Multiple Incident Splines (MISs) algorithm for topological reconstruction of 2D unordered point clouds

R. Furferi, L. Governi, M. Palai and Y. Volpe

Abstract - An automatic 3D model retrieval from freehand conceptual sketches is a key target for both commercial software houses and academic research. Unfortunately, most of the approaches are not suitable for properly translating stylistic sketches into 3D models. In order to carry out this 3D model conversion, the first task to be dealt with is to turn raster data (3D or 2D free-form curves) into vectorial ones. Such a task represents a key issue which has been addressed by a number of authors but still far to be exhaustively worked out. To address this challenge, this work presents a new method that allows to fit 2D unordered point cloud data with Multiple Incident Splines (MISs). At the heart of the proposed approach are two main procedures: the first one is based on Euclidean Minimum Spanning Tree (EMST) and Principal Component Analysis (PCA) for detecting the main local directions of the point cloud and to order its points while preserving original topology; the second is meant to fit ordered point clouds with spline curves providing a robust intersection and vertex detection.

The proposed methodology, tested on a number of case studies, proves to preserve the original topology more efficiently than alternative techniques supplied by commercial vectorization software packages.

Keywords— Image Processing, freehand sketches, curve reconstruction, unorganized point cloud, styling.

I. Introduction

In the last decades design received a major contribution from the use of computers, with remarkable impact in terms of time, cost and reliability. Computer Aided Design (CAD) affords effective solutions for automating the whole product development chain process. In most cases, CAD packages may be considered a starting point for new product development; however, when concept development and/or stylistic design is a primary issue, drawings are executed by stylists mainly using freehand sketches that are not intended as a finished work (Fig.1). In fact, designers produce a multitude of rough sketches early in the design process as a way to explore different shapes and styles [1]. These sketches are handed over to computer modelers that use it as a visual reference for creating 3D models since a key advantage of the conventional CAD modelling rely to faster modification of models and to a more realistic feeling in understanding object style.

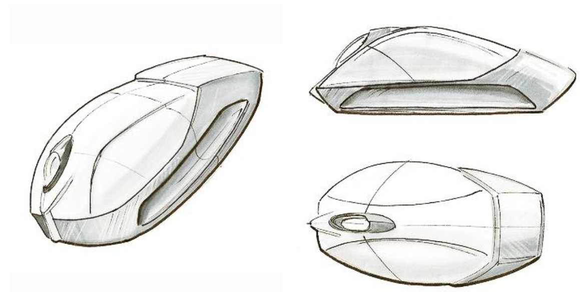


Fig. 1 – Example of a freehand sketch

As a consequence, multiple iterations between style designers and CAD modelers are necessary in order to achieve the desired result [2]. Accordingly, since tridimensional representation of a concept sketch is one of the crucial phases in product development process, it results to be very time consuming and, evidently, characterized by a certain amount of subjectivity in the interpretation of concept sketches. Moreover, due to the significant effort and expertise required for 3D modeling, only a few select candidate concepts will typically pass to the next stage, while many others are prematurely abandoned.

Nowadays, conversion from 2D sketches into 3D CAD models is a key target both for commercial software houses like Dassault Systems® and Autodesk® and for academic research [3-13]. Unfortunately, to the best of authors' knowledge, most approaches are applied to mechanical drawings described by orthogonal projection views where features are, commonly, points, lines, arcs and circumferences. Style objects are, indeed, represented by free-form curves (splines); as a consequence such approaches cannot be suitable for properly translating stylistic sketches into 3D models. In order to carry out this 3D model conversion the first task is to turn raster data (3D or 2D) representing free-form curves into vectorial ones.

One of the most common problem to be faced when vectorizing raster data (e.g. images obtained from blueprints scanning) is the fitting of analytical curves to pixel clusters representing drawing sketches.

Fitting vectorial curves to point datasets has been widely studied and many techniques, theories and commercial software tools have been developed. Depending on input data typology, the research approaches proposed in the scientific literature may be divided into two main families:

1) Ordered point cloud-based approaches; when the order of data points is assumed to be known, the fitting curve can be easily obtained by minimizing error functions or by means of computational geometry methods [14-17].

2) Unordered point cloud-based approaches. For many practical problems, point cloud data are, usually, unordered i.e. the order of data points is unknown. In this case the key problem is to transform the unordered point cloud into an ordered one. The order, of course, needs to be related to the geometry of the point cloud. When the order is known, curve reconstruction can be achieved by interpolating or fitting an ordered point set with a parametric curve.

A number of approaches have been developed for solving this issue; for instance, Pottmann [18] maps data points to binary images, then fits a curve to the image's medial axis. Clustering the point cloud data is another commonly used approach: Yan [19], for instance, presents a fuzzy curve-tracing algorithm that works by means of several clusters identified by fuzzy algorithms. In a more recent work, Liu et al. [20] devise a new algorithm based on the idea that a spline curve can be made to crawl and stretch along the curve shape defined by a point cloud.

Another typical issue for unordered data sets is the presence of noise. Levin [21] provides a Moving Least Squares (MLS) approach so as to clean noisy point cloud data. Lee [22] improves MLS by means of appropriate neighborhoods for regressions; the main weakness of this method is represented by the computational cost of iterating MLS.

A common approach, which can also be found in commercial vectorization software packages, is to convert the original color or gray scale image into a binary ones and, subsequently, to perform a thinning procedure prior to vectorization step. Such an approach, though usually quite effective, is unsuitable in case image gray levels convey relevant geometrical information. Unfortunately, when point clouds present intersections or vertexes, most approaches tend to fail providing the correct topology. Actually, the problem with thinning algorithms is how to deal with junctions [23]; this aspect is confronted in Hough Transform (HT), Orthogonal Zig-Zag (OZZ), Sparse Pixel Vectorization (SPV) and Mesh based methods [24-28]. The quality of lines detected by means of HT-based implementations is far less precise for slanted lines. Moreover, the HT-based methods can yield bars only, and cannot generate polylines. OZZ is able to recovery cross junctions and edges but, like in HT based methods, is affected by poor quality of line geometry. Mesh based approach is also appropriate for junction analysis but results to be a non-trivial challenge when noisy drawings are processed. SPV method and its implementation seem to be the most suitable making use of a series of iterative steps and continuation conditions.

As a consequence alternative approaches for fitting 2D point clouds taking into account intersections and vertexes are still advisable, especially in the outlook of extending 2D reconstruction method to 3D one.

The main idea of the present paper is to provide an algorithm able to fit unordered point cloud data with a set of weighted B-spline curves. The provided method is capable of 1) computing B-spline weights on the basis of raster data gray levels, 2) identifying and reconstructing vertexes and cross junctions by fitting point clouds with multiple incident splines and 3) strongly limiting topological errors in B-spline fitting by using Euclidean Minimum Spanning Tree (EMST) technique.

The provided method has been carried out, by means of a series of Matlab® routines, according to the following tasks:

- Image processing
- EMST + PCA – based polyline extraction
- Gray-level weighted B-spline curves fitting
- Multiple Incident Splines (MISs) algorithm

In order to deeply benchmark the performance of the proposed method, a number of tests have been carried out confronting the devised method with two commercial vectorization software packages. Due to their vectorization techniques, that make use of 2D thinning processes as necessary preprocessing tasks, the obtained vector entities are affected by common thinning problems as spurious branches or topology alterations. These limitations are overcome by using MISs algorithm described in this work.

II. Image Processing

Let $\boldsymbol{J}$ be a digital raster image obtained, for instance, by scanning a sketch or a blueprint (Fig. 1). Generally speaking, the image is treated as a grid of discrete elements (pixels), ordered from top to bottom and left to right.

Sketch contours are represented by a sequence of pixels $j_{m,n}$ characterized by discrete couples of coordinates referred to an orthogonal coordinate system centered on the top left corner of the image. Each pixel is, on its side, described by a triplet of values in the RGB color space [29] or, possibly, by a single brightness value. In the first case, a simple transformation from RGB to gray scale image is required (Fig. 2).

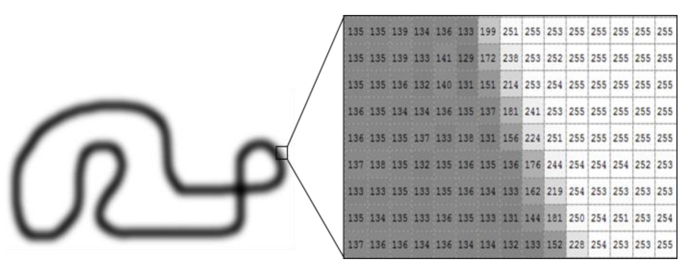


Fig. 2 – Example of a gray scale image. Some details are highlighted.

An adaptive thresholding method [30] is applied according to [31] to determine the pixels composing the sketch contour.

The result of this thresholding is a binary image $\boldsymbol{B}$ where the sketch contour is represented by white pixels (i.e. $b_{m,n}$ = 0) while the background is represented by black pixels (i.e. $b_{m,n}$ = 1).

According to [31], it is possible to assign a weight to each contour pixel on the basis of its brightness by means of the Hadamard product of the 1-complement B' of B and the inverse of the original image J . The result is a new image W (Fig. 3) whose non-zero elements $w_{m,n}$ are the weight values of the contour pixels.

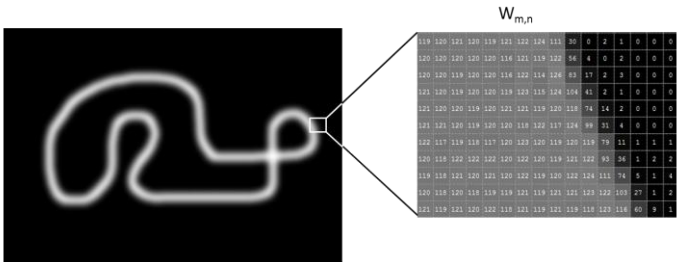


Fig. 3 – Image W. Non zero values represents the weight values of the contour pixels to be used for weighted B-spline reconstruction.

III. EMST+PCA BASED POLYLINE EXTRACTION

In order to extract the local geometric features of the point cloud, a combination of EMST and Principal Component Analysis (PCA) has been used with the aim of overcoming the limitations provided in [31].

By definition, point cloud consists of the only pixels of image B' satisfying the condition $b_{m,n}$ = 1. Obviously, the number of points composing the cloud is:

$$\beta = \sum_{i,j} b_{i,j} \tag{1}$$

The method described in [31] first defines a variable sized cluster C centered in a "seed point" $b_{h,k}$ randomly extracted from the point cloud. Successively, the cluster centroid χ is determined as the weighted mean values of all the points inside the cluster C. Expressing the coordinates of the pixels belonging to the cluster $\boldsymbol{C}$ in a new coordinate system whose origin coincides with χ and whose axis are parallel to the original ones it is possible to define a matrix ψ of the new pixel coordinates. Such a matrix can be processed by means of PCA in order to evaluate its eigenvalues and eigenvectors (i.e. the principal inertia axes of the cluster).

If the ratio between the smallest and the greatest eigenvalue tends to 1 (e.g. is greater than 0.9), the cluster is considered to have no "preferred" orientation; thus the cluster radius has to be increased. Otherwise the cluster radius is considered to be "correct" and, according to the method provided in [32] it is possible to determine the Main Local Direction (MLD) vector of the cluster itself. With this approach it is possible to define a Size Variable Cluster (SVC) whose MLD is determined (Fig. 4).

The two intersection points between the MLD and the circumference with radius r define the geometric centers of two new SVCs (C_{p1} and C_{f1}) to be processed as shown in Fig 5.

Analogously to the first cluster, the two centroids χ_{f1} and χ_{p1} of clusters C_{p1} and C_{f1} are determined. The segments from χ_{p1} to χ and from χ to χ_{f1} define a portion of a starting polyline.

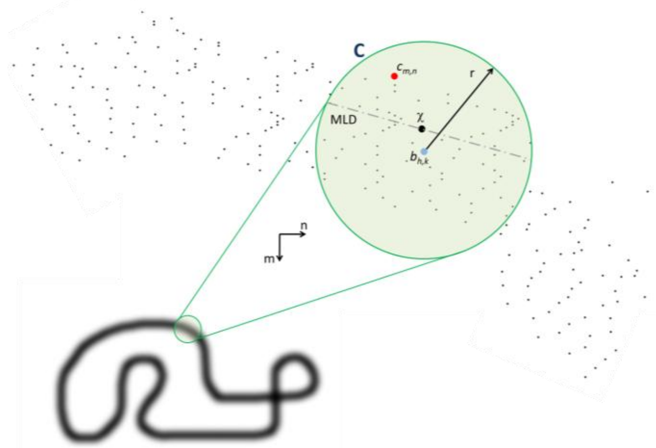


Fig. 4 – First step of the polyline construction procedure applied to a cluster centered in the "seed point" $b_{h,k}$.

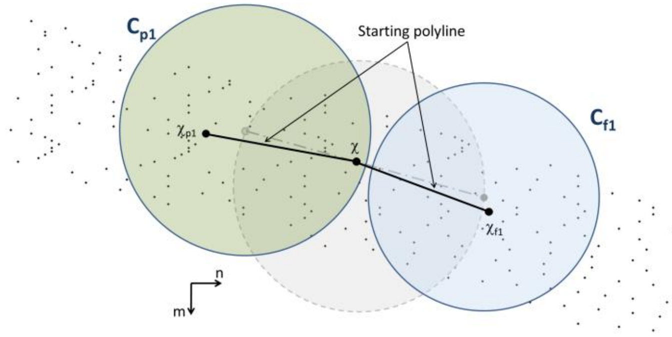


Fig. 5 – Second step of the polyline construction procedure.

Starting from χ_{f1} the procedure is iteratively carried out until all the points of the cloud are used. The final result is a polyline to be fitted by means of weighted B-splines.

Such a method may fail when a cluster encloses sets of points belonging to different topological entities (Fig. 6).

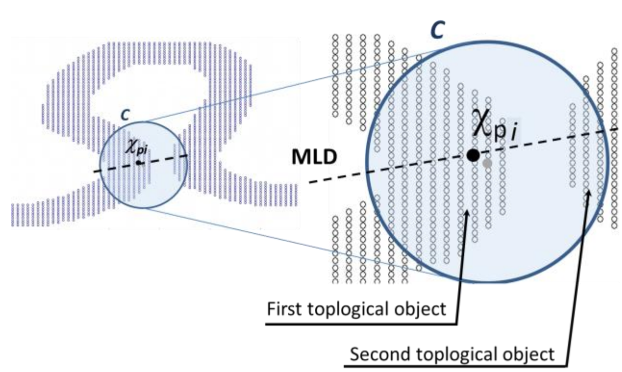


Fig. 6 – Generic cluster C enclosing set of points belonging to two different topological entities.

This is due to the fact that the MLD may be wrongly evaluated as depicted in Fig. 7.

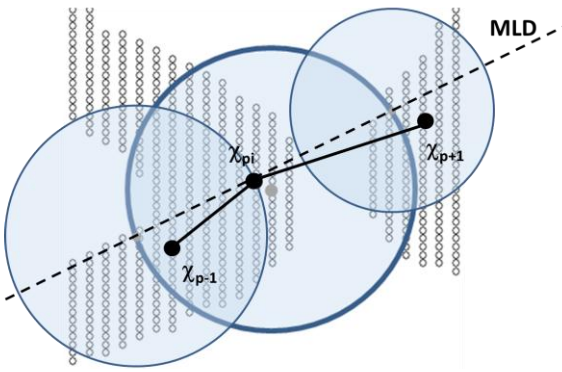


Fig. 7 – Wrong evaluation of the MLD resulting into a wrong polyline construction.

In this eventuality, the new cluster jump from a topological entity to another thus resulting in a wrong polyline reconstruction for the whole point cloud (Fig. 8).

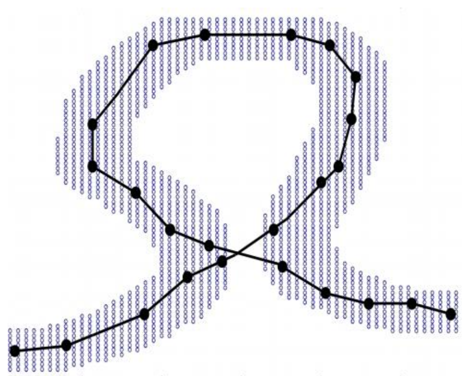


Fig. 8 – Wrong polyline construction.

In order to overcome the limitations of the described method, the EMST technique is applied to the point cloud.

As widely known, given an undirected graph (i.e. not ordered pairs $b_{m,n}$ of vertices belonging, for instance, to a point cloud), a spanning tree of that graph is a sub-graph connecting all the vertices together. A weight to each edge can be assigned in order to represent how unfavorable it is, and a weight to a spanning tree can be also computed as the sum of the weights of the edges in it. The minimum spanning tree (MST) is the spanning tree with minimum weights with respect to any others. If the weight of the edge between each pair of points is the Euclidean distance between those two points the MST becomes the EMST. The result of the EMST application to a point cloud is a set of connected points (using lines) such that the total length of all the lines is minimized and any point can be reached from any other by following the lines (see Fig 9). Mathematically speaking, the EMST provides:

- a $\beta \times \beta$ Adjacency Matrix [33] (AM) where the generic non-diagonal element am_{ij} is equal to 1 if and only if the ith and the jth points are directly connected in the EMST graph;
- a $\beta \times \beta$ weight matrix Λ whose generic element λ_{ij} of each edge i.e. the Euclidean distance between each pair of points (i, j) directly connected.

Fig. 9 – Result of the EMST method application.

According to the results described above, it is possible to define into cluster C a sub-cluster C' as the locus of points $c'_{m,n}$ centered in c_0 satisfying the two following conditions:

$$(h-m)^2 + (k-n)^2 \le r^2 \tag{2}$$

$$\Gamma_{c_0,c_{m,n}} \le \omega \tag{3}$$

Where:

- (m,n) are the coordinates of $c'_{m,n}$;
- (h,k) are the coordinates of c_0;
- r is the cluster radius;
- $\Gamma_{c_0,c_{m,n}}$ is the shortest path between c_0 and $c_{m,n}$ evaluated by means of the Adjacency Matrix AM and the weight matrix Λ;
- ω is a user defined tolerance. In the present work the value of such tolerance is set equal to $3r$.

Respecting Eq. 2 and Eq. 3, each new cluster C' is defined by set of points that belong to the same topological entity (since the path from two points belonging to two different topological entities is greater than the defined tolerance). In other words, points belonging to original cluster C that are topologically "unconnected" with the set of points comprising the "seed point" $b_{h,k}$, are discarded in evaluating the cluster centroid. Accordingly the new cluster centroid $\boldsymbol{\chi'_{pi}}$ (see Fig. 10) is a vector whose elements are the weighted mean values of all the points $c'_{m,n}$ belonging to C':

$$\chi'_{pi} = [\overline{m}, \overline{n}] \tag{4}$$

Where:

$$\overline{m} = \frac{\sum_{c \in C'} m \cdot w_{m,n}}{\sum_{c \in C'} w_{m,n}} \quad , \quad \overline{n} = \frac{\sum_{c \in C'} n \cdot w_{m,n}}{\sum_{c \in C'} w_{m,n}} \tag{5}$$

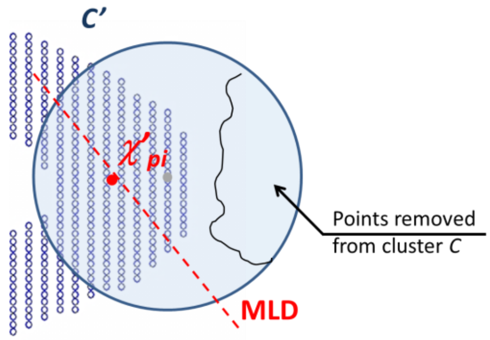
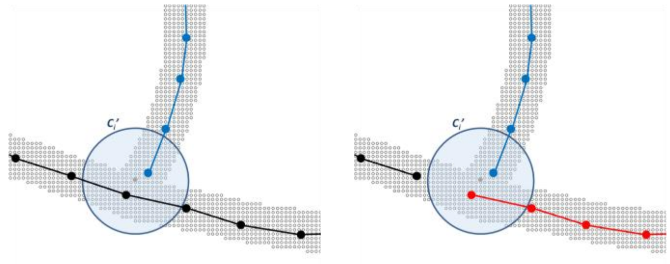


Fig. 10 – Cluster C'. By discarding points belonging to the second topological object the MLD is correctly evaluated.

Analogously to what described in [31], starting from the new first cluster centered in a "seed point" randomly extracted from the point cloud, the devised procedure carries on by determining the MLD vector of the cluster and by defining two new centroids χ'_{f1} and χ'_{p1}. Starting from χ'_{f1}, the procedure is iteratively carried out until, at the ith iteration, one of the following conditions occurs:

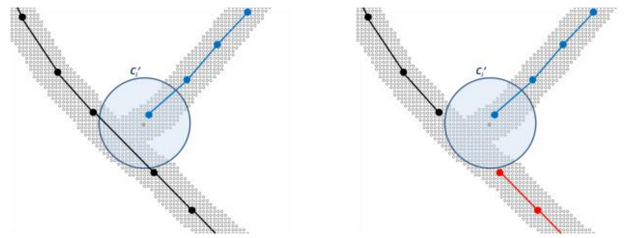
1. the cluster at the ith iteration C'_i contains no point that are not contained in C'_{i-1} (i.e. an endpoint of the point cloud has been reached); in this case the iterative process has to be continued starting from χ'_{p1}.
2. the C'_i region crosses one or more polylines different from the one in process (i.e. point cloud junction has been reached); in this case two additional conditions have to be considered:
   - 2.1. the C'_i region contains one or more vertexes belonging to one of the intercepted polylines; the vertex whose distance from the cluster center is the minimum allowable defines the interruption point of the intercepted polyline (Fig. 11a). Such a polyline is then divided into two new polylines by removing the shortest segment linked to the interruption point (Fig. 11b). As a consequence, points defining the removed segment are discarded when the spline is built;
   - 2.2. the C'_i region contains one or more segments belonging to one of the intercepted polylines (i.e. any vertex of such polylines is enclosed in the cluster but the polyline is crossed by cluster boundaries) as depicted in Fig. 12a. The intercepted polyline is then divided into two new polylines by removing the intercepted segment (Fig. 12b). As a consequence, points defining the removed segment are discarded when the spline is built;

If the point cloud is composed by a single topological entity (i.e. no junctions or vertexes exist), the above conditions assure to trace the only possible polyline since all the points (in the point cloud) have been processed. Vice versa, when more than one topological entity exists, once traced the first polyline, a new seed has to be randomly extracted from the remaining points and a new polyline is evaluated. The procedure goes on until all the points are processed.



(a) (b)

Fig. 11 – Polyline vertex interception.



(a) (b)

Fig. 12 – Polyline edge interception.

Referring to the case of Fig. 6, no jump occurs and the topology of the point cloud is preserved (Fig. 13).

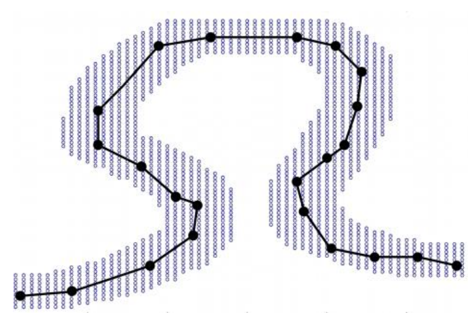


Fig. 13 – Topologically correct polyline

IV. Gray-level weighted B-spline curves fitting

The final result of the previous task is to obtain a single polyline for each topological entity describing the point cloud.

For each polyline, a B-Spline called "ordering curve" (OC), interpolating only the polyline points, is traced (Fig. 14).

The purpose of such B-Splines consists of sorting the point cloud points, according to the order of their projection on the OC.

The point projection problem is to find the closest point on the curve to a given point. In this work this problem has been solved by a numerical, approximated approach as follows:

1. given a cluster C_u selected among the ones defined in section III, the clusters C_{u-1} and C_{u+1} are considered along with their centroids χ_{u-1} and χ_{u+1}.
2. the OC, which by definition interpolates all the centroids χ_i, is uniformly sampled in the interval delimited by χ_{u-1} and χ_{u+1}. The number of OC samples is set equal to three times the global numerosity of the above considered clusters.

3. the projection of each point p belonging to C_u on the OC is the point $p_{s'}$ that is its nearest among the OC sample points p_s:

$$p_{s'} : \|p - p_{s'}\| = \min\{\|p - p_s\|\} \tag{6}$$

4. the procedure carries on iteratively for each cluster. Obviously, the points belonging to previously processed clusters are not further considered.

The final result of this procedure is a set of implicit ordered point clouds (one for each polyline).

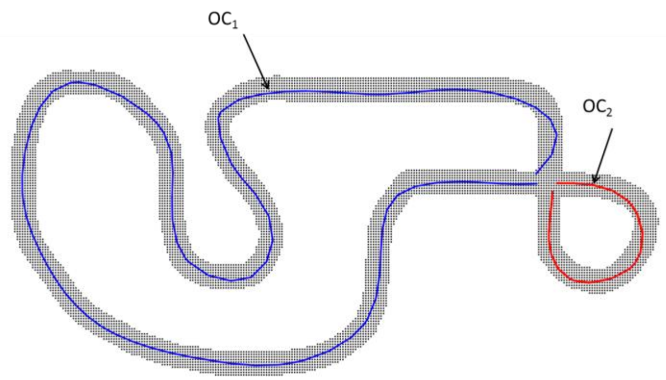


Fig. 14 – Ordering curves

In order to obtain a set of explicit ordered point clouds, a normalized curvilinear coordinate s for each of the OCs has to be defined. Using one of the OC endpoints as the coordinate origin, it is possible to assign a value s for each point projection.

Given an ordered set of points (one for each OC), it is straightforward to fit an approximating B-Spline curve called Fitting Curve (FC). The B-Spline which is used to fit the original points is built so that:

- its parameter t varies between 1 and a with unitary step, where a is the total number of the points belonging to the original point cloud;
- for each integer value t_i of t, the spline point of parameter t_i is biunivocally correspondent to the i^{th} point of the ordered cloud. Thanks to this correspondence, it is immediate to evaluate the Maximum Fitting Error (MFE) as:

$$MFE = \max_{t=1 \to a}\{W(p_{t,1}, p_{t,2}) \cdot \|FC(t) - p_t\|\} \tag{7}$$

The fitting task starts with the generation of a two-knot first approximation FC; the initial knots, which have a multiplicity of 4, are coincident with the two OC endpoints.

This FC, along with the ones subsequently computed, is generated according to Weighted Least Squares (WLS) technique [34] where the weighting coefficient for each point of coordinates (m,n) is $w_{m,n}$ (see section II).

By means of an iterative procedure, FC is updated by adding new knots with the aim of obtaining the necessary degree of approximation; in order to guarantee the desired accuracy, a user defined tolerance value (*tol*) is required. The final result of the procedure is a set of FCs, shown in Fig. 15.

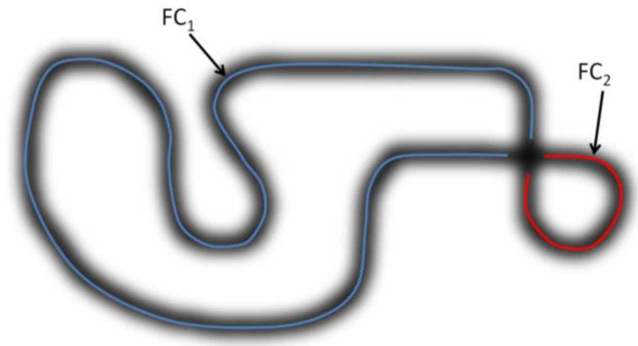


Fig. 15 – Fitting curves

V. Multiple Incident Splines (MISs) algorithm

Once the set of FCs is determined, a junction procedure needs to be used in order to join the respective endpoints which are "sufficiently close" one to the other. A preliminary selection of the endpoints to be joined is also necessary.

The selection-joining procedure can be described as follows:

1. the set E of all the endpoints is defined;
2. until set E is not empty, a random endpoint e_i belonging to E is selected;
   a. a threshold value t is defined so that t=5r, where r is the mean point cloud thickness;
   b. additional endpoints with distance from e_i lower or equal to t are sought, in other words points belonging to $I_{e_i} = \{x \in \Re^2 \mid d(x, e_i) \le t\} = I(e_i, t)$; in case new endpoints (for instance e_j, e_k) are found:
      i. a "union set" US is defined so that US={e_i, e_j, e_k};
      ii. for the newly found endpoints e_j and e_k a new search is performed in I_{e_j} and I_{e_k} for possible endpoints not yet belonging to US, in case new endpoints are found, they are added to US; this step is repeated until none of the endpoints belonging to US contains in its t radius neighborhood new elements of E which do not already belong to US;
      iii. All the endpoints belonging to US are joined according to the procedure described later;
      iv. the endpoints in US are deleted from set E and the procedure is repeated from step 2;
   c. if no endpoints are found in I_{e_i}, e_i is deleted from set E and the procedure is repeated from step 2.

In order to devise an effective joining procedure it is necessary to emphasize that, generally speaking, FCs geometry in the neighborhood of endpoints is often perturbed (Fig. 16). This is due to the presence of points falling in the clusters of a

FC branch but which, in fact, belong to different branches. As a consequence, the joining procedure has been devised so as not to be "confused" by such perturbed regions.

Endpoint joining is performed as follows (see Figs. 16 and 17):

1. the barycentre B_1 of the group of endpoints to be joined (points belonging to US) is computed;
2. the endpoint e_m belonging to US with maximum distance (d_m) from B_1 is identified;
3. $I(B_1, r_1)$ with $r_1=\alpha \cdot d_m$ is defined (α=1.2 was selected, but different values can be assumed according to point cloud characteristics);
4. The barycentre B_2 of the point cloud points inside $I(B_1, r_1)$ is computed;
5. $I(B_2, r_2)$ with d_s is the maximum distance of endpoints in US from B_2 and $r_2=\beta \cdot d_s$ (β =1.2 was selected, but different values can be assumed according to point cloud characteristics);
6. the knots inside $I(B_2, r_2)$ are deleted from the knot sequence of the FCs whose endpoints belong to US; note that, according to this step, the deleted knots are: the first one (or the last one) in the sequence and possible additional knots adjacent to it.
7. each FC identified in step 6 (e.g. FC_i), is extended by using an additional spline segment so that it starts in correspondence of the first undeleted knot of FC_i, is tangent to FC_i in such a knot and ends on B_2.

The FCs resulting from this procedure is the final set of Multiple Incident Splines (MISs) fitting the original point cloud.

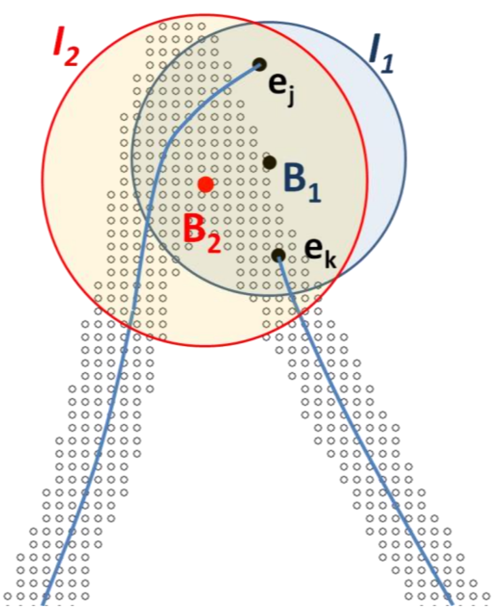


Fig. 16 – FCs endpoints detection

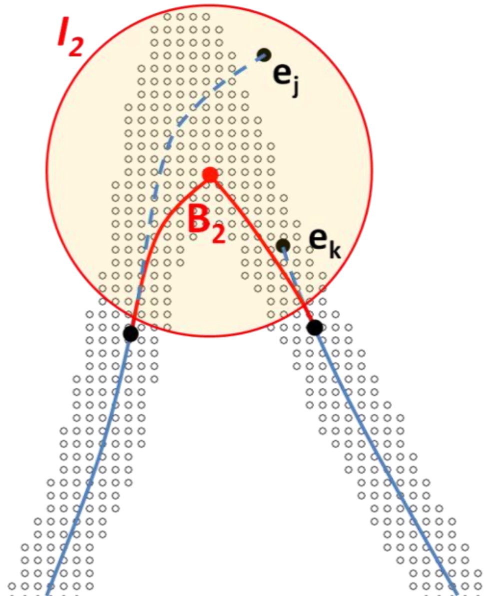


Fig. 17 – FCs endpoints junction procedure

VI. RESULTS

In Figure 18 the result of the proposed method on a case study image is compared against two alternative B-spline fitting techniques provided by two different commercial vectorization software packages, hereafter named SwA and SwB.

Both SwA and SwB make use of a preprocessing phase in which an automatic thresholding is performed. The result of this phase is to provide a binary image loosing the capability of weighting single pixel on the basis of its brightness. Successively a thinning process is performed and a B-spline (or a set of B-splines) is, then, fitted to the thinned image according to a least square method.

As a consequence the algorithms used by SwA and SwB often bring in a set of pathological cases typical of thinning processes (e.g. spurious segments, topological errors, etc.).

The fitting curve (MISs) obtained by means of the solution proposed in this paper proves to be the most effective in preserving the original shape. As shown in case B of Fig. 18, SwB fails in vectorizing intersections and a topological error occurs since two vertexes are built instead of the actual one.

Moreover, the proposed MISs algorithm is particularly suitable for fitting free-form curves. In fact, it is more reliable than the others in detecting the right path when curvature variations happen without false vertex introduction (as shown in case A of Fig. 18). As a consequence the proposed method, dealing with a fundamental task for translating 2D sketches into 3D CAD models, proves to be mainly effective in 3D reconstruction of freehand conceptual sketches (stylistic sketches).

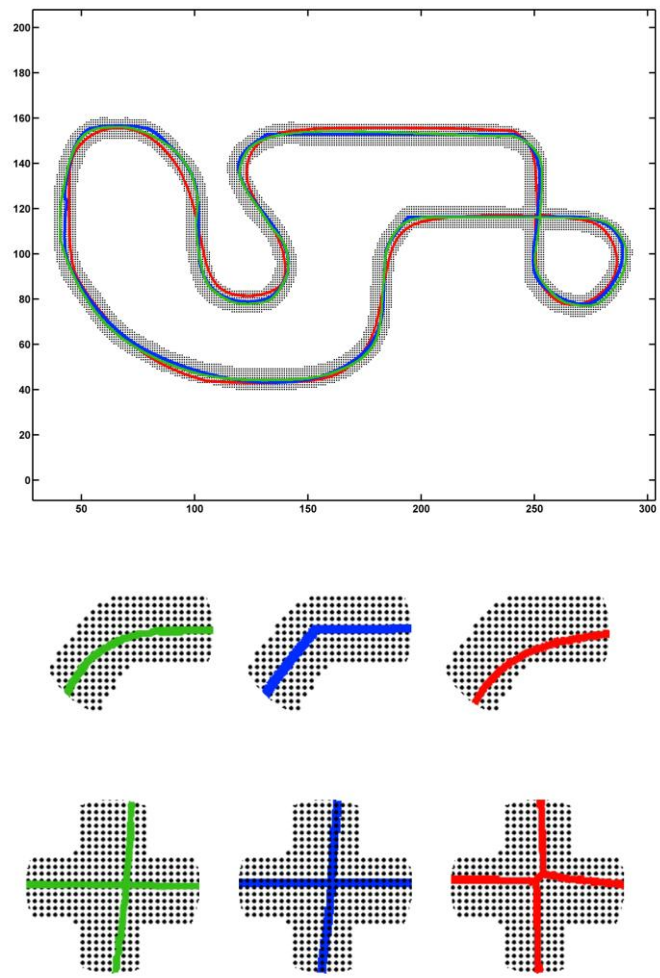


Fig. 18 – Comparison between the proposed method (MISs, green curves) and two different commercial software packages (SwA, blue curves and SwB, red curves).

CONCLUSIONS

In the present work an approach to fit 2D unordered point cloud data with Multiple Incident Splines (MISs), whose weights are computed on the basis of raster data gray levels, is presented.

The provided method has been developed by means of a series of Matlab® routines. In particular, in order to obtain ordered point clouds starting from unordered ones, a novel approach, based on Euclidean Minimum Spanning Tree (EMST) and Principal Component Analysis (PCA), has been devised and tested on a number of case studies. Moreover a novel robust method for fitting ordered point clouds with spline curves has been provided and validated. Such a method is particularly suitable for fitting free-form curves; in fact, it is capable of a more reliable intersection and vertex detection with respect to different approaches also implemented in commercial vectorization software packages.

Future work will be addressed to a 3D generalization of the proposed method in order to reconstruct 3D models starting from 2D freehand sketches provided by stylists.

REFERENCES

[1] L.B. Kara, K. Shimada, “Supporting early styling design of automobiles using sketch-based 3D shape construction”. *Computer-Aided Design and Applications* vol. 5, Issue 6, 2008, pp. 867-876.

[2] L.B. Kara, C.M. D'Eramo, K. Shimada, “Pen-based styling design of 3D geometry using concept sketches and template models”. *Proceedings SPM 2006 - ACM Symposium on Solid and Physical Modeling*, 2006, pp. 149-160.

[3] P. Company, A. Piquer, M. Contero and F. Mnaya, “A survey on geometrical reconstruction as a core technology to sketch-based modeling”. *Computers & Graphics* vol. 29, Issue 6, 2005, pp. 892– 904.

[4] M. Carfagni, R. Furferi, L. Governi, M. Palai and Y. Volpe, “3D Reconstruction Problem": An Automated Procedure”, APPLICATIONS of MATHEMATICS and COMPUTER ENGINEERING, 2011, pp. 99-104.

[5] M. A. Fahiem, A. Shaiq, M. Haq and M. R Sabir “Comparison of 3D Reconstruction Techniques for Engineering Drawings from Orthographic Projections”, *Proceedings of the 6th WSEAS International Conference on Applications of Electrical Engineering*, Istanbul, Turkey, May 27-29, 2007.

[6] Z. Wang and M. Latif, “Reconstruction of 3D Solid Models Using Fuzzy Logic Recognition”, WSEAS Transaction on Circuits and systems, Vol. 3, 2004, pp. 1018-1025.

[7] M. A. Fahiem and N. Kanval, “A novel CSG Approach for 3D Reconstruction of Helix Using Spiral Sweeps”, 7th WSEAS International Conference on APPLIED COMPUTER SCIENCE, Venice, Italy, November 21-23, 2007.

[8] M. A., Fahiem, S.A. Haq and F. Saleemi, “A Review of 3D Reconstruction Techniques from 2D Orthographic Line Drawings”. *Geometric Modeling and Imaging* (GMAI ’07), July 2007, pp. 60–66.

[9] Shixia, L., Shimin, H., and Jiaguang, S., 2002. “Two accelerating techniques for 3D reconstruction”. Journal of Computer Science and Technology, 17(3).

[10] Inoue, K., Shimada, K., and Chilaka, K., 2003. “Solid Model Reconstruction of Wireframe CAD Models Based on Topological Embeddings of Planar Graphs”. Journal of Mechanical Design, 125(3), September, pp. 434–442.

[11] M. Wesley and G. Markowsky, “Fleshing out wire frames”. *IBM Journal of Research and Development*, Vol. 24, Issue 5, 1980, pp. 582–597.

[12] M. Wesley and G. Markowsky, “Fleshing out projections”. *IBM Journal of Research and Development*, Vol. 25, Issue 6, 1981, pp. 934–954.

[13] R. Furferi, L. Governi, M. Palai and Y. Volpe, “From 2D Orthographic views to 3D Pseudo-wireframe: An Automatic Procedure”. *International Journal of Computer Applications* IJCA Vol. 5 Issue 6, 2010, pp. 12–17.

[14] N. Amenta, M. Bern, and D. Eppstein. The crust and the beta-skeleton: combinatorial curve reconstruction. Graphical Models and Image Processing, 60:125–135, 1998.

[15] A. Atieg and G. A. Watson. A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances. Journal of Computational and Applied Mathematics, 158:227–296, 2003.

[16] A. Bjorck. Numerical Methods for Least Squares Problems. Mathematics Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[17] T. K. Dey, K. Mehlhorn, and E. Ramos. Curve reconstruction: connecting dots with good reason. Comput. Geom. Theiry & Appl., 15:229–244, 2000.

[18] H. Pottmann and T. Randrup. Rotational and helical surface approximation for reverse engineering. Computing, 60(4):307–322, 1998.

[19] H. Yan. Fuzzy curve-tracing algorithm. IEEE transactions on Systems, Man,and Cybernetics–Part B: Cybernetics, 31(5):768–780, 2001.

[20] Y. Liu, H. Yang, W. Wang, Reconstructing B-spline curves from point clouds—a tangential flow approach using least squares minimization, Proceedings of SMI, 2005, pp. 4–12.

[21] D. Levin. The approximation power of moving least-squares. Mathematics of Computation, 67(224):1517–1531, 1998.

[22] I.-K. Lee. Curve reconstruction from unorganized points. Computer Aided Geometric Design, 17:161–177, 2000.

[23] L. Wenyin, D. Dori, "From raster to vectors: Extracting visual information from line drawings" *Pattern Analysis and Applications* vol. 2, Issue 1, 1999, pp. 10-21.

[24] D. Dori, "Orthogonal Zig-Zag: an Algorithm for Vectorizing Engineering Drawings Compared with Hough Transform". *Advances in Engineering Software* vol.28, Issue 1, 1997, pp. 11–24.

[25] D. Dori, Y. Liang, J. Dowell, I. Chai. "Spare pixel recognition of primitives in engineering drawings". *Machine Vision and Applications* vol. 6, 1993, pp. 79–82.

[26] W. Liu, D. Dori, "Automated CAD Conversion with the machine drawing understanding system". *Proceedings 2nd IAPR Workshop on Document Analysis Systems*, Malvern, PA, 1996, pp. 241–259.

[27] J.Y. Yoo, M.K. Kim ,Y.B. Kwon. "Information extraction from a skewed form document in the presence of crossing characters". In: K. Tombre, A. Chhabra (eds). *Graphics Recognition – Algorithms and Systems (Lecture Notes in Computer Science, 1389)*. Springer-Verlag, 1998, pp. 139–148.

[28] P. Vaxiviere, K. Tombre, "Cellestin: CAD conversion of mechanical drawings". *IEEE Computer* vol. 25, Issue 5, 1992, pp. 46–54.

[29] S. Süsstrunk, R. Buckley, and S. Swen, "Standard RGB color spaces", Proc. IS T/SID 7th Color Imaging Conf., 1999, pp. 127 - 134.

[30] Shapiro, L.G., Stockman, G.C. "Computer Vision". Prentice Hall, 2002.

[31] R. Furferi, L. Governi, M. Palai and Y. Volpe, "From Unordered Point Cloud to Weighted B-Spline - A Novel Pca-Based Method": An Automated Procedure", APPLICATIONS of MATHEMATICS and COMPUTER ENGINEERING, 2011, pp. 146-151.

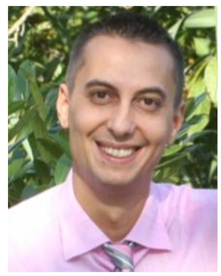
[32] J. Daniels II, T. Ochotta, L.K. Ha, C.T. Silva, "Spline-based feature curves from point-sampled geometry". *Visual Computer* vol. 24, Issue 6, 2008, pp. 449-462.

[33] R. Diestel, *Graph Theory*, 2006. Springer.

[34] A. Nealen, "An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation,", May 2004, URL: http://www.nealen.com/projects/.

Furferi R., PhD in Machine design and Construction (2005) - University of Florence, Italy. Graduated MSc in Mechanical Engineering - University of Florence. After working as a post-doctoral researcher at the Department of Mechanics and Industrial Technologies of the University of Florence, in 2008 he assumed a Faculty position as Assistant Professor for the course "Mechanical Drafting". His main Scientific interests are: development of artificial vision systems for industrial and textile control, artificial neural networks, colorimetry, reverse engineering and rapid prototyping. He is author and co-author of many publications printed in journals and presented on international conferences. Some latest publications described methods for color assessment of textiles, algorithms for 3D reconstruction of objects from orthographic views and ANN-based systems for industrial process. Dr. Furferi is Technical Editor for some journals: Journal of Applied Sciences, Journal of Artificial Intelligence and International Journal of Manufacturing Systems.

Governi L. PhD in Machine design and Construction (2002) - University of Florence, Italy. Graduated M.Sc in Mechanical Engineering - University of Florence. After working as a post-doctoral researcher at the Department of Mechanics and Industrial Technologies of the University of Florence, in 2005 he assumed a Faculty position as Assistant Professor for the courses of "Reverse Engineering" and "Design and modeling methods". His main scientific interests are: machine vision and reverse engineering, colorimetry, tools and methods for product design and development. He is author and co-author of many publications printed in international journals and participated to a number of international conferences. Some latest publications described techniques oriented towards the 3D reconstruction from orthographic views, vision-based product and process assessment and spline-based approximation of point clouds.



Volpe Y. PhD in Machine design and Construction (2006) - University of Florence, Italy. Graduated M.Sc in Mechanical Engineering - University of Florence. He is currently working as a post-doctoral researcher at the Department of Mechanics and Industrial Technologies - University of Florence. He is also Adjunct Professor of the course "Computational Graphics" from the Engineering Faculty of the University of Florence. His main scientific interests are: Computer Aided Design, Image Processing, Virtual Prototyping, FE simulation, Reverse Engineering and Rapid Prototyping. He is author and co-author of many publications printed in international journals and participated to a number of international conferences. Some latest publications described techniques for comfort-oriented design, machine vision-based systems for industrial processes and spline-based approximation of point clouds.

Palai M. PhD candidate in Project and Development of Industrial Products and Processes - University of Florence, Italy. Graduated M.Sc. in Mechanical Engineering - University of Florence. His Doctorate Research deals with 2D to 3D reconstruction, Reverse Engineering and Rapid Prototyping. He is co-author of a publication printed in an international journal and participated to an important international conference where he presented a technique for spline-based approximation of point clouds, a 2d to 3D reconstruction method and a machine vision system for fabric inspection.