

# AUTOMATIC CONSTRUCTION OF NURBS SURFACES FROM UNORGANIZED POINTS

## RECONSTRUCCIÓN AUTOMÁTICA DE SUPERFICIES NURBS A PARTIR DE PUNTOS NO ORDENADOS

NALLIG EDUARDO LEAL NARVÁEZ

*Facultad de Ingeniería, Universidad Autónoma del Caribe, Barranquilla, nleal@uac.edu.co*

ESMEIDE ALBERTO LEAL NARVÁEZ

*Facultad de Ingeniería, Universidad Autónoma del Caribe, Barranquilla, esleal@uac.edu.co*

JOHN WILLIAN BRANCH

*Facultad de Minas, Universidad Nacional de Colombia, Medellín, jwbranch@unal.edu.co*

Received for review May 25 th, 2010, accepted November 23th, 2010, final version November, 25 th, 2010

**ABSTRACT:** Modeling with Non Uniform Rational B-Splines (NURBS) surfaces has become a standard in CAD/CAM systems due to its stability, flexibility, and local modification properties. The advantage of fitting with NURBS surfaces is well known, but it is also known that NURBS surfaces have several deficiencies. A NURBS surface cannot be fitted over an unorganized and scattered set of points and the representation of sharp features like edges, corners, and high curvatures is poor. This paper presents a new method for fitting a NURBS surface over an unorganized and scattered cloud of points, preserving its sharp features. In contrast with other methods, ours does not need either to construct a network of NURBS patches or polygon meshes. By reducing the dimensionality of the point cloud using ISOMAP algorithms, our method detects both regions with lacking points, and regions where the cloud is too dense. Then, the cloud is regularized by inserting and removing points, and it is approximated by a NURBS surface. An evolutionary strategy obtains the weights of the NURBS surface in order to improve the representation of sharp features.

**KEYWORDS:** NURBS, ISOMAP, evolutionary strategies.

**RESUMEN:** El modelamiento con Superficies B-Splines Racionales no Uniformes (NURBS) se ha convertido en un estándar en los sistemas CAD/CAM debido a su estabilidad, flexibilidad y propiedades de modificación local. Las ventajas de ajustar con superficies NURBS son bien conocidas, aunque también son conocidas las limitaciones que éstas presentan. Una superficie NURBS no puede ser ajustada sobre un conjunto de puntos dispersos no ordenados. Adicionalmente, la representación de detalles finos como aristas, esquinas y altas curvaturas, es pobre. Este artículo presenta un nuevo método para ajustar superficies NURBS sobre conjuntos de puntos dispersos no ordenados, preservando los detalles finos. A diferencia de otros métodos, el nuestro no necesita construir una red de parches NURBS ni mayas poligonales. Para reducir la dimensionalidad de la nube de puntos usando el algoritmo ISOMAP, nuestro método detecta regiones con carencia de puntos y regiones donde la nube es muy densa; luego, la nube es regularizada por inserción y remoción de puntos, para finalmente ser ajustada por una única superficie NURBS. Para mejorar la representación de los detalles finos, una estrategia evolutiva obtiene los pesos de la superficie NURBS.

**PALABRAS CLAVE:** NURBS, ISOMAP, estrategias evolutivas.

### 1. INTRODUCTION

NURBS is one of the most employed surface fitting models, provided that it is a standard representation of curves and surfaces [1]; and is widely supported by modern standards like OpenGL and IGES, which are used for graphics and geometric data exchange [2]. In addition, the NURBS surface model has stability, flexibility, local modification properties; and is robust to noise. However, NURBS surfaces models has some disadvantages: the input data points should be mapped on a regular grid structure [3] and the representation of sharp features is poor [14].

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 [4].

In order to fit a NURBS surface to an unorganized and scattered point cloud, several approaches have been presented [3, 4, 5, 6, 7, 8]. 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.

Surface fitting is very important in fields like medicine, engineering, and scientific visualization, among others. However, to fit a surface to a set of sampled data, possibly noisy, always has the underlying problem of a lack of precision in the representation of sharp features like edges and peaks. Therefore, surface fitting processes must be followed in order for an optimization process to improve the representation. Several approaches for optimizing NURBS surfaces have been presented [15, 16, 17, 18].

In this paper, a new method for constructing and optimizing 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 uses ISOMAP algorithm for mapping the point cloud to a two-dimensional space to carry out an analysis of the regularity of the point cloud. 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 evolutionary 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 summary of the ISOMAP algorithm is described. In section 5, a short overview of evolutionary strategies and SOM is presented. In section 6, the stages of our method are explained. In section 7, the results of our method are provided. In section 8, 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 [4]. Constructing NURBS surfaces from point clouds would allow for the incorporation of such information source in CAD systems.

Several approaches have been proposed for solving the problem of constructing NURBS surfaces from scattered and unorganized points. Eck and Hoppe [5] 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 [6] 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 [3] 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 only guarantees G1 continuity. Furthermore, it assumes that the point cloud represents a closed surface.

Gregorski, Hamann, and Joy [4] 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 occluded surfaces, like the surface shown in Figure 1.

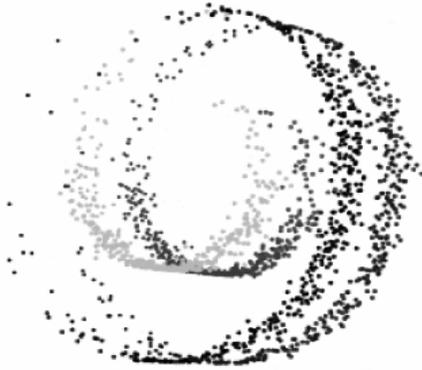


Figure 1. Occluded surface

Bertram, Tricoche, and Hagen [7], and Yvart, Hahmann, and Bonneau [9] 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 triangular B-Spline network to the point cloud.

### 3. RELATED WORK

NURBS, are parametric tensor product curves or surfaces defined by the following expression:

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

where  $w_{ij}$  are the weights,  $P_{ij}$  are the control points, and  $N_{ip}(u), N_{jq}(v)$  are the B-Spline basis functions of order  $p$  and  $q$  respectively, defined over the non periodic node support  $S_u = \{u_0, \dots, u_r\}$  and  $S_v = \{v_0, \dots, v_r\}$  [10], which can be calculated in a recursive way by the Cox and de Boor formula [11] according to (2)

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

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

It is notable how the weighting factors affect the local geometry of the surface. Figures 2a to 2c show the effect over the NURBS curve of assigning the values 0, 0.5, and 1, respectively, to control point 2.

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

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v)w_{i,j}} \quad (3)$$

A NURBS surface is completely determined by its control points  $P_{ij}$ , 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  $w_{ij}$  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.

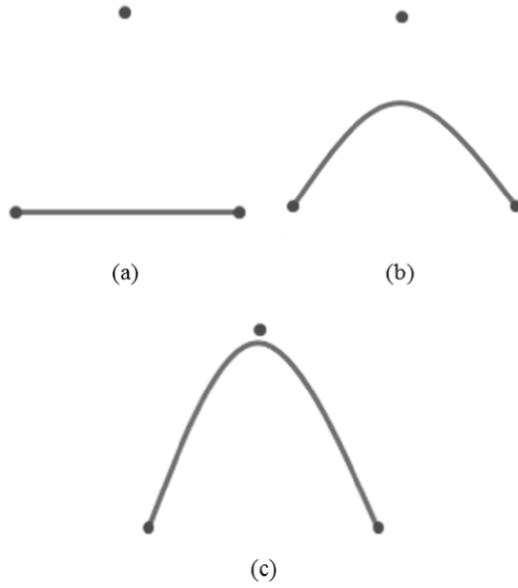
Furthermore, negative values or zeros in the weighting factors can degenerate the construction of the surface.

Figure 2 illustrates the importance of the weighting factors in the NURBS model. The dots 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 2c show the effect over the NURBS curve of assigning the values 0, 0.5, and 1, respectively, to control point 2.

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

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

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



**Figure 2.** Weighting factors effect.

points  $\{\{P_{ij}\}_{i=1}^n\}_{j=1}^m \in R$  are considered during the optimization of (3): we have a linear least-squares problem. 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. ISOMAP

ISOMAP is an isometric mapping method that extends MDS by incorporating geodesic distances in a weighted graph. ISOMAP algorithm takes as input the distances  $d_x(i, j)$  between all pairs  $i, j$  from  $N$  data points in the high-dimensional input space. The algorithm outputs coordinate vectors  $y_i$  in a  $d$ -dimensional Euclidean space that best represents the intrinsic geometry of the data. The only free parameter is  $\epsilon$  (radii of neighborhood) or  $K$  (nearest neighbors).

ISOMAP algorithm steps are summarized in [13] as follows:

##### *Step 1: Construct neighborhood graph*

Define the graph  $G$  over all data points by connecting points  $i$  and  $j$  if [as measured by  $d_x(i, j)$ ] they are closer than  $\epsilon$  ( $\epsilon$ -Isomap), or if  $i$  is one of  $K$ 's nearest neighbors of  $j$  ( $K$ -Isomap). Set edge lengths equal to  $d_x(i, j)$ .

##### *Step 2: Compute shortest paths*

Initialize  $d_G(i, j) = d_x(i, j)$  if  $i, j$  are linked by an edge;  $d_G(i, j) = \infty$  otherwise. Then for each value of  $k = 1, 2, \dots, N$  in turn, replaces all entries  $d_G(i, j)$  by  $\min\{d_G(i, j), d_G(i, k) + d_G(k, j)\}$ . The matrix of final values  $D_G = \{d_G(i, j)\}$  will contain the shortest path distances between all pairs of points in  $G$ .

##### *Step 3: Construct $d$ -dimensional embedding*

The final step applies classical MDS to the matrix of graph distances  $D_G = \{d_G(i, j)\}$ , constructing an embedding of the data in a  $d$ -dimensional Euclidean space  $Y$  that best preserves the manifold's estimated intrinsic geometry. The coordinate vectors  $y_i$  for points in  $Y$  are chosen to minimize the cost function, represented by (4).

$$E = \|\tau(D_G) - \tau(D_Y)\|_L^2 \quad (4)$$

where  $D_Y$  denotes the matrix of Euclidean distances  $\{d_Y(i, j) = \|y_i - y_j\|\}$  a  $\|A\|_L^2$  the  $L^2$  matrix

norm  $\sqrt{\sum_{i,j} A_{ij}^2}$ . The operator  $\tau$  converts distances to inner products.

Let  $\lambda_p$  be the  $p$ -th eigenvalue (in decreasing order) of the matrix  $\tau(D_G)$ , and  $v_p^i$  the  $i$ -th component of the  $p$ -th eigenvector. Then set the  $p$ -th component of the

$d$ -dimensional coordinate vector  $y_i$  equal to  $\sqrt{\lambda_p} v_p^i$

#### 5. evolutionary strategies and som

In this section, a brief background of Evolutionary Strategies and Self Organizing Maps is provided.

##### 5.1 Evolutionary strategies

Evolutionary Strategies ( $ES$ ) were developed in 1964 by Rechenberg and Schwefel at the University of Berlin as an experimental optimization technique [12].  $ES$  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 (GAs) and Genetic Programming (GP), which are known as Evolutionary Algorithms.

The first variant of  $ES$ , called  $(1+1)-ES$ , works based on only two 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  $\sigma^2$ ) 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  $\mu > 1$  descendants per generation. In these new variants, the recombination concept was introduced in order to create individuals as the crossing of parent attributes. After mutation and the individual's evaluation, the descendants replace the parents if the former are better than the latter. Depending on the selection type,  $\mu$  new individuals are selected only from the descendant population  $(\mu,\lambda)$ -ES or  $\mu$  new individuals are selected from the parents and the descendants  $(\mu+\lambda)$ -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 [12]. 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 (5) and (6).

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

$$\sigma_i' = \sigma_i e^{(\tau_0 \cdot N(0,1) + \tau_i \cdot N_i(0,1))} \quad (5)$$

$$x_i' = x_i + \sigma_i' \cdot N_i(0,1) \quad (6)$$

where  $N(0,1)$  is a random variable normally distributed with mean 0 and variance 1, and  $\tau_0, \tau_i$  are constants that control the mutation step.

## 5.2 Som

Artificial neural networks generally consist of simple computational elements called neurons which are highly interconnected to each other. The neurons are arranged in layers that interconnect to other layers. A neuron can receive connections from neurons of other layers and even from neurons of the same layer to which it belongs. SOM (Self Organizing Map, also known as Kohonen network) is a one single layer neural network, denoted by  $\{n_1, n_2, \dots, n_m\}$  where, in its basic configuration, neurons are arranged two-dimensionally, and have some neighboring neurons.

The input patterns to the network is a set of n-dimensional vectors denoted by  $\{x_1, x_2, \dots, x_n\}$ . Each neuron of the network has a weight vector which has the same dimensionality the input patterns have. Basically, SOM is an unsupervised neural network, even though there are several supervised variants known as Learning Vector Quantization (LVQ). The unsupervised SOM is employed for clustering. go to [19] for details.

## 6. 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

### 6.1 Point cloud regularization

The regularity analysis of the point cloud takes place in a two-dimensional space. Such space is given by ISOMAP. The regularization process consists of the following steps:

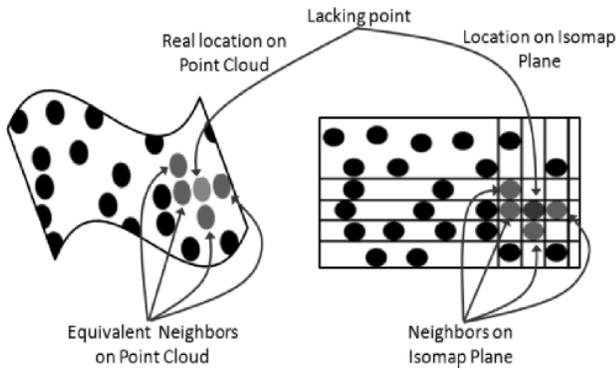
1. Run ISOMAP algorithm on a point cloud and obtain the extreme points.
2. Construct a grid from the extreme points with density  $\rho$  calculated according to (7)

$$\rho = \frac{P}{A} \quad (7)$$

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 as follows: choose the four closest points to the centroid of the cell and compute the equivalent location on the point cloud, using the four correspondent points on the cloud, like Figure 3 shows.
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.



**Figure 3.** Insertion of a point in the cloud

## 6.2 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 (1+1)-ES. It can be described as follows:

Let  $P = \{p_1, p_2, \dots, p_n\}$  be a set of 3D points sampled from a real object, e.g. the regularized points, and  $S = \{s_1, s_2, \dots, s_m\}$  be a NURBS surface that approximates  $P$ . Our problem consists of minimizing the approximation error given by (8).

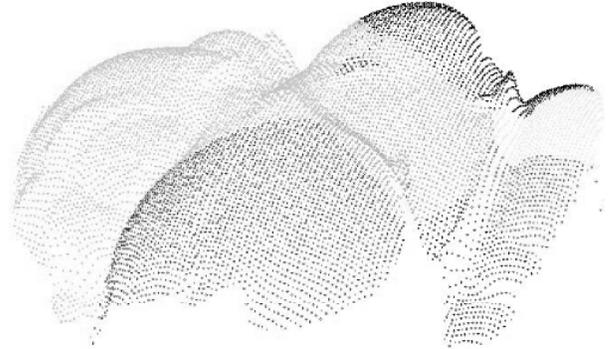
$$E(S) = d_{P,S} < \delta \quad (8)$$

where  $d_{P,S}$  is the total distance between  $P$  and the NURBS approximation surface  $S$ . The parameter  $\delta$  is a given user error tolerance. The evolutionary strategy obtains the control point weights of  $S$ , so that (8) 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 in which a local optimization process will be carried out, in order to reduce 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 the optimization process can be run without distorting the local shape of the surface. Figure 4 shows an example of regions found by SOM.



**Figure 4.** Regions found by SOM.

Once  $P$  is clustered, an evolutionary strategy (1+1)-ES will optimize the local fitting of the NURBS in each cluster. 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  $\delta$ , as shown in Figure 5 (where  $w_i$  are the control point weights and  $\sigma_i$  are the mutation step sizes).

$w_1$	$w_2$	...	$w_n$	$\sigma_1$	$\sigma_2$	...	$\sigma_n$
-------	-------	-----	-------	------------	------------	-----	------------

**Figure 5.** Individual of the evolutionary strategy

*Mutation operator:* uncorrelated mutation with  $n$  mutation step sizes  $\delta$  is applied to the individuals, according to (5) and (6).

*Recombination operator:* the recombination operator is different for object variables  $w_i$  and parameters  $\sigma_i$  a global intermediary recombination is applied to object variables, according to (9), whereas a local intermediary recombination is applied to mutation step sizes  $\sigma_i$  according to (10).

*Selection operator:* the best individuals are selected according to the aptitude function given in (8). 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.

$$b'_i = \frac{1}{\rho} \sum_{k=1}^{\rho} b_{k,i} \quad (9)$$

$$b'_i = u_i b_{k_1,i} + (1 - u_i) b_{k_2,i} \quad (10)$$

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

The algorithm in Table 1 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 SOM
<b>For</b> each cluster <b>do</b>
Set individual size = cluster size
Set population size = $\mu$
Initialize the population randomly
Evaluate the population in the aptitude function (8)
<b>While</b> the stop criterion $\delta$ has not been reached <b>do</b>
<b>For</b> $i = 1$ to $\lambda \cdot 0.9$ <b>do</b>
$Ind_i = mut(Population_{rand(1,\mu)})$
<b>End for</b>
<b>For</b> $i = 1$ to $\lambda \cdot 0.1$ <b>do</b>
$Ind_i = rec(Population_{rand(1,\mu)})$
<b>End for</b>
$Population = select\_from(\mu + \lambda)$
<b>End while</b>
<b>End for</b>

## 7. Experimental results

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 run ISOMAP algorithm on the point cloud to project the cloud on a two-dimensional space,

as shown in Figure 6(b); then, the two-dimensional analysis takes place for inserting and removing points where necessary. The two-dimensional regularization is shown in Figure 6(c), inserted points are shown in red and green color-respectively, whereas the three-dimensional (point cloud) regularization is shown in Figures 6(d) and 7(a). The NURBS surface fitted to the regularized point cloud is shown in Figures 6(e) and 7(b). Finally, the trimmed NURBS surface is shown in figures 6(f) and 7(c).

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 application of the proposed method over complex surfaces.

To verify that the shape of the original point cloud was not distorted, two metrics were defined: i) The relative error  $E_{ddl}$  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 (11), is given in [3]. In our tests, the relative error was 0.031% and the modeling error was 0.01,

$$D_{avg} = \frac{\sum_{i=0}^{N-1} |d_i|}{N} \quad (11)$$

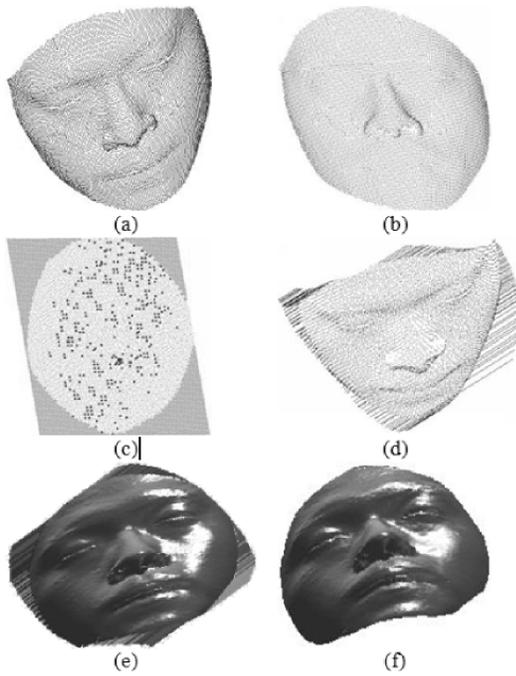
$$E_{avg} = \frac{D_{avg}}{L}$$

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.

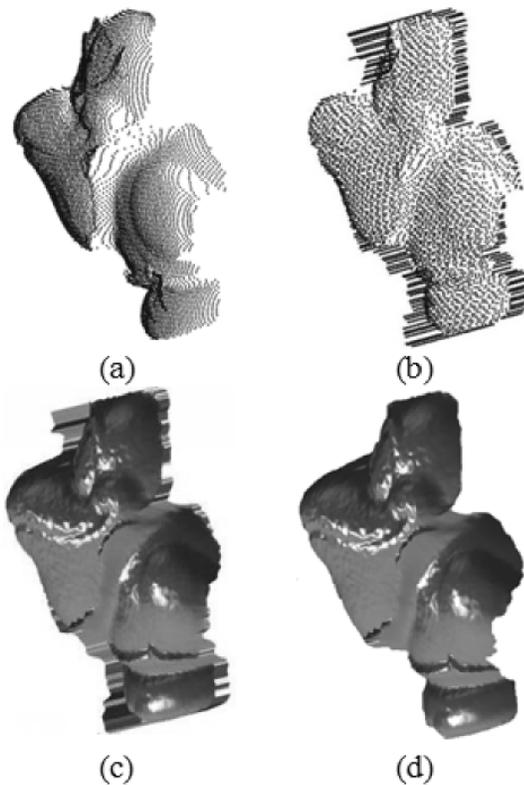
In Table 2, some statistics of the optimization process are presented.

**Table 2.** Statistics of the optimization process

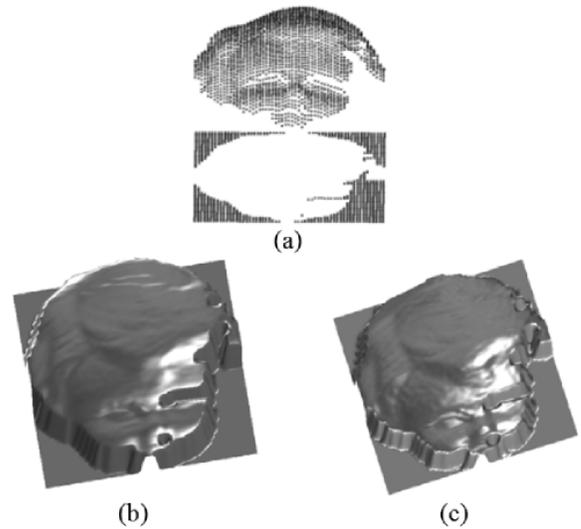
Processed images	50
Average points per image	15K
Average points per cluster	854
Tests per image	15
Time for clustering	10 seconds
Time for optimizing	3.5 minutes
Average generations per test	12
$\mu$	8
$\lambda$	56
Distance reduction	14.2%
Relative error	0.031%
Modeling error	0.01



**Figure 6.** Stages of the proposed method. (a) Original point cloud, (b) Point cloud flattened by ISOMAP, (c) Regularized projection, (d) Regularized point cloud, (e) NURBS surface constructed from the regularized point cloud, (f) Trimmed and optimized NURBS surface



**Figure 7.** Bird model. (a) Original model, (b) Regularized point cloud, (c) NURBS surface, (d) Trimmed NURBS



**Figure 8.** NURBS fitted over a complex surface (a) Original dataset. (b) Non optimized NURBS surface. (c) Optimized NURBS surface

## 8. CONCLUSIONS AND FUTURE WORK

An automatic 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.

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.

Our method cannot be applied over closed surfaces, due to the fact that ISOMAP cannot operate on closed manifolds. In order to apply our method over closed surfaces, future works

Can point in that direction. An additional 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 automatically establishing the number of clusters where the evolutionary strategy is run.

## REFERENCIAS

- [1] L. PIEGL AND W. TILLER. The NURBS Book, Springer-Verlag. 1995.
- [2] 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.
- [3] 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.
- [4] B. GREGORSKI, B. HAMANN, AND K. JOY. Reconstruction of B-spline Surfaces From Scattered Data Points. In Proceedings of Computer Graphics International. 2000.
- [5] M. ECK AND H. HOPPE. Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type. In Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques. ACM Press. 1996.
- [6] V. KRISHNAMURTHY AND M. LEVOY. Fitting smooth surfaces to dense polygon meshes. In SIGGRAPH 96 Conference Proceedings. ACM SIGGRAPH, Addison Wesley. 1996.
- [7] M. BERTRAM, X. TRICOCHÉ, AND H. HAGEN. Adaptive Smooth Scattered-data Approximation for Large-scale Terrain Visualization. In EUROGRAPHICS - IEEE TCVG Symposium on Visualization. 2003.
- [8] X. SHI, T. WANG, P. WU, AND F. LIU. Reconstruction of convergent g1 smooth b-spline surfaces. In Computer Aided Geometric Design. Vol. 21. November. 2004.
- [9] A. YVART, S. HAHMANN, AND G. BONNEAU. Smooth Adaptive Fitting of 3D models using hierarchical triangular splines. In Shape Modeling International. 2005.
- [10] J. CORDERO AND J. PAREJO. Curvas y Superficies para Modelado Geométrico. 1st ed, Alfa Omega, España, 2002.
- [11] C. DE BOOR. On Calculating With B-Splines. In J. Approximation Theory, Vol. 6, No. 1, July, pp. 50-62. 1972.
- [12] T. BÄCK, AND H. SCHWEFEL. Genetic Algorithms in Engineering and Computer Science. John Wiley & Sons Ltd. 1995.
- [13] J. TENENBAUM, V. DE SILVA, AND D. LANGFORD. A Global Geometric Framework for Nonlinear Dimensionality Reduction. In Science 290(5500):2319 – 2323. December 2000.
- [14] H. QIN AND D. TERZOPOULOS. Triangular NURBS and their dynamic generalizations. Computer Aided Geometric Design. Volume 14 Issue 4, May 1997.
- [15] D. ADI., S. SHAMSUDDIN. AND A. ALI. Particle Swarm Optimization for NURBS Curve Fitting. Sixth International Conference on Computer Graphics, Imaging and Visualization, 2009.
- [16] Z. JING, F. SHAOWEI AND C. HANGUO. Optimized NURBS Curve and Surface Fitting Using Simulated Annealing. Second International Symposium on Computational Intelligence and Design, 2009.
- [17] M. SARFRAZ AND M. RIYAZUDDIN. Curve Fitting with NURBS using Simulated Annealing. Applied Soft Computing Technologies: The Challenge of Complexity Advances in Soft Computing, Vol. 34. 2006.
- [18] K. WEINERT AND J. MEHNEN, "Discrete NURBS-Surface Approximation using an Evolutionary Strategy," Tech. Rep. Department of Machining Technology, University of Dortmund, Germany. 2000.
- [19] KOHONEN, T., "The Self-Organizing Feature Map," Pro. of the IEEE, Vol. 78, pp. 1464-1480 (1990)