, a sphere is adaptatively “inflated”, until it reaches the shape of the object to be reconstructed. The inflating algorithm imposes that the inflating direction of each point on the sphere must be within 90° of the viewing direction. And, according to our experiments, it only works well within 45°. Hence, the border of an object was not completely reconstructable from just one cartesian

---

<sup>1</sup>A range image contains the depth information of each image point in the viewing direction (direction towards the sensor). When the viewing direction is  $(0, 0, 1)^T$ , it is known as a cartesian range image, or simply a depth map.

range image, although there exists the depth information of the border in the image. Another drawback of the balloon model is the possible occurrence of self-intersections in the growing model.

As an attempt to overcome these shortages we present in this paper an alternative method to reconstruct 3D objects from range images. Our proposal is based on the local deformation approach and it requires that the objects to be modeled are topologically equivalent to a sphere. Our model can deal efficiently with one view of an object, making the best use of all the range data. The partial reconstructed model may also be improved gradually with new incoming view information. Moreover, the growing dynamics itself avoids self-intersections of the deforming model.

This paper is organized as follows. In section 2 we discuss the related previous work. Next, in section 3, we present our proposed *radial flow model*. Then, in section 4, our implementation is detailed and in section 5, some experimental results are given. Finally, in section 6 some concluding remarks are drawn.

## 2 Previous work

Deformable models have been originally proposed by Terzopoulos et al. [3] to accommodate the shape and motion reconstruction of free-form, approximate axial symmetric, flexible objects from a temporal sequence of binocular image pairs (profile data). In their work, objects were modeled as elastically deformable bodies subject to continuum mechanical laws. Profile information from multiple views were incorporated as constraint forces. The reconstructed model were the one that has minimal elastic energy. Later, Shen and Hogg [15] applied the same idea to reconstruct objects, which are rigid and symmetric about a plane parallel to the direction of motion constrained on a plane.

In order to simultaneously satisfy the reconstruction and recognition requirements, Terzopoulos and Metaxas [4] used deformable superquadrics, which incorporate the global shape parameters of a conventional superellipsoid with the local degrees of freedom of a spline. Also, they fitted models to image data by transforming the data into forces and simulating the model deformation through time with the goal that the final shape should have minimal elastic energy. They have experimented their technique with range data.

Liao and Medioni [13] claimed that the gradient descent minimization algorithm, employed by most of the *global* deformation oriented methods, is less stable, robust and accurate than the Powell mini-

mization algorithm. They also broke the 3D surface reconstruction problem into a set of 2D (linear) B-snake problems to overcome the time and space complexities.

The objects that the abovementioned techniques can reconstruct must have shape similar to the one of the initial guess. In [10], a robust initialization for *global* deformable methods is presented. An algorithmic methodology, that automatically produces a simplicial surface from a set of points in 3D about which we have no topological knowledge, is proposed. It is based on a spatial decomposition and surface tracking algorithm.

Another solution for avoiding a tight initial guess was presented by Chen and Medioni [17]. Their system is not based on global minimization methods, but on the inflating force that drives, on the basis of local measurements, an initial triangular mesh to the image points. During the growth process of the triangular mesh, the triangles will be subdivided dynamically to mold gradually to the object surface. One problem that the authors stated is that the mesh surface may self-intersect during its growth.

This paper presents an improved inflating balloon model. It is called *radial flow model*. A new inflating algorithm is proposed to exploit better the range data and to improve the inflating dynamics. In this way, we expect to achieve a more efficient and robust reconstruction procedure.

## 3 The Radial Flow Model

An icosahedron is placed conveniently at the origin of the reference system of the range data  $R$ . Under the radial inflating force and the surface tension (modeled as spring tension between adjacent vertices), the vertices of the model move towards its potential corresponding point in  $R$ . The velocity of a vertex is estimated on the basis of spring tension. During the inflation process, the triangles may be subdivided adaptively, creating new vertices with a potential correspondent in  $R$ . However, if the growing direction is parallel to a triangle, no subdivision is performed.

Once a vertex reaches its corresponding point, it is considered anchored and thus can no longer move freely. Since the velocity of a vertex depends on the spring tension, when a vertex is anchored, the velocity of its neighborhood is also reduced. The more neighboring vertices are anchored, the velocity of a vertex tends to zero. But, the growing model may be still far away from the shape represented by the image data. To remedy it, a new front with new radial growing direction is created. This process is recursive, until the most of the image points in  $R$  are

almost touched by the triangles of the radial model.

In summary, our reconstruction method comprises the following steps:

1. Initialization
2. Adaptative growing
  - (a) Subdivision
  - (b) Correspondence
  - (c) Growing
  - (d) Anchoring
  - (e) Error determination
3. New growing front determination

In this section each of these steps is explained in details.

### 3.1 Initialization

Since the initial model changes dynamically according to the corresponding image points, any initial 3D model may be used. However, to conform to our radial growing strategy, it is convenient to choose as an initial model the one with uniformly distributed vertices on its surface. We opted for the icosahedron structure. The size of the icosahedron should be such one that the initial model is “covered” by the image data to ensure the “positive” growing, and its edge is shorter than the maximal allowed length (section 3.2.1).

After then, each vertex of the icosahedron is associated to a point in the range data, in order to determine its radial growing direction. Because of the discrete nature of the range data, a tolerance should be given for this seeking process. Small displacements of vertices are allowed to align them radially with their potential corresponding points (section 3.2.2). If, even though, no corresponding point is found, the vertex is removed from the model to avoid spurious faces and the model structure is retriangulated. This is the first growing front.

### 3.2 Adaptative Growing

The growth dynamics emulates surface tension effect. The velocity of a vertex decreases proportionally to the number of anchored vertices in its neighborhood and the subdivision of its adjacent faces may be finer to facilitate its molding to the range data. In this way, we may not only diminish the undesirable results caused by spurious corresponding points, e.g. image noises  $p$  and incorrect associations  $q$ , but also capture abrupt concavity variations in image as well (Fig. 2).

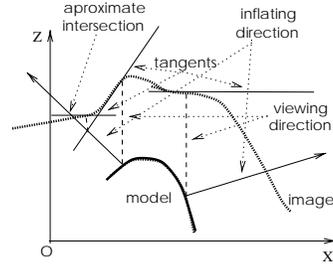


Figure 1: Correspondence in the balloon model.

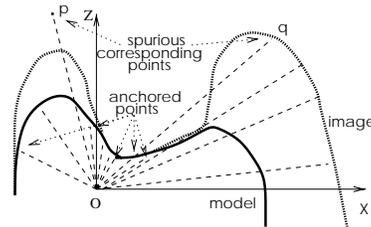


Figure 2: A growing front in the radial model.

Concerning the growth directions, we introduced a new strategy that attempts to improve the efficiency of the balloon model. This strategy requires a new growing procedure, as explained in the subsequent sections.

Chen and Medioni proposed that the vertices of triangulated model move along its surface normal. So long the surface normal is not changed, its growing direction is maintained. The way that they seek for a potential corresponding point in this direction is based on the Newton-Raphson iteration method [6], which uses the projection of the vertex in the direction of the sensor as an initial guess (Fig. 1). To guarantee the convergence of the process, the inflating direction of each point on the sphere must be within  $45^\circ$  of the viewing direction.

In our method, for each front the growing direction of its vertex is radial, from the origin of the front reference system to its potential corresponding point (Fig. 2). It is computed at the moment that the vertex is created and maintained until the vertex is anchored or a new growing front is determined. This strategy simplifies the vertex anchoring algorithm, since it is reduced to the trivial comparison of the coordinates of the vertex with the coordinates of its corresponding point. Another advantage is that the action of our algorithm is not restricted to the solid angle within  $45^\circ$  around the viewing direction.

### 3.2.1 Subdivision

In our approach, a good subdivision scheme would be the one that produces an almost uniformly distributed triangular mesh. It is worth mentioning that, differing from the work of Chen and Medioni, degenerated (thin and long) triangles are allowed in our model to avoid inserting fictitious vertices and to produce undesired side-effects.

From imaginary physical system point-of-view, the purpose of subdividing triangles is to keep constant the stiffness of the spring tension between adjacent vertices. During the growth process, the triangles become naturally bigger in size, and the spring force between adjacent vertices increases, acting

against the inflating force. Also, when the vertices are anchored, the spring tension in their neighborhood increases. Finer subdivision may alleviate these reactions.

From geometrical point-of-view, subdividing triangles allows the model to adapt the local geometry of image data without affecting other parts. We can translate the physical subdivision conditions into a geometrical tolerance parameter: the allowed maximal edge length for each triangle, denoted as *max\_length*. The value of this parameter is adaptively calculated for each front concerning its error to the image data, and for each its triangle regarding the stiffness of the neighborhood of its vertex (number of anchored adjacent vertices).

We subdivide a triangle by bisecting its longest edge. The subdivision is performed when satisfies simultaneously one of the following conditions:

- 1a. larger than  $\max(\max\_length, \min\_error)$  in each front, where *min\_error* is the minimal distance between the front and its corresponding image data (section 3.2.5), or
- 1b. larger than  $\max(\max\_length, \min\_error) * \text{stiff\_factor}$ ,  $\text{stiff\_factor} \leq 1$ , for each triangle, where *min\_error* and *stiff\_factor* are, respectively, the minimal distance to input data and the stiffness factor (section 3.2.3).

and the condition that

2. there is a corresponding point in the image data for the edge to be bisected (section 3.2.2).

Observe that the adjacent triangles of the bisecting triangle may also be bisected or trisected to maintain the model triangulated, as shows Fig. 3.

### 3.2.2 Correspondence

Because the state of each vertex plays a vital role in the growth process (as explained in section 3.2.3

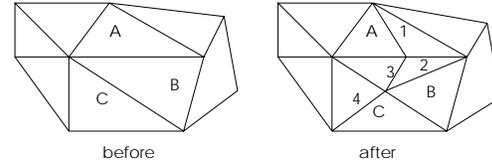


Figure 3: Propagation of subdivisions.

it affects the value of *stiff\_factor*), spurious vertices without corresponding points in image data should be avoided. Therefore, the longest edge criterion is not sufficient for deciding whether a face should be subdivided or not. We just subdivide in cases where we can find a corresponding point in input data for the edge to be subdivided.

The search algorithm of a corresponding potential point for an edge has as input the set of points belonging to the topological disk that has the middle point of the edge, *P*, as center. The radius of this disk is determined concerning the extend of the edge and its adjacent faces. This disk together with the center of the front's radial growing direction builds a solid angle. The subdividing point *P'* is the closest point to *P*, with a corresponding image point located in the solid angle (Fig. 4). In this way, we can also

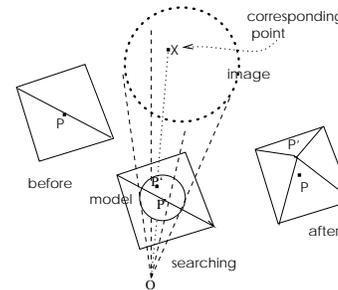


Figure 4: Determination of a subdividing point.

guarantee the topological consistency of the deforming radial flow model in relation to the image data.

### 3.2.3 Growing

The movement of any element *j* on our radial flow model is governed by the motion equation

$$m_j \ddot{v}_j + r_j \dot{v}_j + \vec{g}_j = \vec{f}_j, \quad j = 1, \dots, n, \quad (1)$$

where  $m_j$  and  $r_j$  are the mass and damping coefficient of the element *j*, respectively.  $v_j$ ,  $\dot{v}_j$  and  $\ddot{v}_j$  denote, respectively, its position, velocity and acceleration.  $\vec{f}_i$  and  $\vec{g}_i$  represent the sum of internal forces (e.g. spring tension between adjacent vertices) and the sum of external forces (e.g. inflating force), respectively.

Once our goal is just that our model stops growing (from physical point-of-view, in an equilibrium state) when it achieves the image data, we can simplify the equation (1) by making  $m_j=0$  and  $r_j = 1$ , for all  $j$ , and reduce it to

$$\dot{v}_j = \vec{f}_j - \vec{g}_j, \quad j = 1, \dots, n. \quad (2)$$

Discretizing equation (2), we obtain for each growing iteration  $i + 1$

$$v_j^{i+1} = (\vec{f}_j^i - \vec{g}_j^i) + v_j^i = k_j^i \vec{n}_j^i + v_j^i, \quad (3)$$

where  $\vec{n}_j^i$  corresponds to the radial growing direction of the element  $j$ , and  $k_j^i$ , the growing vector magnitude, which is dependent on the inflating  $\vec{f}_j^i$  and spring  $\vec{g}_j^i$  forces. A rough, but working, approximation for  $k_j^i$  is to make

$$k_j^i = \frac{max\_length}{r_j}, \quad (4)$$

where  $max\_length$  is the allowed maximal length for each triangle and  $r_j$ , the number of neighboring anchored vertices. Since  $\frac{1}{r_j}$  acts as a factor that reduces the velocity magnitude of a vertex when its adjacent spring tension increases (number of anchored vertices increases), we denote this factor as stiffness factor,  $stiff\_factor$ . This concept may aid in reducing the effect of spurious correspondences and spurious image data.

Our radial growing strategy may yield degenerated growing directions, when the image data are not from star-shaped objects. Under degenerated growing direction, we understand that the vector  $\vec{n}_j$  of a vertex becomes parallel to its adjacent triangles (Fig. 5). In this case, no inflating effect is produced.

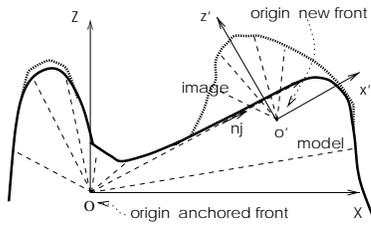


Figure 5: A degenerated direction and a new front.

To avoid these unproductive movements, we make  $r_j = 0$ .

### 3.2.4 Anchoring

As already stated, a vertex  $v_j$  is called anchored when

$$\|v_j^{i+1} - v_j^i\| > \|v_j^i - p_j\|,$$

where  $p_j$  is its potential corresponding point. Otherwise, further iterations are necessary to move  $v_j$  to  $p_j$ . However, there are two situations where a vertex tends to stop without reaching its potential corresponding point.

First, in section 3.2.3 we showed that the movement of a vertex in our radial flow model may tend to zero, when the most of its neighboring vertices are anchored. In order to improve the efficiency, we introduced the minimal stiffness concept,  $min\_stiff$ , under which we consider that the vertex can no longer move, even if the corresponding point is not reached.

Second, the simplification of equation (1) into equation (2), on the one hand, facilitates the growing and anchoring process. On the other hand, we should establish smarter (non-physically-based) convergence tests for the triangles, whether they should be further subdivided or not. This is achieved by introducing the error concept  $-tol$  (section 3.2.5). We consider that a triangle is anchored, when the maximal distance between its inner points and the corresponding image data is smaller than  $tol$ . When a triangle is considered anchored and its adjacent vertices not, our algorithm “pulls” these vertices to their corresponding points and sets them as anchored, in order to guarantee the consistency between image data and our model.

### 3.2.5 Error Determination

The error determination is important for ending our reconstruction process. As presented in section 3.2.4, the error determination is useful for anchoring same faces (and also, their adjacent vertices). This error, called *radial error* corresponds to the average vector magnitude from every image point that lies in a solid angle to the supported plane of the triangle in the radial growing direction. To determine a solid angle, the vertices of the triangle and the origin of the reference system of its front are utilized [1], as depicted in (Fig. 6).

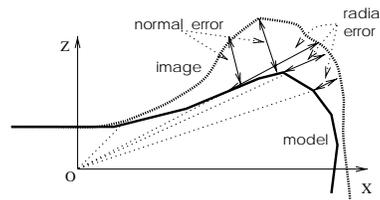


Figure 6: Radial and normal errors.

Fig. 5 illustrates a situation for which we cannot use the zero-velocity criterion to stop the process, since there may be a set of image data that are only reachable if we change the radial growing direction.

To detect the existence of new growing fronts, the distance from the image data to each face, called *normal error*, is computed and compared with the error  $tol = \frac{max\_length}{2}$  (Fig. 6). A new front is always created, whenever at least 20% of the compared points have distance greater than  $tol$ . This guarantees that our algorithm generates, at the worst case, a radial model whose distance to 80% of the range image data points is less than  $0.5 \times$  initial  $max\_length$ .

### 3.3 New growing front determination

When the existence of a new growing front is detected, our system can automatically determine the contours of the new fronts from the faces with normal error greater than the error tolerance  $tol$ . The reference system of each new front is determined, such that every vertex of contour lies on the plane  $z=0$  or above it. This simplifies the growth process, reducing it to a hemisphere.

The reference system of each new front (Fig. 5) is obtained from the baricenter  $P$  of its contour and the normal vector  $\vec{n}$  of the closest polygon to this contour[2]. A vector from baricenter to any contour vertex lies on the plane  $z=0$  defines the x-axis  $\vec{x}$  of the new reference; while the cross product  $\vec{n} \times \vec{x}$  corresponds to the direction of y-axis.

It is worth mentioning that before growing recurrently each new front, geometrical transformations should be applied to the image data, to ensure that both the image data and radial model have the same reference system. After this transformation, the initial correspondence between the existing model vertices in the new front and the image data should be re-established. In this process, we cannot often find a new potential corresponding point to each model vertex. To solve this problem, we may remove these vertices and re-triangulate the new front or we may set these vertices as *don't care* ones, whose behavior is defined by its neighboring vertices and which do not influence the growing process anymore. Due to efficiency reason, we opted for the second solution.

Furthermore, as already observed by Chen and Medioni, this front growing strategy is parallelizable, since the computations of each front are independent from each other.

## 4 Implementation

We have implemented our algorithm in C on a UNIX platform. It is runnable on SUN-SPARC, IBM-AIX, and PC (Linux). In this section we present some design decisions to either improve the algorithm efficiency or the numerical stability.

TDM [14], Topological Data Model, library is used to manage our radial flow model. TDM provides its user a variety of functionalities to create, to manipulate, and to inquire the topology of an object, without knowing the underlying data structure. In our case, it not only ensures the topological consistency (new topological rearrangements) in each subdivision, but also helps in obtaining topological information, such as the oriented contours of a set of triangles and the neighboring data.

For the proposed algorithm we use spherical coordinates  $(\rho, \theta, \phi)$  to simplify the computation of the new position of each vertex towards its potential corresponding image point. Recalling that the growing direction for each front,  $\theta$  and  $\phi$ , is constant, then equation (3) is reduceable to  $\rho_j^{i+1} = k_j^i + \rho_j^i$ .

For deciding whether a new subdivision of a face should be carried out, we impose that a new vertex can only be inserted in a face if  $\frac{\vec{\sigma p} \cdot \vec{n}}{\sigma p} \leq 0.15$ , where  $\vec{\sigma p}$  and  $\vec{n}$  are, respectively, the vector defined by the new potential vertex in the radial direction and the face normal. In this case, we also considered that its adjacent vertices are moving in the direction parallel to it.

Finally, due to the performance reason, we introduced a maximal growing step number,  $max\_step$ , to control the number of recurrences for finding new growing fronts.

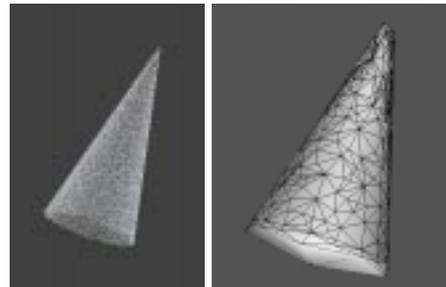


Figure 7: Cone.

## 5 Experimental Results

To evaluate the applicability of our algorithm, we carried out experiments on both synthetic and real range data. To demonstrate the performance of our algorithm in handling efficiently all the information contained in a range image, just one cartesian image was used for each reconstruction.

For illustration, we include in this section the radial flow models reconstructed from two synthetic images and three real images, by attributing initial  $max\_length=0.2$  and  $max\_step = 6$ . For each of them we present two data formats: input (range) data and

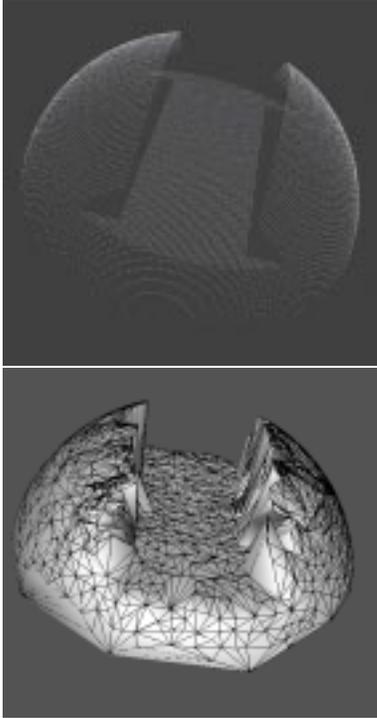


Figure 8: Esrcub.

the corresponding radial flow model rendered by Geomview [9].

We first tested the algorithm on the synthetic image of a cone (Fig. 7). In this case, we observe that our algorithm works well for adaptative computation of *stiff\_factor* and the error determination. Surface tension acts against the growing force, reducing the velocity of the vertices towards the cone apex to zero. But, through the error determination, two new growing fronts were recursively detected until the neighborhood of the cone apex was actually achieved (Table 1).

Then, we tested the performance of our algorithm in handling discontinuities in the range data (Fig. 8). Again, the surface tension was important for controlling the movement of a vertex towards a non-valid corresponding range image point and allowed the “valley” in the object to be reconstructed even though the range image does not contain the information about the lateral faces. Note that our radial model could not only reconstruct approximately the borderline of the discontinuities, but also “filled” the “missed” faces that should exist between these borders.

In order to test our algorithm for complex objects, we reconstructed the real image data of Chopin bust (Fig. 9) and the Bigweye (Fig. 10) taken from MSU/WSU range database [11]. In the both cases,

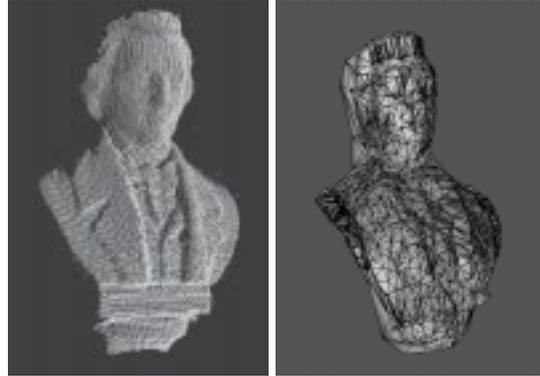


Figure 9: Chopin's bust.

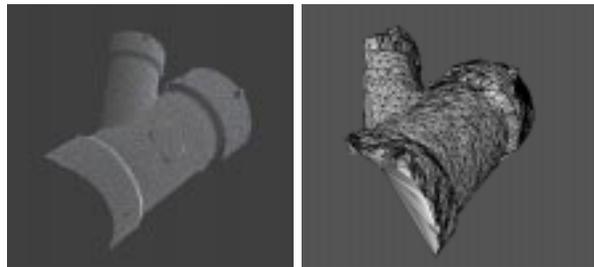


Figure 10: Bigweye.

we experimented the capability of our algorithm for reconstructing a non-star shaped object and for avoiding self-intersections in the regions with closely spaced features.

Finally, we experimented our algorithm for a part of the Chopin image data, restricted to its head (Fig. 11). The objective of this experiment was to compare the performance of our algorithm for star and non-star shaped objects. In this case, less growing steps were used to reconstruct much more details (Table 1).

## 6 Conclusions

We presented a new reconstruction algorithm for the class of objects topologically equivalent a sphere. It is based on a local deformation approach, using a *radial flow model*. Our algorithm can automatically adapt the maximal allowed edge length and growing velocity from one unique variable defined by the user, *max\_length*.

From our experiments, we can state that our reconstruction algorithm behave well for local surface properties, even when we deal with non-star shaped objects. In comparison with the work of Chen and Medioni, two qualitative improvements were achieved in our work: better use of the range (input) data and avoidance of self-intersections during the reconstruct-

image	size (# points)	# vertices	# faces	# growing fronts	# growing steps	# growing iterations
Cone	128 × 128 (12303)	197	390	3	2	15
esfcub	128 × 128 (12449)	849	1694	43	4	86
Bust	356 × 232 (45191)	*1860	3716	151	6	172
Head	131 × 142 (13026)	1116	2228	11	2	25
Bigwye	185 × 225 (21345)	1804	3604	149	5	185

Table 1: Computation results

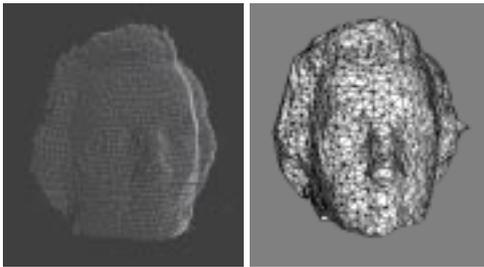


Figure 11: Chopin's head.

tion of closely spaced features..

As further work, we intend to extend our algorithm in order to support the reconstruction of other topological classes of objects.

## 7 Acknowledgements

We acknowledge Clésio L. Tozzi for very valuable discussions. We also thank CAPES/PICD for financial support to the first author during her staying at Unicamp.

## References

- [1] A. Glassner. *Graphics Gems*. Academic Press, Inc., Palo Alto, California, 1990.
- [2] D.F.Rogers. *Procedural Elements for Computer Graphics*. McGraw-Hill Book Company, Singapore, 1985.
- [3] D. Terzopoulos, A.Witkin and M.Kass. Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion. *Artificial Intelligence*, vol. 38, pp. 91-123, 1988.
- [4] D.Terzopoulos and D.Metaxas. Dynamic 3D Models with Local and Global deformations: Deformable Superquadrics. *IEEE Trans. on patt. Anal. Machine Intell.*, vol. 13, no. 7, pp.703-714, 1991.
- [5] E.Gülch. Deformable Models as a Photogrammetric Measurement Tool - Potential and Problems. *International Archives of Photogrammetry and Remote Sensing*. vol. xxxi, part B3, pp.279-284, Vienna, 1996.
- [6] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, Inc., New York, 1993.
- [7] H.T.Tanaka. Accuracy-Based Sampling and Reconstruction with Adaptive Meshes for Parallel Hierarchical Triangulation. *Computer Vision and Image Understanding*. vol. 61, no. 3, pp.335-350, 1995
- [8] I.Weiss. Shape Reconstruction on a Varying Mesh. *IEEE Trans. on patt. Anal. Machine Intell.*, vol. 12, no. 4, pp.345-361, 1990.
- [9] <http://www.geom.umn.edu/software/download/geomview.html>.
- [10] M.Algorri and F.Schmitt Deformable Models for reconstructing Unstructured 3D Data. *Proceedings Lecture Notes in Computer Science*, vol. 905, *Computer Vision - ECCV'95*, Springer-Verlag, Berlin Heidelberg, pp.420-428, 1995.
- [11] MSU/WSU range database: USC Image (composer). <http://www.eecs.usu.edu>.
- [12] W.Brent Seales and O.D.Faugeras. Building Three-Dimensional Object Models from Image Sequences. *Computer Vision and Image Understanding*. vol. 61, no. 3, pp.308-324, 1995
- [13] W.L.Chia and G.Medioni. Surface Approximation of Complex 3-D Objects, 1994, <http://iris.usc.edu/Outlines/papers.html>.
- [14] Wu S.Ting. Non-manifold data models: Implementational issues. *Proceedings of MICAD'92*, pp. 38-56. 1992.
- [15] X.Shen and D.Hogg. Shape Models from Image Sequences. *Proceedings Lecture Notes in Computer Science*, vol. 800, *Computer Vision - ECCV'94*, Springer-Verlag, Berlin Heidelberg, pp.225-230, 1994.
- [16] Y.Chen and G.Medioni. Object Modeling by Registration of Multiple Range Images. *Proc. IEEE International Conf. on Robotics and Automation*, pp.2724-2729, Sacramento, California, 1991.
- [17] Y. Chen and G. Medioni. Description of Complex Objects from Multiple Range Images Using an Inflating Balloon Model. *Computer Vision and Image Understanding*. vol. 61, no. 3, pp.325-334, 1995