Reconstruction of 3D virtual buildings from 2D architectural floor plans

Conference Paper · January 1998

DOI: 10.1145/293701.293704 · Source: DBLP

3 authors:

Clifford So
The Chinese University of Hong Kong

George Baciu
The Hong Kong Polytechnic University

Hanqiu Sun
The Chinese University of Hong Kong

Reconstruction of 3-D Virtual Reality Model from 2-D Architectural Floor Plan

Clifford So and George Baciu
Computer Science Department
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Email: { cliffso , gbaciu } @cs.ust.hk

Abstract

Visualizing architectural constructions with the help of Virtual Reality (VR) technology is becoming more demanding for architects and designers, as it provides a more realistic mean to preview their designs. In order to preview in Virtual Reality, one needs to reconstruct the VR model from the 2-D architectural drawings, but in the conventional method, a huge amount of manual tasks are needed by making use of some common 3-D authoring tools avaliable in the market. Therefore, this gives us a great incentive to examine this VR application in such a practical case. In this paper, we try to clarify the whole conventional reconstruction process in general and give semi-automated solutions to it. Three core streams of tasks in the process are identified: wall extrusion, object mapping, and ceiling/floor reconstruction. We also implement an experimental automation program and go though the whole process again. Results show that the reconstruction process is speeded up by a large portion with the help of our semi-automated solutions comparing to the conventional process.

Keywords: Virtual Reality Application, 3-D Extrusion, 3-D modeling, Architectural Design, Floor Plan

1 Introduction

The application of Virtual Reality (VR) in architectural constructions visualization is becoming vital to architects and designers: it provides a more realistic mean for previewing the constructions such that a plain 2-D drawing cannot provide. To preview their designs in Virtual Reality, a VR model is needed. By the introduction of some common VR authoring tools like Kinetix 3D Studio Viz [?] [?] and Microsoft Softimage 3D [?], designers can reconstruct the VR models from the existing 2-D architectural floor plan drawings, but this usually involves a huge amount of manual processes. Works like building 3-D walls, adding 3-D objects and polishing the interior by texture mapping are common and take a long time.

Having the increasing demand for design visualization in VR, we would like to examine the general conventional VR reconstruction procedures taken by most designers and, most importantly, to see how to automate them. We will start by discussing the general framework of the manual reconstruction process from an existing 2-D architectural floor plan drawing. Afterwards, we will try to give semi-automated methods to the process.

In section 2, we will have a dicussion on the conventional manual process. In section 3, we will propose semi-automated methods to the conventional process. In section 4, we will demonstrate our semi-automated reconstruction process using an existing architectural floor plan drawing. In section 5, we will conclude and comment on our solutions.

2 Conventional Manual VR Reconstruction Process

In the conventional reconstruction of VR model from 2-D architectural CAD drawings, a large amount of manual tasks has to be accomplished in the both CAD and 3-D tools. We have identified, in general, three core streams of tasks in the process. The three streams are: wall extrusion, object mapping and ceiling/floor reconstruction. Wall extrusion is the process of converting the 2-D plan view of the wall structure into 3-D wall model. Object mapping is the process of placing every featured 3-D object in the way that specified in the floor plan. Ceiling/floor reconstruction involves the outlining of the perimeter of the model.

Figure 1 shows an overview of the manual reconstruction process described above, words in boxes indicate tasks and arrows indicates data flow. Tasks in different streams are accomplished in either the CAD tool or the 3-D authoring tool.

The reconstruction process begins with the wall extrusion. Wall polylines are extracted from the 2-D floor plan in the CAD tool and then ex-

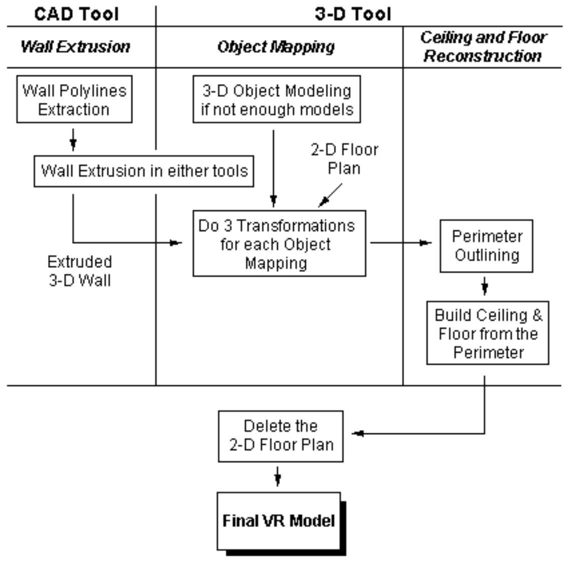

Figure 1: Conventional Manual VR Reconstruction Process

truded in either the CAD or the 3-D tool. In prior to the object mapping, if there are not enough 3-D object models, the designers should create an enough number of them. To accomplish object mapping, designers map every 3-D object with its own transformations. The reconstruction of the ceiling and floor is followed after outlining the perimeter. Note that the 2-D floor plan imported together with the 3-D extruded wall in the 3-D tool in the object mapping process will be deleled before the VR model is finalized, we will explain the need of the 2-D floor plan later.

2.1 Wall Extrusion

Wall extrusion is the process of converting the plan view of the wall structure into 3-D wall model, every CAD tool and 3D authoring tool provides this function. Figure 2 shows the wall extrusion effect. To perform the extrusion in the 3-D tool, we need a clean plan view of the 2-D wall polyline structure ready in the tool. The wall polylines can be imported from the 2-D drawing file and then extruded in the 3-D tool. In this case, we need to extract the wall polylines manually in prior in the CAD tool. And then export the wall polylines to the 3-D tool for extrusion. On the other hand, in the situation that if we want to extrude the walls in the CAD tool, we still need to select all of the wall polylines out of the drawing to perform the action. So we can see that manual extraction of wall polylines is always needed in prior to the wall extrusion in both cases.

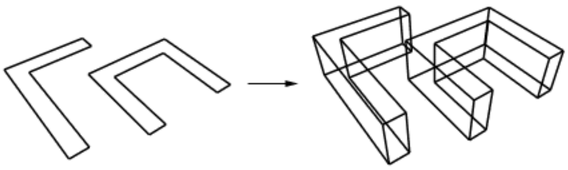

Figure 2: Wall Extrusion effect

However, every CAD drawing exists today is not absolutely well structuralized. Mistakes like loose layer and object definition, placing objects in incorrect layers, redundant objects overlapped between layers, and missing drawings can even be found in current professional architectural drawings. One possible reason is that the designers may only concentrate on the overall drawing outlook, but neglecting its internal data organization. Besides, incorrect input methods, unfamiliarity to the CAD tool and carelessness might also cause problems. The loose structure inside a bad CAD drawing might prevent a direct manual extraction of all the wall polylines, because they maybe scattered between different layers. In practice, designers group all of these wall polylines back to a specific wall layer as extraction.

The wall layer will then be a very effective carrier for the wall structure. When the designer like to perform the extrusion in the 3-D tool, he could simply export the wall layer from the CAD tool and import it to the 3-D tool for extrusion. When the designer like to extrude the wall polylines in the CAD tool, he could simply select the whole wall layer and have all the polylines extruded.

2.2 Object Mapping

Object mapping is the process to place pre-built 3-D object models in the 3-D virtual world in the way that specified in the floor plan drawing. Object models like doors, windows and common furniture are placed in the virtual world at its correct position, orientation and scaling. Every identical object is mapped with the same 3-D model. Figure 3 shows the placing of 3-D object models in the virtual world. The designer should obtain or create enough objects in prior to object mapping.

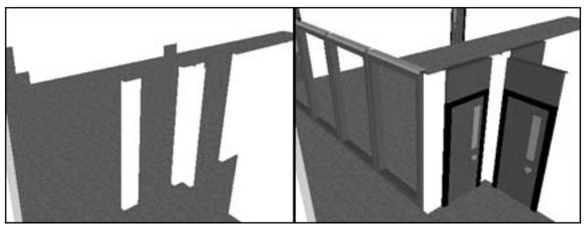

Figure 3: Before and after object mapping

To map the objects manually, in practice, designers would usually import the extruded wall and the 2-D floor plan together in the 3-D authoring tool and overlap them in the plan view. By comparing between the extruded wall and the 2-D floor plan, the correct position of every object with respect to the extruded wall is revealed. The designer could fit a 3-D object model in each 2-D object marking.

However, this manual mapping process requires a tremendous portion of the time, especially when exact positioning has to be followed. Architectural floor plan drawing having over hundreds of objects (like furniture) is common. For each manual object mapping, designers should have to go through three transformations: translation, rotation, and scaling in order to match the different position, orientation and scale of each independent object specified in the floor plan. If an exact positioning is required, the designers have to input the transformation parameters by keyboard entry and that even lengthens the processing time. We estimate in general that the minimum time needed to handle one object mapping for an experienced designer is one minute. So a designer works on only object mapping process for over a few hours is very common.

2.3 Ceiling and Floor Reconstruction

Ceiling and floor are normally not considered in 2-D CAD drawings but contrastably significant to VR models. In some 2-D floor plan drawings, designer may represent the floor region in grid lines. Figure 4 shows an example of the floor grid found in a floor plan drawing. In order to reconstruct the ceiling and floor in the 3-D tool, the designers have to firstly outline the perimeter. He could outline the perimeter by his own understanding of the grid region or any other suitable shape if the grid region does not exist. So the designer will keep the overlapped 2-D floor plan in the 3-D tool until the ceiling and floor are reconstructed. After the floor surface is built, a copy of it with inverse normal is then created at the height of the extruded wall as the ceiling. Designers would also like to apply different texture maps on them to give realistic appearance.

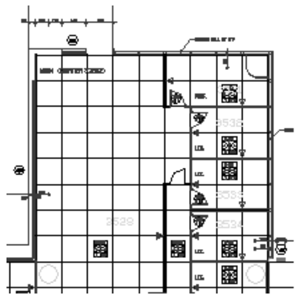

Figure 4: Grid Region of floor

3 Semi-Automated Reconstruction Process

Conventional manual VR reconstruction requires considerable human resources and intervention. In fact, we can develop certain automation functions to replace manual processing, and thus shrink the time in which the designers spent on the CAD and the 3-D tools. Semi-automation solutions can be easily developed from simple computer graphics knowledge and computational geometry. Manual operations will be remained on some preprocessing tasks. We will see that basically the manual preprocessing are to ensure that the input resources are clean and well structuralized.

Figure 5 shows our semi-automated process. It begins with a chain of unavoidable manual preprocessing in the CAD tool: wall polylines extraction, objects reorganization, and perimeter outlining. In the 3-D tool, we also need to calibrate 3-D models to their master object definition (MOD) at the very first time, we will detail on this later. Also we need to provide a list of modeling parameters, such as the extrusion height, the object mapping relationships (MOD name to 3-D model name), ceiling textures, etc, to the automation process. The automation process should be able to reproduce the required VR model after it interprets our 2-D floor plan, 3-D object models and modeling parameters.

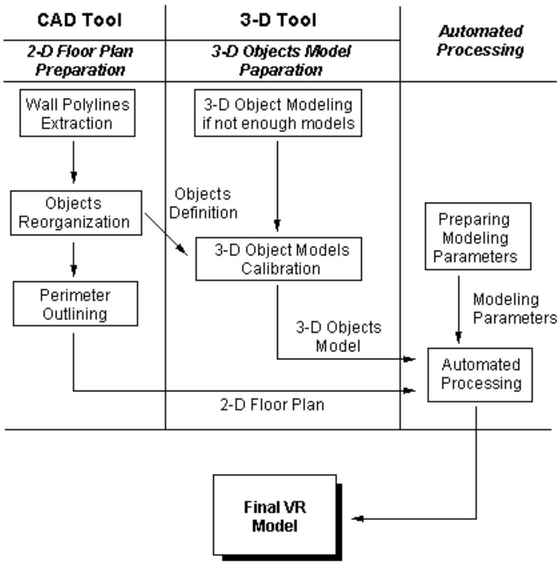

Figure 5: Semi-Automated Reconstruction Process

3.1 Wall Extrusion

As mentioned in the previous section, wall polylines should be extracted first in the CAD tool before actually performing the wall extrusion. We intend to automate the extrusion process, but still leave the extraction of polylines to be a manual preprocessing task.

The reason for leaving the extraction task to be manual is that only the designers know the exact meaning of different misplaced polylines in different layers. In case of the wall polylines are scattered among several unrelated layers, it is highly difficult to extract them precisely with pattern recognition techniques, because wall structure polylines come in complete variety of shapes and dimensions, they are easily to be misclassified to the other objects in the drawings or vice versa. Therefore we do not intend to propose any pattern recognition functions in this paper. Following a good designer practice, designer should extract the wall polylines in prior and grouped into a dedicated wall layer without any other noise drawings.

After we have obtained the completed set of wall polylines, we can extrude them by the following automated method. Let us consider an instance of wall polyline in figure 6, we assume that every polyline is closed, non-intersecting and non-collinear such that it exhibits two properties of a normal wall: 2-sided and non-zero thickness. Let V be the set of the vertices in the polyline, $\{v_0, v_1, ..., v_n\}$. Consider a patch of extruded wall between v_i and v_{i+1}, in order for it to be visible from outside, it should have its normal pointing outward. To specify the normal to point outward, we should output the patch vertices in the order $\{v_i, v_i + \vec{h}, v_{i+1} + \vec{h}, v_{i+1}\}$ if a quadrilateral is needed [?], where $\vec{h}$ is the extrusion vector pointing out of the paper and orthogonal to the floor plan. However, in other scenario that the ordering of the original polyline vertices is anticlockwise as shown in figure 7, all the normals will point inward by following this patch output scheme. So, in prior, we need to ensure that the original ordering is clockwise. If we discover that it is anticlockwise, we should reverse the vertex chain before output to patches.

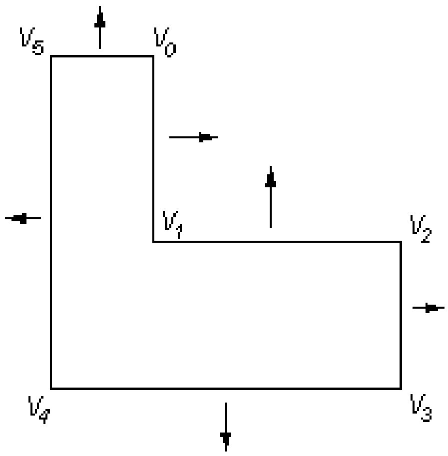

Figure 6: Wall polyline with clockwise vertex order

To check the ordering, let v_k be the rightmost vertex with the maximum y-coordinate in the polyline (For examples, in figure 6, $v_k = v_2$ and in figure 7, $v_k = v_1$.), and calculate $(\overrightarrow{v_k - v_{k-1}}) \times (\overrightarrow{v_{k+1} - v_k})$. Clockwise vertex order always gives negative value and anticlockwise vertex order always gives positive value.

3.2 Object Mapping

The manual object mapping procedure is most time consuming and tedious as we mentioned before and therefore automating this task is mandatory. In fact, we can use basic computer graphics transformation matrices to automate this task. But before entering the automated process, a little manual refinement on the 2-D floor plan drawing and some manual calibrations on the 3-D object models are needed.

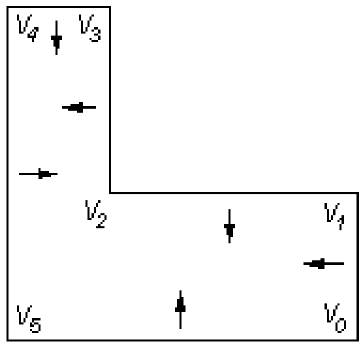

Figure 7: Wall polyline with anticlockwise vertex order

The 2-D floor plan drawing should be object-oriented. We cannot identify any ungrouped drawings which actually composing an object, unless they are grouped and defined as an identifiable single entity. We call this grouped drawings composing the object to be an object definition. We also only need one single 2-D master object definition (MOD) for all identical objects (e.g. same wooden doors, same glass tables). Different MODs will have different names but any other identical objects copied from those MOD will have the name. This is to ensure that every identical object will be mapped with the same 3-D model. Every copied object can have its own positions, orientations and scales specified in the drawing. We leave this object reorganization task to be done in manual because of the same argument in the wall polylines extraction.

We also assume that the initial object center, orientation and dimensions of the plan view of the 3-D object models match to their MOD in the floor plan drawing. An example of a door definition is given in figure 8. This requirement may induce a little manual transformation adjustment of each 3-D model, but this calibration is required to do once, after then, all the automated object mappings will work based on this initial calibration. One may suggest to even automate this calibration of 3-D object models, but figure 8 already provides a good example that the plan view of the 3-D model does not always similar to its MOD. It is also impossible to match the shape if the MOD is even other arbitrary symbol or marking.

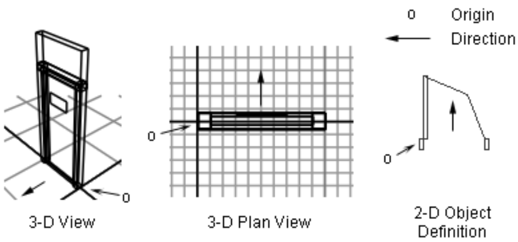

Figure 8: 3-D object model match with its 2-D object definition

Having these two basic manual preparations, we can carry out the object mapping process automatically for each object i by the following simple transformation formula:

$$P'_{ij} = T_i \times R_i \times S_i \times P_{ij} \quad (1)$$

, where P_{ij} is the jth-vertex in the 3-D object model i, P'_{ij} is the transformed new vertex, T_i, R_i and S_i are the translation matrix, rotation matrix, scaling matrix respectively of the object i.

All the transformation parameters are directly read from the 2-D floor plan file. The translation about the z-axis is zero. The rotational axis is along the z-axis. But we need to pay a more attention to the scaling parameter. When we are encountering common objects, we can assume the scaling factor of the z-axis is one. But for the objects, like doors and windows, which totally fill in the space carved from the wall, should be stretched to have the same height as the wall. So the z-axis scaling factor of this kind of objects should be $|\vec{h}|$/(height of the 3-D object model).

3.3 Ceiling and Floor Reconstruction

In the conventional manual process, the desiger needs to outline the perimeter first in the 3-D tool and then reconstruct the ceiling and the floor from it. We choose to automate the reconstruction of the ceiling and floor but remain the outlining of the perimeter to the designer, because floor grid lines are not necessary contained in the floor plan drawings, and we want to provide flexiblity that the floor can extend outside of the interior of the construction. In this case, we require two perimeters (ceiling perimeter and floor perimeter) should be outlined in the CAD tool, in prior to the automated reconstruction of ceiling and floor.

Automated reconstruction of ceiling and floor is equal to generating the trianglated surfaces bound by the perimeters. We can apply the advancing front algorithm[**?**] in this case. This algorithm also supports surface with holes. Figure 9 shows the meshing effect of this algorithm. The algorithm starts by generating internal nodes by a scan-line sweeping along the x-axis. Afterwards, advanced front meshing algorithm is used to connect the internal nodes and the nodes on the perimeter or hole boundary. The result is a surface with internal triangle meshing is formed. We can use a lower internal node density in both x and y direction to decrease the total number of internal nodes in the surface. After the triangle mesh of both perimeters are built, we output the triangle vertices of the floor in anticlockwise order and output the triangle vertices of the ceiling in clockwise order.

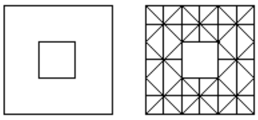

Figure 9: The result of advancing front meshing of a square with a hole

3.4 Modeling Parameter Preparation

We need to inform the automation process the essential modeling parameters and the location of the resources. Core parameters like wall extrusion height, object mapping relationships are mandatory. Location of the resources like wall layer numbers, perimeter layer number, external paths to floor and ceiling texture image should also be included.

4 Results

An experimental program was implemented having all the automation functions described above. We choose AutoCAD DXF as the 2-D floor plan input format and VRML 2.0 as the 3-D model output format as they are widely commonly used. To demonstrate our semi-automated process, we are going to convert our department floor plan with the help of the automation program. This section will describe the actual procedures we have performed.

4.1 Manual Floor Plan preprocessing in the CAD tool

The original AutoCAD floor plan is shown in figure 10. We begin by manual wall polyline extraction. Several wall structure layers originally exist in the drawing are identified. But a few misplaced polylines are also discovered in different layers, we reassociate them back to the wall layer. The overall wall layer is shown in figure 11.

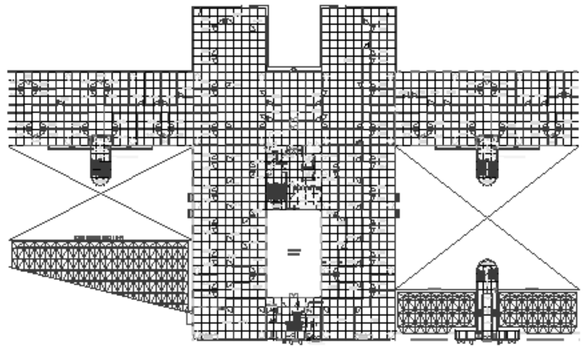

Figure 10: The original department 2-D floor plan drawing

Seven kinds of objects are defined in the floor plan drawing, such as doors, windows, lift and stair. We record the center, orientation, and size of each MOD for further 3-D object model calibration. Also, we reassociate some meaningful ungrouped object drawings back to MODs or copies of another MODs for object mapping propose. An example is shown in figure **??**. The left side shows a window's original shape. However in the middle, it shows that the window is actually drawn by separated polylines (indicated by the highlights of the vertices on the polylines). So we group them to form a MOD of this window (provided that there is originally no other MOD having the same shape), which shows at

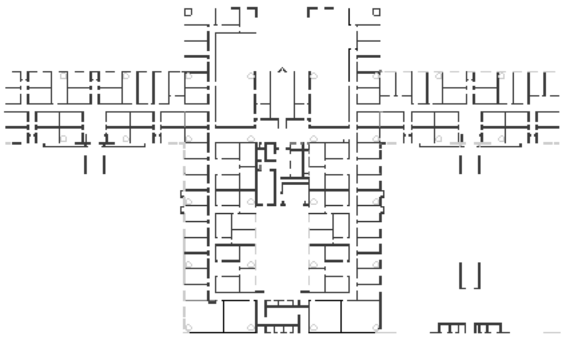

Figure 11: The overall wall layer of the department floor plan

the right side. For the other ungrouped drawings which also describing the same window, we delele them and replace them with the copies of this window's MOD.

We also outline the floor and ceiling perimeters by following the grid region and the overall wall structure. The trianglated floor and ceiling surfaces will be generated automatically with the advancing front algorithm later.

Generally, the original AutoCAD drawing is already in good structure and not much extra manual correction has to be done. The total time needed in the experiment is less than an hour. The whole 2-D floor plan is then exported to DXF file format and ready to be interpreted by the automation program.

4.2 3-D Object Modeling and Calibration

For the seven kinds of the objects, we obtained one 3D Studio MAX model for each. We calibrate them with the center, orientation and scale information recorded from the AutoCAD. We compare the plan view of the 3-D model to its

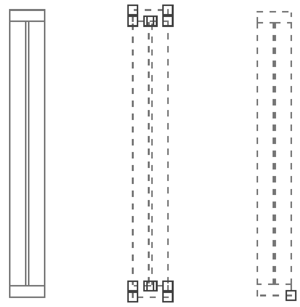

Figure 12: The original window drawing is at the left side. The middle window shows that it is actually composed of seperated polylines. The right window shows it is grouped and defined as a MOD

MOD. With reference to figure refMatch, the original door 3-D model is shown in the left side of figure refCalibration, it is calibrated and transformed to the correct position showing at the right side. Each model is then exported to separate VRML 2.0 format file and is ready for further automatic object mapping. We take 5 minutes for each model calibration.

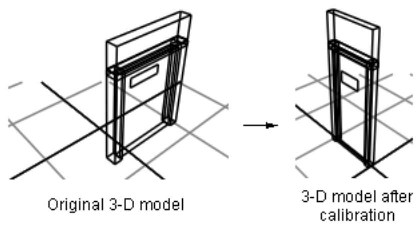

Figure 13: The original door model at the left are calibrated to its MOD shown in figure ref-fig:Match. The door model at the right side is the calibrated one.

4.3 Modeling Parameter Preparation

Parameters like wall extrusion height, object mapping relationships, wall layer number, perimeter layer number are included in the parameter file. We use 5 minutes to fill in the information in the parameter file and save it for the interpretation of the automation program.

4.4 Final VR Model

Automation process go smoothly and it transform the department floor plan (about 300 wall polylines, 260 objects, and 320 perimeter vertices) in 5 seconds using a Pentium Pro processor-based computer. So the whole semi-automated process time including the manual preprocessing work is about half an hour. While for the conventional manual process, we estimate at least an extra 5 hours is needed to manually map the 260 objects by an experienced designer in 3D Studio MAX, even we neglect the manual wall extrusion and ceiling/floor reconstruction time in the 3-D tool. We even expect that much more follow-up rechecking have to be done to ensure that no manual errors are hidden in the whole model.

The VRML model is successfully generated and viewed by a VRML browser, which is shown in figure 14, reffig:Final2 and reffig:Final3.

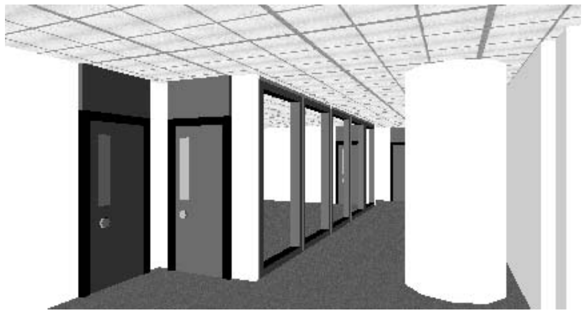

Figure 14: Snapshot 1 of the department VR model

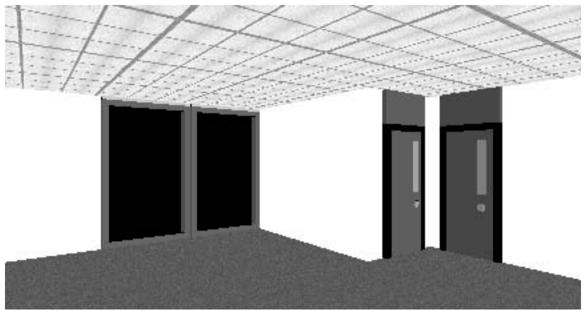

Figure 15: Snapshot 2 of the department VR model

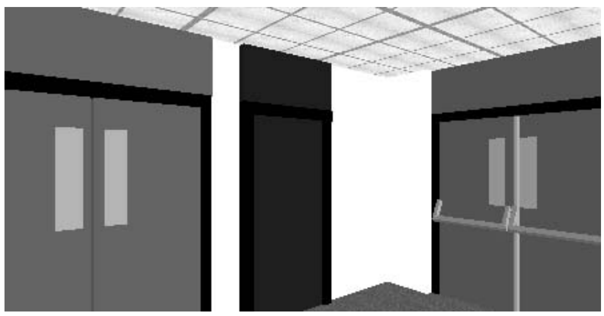

Figure 16: Snapshot 3 of the department VR model

5 Conclusions

We have reviewed the conventional manual process of VR model reconstruction and gave certain ideas to semi-automate the three internal different streams of tasks: wall extrusion, object mapping and ceiling/floor reconstruction. By experimenting with our department floor plan, we have seen that apart from the manual preprocessing time, we have nearly shrunk the processing time of the automated tasks into instant. The original time spent on the 3-D tool for object mapping is eliminated, and hence we saved a lot of labor in dealing with the 3-D tool.

For our solutions, we conclude that we should still leave some preprocessing tasks to be done in manual. Due to the variety of objects drawn in different shapes and dimensions in the drawing, no single generic error-free pattern recognition engine can be developed to extract wall polyline information or to calibrate the 3-D object models to their MODs. Fortunately, most of the CAD drawings exist today are quite well structured, such as the department floor plan we have experimented. Cleaning up and structuring the CAD drawings would not be a long and difficult task for an experienced CAD designer.