Adapting Geometrical Deformable Models to Multiple Range Images

STEFAN GROSSKOPF[1] PETER JOHANNES NEUGEBAUER[1]

[1] Fraunhofer Institute for Computer Graphics
Rundeturmstr. 6, D-64283 Darmstadt, Germany
{grosskop, neugeb}@igd.fhg.de
http://www.igd.fhg.de/www/igd-a7

Abstract. Recent research topics deal with the problem how to reconstruct real-world objects from a set of multiple range images showing only portions of the object. Here we present an approach how to generate and adapt a geometrical deformable model (GDM) to a set of already registered range images in order to reconstruct the complete object. From the range images we derive a signed distance function which implicitly defines the surface of the object. Then, an intermediate volume is carved out and a sparse triangle mesh is generated. The proposed GDM scheme refines the initial roughly approximated mesh by deformation and adaptive subtriangulation. Due to the adaptive improvement of the mesh up to the desired degree of accuracy, our method describes an efficient way how to reconstruct the object in user-definable accuracy.

Keywords: surface reconstruction, surface meshing, range images, geometrical deformable models

1 Introduction

The reconstruction of complete object geometries with a 3D scanner device is generally not possible within one scan. Instead, the object has to be scanned from several directions in order to capture its complete geometry. The resulting range images must be registered and can subsequently be integrated into a model of the object surface.

Due to some amount of noise in the original data and due to partially incomplete captured surface portions, there is a need for interpolating the object surface at falsified or undefined gaps. A uniform approach solving this problem is the geometrical deformable model (GDM). The GDM was first described by Miller et al. [3] for the segmentation of volumetric data sets. Basically, a GDM is a triangle mesh that dynamically deforms by moving each mesh vertex in the direction of steepest descent along the surface of a cost function. The cost function integrates all constraints on the shape and position of the mesh into a consistent mathematical model. By minimizing the total costs the best solution is achieved.

However, a crucial drawback of this approach is the smoothing of fine details of the surface even in regions where it is defined properly, e.g., sharp edges. It is caused by improper weighting of internal cost terms which are intended to preserve the mesh smoothness and topology. We propose a deformation scheme that moves the vertices under constraint to minimize the external cost term exclusively. The presented optimization procedure achieves high quality of the mesh by moving the vertices along two types of forces, a spring force and an expansion force. The spring force maintains the mesh regularly whereas the expansion force drives the mesh towards the surface.

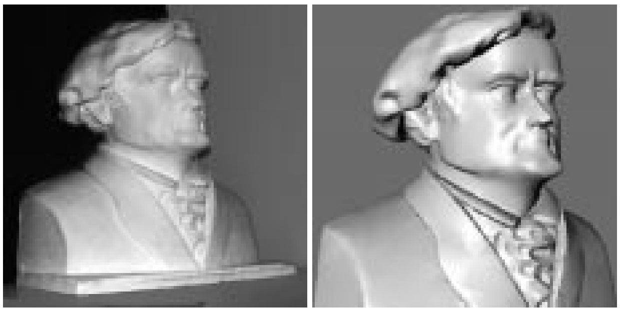

Figure 1: (a) plaster bust of composer Richard Wagner (b) 3D model reconstructed from 27 scanned range images

The remainder of this paper is organized as follows. Section 2 gives an overview of the processing steps. Section 3 discusses the definition of the implicit surface from multiple registered range images. Section 4 deals with the generation of the template mesh. Section 5 discusses topological improvements of the mesh applied during the deformation process. Section 6 presents two approaches to improve the vertex positions of a given mesh: a fast one and the proposed GDM approach. In section 7 results are shown and discussed.

2 Overview

The processing steps and intermediate representations that yield an accurate, sparse triangulated surface starting from a set of range images is shown in Fig. 2. The first step is the registration of range images [5] and results in the model cluster. The cluster holds the information needed to define an implicit surface, i.e., the range images and

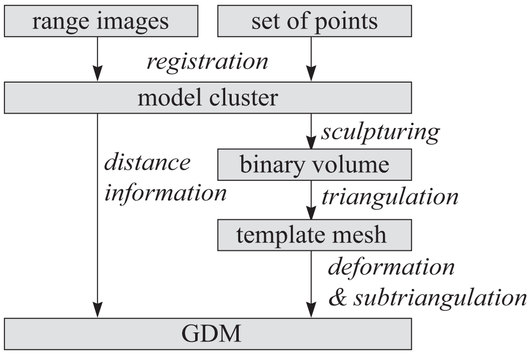

Figure 2: Pipeline for adapting the GDM

their attached transformation matrices. For the calculation of Euclidean distances to the surface and for higher performance it also contains a distance transformed volume that assigns every voxel its distance to the nearest surface point. In a subsequent step, a binary volume is sculptured. By using an octree representation during sculpturing we are able to generate a volume of arbitrary resolution. By application of the marching cube algorithm a template triangle mesh is generated from the volume. This template is then adapted to the surface by deformation and subtriangulation.

3 Definition of the Implicit Surface

3.1 Distance Function

Initially, the surface of the object to be reconstructed is given by a number of range images that are already registered. The parameter grid of the range images is defined by the coordinate system of the scanner, e.g., a cylindrical or a perspective system. By interpolating between the grid points the surface can be continuously completed.

Now, we convert the range images to a signed distance function. For the definition of the distance we define for each range image a function $g_i(\boldsymbol{x})$ that measures the signed projection distance between a point in space and the interpolated surface in the range image. The corresponding point in the range image is found by projecting the point $\boldsymbol{x}$ onto the parameter grid of the range image. Thus, a positive distance indicates that the point lies between the scanner and the surface and consequently is visible whereas a negative distance indicates that the point lies below the surface and is invisible.

The synthesis of multiple range images is achieved by combining the functions $g_i(\boldsymbol{x})$ as shown in eq. (1).

$$f(\boldsymbol{x}) = \max_i \; \{ g_i(\boldsymbol{x}) \} \qquad (1)$$

The implicitly defined surface is given by the zero crossings of $f(\boldsymbol{x})$. It is now possible to calculate the intersection between an arbitrary ray and the surface of the object.

3.2 Distance Transformation

The distance function discussed in the previous subsection calculates the projection distance, which has the same zero crossings as the Euclidean distance. The function consequently is suitable for the definition of the implicit surface and for the definition of the visibility of a point. However, in order to approximate the Euclidean distance from a point to the nearest surface point, we propagate distances into space by calculating a distance transformed volume. This has the positive side effect that it also reduces the effort of calculating the distance, which otherwise depends linearly on the number of range images. In order to prevent loss of information the projection distance within the voxels that contain portions of the surface can be calculated additionally.

The distance transformed volume is generated by a floating point number based chamfering distance transformation. The distance is at first defined only for voxels that contain parts of the object surface. For initialization of these voxels the projection distance to the center of gravity of the voxel is calculated and stored to the voxel. By application of a two pass transformation algorithm [1] the distances are propagated successively into the neighborhood. After the distance transformation is completed, the invisible voxels are defined as to be negative.

4 Generating the Template Mesh

In order to achieve a first approximation of the object surface and to derive the topology of the object up to the desired level of detail, we use a sculpturing approach to build an intermediate volumetric model. For each voxel $\boldsymbol{x}$ of the volume the distance function $f(\boldsymbol{x})$ can be evaluated and we thus obtain the binary decision whether the point does belong to the object. In some cases, there remain some volumetric regions that do not belong to the object because the associated voxels e.g., lie on the back-face of the object, far below one of the range images defining the front side. However, after generating the intermediate volumetric model, the marching cube algorithm with a look-up table that resolves ambiguous cases [4] can be applied to generate a polygonal representation. In order to delete wrong volumetric regions all connected meshes are detected and all but the largest mesh are deleted. The accuracy of this polygonal mesh is improved by moving the vertices of the mesh onto the surface implicitly defined by the registered range images [5].

5 Improving the Mesh Topology

To be able to approximate fine details of the surface our scheme refines the grid at surface portions with high curvature and removes triangles where the reconstructed surface is nearly flat. This benefits for a compact representation and accelerates operations performed on the mesh.

By merging those points connected by very short edges and deleting the corresponding triangles, the number of triangles can be reduced very easily. In addition, standard mesh simplification and optimization algorithms such as edge swapping [6, 7] are applied.

5.1 Subdivision

Triangles are splitted into a number of faces if the distance of one of the centers of gravity of the three edges or of the center of gravity of the triangle is larger than a specified threshold value. The new vertices are found as the intersection of the mesh normal with the implicit surface. After determination of the point locations one of the subtriangulation schemes of Fig. 3 is chosen and the splitted triangle is replaced by the new triangles.

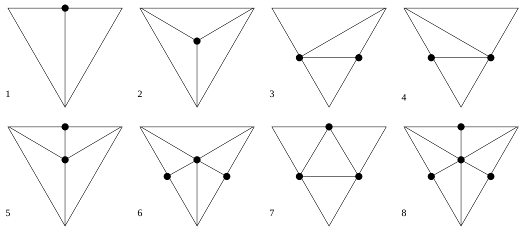

Figure 3: Subdivision configurations

6 Improving Vertex Positions

6.1 A Fast Approach: Smooth and Reproject

The initial triangulation can be improved by shifting the vertices of the mesh onto the center of gravity of the surrounding polygon (smoothing). We then define a ray that runs through the shifted vertex and parallel to the mean normal of the neighboring triangles. The vertex is now re-projected onto the surface by finding the nearest surface intersection. Hence we derive triangles of approximately equal size and inner angles. The visualization of the object appears to be greatly improved as small noise in the vertex coordinates only slightly influences the surface normal of the triangle. On the other hand, vertices may be delocalized apart from small step edges of the surface.

6.2 GDM

6.2.1 Force Definition

The deformation of the GDM from the initial template is driven by the simulation of two forces. On one hand the edges act like springs. According to equation (2) the spring forces are normalized in order to yield equilateral triangles.

$$F_{\text{spring}}(\boldsymbol{x}_i) = \sum_{(i,j)\in Vertices} \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{|\boldsymbol{x}_j - \boldsymbol{x}_i|} \qquad (2)$$

On the other hand a pressure force defined along the surface normals causes a smooth deformation of the GDM as it is done for reprojection.

6.2.2 Optimization Procedure

The deformation of the GDM is performed iteratively by moving each vertex along the direction of each force. The step size for this is chosen individually for both forces at each vertex. Two strategies are sensible.

1) As a first step, move the vertex along the spring force. The step size has to be smaller than the distance value of the starting point and shorter than the distance to the center of gravity of the surrounding polygon. If a defined distance value calculated for all adjacent triangles increases, this step is not performed. In a second step, move the vertex along the positive normal direction. The step size has to be smaller than or equal to the distance value of the starting point. If the distance value of the adjacent triangles increases, the negative normal direction is tested.

2) Since the second step of the above procedure assures for minimizing the distance of the adjacent triangles to the surface, the constraint for the movements along the spring force can be relaxed. This is done by a stochastic approach. We adapt an acceptance criteria from simulated annealing [2] which is shown in equation (3). If the probability P_{ij} is larger than an equally distributed random number in the interval of $[0\ldots1)$ the new state is accepted, otherwise rejected.

For the results shown in the next section the latter method has been used. In equation (3) T indicates the temperature of the system and $h(.)$ indicates the Euclidean distance to the surface as mentioned in section 3.2.

$$P_{ij} = \begin{cases} 1 & \text{if } h(\boldsymbol{x}_i) \le h(\boldsymbol{x}_j) \\ e^{(h(\boldsymbol{x}_j) - h(\boldsymbol{x}_i))/T} & \text{if } h(\boldsymbol{x}_i) > h(\boldsymbol{x}_j) \end{cases} \qquad (3)$$

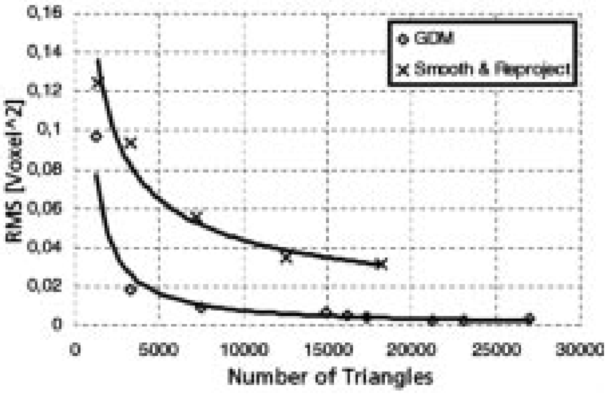

Figure 4: Approximation error for both vertex optimization methods

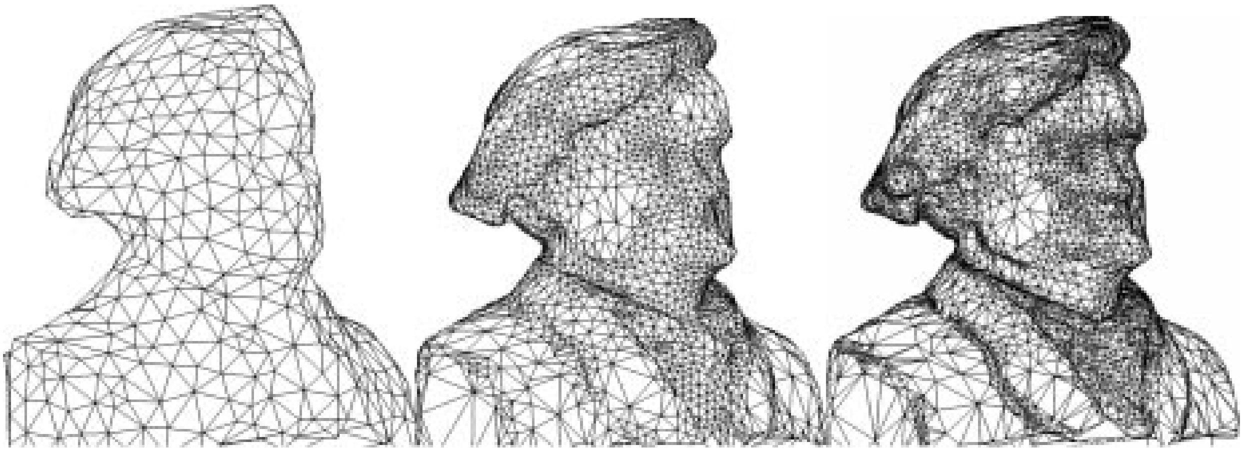

Figure 5: Three stages of the GDM-adaption process with (a) 1300 (b) 7500 (c) 16000 triangles

In order to calculate the distance of an arbitrary point to the surface, the distance transformed volume is sampled at each vertex point and additionally at the center of gravity of each triangle. From these sample points the root mean square distance is calculated. As can be seen later the minimization of this average value results in high quality surface approximation. The procedure terminates if the average vertex movement is below a given threshold value. The adaption algorithm is summarized as follows:

```
Adapt Mesh (T)
  loop {
       loop {
            remove short edges (T);
            remove redundant points (T);
            swap edges (T);
       } until no more vertices are removed;
       improve vertex positions (T);
       subdivision (T);
       swap edges (T);
       improve vertex positions (T);
  } until the required accuracy is reached;
```

7 Results

Results are presented for a bust of the composer Richard Wagner. The bust was reconstructed from 27 range images. The topology of the bust and a first approximation of its shape was sculptured in a volume of $23 \times 25 \times 16$ voxels. Fig. 5(a) shows the triangulation generated with the marching cube algorithm. Small edges have already been eliminated. Afterwards our GDM adaption procedure was applied to the data set as it is presented in Fig. 5(b) and (c).

Simultaneously the root mean square approximation error has been calculated as distance value of each triangle center of gravity resp. vertex point during each subdivision step. As can be seen from Fig. 4 the GDM approach leads to far lower approximation errors than the faster approach of smooth and reproject. Dependent on the object, approximately 4 times the number of triangles are needed for the smooth and reproject approach in order to achieve the same quality for the reconstructed object. However, the GDM approach requires far more computation time. Whereas the smooth-and-reproject approach takes in total only a few minutes for the mesh optimization and refinement, the GDM approach is approx. 100 times slower. Thus, we have a strong tradeoff between run time and accuracy.

References

[1] G. Borgefors. Distance transformations in arbitrary dimensions. *CVGIP*, 27(3):321–345, Sep. 1984.

[2] L. Ingber. Simulated annealing: Practice versus theory. *Mathl. Compu. Modelling*, 18(11):29–57, 1993.

[3] J.V. Miller. On GDM's: Geometrically deformed models for the extraction of closed shapes from volume data. Master's thesis, Rensselaer Polytechnic Institute, Troy, New York, December 1990.

[4] C. Montani, R. Scateni, and R. Scopigno. A modified look-up table for implicit disambiguation of marching cubes. *Visual Computer*, 10(6), 1994.

[5] P. J. Neugebauer. Reconstruction of Real-World Objects via Simultaneous Registration and Robust Combination of Multiple Range Images. *International Journal of Shape Modeling*, 3(1&2):71–90, 1997.

[6] W.J. Schroeder, J.A. Zarge, and W.E. Lorensen. Decimation of Triangle Meshes. In *Computer Graphics (SIGGRAPH 92 Proc.)*, volume 26, pages 65–70, July 1992.

[7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh Optimization. In *Computer Graphics (SIGGRAPH 93 Proc.)*, volume 27, pages 19–26, August 1993.