

RECONSTRUCTION OF CONSISTENT 3D CAD MODELS FROM POINT CLOUD DATA USING *A PRIORI* CAD MODELS

Aurélien BEY^{1,2,3}, Raphaëlle CHAINE^{1,2}, Raphaël MARC³, Guillaume THIBAUT^{3,4}, Samir AKKOUCHE^{1,2}

¹Université de Lyon, CNRS

²Université Lyon 1, LIRIS, UMR5205, F-69622, France

³Électricité De France Recherche & Développement, F-92140, France

⁴CNRS, Laboratoire de Physiologie de la Perception et de l'Action, UMR7152, F-75231, France

KEY WORDS: Industrial Plants Modeling, Point Cloud, CAD Model, Reverse Engineering, Bayesian Reconstruction

ABSTRACT:

We address the reconstruction of 3D CAD models from point cloud data acquired in industrial environments, using a pre-existing 3D model as an initial estimate of the scene to be processed. Indeed, this prior knowledge can be used to drive the reconstruction so as to generate an accurate 3D model matching the point cloud. We more particularly focus our work on the cylindrical parts of the 3D models. We propose to state the problem in a probabilistic framework: we have to search for the 3D model which maximizes some probability taking several constraints into account, such as the relevancy with respect to the point cloud and the *a priori* 3D model, and the consistency of the reconstructed model. The resulting optimization problem can then be handled using a stochastic exploration of the solution space, based on the random insertion of elements in the configuration under construction, coupled with a greedy management of the conflicts which efficiently improves the configuration at each step. We show that this approach provides reliable reconstructed 3D models by presenting some results on industrial data sets.

1 INTRODUCTION

1.1 Context

3D CAD models of industrial environments such as power plants, oil and gas refineries or pharmaceutical installations are used to serve several purposes: simulation of maintenance operations, training, etc. Some of these applications require the 3D models to match precisely (up to one centimeter) the actual geometry of the facility they represent. Such models are called "as built" models and are usually constructed from 3D point clouds. In our case, the point cloud \mathcal{P} that we use come from terrestrial LIDAR acquisitions. The 3D CAD models which are considered here are CSG models made of simple primitive shapes such as planes, cones, tori and cylinders, that are truncated and assembled. Each primitive shape is a parametric entity defined by its position, orientation and some other geometric parameters inherent to its type.

Besides the point cloud \mathcal{P} , we consider that an *a priori* 3D CAD model \mathcal{M}_0 is also available. While the point cloud describes the actual state of the scene to be reconstructed, the *a priori* 3D model stands for a theoretical state of the facility: it is a rough estimate of the solution we are searching for (cf. Figure 1a). However, there may be some significant differences between \mathcal{M}_0 and the actual scene represented by \mathcal{P} : some elements of \mathcal{M}_0 may be present in the facility at a different place, with a different orientation or with a slightly different geometry. Moreover, the number of primitive shapes required to describe the whole actual scene may be different from the number of shapes in \mathcal{M}_0 . Indeed, some elements of \mathcal{M}_0 may be missing or duplicated in the facility. Inversely, some components in the facility may have no match in \mathcal{M}_0 . The *a priori* model \mathcal{M}_0 can be an existing generic 3D representation of the site (a 3D model that describes the structure of a typical facility), but it can also come from the "as built" reconstruction of another facility which looks like the one to be processed.

We focus our work on the reconstruction of right circular cylinders parts, because many shapes in industrial environments are

cylindrical and such components are very frequently used in piping and equipment design. Moreover, recognized cylinders can be subsequently used as key constraints for the detection of other shapes (especially tori and cones).

Hence given \mathcal{P} and \mathcal{M}_0 , we aim at constructing a reliable 3D model \mathcal{X} made of cylinders, such that:

- \mathcal{X} fits well to \mathcal{P} : each cylinder is close to its points,
- each cylinder of \mathcal{X} matches a cylinder from \mathcal{M}_0 : we consider that the cylinders which are to be reconstructed must appear in \mathcal{M}_0 up to some specified change tolerances, so as to provide reliable results,
- \mathcal{X} is consistent: the connections between its cylinders are relevant with respect to the constraints observed in industrial facilities. These connections constraints are also some kind of *a priori* knowledge about the scene to be reconstructed, besides the model \mathcal{M}_0 .

Our contributions to the reverse engineering field are the expression of a new probabilistic framework for this reconstruction problem, as well as a new approach to perform the optimization stated in this framework. Our method introduces the ideas of checking the reconstructed model consistency and guiding the reconstruction by using the *a priori* 3D model. Hence, the reconstructed model does not only fit well to the point cloud, but is also relevant with respect to the industrial environments constraints, which is usually not guaranteed by existing approaches, and its similarity with the *a priori* 3D model can also be ensured.

1.2 Related work

To our knowledge, the work of (Bosché, 2010) is the only reverse engineering approach that uses *a priori* CAD models as an initial estimate of the point cloud to be reconstructed. The 3D model is fitted to the point cloud using a registration approach based on the ICP algorithm, which is only supposed to converge when the data to be registered are close enough. So this approach cannot handle significant changes.

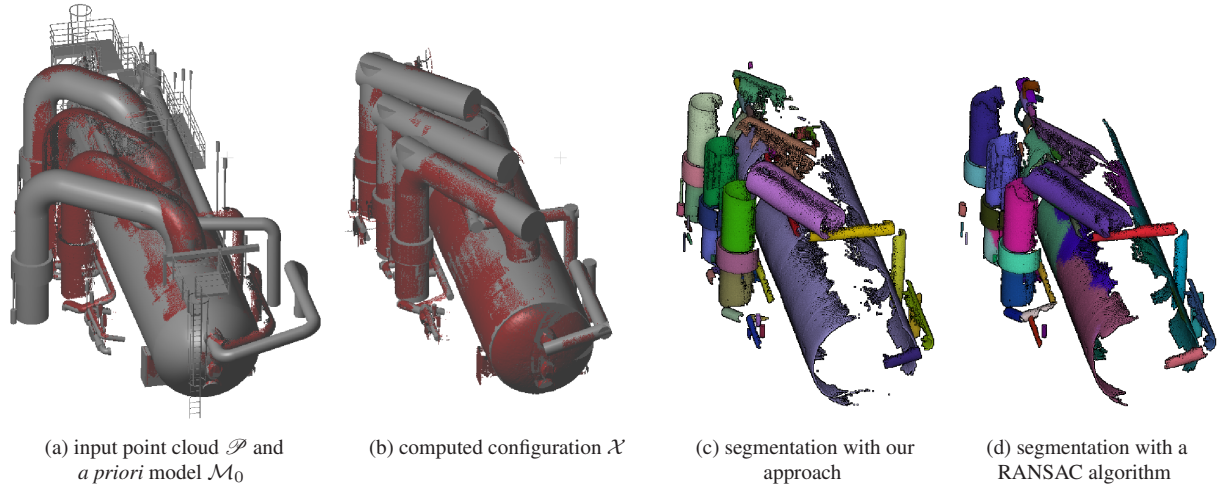


Figure 1: Reconstruction results on an industrial dataset. The *a priori* 3D model (grey in Figure (a)) has 626 shapes, among which there are 391 cylinders, and presents some obvious differences with respect to the point cloud (red: 990323 points). Most of the cylinders in the *a priori* model are not present in the point cloud. The computed configuration (grey in Figure (b)) has 68 cylinders, and the corresponding segmentation is shown in Figure (c): each segment is displayed with its own random color. These results can be compared with the segmentation provided by the algorithm of (Schnabel et al., 2007) shown in Figure (d).

(Schmittwilken and Plümer, 2010) also use strong prior informations such as the size of windows to classify distinct parts of building facade, but their work is very specific to the facade parts classification problem, although interesting ideas could also be used in another context. There also exists some researches that drive the surface reconstruction from point clouds by using prior knowledge (Pauly et al., 2005, Jenke et al., 2006, Gal et al., 2007). These works use 3D meshes to model the surfaces whereas we would like to build the scenes with primitive shapes, which gives a level of representation more adapted to the targeted applications.

Two approaches are widely used to handle the shapes recognition problem : the Generalized Hough Transform (GHT) and the RANdom SAMple Consensus (RANSAC) algorithm. The GHT is a vote based-approach: each point votes for a set of shapes in a discretized parametric space, and the shapes whose associated parametric description correspond to a peak of votes can be considered as present in the point cloud. The main drawback of this method, when naively used, is its prohibitive complexity since the whole shape parametric space must be scanned (5 dimensions for the cylinder). The complexity can be lowered in the case of the cylinder by resolving some parts of the problem in lower dimensions spaces (Rabbani and Van Den Heuvel, 2005). This method provides nice results on part of industrial scenes, but the detection of accurate cylinders with different sizes (especially the smallest ones) in large scenes remains challenging. The RANSAC algorithm (Fischler and Bolles, 1987) is a stochastic approach which consists in randomly picking minimal points sets from which candidate shapes are built. Without improvement, most of the generated candidates are not relevant and the number of trials that is required to find a correct shape can be huge. (Schnabel et al., 2007) propose an efficient algorithm that decreases the minimal number of trials that ensures shapes recognition. The authors also provide statistical tools that speed up the score computation of candidate shapes. This approach has shown good results on the data sets we have tested, although there are still some precision issues that lead to irrelevant shapes.

We finally would like to mention the work of (Fisher, 2004) that points out the interest of using *a priori* knowledge in the reverse engineering process, so as to provide high quality results. The

authors notably propose to take shapes relationships into account, by adding constraints in the localized shape fitting problem.

1.3 Overview

The use of a prior knowledge lends itself quite well to the Bayesian formulation (Diebel et al., 2006) of the reconstruction problem which states this task as the research of the most probable configuration (cf. Section 2.2). The underlying probability must be defined to set the requirement that we want the reconstructed model to fulfill. Therefore we propose a probability that embeds several kinds of constraints (cf. Section 2.3), such as the quality of the fitting to the point cloud, the similarity with the *a priori* 3D model and the quality of the connectivity between shapes in the configuration.

Once the probability has been defined, we have to find an efficient way to compute the optimal configuration among the huge and complex set of possible solutions. For that purpose, we propose a new iterative optimization approach based on the stochastic exploration of the configuration space (cf. Section 3.1). A candidate cylinder is randomly generated at each step, in order to insert it into the configuration. As its insertion may decrease the probability of the current configuration, we propose a greedy method to handle conflicts with the cylinders already included in the configuration: the candidate cylinder is accepted if it increases the probability once the conflicts have been resolved. We also present a cylinder generation algorithm that favors the relevant cylinders appearing, by using both \mathcal{P} and \mathcal{M}_0 to guide the random generation (cf. Section 3.2).

2 A PROBABILISTIC APPROACH

2.1 Introduction about data and notations

Let $\mathcal{P} = \{(\mathbf{p}_0, \mathbf{n}_0), \dots, (\mathbf{p}_{n-1}, \mathbf{n}_{n-1})\}$ be a point cloud that stands for the actual state of a facility. Each pair $(\mathbf{p}_j, \mathbf{n}_j) \in \mathcal{P}$ consists of a 3D point \mathbf{p}_j and its associated normal vector \mathbf{n}_j , which can either be provided with LIDAR data or computed using a local plane fitting approach (Mitra and Nguyen, 2004). The *a priori* 3D CAD model $\mathcal{M}_0 = \{C^0, \dots, C^{m-1}\}$ is a set of

primitive shapes, among which we only consider right circular cylinders. Each cylinder \mathcal{C} considered in this paper is truncated and defined by its center of mass \mathbf{c}_C , its axis direction \mathbf{a}_C , its radius r_C and its length l_C . We aim at finding the most probable configuration $\mathcal{X} = \{\mathcal{C}_0, \dots, \mathcal{C}_{x-1}\}$, in which each \mathcal{C}_j is a cylinder.

2.2 Reconstruction as an optimization problem

Considering the reconstruction problem from the probabilistic point of view, we are searching for the configuration that maximizes some posterior probability $\pi(\cdot)$ which is proportional, up to some constant normalization factor, to the product of a **Data** likelihood term P_D and a **Prior** term P_P according to the Bayes rule:

$$\pi(\mathcal{X}|\mathcal{P}, \mathcal{M}_0) \propto P_D(\mathcal{P}|\mathcal{X}, \mathcal{M}_0) \times P_P(\mathcal{X}|\mathcal{M}_0) \quad (1)$$

" \propto " stands for the proportionality relation : the constant normalization term is not involved in the optimization problem.

$P_D(\mathcal{P}|\mathcal{X}, \mathcal{M}_0)$ is in fact independent of \mathcal{M}_0 since \mathcal{P} does not depend on \mathcal{M}_0 . It quantifies how well \mathcal{P} fits to \mathcal{X} , whereas P_P defines some prior density over the whole space of possible configurations. We have to ensure that the prior term P_P favors consistent configurations and takes the similarity with \mathcal{M}_0 into account. Using the densities:

$$P_D(\mathcal{P}|\mathcal{X}) = e^{-\lambda_D \cdot E_D(\mathcal{X}, \mathcal{P})} \quad (2)$$

$$P_P(\mathcal{X}|\mathcal{M}_0) = e^{-E_P(\mathcal{X}, \mathcal{M}_0)} \quad (3)$$

we have to find the configuration \mathcal{X} that minimizes the energy:

$$\begin{aligned} E(\mathcal{X}, \mathcal{P}, \mathcal{M}_0) &= -\log(\pi(\mathcal{X}|\mathcal{P}, \mathcal{M}_0)) \\ &= \lambda_D \cdot E_D(\mathcal{X}, \mathcal{P}) + E_P(\mathcal{X}, \mathcal{M}_0) \end{aligned} \quad (4)$$

where λ_D specifies the relative importance of the data fitting energy E_D with respect to the prior energy E_P .

From equations 2 and 3, we can see that the **Data** fitting energy E_D measures the quality of fitting of \mathcal{P} to \mathcal{X} , while the **Prior** energy E_P measures the compliance of \mathcal{X} with the *a priori* expectations.

2.3 Energies formulation

2.3.1 Data fitting term Given a threshold ϵ which corresponds to an estimate of the noise in \mathcal{P} , we define the data fitting energy E_D as follows:

$$E_D(\mathcal{X}, \mathcal{P}) = \frac{1}{n} \sum_{\mathcal{C}_i \in \mathcal{X}} \sum_{\mathbf{p} \in \mathcal{P}} e_D(\mathcal{C}_i, \mathbf{p}) \quad (5)$$

remind n is the number of elements in \mathcal{P} . If $d(\mathcal{C}_i, \mathbf{p})$ (shortened $d_{\mathbf{p}}^i$) is the distance between \mathbf{p} and \mathcal{C}_i , then we define e_D as follows:

$$e_D(\mathcal{C}_i, \mathbf{p}) = \begin{cases} \left(\frac{d_{\mathbf{p}}^i}{\epsilon}\right)^2 - 1 & \text{if } d_{\mathbf{p}}^i < \epsilon \text{ and } d_{\mathbf{p}}^i = \min_{\mathcal{C}_k \in \mathcal{X}} d_{\mathbf{p}}^k \\ \omega_D & \text{if } \epsilon \leq d_{\mathbf{p}}^i \leq 3\epsilon \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Any point \mathbf{p} that falls into a cylinder medium-range neighborhood $[\epsilon, 3\epsilon]$ brings a constant positive penalty ω_D . Otherwise, if the point is close enough to a cylinder, it brings a negative energy which tends to its minimum (-1) as the distance between the point and its closest cylinder decreases. The points that do not meet one of these two cases do not contribute to the energy. This

measurement encourages cylinders to be close to their matching points, but it also penalizes those that do not fulfill an emptiness constraint in their medium-range neighborhood.

2.3.2 Prior term The prior density quantifies the relevancy of \mathcal{X} regardless of \mathcal{P} . It induces a probability over the whole configuration space, and thus defines the relevancy of any configuration based on *a priori* criteria. The two main criteria we introduce in the reconstruction process are the following: a configuration is likely if its elements connect well one to each other, and if it "looks like" the initial estimate \mathcal{M}_0 up to some specified tolerances σ_A , σ_R and σ_C . Therefore it is quite natural to split the prior energy E_P into two separated terms: a **Topological** term E_T that handles objects connectivity, and a **Geometrical** term E_G that handles the similarity with \mathcal{M}_0 :

$$E_P(\mathcal{X}, \mathcal{M}_0) = \lambda_T \cdot E_T(\mathcal{X}) + \lambda_G \cdot E_G(\mathcal{X}, \mathcal{M}_0) \quad (7)$$

where the factors λ_T and λ_G modulate the relative importance of each term.

We define E_G element-wise: each cylinder of \mathcal{X} is compared with its most likely match in \mathcal{M}_0 and brings the corresponding energy:

$$E_G(\mathcal{X}, \mathcal{M}_0) = \sum_{\mathcal{C}_i \in \mathcal{X}} \min_{\mathcal{C}^j \in \mathcal{M}_0} e_G(\mathcal{C}_i, \mathcal{C}^j) \quad (8)$$

where e_G splits into three terms (one for each of the main cylinder parameters):

$$e_G(\mathcal{C}_i, \mathcal{C}^j) = e_G^A(\mathcal{C}_i, \mathcal{C}^j) + e_G^R(\mathcal{C}_i, \mathcal{C}^j) + e_G^C(\mathcal{C}_i, \mathcal{C}^j) \quad (9)$$

The angular term e_G^A decreases as the axes directions align:

$$e_G^A(\mathcal{C}_i, \mathcal{C}^j) = \left(\frac{\arccos(|\mathbf{a}_{\mathcal{C}_i} \cdot \mathbf{a}_{\mathcal{C}^j}|)}{\sigma_A} \right)^2 \quad (10)$$

The radius term e_G^R decreases as the radii tend one to the other:

$$e_G^R(\mathcal{C}_i, \mathcal{C}^j) = \left(\frac{r_{\mathcal{C}_i} - r_{\mathcal{C}^j}}{\sigma_R \cdot r_{\mathcal{C}^j}} \right)^2 \quad (11)$$

The position term e_G^C decreases as \mathcal{C}_i position gets closer to \mathcal{C}^j axis:

$$e_G^C(\mathcal{C}_i, \mathcal{C}^j) = \left(\frac{\|(\mathbf{c}_{\mathcal{C}_i} - \mathbf{c}_{\mathcal{C}^j}) \times \mathbf{a}_{\mathcal{C}^j}\|}{\sigma_C} \right)^2 \quad (12)$$

In the above expressions, " \cdot ", " \times " and " $\| \cdot \|$ " respectively denote the usual scalar product, cross product and Euclidean norm in \mathbb{R}^3 .

The topological energy E_T is much more complex because the set of connections that should be encouraged and those that should be penalized is harder to define. We make the following assumptions based on expert knowledge:

- the axes of two connected cylinders should "intersect",
- connected cylinders should have a negligible overlapping volume,
- the angle between two connected cylinders should be either right (90°) or flat (0°),
- the connections involving two cylinders with the same radius are relevant, although connections between two cylinders with different radii are not forbidden (e.g. T-junctions).

Thus the energy E_T does not depend on \mathcal{M}_0 , and is defined as

the sum of connections energies:

$$E_T(\mathcal{X}) = \sum_{(C_i \in \mathcal{X}, C_j \in \mathcal{X}), i \neq j} e_T(C_i, C_j) \quad (13)$$

With:

$$e_T(C_i, C_j) = \begin{cases} 0 & \text{if } C_i \text{ does not intersect } C_j \\ g_\tau(C_i, C_j) & \text{if } 0 < \mathcal{V}(C_i \cap C_j) < \frac{1}{2} \\ \omega_T & \text{if } \frac{1}{2} \leq \mathcal{V}(C_i \cap C_j) \end{cases} \quad (14)$$

$\mathcal{V}(C_i \cap C_j)$ is the maximum between the proportion of C_i volume intersecting C_j and the proportion of C_j volume intersecting C_i . It can be grossly approximated using a Monte Carlo approach: a few hundreds of points are randomly picked in each cylinder, and we count the proportion of these points that falls into both cylinders simultaneously (the maximum of the two resulting proportions is kept). This way, overlapping cylinders can be strongly penalized using the positive constant ω_T . The cylinders that do not intersect are not taken into account for the topological energy computation. When the cylinders connection seems to be relevant, we define an energy that favors connections which fulfill the remaining requirements mentioned above (axes intersections, right angles, similar radii):

$$g_\tau(C_i, C_j) = g_\tau^L(C_i, C_j) + g_\tau^R(C_i, C_j) + g_\tau^A(C_i, C_j) \quad (15)$$

Again, we have three distinct terms handling several constraints. As the cylinder center \mathbf{c}_{C_i} (resp. \mathbf{c}_{C_j}) and its axis direction \mathbf{a}_{C_i} (resp. \mathbf{a}_{C_j}) define a 3D line, and provided a distance function $d((\mathbf{c}_{C_i}, \mathbf{a}_{C_i}), (\mathbf{c}_{C_j}, \mathbf{a}_{C_j}))$ between two 3D lines, the axis intersection constraint is quantified by the following energy:

$$g_\tau^L(C_i, C_j) = 3 - 4 \left(\frac{3}{4} \right) \left(\frac{d((\mathbf{c}_{C_i}, \mathbf{a}_{C_i}), (\mathbf{c}_{C_j}, \mathbf{a}_{C_j}))}{\tau \times \min(r_{C_i}, r_{C_j})} \right)^2 \quad (16)$$

where τ is a tolerance ratio specified by the user.

The angular term g_τ^A favors right or flat angles (negative energy) while disadvantaging other angles (positive energy):

$$g_\tau^A(C_i, C_j) = 3 - 4 \left(\frac{3}{4} \right) \left(\frac{\arccos(|\mathbf{a}_{C_i} \cdot \mathbf{a}_{C_j}|)}{a} \right)^2 - 4 \left(\frac{3}{4} \right) \left(\frac{\arccos(|\mathbf{a}_{C_i} \cdot \mathbf{a}_{C_j}|) - \frac{\pi}{2}}{a} \right)^2 \quad (17)$$

a being an angular tolerance specified by the user.

Eventually, the radius term g_τ^R favors connections where the involved cylinders have the same radius, and is quite neutral when radii are different. For that purpose, we consider that the radii difference should not be much greater than the ϵ parameter introduced in section 2.3.1:

$$g_\tau^R(C_i, C_j) = -e - \frac{1}{2} \left(\frac{r_{C_i} - r_{C_j}}{\epsilon} \right)^2 \quad (18)$$

The terms g_τ^L and g_τ^A define hardcore constraints: when one is not fulfilled, the corresponding energy tends to its upper bound (3), and the sum g_τ is turned into a positive penalty because each of the two other terms is greater than -1. This explains why we use the values 3 and 4 in Equations 16 and 17, although they could be substituted for an other pair $(\gamma, \gamma + 1)$, as long as $\gamma > 2$.

3 RESOLUTION

3.1 Greedy optimization

We have to generate some configuration \mathcal{X} that minimizes the energy E (cf. Section 2.2), but we do not precisely know beforehand how many cylinders there are in \mathcal{X} . Usual numerical optimization tools can hardly handle such a task.

In (Descombes et al., 2008, Lafarge et al., 2010), the authors solve similar problems by using approaches based on time reversible Markov chains in a simulated annealing scheme. Indeed, the configurations space can be explored by randomly jumping from a configuration to a new one, each jump being the insertion or the removal of a random element ("birth and death" approach), and randomly accepting these transitions depending on the energy evolution. This process stands for a reversible Markov chain whose stationary distribution corresponds to the energy to be minimized, which ensures the convergence of the process to the global optimum.

As far as we are concerned, we think that the generation of an element (cylinder) should be driven by the available knowledge \mathcal{M}_0 (cf. Section 3.2). The blind generation of a cylinder by uniformly sampling its parameters would cause the resolution to take a prohibitive time, since such a cylinder is unlikely to optimize the energy. But by introducing a bias in the birth kernel, we do not ensure the time reversibility of the resulting Markov chain. Hence there is no guarantee about the process convergence. Moreover, the simulated annealing algorithms rely on a cooling schedule whose role is critical in the convergence, and which requires the user to specify additional parameters, which is really difficult for someone who has not a deep knowledge of it.

Algorithm 1 Greedy energy minimization algorithm

Require: \mathcal{P} : point cloud to be reconstructed

\mathcal{M}_0 : 3D model, gross estimate of \mathcal{P}

Ensure: \mathcal{X} fits to \mathcal{P} , "looks like" \mathcal{M}_0 and is consistent

```

1:  $\mathcal{X} \leftarrow \emptyset$ 
2: repeat
3:   for all  $C^i \in \mathcal{M}_0$  do
4:     Randomly build a cylinder  $C$  from  $\mathcal{P}$  using  $C^i$ 
5:      $\mathcal{X}^* \leftarrow \mathcal{X} \cup \{C\}$ ,  $E^0 \leftarrow E(\mathcal{X}^*, \mathcal{P}, \mathcal{M}_0)$ 
6:     compute the set  $\mathcal{K}$  of cylinders in  $\mathcal{X}$  that intersect  $C$ 
7:     sort  $\mathcal{K}$  in decreasing  $e_T(C, C_j \in \mathcal{K})$  order
8:      $\mathcal{K}^* \leftarrow \emptyset$ ,  $\mathcal{Q} \leftarrow \emptyset$ 
9:     for all  $C_j \in \mathcal{K}$  do
10:       $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{C_j\}$ ,  $E^j \leftarrow E(\mathcal{X}^* - \mathcal{Q}, \mathcal{P}, \mathcal{M}_0)$ 
11:      if  $E^j < E^0$  then
12:         $E^0 \leftarrow E^j$ ,  $\mathcal{K}^* \leftarrow \mathcal{K}^* \cup \mathcal{Q}$ 
13:      end if
14:    end for
15:    if  $E^0 < E(\mathcal{X}, \mathcal{P}, \mathcal{M}_0)$  then
16:       $\mathcal{X} \leftarrow \mathcal{X}^* - \mathcal{K}^*$ 
17:    end if
18:  end for
19: until  $\mathcal{X}$  has not been changed
```

Therefore instead of using a probabilistic minimization algorithm, we propose a greedy minimization method (Algorithm 1) based on the stochastic exploration of the solution space : the decision whether a transition should be accepted or not is taken in a deterministic way, so as to minimize E at each step. When a new cylinder C is generated (cf. Section 3.2), we try to insert it in the configuration \mathcal{X} . This insertion may increase the energy although C may be a perfect candidate, due to conflicts with some cylinders in \mathcal{X} . Therefore before taking any decision, we have

to resolve these conflicts. Let $\mathcal{K} \subset \mathcal{X}$ be the set of cylinders intersecting \mathcal{C} . We have to find the subset $\mathcal{K}^* \subset \mathcal{K}$ that minimizes $E(\mathcal{X} - \mathcal{K}^* \cup \{\mathcal{C}\}, \mathcal{P}, \mathcal{M}_0)$, i.e. the optimal removal of conflicting cylinders. Theoretically, we should test all the parts of \mathcal{K} to find \mathcal{K}^* . To avoid combinatorial explosion, we use a greedy heuristic: the cylinders in \mathcal{K} whose topological energy e_T against \mathcal{C} is high are removed first. By removing the conflicting cylinders in this order, we can keep track of the set which gets to the smallest energy in a time which is proportional to $|\mathcal{K}^*|$ instead of $2^{|\mathcal{K}^*|}$. Finally, once the conflicts have been resolved, the cylinder is accepted only if it decreases the energy.

3.2 Cylinders generation

In our case, an estimate of the solution we are searching for is available: this *a priori* knowledge \mathcal{M}_0 can be used to generate relevant cylinders from the point cloud, and thus increase the convergence rate of the optimization process. Given a cylinder $\mathcal{C}^i \in \mathcal{M}_0$, we can build a candidate cylinder using a RANSAC-like algorithm:

1. use the *a priori* cylinder \mathcal{C}^i to locate the area where the candidate is going to be generated:
 - (a) take a first point \mathbf{q}_0 close to \mathcal{C}^i , using a rejection test: randomly pick points until the current one is close enough to \mathcal{C}^i and its normal is relevant with respect to \mathcal{C}^i ,
 - (b) shift \mathbf{q}_0 along its normal vector \mathbf{n} : $\mathbf{c} \leftarrow \mathbf{q}_0 \pm r_{\mathcal{C}^i} \mathbf{n}$. Then take the spherical neighborhood of \mathbf{c} with a radius proportional to $r_{\mathcal{C}^i}$. We use KD-Trees to handle such queries,
2. build a candidate from a non-minimal points set:
 - (a) among the selected neighbors, choose κ that are likely to lie on the same surface as \mathbf{q}_0 . This likelihood can be estimated using a computation of the local cylinder axis based on the Gaussian map (Rabbani and Van Den Heuvel, 2005), then projecting the points into a plane orthogonal to this axis, and performing a circle detection among projected neighbors (using a RANSAC algorithm for instance). The probability for each neighbor to be picked is then computed based on the distance of its projection to the detected circle with respect to the expected noise ϵ .
 - (b) build the cylinder \mathcal{C} that minimizes the mean square distance to the κ selected points. The length of this cylinder is computed using its ϵ -inliers in \mathcal{P} (points whose distance to \mathcal{C} is smaller than ϵ).

This algorithm randomly generates relevant cylinders that tend to "look like" the input cylinder \mathcal{C}^i (thanks to step 1), and fit well to \mathcal{P} (step 2). Indeed, we use \mathcal{C}^i parameters to drive the points selection step. And since the point selection process is reliable, we can build cylinders from many points (we use $\kappa = 20$), while usual RANSAC algorithms use minimal sets of points (2 to 7). It makes the generation step more precise and robust against noise and normals uncertainties. Thus, the probability for a cylinder to be generated increases as its energies E_D and E_G decrease. Moreover, the set of cylinders that can be built with this method is finite and countable, and its cardinality is bounded by the number of parts of \mathcal{P} having κ elements. In particular, the cylinders that do not minimize the mean square distance to κ points of \mathcal{P} have a zero probability to be generated. So the proposed method defines a probability triple over the cylinders space, with a finite countable set of events having a non-zero probability, and it can be used for the efficient exploration of the solution space in order to solve the Bayesian problem stated in this paper.

4 RESULTS

4.1 A few words about the parameters setting

We have introduced several parameters in the previous sections. Some of them have to be specified by the user, since they bring knowledge about the problem that could hardly be computed. Thus ϵ specifying the noise in \mathcal{P} , σ_A , σ_R and σ_C specifying uncertainties about the initial estimate \mathcal{M}_0 , and the weighting parameters λ_D , λ_G and λ_T have to be set by the user at the beginning of the algorithm. However, it is difficult to set the energies weights λ_D , λ_G and λ_T , and some approach that automatically estimates these parameters would be helpful.

Some other parameters can either be guessed based on assumptions, or can be set regardless of the input data \mathcal{P} and \mathcal{M}_0 . For instance, a topological tolerance $\tau = 0.1$ can be used: in any case, the distance between connected cylinders axes should not exceed a tenth of the smaller cylinder radius (cf. equation 16), and $\alpha = 5^\circ$ should also be convenient for most cases (cf. equation 17). Assuming that at most 10% of any cylinder inliers may lie into the medium-range neighborhood, we propose to set $\omega_D = 12$. Finally, setting $\omega_T = 4$ enables the removal of overlapping cylinders even if they have up to 4 perfect connections with some other cylinders, which should be sufficient in many scenes (otherwise the greater ω_T , the better).

4.2 Results on industrial scenes

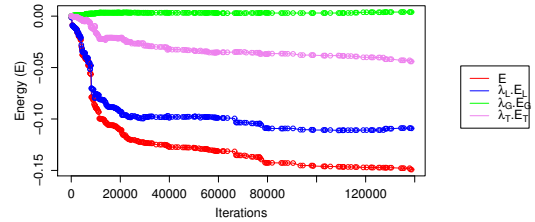


Figure 2: Evolution of the energies during the reconstruction process, for the dataset presented in Figure 1. Here, we use the weighting coefficients $\lambda_D = 0.1$, $\lambda_T = 0.7$ and $\lambda_G = 0.2$.

We present the results that we got on an industrial dataset shown in Figure 1, in order to prove the ability of our method to generate consistent model from LIDAR data. We use an *a priori* 3D model which is roughly registered with the point cloud (the greatest registration errors are about 10 cm), and which is different from the 3D point cloud: some pipes are missing, and other ones are significantly modified (reoriented and shifted). Therefore we use permissive change tolerances: $\sigma_A = 30^\circ$, $\sigma_R = 0.2$ and $\sigma_C = 3.5$ m. The reconstructed configuration contains pipes that were initially missing, as well as those that were displaced. Moreover, we can see that it is consistent: its cylinders are well connected, and there are no undesirable overlaps. It also fits quite well to the point cloud. Another interesting aspect of this algorithm is that it enables the reconstruction of a large variety of cylinders: we observe that it detects cylinders of radius 10 mm as well as a cylinder of radius 2300 mm, although the detection of the smallest elements is not very precise because their radii are smaller than the specified noise (here, we use $\epsilon=30$ mm).

The evolution of the energies during the resolution process is shown in Figure 2. It takes about 140000 iterations (by iteration, we mean a generated candidate) for the algorithm to converge (about 1700 seconds with a 2.4 GHz processor).

We also compare our approach with a reference RANSAC algorithm (Schnabel et al., 2007) which detects tori, cylinders, planes and spheres without using any prior knowledge, and which is faster than ours (about 30 seconds to perform the detection). The segmented cylinders are shown in Figure 1d, but we do not consider the other shapes handled by the RANSAC algorithm. We can see by comparing the segmentation results that our approach manages to correctly detect the biggest cylinders, whereas the RANSAC algorithm detects several pieces of overlapping shapes (many colours standing for a single cylinder), especially for the cylinder at the center of the scene. The following table shows a few comparisons between the RANSAC method (we only consider cylinders) and our approach. We can see that our method works better in fitting the cylinders to their points (second line), as well as in avoiding conflicts between shapes (fourth line) and optimizing the distribution of points on the shapes surfaces (third line).

	RANSAC	Bayesian
Amount of cylinders	42	68
Average distance to ϵ -inliers	14.58 mm	10.12 mm
Average amount of points per surface unit	21,18 points/dm ²	25,81 points/dm ²
Points that simultaneously match more than one cylinder	5.67%	0.85%

Finally, we have run our point cloud reconstruction approach with a part of the *a priori* 3D model (Figure 3). As expected, the algorithm manages to find only the parts that match the incomplete input CAD model.

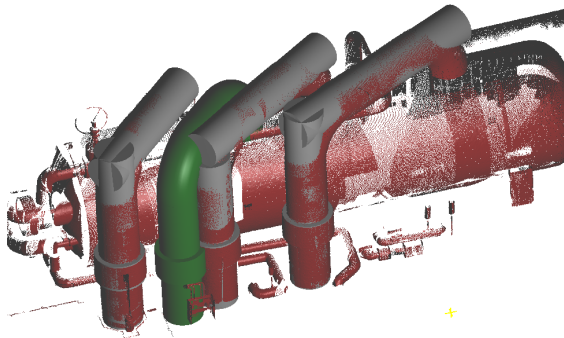


Figure 3: Reconstruction from a partial 3D model. As it searches for a model that is similar to the *a priori* (green), our approach only retrieves the appropriated parts (grey) in the point cloud (red). We use the same parameters as for the scene in Figure 1.

5 CONCLUSION

We have stated the reconstruction problem as the search of a 3D CAD model that fulfills several requirements, among which some are based on *a priori* knowledge. We have more particularly focused our approach on the use of an existing 3D CAD model which roughly describes the scene we aim at reconstructing. This *a priori* knowledge is used to improve the reliability of reconstructed models. Indeed, the existing reconstruction techniques mostly deal with the quality of fitting to the input point cloud, whereas the Bayesian formulation that we propose allows us to embed higher level constraints, such as the consistency of the resulting model or the compliance with the *a priori* CAD model. Moreover, we use the *a priori* CAD model to focus the reconstruction on parts of the scene that we know to be relevant.

The posterior probability that we have proposed define the three requirements we want the reconstructed model to fulfill. Besides,

we have proposed a new approach which searches for the optimal model with respect to this probability. This approach has shown good results on the industrial datasets that we have tested so far: the resolution method actually provides solutions that have high probabilities (low energy), and the resulting CAD models fit well to the point clouds, their shapes are well connected and each shape actually matches one from the *a priori* CAD models.

To conclude, we think that our approach could be extended to other kind of shapes, such as planes. But handling the components interactions gets much more complex as the number of shape types increases.

REFERENCES

- Bosché, F., 2010. Automated Recognition of 3D CAD Model Objects and Calculation of As-built Dimensions for Dimensional Compliance Control in Construction. Elsevier Journal of Advanced Engineering Informatics.
- Descombes, X., Minlos, R. and Zhizhina, E., 2008. Object Extraction Using a Stochastic Birth-and-Death Dynamics in Continuum. Journal of Mathematical Imaging and Vision 33(3), pp. 347–359.
- Diebel, J., Thrun, S. and Brünig, M., 2006. A Bayesian method for probable surface reconstruction and decimation. ACM Transactions on Graphics (TOG) 25(1), pp. 39–59.
- Fischler, M. A. and Bolles, R. C., 1987. Random Sample Consensus : a Paradigm for Model Fitting with applications to Image Analysis and Automated Cartography. Readings in computer vision: issues, problems, principles, and paradigms pp. 726–740.
- Fisher, R., 2004. Applying knowledge to reverse engineering problems. Computer-Aided Design 36(6), pp. 501–510.
- Gal, R., Shamir, A., Hassner, T., Pauly, M. and Cohen-Or, D., 2007. Surface Reconstruction using Local Shape Priors. Proceedings of the fifth Eurographics symposium on Geometry processing pp. 253–262.
- Jenke, P., Wand, M., Bokeloh, M., Schilling, A. and Straßer, W., 2006. Bayesian point cloud reconstruction. Computer Graphics Forum 25(3), pp. 379–388.
- Lafarge, F., Gimel'farb, G. and Descombes, X., 2010. Geometric feature extraction by a multimarked point process. IEEE transactions on pattern analysis and machine intelligence 32(9), pp. 1597–609.
- Mitra, N. J. and Nguyen, A., 2004. Estimating Surface Normals in Noisy Point Cloud Data. Special Issue of International Journal of Computational Geometry and Applications 14, pp. 261–276.
- Pauly, M., Mitra, N., Giesen, J., Gross, M. and Guibas, L., 2005. Example-based 3D scan completion. Proceedings of the third Eurographics symposium on Geometry processing pp. 23–32.
- Rabbani, T. and Van Den Heuvel, F., 2005. Efficient hough transform for automatic detection of cylinders in point clouds. ISPRS WG III/3, III/4 3, pp. 60–65.
- Schmittwilken, J. and Plümer, L., 2010. Model-based reconstruction and classification of facade parts in 3d point clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38(1), pp. 269–274.
- Schnabel, R., Wahl, R. and Klein, R., 2007. Efficient RANSAC for Point-Cloud Shape Detection. Computer Graphics Forum 26(2), pp. 214–226.