Implementation of Automatic Annotation Methods for Revit Piping (Post-Print)

Authors: Li Changhua; Li Tong; Zhou Fangxiao; Cui Huanhuan

Date: 2017-11-06T00:00:00+00:00

Abstract

Based on the practical engineering requirements for Revit pipeline batch automatic annotation, this paper categorizes parallel pipeline annotation into two classes: annotation without diameter change and annotation with diameter change, and expounds upon three conditions that lead to pipeline diameter variation. Corresponding automatic pipeline annotation methods are employed for different pipeline arrangement scenarios. Additionally, this paper proposes a perpendicular line method to resolve the batch annotation challenges for parallel pipelines with multiple diameter changes. Experimental results demonstrate that these methods can rapidly and efficiently annotate pipelines, simplify operations, and improve annotation efficiency compared to traditional annotation methods, thereby providing valuable experience for future secondary development.

Full Text

Vol. 8 No. 1, February 2016

Realization of Revit Pipeline Auto-Labeling Method

Li Changhua, Li Tong, Zhou Fangxiao, Cui Huanhuan
(Xi' an University of Architecture and Technology, Xi' an 710000)

Abstract

Based on the practical engineering requirements for batch automatic labeling of pipelines in Revit, this paper categorizes parallel pipeline labeling into two types: non-diameter-changing and diameter-changing labels, and elaborates on three situations that cause pipeline diameter changes. Corresponding automatic pipeline labeling methods are adopted for different pipeline arrangement scenarios. This paper also proposes a method using perpendicular lines to solve the batch labeling problem for parallel pipelines with multiple diameter changes. Experimental results demonstrate that these methods can label pipelines quickly and efficiently. Compared with traditional labeling methods, they simplify operations, improve labeling efficiency, and provide valuable experience for future secondary development.

Keywords: Revit Secondary Development; Building Information Modeling; Pipeline Labeling

---

Introduction

The rise of BIM technology represents another evolution of computer-aided technology in the construction industry. As modern buildings become increasingly complex and design requirements continuously escalate, Revit' s native functionality can no longer satisfy current architectural design demands, making secondary development imperative. Revit secondary development not only inherits the software' s inherent advantages but also enables the creation of customized plugins tailored to different drafters' needs, thereby significantly improving drawing efficiency.

The Revit API provides two approaches for secondary development: External Command and External Application. External Command inherits from the IExternalCommand class and is loaded through the Add-in Manager, primarily used for implementing specific functional applications. External Application inherits from the IExternalApplication class, characterized by using the OnStartup() and OnShutdown() functions to enable the plugin to launch with Revit and exit when Revit closes. Plugins inheriting from External Application typically involve menu or toolbar development and require corresponding addin files in Revit' s autostart folder.

Pipeline labeling in Revit is essentially the work of intuitively and effectively presenting pipeline diameter information as annotations in mechanical and electrical engineering drawings. Automatic pipeline labeling is an effective means for designers to shorten drawing time when creating MEP drawings, as it enables construction personnel to clearly understand the effective information of each pipeline. However, Revit' s built-in labeling functionality fails to meet designers' requirements in terms of both operational difficulty and aesthetics. Therefore, it is necessary to design a Revit-based automatic labeling plugin to satisfy designers' needs in pipeline annotation. The pipeline automatic labeling method addresses corresponding problems through classes encapsulated in the Revit API provided by Autodesk.

Implementation of Revit Pipeline Labeling Function

A family is a group of elements containing a common set of parameters and associated graphical representations. Autodesk Revit families are mainly divided into three categories: built-in families, system families, and annotation component families. Pipeline auto-labeling families belong to annotation component families, characterized by high customizability. The implementation of Revit pipeline labeling functions is accomplished by loading pre-made families. In family creation, required category parameters can be selected. Generally, pipeline labeling only needs to display pipeline parameters, types, and dimensions. In this paper, the pipe labeling family created includes two category parameters: size and tag.

Classification of Revit Pipeline Auto-Labeling

Pipeline Classification Revit pipeline auto-labeling is mainly divided into two categories: single pipeline auto-labeling and batch auto-labeling for parallel pipelines. Single pipeline labeling is relatively simple and can be completed using Revit' s built-in labeling function. The Revit MEP pipeline labeling discussed in this paper is a plugin development based on External Command with batch auto-labeling capabilities. Practical operations are divided into two scenarios: auto-labeling without diameter changes and auto-labeling with diameter changes.

For pipelines without diameter changes, the batch auto-labeling method obtains appropriate labeling points, projects these points onto the pipelines to acquire distance and leader information for each pipeline, and then uses dimension labeling methods to annotate the pipelines. For batch labeling of parallel pipelines with diameter changes, the perpendicular line method from labeling points to pipelines is used to obtain the current group of parallel pipelines, followed by dimension labeling methods.

Pipeline Diameter Change Conditions Pipeline diameter changes can be categorized into three types: first, changes in pipeline parameters (such as pipe diameter size); second, changes in pipeline arrangement; and third, branch changes in pipelines (such as tees and crosses). Different diameter changes require different judgment methods. The first case can be addressed by using the perpendicular line method from labeling points to pipelines for grouped labeling. For the second and third cases, when directly using labeling points to draw perpendicular lines to pipelines, the labeling points may simultaneously create perpendicular lines to multiple pipelines due to changes in pipeline arrangement direction or the presence of tees/crosses, resulting in labeling confusion (as shown in [Figure 1: see original paper]). Therefore, for these latter two cases, pipelines can be regrouped by obtaining two labeling points on both sides of the pipelines.

Perpendicular Line Method for Multi-Diameter Labeling Problems

During batch labeling of parallel pipelines with multiple diameter changes, labeling leaders may become tilted due to two reasons: first, when projecting from labeling points to pipelines, the projection falls not on the pipeline itself but on its extension; second, when pipelines contain bends, tees, or crosses, projecting from labeling points may simultaneously project onto two or more pipelines, also causing leader tilt. These two issues can be resolved through a grouping and filtering method: first, obtain one labeling point on each side of the parallel pipeline group and draw a line between them without extending it; then group pipelines that intersect with this line, and finally apply the perpendicular line method to solve the leader tilt problem.

For the perpendicular line method discussed in this paper, we define: let P(pointX, pointY) be a labeling point outside the pipeline, A(x1, y1) and B(x2, y2) be the two endpoints of the pipeline, and N(x, y) be the foot of the perpendicular. Then vector PN = (x - pointX, y - pointY), AB = (x2 - x1, y2 - y1), AN = (x - x1, y - y1). Since PN is perpendicular to AB, we have PN • AB = 0, which yields:

$$(x - pointX)(x2 - x1) + (y - pointY)(y2 - y1) = 0$$

Pipeline positions have the following cases:

1. Parallel pipelines are parallel to the x-axis: the foot coordinate is N(pointX, y1).
2. Parallel pipelines are perpendicular to the x-axis: the foot coordinate is N(x1, pointY).
3. Parallel pipelines are neither parallel nor perpendicular to the x-axis: the foot coordinate is N(k(x2 - x1) + x1, k(y2 - y1) + y1), where the slope k is:

$$k = \frac{(pointX - x1)(x2 - x1) + (pointY - y1)(y2 - y1)}{(x2 - x1)^2 + (y2 - y1)^2}$$

For the above cases, this paper proposes a multi-diameter pipeline batch labeling algorithm, detailed in .

TABLE:1 Multi-diameter Pipeline Batch Labeling Algorithm

Input: Labeling points P and Q
Output: Batch pipeline labeling

1. Obtain one labeling point on each side of the parallel pipeline group, P and Q.
2. Draw a line through the two labeling points without extending it.
3. Obtain pipeline information that intersects with the line as a group.
4. Apply the perpendicular line method to find the perpendicular from labeling points to pipelines and draw perpendicular lines.
5. Create leaders through the perpendicular feet and annotate each pipeline sequentially.

Fast Labeling

The fast labeling algorithm is an improvement based on the algorithm in . It calculates the position coordinates of corresponding labeling points through a point on the pipeline, then solves the labeling problem using the perpendicular line method. As shown in [Figure 2: see original paper], the main problem fast labeling needs to solve is obtaining the coordinates of labeling point P(X, Y). Q(x1, y1) and R(x2, y2) are the two endpoints of the pipeline, and O is the labeling position on the pipeline. Let $\frac{QR}{QP} = \lambda$ (0 <   < 1).   can be initialized, and during labeling, the leader length h can be initialized. Then the pipeline and point P can be abstracted as a triangle (as shown in [Figure 2: see original paper]) to solve for the labeling point coordinates.

Based on the discussion above, the pipeline position leads to corresponding analysis:

1. Pipeline parallel to x-axis: labeling point P((1-)x1 + x2, (1-)y1 + y2 + h), (0 <   < 1, h > 0).
2. Pipeline perpendicular to x-axis: labeling point P((1-)x1 + x2 + h, (1-)y1 + y2), (0 <   < 1, h > 0).
3. Pipeline neither parallel nor perpendicular to x-axis: labeling point P((1-)x1 + x2 + h · cos , (1-)y1 + y2 + h · sin), where cos = $\frac{y2-y1}{\sqrt{(x1-x2)^2+(y1-y2)^2}}$ and sin = $\frac{x2-x1}{\sqrt{(x1-x2)^2+(y1-y2)^2}}$.

From the geometric relationship, we can derive:

Let QR · OP = 0 and |OP| = h. From the equations:

$$(x2 - x1)[x - ((1-\lambda)x1 + \lambda x2)] + (y2 - y1)[y - ((1-\lambda)y1 + \lambda y2)] = 0$$

$$[x - ((1-\lambda)x1 + \lambda x2)]^2 + [y - ((1-\lambda)y1 + \lambda y2)]^2 = h^2$$

Through calculation, we obtain the coordinates of labeling point P:

$$x = \frac{h(y2 - y1)}{\sqrt{(x1 - x2)^2 + (y1 - y2)^2}} + (1-\lambda)x1 + \lambda x2$$

$$y = \frac{h(x2 - x1)}{\sqrt{(x1 - x2)^2 + (y1 - y2)^2}} + (1-\lambda)y1 + \lambda y2$$

where (0 <   < 1, h > 0).

A line drawn between points P and O serves as the labeling leader, and the dimension labeling method can then be used to achieve automatic labeling.

Experimental Results and Analysis

Based on the labeling methods provided in this paper, testing was conducted using Autodesk Revit 2014, with code written in C# using VS2010.NET. The main programming steps are: 1) filter all pipelines in the current view; 2) implement batch labeling of parallel pipelines using Algorithm 1; 3) check if finished, otherwise continue with step 2. The overall process is shown in [Figure 3: see original paper].

The test data comes from an actual Revit pipeline labeling project and self-constructed multi-diameter parallel pipeline groups. The batch labeling completion effect is shown in [Figure 4: see original paper].

A comparison between automatic batch labeling and Revit' s built-in labeling efficiency reveals that automatic batch labeling primarily requires manually selecting two labeling points above and below the pipelines. After selection, an entire group of parallel pipelines can be batch-labeled. Revit' s built-in method can only label pipelines one by one, with lower efficiency and neatness compared to the proposed batch auto-labeling method. The fast labeling method mentioned in this paper offers the highest labeling efficiency by eliminating the manual selection of labeling points required in automatic batch labeling. However, its labeling position is uniformly fixed at a certain point on the pipeline, resulting in lower aesthetics and suitability compared to automatic batch labeling. Specific comparisons are shown in and .

TABLE:2 Manual vs. Automatic Labeling Efficiency Comparison

TABLE:3 Manual vs. Automatic Labeling Aesthetics Comparison

Conclusion

This paper primarily elaborates on the implementation method for batch automatic pipeline labeling in Revit, while providing a brief evaluation of the software' s built-in labeling method. Pipeline automatic labeling mainly addresses two scenarios: single pipeline auto-labeling and batch pipeline auto-labeling. Batch labeling can be further subdivided into single pipeline multi-diameter labeling, non-diameter-changing parallel pipeline batch labeling, and diameter-changing parallel pipeline batch labeling. These labeling scenarios essentially cover all current requirements of drawing engineers. For multi-diameter pipeline batch labeling, this paper proposes using the perpendicular line method to group pipelines, thereby solving the problem of tilted labeling leaders.

The Revit secondary development plugin for batch labeling proposed in this paper significantly improves efficiency compared to using Revit's native interactive interface labeling methods. The approach demonstrates clear targeting, makes labeling more intelligent, simplifies operations, and satisfies special requirements of some drawing engineers, thereby shortening project cycles and saving time and costs. This provides valuable reference for Revit pipeline labeling and other application areas.

References

[1] He Guanpei. BIM and BIM-related software[J]. Journal of Information Tech-

nology in Civil Engineering and Architecture, 2010, 2(3): 111-117.
[2] Xu Jian. Research on secondary development of Revit system software[J]. Railway Technical Innovation, 2014.
[3] Lin Jiangfeng. Secondary development of conventional annotations and labels in water conservancy projects based on Revit[J]. Henan Science and Technology, 2015.
[4] Ye Xiongjin. The learning path from beginner to master in Revit secondary development[EB]. CSDN Blog, 2011.
[5] Autodesk Asia Pte Ltd. Autodesk Revit Secondary Development Basic Tutorial[M]. Tongji University Press, 2015(8).
[6] Xiao Ying. Dimension labeling in AutoCAD based on VB[J]. Manufacturing Technology & Machine Tool, 2007(12).
[7] Xu Jian. Research on secondary development of Revit system software[J]. Railway Technical Innovation, 2014(5).
[8] Chen Zhenyu. Feasibility analysis of Revit MEP-assisted design[J]. Fujian Construction Science & Technology, 2012(5).

Realization of Revit Pipeline Auto-Labeling Method

Li Changhua, Li Tong, Zhou Fangxiao, Cui Huanhuan
(Xi' an University of Architecture and Technology, Xi' an 710000, China)

Abstract: Based on the actual engineering requirements of Revit pipeline auto-labeling, this paper divides parallel pipeline labels into two categories of no-adjustable and adjustable parallel pipeline labels and describes three situations which will lead to pipeline diameter changes. In addition, the paper puts forward a method of using lead vertical to solve the problem of the changeable diameter parallel pipes labeling. The experimental results show that this kind of labeling methods greatly increase labeling efficiency, and compared to the use of Revit interactive interface, they simplify the operation, improve the labeling efficiency and provide some experiences for secondary development in the future.

Key Words: Revit Secondary Development; Building Information Modeling; Pipeline Labeling

Note: Figure translations are in progress. See original paper for figures.