Fitting Surface of Free Form Objects using Optimized NURBS Patches Network with Evolutionary Strategies $(\mu + \lambda) - ES$

John William Branch
Escuela de Sistemas
Universidad Nacional de Colombia - Sede Medellín, Colombia
jwbranch@unalmed.edu.co

Flavio Prieto
Departamento de Eléctrica, Electrónica y Computación
Universidad Nacional de Colombia - Sede Manizales, Colombia
faprietoo@unal.edu.co

Pierre Boulanger
Department of Computing Science
University of Alberta, Canada
pierreb@cs.ualberta.ca

Abstract

We propose an algorithm to produce a 3-D CAD model from a set of range data, based on non-uniform rational B-splines (NURBS) surface fitting technique. Our goal is to construct continuous geometric models, assuming that the topology of surface is unknown. In our approach, the triangulated surface is partitioned in quadrilateral patches, using Morse theory. The quadrilateral obtained mesh is regularized by means of the use of geodesic curves and B-splines to obtain a new adequate grid on which to draw NURBS surfaces. Such NURBS surfaces are optimized by means of evolutionary strategies. Further, the patches are smoothly joined guaranteeing continuity C^1.

1 Introduction

Computer-aided geometric design and computer-aided manufacturing systems are used in numerous industries to design and create physical objects from digital models. Typically, the process consist of constructing complex objects by a combination of simple geometrical primitives. Many of these primitives are combined by boolean operations or by specifying a boundary representation where the topology and the geometry of the object are well known. However the reverse problem, which is of inferring a geometric model from an existing physical object digitized by a 3-D sensor, is a much harder problem as it is ill-posed. This paper addresses the problem of recovering 3D shape by using NURBS surfaces defined topologically as a network of quadrilaterals curves over the surface. The specification of the problem to be solved can be stated as follows:

"Given a set of sample points X assumed to lie on or near an unknown surface U, create a surface model S approximating U" [7].

In the general surface reconstruction problem, we consider that the points X are noisy. No structure or other information is assumed. The surface U -assumed to be a manifold- may have arbitrary topology, including boundaries, they contain sharp features such as creases and corners. Since the points X are noisy samples, we do not attempt to interpolate them, but instead find an approximating surface. Of course, a surface reconstruction procedure cannot guarantee recovering U exactly, since it is only given information

BMVC 2007 doi:10.5244/C.21.99

about U through a finite set of noisy sample points. The reconstructed surface S should have the same topological type as U and be close to U.

This paper is organized as following: Section 2, describes the Morse theory for triangular meshes. Section 3, a review of the pertinent literature in 3-D reconstruction. Section 4 describes the method for the adjustment of surfaces by means of optimized NURBS patches. Section 5 describes the experimental results of the proposed algorithm, and finally, in Section 6 a conclusion is presented.

2 Morse Theory for Triangular Meshes

The application of the Morse theory for triangular meshes implies to discretize Morse analysis. The Laplacian equation is used to find a Morse function which describes the topology represented on the triangular mesh. In this sense, additional points of the feature of the surface might exist, which produce a basis domain which adequately represents the geometry of the topology itself and the original mesh. The mesh can also be grouped into improved patches. In this work, Morse theory is applied by representing the saddle points and its borders by a Morse function which can then be used to determine a number of critical points.

This approximation function is based on a discrete version of the Laplacian, to find the harmonic functions. In many ways, Morse theory relates the topology of a surface S with its differential structure specified by the critical points of a Morse function $h : S \to \mathbb{R}$ [17] and is related to the mesh spectral analysis.

The spectral analysis of the mesh is performed by initially calculating the Laplacian. The discrete Laplacian operator on piecewise linear functions over triangulated manifolds is given by:

$$\Delta f_i = \sum_{j \in N_i} W_{ij}(f_j - f_i) \tag{1}$$

where N_i is the set of vertices adjacent to vertex i and W_{ij} is a scalar weight assigned to the directed edge (i, j). For graphs free of any geometry embedding, it is customary to use the combinatorial weights $W_{ij} = 1/\deg(i)$ in defining the operator. However, for 2-manifold mapped in $\Re^3$, the appropriate choice is a discrete sets of harmonic weights, suggested by Dong [14] and is the one used in this paper (see Equation 2):

$$W_{ij} = \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}). \tag{2}$$

Here α_{ij} and β_{ij} are the opposite angles to the edge (i, j).

Representing the function f, by the column vector of its values at all vertices $f = [f_1, f_2, \ldots, f_n]^T$, one can reformulate the Laplacian as a matrix $\Delta f = -Lf$ where the Laplacian matrix L each elements are defined by:

$$L_{ij} = \begin{cases} \sum_k W_{ik} & \text{if } i = j, \\ -W_{ij} & \text{if } (i, j) \text{ is an edge of } S, \\ 0 & \text{in other case.} \end{cases} \tag{3}$$

where k is the number of neighbors of the vertex i. The Eigenvalues $\lambda_1 = 0 \leq \lambda_2 \leq \ldots \leq \lambda_n$ of the matrix L forms the *spectrum* of mesh S. Besides describing the square of the frequency and the corresponding eigenvectors $e_1, e_2, \ldots, e_n$ of L, one can define piecewise linear functions over S using progressively higher frequencies [6].

3 Literature Review

A wide gamut of algorithms for surface reconstruction have been proposed in the literature in recent years [3] [7] [1].

Loop [5] generates B-spline surfaces on irregular meshes. These meshes do not require a known object topology, and therefore, they can be configured arbitrarily without carrying a sequence of the 3D coordinates of the points set. The advantage of this method is that it uses different spline types for the surface approximation. The algorithm was tested using synthetic data with low curvature.

Eck and Hoppe [11] present the first solution to the fitting problem of B-spline surfaces on arbitrary topology surfaces from disperse and unordered points. The method builds an initial parametrization, which in turn is re-parametrized to build a triangular base, which is then used to create a quadrilateral domain. In the quadrilateral domain, the B-spline patches adjust with a continuity degree of C^1. This method, although effective, is quite complex due to the quantity of steps and process required to build the net of B-spline patches on the adjustment surface.

Krishnamurthy and Levoy [15] presented a novel approach to adjust NURBS surface patches on cloud of points. The method consists of building a polygonal mesh on the points set first. Then on this mesh, a re-sampling is performed to generate a regular mesh, on which NURBS surfaces patches can be adjusted. The method has poor performance when dealing with complex surfaces. Other limitations are the impossibility to apply the method to surfaces having holes, and the underlying difficulty to keep continuity on the NURBS surface patches.

Park [8] proposed a two-phase algorithm. In the first phase, a grouping of the points is performed by means of the k-means algorithm to create a polyhedral mesh approximation of the points, which is later reduced to a triangular mesh, on which a quadrilateral mesh is built. In the second phase, the initial model is used to build a net of NURBS patches with continuity C^1. Park's proposal assumes that the cloud-of-points is closed in such a way that the NURBS patches network is fully connected. This implies that the proposed method is not applicable to open surfaces. The use of NURBS patches implies an additional process keeping continuity at the boundary, making the method computationally expensive even when the irregularity of the surface does not require it.

Boulanger *et al.* [12] describe linear approximation of continuous pieces by means of trimmed NURBS surfaces. This method generates triangular meshes which are adaptive to local surface curvature. First, the surface is approximated with hierarchical quadrilaterals without considering the jagged curves. Later, jagged curves are inserted and hierarchical quadrilaterals are triangulated. The result is a triangulation which satisfies a given tolerance. The insertion of jagged curves is improved by organizing the quadrilaterals' hierarchy into a *quad-tree* structure. The quality of triangles is also improved by means of a Delaunay triangulation. Although this method produces good results, it is restricted to surfaces which are continuous and it does not accurately model fine details, limiting its application for objects with an arbitrary topology.

Gregorski [4] proposes an algorithm which decomposes a given point-set into a data structure *strip tree*. The *strip tree* is used to adjust a set of minimal squares quadratic surfaces to the points cloud. An elevation to bi-cubic surfaces is performed on the quadratic surfaces, and they are merged to form a set of B-spline surfaces which approximates the given point-set. This proposal can not be applied to closed surfaces or surfaces which curve themselves. The proposal is highly complex because it has to perform a degree elevation and a union of patches on B-spline patches at the same time that a continuity degree C^1 is performed among adjacent patches.

Bertram [10] proposes a method to approximate in an adaptive way to disperse points by using triangular hierarchical B-splines. A non-uniform distribution of sampling on the surface is assumed, in such a way that zones with a high curvature present a denser sampling that zones with a low curvature. This proposal uses patches for data adjustment which add quality to the solution.

A different approach is presented by Yvart *et al.* [2], which uses triangular NURBS for dispersed points adjustment. Triangular NURBS do not require that the point-set has a rectangular topology, although it is more complex that NURBS. Similar to previous works, it requires intermediate steps where triangular meshes are reconstructed, re-parametrization processes are performed, and continuity patches G^1 are adjusted to obtain a surface model.

4 Approximation of Smooth Surfaces Using Morse Theory

The majority of the literature on re-meshing methods, focuses on the problem of producing well formed triangular meshes. However, the ability to produce quadrilateral meshes is of great importance as it is a key requirement to fit NURBS surface on a large 3–D mesh. Quadrilateral topology is the preferred primitives for modelling many objects and in many application domains. Many formulations of surface subdivision such as SPLINES and NURBS, require complex quadrilateral bases. Recently, methods to automatically quadrilateralize complex triangulated mesh have been developed such as the one proposed by Dong *et al.* [14].

In this section, a method for the surface approximation by means of optimized NURBS patches from complex quadrilateral bases on triangulated surfaces of arbitrary topology is proposed. This process of quadrilateralization produces regions composed exclusively of smooth quadrilaterals. To decompose the triangulated surface into quadrilateral patches, Morse theory and spectral mesh analysis are used. The quadrilateral border joining the critical points are regularized by computing geodesic curves between each corner and then B-splines approximate those geodesics. Following the geodesic curves approximation a NURBS surface is then fitted by changing the NURBS's weight to represent the data inside the quadrilateral region. Such NURBS surfaces fitting is non-linear and an evolutionary strategy optimization method is used to minimize the distance between the surface and the points inside the quadrilateral region. The optimization also takes into account the smooth joint at the boundary to guarantee C^1 continuity.

4.1 Quadrilateralization of Triangular Mesh

One of the first step of our algorithm consist of converting a triangular representation into a network of quadrilateral that is a complete description of the object's geometry. This is necessary as the representation by means of NURBS patches requires building a regular base on which the NURBS surfaces sits. Because of the complex and diverse forms of free-formed objects, obtaining a quadrilateral description of the whole surface is not a trivial task.

4.1.1 Localizing Critical Points

Initially, the quadrilateral's vertices are obtained as a critical point-set of a Morse function. Morse's discrete theory guarantees that, without caring about topological complexity of the surface represented by triangular mesh, a complete quadrilateral description is obtained. That is to say, it is possible to completely divide objects' surfaces by means of rectangles. In this procedure, an equation system for the Laplacian matrix is solved by calculating a set of eigen-values and eigen-vectors for each matrix (Equation 3) [16].

A Morse-Smale complex is obtained from the connection of a critical point-set which belongs to a field of the Laplacian matrix. The definition of a field of the matrix is obtained by selecting the set of vectors associated to a solution value of the equation. As Morse function represents a function in the mesh, each eigen-value describes the frequency square of each function. Thus, selecting each eigen-value directly indicates the quantity of critical points which the function has. For higher frequency values, a higher number of critical points will be obtained. This permits representing each object with a variable number of surface patches. The eigen value computations assigns function values to every vertex of the mesh, which permits determining whether a vertex of the mesh is at critical points of the Morse function. In addition, according to a value set obtained as the neighborhood of the first ring of every vertex, it is possible to classify the critical points as maximum, minimum or "saddle points." Identification and classification of every critical point permits building the Morse-Smale complex.

4.1.2 Critical Points Interconnection

Once critical points are obtained and classified, then they should be connected to form the quadrilateral base of the mesh. The connection of critical points is started by selecting a "saddle point" and by building two inclined ascending lines and two declined descending lines. Inclined lines are formed as a vertex set ending at a maximum critical point. In addition, a descending line is formed by a vertex path which ends at a minimum critical point. One can then join two paths if both are ascending or descending.

After calculating every paths, the triangulation of K surface is divided into quadrilateral regions which forms Morse-Smale complex cells [16]. Specifically, every quadrilateral of a triangle falls into a "saddle point" without ever crossing a path. The complete procedure is described in Algorithm 1.

Algorithm 1: Bulding method of MS cells.

```
Critical points interconnection();
begin
    Let T={F,E,V} M triangulation;
    Initialize Morse-Smale complex, M=0;
    Initialize the set of cells and paths, P=C=0;
    S=SaddlePointFinding(T);
    S=MultipleSaddlePointsDivission(T);
    SortByInclination(S);
    for every s ∈ S in ascending order do
        CalculeteAscedingPath(P);
    end
    while exists intact f ∈ F do
        GrowingRegion(f, p0, p1, p2, p3);
        CreateMorseCells(C, p0, p1, p2, p3);
    end
    M = MorseCellsConnection(C);
end
```

4.2 Regularization of the Quadrilateral Border Curves

Because the surface needs to be fitted using NURBS patches, it is necessary to regularize the quadrilateral curves obtained from the mesh. The curves are regularized and fitted by b-splines using the following Algorithm 2.

Algorithm 2: Quadrilateral mesh regularization method..

```
Regularization();
begin
    1. Quadrilateral selection;
    2. Selection of a border of the selected quadrilateral and its opposite;
    3. Regularization using B-splines with lambda density;
    4. Regularized points match by means of geodetics FMM;
        4.1 Smoothing of geodetic with B-splines;
    5. Points generating for every B-spline line with lambda density;
end
```

One of the quadrilateral border is selected from the mesh, and later a border is selected from each quadrilateral border and its opposite. The initially selected border is random. The opposite order is searched as one which does not contain the vertices of the first one. If the first selected border has vertices A and B, it is required that the opposite border does not contain vertices A and B, but the remaining, B and C.

Later, B-splines are fitted on selected borders with a λ density, to guarantee the same points for both borders are chosen, regardless of the distance between them. In general, a B-spline does not interpolate every control point; therefore, they approximate curves which permit a local manipulation of the curve, and they require fewer calculations for coefficient determination.

Having these points at selected borders, it is required to match them. This is done with FMM (*Fast Marching Method*). This algorithm is used to define a distance function from an origin point to the remainder or surface with a computational complexity of $O(n \times \log n)$. This method integrates a differential equation to obtain the geodetic shortest path by traversing the triangle vertices.

At the end of the regularization process, B-splines are fitted on geodetic curves and density λ points are generated at every curve which unite the border points of quadrilateral borders, to finally obtain the grid which is used to fit the NURBS surface.

4.3 Fitting of Optimized NURBS Patches Using an Evolutionary Algorithm

This section presents a method based on an evolutionary strategy (ES), to determine the weights of control points of a NURBS surface, without modifying the location of sampled points of the original surface. The main goal is to reduce the error between the NURBS surfaces and the data points inside the quadrilateral regions. In addition, the algorithm make sure that the C^1 continuity condition is preserved for all optimized NURBS patches. The proposed algorithm is described in Algorithm 3.

Algorithm 3: Optimization and continuity method of NURBS patches method.

`Adjustment by optimized NURBS patches();`
begin
- 1. Optimization of the NURBS patches;
  - 1.1. Multiple ES usage with deterministic replacement by inclusion;
  - 1.2. Application of ES to control weights of NURBS;
- 2. Union of NURBS patches with continuity C^1;
  - 2.1. Check continuity between axis;
  - 2.2. Check continuity at vertices;

end

4.3.1 Optimization of NURBS Parameters

A NURBS surface is completely determined by its control points $\mathbf{P}_{i,j}$. The main difficulty in fitting NURBS surface locally is in finding an adequate parametrization for the NURBS and the ability to automatically choose the number of control points and their positions. The NURBS's weight function $w_{i,j}$ determine the local influence degree of a point in surface topology. Generally, weights of control points for a NURBS surface are assigned in an homogeneous way and are set equal to 1, reducing NURBS to simple B-spline surface. The determination of NURBS control points and weights for arbitrarily curved surfaces adjustment is a complex non-linear problem.

The optimization process is formally described as follows: Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of 3-D points sampled from a real object, which has rectangular topology, and $S = \{s_1, s_2, \ldots, s_m\}$ be a NURBS surface that approximates P, our problem consist of minimizing the approximation error given by 4

$$E(S) = d_{P,S} < \delta \tag{4}$$

where $d_{P,S}$ is the total distance between P and the NURBS approximation surface S. The parameter δ is a given user error tolerance. It is attempted obtain the configuration of S so that 4 is true.

Since the influence of the NURBS surface control points is only local, the sampled points P will be divided in clusters where will carry on a local optimization process, which reduces the computational cost of the proposed method.

The optimization process starts with a clustering of the set of points P such clustering will be achieved by a SOM. The objective of the SOM is to find homogeneous regions where run the optimization process without distort the local shape of the surface. The points of P will be presented to the SOM as the training patterns. It is hoped at last of the training the SOM have found the homogeneous regions where run the optimization process.

Once clustered P an evolutionary strategy $(\mu + \lambda) - ES$ will optimize the local fitting of the NURBS in each cluster. The evolutionary strategy configuration is as follow:

Individuals: the individuals of the strategy are conformed by the weights of the cluster points and the mutation steps σ, like shows Figure 1.

$I =$	w_1	w_2	$\dots$	w_n	σ_1	$\dots$	σ_n

Figure 1: Individual of the strategy

where w_i are the control point weights and σ_i are the mutation step sizes.

Mutation operator: uncorrelated mutation with n mutation step sizes σ's is applied to the individuals.

Recombination operator: the recombination operator is different for object variable w_i than parameters σ_i. A global intermediary recombination is applied to object variables, according to 5, whereas a local intermediary recombination is applied to mutation step sizes σ_i, according to 6.

$$b'_i = \frac{1}{\rho} \sum_{k=1}^{\rho} b_{k,i} \tag{5}$$

$$b'_i = \mu_i b_{k_1,i} + (1 - \mu_i) b_{k_2,i} \tag{6}$$

where i is the allele of the individual, b_i is the value of the allele, ρ is the size of the recombination pool and μ is a random number uniformly distributed in $[0,1]$.

Selection operator: the best individuals according to the aptitude function given in 4. In order to perform a fast compute of the distance between the points P and the NURBS surface S, the points of S are store in a kd-tree structure, so that the searching process for finding the nearest points between P and S is $\log(n)$ order. The Algorithm 4 summarizes the optimization process.

Algorithm 4: Perform a clustering of P by using SOM.

begin
  for *each cluster* **do**
    Set individual size = cluster size;
    Set population size = μ;
    Initialize randomly the population;
    Evaluate the population in the aptitude function 4;
    while *the stop criterion* δ *do not reached* **do**
      for i = 1 *to* $\lambda \cdot 0.9$ **do**
        $Ind_i = mut(Population_{rand(1,\mu)})$;
      end
      for i = 1 *to* $\lambda \cdot 0.1$ **do**
        $Ind_i = rec(Population_{rand(1,\mu)})$;
      end
      $Population =$ select from$(\mu + \lambda)$;
    end
  end
end

4.3.2 NURBS Patches Continuity

Continuity in regular cases (4 patches joined at one of the vertex) is a solved problem [11]. However, in neighborhoods where the neighbors' number is different from 4 ($v \geqslant 3 \rightarrow v \neq 4$), continuity must be adjusted to guarantee a soft transition of the implicit surface function between patches of the partition. In this paper, C^1 continuity between NURBS patches is guaranteed, using Peters continuity model [9] which guarantees continuity of normals between bi-cubical spline functions. Peters proposes a regular and general model of bi-cubical NURBS functions with regular nodes vectors and the same number of control points at both of the parametric directions. In our algorithm, Peter's model was adapted by choosing generalizing

NURBS functions, with the same control points number at both of the parametric directions, bi-cubic basis functions and regular expansions in their node vectors.

Continuity Along the Quadrilateral Boundaries: To guarantee C^1 continuity between the boundaries of neighboring patches, extreme control points which affect the continuity between patches must be found. Due to data ordering within the proposed parametrization schema, two adjacent patches will have the same number of control points at the common axis, regardless of their disposition. To adjust continuity between axes, control points are calculated on the analyzed boundary, to make it co-lineal with neighboring control points on adjacent patches.

Equation 7 illustrates the new position for a control point at given P_{eje} axis, where P_A^{vec} is the neighbor point to P_{eje} at patch B. The new control point P_{eje} is the medium point between the two adjacent control points P_A^{vec} y P_B^{vec} which guarantees that control points on the axis and their adjacent neighbors at each patch is co-linear.

$$P_{eje} = \frac{P_A^{vec} + P_B^{vec}}{2} \tag{7}$$

Continuity at Quadrilateral Vertices: Continuity at vertices of quadrilateral regions is guaranteed by making sure that every adjacent control points at each vertices is co-planar.

Under the continuity criteria proposed by Peters, continuity at quadrilateral vertices are generalized, that is to say, the adjustment process is the same regardless of the number of patches which can be found at a given vertex. We have $\pi^T P = 0$, where π is a given plane and P is a point on the plane. If the system of equations is over-determined with more than four points, the equation which best adjusts a given point-set can be found.

Equation 8 represent the over-determined system where $P = [P_1, P_2, \ldots, P_n]^T$ with $n \geq 4$ are control points at the vertices. The equation is solved using Singular Values Decomposition SVD, with the last column of matrix P the equation of the plane which is adjusted to point-set P in the quadratic mean square error sense. [13].

$$\begin{bmatrix} P_x^1 & P_y^1 & P_z^1 & 1 \\ P_x^2 & P_y^2 & P_z^2 & 1 \\ \ldots & \ldots & \ldots & 1 \\ P_x^n & P_y^n & P_z^n & 1 \end{bmatrix} \begin{bmatrix} \pi_x \\ \pi_y \\ \pi_z \\ \pi_z \end{bmatrix} = 0 \tag{8}$$

Continuity is adjusted by projecting control points P onto the plane given by Equation 8:

$$\pi = n_1(x - x_o) + n_2(y - y_o) + n_3(z - z_o) \tag{9}$$

where $N = [n_1, n_2, n_3]$ is the plane's normal and $P_0 = [x_0, y_0, z_0]$ is a point on the plane. The projection $P_I = [x_I, y_I, z_I]$ of a point $P = [P_x, P_y, P_z]$ on the plane is given by:

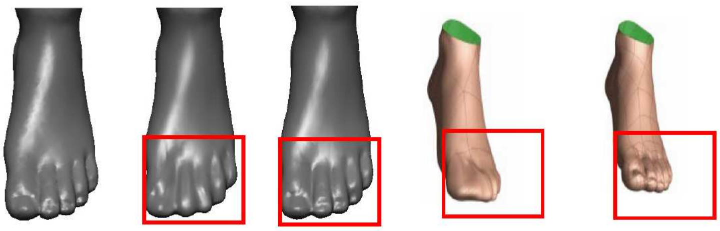
$$x_I = P_x + n_1 t_I \quad y_I = P_y + n_2 t_I \quad z_I = P_z + n_3 t_I \tag{10}$$

where t is the parametric value of the straight line which passes through point P in the direction of the plane's normal **N**.

Using Equation 10, it is possible to project control points on the given plane, which guarantee the continuity of normals at vertices of the quadrilateral partition, ensuring that every adjacent control points are co-planar.

5 Experimental Results

Tests were performed using a 3.0 GHz dual Opteron processor computer, with 1.0 GB RAM, running Microsoft Windows XP operating system. The methods were implemented using C++ and MATLAB. The data used were obtained with Kreon range scanners, available at the Advanced Man-Machine Laboratory – Department of Computing Science, University of Alberta, Canada.



(a) (b) (c) (d) (e)

Figure 2: Comparison Between Branch's Method and Eck and Hoppe's Method. a) Triangulated model, b) 27 patches model (Branch's method without optimize), c) 27 patches model (Branch's method optimized), d) 29 patches model (Eck and Hoppe's method without optimize), e) 156 patches model (Eck and Hoppe's method optimized)

5.1 Comparison Between Branch's Method and Eck and Hoppe's Method

The work by Eck and Hoppe [11] performs the same adjustment by means of a network of B-spline surface patches adaptatively refined until they obtain a given error tolerance. The process of optimization performed by Eck and Hoppe reduces the error by generating new patches, which considerably augments the number of patches which represent the surface. The increment of the number of patches reduces the error because the regions to be adjusted are smaller and more geometrically homogeneous. In the method proposed in this paper, the optimization process is focused on improving the adjustment for every patch by modifying only its parameterization (control points weight). Because of that, the number of patches does not augment after optimization process. The final number of patches which represent every object is determined by the number of critical points obtained in an eigenvector associated with the eigenvalue (λ) selected from the solution system of the Laplacian matrix, and it does not change at any stage of the process.

Figure 2 contains a couple of objects (foot and skidoo) reported by Eck and Hoppe. Every object is shown triangulated starting with the points cloud. The triangulation is then adjusted with a patch cloud without optimizing and the result obtained after optimization. The adjustment with the method proposed in this paper, represents each object, with 27 and 25 patches, while Eck and Hoppe use 156 and 94 patches. This represents a reduction of 82% and 73% fewer patches respectively, in our work.

With respect to the reduction of the obtained error in the optimization process in each case, with the proposed method in this paper, the error reduces an average of 77%, a value obtained in an experimental test with 30 range images. Among these appear the images included in Figure 2. The error reported in Eck and Hoppe for the same images of Figure 2 allow a error reduction of 70%. In spite of this difference which is given between our method with respect to Eck and Hoppe's method, we should emphasize that error metrics are not the same, Eck and Hoppe's method is a measurement of RMS, ours method corresponds to an average of distances of projections of points on the surface.

Another aspect to be considered in the method comparison is the number of patches required to represent the object's surfaces. In Eck's work, the number of patches used to represent the object's increase is an average of 485% in relation to the initial quadrilaterization, while in the method proposed in this paper, the number of patches to represent the surface without optimization, and the optimized one, is constant.

6 Conclusion

The methodology proposed in this paper for the automation of reverse engineering of free-form three-dimensional objects has a wide application domain, allowing adjustment of surfaces regardless of topological complexity of the original objects.

A novel method for fitting triangular mesh using optimized NURBS patches has been proposed. This method is topologically robust and guarantees that the complex base be always quadrilateral creating a network of surfaces which is compatible with most commercial CAD systems.

In the proposed algorithm, the NURBS patches are optimized using multiple evolutionary strategies to estimate the optimal NURBS parameters. The resulting NURBS are then joined, guaranteing C^1 continuity. The formulation of C^1 continuity presented in this paper can be generalized, because it can be used to approximate regular and irregular neighborhoods which present model processes regardless of partitioning and parametrization.

References

[1] Arge A. Approximation of scattered data using smooth grid functions. *Comp. App.l. Math.*, 59:191–205, 1995.

[2] Yvart A., Hahmann S., and Bonneau G. Smooth adaptive fitting of 3-d models using hierarchical triangular splines. pages 13–22, Boston, USA, 2005.

[3] Baxter B. *The Interpolation Theory of Radial Basis Functions*. PhD thesis, Trinity College, University of Cambridge, UK, 1992.

[4] Gregorski B., Hamann B., and Joy D. Reconstruction of b-spline surfaces from scattered data points. pages 163–170, Geneva, Switzerland, June 2000.

[5] Loop C. Smooth spline surfaces over irregular meshes. pages 303–310, Orlando, USA, July 1994.

[6] Weber G., Scheuermann G., Hagen H., and Hamann B. Exploring scalar fields using critical isovalues. pages 171–178, Boston, USA, October 2002.

[7] Hoppe H. *Surface Reconstruction From Unorganized Points*. PhD thesis, Washington University, USA, 1994.

[8] Park I., Lee S., and Yun I. Constructing nurbs surface model from scattered and unorganized range data. pages 312–320, Ottawa, Canada, October 1999.

[9] Peters J. Constructing c^1 surfaces of arbitrary topology using bicuadric and bicubic splines. *Designing Fair Curves and Surfaces*, 277-293, 1994.

[10] Bertram M., Tricoche X., and Hagen H. Adaptive smooth scattered-data approximation for large-scale terrain visualization. *Proc. Symp. on Data Visualisation*, pages 177–184, May 2003.

[11] Eck M. and Hoppe H. Automatic reconstruction of b-spline surface of arbitrary topological type. *Proc. 23rd International Conference on Computer Graphics and Interactive Techniques*, pages 325–334, August 1996.

[12] Boulanger P. Triangulating nurbs surfaces, curve and surface design. Technical report, Vanderbilt University Press, Nashville, Tennesee, USA, 2000.

[13] Hartley R. and Zisserman A. *Multiple view geometry in computer vision*. Cambridge University Press, second edition edition, 2003.

[14] Dong S., Bremer P., Garland M., Pascucci V., and Hart J. Quadrangulating a mesh using laplacian eigenvectors. Technical report, University of Illinois, USA, 2005.

[15] Krishnamurthy V. and Levoy M. Fitting smooth surfaces to dense polygon meshes. *Proc. 23rd International Conference on Computer Graphics and Interactive Techniques*, pages 313–324, August 1996.

[16] Ni X., Garland M., and Hart J. Fair morse functions for extracting the topological structure of a surface mesh. *ACM Trans. Graph.*, 23(3):613–622, 2004.

[17] Ni X., Garland M., and Hart J. Simplification and repair of polygonal models using volumetric techniques. *Proc. SIGGRAPH*, TOG 23,3,613-622, 2004.