"MODEL SPELL CHECKER" FOR PRIMITIVE-BASED AS-BUILT MODELING IN CONSTRUCTION

권 순 욱* · 프레데릭 보쉐** · 허 영 기***
Kwon, Soon-Wook · Bosche, Frederic · Huh,Youngki

Abstract

This research investigates a Modeling Spell Checker that, similarly to Word Spell Checker for word processing software, would conform as-built 3D models to standard construction rules. The work is focused on the study of pipe-spools. Specifically pipe diameters and co-planarity are checked and corrected by the Modeling Spell Checker, and elbows are deduced and modeled to complete models. Experiments have been conducted by scanning scenes of increasing levels of complexity regarding the number of pipes, the types of elbows and the number of planes constituting pipe-spools. For building models of pipes from sensed data, a modeling method, developed at the University of Texas at Austin, that is based on the acquisition of sparse point clouds and the human ability to recognize geometric shapes has been used Results show that primitive-based models obtained after scanning construction sites can be corrected and even improved automatically, and, since such models are expected to be used as feedback control models for equipment operators, the higher modeling accuracy achieved with the Modeling Spell Checker could potentially increase the level of safety in construction. Results also show that some improvements are still needed especially regarding the co-planarity of pipes. In addition, results show that the modeling accuracy significantly depends on the primitive modeling method, and improvement of that method would positively impact the modeling spell checker.

Keywords : Merging Objects, Sparse Point Clouds, Workspace Modeling, Construction Automation

1. Introduction

Recent studies conducted by the Construction Industry Institute (CII) have indicated that for a typical $100 million construction project, between $500 000 and $1 million are spent for just keeping track of where things are on the site --typically tens of thousands of items -- and for monitoring the status of construction activities. Additional expenses are directed to the establishment of the state of the infrastructure following the actual construction work. Approximately 2% of all construction work must be devoted to intensive manual quality control and tracking of work completion, including operations involving earthmoving and bulk material handling. Any technology that can reduce this burden and decrease time to sense the state of construction sites will offer a significant competitive edge (Cheok et al., 2000). At the same time, a large number of research studies are conducted with the aim of improving safety and productivity of heavy equipment. These studies focus on the implementation of either semi or fully automated systems. Although semi and fully automation approaches don' t usually focus on the same applications, they are both limited by the shortcomings of existing methods for rapidly and accurately modeling the construction workspace.

Construction sites are usually fast changing scenes and workspace models for real-time applications need to be frequently updated, at least every hour, and with a level of accuracy that permits safe and productive labor. Any technology allowing fast and accurate construction site modeling would be a major step forward in the development and use of such real-time systems. A recently developed method for workspace modeling, using the human ability to recognize targets, has been developed at the University of Texas at Austin (UT) (Cho et al, 2001, Kwon, 2003). This method takes

---

* 정회원, 한국건설기술연구원 선임연구원, 공학박사 swkwon@kict.re.kr
** 학생회원, 미국 텍사스 대학교 토목공학과 연구원
*** 정회원, 미국 텍사스 대학교 토목공학과, 공학박사

advantage of the human ability to recognize geometric forms and distinguish target from non-target objects which can significantly reduce acquisition and modeling processing times (Kwon, 2003, McLaughlin, 2001). Rapid and accurate methods for fitting and matching primitives to "sparse range point clouds" is a major feature of the method (Kwon, 2003). The main advantage of this method is that it requires only a few scanned points to model objects on a construction site so that it is much less computationally intensive than the other approaches. In the rest of the thesis, this modeling method will be referred as the "primitive modeling".

However, the primitive modeling method, that models individual objects with a high level of accuracy, doesn' t address complex construction scenes that include combinations of primitive forms. As an example, the method would not exactly model a structure in concrete, but the numerous beams and columns constituting it. In order to achieve a higher level of accuracy and display the entire structure as one single object instead of an agglomeration of objects, "beams" and "columns", it seems necessary to merge, or combine, the primitives into more complex objects by conforming them to construction material/method related constraints. In addition, the models obtained by primitive modeling are different from those drawn by 3D CAD engines. Since they are obtained after scanning sites using laser range finders, their precision is not optimal as those developed on 3D CAD engines whose dimensions and orientations are perfect. Moreover, workspaces cannot generally be fully scanned because of the presence of scanning obstacles. These problems reveal a critical need for a method analyzing as-built models, and conforming them to some standard rules, so that more reliable results could be provided to automation or as-built modeling applications. This method would check the modeling "syntax" and implement the connection of primitives. It should be able to check standard compliance of a model and its components (primitives), and then reconstitute as-built objects by moving (translating and/or rotating) the primitives and deducing missing parts. This main idea of this tool bears similarity to the concept of "word spell checker" for document editing software (like Microsoft® Word®). The purpose of word spell checker is to correct vocabulary and grammar. While correcting vocabulary consists in checking the spell of each word separately, correcting grammar consists in conforming groups of words, sentences, to standard rules. Similarly, this research proposes a method that would separately check the dimensions of primitives (vocabulary) and the organization of primitives within pipe-spools (grammar) by conforming them to standard constants and constraints. By way of analogy to word spell checker, this proposed method can be called "As-built 3D Modeling Spell Checker".

The modeling spell checker was carried out on the basis of primitive modeling. Its purpose is to correct primitive data acquisition and modeling errors, and recombine primitives by modeling missing parts. The study has been limited to one type of primitives: cylinders. Pipes present a great variety of assemblies constituting a consistent area of application in construction. Therefore, if the results satisfy the expected modeling improvements, it would be justified and easy to develop equivalent methods for the other types of primitives. In accordance with the objectives, the modeling spell checker has been developed in three phases. Methods for (a) correcting objects dimensions, (b) correcting objects positions, and (c) merging objects were successively developed on the Matlab environment. To test the accuracy and conservativeness of the model check method along its development, experiments with increasing levels of complexity regarding the number of cylinders as well as their relative positions were conducted. Tests were first performed computationally through simulations, then on an indoor test bed composed of several types of pipes and elbows, and finally on an outdoor real pipe network scene.

2. General Approach

This work is based on the observation that construction materials and as-built present geometrical particularities that conform to the application of standard constant and constraints. These particularities include dimension and orientation. For instance, pipe diameters conform to standard values, as shown in Table 3.1 below. These standards are used to check the dimensions of the modeled pipes in this work, which corresponds to the use of the vocabulary-checking feature of word spell checkers.

The "merging" part of the work is based on the same assumption. Pipespool elbows are defined by the diameters of the pipes that they connect and their relative angle. Thus, automatically drawing elbows (meaning merging pipes) is possible even without additional information about the elbows. This operation corresponds to the use of the grammar checking feature of word spell checkers. The major problem in this research is that models aren' t absolutely accurate as they would be in architectural CAD drawings. Indeed, due to previously mentioned modeling errors and the fact that primitives are modeled separately, they are hardly ever in correct position to be merged. Fitted cylinders that need to be connected may not fulfill merging criteria (for instance their axes must have an intersection point). Therefore, the issue is to develop a tool, based on heuristics, that automatically and intelligently analyzes groups of pipes, compares the positions and orientations of one to each other, discovers how they are likely to be actually connected, and finally moves them accordingly. Once this aspect is treated, merging them becomes possible.

3. Program Structure Code Description

The Modeling Spell Checker is a modeling tool that aims to correct models obtained by scanning pipe-spools on construction sites. Workspace models are created by following the primitive method developed at UT. Sparse range point clouds are acquired for each pipe by using the laser mounted on the pan and tilt unit that is controlled by a trackball. Then, cylinders are fitted to these clouds. The coordinate positions and dimensions of the modeled cylinders constitute the input of the Modeling Spell Checker. The output is a corrected and improved displayed model. While dimensions and positions of the cylinders are corrected, elbows are deducted and drawn.

The entire modeling tool has been developed in Matlab®. In very general lines, the structure of the Modeling Spell Checker is as follows. The entire program consists of 17 M-files (Matlab®) and one spreadsheet (Excel®) contained in a folder. These 18 files are organized as shown in the organization tree in Figure 1 below. Their step numbers ("step xxx") convey that organization. The files include both acquisition (Step 100) and spell checking processes (Step 200). While the codes used for data acquisition will be briefly presented in section 3.5.1, the spell checker will be presented in details in the next six sections. In order to run the program, the operator has to launch the M-File "Step_000_Scene_Modeling" in the Matlab® environment. The other files are then run automatically, and the operator just has to answer the questions asked during the acquisition process.

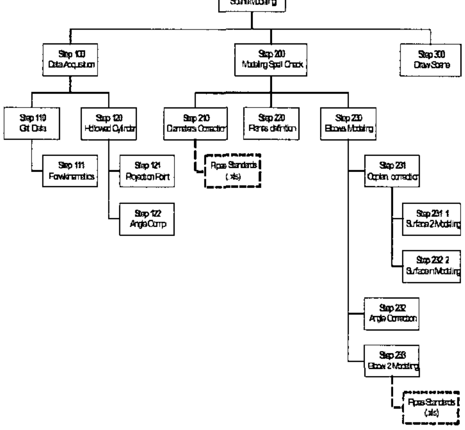


Figure 1. Organization tree of the files used in the whole modeling Program

3.1 Algorithm Description

The developed program runs in six steps. It successively: (1) acquires objects data and fits primitives (primitives modeling work), (2) corrects their diameters, (3) defines coplanar pipes, (4) corrects the co-planarity of pipes, (5) corrects relatives angles, (6) merges pipes, and finally (7) displays the results.

These efficient algorithms allow modeling one cylinder (Steps 110, 111, 120, 121 and 122). However, modeling a pipe-spool requires modeling several cylinders. By using the modeling algorithm in a loop it is possible to model the entire pipe-spools. This implies that the fitted cylinders are modeled separately, and they need to be connected. Consequently, the modeling spell checker needs to know, by using another approach, which primitives are members of the same pipe-spool. The algorithm "Step_100_Data_Acquisition" groups cylinders belonging to common pipe-spools and saves information relating to their network relationships. A process diagram of that algorithm is displayed in Figure 2 below. It is based on the loop previously mentioned allowing the operator to model new cylinders until he or she reaches the end of the pipe-spool. It also includes a step called "Define modeled cylinder it is linked to". Here, the operator has to give the number of a cylinder that has already been modeled.

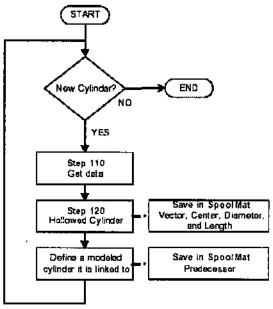


Figure 2. Methods used for the algorithm "Step__100__Data __Acqusition"

These efficient algorithms allow modeling one cylinder (Steps 110, 111, 120, 121 and 122). However, modeling a pipe-spool requires modeling several cylinders. By using the modeling algorithm in a loop it is possible to model the entire pipe-spools. This implies that the fitted cylinders are modeled separately, and they need to be connected. Consequently, the modeling spell checker needs to know, by using another approach, which primitives are members of the same pipe-spool. The algorithm "Step_100_Data_Acquisition" groups cylinders belonging to common pipe-spools and saves information relating to their network relationships (Figure 3). A process diagram of that algorithm is displayed in Figure 3.6 below. It is based on the loop previously mentioned allowing the operator to model new cylinders until he or she reaches the end of the pipe-spool. It also includes a step called "Define modeled cylinder it is linked to". Here, the operator has to give the number of a cylinder that has already been modeled and linked to the one he or she is currently modeling. For simple pipe-spools where every pipe is only linked to one other, this step would be unnecessary since the operator could simply be asked, before starting the acquisition, to model pipes in the order they appear in spools. However, in the case a spool has elbows connecting more than 2 pipes, this tool is very important. Without it, the program would not be able to figure out the way the pipes are organized. So, as example, if the operator is modeling a fifth pipe which is connected to the fourth one, he or she must enter "4" as input to the question "Define a modeled cylinder that is linked to the current one".

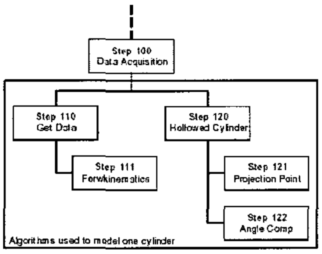


Figure 3. Algorithms tree of the "Step__100__Data__Acquisition"

The result of the step 100 algorithm is a matrix "SpoolMat" including the vectors (in Cartesian coordinates), centers (in Cartesian coordinates), diameters, lengths and predecessors of all the scanned pipes constituting a pipe-spool. The matrix is organized as shown in the Figure 4 below and is the basis of work of the modeling spell checker. Once the acquisition of one pipe-spool is done, the correction starts: step 200 Modeling spell check. This parent algorithm simply launches the algorithms "step 210", "step 220" and "step 230" one after the other. These are described in the next paragraphs and are run automatically without requiring any operator input.

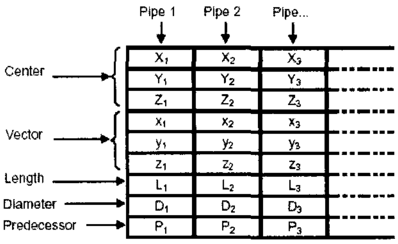


Figure 4. Organization of the SpoolMat matrix

3.2 Diameters Correction

The first step to be implemented is to correct the diameters of the cylinders. This makes sense since, as explained previously, the next steps are all based on pipe diameter standards. The diameter correction process consists of one Matlab algorithm file (Mfile), "Step_210_Diameters _correction", but calls the database containing the pipe standards. The method is explained in Figure 5.

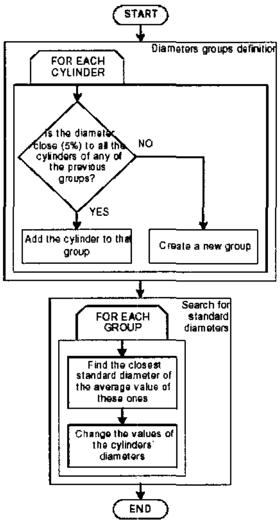


Figure 5. Methods used for the algorithm "Step__200__Diameter__Correction"

The choice for a 5% interval comes from the results obtained in the primitive modeling research. Indeed, the hollow cylinder fitting method has shown a maximum of 5% error in diameter prediction. Therefore, though this method may fail sometimes, it appeared to be more logical and safer to choose the value 5% to define the acceptance level (Figure 6)

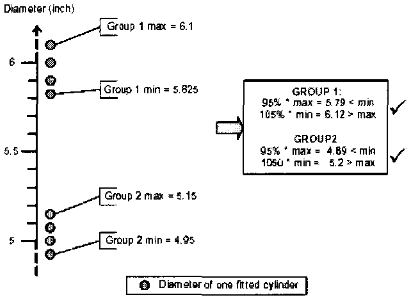


Figure 6. Example of pipe grouping regarding their diameters

3.3 Plane Definition for Correction of Co-planarity

The Problem of Achieving Co-planarity Besides diameter correction, the co-planarity of cylinders is another issue that must be corrected. Experiments implemented in the primitive modeling work show that the cylinder fitting method determines cylinders' axes with deviation errors of 3% maximum. The result at the pipe-spool level is that cylinders will probably not be co-planar, which is a requirement to merge them. For instance, pipes in chemical plants are often not properly aligned in order to avoid stagnant fluids in pipes. Consequently, trying to correct these apparent geometrical imperfections won' t accurately reflect reality and models may appear different from actual pipe-spools. Besides, a requirement that always has to be fulfilled in order to merge two cylinders with a torus is that they must be co-planar. Geometrically, it means that their axes must have an intersection point. Without this compliance, two cylinders cannot be accurately connected. This reason further suggested the implementation of the correction of co-planarity. Thus, another formulation of the problem is to: "define the optimized way to move a series of non co-planar segment in order to correct co-planarity". The problem is even more complicated: as shown in the example in Figure 7, one spool can consist of groups of pipes belonging to different planes. Consequently, the first step to correct co-planarity is to define the groups of pipes that can be considered co-planar. This is achieved by the algorithm "Step_220_Planes_definition" presented below.

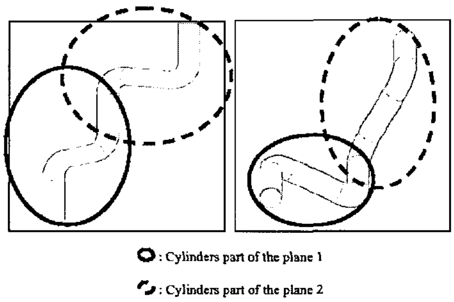


Figure 7. Example of pipe-spool structured on two planes

3.4 Co-planarity Correction

Once the planes are defined, the cylinders' co-planarity and angularity between the pipes can be corrected and the elbows modeled. The method used here is an elbow-by-elbow process presented in the "Step_230_Elbows_modeling" algorithm (See Appendix A.11). Figure 8 displays a more comprehensive process chart. Once two cylinders (C1 and C2) that must be connected are identified, co-planarity and relative angular position are checked, and finally they are merged by modeling an elbow. It must be noticed that, as it will be seen later, step 231 corrects the co-planarity for all the pipes within a pipe-spool.

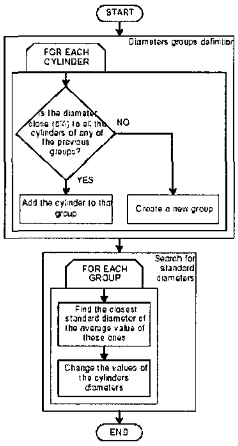


Figure 8. Method used for the "Step_230_Elbows_Modeling" algorithm

The reason for correcting the co-planarity of all the pipes after modeling each elbow is that modeling an elbow requires moving cylinders that may belong to one of the following planes. Afterwards, this modification implies that the planes must be re-optimized.

The first major action is to correct co-planarity. This is done by utilizing algorithm "Step 231_0_Coplan._Correction" (Appendix A.12) that defines the optimal average plane for each group of supposedly coplanar cylinders. Two methods have actually been developed to evaluate an optimal average plane from the primary axes of all pipes constituting a pipe-spool. The existence of two methods (that will be presented afterwards) is related to the number of cylinders constituting the plane (two or more than two cylinders), but is justified by a more complicated problem. As it has been previously used in order to define planes, two consecutive planes always share one cylinder. Therefore, if two optimal planes are defined separately (without referring one to each other),the second optimal plane may require projecting the shared cylinder in a way that it won' t be on the optimal first plane anymore.

The example in Figure 9 geometrically reveals this issue. A solution would be to use the projection of that primary vector on the edge belonging to both Planes 1 and 2. However, this would significantly impact the next step of the process that corrects relative angles. Indeed, by projecting an axis on the edge, all its degrees of freedom are removed, and this freedom will be necessary in the next stage correcting relative angular positions. In order to simplify the problem without impacting the optimization level, planes are processed one after the other and the axis of the shared cylinder is projected on the first processed plane. The optimization of the average

Plane 2, then, has to be recalculated considering the fixed shared vector. This is illustrated in the Figure 10 using the previous example.

The two methods, to evaluate an optimal average plane, study separately the cases where only two or more than two pipes constitute the plane to be optimized, and are presented below. Plane defined by only two cylinders. This corresponds to the "Step_231_1_Surface_2_modeling" algorithm. Since one of the cylinders is fixed by a previously optimized plane, only the second one has to be moved.

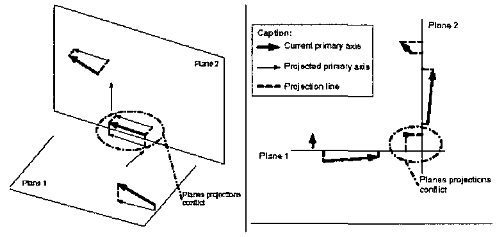


Figure 9. Conflict while separately correcting the co-planarity of the consecutives

The method used to optimize this movement is presented in Figure 11 below. The smallest vector d connecting the two axes of the pipes, V1 and V2, is located (left) and used to translate the second vector V2 to V2' (right). This corresponds to the "Step_231_Surface_N_modeling" algorithm. The method used in the case of two cylinders cannot be used here. While the first vector defining the average plane can be chosen as the fixed vector already belonging to an optimized plane, the second vector is identified using a least squares method applied to the cylinders' centers.

This is done by priory calculating the vector normal of the optimized plane that minimizes the sum of the square of the distances of the centers to the plane. Therefore, the second vector defining the plane is the cross-product of the first vector and the normal. While these vectors define the orientation of the optimized plane, its position is defined by the center of the first cylinder. Then, all the vectors are projected on this optimized plane.

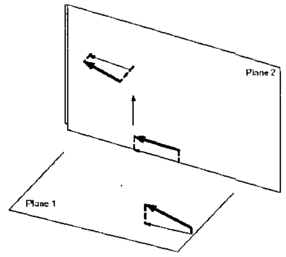


Figure 10. Calculation of an optimal average plane that shares an axis with an already calculated one

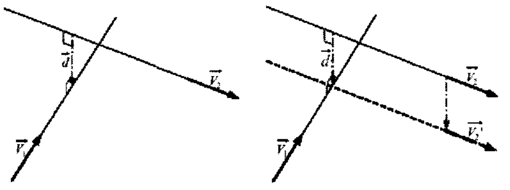


Figure 11. Optimization of a plane defined by only two cylinders

4. Experimental Plan and Results

The first part of this section presents experiments conducted through simple simulation and indoor scanning and discusses how results from these experiments helped improving the method. Finally, two more complex indoor and outdoor tests are described. Their results, which indicate the advantages and limitations and areas for improvement of the proposed method, are thoroughly analyzed in the following sections.

4.1 Initial Experiments Computer Simulation

The simulations were carried out by creating SpoolMat matrices based on CAD models that include only two cylinders. The consistency and strength of the first algorithms checking only the co-planarity and the angularity was tested by changing parameters of the two cylinders such as:

- Relative position of the cylinders
- Relative orientation of the two unit vectors of the two cylinders
- Lengths of the cylinders

Figure 12 displays an example where only the vectors of the cylinders are displayed. Since there are two cylinders, only one translation is needed to correct the co-planarity: the second cylinder's axis is translated by using the vector, corresponding to the shortest distance between the two axes of the cylinders. Then, the second cylinder is rotated so that the relative angle between the two cylinders is 90°.

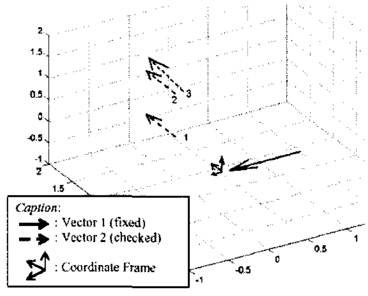


Figure 12. Simulation with two cylinders to check co-planarity (position 1 to 2)

This experiment revealed limitations and brought up two new issues. The first was that the Matlab environment doesn' t include a built-in algorithm to display a cylinder in a random position in space. The same problem appeared with elbows (torus). In this case, no algorithm has been created to address this issue. The second issue had to do with the method for correcting the positions and orientations of cylinders connected at an elbow. Several methods were considered, like calculating the average plane and project both pipes on it, but the one presented in section 3.3.1 seemed to be the most appropriate and was finally used in the following experiments.

4.2 Final Experiments

Though the previous experiments display great results, they didn' t include situations where a pipe-spool consisted of pipes beginning. Therefore, indoor tests were conducted to support and test each step of the development of this feature by handling pipes located on different planes. Once the entire modeling spell check seemed to be adequately consistent, it has been tested on pipe structures in an outdoor site. located on several different planes. Figure 13 shows the experimental setup used in indoor and outdoor tests.

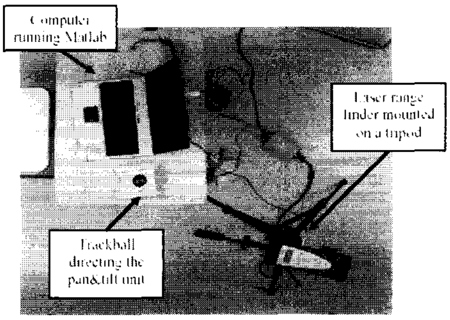


Figure 13. Experimental setup

4.2.1 Indoor Testing

The example displayed in Figure 14 has been tested many times until consistent results were obtained. This example is simple but includes many particularities. It consists of three pipes belonging to two different planes and with two different relative angles (45° and 90°).

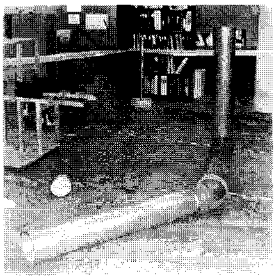


Figure 14. Pipe spool used to test pipes positioned in several different planes

Since the pipes had ratios of length over diameter (L/D) equaling four, the results of the primitive modeling algorithms were consistent. Indeed, while being tested, the primitive modeling method showed more consistency with average ratios than low ones (see Table 1). The results obtained from the modeling spell checker are shown in Figure 15.

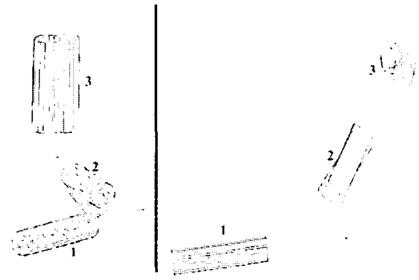


Figure 15. Results from two different views, showing the cylinders before (with spaced stripes) and after (with dense stripes) applying the modeling spell checker

Table 1. Impact of ratio Length/Diameter (L/D) on the cylinder modeling method

	10 scanned pts. per cylinder		
	Percentage of error (%)		
	Radius	Length	Deviation of Axis
L/D = 1.0	11.50	11.00	14.19
L/D = 1.5	4.75	8.67	2.66
L/D = 2.0	4.30	5.35	2.58
L/D = 2.5	3.00	2.86	1.94

4.2.2 Outdoor Testing

As a final and determinant-testing step, outdoor experiments have been conducted. They took place in the surrounding of the UT FSCAL laboratories on the campus of the University of Texas at Austin. A scanned scene, photographed from the position of the laser scanner, is displayed in Figure 16.

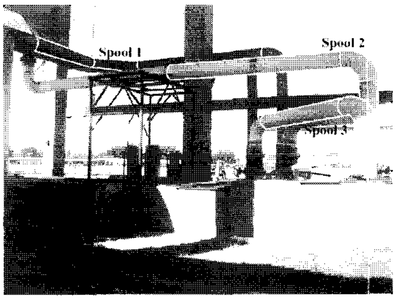


Figure 16. Highlighted scanned pipe-spools in the entire outdoor scene

In one of the experiments, three pipe-spools were scanned, as highlighted in Figure 16. While pipe-spools 2 and 3 consist of only two pipes of eight-inch diameters each, pipe-spool 1 consists of three ten-inch pipes displayed on two different planes. Figure 17 below also displays the final result of the experiment. Algorithmic text relevant of the experiment is also displayed in the Matlab environment.

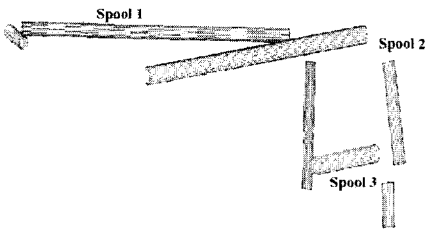
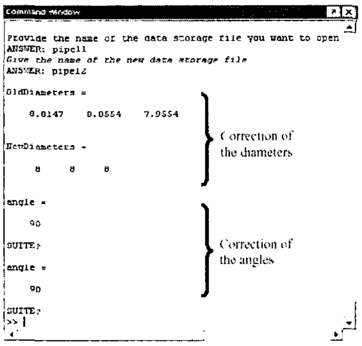


Figure 17. Results applying the modeling spell checker

An example, showing a print screen obtained at the end of the process for pipe-spool 1, is displayed in Figure 18. It shows that the diameters were corrected (Old Diameters to New Diameters) and the two angles corrected to 90°.



```
Provide the name of the data storage file you want to open
ANSWER: pipe11
Give the name of the new data storage file
ANSWER: pipe12

OldDiameters =

    8.0147    8.0554    7.9554


NewDiameters =

     8     8     8


angle =

    90

SUITE?

angle =

    90

SUITE?
>> 
```

Figure 18. Captured screen showing the correction of the diameters and angles

An explanation for these errors has been identified: the quality of accuracy of the equipment had a huge impact on the results. The pan & tilt unit and the tripod supporting have play in them, which affected the accuracy of the sense data acquired from objects located at a great distance from the laser equipment. Indeed, the error due to a small angular deviation in the equipment setup increases as the distance of the equipment from the measured objects increases. Ranges of errors are given in Table 2 and indicate the importance of precise equipment. The current pan & tilt unit and tripod actually showed such instability.

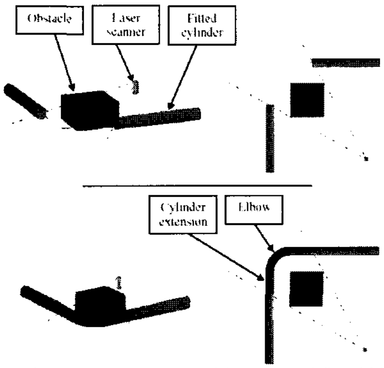
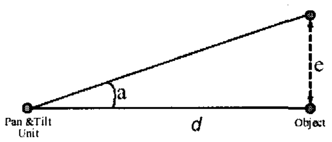


Figure 19. Example of how the method helps modeling hidden parts of a pipe-spool (Up: Model before applying spell checker, Down: Model after applying spell checker)

Table 2. Impact e due to a loose of a in the tripod and pan & tilt unit, modeling results for an object at a distance d from the laser

Loose α (Degree)	Distance d (meter)	Error e (centimeter)
0.2	1	0.35
	5	1.7
	10	3.5
	20	7.0
	30	10.5
0.5	1	0.87
	5	4.4
	10	8.7
	20	17.4
	30	26.2



5. Conclusions

Regarding the correction of diameters, it has been shown in chapter four that the algorithms showed a good consistency even when applied in cases presenting large errors (up to 10% difference with the standard values). Although it is one of the three sub-objectives and doesn't have a great direct impact on the improvement of the quality of the models, it was very critical since the results of the others are very dependant on those ones. This part was even more successful since the choices, made at the beginning of the research, appeared to be the most effective ones. The implementation of the deduction of non-scanned parts, the third objective, also showed great results. This is not very surprising because it didn't require any special development. All the requirements for implementing this part must in fact be met within the second objective.

The correction of the positions of a series of pipes, the second objective, required a lot of attention and time. Correcting simultaneously co-planarity and angularity is a very complex optimization problem. It has not been addressed but an efficient solution has been developed. It processes pipes one-by-one, instead of the entire pipe-spool at one time. The results approach optimized solutions so that it doesn' t significantly reduce the quality of the results. It is acknowledged here that this objective has only partially been met.

In general, it has been demonstrated in this research that an as-built Modeling Spell Checker can be developed to be applied with as-built modeling products. It is very fast (not more than 5% of the time required for the acquisition) and improves the quality of models. Nonetheless, a Modeling Spell Checker cannot completely correct the errors due to the quality of the equipment and the modeling algorithms. If the equipment and the modeling method (fitting method here) create more error than what the modeling spell checker can support, the Modeling Spell Checker may deteriorate the quality of the results. Consistent and accurate acquisition equipment and methods are a major requirement for the use of modeling spell checker.

Acknowledgement

This paper is based on the research funded by the National Science Foundation (Grant#: CMS-0000137) and the National Institute Standard and Technology (Project NBR: NA1341-02-W-0742). The authors gratefully acknowledge their financial support and encouragement throughout this study.

References

1. CII, "Radio Frequency Identification Tagging", RFID Tagging Research Team-Research Summary 151-1, March 2000
2. Cheok, G., Lipman, R., Witzgall, C., Bernal, J., Stone, W., "Non-Intrusive Scanning Technology for Construction Status Determination", National Institute of Standards and Technologies (NIST), 2000
3. Cho, Y., Haas, C. T., Saidi, K., Liapi, K., and Sreenivasan, S. V., "Rapid Local Area Modeling for Construction Automation," Proceedings of 18th International Symposium on Automation and Robotics in Construction (ISARC), Krakow, September 2001
4. Cyra Technologies Inc., 4550 Norris Canyon Road, San Ramon, CA 94583, www.cyra.com
5. Kwon, S., "Human-Assisted Fitting and Matching of Objects to Sparse Point Clouds for Rapid Workspace Modeling in Construction Automation", PhD' s dissertation, Department of Civil Engineering, The University of Texas at Austin, Texas, 2003