



Cost estimation of custom hoses from STL files and CAD drawings

Joan Serrat*, Felipe Lumbreras, Antonio M. López

Computer Vision Center and Computer Science Dept., Universitat Autònoma de Barcelona, 08193 Cerdanyola, Spain

ARTICLE INFO

Article history:

Received 27 December 2011
 Received in revised form 21 September 2012
 Accepted 28 November 2012
 Available online 17 January 2013

Keywords:

On-line quotation
 STL format
 3D reconstruction
 Regression
 Gaussian process

ABSTRACT

We present a method for the cost estimation of custom hoses from CAD models. They can come in two formats, which are easy to generate: a STL file or the image of a CAD drawing showing several orthogonal projections. The challenges in either cases are, first, to obtain from them a high level 3D description of the shape, and second, to learn a regression function for the prediction of the manufacturing time, based on geometric features of the reconstructed shape. The chosen description is the 3D line along the medial axis of the tube and the diameter of the circular sections along it. In order to extract it from STL files, we have adapted RANSAC, a robust parametric fitting algorithm. As for CAD drawing images, we propose a new technique for 3D reconstruction from data entered on any number of orthogonal projections. The regression function is a Gaussian process, which does not constrain the function to adopt any specific form and is governed by just two parameters. We assess the accuracy of the manufacturing time estimation by k -fold cross validation on 171 STL file models for which the time is provided by an expert. The results show the feasibility of the method, whereby the relative error for 80% of the testing samples is below 15%.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Price estimation of parts from digital models like CAD files or blueprints before their actual manufacture is a key task for many companies. In today's global competitive market, timely delivering accurate quotations to potential clients may make the difference of a company with respect to its competitors. This is even more necessary for small and medium-size enterprises for which a large share of their production consists in short series of custom parts. This means that price has to be estimated accurately, fast and frequently from part models provided by potential customers, being parts always different in size, shape and materials.

The quotation process includes as its main component the estimation of the unitary manufacturing cost, to which the profit margin is added in order to get the final price. Cost estimation in those companies is often performed by an expert on the basis of his/her experience over the years. This has some drawbacks: companies come to strongly depend on such qualified personnel, experts spend a large amount of time generating quotations rather than working on other important tasks [1], and their estimations may have a subjective component. Finally, in the event of peaks in the demand of bids, they act as involuntary bottlenecks of the production process because customers can not directly obtain their quotations by other means.

1.1. Objective

This article deals with the problem of cost estimation for a specific type of parts, namely custom hoses. They are used to transport liquids and gases often at high temperature and pressure, in a variety of sectors like automotive, nautical, chemical and medical industries. Their changing shape and production in short series explain why they are manually manufactured: a steel forming tool is first made that is used for molding the hose into shape by hand, wrapping a narrow band of some kind of textile material like rubber or silicone, to be later melted in an oven.

The cost depends on the amount of material employed and also on the manufacturing time. One may wonder whether the labor time cost is significant with respect to the materials cost. According to consultations we made to a leading company of this sector, in developed countries it may account for more than 60% of the total unitary cost, pushing this kind of industries into offshoring. While the materials cost can be readily computed from the part surface, labor time cost is much more difficult to determine because it is related in an unknown way to the part size and shape. Specifically, we only know from the expert that the larger the part, the longer it takes to manufacture. Likewise, the more bends and the higher their curvature, the longer it takes because it is harder to uniformly wrap the covering material into the mold.

Thus, the objective is to derive a procedure for the automatic estimation of the surface and the manufacturing time from a CAD model of a hose. The models may come in two formats, the easiest

* Corresponding author. Tel.: +34 3 935811828; fax: +34 3 935811670.
 E-mail address: joans@cvc.uab.es (J. Serrat).

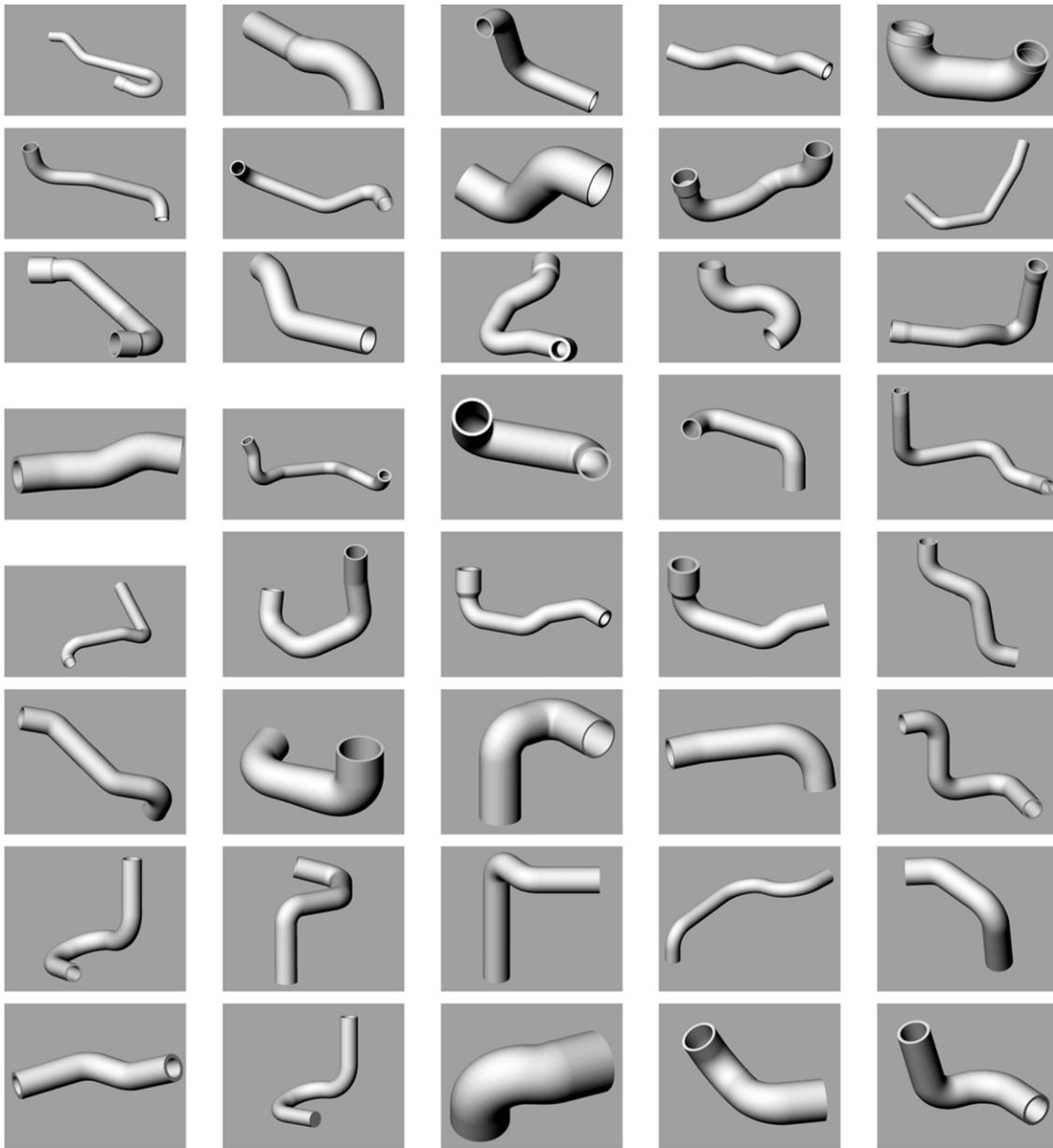


Fig. 1. 3D rendering of some STL models. Models have been rescaled for better visualization. The size of the working set of models ranges from 13 cm to 1.2 m long and 8 mm to 20 cm of inner diameter. Note that section diameter may change along one same model, like in top-right and center.

to generate and send electronically by a customer: stereolithography files (STL) or an image of the CAD drawing showing orthogonal projections. STL is a file format to describe the surface geometry of an object and it is supported by most CAD software packages. CAD designs can be printed and scanned, or simply saved in some image format like TIFF, JPEG, etc. Fig. 1 shows a gallery of tubes rendered from their STL models and Fig. 6a shows an image of a scanned blueprint.

Furthermore, this procedure must be implemented as a platform independent web application so that manufacturing companies representatives and customers alike may get quotations on-line from their web browsers without the hassle of software installation, version updating and hardware dependency problems.

As a final and minor goal, the system must be able to render realistic orthogonal views of the model like that of Fig. 6b. Generating them from the STL file is fairly easy. However, it is not from CAD drawing images because it requires the reconstruction of

the 3D model from orthogonal views. Rendered model images are to be included in the quotation document, to show the customers that the part shape has been understood well and support the quotation result.

1.2. Related work

In order to deal with the issues derived of cost estimation from part models by an expert, several researchers have addressed its automation, notably in the domains of rapid prototyping and tooling, and machining manufacturing. Refs. [2,1] are nice and through reviews of the works on these two fields, respectively. For this reason, and because we address a different type of production, we will only pay attention to the two aspects most related to our work: the classification of cost estimation methods and web-based quotation systems.

Automated cost estimation methods have been grouped into the three following categories [3,1]:

- *Analogy*. The values of part attributes related with, for example, shape, size and material characteristics, have been previously stored in a database along with their actual cost. When a new part description arrives, the most similar part in the database is retrieved and its registered cost is somehow adapted to produce the estimation for the new part. The success of this approach relies on the proper selection of the part attributes, the similarity measure, the database size and a balanced distribution of shapes within it. Two representative examples are [4,5].
- *Analytic*. The part and/or the manufacturing process is divided into components or simpler subtasks so that for each one it is possible to calculate the cost deterministically. Later, all costs are aggregated.
- *Parametric*. Like in analogy, a set of part attributes is previously selected because they are considered to be highly related to the cost. For instance, volume, area, surface curvature or type of finishing. But instead of trying to retrieve the most similar part in a database, they are used to directly estimate the cost through regression analysis. This kind of methods draw from statistical regression and machine learning techniques [17].

With regard the adoption of the analogy approach for our problem, we have found impossible to define a good similarity measure, given the variety of shapes and sizes. Another reason is that fairly *globally* similar shapes may have associated quite different times because it seems to also depend on *local* shape characteristics like surface curvature. A third obstacle is that it is not clear at all how to adapt the time of the most similar part to the present object. Analytic methods are better suited to manufacturing processes which can be decomposed in distinct, cost-quantifiable tasks, which is not the case. The parametric approach, instead, has shown to fit well to our problem: we have been able to select both global and local features and learn a regression function from a training set of samples which accurately estimates the manufacturing time.

There are web applications for cost estimation from STL models. One is QuickParts, which can be found in [15] although no description on how it works is provided, beyond that it performs some kind of geometric analysis. Conversely, Lan et al. [16] detail a web-based automated quotation system which can provide instant price quotations. However, both works are specific of rapid prototyping and machining parts.

1.3. Contribution and system overview

In the following sections we will describe the most important components of the cost estimation system we have built for customized hoses. Two of them have in common the goal of extracting one same 3D high-level and complete description of the part to quote: the 3D curve of the tube medial axis and the diameter of section at each point along it. They will allow to compute the surface – and thus the materials cost –, and also the value for attributes to be subsequently used for time regression analysis, namely, medial axis length, curvature and diameter mode. The first component obtains the medial axis from an STL file (Section 2), which contains an unsorted list of triangles covering the part surface, specified by their vertices and normal in 3D space. Therefore, it is far from trivial to get the central axis curve. The second component does the same but from an image of the CAD drawing showing three orthogonal views. We have devised a semi-automated algorithm for the 3D reconstruction of the model which works with one or more orthogonal projections in a CAD drawing image (Section 3).

The third main component is the time regression. We are given a large number of samples consisting of STL files plus their time as estimated by a human expert. From this training set, we extract the

value of a few relevant features related to the part size and shape in order to learn a Gaussian process. We can then estimate the cost from the features for new parts (Section 4). In Section 5 we assess the accuracy of the cost regression by means of a well known testing methodology in machine learning, *k*-fold cross validation, and justify the choice of the only two Gaussian process hyperparameters, the scale and the data noise variance.

We apply these techniques to the specific problem of hose quotation. However, we believe they are useful in a broader sense. The algorithm for medial axis extraction is applicable to any tubular shape described by points on its circular sections, and the same can be said of the 3D reconstruction. Regression through a Gaussian process from geometric features can be applied to other shapes for cost estimation also, avoiding the need to study each subtask like in the analytic approach. It has the advantages of being an almost-free parameter regression method, not assuming any specific class for the regression function, and its few parameters (two, in the simplest setting) can be easily tuned to the available data.

2. Medial axis extraction from STL models

The STL file format describes the approximate shape of an object in a very simple way: an unsorted list of triangles whose mesh covers the surface, each specified by its three vertices and unitary normal vector. We aim at obtaining a high-level shape description, namely, the medial axis defined as an ordered list of circular section centers and their corresponding diameters (Fig. 2), from one of the tube ends to the other.

The medial axis line, also called curve-skeleton in the literature, is a concise representation of 3D shapes that has attracted much attention because of its applications to shape compression, retrieval, visualization and animation. We refer the reader to the excellent survey [6] which reviews and categorizes many methods according to several properties. Relevant here is the class of geometric methods, those accepting as input polygonal meshes. They work directly on the mesh domain without having to sample the model in order to produce a (maybe huge) binary volumetric or voxel-based representation, like in [7]. Among them, a successful approach consists on making the vertices or patches to iteratively evolve or collapse from the shape boundary into the shape centerline [8–10]. They are effective but complex methods because they intend to work with generic closed shapes. In our case, however, we can take advantage of a strong constraint: our shapes are tubes, that is, a succession of circular sections orthogonal to the medial axis possibly with varying radius. This has allowed us to come up with a specialized but much simpler algorithm.

We decompose the problem in two parts: finding circular sections and then sorting their centers to form a smooth polyline curve (a list of straight segments) in space. For the first, we have adapted RANSAC [18,19], a generic robust hypothesis generation and verification method used to fit parametric shapes to data which may contain a large proportion of outliers. For the sake of completeness, we will summarize it. RANSAC is composed of two steps:

1. Hypothesize: randomly select a minimum set of data samples so as to fit the parametric model to them (i.e., find the parameters values). In our case, three vertices suffice to define a circle in 3D space.
2. Verification: select the data points close enough to the instantiated model, which will form its consensus or support set.

They are repeated iteratively until the size of the support set rises above a certain threshold. Then, the consensus set is

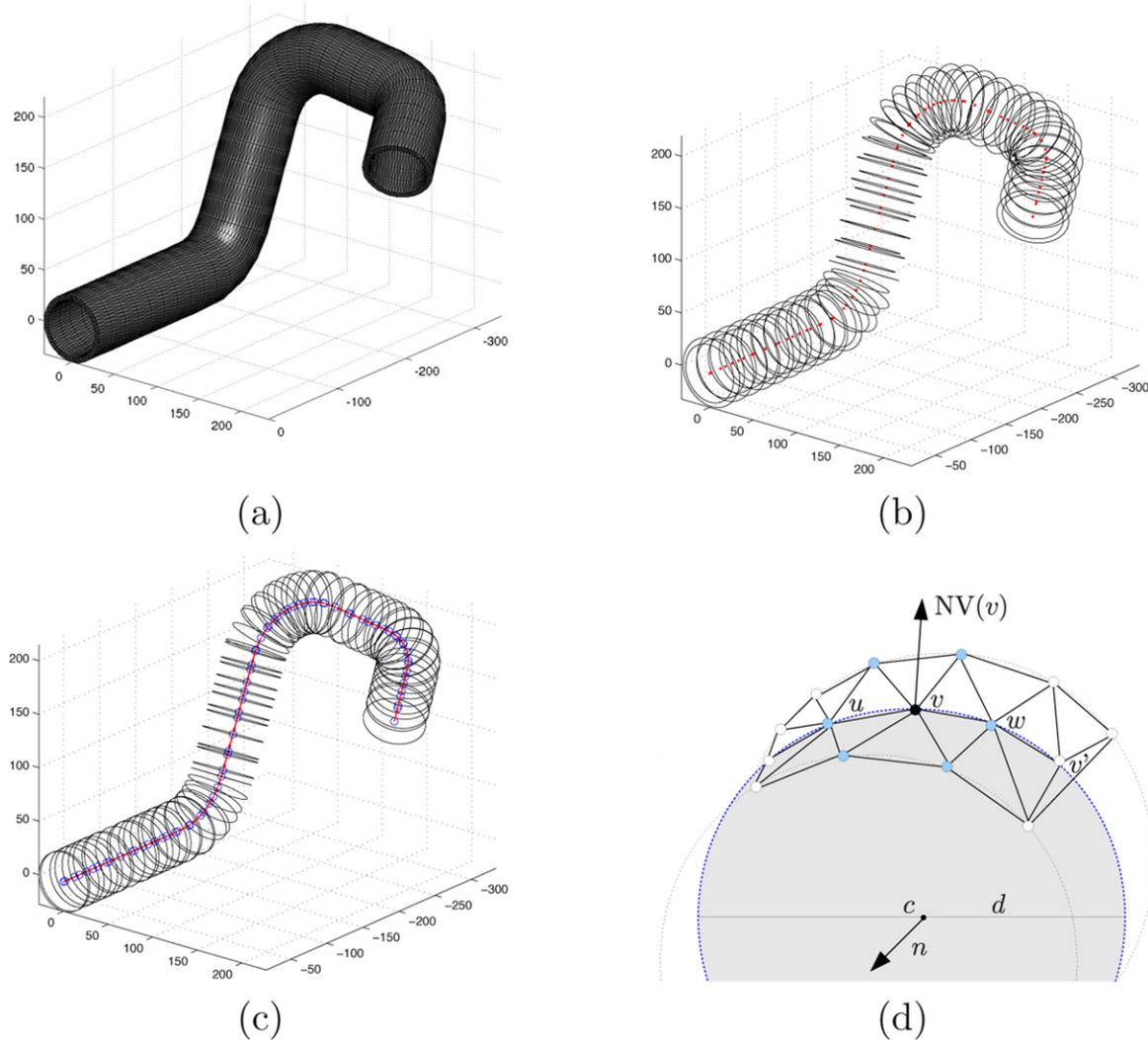


Fig. 2. Medial axis extraction. (a) STL input model, (b) extracted inner and outer circular sections, (c) medial axis and (d) variables of Algorithm 1.

considered the inliers set and the model is fitted again to all of them.

Our data are not samples of a single parametric model but of a large number of 3D circles, and all or at least many of them, closely spaced, have to be found in order to build the medial axis from their centers. Trying all triplets of vertices for every circle is clearly infeasible because even mid-size models have thousands of vertices. Moreover, CAD models represent the hose thickness and thus we have an inner and an outer tube surfaces, both covered by the triangle mesh (Fig. 2a). It is absolutely necessary to prune the number of triplets to try. We achieve it in two ways (Algorithm 1). The first is, for each vertex v , to select the two remaining vertices u, w among the small set of vertices which are connected with it in the mesh (in light grey in Fig. 2d). This boosts the probability that each triplet forms an actual circular section. The second is to remove from the set of vertices to choose from, those that have already been found to belong to the support set of some circle. These two simple strategies allow to find the circular sections of meshes with tens of thousands of vertices in a matter of seconds. Three additional constraints help to further filter out infeasible triplets satisfying the two former conditions. They are based on: (1) a minimum distance between the points in the triplet, (2) the parallelism of their normals, and (3) the orthogonality between the normal to the plane containing the fitted circle, and each vertex normal (see Algorithm 2 and Fig. 2d).

The second part, sorting the sections, is solved as follows. Consider the weighted undirected graph whose nodes are the section centers, there is one edge between every pair of nodes (i.e., the graph is fully connected) and the edge weight is the distance between the two nodes it connects. We would like to remove all edges but those such that each section center remains linked to the closest one in space, thus forming a chain of segments. It turns out that the minimum spanning tree (MST) of such a graph does provide this result. The MST of a graph is a connected subgraph with no cycles that includes all the nodes and at the same time minimizes the sum of weights of its edges. There are several algorithms for the computation of the MST, like Kruskal's or Prim's, which have a low complexity, on the order of $E \log E$ and $E \log N$, respectively, where E and N are the number of edges and nodes [20].

Note that as a consequence of the double hose surface, the circles diameter found in the first part can be either the inner or outer section diameter (Fig. 2b). Once sorted the sections, a simple local minimum filter on the list of diameters gets the inner diameter for each section (Fig. 2c).

Algorithm 1. Medial axis from STL (part 1). Some constants below, like $maxDist$ and $minSupport$, depend on the density of vertices specified in a CAD software when saving a model in STL format.

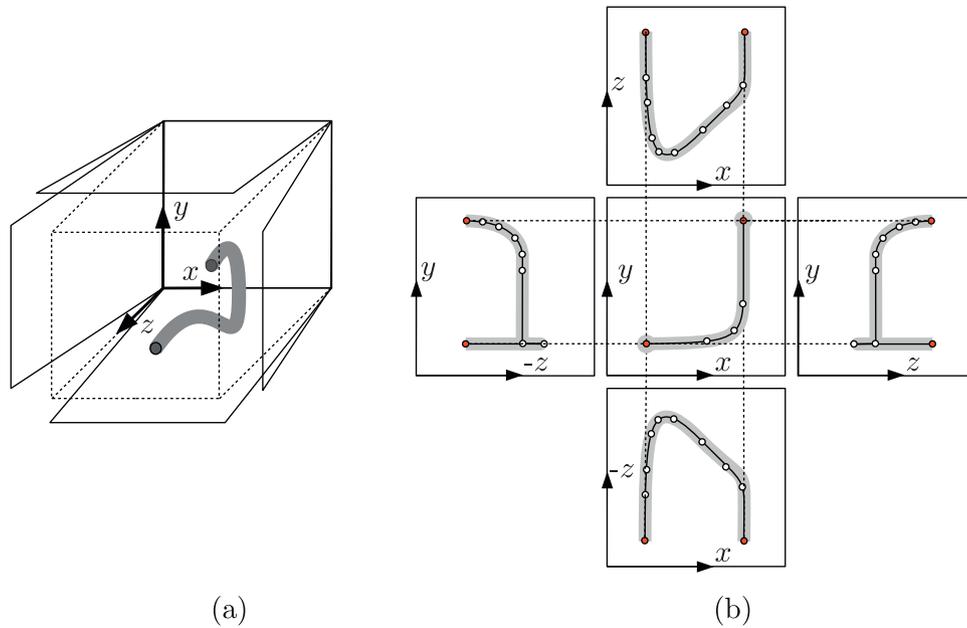


Fig. 3. Spatial arrangement of projections in a CAD drawing with xy central projection in the middle and matching of end points.

```

function MedialAxisSTL( $V, T, N$ )

   $V$  list of vertices  $v_i \in \mathbb{R}^3$ 
   $T$  list of triangles  $t_j = (i_{1j}, i_{2j}, i_{3j}), 1 \leq i_{1j}, i_{2j}, i_{3j} \leq |V|$ 
   $N$  list of triangles normals,  $n_j \in \mathbb{R}^3, \|n_j\| = 1, j = 1 \dots |T|$ 

   $C$  sorted list of section centers  $c_k \in \mathbb{R}^3$ 
   $D$  corresponding list of section diameters  $d_k, k = 1 \dots |C|$ 
for all  $v \in V$  do
   $isSupport(v) \leftarrow false$ 
   $NV(v) \leftarrow$  normal of first triangle found with vertex  $v$ 
end for
 $k \leftarrow 0$   $\triangleright$  number of sections found
for all  $v \in V$  do
  if  $\neg isSupport(v)$  then
   $\triangleright v$  does not belong to any section already found
   $A \leftarrow ADJACENTVERTICES(v)$ 
   $\triangleright$  list of indices in  $V$  of vertices sharing an edge with  $v$ 
  for all  $(i, j), 1 \leq i < j \leq |A|$  do
   $u \leftarrow V[i]$ 
   $w \leftarrow V[j]$ 
  if  $TRIPLETISFEASIBLE(u, v, w)$  then
   $c, d, n \leftarrow FIT3DCIRCLE(u, v, w)$ 
   $\triangleright c$  3D center,  $d$  diameter,  $n$  normal to circle's plane
  if  $CIRCLEISFEASIBLE(d, n, v)$  then
   $S \leftarrow \emptyset$   $\triangleright$  support for circle
  for all  $v' \in V$  do
  if  $DISTTOCIRCLE(v', c, r, n) < maxDist$  then
   $S \leftarrow S \cup v'$ 
  end if
  end for
  if  $|S| > minSupport$  then
   $k \leftarrow k + 1$ 
   $C[k] \leftarrow c$ 
   $D[k] \leftarrow d$ 
  for all  $v' \in S$  do
   $isSupport(v') \leftarrow true$ 
  end for
  break  $\triangleright$  leave for all  $(i, j)$ , stop looking for feasible triplets
  end if
  end if
  end if
  end for
end for
for  $i \leftarrow 1$  to  $k$  do
  for  $j \leftarrow 1$  to  $k$  do
   $dist[i, j] \leftarrow \|C[i] - C[j]\|$ 

```

```

   $\triangleright$  adjacency matrix of fully connected graph of centers
end for
end for
 $distMST \leftarrow MINIMUMSPANNINGTREE(dist)$ 
 $\triangleright$  adjacency matrix of minimum spanning tree graph
 $C, D \leftarrow TREETRAVERSAL(distMST, C, D)$ 
return  $C, D$ 

```

Algorithm 2. Medial axis from STL (part 2).

```

function TRIPLETISFEASIBLE( $u, v, w$ )

   $u, v, w$  three different vertices

  output:
  could they belong to one same circular section? (true,false)
   $n_u \leftarrow NV(u)$ 
   $n_v \leftarrow NV(v)$ 
   $n_w \leftarrow NV(w)$ 
  return  $\min(\|u - v\|, \|u - w\|, \|v - w\|) > minDistance \wedge$ 
   $\min(|n_u \cdot n_v|, |n_u \cdot n_w|, |n_v \cdot n_w|) > minParalelism$ 
function CIRCLEISFEASIBLE( $d, n, v$ )

   $d, n$  circle diameter and plane normal
   $v$  vertex defining the circle

  output:
  is the circle an admissible circular section, consistent with  $v$  ?
  return  $d > minDiameter \wedge d < maxDiameter \wedge$ 
   $|n \cdot NV(v)| < minOrtho$ 

```

3. 3D reconstruction from CAD drawings

The second source of hose descriptions are images of CAD drawings showing orthogonal views, like in Fig. 6a. Like STL files, they are easy to obtain, for instance by scanning a printed design, saving it to a file in some common image format (TIFF, JPEG, etc.) or printing the screen while running the CAD application. If the model is geometrically simple, it is even possible to draw by hand a sketch of several projections on a sheet of paper and scan it.

CAD drawings of hoses typically show several orthogonal projections, distributed spatially like in Fig. 3. That is, two adjacent views share one common axis. Of course, designs do not always include all the views but only those necessary to convey the complete geometry of the hose. Flat tubes, devoid of torsion, need a single view whereas tubes with two or more bends with torsion

need two or more of them. We assume the projections in a design follow this same pattern, as illustrated in Figs. 5a and 6a.

Reconstruction of mechanical parts from CAD drawing views has been a recurring theme since the early days of document analysis. The survey [11] summarizes a myriad of proposals that have been published over the years, categorized by different criteria. Posterior works on this topic include [12–14]. The reconstruction of 3D hoses from CAD drawing projections shares some aspects with them: the feasibility of reconstruction from dihedral views, from one or multiple views, and the inclusion of curves in the definition of the part. However, there are also some differences that motivate a specific solution: our goal is to explicitly recover the medial axis and not the whole surface, the need to obtain a reconstruction from inaccurate and manually input points, the a priori unknown number of projections, and the strong constraint that our shapes have tubular form.

In order to reconstruct the 3D medial axis and the diameters of section from a CAD drawing image, we ask the user to input (Fig. 5a):

- The bounding box enclosing each of the projections, specified by two corners diagonally opposed. Not all of those depicted in the design may be necessary, but often the 3D reconstruction can be successfully computed with less of them. However, it is necessary that the central xy view in Fig. 3b is included if the reconstruction has to be performed on the basis of two or more projections.
- A sequence of points on the medial axis within each bounding box, from one end to the other. In order to get an accurate reconstruction, the higher the axis curvature (that is, the sharper the bending), the more points to enter, whereas straight stretches are fully specified by two points.
- A section is specified by a segment orthogonal to the 2D input medial axis polyline in some of the selected projections. Therefore, if the diameter of section is constant along the whole tube, a unique segment suffices. If not, the user may input several such segments and the diameter is linearly interpolated in-between.
- Finally, the two end-points of one segment corresponding to a longitudinal dimension plus its equivalence in millimeters or

inches, in order to compute the scale factor to convert pixels to longitudinal units.

The case of a single projection of a planar tube is straightforward, because we can readily compute the 3D coordinates of the medial axis points entered by the user. The following steps describe the method in the case of two or more projections:

End-points matching. The goal is to find the correspondences between the two end points, the first and last clicked along the medial axis in each selected projection, which are at the two extremes of the tube (Fig. 3b). This can be done thanks to the disposition of views in the design: the inclusion of the central view xy plus the fact that adjacent views share one common axis. In effect, from the number of views and their relative position we can identify the 3D axes x, y, z of the two axis of each bounding box. For instance, the axes of three projections in L shape are, from top to bottom and left to right, xz, xy, zy . This allows to match points in adjacent projections as those having a similar value for the common coordinate, provided a sufficient number of projections have been considered to avoid ambiguities.

3D point sampling. Consider the polyline defined by the sorted list of points along the medial axis entered for each projection. We now sample N equispaced points along each of them. If we suppose they are corresponding, we can compute their world 3D coordinates thanks to the labeling of each view's axis as x, y or z and the scale factor. This is an initial estimation $p_i^1, i = 1 \dots N$ of the medial axis curve that the following step will deform towards its real shape (Fig. 5b).

Iterative estimation. Let p_i^n be the sequence of points along the medial axis which constitute the present solution and which have to be updated to $p_i^{n+1}, i = 1 \dots N$ so that they get closer to the medial axis in each view. All p_i^n are projected to every orthogonal view (grey circles in Fig. 4), giving rise to 2D points r_i^k , where k indexes projections. Then, 2D vectors u_i^k are computed between r_i^k and the closest point to the polyline defined by the points manually entered by the user on the k th view (white circles). Each point p_i^n is moved in the direction of the 3D displacement vector v_i obtained by adding all the vectors u_i^k . Finally, the resulting sequence is smoothed by weighted moving average. Algorithm 3 summarizes this step.

Once we know the 3D medial axis, it is easy to compute the section diameter for each of those points Fig. 6. Then, one can

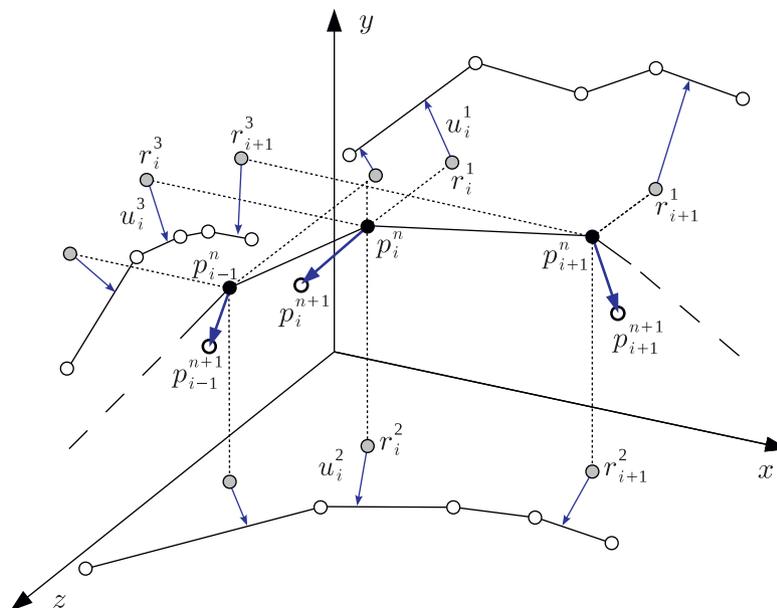
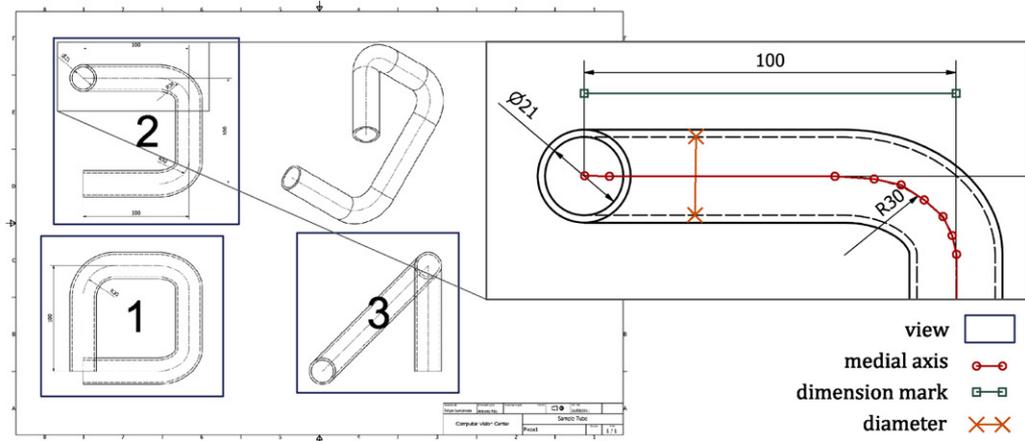
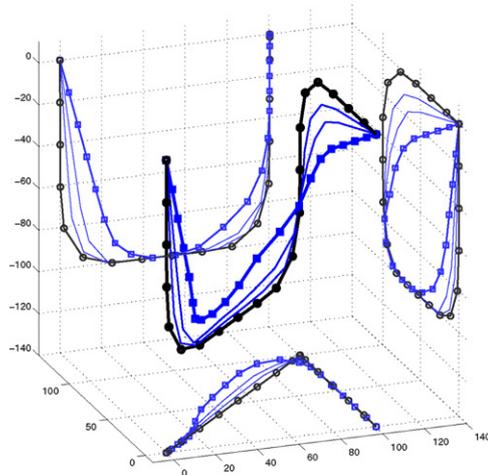


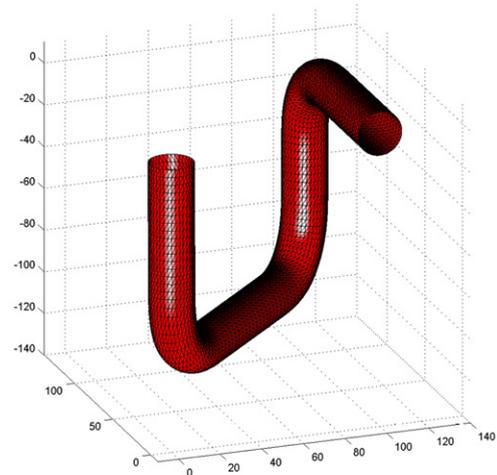
Fig. 4. Variables involved in each iteration of the 3D medial axis reconstruction, see text.



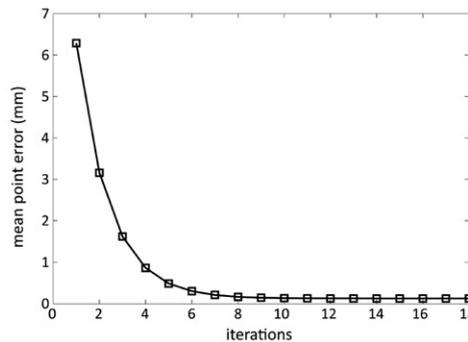
(a)



(b)



(c)



(d)

Fig. 5. (a) Manual user input on a CAD drawing image, (b) evolution of the 3D medial axis polyline at the first (solid blue curve), two intermediate and last (black) iteration, (c) reconstruction and (d) convergence to the true medial axis measured as average distance to the closest medial axis ground truth with respect to the number of iterations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

sample points along the perimeter of each circular section and generate a triangular mesh to visualize the reconstructed model, as shown in Fig. 5 c.

Algorithm 3. Iterative 3D reconstruction. Constants $\alpha = 0.5$, $\beta = 0.4$, $minDif = 0.05$, $maxNumIter = 20$.

function RECONSTRUCT(P, I, A)
input:

P initial list of N sampled 3D medial axis points, $P = \{p_i \in \mathbb{R}^3, i = 1 \dots N\}$
 $I_k = \{(a_{ki}, b_{ki}), i = 1 \dots n_k\}$ user input points along 2D medial axis of k th projection, n_k number of user input points in k th projection
 $I = \{I_k, k = 1 \dots n_p\}$, n_p number of projections
 $A_k = \{(\eta_k, \nu_k) \in \pm \{x, y, z\} \times \pm \{x, y, z\}\}$ axes of the k th projection
 $A = \{A_k, k = 1 \dots n_p\}$ axes for all views

output:

P^i final list of N medial axis points
 $n \leftarrow 1 \triangleright$ number of iterations

```

 $P^1 \leftarrow P$  ▷ initialization of  $p_i^1$ ,  $i = 1 \dots N$ 
repeat
  for  $i = 1, N$  do ▷ for each point in the present solution
     $v_i \leftarrow (0, 0, 0)$ 
    for  $k \leftarrow 1, n_p$  do ▷ for each projection
       $r_i^k \leftarrow \text{PROJECT}(p_i^k, A_k)$ 
       $u_i^k \leftarrow \text{NEARESTPOINT}(r_i^k, I_k) - r_i^k$ 
       $v_i \leftarrow \text{ADDTo3DVECTOR}(v_i, u_i^k, A_k)$ 
    end for
     $q_i \leftarrow q_i + \alpha v_i$ 
  end for
   $\text{dif} \leftarrow 0$ 
  for  $i = 2, N - 1$  do ▷ update all points but the two ends
     $p_i^{n+1} \leftarrow q_i + \beta((q_{i+1} - q_i) + (q_{i-1} - q_i))$ 
     $\text{dif} \leftarrow \text{dif} + \|p_i^{n+1} - p_i^n\|_2$ 
  end for
   $n \leftarrow n + 1$ 
   $\text{dif} \leftarrow \text{dif} / \sum_{i=2}^N \|p_i^n - p_{i-1}^n\|_2$ 
until ( $\text{dif} < \text{minDif}$ )  $\vee$  ( $n > \text{maxNumIter}$ )
return  $P^n$ 

```

4. Time regression

Once we have obtained the medial axis representation of a tube we are ready to compute relevant shape and size features from it and then estimate the manufacturing time. But which features? And how to estimate that time? One has first to find good, relevant features on which the time depends. Then, learn a good prediction (regression) function from a training set, consisting of a number of tubes for which the STL files and the expert's estimated times are available.

As for the features, what matters according to the expert are the size and the shape 'complexity'. The former clearly depends on the medial axis length and the sections diameter. The later is more difficult to precise. A tube mold takes more time to wrap if it has more bends, the bending is sharp, or the section diameter at it is large. Therefore, we have selected as candidate features the medial axis length, the most frequent of the section diameters and the average of the medial axis curvature sampled along it. One problem we have to face is the limited size of the available training set, which precludes the use of many features because of the curse of dimensionality.¹ Therefore, we have experimented with different combinations of the former list of features to come up with just two features: area (product of length and diameter) and the ratio between mean medial axis curvature and representative diameter. Another advantage of reducing the number of features to two is that their values for the training and testing samples and also the regression function, which is a surface, can be visualized (Fig. 7a).

We perform regression through a Gaussian process (GP) because of several advantages it offers. Firstly, it allows a sound Bayesian formulation of the regression problem in terms of obtaining the value which maximizes a certain predictive probability distribution conditioned to (given) the training data. Thus, regression not only produces a prediction but also a measure of its uncertainty. Second, the regression function is not restricted to belong to a certain class of functions like linear, polynomial, B-splines, etc. but a GP simply let us impose a certain degree of smoothness. Third, they are governed by just a couple of parameters, in the simplest version we use. Finally, in spite of their theoretical complexity, they are simple to implement and fast to train and compute. For the sake of completeness, we present in the following the very basics of regression through Gaussian processes. A complete development can be found in [21] and a quick introduction in [22].

¹ When the number of dimensions increases, the volume of the space increases exponentially and the available data becomes too sparse, losing their structure and hampering the predictive capacity of any regression method.

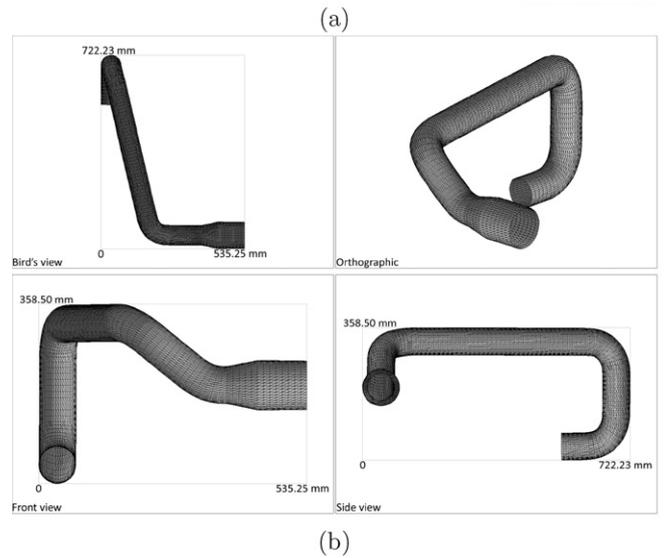
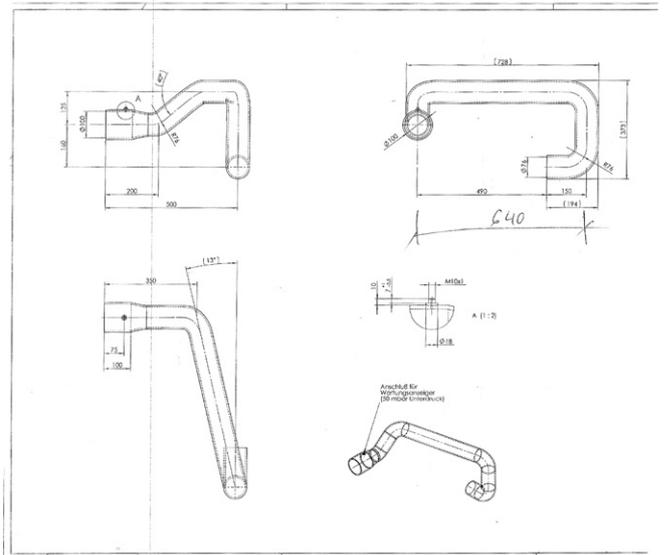


Fig. 6. (a) Scanned image of a blueprint with the three orthogonal views in European (first angle) system and (b) rendering of its 3D reconstruction from medial axis and diameter segments input by mouse clicking.

Suppose we are given a set of training samples $\{(\mathbf{x}_i, y_i), i = 1 \dots n\}$ and let X and \mathbf{y} be the matrix and column vector built by stacking the \mathbf{x}_i and y_i , respectively. Further, suppose that the training data is not noise-free but $y_i = f_i + \varepsilon$, being ε a random Gaussian noise with variance σ_{noise}^2 . We want to estimate the value f_s of the regression function at some point \mathbf{x}_s . Then, it can be shown that the most probable value for the prediction of f_s is \bar{f}_s with an uncertainty of $\text{var}(f_s)$, being

$$\bar{f}_s = K(\mathbf{x}_s, X) (K(X, X) + \sigma_{\text{noise}}^2 I)^{-1} \mathbf{y} \quad (1)$$

$$\text{var}(f_s) = K(\mathbf{x}_s, \mathbf{x}_s) - K(\mathbf{x}_s, X)^\top (K(X, X) + \sigma_{\text{noise}}^2 I)^{-1} K(X, X) \quad (2)$$

$K(P, Q) = [k(p_i, q_j)]$, $p_i \in P$, $q_j \in Q$, that is, the matrix resulting from the evaluation of k for every possible pair of points in $P \times Q$. k is the so called covariance function for which we have made a simple and common choice, the square exponential function

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right) \quad (3)$$

In practice, since $(K(X, X) + \sigma_{\text{noise}}^2 I)$ is symmetric and semi-positive definite, those computations can be performed efficiently

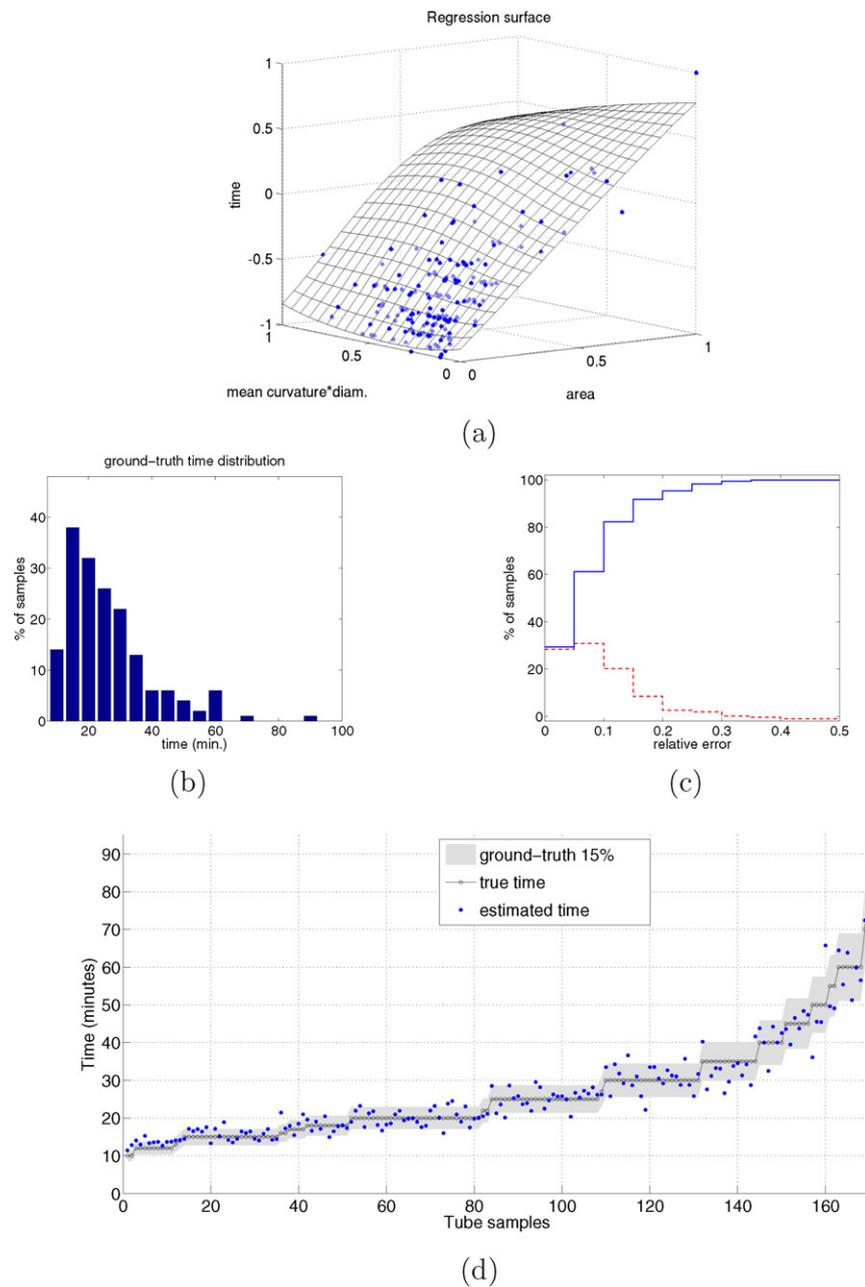


Fig. 7. (a) Regression surface for $\sigma_{\text{noise}} = 0.2$, $l = 0.6$ and feature values for the training set after normalization, (b) time distribution of the available samples, (c) absolute relative error distribution (dashed red) and accumulated distribution (continuous blue) and (d) ground-truth and predicted time for all the samples in 10-fold cross-validation and the chosen parameters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

by Cholesky decomposition. Note that in order to predict f_s given some x_s , we just need to compute the product of the row vector $K(x_s, X)$ with the previously stored fixed column vector $(K(X, X) + \sigma_{\text{noise}}^2 I)^{-1} \mathbf{y}$. In next section we detail how we have set the only two parameters σ_{noise} and l .

5. Results

The available sample set consists of 171 STL files of assorted sizes and shapes plus the corresponding expert's predicted time. The medial axis length and representative section diameter range from 13 cm to 1.2 m and from 8 mm to 20 cm, respectively. The number of bends, with or without torsion is 1–5. We partition this set into training and testing sets according to the k -fold cross-validation technique, commonly used to estimate the accuracy of a predictive model when the samples are scarce [17]. The whole

sample set is randomly divided into k subsets. Of them, $k - 1$ are used for training and the remaining once for testing. The process is repeated k times, using each subset exactly one as testing set. Thus, in the end, one gets a unique estimation for each sample after training with other samples. We have exhaustively explored the intervals $[0.1, 0.6]$ for σ_{noise} and $[0.05, 2.0]$ for l at fine steps of 0.05. For each possible pair of values we have performed k -fold cross-validation with $k = 10$, and computed the prediction error for each sample in the testing set. The combination with more samples with an absolute relative error less than 15% has been chosen, $\sigma_{\text{noise}} = 0.2$, $l = 0.6$. This threshold was considered by the expert as the acceptable error upper bound. Fig. 7a shows the resulting smooth regression surface. We observe that the samples density in the parameter space is not uniform but lower for large values of either of the two features or the time (also shown in Fig. 7b), thus producing less accurate predictions. Nevertheless, we overall

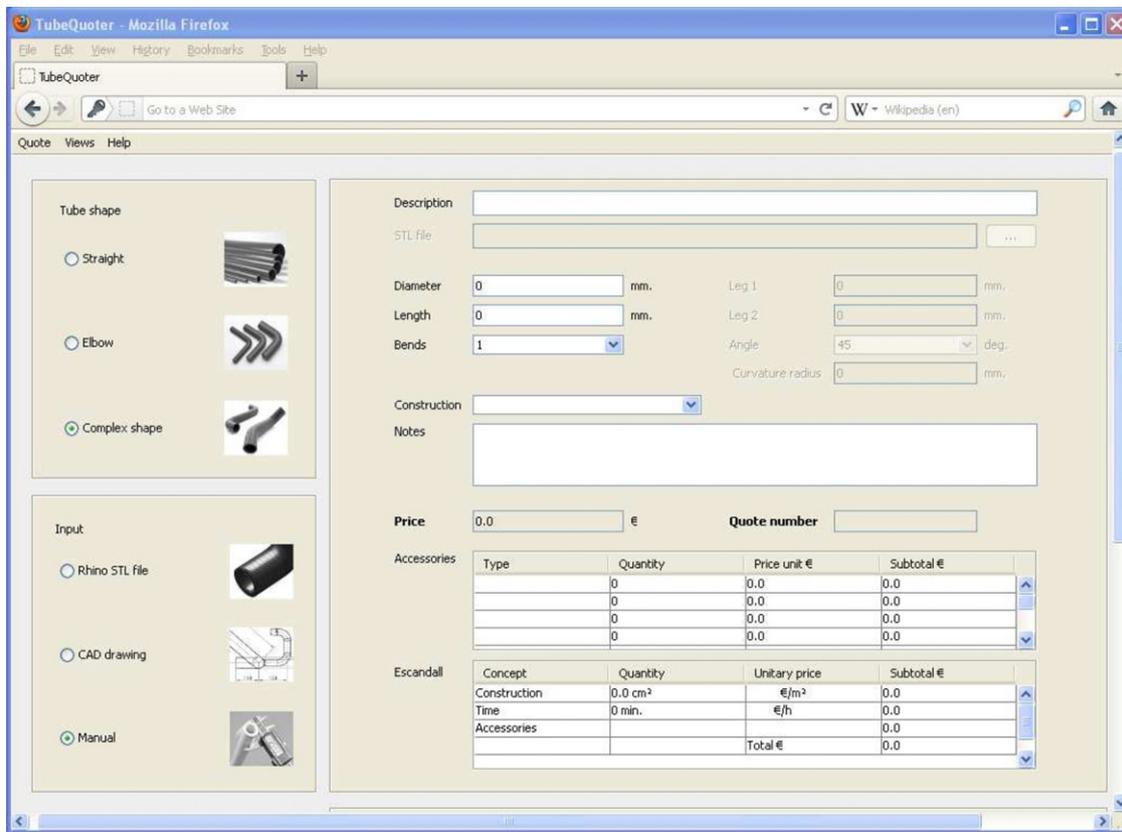


Fig. A.1. Graphical user interface of the applet.

obtain a good prediction as illustrated in Fig. 7c: 61%, 82% and 92% of samples have a relative error less than 10%, 15% and 20%, respectively.

From Fig. 7d it can be appreciated that the times decided by the expert, which we take as ground-truth, are multiple of 5 min for times greater than 20 min. The reason argued is the difficulty to decide the time more precisely for the largest and/or more complex parts. On this regard, 89% of the predictions produced by the Gaussian process lie within the expert's time ± 5 min.

6. Conclusions

We have presented a method for the quotation of custom hoses from STL files and images of CAD drawings. In both cases, it is based on the extraction of the 3D medial axis and section diameters. Size and geometry features are then computed, and a regression function, learned from a training set, is able to estimate the manufacturing time with a low relative error, assessed by k -fold cross validation. The 3D reconstruction from either STL files or CAD drawing images allows to accurately estimate the tube surface and thus also the cost of the necessary material to wrap the hose, which is in a non-negligible part of the total cost. We believe this approach of cost estimation can be extended to other tubular-like parts for which the same types of inputs are available and the fabrication process is not amenable to analytic cost estimation. Future extensions of the present work are dealing with hoses with branches and also non-circular (e.g. elliptical) sections, which are less common but nevertheless existing variants of custom hoses.

Acknowledgements

This research was partially supported by projects TRA2011-29454-C03-01, TIN2011-29494-C03-02 and Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

Appendix A. Implementation

The method presented has been implemented as a Java applet, and thus can be accessed by authorized users from any browser with Java runtime support. Fig. A.1 shows the main window of the graphical user interface. Note that there is a third input type: manually entering the length, diameter and number of bends. It also produces a price estimation by means of another learned Gaussian process regression function, though with a lower accuracy. The interface shows also the possibility of getting quotations for straight tubes and elbows, again with a different regression function in the later case.

Appendix B. Supplementary Data

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.compind.2012.11.009>.

References

- [1] A. García-Crespo, B. Ruiz-Mezcua, J. López-Cuadrado, I. González-Carrasco, A review of conventional and knowledge based systems for machining price quotation, *Journal of Intelligent Manufacturing* 22 (6) (2011) 823–841.
- [2] H. Lan, Web-based rapid prototyping and manufacturing systems: a review, *Computers in Industry* 60 (9) (2009) 643–656.
- [3] A. Niazi, J.S. Dai, S. Balabani, L. Seneviratne, Product cost estimation: technique classification and methodology review, *Journal of Manufacturing Science Engineering* 2 (128) (2006) 563–575.
- [4] K.P. Zhu, Y.S. Wong, W.F. Lu, H.T. Loh, 3D CAD model matching with 2D affine invariant features, *Computers in Industry* 61 (5) (2010) 432–439.
- [5] J. Pu, K. Ramani, On visual similarity based 2D drawing retrieval, *Computer-Aided Design* 38 (3) (2006) 249–259.

- [6] N.D. Cornea, D. Silver, P. Min, Curve-skeleton properties, applications, and algorithms, *IEEE Transactions on Visualization and Computer Graphics* 13 (3) (2007) 530–548.
- [7] G. Zwettler, F. Pfeifer, R. Swoboda, W. Backfrieder, Accelerated skeletonization algorithm for tubular structures in large datasets by randomized erosion, in: *International Conference on Computer Vision Theory and Applications*, 2008, 74–81.
- [8] A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt, On-the-fly curve-skeleton computation for 3D shapes, *Computer Graphics Forum* 26 (3) (2007) 323–328.
- [9] M. Sabry Hassouna, A.A. Farag, On the extraction of curve skeletons using gradient vector flow, in: *International Conference on Computer Vision*, 2007, 1–8.
- [10] O.K. Au, C.L. Tai, H.-K. Chu, D. Cohen-Or, T.-Y. Lee, Skeleton extraction by mesh contraction, *ACM Transactions on Graphics* 27 (3) (2008), 44:1–44:10.
- [11] P. Company, A. Piquer, M. Contero, F. Naya, A survey on geometrical reconstruction as a core technology to sketch-based modeling, *Computers & Graphics* 29 (6) (2005) 892–904.
- [12] M. Weiss-Cohen, 3D reconstruction of solid models from engineering orthographic views using variational geometry and composite graphs, *Computer-Aided Design & Applications* 4 (1) (2007) 159–167.
- [13] R. Furferi, L. Governi, M. Palai, Y. Volpe, 3D model retrieval from mechanical drawings analysis, *International Journal of Mechanics* 5 (2) (2011) 91–99.
- [14] M. Carfagni, R. Furferi, L. Governi, M. Palai, Y. Volpe, 3D reconstruction problem: an automated procedure, in: *Proc. WSEAS International Conference on Computer Engineering and Applications*, vol. 9, 2011, 9–104.
- [15] Quickparts. <http://www.quickparts.com>.
- [16] H. Lan, Y. Ding, J. Hong, H. Huang, B. Lu, Web-based quotation system for stereolithography parts, *Computers in Industry* 59 (2008) 777–785.
- [17] T. Hastie, R. Tibshirani, J.H. Friedman, *The Elements of Statistical Learning*, Springer, 2003 Available on-line at <http://www.stat.stanford.edu/~tibs/ElemStatLearn/>.
- [18] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM* 24 (1981) 381–395.
- [19] D.A. Forsyth, J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Englewood Cliffs, 2002.
- [20] R. Sedgewick, K. Wayne, *Algorithms*, 4th edition, Addison-Wesley, Boston, 2011.
- [21] C.E. Rasmussen, C.K.I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2005 Available on-line at <http://www.GaussianProcess.org/gpml>.
- [22] C.E. Rasmussen, *Gaussian Processes in Machine Learning*, Vol. 3176 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 2004, pp. 63–71.



Joan Serrat received the Ph.D. degree in computer science from the Universitat Autònoma de Barcelona (UAB), Spain, in 1990. He is currently an Associate Professor with the Department of Computer Science, UAB, where he is also a member of the Computer Vision Center. He has been the head of several machine vision projects for local industries and served as board member of the Spanish Chapter of the International Association for Pattern Recognition. He has coauthored more than 40 papers. He is the holder of three patents. His current research interest is the application of probabilistic graphical models to computer vision problems such as feature matching, tracking, and video alignment.



Felipe Lumbresas received the electromechanical B.Sc. degree in physics from the Universitat de Barcelona in 1991 and the Ph.D. degree in computer science from the Universitat Autònoma de Barcelona in 2001. He is currently an associated professor in the Computer Science Department and member of the Computer Vision Center. His research interest include wavelet and texture analysis, 3D reconstruction, and computer vision for automotive applications.



Antonio M. López received the B.Sc. degree in computer science from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 1992 and the M.Sc. degree in image processing and artificial intelligence and the Ph.D. degree from the Universitat Autònoma de Barcelona (UAB), Cerdanyola, Spain, in 1994 and 2000, respectively. Since 1992, he has been giving lectures with the Department of Computer Science, UAB, where he is currently an Associate Professor. In 1996, he participated in the foundation of the Computer Vision Center, UAB, where he has held different institutional responsibilities and is currently responsible for the research group on Advanced Driver Assistance Systems by computer vision. He has been responsible for public and private projects and is a coauthor of more than 50 papers, all in the field of computer vision.