3D SURFACE GENERATION FROM POINT CLOUDS ACQUIRED FROM A VISION-BASED LASER SCANNING SENSOR

Gerardo Antonio Idrobo Pizo, gerardo_idrobo@unb.br
José Mauricio S. T. Motta, jmmotta@unb.br
Universidade de Brasília, Faculty of Technology, Department of Mechanical Engineering
70.910-900, Brasilia – DF, Brazil

Abstract. *This paper presents special algorithms to be used with a specially developed vision-based laser scanning optical sensor to generate 3D surface digital models aiming at repairing defects on turbine blade geometric profiles with welding robots. Currently, most of the systems for 3D surface modeling based on vision make use of angular position sensors for measuring the rotation of a source of light. These sensors impose restrictions in the volume and range of work of the scanner and the reconstruction is typically based on equations of triangulation which produce measurement errors due to the non-linear relationship between angle and distance, increasing severely the errors as a function of the distance from the object targeted. To overcome these difficulties, specialized computer vision algorithms were developed in this research to eliminate the use of angular measuring sensors and positioning systems, reducing costs and errors and also allowing larger measurement distances. The system is also self-calibrated with no need of other measurement systems for camera and laser calibrations. The system developed so far capture object target points and calculates their 3D position producing a cloud of points. To generate a 3D surface with sufficient accuracy from the cloud of points, triangular meshes are generated to be used with NURBS modeling and these parametric surfaces are optimized by using adaptive filters. The system is compact, easy to transport and it is shown the complete system architecture with two laser diodes and a single camera. Experimental procedures have been specially developed to estimate the accuracy of the measurement system by locating the center of spherical targets from the surface generated from the sphere, with promising results. Theoretical basis, hardware implementation and practical results obtained are also discussed.*
Keywords: 3D Vision Sensor; Robot Vision; Surface Modeling; Laser Scanning

1. INTRODUCTION

This article presents and discusses an ongoing research on 3D surface modeling from point clouds acquired from a special vision sensor system embedded on a robot arm dedicated to repair surface flaws on hydraulic turbine blades capable to make weld beads in layers. For the robot to have the 3D surface map of the blade it is necessary to develop a computer system to transform a 3D cloud of points in a 3D surface digital model.

The exact representation of the original surface is not always possible, but only an approximation of it. This approximation should keep not only the geometric information of the real object, but also needs to offer a realistic and precise map of the surface.

Surface modeling can be performed by interpolation or by approximation routines. Interpolation requires that the surface passes exactly through all the points, but if the data contain some noise added during the acquisition process, an approximated surface is more appropriate. In this case, it is only required that the surface passes close to the control points.

To assess the quality of a 3D object model is not easy when the relations of connectivity between points that form the model are not known, and mostly, when the cloud of points used to model the object is contaminated with noise from the image acquisition process. Such noise can be eliminated or attenuated through filtering. The main objective of filtering is to improve the quality of the signal according to a performance criterion (Chui *et al*, 2008).

This article discusses adaptive filters as one way to tackle the problem of image noise filtering, since many of the noise that pollute the cloud of points appear as a stochastic process that cannot be attenuated successfully through common digital filters (low pass, high pass, band pass). (Manolakis *et al*, 2000). This article also addresses the combination of two techniques for 3D surface modeling of objects: reconstruction of surfaces by triangular mesh grids and parametric functions - NURBS (Non-Uniform Rational B-Splines) (Farin, 2001, Anderson & Crawford-Hines, 2000, Roth, 1998). The algorithm performance is evaluated experimentally by comparing the geometric dimensions of a sphere model constructed by the system to the actual sphere previously measured with standard measurement devices, showing that the system can be used to construct 3D object surface models with known geometric accuracy.

2. 3D VISON SENSOR AND IMAGE FILTERING

The 3D vision sensor constructed and used to acquire cloud of points is a laser scanner based system with a high resolution camera and two laser diodes to project two light planes on the object and uses computer vision algorithms to extract information directly from the image. With this arrangement it is possible to remove sensors to measure angular positions of the laser diode, improving accuracy. Some details of this system can be found in the work of Ginani and Motta (2007). From the geometric distortions of the projected light in the object and knowing the intrinsic camera

parameters it is possible to calculate the 3D coordinates of the object surface points, generating point clouds from the light strips scanning the object surface (Fig. 1). However, as in any other vision system, the camera and the environment generate noise.

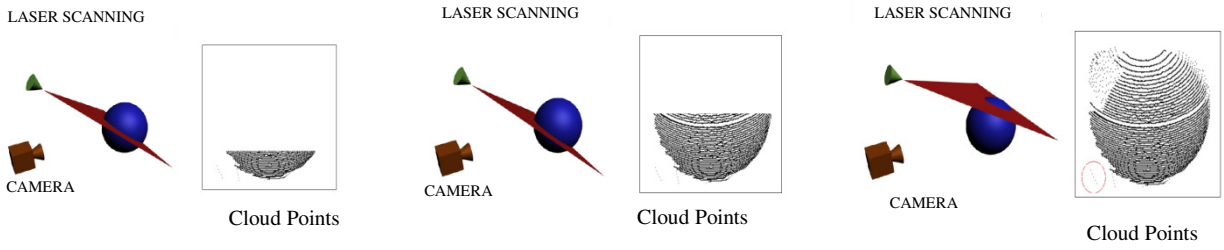


Figure 1. Vision sensor hardware VISSCAN-3D

The cloud of points acquired can be modeled as the output of an input image *x(n)* which passes through a FIR (finite-duration impulse response) filter (corresponding to the image processing) with impulse response (Fig. 2), (Haykyn, 2001), $h = [h_0\ h_1\ ...\ h_{N-1}]^T$ with finite length N. Random noise *s(n)* can be added to the output of the impulse response and

$$d(n) = h^T x(n) + s(n) \tag{1}$$

is the cloud of points contaminated by noise.

The attenuation of the random noise signal *s(n)* is proposed here to be achieved as in Fig. 2. The output of the adaptive filter *w* is *y(n)*

$$y(n) = w^T x(n) \tag{2}$$

Where $w = [w_0\ w_1\ ...\ w_{N-1}]$. This estimative is subtracted from the contaminated signal, generating the error

$$e(n) = d(n) - y(n) = (h^T - w^T)x(n) + s(n) \tag{3}$$

In an ideal case $w = h$ and $e(n) = s(n)$, such that the noise would be completely eliminated. However, the approximation of *h* by a filter *w* of finite length restricts full noise elimination.

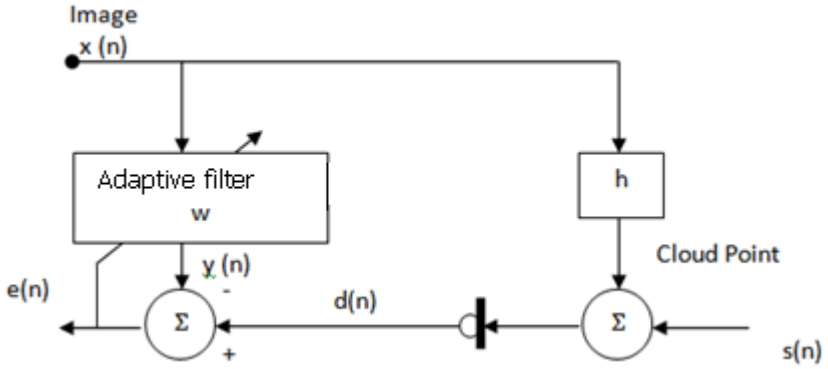


Figure 2. Block diagram showing an adaptive filter to attenuate noise in point clouds

2.1. Adaptive Filtering

The choice of an adaptive filter (Haykyn, 2001; Diniz, 2002) is often based on speed of convergence and accuracy, numerical complexity and structure stability.

The speed of convergence is defined as the number of iterations required for the algorithm to be close enough to the optimal solution with stationary signals. The higher the speed of convergence, the faster the algorithm adapts to an environment with unknown stationary characteristics.

Numerical complexity is the number of operations to perform a complete iteration of the algorithm. Depending on the adaptive algorithm, there may be numerical instability in the filter. The accumulation of errors due to accuracy restrictions can make the algorithm to move away from the ideal state. It is possible that the deviation is of such a nature that the errors will accumulate indefinitely and the filter will be left to overflow and instability.

The chosen adaptive algorithm was the LMS (Least Mean Square) as it exhibits a higher convergence speed (about 10 times higher) than other known algorithms such as Kalman and Wiener filters and slightly lower than RLS (Recursive Least Square), mainly due to its simplicity.

With the LMS algorithm, the vector of coefficients is updated in the opposite direction to the gradient vector obtained from the instantaneous quadratic error, i.e. (Fisher et. al.,2000):

$$w(n+1) = w(n) - \mu' \nabla_w [e^2(n)] \quad (4)$$

, where μ' is the step of the algorithm. Thus, the equation for updating the coefficients of the LMS algorithm is given by:

$$w(n+1) = w(n) - \mu e(n) x(n) \quad (5)$$

with $\mu = 2u'$

3. ORGANIZING POINT CLOUDS WITHIN A TRIANGULAR MESH

Over the years, several techniques have been proposed to solve the problem of surface reconstruction from point clouds, using concepts from computational geometry to models based on partial differential equations (Roth,1998). Most of the methods can be classified as implicit or parametric (Hoppe,1994;Yvart ,2005). More recently, Boissnoat and Cazalis (2002) classified the methods into four groups: 1) Methods based on local projections: approximation is performed treating the surface as a function defined in a local reference domain; 2) Sculpting Methods: the methods are based on the removal of triangles from a spatial arrangement, as the Delaunay triangulation; 3) Implicit Methods: the methods are based on an implicit definition of a function of distance with signal f and in the realization of a triangulation approximation of the isosurface $f(x) = c$. This process of approximation of the function f is performed using techniques of extraction of isosurfaces; 4) Methods based on deformable models: from an initial shell (membrane) deformations are applied to minimize a function of energy and get closer to surface.

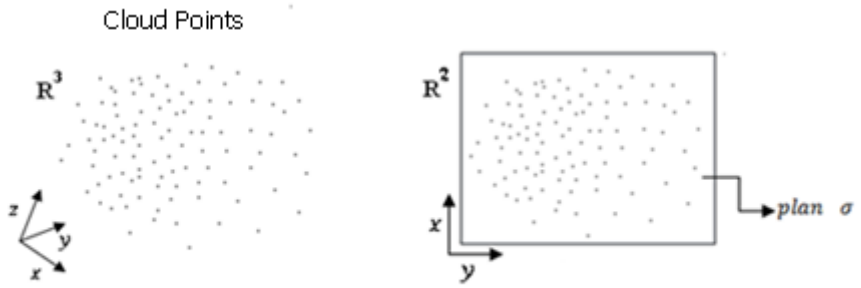
All kinds of methods listed above are based on the functional characteristics of the algorithm. However, other types of classification are possible, such as divisions based on computational complexity or quality demanded by the sampling algorithms.

Among the previous methods, a method based on deformable models was chosen because it offers more speed in the process of organizing the point clouds within a mesh. Also, it minimizes the computational cost as the 3D surface is modeled.

The algorithm is divided into three steps, and each step is divided into several sub-steps (Chui *et al* , 2008): (a) Projection of the cloud of points on the mesh grid, (b) Nodal point repositioning, (c) Reconstruction of the 3D triangular mesh from the 2D triangular mesh.

3.1. Projection of the cloud of points on the mesh grid

1. A cloud of points is projected from R^3 to R^2 on the plane σ (Fig. 3a and Fig. 3b).
2. Adjust the cloud of points inside a parallelogram (angle of 60°, Fig. 3c). This polygon involves all points of the sample that are to be analyzed in all subsequent steps of the algorithm.
3. Divide the polygon into horizontal and vertical lines separated of a distance t. This separation influences the resolution of the 3D surface (Fig 3d and Fig 3e).
4. Diagonal lines are projected from the left to the right of the polygon, forming equilateral triangles Fig. 3f.



(a) (b)

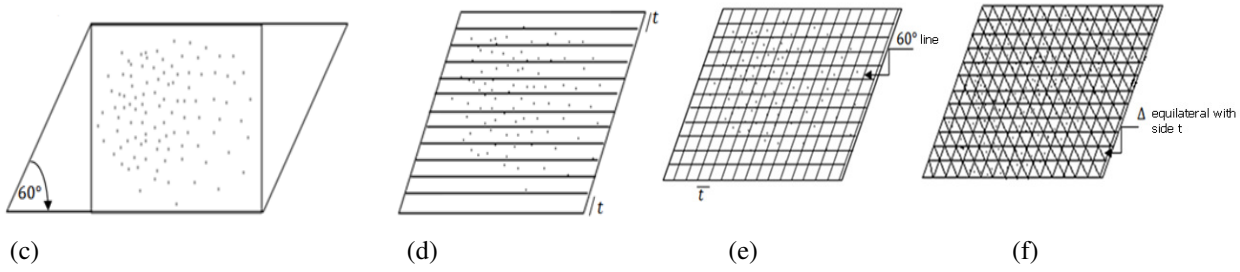


Figure 3. Projection of the cloud of points on the mesh grid

3.2. Nodal Point Repositioning

Nodal point repositioning is a essential step to adjust a mesh nodal point to the a cloud point. In each mesh nodal point a circle is drawn connecting the adjacent nodal points, in order to select only one point among many that are within the area enclosed by the circle. The point selected is the one that is closer to the node of the mesh inscribed in the corresponding circle Fig. 4a. The procedure is repeated for the other node points, and those points that have not been selected will be deleted.

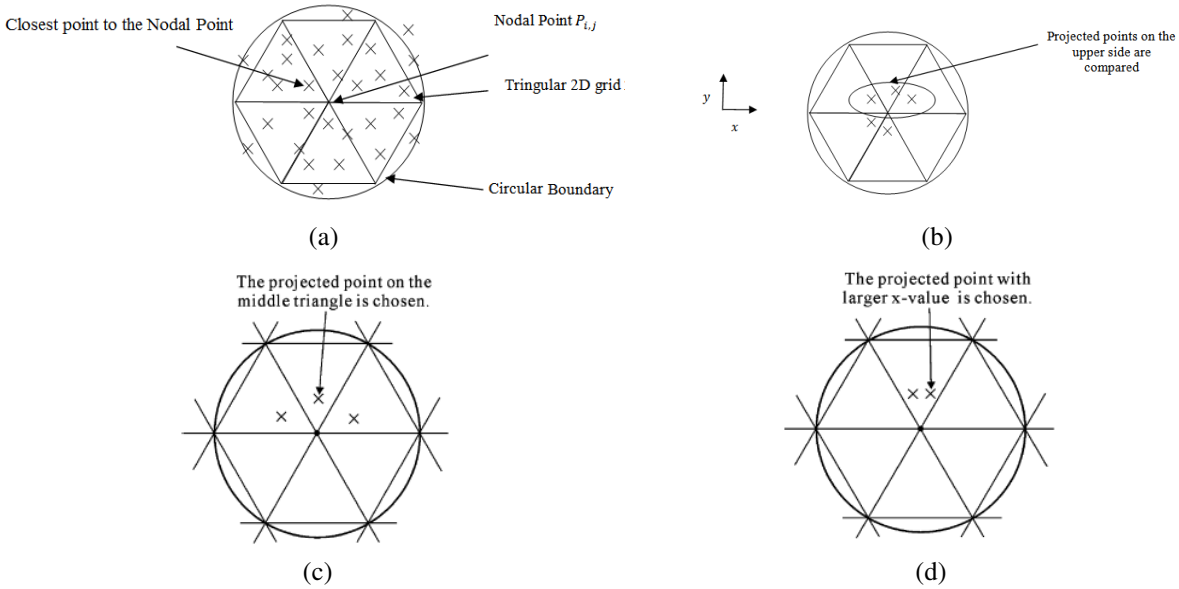


Figure 4: Repositioning of nodal points. (a) Point selection approach, (b) An example case for choosing projected points using Criterion A, (c) A projected point is selected as the new nodal point position using Criterion B, (d) Projected point being chosen using Criterion C.

For example, in Fig. 4a, the nearest projected point will be considered as the new position of the nodal point $P_{i,j}$. However, if there are two or more points satisfying closer distances, the following criteria should be established:

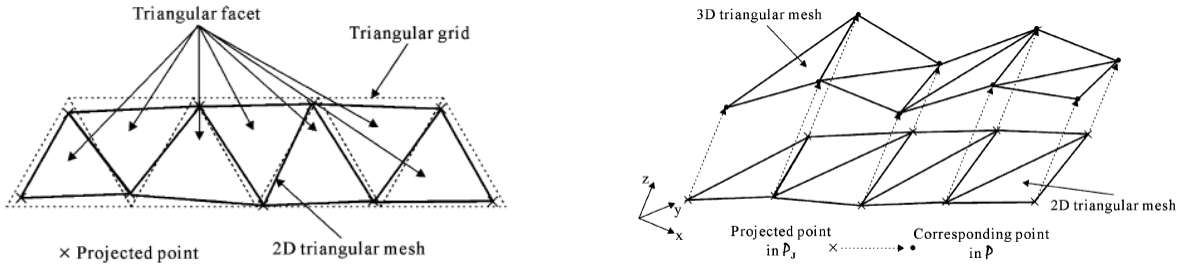
Criterion A. If two or more projected points are equal in distance to *P(i, j)* but are located in different triangles, the projected point at the top (i.e., higher *y*) will be chosen. For example, in Fig. 4b, five points are projected in equal distances from the nodal point *P(i, j)*. Items found in the upper nodal point will be selected. The choice of the new position of the nodal point will be explained using Criterion B.

Criterion B. If two or more projected points are equal in distance in the same horizontal line Fig. 4c, but in several different triangles, the projected point in the triangle is chosen as the new position of the nodal point $P_{i,j}$.

Criterion C. If two or more projected points are equal in distance in the same horizontal line and same triangle, the projected point with higher value *x* is chosen Fig. 4d.

3.3 Reconstruction of the 3D triangular mesh from the 2D triangular mesh.

The 3D triangular mesh can be reconstructed next to the process of displacing the nodes of the 2D mesh (Fig. 5a) by using the depth values of each displaced nodal point (criterion A, criterion B or criterion C) (Fig. 5b).



(a) (b)

Figure 5. (a) 2D triangular mesh constructed from the projected points next to nodal points repositioning. (b) 3D triangular mesh is constructed based on its corresponding 2D triangular mesh.

4. NURBS SURFACE REPRESENTATION

The construction of a 3D model using parametric surfaces is a process that requires an adjustment of approximation between the geometrical structures of a structured mesh in the space R^3 (Hoppe, 1994).

NURBS (Non Uniform Rational B-Spline) is an industrial standard tool for modeling and design of simple and complex geometries. The reasons for such widespread use are: a) based on common mathematical methods to represent free form objects; b) provides a high flexibility in the design of shapes; c) the processing time is reasonably small; d) are generalizations of curves and surfaces of B-spline and Bezier.

The NURBS surfaces are curve sets of the same type that can be defined as (Park *et al*, 1999).:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} T_{i,p}(u)T_{j,q}(v)W_{i,j}D_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} T_{i,p}(u)T_{j,q}(v)W_{i,j}} \tag{6}$$

where $D_{i,j}$ are the control points; $W_{i,j}$ are the weights defining how significant will be the control point on the curve; $T_{i,p}$ and $T_{j,q}$ are the B-spline functions defined in two parametric directions. The number of control points is defined by the surface degree in each parametric direction ("u" and "v"). Therefore, in each direction there is a number of control points. The B-spline functions are functions of the nodes (t_i) that are increasing sequences. So, one can define a B-spline function as:

$$T_{i,p}(u) = \begin{cases} 1 & t_i \le u \le t_{i+1} \\ 0 & else \end{cases} \tag{7}$$

with

$$T_{i,p}(u) = \frac{u-t_i}{t_{i+k-1}-t_i}\cdot T_{i,k-1}(u) + \frac{t_{i+k}-u}{t_{i+k}-t_{i+1}}\cdot T_{i+1,k-1}(u) \tag{8}$$

The nodes are represented by a list of numbers that is commonly called the node vector. The node vector needs to be a sequence of numbers equal or increasing (uniformly or not).

4.1. Problem Formulation

Given a sequence of samples (u_i, M_i) with ($i = 0, ..., m$), a surface $X(u,v) = X_{W,D,T}(u,v)$ is to be found that best fits to samples M_i, minimizing the mean square error:

$$\min_{W,D,T} \sum_{i=0}^{m} \left\| X_{W,D,T}(u_i, v_i) - M_i \right\|^2 \tag{9}$$

, where W are the weights, D are the control points and T is the node parametric vector.

Randrianarivony and Brunet (2002) addresses the Eq. (9) as a function of the node parametric vector T, since there is no linear dependence between X and the parameters W, D, T. They give solution to the problem, simplifying the task and adding some penalties that guarantee the mathematical formulation of NURBS functions (Loop, 1990). The

solution of Eq. (9) is subject to the restrictions of a solution with a sequence T of no decreasing nodes. Thus, the problem can be considered a typical reduction by least squares:

$$X_T(u,v) = \left(X_{T,x}(u,v), X_{T,y}(u,v), X_{T,z}(u,v)\right) with\ M_i = (M_{i,x}, M_{i,y}, M_{i,z}) \tag{10}$$

To linearize the equations, a matrix vector is defined as

$$S = \begin{Bmatrix} S_{3i}(T) = X_{T,x}(u,v) - M_{ix} \\ S_{3i+1}(T) = X_{T,y}(u,v) - M_{iy} \\ S_{3i+2}(T) = X_{T,z}(u,v) - M_{iz} \end{Bmatrix} \tag{11}$$

Replacing S in Eq. (11), one obtains

$$\min_T \sum_{i=0}^{3m+2} S_i^2(T) \tag{12}$$

The solution of Eq. (12) should guarantee a sequence of increasing T nodes, for which a function is defined as

$$R(x) = \begin{Bmatrix} 0 & se & x > 0 \\ -x^3 & se & x \leq 0 \end{Bmatrix} \tag{13}$$

In addition, the penalty function can be formulated as

$$\min_T \sum_{i=0}^{3m+2} [S_i(T) + \alpha R(T)]^2 \tag{14}$$

where α is a very large positive number (≈1 x 10^{10}). Finally, $r_i(T) = S_i(T) + \alpha R(T)$ in

$$\min_T \sum_{i=0}^{3m+2} [r_i(T)]^2 \tag{15}$$

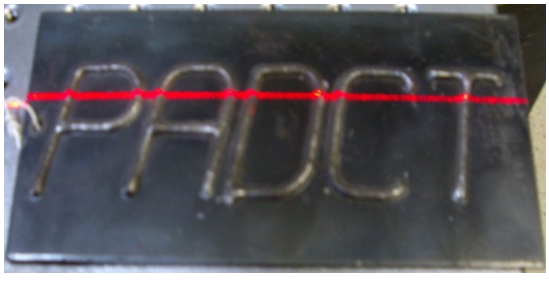
Equation (15) can be solved by an optimization method based on nonlinear least squares and the Levenberg-Marquardt algorithm (Randrianarivony & Brunnett (2002)), which is not addressed in this article.

4. RESULTS

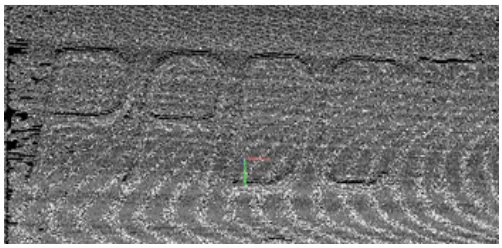
Figure 6 presents several steps of the 3D surfaces fitting process from cloud of points. The steps are described below:

1. Figure (6a) shows the object 3D surface to be reconstructed, a metal plate with a weld bead on it. The object dimensions were measured by means of a caliper guide as: 74 ± 0.01 mm by 54 ± 0.01 mm and a maximum height of 12 ± 0.01 mm.
2. Figure (6.b) shows the light stripe on the object of the object, projected by the laser scanning system.
3. Figures (6.c) and (6.d) shows the processed images and the range images as point clouds. Some noise can be noted visually.
4. Figures (6.e) and (6.f) shows a plot of a set of points from the light stripe on the object surface (Figure 6b) before and after adaptive filtering with the LMS algorithm.
5. Figures (6.g), (6.h) (6.i) and (6.j) present the construction of a 3D model using meshes and parametric surfaces. It was assigned a square grid (4mm) as in figures (3.e) and (3. f) for filtering the data. The initial number of points in the point clouds was 99,278 and at the end of the process of assigning the 3D mesh nodes (Fig. 5.b) the number of points decreased to 2,350 points.

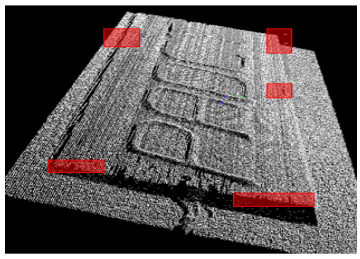
(a)



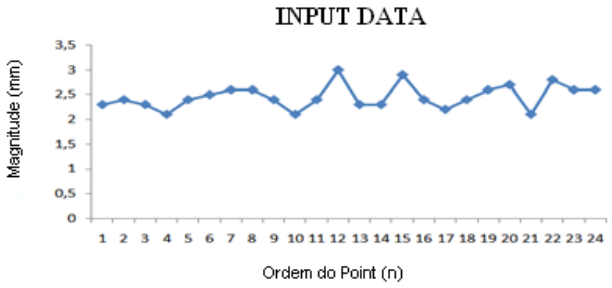
(b)



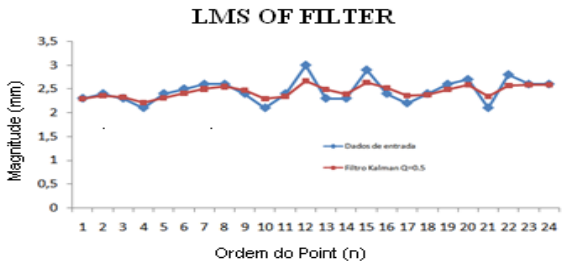
(c)



(d)

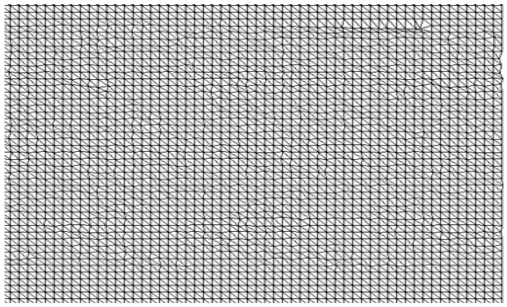


(e)



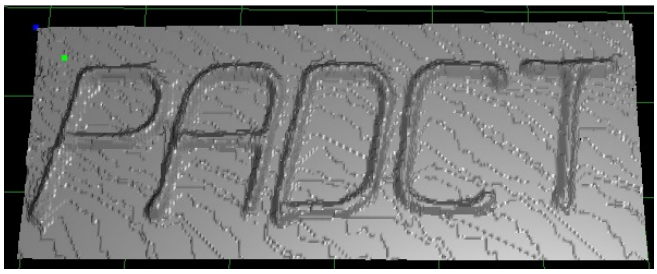
(f)

(g)



(h)

(i)

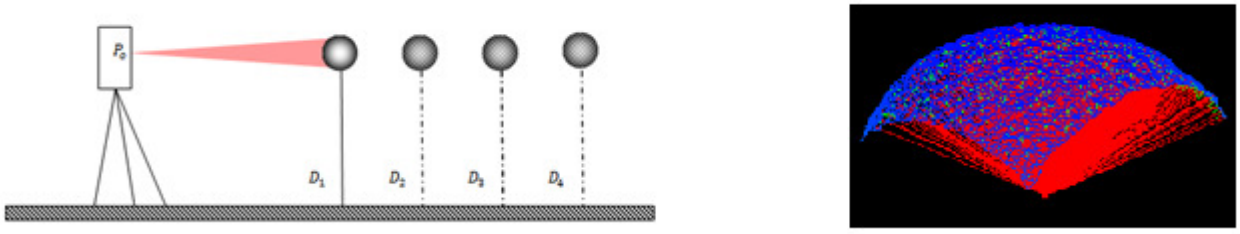


(j)

Figure 6. Several steps for surface fitting in point clouds: (a) Weld bead; (b) Laser scanning on the weld bead; (c) Point cloud acquired from the weld bead image; (d) Noise detected visually in the point cloud; (e) Plot of a section of the weld bead in Z coordinates; (f) Plot of the section filtered with the adaptive LMS algorithm; (g) Point cloud projected on the 2D plane; (h) 2D triangular mesh; (i) 3D triangular mesh; (j) Parametric surface fitted in the 3D triangular mesh

6. EXPERIMENTAL EVALUATION

It is extremely important to assess the resolution of a vision sensor as a function of the distance from the target object. Thus, depending on the details needed in the object image, the distance can be chosen properly. For an assessment of that resolution and measurement accuracy, a spherical target object was used, with $36.10^{\pm 0.05}$mm of diameter (a roller ball of an electric generator bearing). The ball was scanned by the sensor and point clouds were generated at different distances from a reference (Fig. 7a).



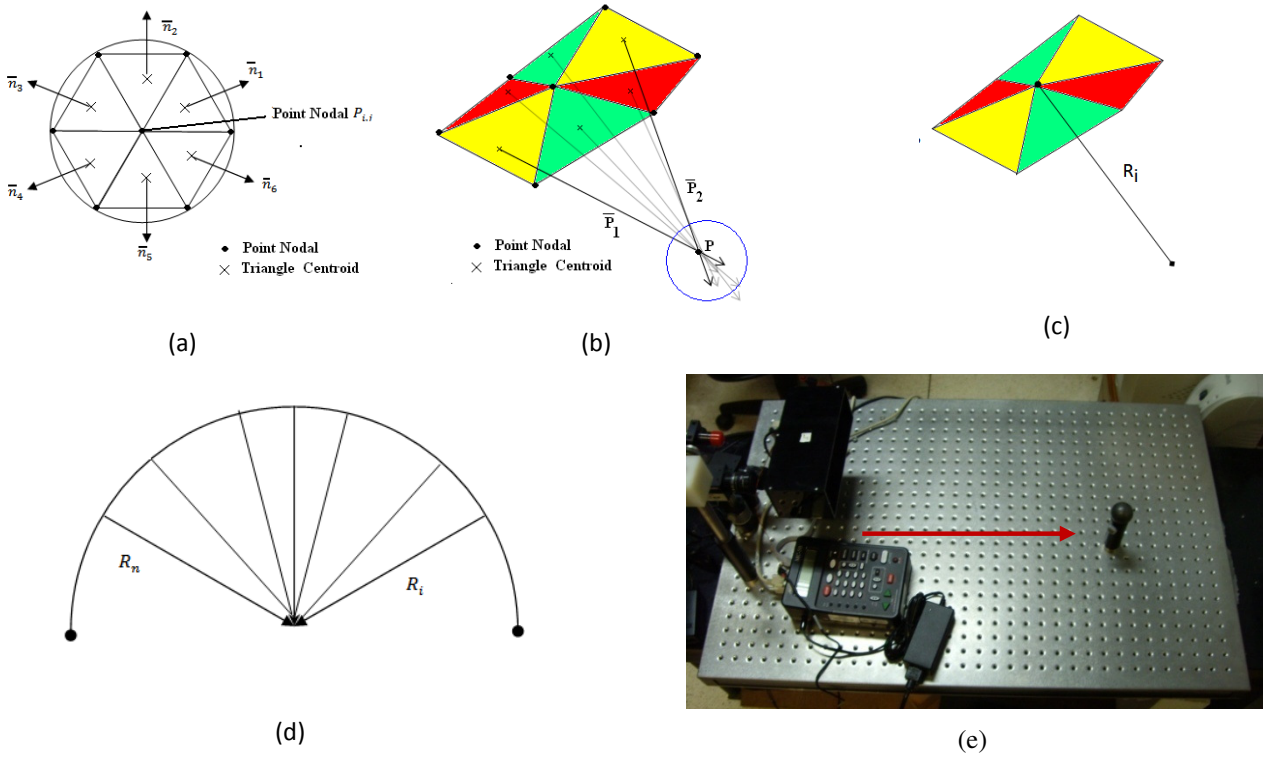
(a) (b)

Figure 7. (a) Acquisition of point clouds at different distances between point P_0 and the sphere; (b) Point cloud acquired from the sphere image.

Next to scanning the metal sphere at different distances, the sphere radius was calculated at each distance (Fig. 7b) by using point triangulation (Forsyth and Ponce, 2003), explained next.

The experimental assessment was conducted using the tools developed to generate the meshes (section 6). For each surface patch connected to a node the average normal vector was calculated as shown graphically in (Fig. 8a). Then, the normal vectors to each adjacent and opposite surface patches of a pair connected to the node are projected to intersect at the center of the sphere (Fig. 8b).

Due to the finite precision of the system and to noise, parameters and image locations are known only approximately and the two rays will not actually intersect in space; their intersection can only be estimated as the point of minimum distance from both rays. So, the intersection point of the prolonged normal vectors arising from the centroid of each of the two patches, p_1 and p_2, can be estimated as the midpoint of a segment perpendicular to both rays (Fig. 9).



(a) (b) (c)

(d) (e)

Figure 8. Several steps to estimate the sphere radius experimentaly.

The vectors shown in (Fig. 9) can be expressed in the same coordinate system as

$$ap_1 + c(p_1 \times R^T p_2) = T + bR^T p_2 \tag{16}$$

, where T is the translation vector between p_1 and p_2 and R the rotation matrix between the two patch planes π_1 and π_2

; *a, b* and *c* are the parameters to be solved to find the endpoints of the segment *s*. The intersection point is the midpoint of the segment *s*.

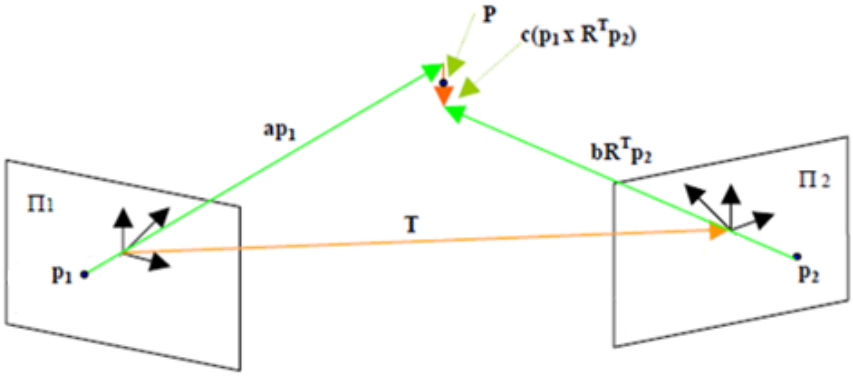


Figure 9. Triangulação dos pontos p_1, $p_2 e\ P$

The radius associated to each pair of patches is the distance from the midpoint of *s* to the node point. This process is repeated for the other two pair of patches connected to the same node, as shown in (Fig. 8.b). The radius associated to each node is calculated as the average radius of the three pairs of patches, as shown in (Fig. 8.c). The sphere radius was estimated as

$$\overline{\mathrm{R}} = \frac{\sum_{j=1}^{n} R_j}{n} \tag{17}$$

, where *n* is the number of nodes. The standard deviation is calculated from

$$Sd = \sqrt[2]{\frac{1}{n-1}\left(\sum_{i=1}^{n}(\mathrm{R_i} - \overline{\mathrm{R}})\right)} \tag{18}$$

Table (1) presents the results of the sphere radius values calculated at different distances. The distances between the sensor and the sphere were measured using a breadboard with holes spaced by 25.0 ± 0.5 mm as shown in (Fig. 8e). The sphere was positioned on a metal post. The initial distance was estimated using a manual caliper.

Table 1. Dimension errors of the vision system associated to surface modeling from point clouds

Distances (mm) Sphere-Sensor	$\overline{\mathrm{R}}$ (mm) Estimated Average Radius	$\lvert(\mathrm{r} - \overline{\mathrm{R}})\rvert$ Error (mm)	*Sd* (mm) Standard Deviation	Number of Nodes
250.00	17,93	0,32	0,287	472
275.05	17,93	0,32	0,336	452
300.05	17,84	0,41	0,343	428
325.50	17,84	0,41	0,411	436
350.20	17,73	0,52	0,412	450
375.00	17,69	0,56	0,512	472
400.05	17,67	0,58	0,528	448
425.15	17,60	0,65	0,628	426
450.05	17,54	0,71	0,684	388
475.10	17,48	0,77	0,762	324
500.30	17,39	0,86	0,821	326

r = measured sphere radius(18,25mm)

The results show that the average errors found are within the range of 1mm at distances of 500 mm. The results, of course, are dependent on the lens size, since the image size is the parameter to influence the object image physical resolution. It can be observed that the average error did not increase at the same ratio as the distance sensor-sphere, but the standard deviation. So, with the lens used (focal length = 8.5mm), the vision system dimensional average accuracy was evaluated as approximately 1:500.

7. CONCLUSIONS

This article presented the development of a system model to fit 3D surfaces to point clouds acquired by a special vision sensor used in robot applications where surface details have to be mapped for welding tasks. It is presented and discussed a technique to generate 3D surfaces from point clouds by using parametric surfaces – NURBS – with control points defined as nodal points of 3D triangular meshes. The triangular meshes are constructed from the projection of point clouds onto 2D meshes, next cloud points are selected when closer to the nodal points of the 2D mesh, then the 2D mesh is deformed to match the nodal points with the selected cloud points and finally the 2D mesh is projected back into the 3D space. Parametric surfaces are then approximated to the control points to fit the 3D surface to the mesh grids. A formulation of adaptive filters for point clouds in images is discussed and the model used presented.

The results of the constructed model are shown by means of digital representation from range images along all the surface fitting steps. Experimental results were presented to assess the 3D accuracy of the vision system by comparing 3D data points from the surface fitted with the measured dimensions of a metal sphere at several distances from the vision sensor. Results showed that the system accuracy is fairly enough to guarantee weld quality with sub-millimeter positioning errors.

8. ACKNOWLEDGEMENTS

ELETRONORTE (Electrical Power Plants of the North of Brazil) and FINATEC (Scientific and Technological Enterprises Foundation) are thanked for their financial support and collaboration. Our gratitude is especially conveyed to the staff of ELETRONORTE for their technical support and interest.

9. REFERENCES

Anderson, C.W.,Crawford-Hines, 2000, “Fast generation of NURBS surfaces from polygonal mesh models of human anatomy”. Technical Report CS-99-101, Fort Collins, Colorado State University, USA

Loop, C., 1990, “Generalized B-spline surfaces of arbitrary topology”, Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques, Dallas, TX, USA pp. 347 - 356

Boissonnat, J.-D. and Cazals, F., 2002,”Smooth surface reconstruction via natural neighbour interpolation of distance functions”, Proceedings of the 16th Annual Symposium on Computational Geometry, Clear Water Bay, Kowloon, Hong Kong, pp. 223 – 232.

Chui, K.L., Chiu, W.K. and Yu, K. M., 2008, “Direct 5-axis tool-path generation from point cloud input using 3D biarc fitting”, Robotics and Computer-Integrated Manufacturing, Vol. 24 , n. 2, p.p. 270-286.

Diniz, P.S.R., 2002, “Adaptative filter theory: Algorithm and practical implementations”, Kluwer Academic Publishers, USA, 559 p.

Haykyn,S., 2001, “Adaptive filter theory”, 4th ed., Prentice Hall, USA, 978 p.

Hoppe, H., 1994, “Piecewise smooth surface reconstruction”, Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, Orlando, Florida, pp. 295 – 302.

Farin, G. (2001). “Shape”. Math. Unlimited-2001 and Beyond, 3a. edição, pp 463-466.

Fisher, M, Mandic, D, Bangham, J.A. and Harvey, R., 2000.“Visualizing error surfaces for adaptive filters and other purposes”. IEEE International Conference on Acoustic, Speech and Signal Processing, Istanbul, Turkey, 4 p.

Forsyth, D and Ponce, J., 2003, “Computer Vision: A Modern Approach”, Prentice-Hall, New Jersey, USA.

Ginani, L. and Motta, J. M S. T.., 2007, “A Laser Scanning System for 3D Modeling of Industrial Objects Based on Computer Vision”, Proceedings of 19th International Congress of Mechanical Engineering, Vol. 1, Brasilia-DF, Brazil, 10 p.

Manolakis, D. G., Ingle, V.K. & Kogon, S. M. (2000). *Statistical and Adaptative Signal Processing: Spectral Estimation, Signal Modelig, Adaptative Filtering and Array Processing*. McGraw-Hill, EUA.

Park. K., Yun, I. D. and Lee, S. U., 1999, “Constructing NURBS surface model from scattered and unorganized range data ”. Proceedings of the Second International Conference on 3D Digital Imaging and Modeling, pp. 312-320.

Randrianarivony, M. e Brunnett, G., 2002, “Parallel implementation of surface reconstruction from noisy samples”, Preprintreihe des Chemnitzer SFB 393, 02-16.

Roth, G. and Boulanger, P. 1998, “Cad model building from multiple range images”, Proceedings of Vision Interface, Vancuver, BC, pp. 431-438.

Yvart, A., Hahmann, S. and Bonneau, G.-P., 2005, “Smooth adaptive fitting of 3D models using hierarchical triangular splines ”, Proceedings of the International Conference on Shape Modeling and Applications 2005, pp. 13-22.

5. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.