CILAMCE 2017
Florianópolis - Santa Catarina - Brazil

A MESH GENERATION ALGORITHM FOR NURBS BASED METHODS THROUGH IMAGE ANALYSIS

Mateus Rauen

Roberto Dalledone Machado

Marcos Arndt

mateusrauen@gmail.com

Universidade Federal do Paranaá, Programa de Pos-Graduação em Métodos Numéricos em Engenharia

R. Cel. Francisco Heraclito dos Santos, 210, Jardim das Américas, Zip Code 81531-990 Curitiba, Paraná, Brazil ?

Abstract. *Isogeometric analysis and CAD based methods appeared with the philosophy of global mapping, which led to some advantages related with mesh creation, refinement and geometry exactness. But the CAD mesh extraction from a given representation for the use in numerical methods requires yet some improvements and technologies. This paper presents a simple algorithm to extract CAD meshes from an image representation. Based on color filters and boolean representations, the image contour is defined and scaled. A point cloud algorithm based on the image extraction is developed and a fitting algorithm is tested for surfaces. Based on numerical experiments the mesh obtaining is tested and validated. The method is capable to accurate fit on geometry and the convergence condition depends directly on the number of cloud points.*

Keywords: *CAD, Mesh creation, Isogeometric, Image Analysis*

1 INTRODUCTION

In the last few years the relation between numerical analysis and CAD meshes became closer with the advent of isogeometric analysis (IGA) (Hughes et al., 2005). The possibility to discretize a mesh in the parametric space gave a global mapping context to the numerical methods, which allowed to introduce CAD mapping in the numerical integrations. This leads to advantageous facts related with geometry exactness and automatic mesh refinement, which those concepts derives from CAD classical literature (Piegl and Tiller, 1997) (Farin, n.d.) by the use of NURBS (non uniform rational b-spline) shape functions. Concerning numerical verification on IGA, several applications on method shown robustness and advantages related with mesh manipulation (Cottrell et al., 2009) (Shojaee et al., 2012) (Cottrell et al., 2006).

As the CAD mesh gives exact geometry, the isogeometric concept on numerical application is to use at least a minimum input that could describe the exact geometry. But concerning a general object representation, the determination of a CAD geometry input is a problem to be solved. Otherwise the communication between CAD and IGA meshes from a general representation is a actual discussion and yet requires developments and improvements. This paper tests a simple algorithm to extract a CAD mesh from an image containing a general representation.

2 ISOGEOMETRIC ANALYSIS

Isogeometric analysis (IGA) is discrete numerical method to solve partial diferential equations where the domain analysis is exactly described by NURBS functions, from CAD classical literature Farin (1999), Piegl and Tiller (1997). Like FEM, IGA is also origined by an weighted formulation and discretization. The most remarkable diference between IGA and FEM is in the discretization strategy Hughes et al. (2005). The usual implementation of FEM is to divide the physical domain in a finite number of locally mapped isoparametric elements. Figure 1 shows the classical FEM mesh mapping strategy, with a local context of mapping and global node coupling. Isogeometric analysis adopts the global mapping concept where the whole parametric space is mapped to the physical domain. Figure 2 shows the basic IGA strategy. Is important to note that FEM performs the discretization process in the physical domain, while IGA performs the discretization in the parametric domain which allows the global mapping by NURBS exact describing geometry.

2.1 B-spline shape functions

The fundamental functions to create the solution and geometric space in isogeometric analysis are the b-splines shape functions. In parametric space, the subdivision of elements is performed by the knots concept. These knots consists in a set of coordinates in the parametric space which holds the number of functions and its continuity. The b-splines shape functions is a set of parametric polynomials defined by a polynomial degree p and a set of knots $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$ where knots can appear repeatedly but in an ascending order, and n is the number of b-splines shape functions. The shape functions are recursively defined by De-Boor (1972):

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{otherwise} \end{cases} , \tag{1}$$

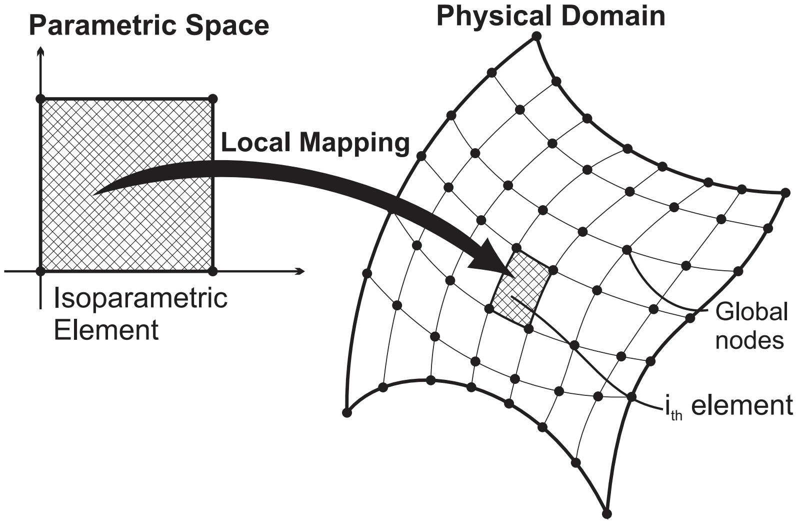


Figure 1: Classical mapping scheme in FEM

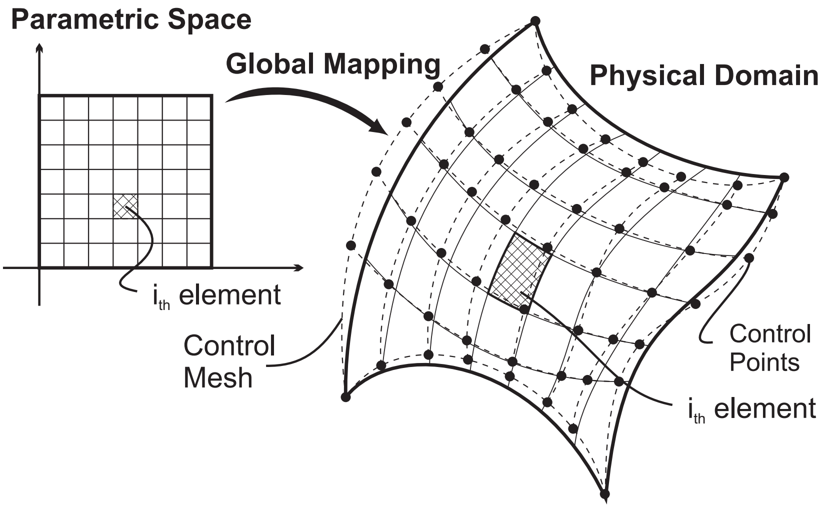


Figure 2: IGA mapping strategy

starting from an null degree iteration. The iteration for p_{th} degree is given by:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{2}$$

The b-splines shape functions has some important properties. Namely:

- 1) The continuity of the b-splines set in the knot ξ_i are related with its multiplicity in Ξ, i.e the functions have $p - m_i$ continuous derivatives at the knot ξ_i, where m_i is the knot multiplicity;
- 2) The support of a NURBS shape function $N_{i,p}$ is defined in the interval $[\xi_i, \xi_{i+p+1}]$;
- 3) The b-splines shape functions follows the partition of unity in parametric space, i.e $\sum_{i=1}^{n} N_{i,p}(\xi) = 1$.

The usual knot configuration in isogeometric analysis is to use the open concept Farin

(1999), i.e $p+1$ edge knots. Interior knots also controls the continuity, being that ξ_i repeated p times created an C^0 continuity in that point.

2.2 Curve representation and NURBS functions

A curve defined in a physical domain is mapped by a set o b-splines shape functions by the relation:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} N_{i,p}\,\mathbf{B}_i, \tag{3}$$

where $\mathbf{B}_i$ is the i_{th} control point set which defines the curve.

Non uniform rational b-splines (NURBS) functions are originated from b-splines, to increase CAD representation of objects. The control points, used to map a curve in a physical domain appear weighted, being possible to represent a more complex curve than a *beziÃľr* curve Piegl and Tiller (1997). The NURBS functions are defined as a weighted rational set $R_{i,p}$, based in a b-spline set $N_{i,p}$. Given a set of n control points $\mathbf{B}$ and a set of n weights $\mathbf{w}$, NURBS set is the defined as:

$$R_{i,p} = \frac{N_{i,p}\,w_i}{\sum_{\hat{i}=}^{n} N_{\hat{i},p}\,w_{\hat{i}}}. \tag{4}$$

A curve origined by a NURBS set is defined by:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_{i,p}\,\mathbf{B}_i. \tag{5}$$

The insertion of weights and the rational concept implied a more flexible possibility to perform curves construction, being possible to control curvature radius, which comes from the projective concept given by weights Farin (1999). From a practical view of implementation in isogeometric analysis (fig. 3), the construction of a NURBS curve can be though as a transformation which maps a point in the parametric space ξ directly to the physical space $\mathbf{C} = (x, y)$.

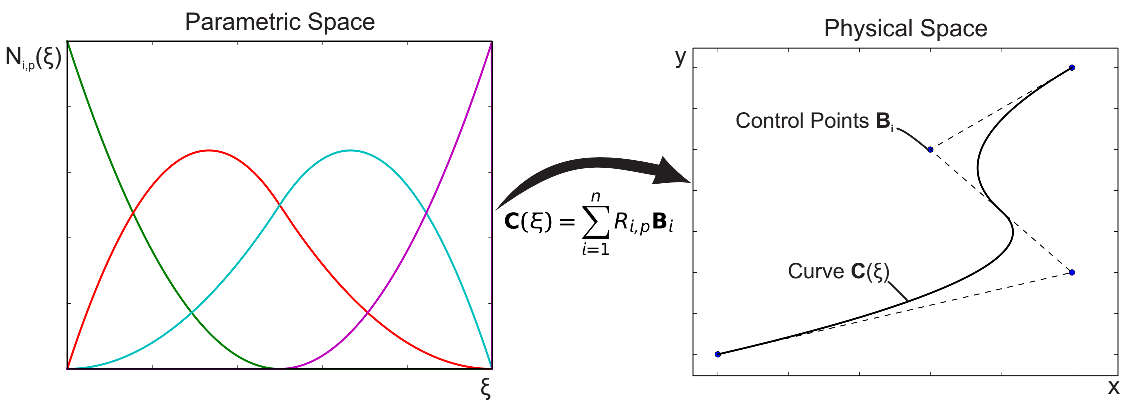


Figure 3: Basic NURBS curve mapping scheme

2.3 Surface Representation

In a analogous way as curve representation, a NURBS surface is defined by:

$$\mathbf{S}(\xi,\eta) = \sum_{j=1}^{m}\sum_{i=1}^{n} \hat{R}_{i,j,p,q}(\xi,\eta)\,\mathbf{B}_{i,j}, \tag{6}$$

with

$$\hat{R}_{i,j,p,q}(\xi,\eta) = \frac{R_{i,p}(\xi)R_{j,q}(\eta)w_{i,j}}{\sum_{\hat{j}=1}^{m}\sum_{\hat{i}=1}^{n} R_{\hat{i},p}(\xi)R_{\hat{j},q}(\eta)w_{i,j}}, \tag{7}$$

where $R_{i,p}(\xi)$, $R_{j,q}(\eta)$ denotes different NURBS spaces on ξ and η.

2.4 Refinements in IGA and CAD

Refinement procedures in IGA are related with the modifications of the parameters Ξ (knot vector) and p (polynomial degree). In CAD procedures, several refinement techniques are discussed and exampled by Piegl and Tiller (1997) aiming to locate control points and weights properly by impose equality in curve equations. These algorithms implies in automatic mesh refinement.

Considering NURBS curve refinement in isogeometric analysis, three kind of basic refinement procedures are possible to perform (Cottrell et al., 2007), namely: h refinement, *p* refinement and *k* refinement. Like FEM, IGA h refinement is related with point increasing without change polynomial degree. In IGA this means to add new knots in the knot vector and recompute the functions, by the recursive formula (eqs. 1,2) and the points and weights by the algorithms given by Piegl and Tiller (1997). This results in a higher amount of b-splines shape functions, control points, weights and degrees of freedom in the system. Refinements *k* and *p* are related with knot vector changing and also polynomial degree increasing. The basic difference between these refinements techniques is that *p* refinement performs the polynomial degree increasing aiming to maintain b-spline shape functions continuity at certain knots, a necessary procedure to maintain some kind of geometries like squares and cusps. An detailed guide about geometry and mesh procedures in NURBS construction is given by Farin (n.d.). Refinement *k* is related with degree elevation followed by knot insertion, which generates higher continuity functions at the knots. Several works present the advantage of *k* refinement over *p* and *h* refinements in terms of convergence, due mostly to its continuity growing (property 1 of eqs. 1 and 2) with fewer degrees of freedom than *p* refinement (Cottrell et al., 2007, Hughes et al., 2005). In structural dynamics the works of Cottrell et al. (2006) and recently Rauen et al. (2017) reinforce this statement.

3 MESH GENERATION ALGORITHM

3.1 Color Filtering and Boolean Image

An image can be mathematically treated as a matrix I_m which carries image information pixel by pixel. The matrix I_m naturally has dimensions n_x, n_y, where denotes the number of

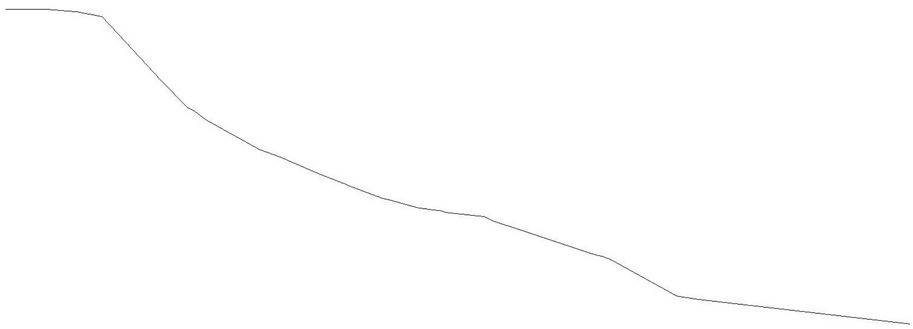


Figure 4: Example image representation

pixels in x, y direction. Several edge detection methods and topology schemes are discussed by Vizilter et al. (2015) and an interesting state of art is presented by Maini and Aggarwal (2002). But the technique in this algorithm extends to take some cares in the image production. First the image representation should be draw at a high color difference between the background a low thickness line is recommended. For purposes of accurate generation of the cloud points is also recommended that the image length has a correlation with known measures in the real space. A example is given in fig. 4

The boundary creation routine in to detect the points with high difference on the color gradient (Dutta and Chaudhuri, 2009). This verification can be done pixel wise or in a discrete manner by apply Sobel Operator (Maini and Aggarwal, 2002), which turns feasible to high definition images. In a pixel wise manner of verification, the coordinates (i, j) of matrix I_m which contains drawing points is stored on vector P_i.

$$\mathbf{P} = \left\{ P_1 \quad P_2 \quad \dots \quad P_m \right\} = \left\{ \begin{pmatrix} i_1 \\ j_1 \end{pmatrix} \quad \begin{pmatrix} i_2 \\ j_2 \end{pmatrix} \quad \dots \quad \begin{pmatrix} i_m \\ j_m \end{pmatrix} \right\} \tag{8}$$

The point cloud P is so mapped in the real space by the scaling equations:

$$X = X_0 + \frac{X_1 - X_0}{n_x} j \tag{9}$$

$$Y = Y_1 + \frac{Y_0 - Y_1}{n_y} i, \tag{10}$$

where X_0, X_1, Y_0 and Y_1 represents the coordinates of the image limit vertex on the real space by the scheme shown in fig. 5.

Figure 5: Image edges on scaling process

By the adapting of matrix coordinates on $\mathbf{P}$ on the scaling process, a cloud point X_b, Y_b is created by the boundary extraction. The figure 6 shows an example of a dense point cloud by an image analysis with $1707x815$ pixels.

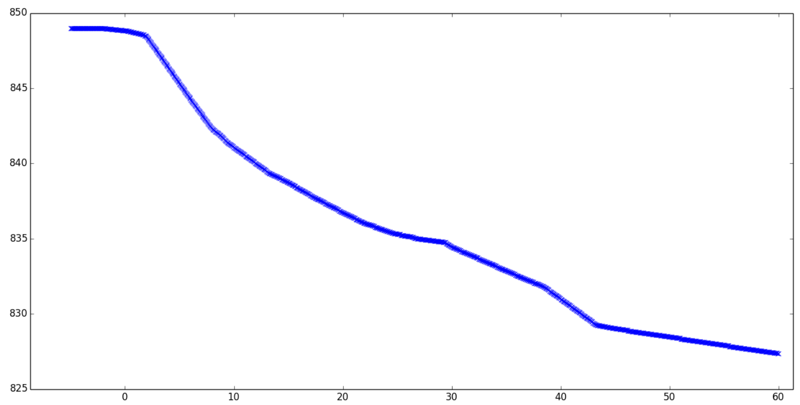


Figure 6: Scaled point cloud

3.2 Cloud points on rectangular based surfaces

As a generalization to a surface representation a color mask can be applied with edge detection (Dutta and Chaudhuri, 2009) aiming to identify the boundary limits related with the parametric surface. Figure 7 brings an example of these scheme.

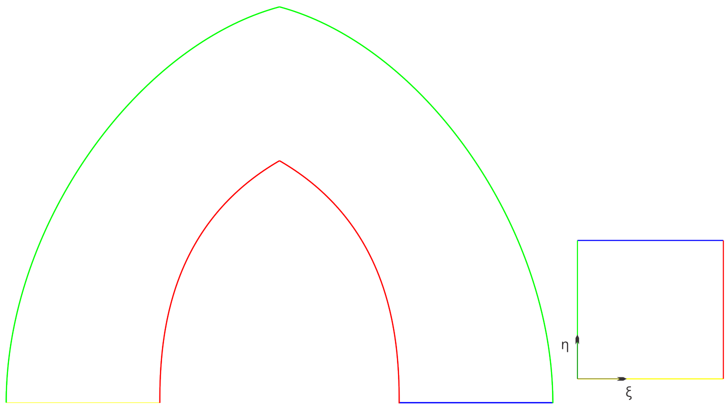


Figure 7: Boundary color identification example

The color filtering allows to create point clouds of each surface boundary in a separated way. Considering a coarser mesh, the points can be discretized proportionally by a set of collocation points in the parametric space:

$$\bar{u} = \{\xi_1, \xi_2, \ldots, \xi_{\bar{n}}\} \tag{11}$$

$$\bar{v} = \{\eta_1, \eta_2, \ldots, \eta_{\bar{m}}\} \tag{12}$$

Suposing point representations for each surface boundary P_B for bottom limit (yellow), P_R for right limit (red), P_T for top limit (blue) and P_L for left limit (green), the cloud on boundaries is set by choose points in the same proportion on $\bar{u}$ and $\bar{v}$, i.e:

$$\mathbf{L} = P_L(l) \ \text{ with } \ l = round(\bar{v}n_l) \tag{13}$$
$$\mathbf{R} = P_R(r) \ \text{ with } \ r = round(\bar{v}n_r) \tag{14}$$
$$\tag{15}$$

where n_l, n_r, n_b, n_t denotes the total number of cloud points on each boundary. The procedure to insert points inside the surface is to perform distribution of $\bar{u}$ on the mean of y points:

$$\begin{pmatrix} X_1 & X_2 & \ldots & X_{\bar{n}} \\ Y_1 & Y_2 & \ldots & Y_{\bar{n}} \end{pmatrix} = \bar{u} \begin{pmatrix} R_x(i) - L_x(i) \\ R_y(i) - L_y(i) \end{pmatrix} \tag{16}$$

with $i = 1, \ldots, \bar{m}$.

By taking this philosophy fig. 8 shows two examples of discrete distribution of a point cloud on the example of fig. 7, a scaling of $22x15$ measure units. The distribution logic on the collocation points allows also to create non uniform point clouds.

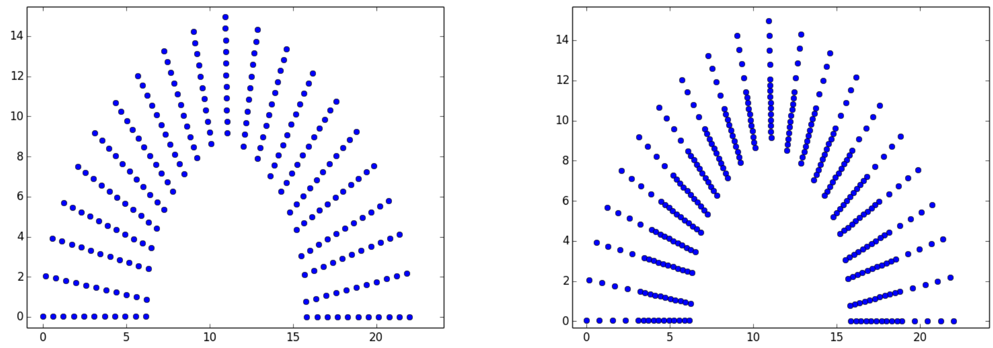


Figure 8: Uniform and deformed point clouds on surface

4 FITTING ALGORITHM

The mesh construction algorithm is to interpolate a NURBS set on the point cloud by least-squares minimization technique (Varady et al., 1997), which gives to the image a reverse engineering context where accuracy parameters should be investigated.

The matrix representation on the least-squares by NURBS approach is developed by Ma and Kruth (1998), which use a b-spline set as input, taking control points and weights as unknows:

$$\begin{cases} \mathbf{b}^T(\cdot_i)\mathbf{X} = \bar{x}_i \cdot \mathbf{b}^T(\cdot_i) \cdot \mathbf{w} \\ \mathbf{b}^T(\cdot_i)\mathbf{Y} = \bar{y}_i \cdot \mathbf{b}^T(\cdot_i) \cdot \mathbf{w} \end{cases} \tag{17}$$

with $i = 1, 2, \ldots, m$, where m now is the amount of cloud points, $\bar{x}_i, \bar{y}_i$ is the coordinates of the $i_t h$ point cloud, and $\mathbf{b}^T(\cdot_i)$ is the b-spline parameters on $(\cdot_i) = \bar{u}_i$ for curves and $(\cdot_i) = (\bar{u}_i, \bar{v}_i)$ for surfaces. The parameters $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{w}$ denotes the control point coordinates and the weight set. The system is can be algebraic rewritten as:

$$\begin{bmatrix} \mathbf{B} & \mathbf{0} & -\bar{\mathbf{X}} \cdot \mathbf{B} \\ \mathbf{0} & \mathbf{B} & -\bar{\mathbf{Y}} \cdot \mathbf{B} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{w} \end{bmatrix} = [\mathbf{0}]_{3nx1}, \tag{18}$$

with

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1(\cdot_1) & \mathbf{B}_2(\cdot_1) & \ldots & \mathbf{B}_n(\cdot_1) \\ \mathbf{B}_1(\cdot_2) & \mathbf{B}_2(\cdot_2) & \ldots & \mathbf{B}_n(\cdot_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_1(\cdot_m) & \mathbf{B}_2(\cdot_m) & \ldots & \mathbf{B}_n(\cdot_m) \end{bmatrix} \tag{19}$$

and

$$\bar{\mathbf{X}} = diag\{\bar{x_1},\ \bar{x_2},\ \ldots,\ \bar{x_m}\} \tag{20}$$

$$\bar{\mathbf{Y}} = diag\{\bar{y_1},\ \bar{y_2},\ \ldots,\ \bar{y_m}\} \tag{21}$$

The fitting is so splited in two subproblems by multiply the system by $\mathbf{B}^T$ (Ma and Kruth, 1998). The weights are firstly solved by:

$$\mathbf{M} \cdot w = \mathbf{0} \tag{22}$$

with

$$\mathbf{M} = \mathbf{B}^T\bar{\mathbf{X}}^2\mathbf{B}+\mathbf{B}^T\bar{\mathbf{Y}}^2\mathbf{B}-(\mathbf{B}^T\bar{\mathbf{X}}\mathbf{B})(\mathbf{B}^T\mathbf{B})^{-1}(\mathbf{B}^T\bar{\mathbf{X}}\mathbf{B})-(\mathbf{B}^T\bar{\mathbf{Y}}\mathbf{B})(\mathbf{B}^T\mathbf{B})^{-1}(\mathbf{B}^T\bar{\mathbf{Y}}\mathbf{B}). \tag{23}$$

The weight estimation is based on the eigenvectors of $\mathbf{M}$ and no necessary leads to positive values. Algebraic treatment of these eigenvectors are developed by Ma and Kruth (1998) aiming to obtain positive weights by a linear combination of eigenvectors of $\mathbf{M}$. An advantageous fact in this weight obtaining algorithm is the need to invert the matrix $\mathbf{B}^T\mathbf{B}^T$.

By definition of the weight set $\mathbf{w}$, the control points are defined by solve the system:

$$\begin{cases} \mathbf{B}^T\mathbf{B}^T \cdot \mathbf{X} = \mathbf{B}^T\mathbf{B}^T \cdot \bar{\mathbf{X}}\mathbf{w} \\ \mathbf{B}^T\mathbf{B}^T \cdot \mathbf{Y} = \mathbf{B}^T\mathbf{B}^T \cdot \bar{\mathbf{Y}}\mathbf{w} \end{cases} \tag{24}$$

5 NUMERICAL EXPERIMENTS

5.1 Curve Example

The geometry fitting algorithm is tested for both point clouds exampled in section 4. First the curve fitting with 80 cloud points and b-spline basis in degree $p = 2$ is tested under 10 and 20 elements, the curve results with the points and weights defined is plotted against the image cloud, in red line, by fig. 9.

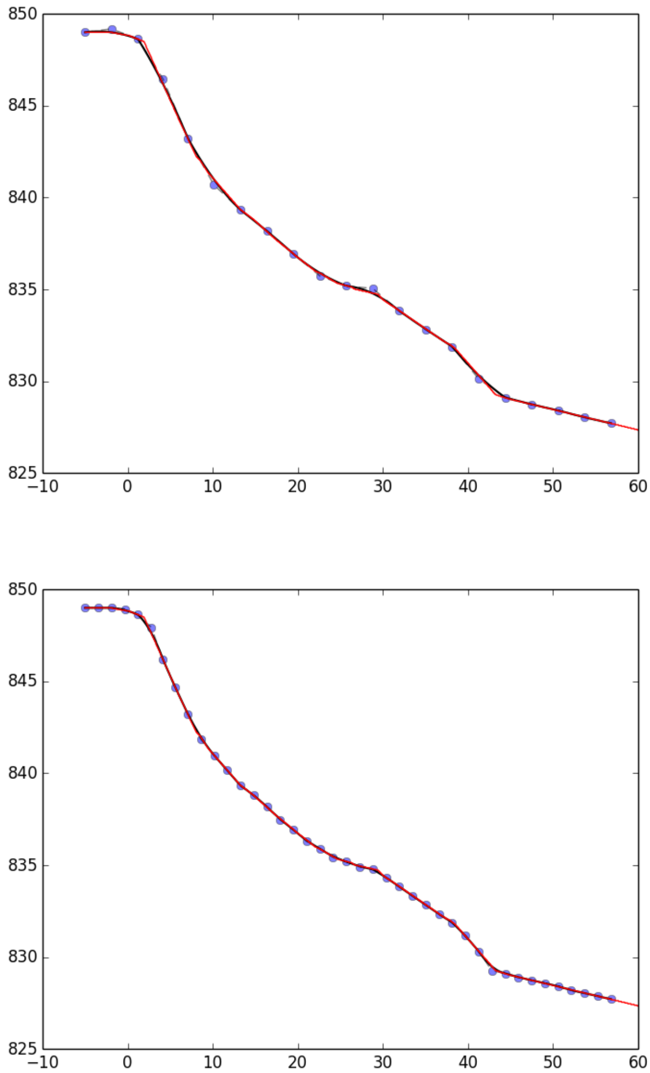
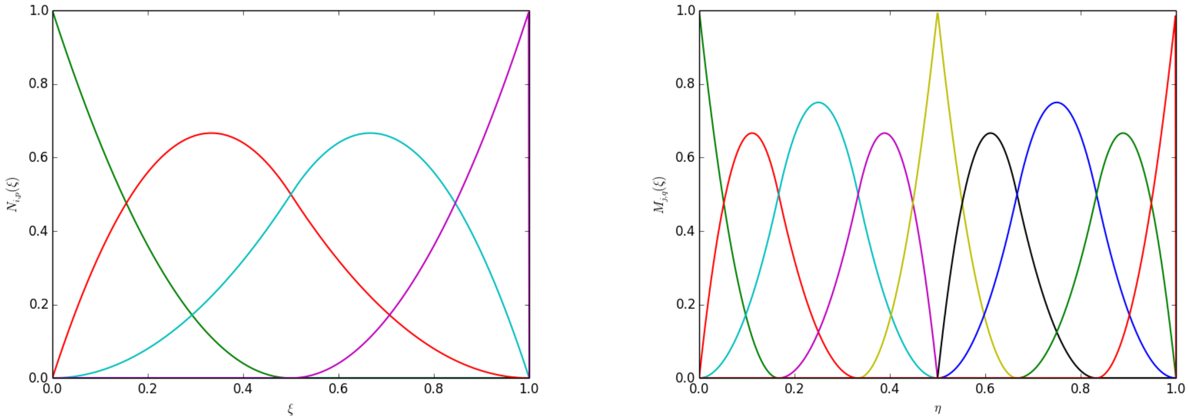


Figure 9: Curve fitting results with b-splines $p = 2$

The image shows that refinement on b-splines basis accurate adequate to the image boundary.

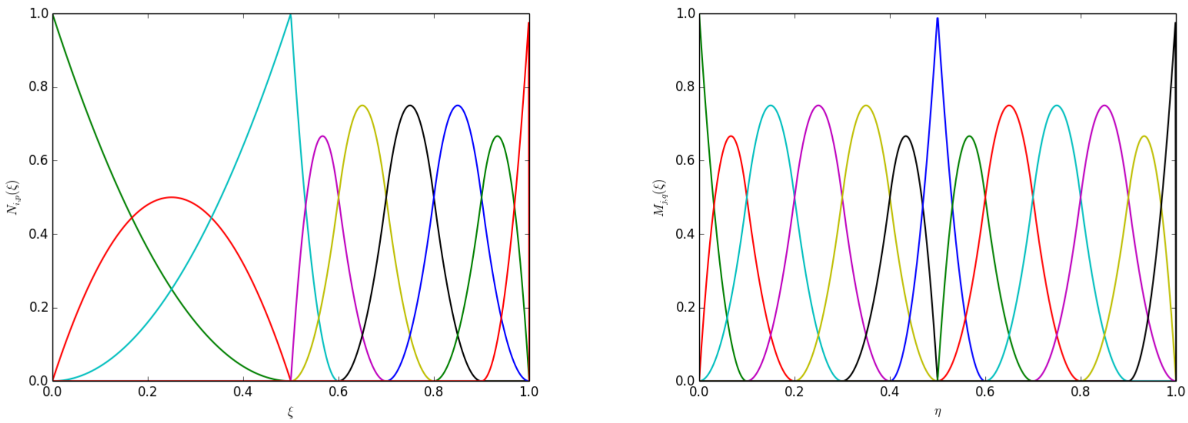
5.2 Surface Example

To test the algorithm for surface fitting, the Church Arc on fig. 8 is exampled under two b-splines basis: one consider a coarse mesh and other considering a deformed mesh to test the algorithm capabilities. Figures 10 and 11 shows graphically the input on basis functions:



Basis functions on ξ Basis functions on η

Figure 10: Basis function on coarse mesh on surface example



Basis functions on ξ Basis functions on η

Figure 11: Basis function on deformed mesh on surface example

Figure 12 shows the coarse and deformed mesh obtained with the fitting process. Figures 13 and 14 shows the fitted surface boundary compared with the dense cloud point vector obtained by the image.

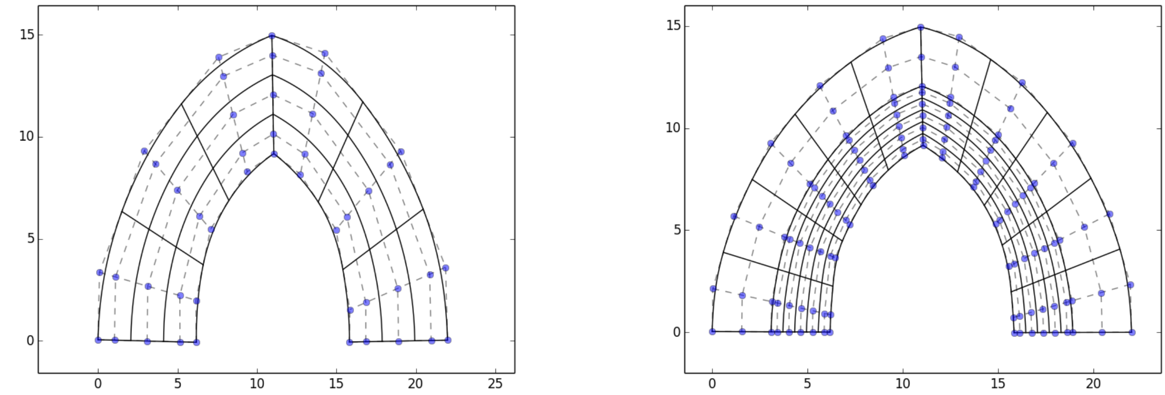


Figure 12: Basis function on coarse mesh on surface example

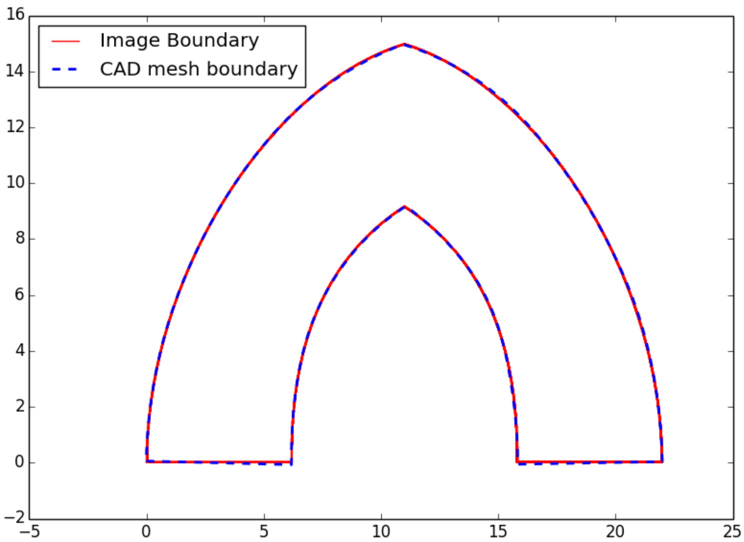


Figure 13: Fitted boundary obtained by coarse mesh compared with image boundary

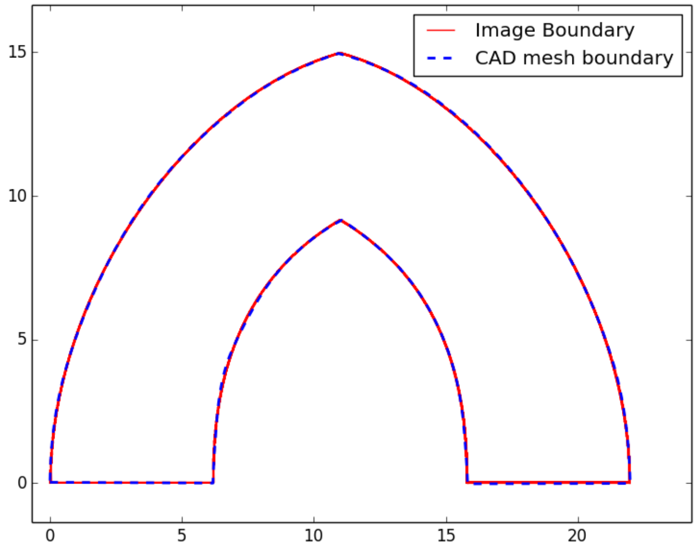


Figure 14: Fitted boundary obtained by deformed mesh compared with image boundary

6 CONCLUSIONS

This paper developed a coherent cloud point generation algorithm based on image analysis in a reverse engineering point of view. The numerical experiments shows a good performance on the cloud point data generation and also a accurate capacity on NURBS fitting. Even with accurate condition, the exploration on weight estimation is suggested on future works, where in the development in the performed experiments weight obtaining appeared to be a sensible point and a possible computational issue. The algorithm presented a capability to describe a CAD mesh by an image, being a useful tool in mesh creation for isogeometric analysis and NURBS based numerical methods.

REFERENCES

Cottrell, J. A., Bazilevs, Y. and Hughes, T. J. R. (2009), *Isogeometric Analysis: Toward Integration of CAD and FEA*, 1st edition edn, John Wiley & Sons, USA.

Cottrell, J. A., Hughes, A. T. J. R. and Reali, A. (2007), 'Studies of refinement and continuity in isogeometric structural analysis', *Computer Methods in Applied Mechanics and Engineering* **196**, 4160–4183.

Cottrell, J. A., Reali, A., Bazilevs, Y. and Hughes, T. J. R. (2006), 'Isogeometric analysis of structural vibrations', *Computer Methods in Applied Mechanics and Engineering* **195**, 5257–5196.

De-Boor, C. (1972), 'On calculation of b-splines', *Journal of Approximation Theory* **6**, 50–62.

Dutta, S. and Chaudhuri, B. B. (2009), A color edge detection algorithm in rgb color space.

Farin, G. (n.d.), *Curves and Surfaces for CAGD: a practical guide*, 5th edition edn, Academic Press.

Farin, G. E. (1999), *NURBS: from projective geometry to practical use*, 2 edn, A K Peters, Natick.

Hughes, T. J. R., Cottrell, J. A. and Bazilevs, Y. (2005), 'Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement', *Computer Methods in Applied Mechanics and Engineering* **194**, 4135–4195.

Ma, W. and Kruth, J. P. (1998), 'Nurbs curve and surface fitting for reverse engineering', *International Journal of Advanced Manufacturing Technology* **14**, 918–927.

Maini, R. and Aggarwal, H. (2002), 'Study and comparison of various image edge detection techniques', *International Journal of Image Processing* **3**, 1–12.

Piegl, L. and Tiller, W. (1997), *The NURBS Book*, Springer-Verlag, Germany.

Rauen, M., Machado, R. D. and Arndt, M. (2017), 'Isogeometric Analysis of Free Vibration of Framed Structures : Comparative Problems', *Engineering Computations* **34**(2).

Shojaee, S., Valizadeh, N., Izadpanah, E., Bui, T. and Vu, T. V. (2012), 'Free vibration and buckling analysis of laminated composite plates using the NURBS-based isogeometric finite element method', *Composite Structures* **94**(5), 1677–1693.

Varady, T., Martin, R. R. and Cox, J. (1997), 'Reverse engineering of geometric models - an introduction', *Computer-Aided Design* **29**(4), 255–268.

Vizilter, Y., PytâĂŹev, Y., Chulichkov, A. and Mestetskiy, L. (2015), 'Morphological image analysis for computer vision applications', *Computer Vision in Control Systems* .