Simple Method for Constructing NURBS Surfaces from Unorganized Points

Nallig Leal, Esmeide Leal, John William Branch

Universidad Autónoma del Caribe
Calle 90 No 46 – 112
Tel. 57-5-3671286
Barranquilla – Colombia – Suramérica
{nleal,esleal}@uac.edu.co

Universidad Nacional de Colombia – Sede Medellín
Medellín – Colombia – Suramérica
Carrera 80 No 65-223
jwbranch@unalmed.edu.co

Summary. In this paper, a new method for constructing NURBS surfaces from scattered and unorganized points is presented. The method is easy to implement and allows fitting a NURBS surface to a scattered point cloud without constructing either NURBS patches networks or polygon meshes. Based on the projection of the points onto the regression plane of the cloud, the method detects both regions where the cloud lacks points, and regions where the cloud is too dense. Then, by inserting and removing points, the point cloud is regularized and approximated by a NURBS surface. In order to reduce the approximation error, an evolutionary strategy obtains the weights of the NURBS surface so that the distance between the point cloud and the NURBS is minimal. Finally, the points inserted outside the bounds of the point cloud are removed by trimming the NURBS surface. Numerical and graphical results are provided, demonstrating that the method produces valid surfaces with low fitting error.

Keywords: Surface fitting, NURBS surfaces, Points regularization, PCA, Evolutionary Strategies.

1. Introduction

3D reconstruction is a process by which objects are reproduced in the computer memory, keeping its physical characteristics (dimensions, volume and shape). 3D reconstruction is a difficult task that covers, in general terms, five stages: data acquisition, registration, integration, segmentation, and surface fitting [1]. The approach presented in this paper deals with the surface fitting stage, in which the computational model of the object is obtained.

NURBS is one of the most employed surface fitting models, provided that it is a standard representation of curves and surfaces [2] and is widely supported by modern standards like OpenGL and IGES, which are used for graphics and geometric data exchange [3]. In addition, the NURBS surface model has stability, flexibility, local modification properties and is robust to noise. Yet, the NURBS surface model has a disadvantage: the input data points should be mapped on a regular grid structure [4].

In the 3D reconstruction process, the registration and integration stages produce massive scattered and unorganized point clouds that cannot be mapped on a regular grid structure. Such point clouds cannot be fitted by a NURBS surface and are not suitable for usage in computer-aided design (CAD) systems [5].

In order to fit a NURBS surface to an unorganized and scattered point cloud, several approaches have been presented [4, 5, 6, 7, 8, 9]. Such approaches fit to the cloud a network of NURBS patches with some degree of continuity between them.

The construction of the network requires constructing polygon meshes and complicated procedures of refinement, reparameterization and maintainability of the continuity between the patches, which is computationally expensive in terms of memory and processing.

In this paper a new method for constructing NURBS surfaces from scattered and unorganized points is presented. In contrast with others, our method does not need to construct a network of NURBS patches. Furthermore, previous construction of polygon meshes, mesh refinement and data reparameterization are not required.

Our method first detects the global bias of the point cloud fitting a regression plane to it by using weighted principal components analysis. Then, all the points are projected onto the plane and a two-dimensional regularity analysis of the point cloud is made. The analysis consists of detecting regions with low point density and regions with high point density. By inserting and removing points, based on the two-dimensional analysis, the point cloud is regularized. In order to reduce the fitting error, an evolu-

tionary strategy obtains the weights of the control points that belong to the cloud. Finally, the points inserted outside the cloud boundary are removed by trimming the NURBS surface.
The main contribution of our work is that we proposed an automatic, efficient and simple method for constructing a complete NURBS surface from a point cloud.

The remainder of this paper is organized as follows. In section 2, related work dealing with reconstruction of NURBS surfaces from scattered and unorganized points is presented. In section 3, the fundamentals of NURBS surfaces are presented. In section 4, a short overview of evolutionary strategies is presented. In section 5, the stages of our method are explained. In section 6, results of our method are provided. In section 7, conclusions and future work are discussed.

2. Related Work

Point clouds are considered the main information source in the 3D reconstruction process; unfortunately, such point clouds are not suitable for integration into CAD systems [5]. Constructing NURBS surfaces from point clouds would allow the incorporation of such information source in CAD systems.

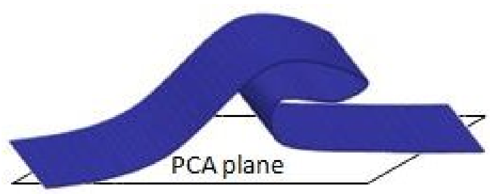

Fig. 1. Occluded surface with respect to its PCAP.

Several approaches have been proposed for solving the problem of constructing NURBS surfaces from scattered and unorganized points. Eck and Hoppe [6] solved the problem by generating a network of B-Spline patches. They first construct a triangular mesh over the cloud and project the points onto the mesh to obtain an initial parameterization. Then, a merging process from triangular mesh is carried out for constructing a quadrilateral mesh. Over the quadrilateral domain, a network of B-Spline patches is built. In order to reach a user-specified error tolerance, a refinement process takes place. This process adaptively subdivides the quadrilateral domain producing smaller quadrilateral subfaces. A new network of B-Spline patches is necessary to fit the refined surface. Even though this is an effective method for reconstructing B-Spline surfaces, e.g. NURBS, from point clouds, it is computationally expensive in terms of memory and processing needed. Furthermore, only G1 continuity between the patches is guaranteed.

Krishnamurthy and Levoy [7] proposed an approach that constructs a polygon mesh which is resampled to produce a regular grid where NURBS surface patches can be fitted. The performance of the approach is poor when it operates on complex surfaces, and it cannot be applied over surfaces with holes.

Park, Yun and Lee [4] presented a two-stage algorithm. The initial stage, named model approximation, employs K-means clustering to obtain regions where polygon meshes are constructed, subdivided in triangular meshes, and blended in quadrilateral meshes. The meshes produced by the initial stage are represented by a hierarchical graph structure. The second stage takes the hierarchical graph structure to construct a NURBS patch network. This method, like Eck and Hoppe's method, is computationally expensive and just guarantees G1 continuity. Furthermore, it assumes the point cloud represents a closed surface.

Gregorski, Hamann and Joy [5] proposed an approach that subdivides the set of points into a strip tree structure. This structure is used to fit quadratic surfaces which are degree-elevated and blended into B-Spline patches. This approach cannot be applied either on closed surfaces or on surfaces that are occluded with respect to its principal component analysis regression plane (PCAP), like the surface shown in Figure 1.

Bertram, Tricoche and Hagen [8], and Yvart, Hahmann, and Bonneau [10] proposed approaches that use triangular B-Spline patches instead of NURBS patches to fit point clouds. Their approaches, like the aforementioned ones, construct polygon meshes and use expensive processes for fitting a network of triangular B-Spline to the point cloud.

3. NURBS

NURBS, Non Uniform Rational B-Splines, are parametric tensor product curves or surfaces defined by the following expression

$$S(u,v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j}} \tag{1}$$

where $w_{i,j}$ are the weights, $P_{i,j}$ are the control points and $N_{i,p}(u), N_{j,q}(v)$ are the B-Spline basis functions of order p and q respectively, defined over the non periodic node support $S_u = \{u_0,...,u_r\}$ and $S_v = \{v_0,...,v_r\}$ [11], which can be calculated in a recursive way by the Cox and de Boor formula [12] according to (2)

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{in other cases} \end{cases} \tag{2}$$

$$N_{i,p}(u) = \frac{(u-u_i)N_{i,p-1}(u)}{u_{i+p}-u_i} + \frac{(u_{i+p+1-}u)N_{i+1,p-1}(u)}{u_{i+p+1}-u_{i+1}}$$

A NURBS surface is completely determined by its control points $P_{i,j}$, i.e. the surface changes in a predictable way according to control points movement. This is known as the local support property and allows the surface to be affected, only locally, by the movement of a control point. The main difficulty when fitting NURBS surfaces is to obtain a suitable parameterization and automatically choose the number of control points and their positions to define the surface topology.

The weighting factors $w_{i,j}$ of NURBS surfaces play an important role in the fitting process, since these factors determine how much a control point influences the shape of the surface locally. When the weighting factors of NURBS surfaces are assigned in a homogeneous way and their values are one, the NURBS model is reduced to a particular case known as B-Spline surfaces, which are limited in the representation of free-form and conic surfaces. If we want an approximately close enough set of data that represents free-form surfaces using NURBS, it is necessary to manipulate the NURBS parameters, but as previously mentioned such manipulation implies dealing with non linear problems during the fitting process.

Furthermore, negative values or zeros in the weighting factors can degenerate the construction of the surface.

Figure 2 illustrates the importance of the weighting factors in the NURBS model. The circles represent control points and the line represents a NURBS curve. It is notable how the weighting factors affect the local geometry of the surface. Figures 2a to 2d show the effect over the NURBS curve of assigning the values 0, 0.5, 1 and 2, respectively, to control point 3.

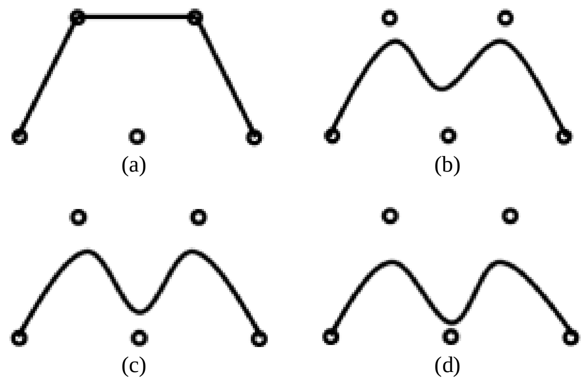

Fig. 2. Weighting factors effect.

When fitting data points using NURBS, we attempt to minimize:

$$\delta = \sum_{l=1}^{np} \left(z_l - \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j}} \right)^2 \tag{3}$$

where np is the number of control points. If the number of knots and their positions are fixed, the set of weighting factors is known and only the control points $\left\{\left\{P_{i,j}\right\}_{i=1}^{n}\right\}_{j=1}^{m} \in R$ are considered during the optimization of (3), we have a linear problem of least squares. But if the knots or the weights are unknown, it will be necessary to solve a non linear problem. In many applications the knots location is not necessary; therefore, knots values are obtained using some heuristic techniques.

4. Evolutionary Strategies

Evolutionary Strategies (ES) were developed in 1964 by Rechenberg and Schwefel at the University of Berlin as an experimental optimization technique [13]. ES try to imitate, in contrast with Genetic Algorithms, the effects of the genetic procedures in the phenotype. ES belong to a kind of probabilistic numerical optimization algorithm, which include Evolutionary Programming (EP), Genetic Algorithms (GA) and Genetic Programming (GP), which are known as Evolutionary Algorithms.

The first variant of ES, called $(1+1)-ES$, works based on two only individuals, a parent and a descendant per generation. The descendant is created by applying variations, called mutations, binomially distributed (with mean equal to zero and variance σ^2) to the parent. The descendant can be the parent in the next generation if it is better than the parent;if the opposite is true, the parent will be the survivor for the next generation.

$(1+1)-ES$ was replaced by $(\mu+\lambda)-ES$ and $(\mu,\lambda)-ES$ variants, with $\mu>1$ parents and $\lambda>1$ descendants per generation. In these new variants, the recombination concept was introduced, in order to create individuals as the cross of the parent attributes. After mutation and the individuals evaluation, the descendants replace the parents if the former are better than the latter. Depending on the selection type, μ new individuals are selected only from the descendant population $(\mu,\lambda)-ES$ or μ new individuals are selected from the parents and the descendant $(\mu+\lambda)-ES$. Beside mutation and recombination, $(\mu+\lambda)-ES$ and $(\mu,\lambda)-ES$ control the size of the mutation step by an auto-adaptation process that learns the mutation step size, and optionally the covariance, during the evolutionary searching process [13]. ES use three main operators for changing the population until a stop criterion is reached.

Recombination: new individuals are produced by crossing the information contained in the parents. Depending on the individual variable representation some algorithms can be applied for recombination purposes: discrete recombination, local intermediate recombination, global intermediate recombination, point crossover and n-point crossover. The recombination operator allows for the exploration of the searching space.

Mutation: After recombination, the descendants are changed with a probability p, by introducing small variations known as mutations. Mutation allows for the introduction of new possible solutions and the exploitation near to a given solution (individual). Mutation follows the scheme given in (4) and (5).

Selection: choose the best individuals of the population according to a fitting criterion.

$$\sigma_i' = \sigma_i e^{(\tau_0 \cdot N(0,1) + \tau_i \cdot N_i(0,1))} \tag{4}$$

$$x_i' = x_i + \sigma_i' \cdot N_i(0,1) \tag{5}$$

where $N(0,1)$ is a random variable normally distributed with mean 0 and variance 1, and τ_0, τ_i are constants that control the mutation step.

5. NURBS fitting to unorganized points

Our method for constructing NURBS surfaces from scattered and unorganized point clouds is roughly made up of two stages. The first one regularizes and fits the NURBS surface. The second one optimizes the NURBS surface fitting and trims the surface in order to eliminate points inserted outside the cloud during the regularization process. The following subsections explain each of these stages.

5.1. Point cloud regularization

The regularity analysis of the point cloud takes place in a two-dimensional space. Such space is given by the principal components analysis regression plane (PCAP), which is expanded along the directions of higher dispersion of the cloud. The regularization process consists of the following steps:

1. Project the points of the cloud onto the PCAP and obtain the extreme projected points.
2. Construct a grid from the extreme points with density ρ calculated according to (6)

$$\rho = \frac{P}{A} \tag{6}$$

   where P is the number of points and A is the area of the ellipse determined by the extreme points.

3. For each grid element, do steps 4 to 5.
4. If no point is present, insert one point in its centroid, according to (7) if the inserted point is inner to the cloud (Figure 3) or according to (8) if the inserted point is outer to the cloud (Figure 4).

$$P_{ins} = AVG_{Neigh} + P_{grid} - Pj(AVG_{Neigh}, Plane_{PCA}) \tag{7}$$

$$P_{ins} = P_{ext} + \lambda \vec{B} \tag{8}$$

where AVG_{Neigh} is the neighborhood mean of the point for inserting, P_{grid} is the centroid of the empty element analyzed, $Plane_{PCA}$ is the PCAP, Pj is the projection function, P_{ext} is the point of the cloud closest to the point for inserting, λ is the distance between them and $\vec{B}$ is the PCAP basis vector in the inserting direction.

5. If there are two or more points in the element, remove the points located farthest from the centroid.

Once the regularization process has been finished, a NURBS surface can be fitted to the entire cloud. In order to reduce the fitting error of the surface, due to the smoothing properties of NURBS, an optimization process is carried out for preserving the sharp features of the original point cloud.

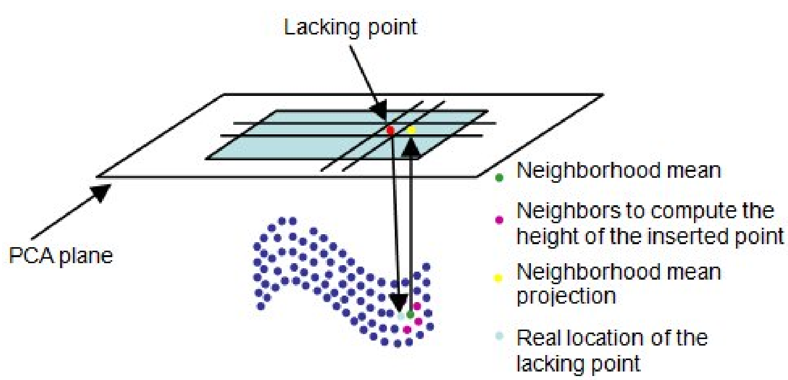

Fig. 3. Insertion of a point inside the cloud

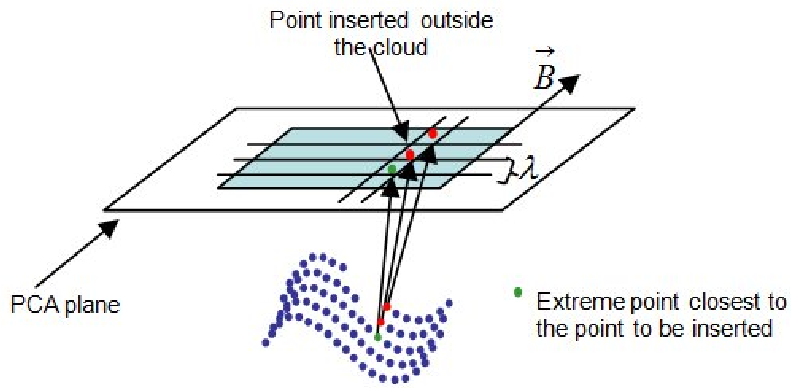

Fig. 4. Insertion of a point outside the cloud

5.2. Surface Optimization

The surface optimization process is devised to maintain the influence of the original point cloud stronger than the points inserted, so that the topology of the cloud is not affected. Our optimization process is carried out by a $(\mu+\lambda)-ES$. It can be described as follows.

Let $P=\{p_1, p_2, \ldots, p_n\}$ be a set of 3D points sampled from a real object, e.g. the regularized points, and $S=\{s_1, s_2, \ldots, s_m\}$ be a NURBS surface that approximates P. Our problem consists of minimizing the approximation error given by (9).

$$E(S)=d_{P,S}<\delta \tag{9}$$

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. The evolutionary strategy obtains the control point weights of S, so that (9) is true.

The evolutionary strategy will only obtain the weights of the points that belong to the point cloud. The weights of the inserted points will be assigned as follows: i) if the point was inserted inside the cloud, its weight will be the mean of the weights of the neighbor points. ii) If the point was inserted outside the cloud, its weight will be zero.

Since the influence of the NURBS surface control points is only local, the sampled points P will be divided into clusters where a local optimization process will be carried out, 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 k-means.

The objective of the k-means is to find homogeneous regions where the optimization process can be run without distorting the local shape of the surface. By the last run of the k-means clustering algorithm, it is expected to have found the homogeneous regions where the optimization process was run, as shown in Figure 5.

Once P is clustered an evolutionary strategy $(\mu+\lambda)-ES$ will optimize the local fitting of the NURBS in each cluster. At the boundaries of the clusters, the geometry of the surface is not affected since the evolutionary strategy only modified local weights of the NURBS surface control points, it does not modified de points. The evolutionary strategy configuration is as follows:

Individuals: the individuals of the strategy are conformed by the weights of the cluster points and the mutation steps σ, as shown in Figure 6 (where w_i are the control point weights and σ_i are the mutation step sizes).

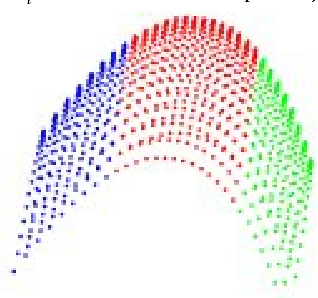

Fig. 5. Clusters found by k-means.

Mutation operator: uncorrelated mutation with n mutation step sizes σ is applied to the individuals, according to (4) and (5).

Recombination operator: the recombination operator is different for object variables w_i and parameters σ_i. A global intermediary recombination is applied to object variables, according to (10), whereas a local intermediary recombination is applied to mutation step sizes σ_i, according to (11).

Selection operator: the best individuals according to the aptitude function given in (9). In order to perform a fast computation of the distance between the points P and the NURBS surface S, the points of S are stored in a kd-tree structure, so that the searching process for finding the nearest points between P and S is of $\log(n)$ order.

$$b'_i = \frac{1}{\rho}\sum_{k=1}^{\rho} b_{k,i} \qquad (10)$$

$$b'_i = u_i b_{k_1,i} + (1 - u_i) b_{k_2,i} \tag{11}$$

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].

w_1	w_2	$\cdots$	w_n	σ_1	σ_2	$\cdots$	σ_n

Fig. 6. Individual of the evolutionary strategy.

The algorithm in Table I summarizes the optimization process. After the optimization process, the optimized surface is trimmed to eliminate the points inserted outside the cloud.

Table 1. Optimization process

```
Perform a clustering of P by using k-means
For each cluster do
  Set individual size = cluster size
  Set population size = μ
  Initialize the population randomly
  Evaluate the population in the aptitude function (8)
  While the stop criterion δ has not been reached do
    For i = 1 to λ·0.9 do
      Ind_i = mut(Population_rand(1,μ))
    End for
    For i = 1 to λ·0.1 do
      Ind_i = rec(Population_rand(1,μ))
    End for
    Population = select from(μ + λ)
  End while
End for
```

6. Experimental results

The proposed method was evaluated in scanned data. Two different models are used to show the method results. The Face model, obtained from Alexander Belyaev's web site from the Heriot-Watt University (www.eps.hw.ac.uk/~belyaev), illustrate the complete NURBS constructing process. The Angel model, obtained from Ohio State University, illustrates the results of the optimization process. The method was implemented in a 1.4GHZ Pentium M with 512MB of RAM.

The first stage, point cloud regularization, takes seven seconds to process 15K points. We first project the cloud onto its PCAP, as shown in Figure 7(b); then, the two-dimensional analysis takes place for inserting and removing points where necessary. The two-dimensional regularization is shown in Figure 7(c) (inner and outer inserted points are shown in red and green color respectively), whereas the three-dimensional (point cloud) regularization is shown in Figure 7(d). The NURBS surface fitted to the regularized point cloud is shown in Figures 7(e).

The second stage, surface optimization, takes 3 minutes for processing 15K points. After 10 generations average, the evolutionary strategy reached the minimum, i.e. the distance between P and the optimized NURBS surface S reached an average of 14% less than the distance between P and the non optimized NURBS surface. Figure 8 shows the effectiveness of the proposed method application over complex surfaces. The profiles presented in Figures 8(a) – 8(c) show the complexity of the surface. Figure 9 show the improvement of the sharp features obtained with our method. In Figure 9(b) the NURBS surface points follow the control points (red circles) closer than in Figure 9(a), which improves sharp feature representation.

To verify that the shape of the original point cloud was not distorted, two metrics were defined. i) The relative error E_{bdl} between the diagonal length of the bounding box of P and the diagonal length of the bounding box of S. ii) The normalized modeling error E_{avg} , according to (12), is given in [4]. In our tests, the relative error was 0.031% and the modeling error was 0.01,

$$D_{avg} = \frac{\sum_{i=0}^{N-1} |d_i|}{N}$$
$$E_{avg} = \frac{D_{avg}}{L} \tag{12}$$

where d_i and N denote the signed distance from the data x_i and the number of the total data, respectively. L is the bounding box length. In Table II, the execution time of our method stages are summarized. In Table III, some statistics of the optimization process are presented.

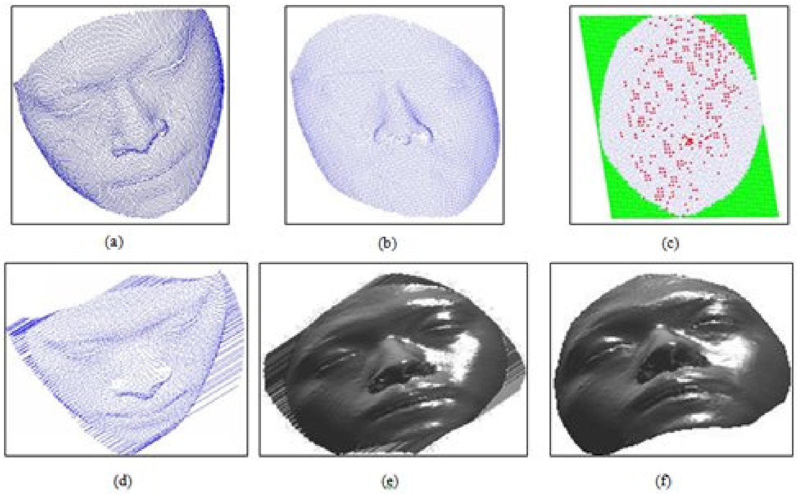

Fig. 7. Stages of the proposed method evaluated on scanned data. (a) Original point cloud (b) Point cloud projected onto its PCAP (c) Regularized projection (d) Regularized point cloud (e) NURBS surface constructed from the regularized point cloud (f) Trimmed and optimized NURBS surface.

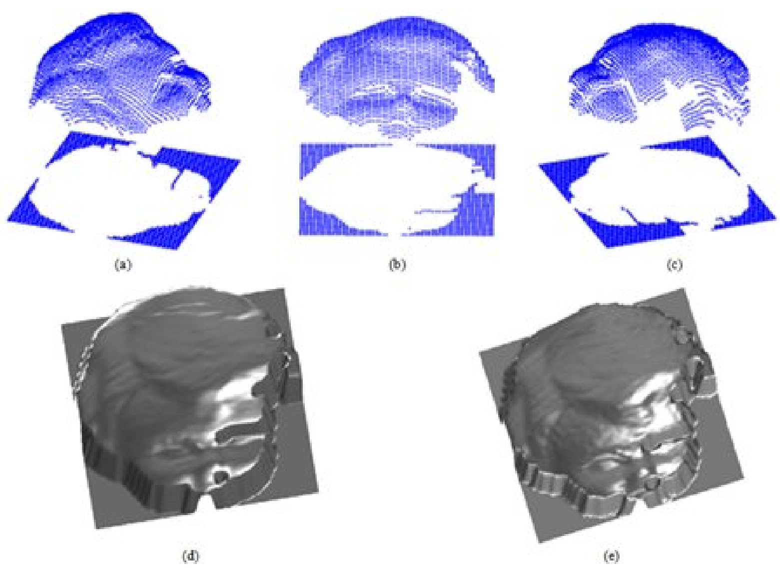

Fig. 8. NURBS fitted over a complex surface applying the proposed method. (a) Right profile of the original dataset (b) Frontal profile of the original dataset (c) Left profile of the original dataset (d) Non optimized NURBS surface (d) Optimized NURBS surface.

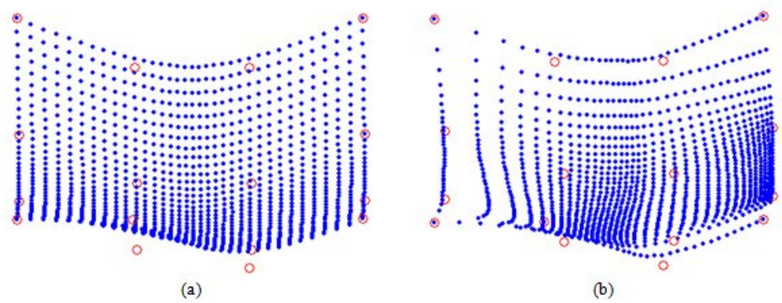

Fig. 9. Detail of sharp features. (a) Detail of a Non optimized NURBS surface fitted to a set of control points. (b) Detail of an optimized NURBS surface fitted to a set of control points.

Table 2. Execution time of the method stages

Stage	Execution Time
Regularization	7 seconds
Optimization	3 minutes

Table 3. Statistics of the optimization process

Processed images	30
Average points per image	15K
Average points per cluster	854
Tests per image	12
Time for clustering	5 seconds
Time for optimizing	3 minutes
Average generations per test	10
μ	5
λ	35
Distance reduction	14%
Relative error	0.031%
Modeling error	0.01

7. Conclusions and future work

A new simple method for constructing NURBS surfaces from scattered and unorganized point clouds was presented. Both our method applicability in regular and irregular surfaces and the effectiveness of the method improving NURBS surface sharp feature representation were shown. It was demonstrated that it is possible to fit complete NURBS surfaces to point clouds, without previous construction of either polygon meshes or NURBS patch networks; as a result, the proposed method is computationally efficient.

Our method cannot be applied over surfaces occluded with regard to its principal component analysis regression plane, like the surface shown in Figure 1. Occluded surfaces with regard to its principal component analysis regression plane generate collisions in step one of the regularization process and such collisions are eliminated by deleting points, which can eliminate a complete segment of the surface.

The optimization method could be used for optimizing approaches that use NURBS patches. In those approaches, the clustering process would not be necessary, since the optimization process would be carried out for each NURBS patch.

In order to apply our method over occluded surfaces, some future work can point to that direction. The work of Tenenbaum, De Silva and Langford [14] can be used to analyze the cloud regularity in a parametric space without loosing information, which allows constructing the NURBS surface over closed and occluded surfaces. Some additional future work could focus on i) detecting holes in the cloud before regularizing and then trimming them after the regularization process, and ii) improving the optimization process by establishing automatically the number of clusters where the evolutionary strategy is run

References

[1] A. Myers. Introductory literature review surface reconstruction from three dimensional range data. *Technical report*, The University of Adelaide, Department of Computer Science. 1999

[2] L. Piegl and W. Tiller. *The NURBS Book*, Springer-Verlag. 1995.

[3] M. Ristic. Efficient fitting of Non-Uniform Rational B-Spline surfaces using non-organized 3D data. In *SPIE'S. Vol. 11. No. 1*. December. 2000.

[4] I. Park, I. Yun, and S. Lee. Constructing NURBS Surface Model from Scattered and Unorganized Range Data. In *Second International Conference on 3-D Digital Imaging and Modeling*. 1999.

[5] B. Gregorski, B. Hamann, and K. Joy. Reconstruction of B-spline Surfaces From Scattered Data Points. In *Proceedings of Computer Graphics International.* 2000.

[6] M. Eck and H. Hoppe. Automatic Reconstruction of B-Spine Surfaces of Arbitrary Topological Type. In *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques.* ACM Press. 1996.

[7] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *SIGGRAPH 96 Conference Proceedings*. ACM SIGGRAPH, Addison Wesley. 1996.

[8] M. Bertram, X. Tricoche, and H. Hagen. Adaptive Smooth Scattered-data Approximation for Large-scale Terrain Visualization. In *EUROGRAPHICS - IEEE TCVG Symposium on Visualization*. 2003.

[9] X. Shi, T. Wang, P. Wu, and F. Liu. Reconstruction of convergent g^1 smooth b-spline surfaces. In *Computer Aided Geometric Design. Vol. 21*. November. 2004.

[10] A. Yvart, S. Hahmann, and G. Bonneau. Smooth Adaptive Fitting of 3D models using hierarchical triangular splines. In *Shape Modeling International*. 2005.

[11] J. Cordero and J. Parejo. Curvas y Superficies para Modelado Geométrico. 1st ed, Alfa Omega, España, 2002.

[12] C. De Boor. On Calculating With B-Splines. In J. Approximation Theory, Vol. 6, No. 1, July, pp. 50-62. 1972.

[13] T. Bäck, and H. Schwefel. Genetic Algorithms in Engineering and Computer Science. John Wiley & Sons Ltd. 1995.

[14] J. Tenenbaum, V. De Silva, and D. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. In *Science 290(5500):2319 – 2323*. December 2000.