

Received 22 November 2022, accepted 10 December 2022, date of publication 15 December 2022,
date of current version 3 February 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3229777

RESEARCH ARTICLE

Study on Construction of Three-Dimensional Interaction Virtual Reality of Corridor Based on Computer Vision

XIAOFAN LIU^{1,2}, JINYE WANG³, SHENGXIN HUANG⁴, TINGTING ZHANG^{2,5}, AND JINJIN LU²

¹Baku Technology Studio, Nanning College of Technology, Guilin 541004, China

²School of Civil Engineering, Nanning College of Technology, Guilin 541004, China

³College of Plant and Ecological Engineering, Guilin University of Technology, Guilin 541004, China

⁴Guangxi Water and Power Design Institute Company Ltd., Nanning 530000, China

⁵College of Arts, Guilin University of Technology, Guilin 541004, China

Corresponding author: Jinye Wang (2005010@glut.edu.cn)

This work was supported in part by the Major Research Projects of the Nanning College of Technology under Grant KYZ202201 and Grant KY202101, and in part by the Young and Middle-Aged Teacher in Guangxi Universities Scientific Research Basic Capacity Improvement Project under Grant 2021KY1670.


ABSTRACT The current methods of construct three-dimensional interaction virtual reality include Photogrammetry, Point Cloud Modeling and Building Information Modeling. However, due to the limitation of space size, these methods cannot construct three-dimensional interaction virtual reality modeling effectively, like corridor. In this paper, the method is proposed to construct corridor three-dimensional interaction virtual reality. The method based on computer vision includes data acquisition, texture stitch, path selection, and model construction. In the process of acquiring data, firstly, we obtain the perspective transformation parameters by pre-establishing the check field. Secondly, we correct the acquired data in real-time by using perspective transformation. In the process of stitching texture, according to the number of feature points in the images, we use feature/direct stitch to generate texture. In the process of selecting path, we identify the targets in image and build buffers. The absolute error and root mean square error are used to measure the accuracy of corridor three-dimensional interaction virtual reality. The experimental result shows that the absolute error of the model constructed in this paper is about 0.0507 to 0.1691. And the root mean square error is about 0.1203 to 0.1318.

INDEX TERMS Construct three-dimensional interaction virtual reality, corridor, check field, feature points, path selection, real-time.

I. INTRODUCTION

With the advent of the digital age, three-dimensional interaction virtual reality is needed in many aspects, such as digital architecture, digital landscape, digital city, etc. [1]. three-dimensional interaction virtual reality of narrow space also has a huge demand [2]. The corridor three-dimensional interaction virtual reality is being mentioned as an important part of digital architecture increasingly.

For this reason, the researchers have made a lot of efforts to construct three-dimensional interaction virtual reality.

The associate editor coordinating the review of this manuscript and approving it for publication was Wenming Cao .

Construct three-dimensional interaction virtual reality can be categorized into three types of methods: The model construction based on using photogrammetry [3]. The model construction based on using point cloud modeling [4]. The model construction based on using Building Information Modeling [5].

Construct three-dimensional interaction virtual reality based on photogrammetry is mostly performed by drones/cars to carry sensors. Therefore, the method is suitable for us to construct the model in large-scale-area, such as digital city model, etc. [6]. The method includes data acquisition [7]; Automatic extraction of three-dimensional information of elements (including contour lines and elevation); Model

construction based on elevation and coordinates [8]. The study on photogrammetry included three main aspects: the clarity of the scene texture, the addition of elevation data, and the geometric repair with the model [9]. Zhou et al. proposed to use photogrammetry for three-dimensional interaction virtual reality construction, a local-to-global radiometric balance is needed to improve the scene clarity of the model [10]. However, the effect of radiometric balance is affected by the image resolution. When the image has high resolution, the arithmetic of method will increase, and the visual effect will be reduced accordingly [11]. Therefore, this method is not suitable for applications with high-precision sensor. Nex et al. proposed to use Unmanned Aerial Vehicle for aerial triangulation encryption in the process of constructing three-dimensional interaction virtual reality. The Triangulated Irregular Network model will have more reference points and is used for the digital elevation model [3]. However, on the one hand, this method is affected by the image resolution; on the other hand, it is impossible to construct three-dimensional interaction virtual reality in a narrow space [12]. Wu et al. proposed a three-dimensional-assisted editing system to construct a white model. The method stretches the entity in image and repairs the white model to make three-dimensional interaction virtual reality with high accuracy [13]. However, this method belongs to optimization after modeling. Therefore, the method cannot be realized in the modeling process.

Using point cloud data to construct three-dimensional interaction virtual reality is suitable for small-scale scenes, such as items [14]. Firstly, the method constructs the white model by using point cloud data. Secondly, the textures are pasted on the white model. The Light Detection and Ranging emits laser and receives back the point cloud data to construct a white model based on the bit depth [15]. Carvalho et al. proposed the advantage of this method is that the laser reception is faster and can be used for real-time modeling. The disadvantage of this method is that the Light Detection and Ranging machines are expensive [16]. Therefore, the method is too costly for narrow corridors modeling which do not require high accuracy. In the process of building three-dimensional interaction virtual reality by using point cloud data, the researchers did not make changes to data acquisition style, but mainly focused on how to make the model obtain texture. Niu et al. proposed to perform object recognition by using Region-Convolution Neural Network to distinguish the target from the scene. So that, the background information in three-dimensional interaction virtual reality is eliminated [17]. However, this method relies on training feature data and cannot effectively perform recognition in a scene with few feature like a corridor [18].

Using Building Information Modeling to build three-dimensional interaction virtual reality is applicable to buildings. Building Information Modeling transforms building entities into data and build three-dimensional models [19]. The method constructs the white model by inputting data from the researcher and merging the data

through photogrammetry or three-dimensional laser scanning as described above for the purpose of modeling. Building Information Modeling is less efficient in building three-dimensional interaction virtual reality. However, it is more inclined to model management and updating [20].

Above of all, modeling based on the photogrammetry has the advantage of texture realism. However, firstly, this method is suitable for constructing large-scale scenes three-dimensional interaction virtual reality. Secondly, the real-time coordinates are difficult to acquire in the corridor. Modeling based on using point cloud data has the advantages of fast data acquisition and high construction accuracy. However, using Light Detection and Ranging to acquire points cloud data has some disadvantages like being too expensive and small areas for a single acquisition. Therefore, the method is not suitable for constructing corridor three-dimensional interaction virtual reality. Modeling based on using Building Information Modeling has some advantages of easy operation and allows for model management and updates. However, this method is less efficient in building three-dimensional interaction virtual reality.

To solve the above problems, this paper proposes the method to construct three-dimensional interaction virtual reality of corridor based on computer vision. The main contributions of our study are summarized as follows.

1. This paper proposes a method to construct three-dimensional interaction virtual reality which can directly construct the model with Level of Detail 200 (LOD 200) accuracy. The method compare with the traditional Building Information Modeling has more efficiency and higher texture clarity.
2. The method proposed in this paper compare with photogrammetry can automatically realize the route selection, which does not need to plan the route in advance. And there is no requirement for elevation data to construct the corridor model.

II. THREE-DIMENSIONAL INTERACTION VIRTUAL REALITY CONSTRUCTION FOR THE CORRIDOR

The proposed method in this paper is achieved by acquiring data, perspective transforming, stitching texture, constructing buffers, selecting route and building model. In the process of stitching texture, we were used both feature stitch and direct stitch to produce texture due to the feature information in the images are uncertainty. In the process of constructing buffers, we were used scene information as the object to build line buffers/polygon buffers to achieve route selection. The flowchart of construct the corridor three-dimensional interaction virtual reality is shown in Fig. 1. It should be noted that considering the complexity and specificity of the three-dimensional interaction virtual reality, the proposed corridor in this paper is defined as a regular type corridor (approximate to a type like cube).

A. PERSPECTIVE TRANSFORMATION

The projection of the image taken by the sensor is central. The central projection will result in distortion of the image [21].

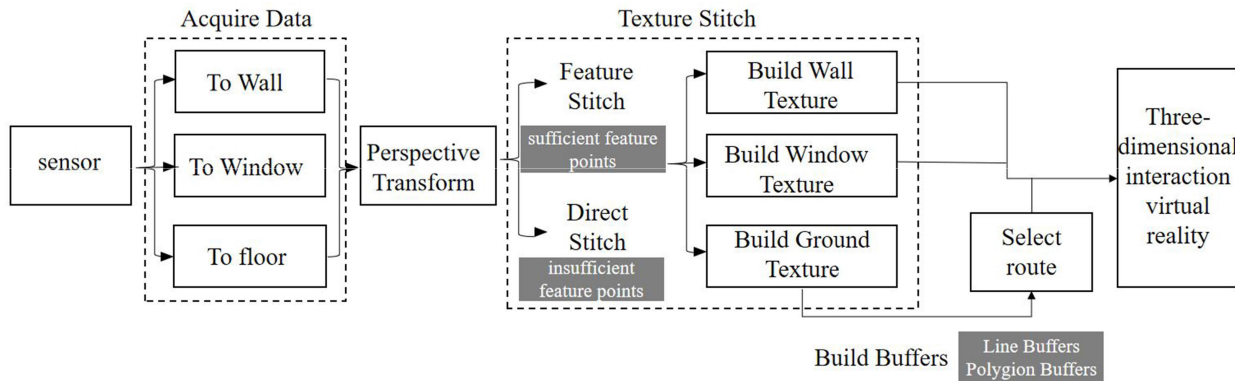


FIGURE 1. The flowchart for constructing the corridor three-dimensional interaction virtual reality.

However, the texture of the three-dimensional interaction virtual reality needs to minimize the distortion. Therefore, we use the perspective transformation to correct the image.

Step 1 (Rigid Transformation): The rigid transformation contains the rotate and translate.

Set (x, y) represents the pixel coordinate in image coordinate system, the image is rotated by an angle of α . The coordinate after rotating can be expressed as follows.

$$\begin{aligned} x_1 &= x \times \cos \alpha + y \times \sin \alpha \\ y_1 &= y \times \cos \alpha - x \times \sin \alpha \end{aligned} \quad (1)$$

where (x_1, y_1) represents the coordinate obtained by (x, y) after rotating.

Further, (x_1, y_1) is translated by a units in the X-axis direction and b units in the Y-axis direction of the image coordinate system. The pixel coordinates after translating can be expressed as:

$$\begin{aligned} x_2 &= x_1 \pm a \\ y_2 &= y_1 \pm b \end{aligned} \quad (2)$$

where (x_2, y_2) represents the coordinate of (x_1, y_1) after translating.

Step 2 (Affine Transformation): The affine transformation is used to correct the image distortion. The affine transformation is shown in the following equation and is solved using the least squares.

$$\begin{aligned} x_3 &= f_1 \times x_2 + f_2 \times y_2 \\ y_3 &= f_3 \times x_2 + f_4 \times y_2 \end{aligned} \quad (3)$$

where (x_3, y_3) represents the coordinate of (x_2, y_2) after affine transforming, $f_1, f_2, f_3,$ and f_4 represent the affine parameters. It should be noted that the affine parameters can be inferred from the coordinates before and after the affine transforming.

Step 3 (Scaling Transformation): The scaling transformation makes the real distance of (x_3, y_3) in the row and column directions the same.

Set the scaling parameters of the image as r_1 and r_2 . Then r_1 and r_2 can be expressed as follows.

$$r_1 = s_0/m$$

$$r_2 = \frac{s_0/m}{s_1/n} \times r_1 \quad (4)$$

where s_0 represents the actual distance in the row direction within the range shown in the image, s_1 represents the actual distance in the column direction within the range shown in the image, m and n represent the size of the image.

According to (3) and (4), the coordinate after scaling can be expressed as follows.

$$\begin{aligned} x_4 &= x_3 \times r_1 \\ y_4 &= y_3 \times r_2 \end{aligned} \quad (5)$$

where (x_4, y_4) represents the coordinate of (x_3, y_3) after scaling.

Combining (1) to (5), the (x, y) after perspective transforming is shown in (6).

$$\begin{aligned} x_4 &= r_1 [f_1(x \cos \alpha + y \sin \alpha \pm a) + f_2(y \cos \alpha - x \sin \alpha \pm b)] \\ y_4 &= \frac{s_1/m}{s_2/n} r_1 [f_3(x \cos \alpha + y \sin \alpha \pm a) \\ &\quad + f_4(y \cos \alpha - x \sin \alpha \pm b)] \end{aligned} \quad (6)$$

From (6), the coefficients of perspective transformation are $a, b, f_1, f_2, f_3, f_4, r_1$.

According to the least squares, at least 4 pairs of control points are required in the image for the solution. We construct the check field and determine the pose parameters, so that each image taken by the sensor can be solved continuously.

B. TEXTURE STITCH

The successive images taken by the sensor are stitched together to produce textures. Due to the different number of feature points in the images, we use both feature stitch and direct stitch to produce textures.

1) FEATURE STITCH

In the process of feature stitching, firstly, is necessary to extract feature information from adjacent images respectively. The extracted information includes the outline or point features of the image.

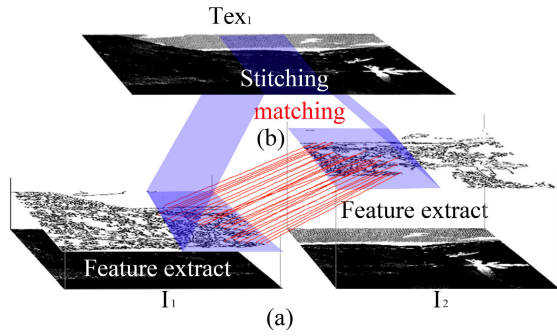


FIGURE 2. The principle of feature stitching. (a) feature extraction is performed on the image (b) texture is produced by adjacent images feature matching and stitching.

Secondly, the features points are matched, and the images are stitched together to produce textures. The principle is shown in Fig. 2.

As shown in Fig. 2, we refer to the texture as Tex_1 and the images as I_1 and I_2 . Combine with Fig. 2(a), adjacent images are matched by using the computational theory of edge detection by Canny [22].

The computational theory of edge detection by Canny achieves feature extraction by computing the Mag_I and direction $\theta_{i,j}$ of the gradient. Specifying leftward and downward of point $I(x_4, y_4)$ in I_1 as the positive direction. the gradient direction can be shown by the following equation [23].

$$\theta_{i,j} = \begin{cases} \theta' & \theta' \in [0, 2\pi] \\ \theta' + 2n\pi & \theta' \notin [0, 2\pi] \end{cases} \quad (7)$$

where $\theta_{i,j}$ represents the gradient direction of $I(x_4, y_4)$.

And:

$$\theta' = \begin{cases} \frac{3}{2}\pi & y_4 > 0, x_4 = 0 \\ -\arg \operatorname{tg} \frac{Y}{X} & x_4 > 0 \\ \frac{\pi}{2} & y_4 < 0, x_4 = 0 \\ \pi - \arg \operatorname{tg} \frac{Y}{X} & x_4 < 0 \end{cases} \quad (8)$$

where,

$$X = [I(x_4 + 1, y_4) - I(x_4 - 1, y_4)]/2, \\ Y = [I(x_4, y_4 + 1) - I(x_4, y_4 - 1)]/2, \\ I(x_4, y_4) \text{ from (6) to obtain.}$$

The Mag_I can be expressed as follows.

$$Mag_I = \sqrt{X^2 + Y^2} \quad (9)$$

Since the similarity of feature points in adjacent images are determined by both the gradient direction and the gradient value. Therefore, we introduce two indexes $S\theta$ and SM . $S\theta$ and SM are expressed them as follows [24].

$$S\theta = |\theta_p - \theta_q| \\ SM = \frac{|Mag_p - Mag_q|}{Mag_p + Mag_q} \quad (10)$$

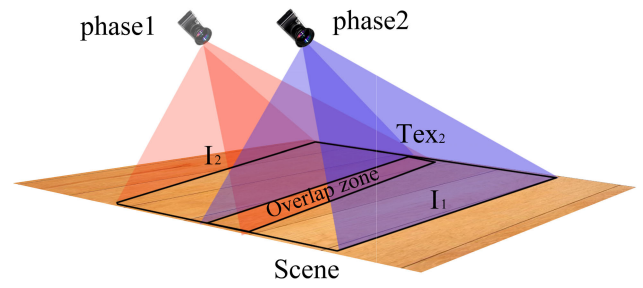


FIGURE 3. The principle of direct stitching to produce texture.

where, when the value of $S\theta$ and SM is smaller, the higher matching of feature points is achieved.

We name the texture after stitching by (10) as Tex_1 , and set the size of Tex_1 is $M \times N$ ($M \in [0, 2m), N \in [0, n]$).

2) DIRECT SITICH

Feature stitch is suitable for stitching images with sufficient feature points. If there are few feature points, feature stitch cannot be achieved effectively. Further, we propose direct stitch. Direct stitch is based on the motion state of the sensor and the time of shoot to achieve image stitching. The principle is shown in Fig. 3.

As shown in Fig. 3, set the interval time between phase 1 and phase 2 as t , the running speed of the sensor as vic , the distance parameters of the image are corrected by (6) as A , then the length of the overlapping zone can be calculated by the following equation.

$$k = A \times vic \times t \\ A = \frac{s_0}{p_0} \quad (11)$$

Where k represents the length of the overlapping zone, p_0 represents the pixel distance.

The overlapping zone after stitching can be expressed as follows.

$$\begin{cases} ro \in [0, k] \\ co \in [0, n] \end{cases} \quad (12)$$

where ro represents the rows in the overlapping zone, co represents the columns in the overlapping zone.

The texture after stitching can be expressed as follows.

$$Tex_2 = \begin{bmatrix} I_2 \\ I_1 \end{bmatrix}_{\substack{(m-s) \times n \\ (m-k) \times n}} \quad (13)$$

where, Tex_2 represents the texture after direct stitching, I_1 represents the image taken by the sensor in phase 1 and I_2 represents the image taken by the sensor in phase 2. Set the size of Tex_2 is $M \times N$, and $M = 2m - k, N = n$.

When the sensor is moving by homogeneous motion, the texture after stitching by two methods satisfies the following equation.

$$Tex_1 = Tex_2 \quad (14)$$

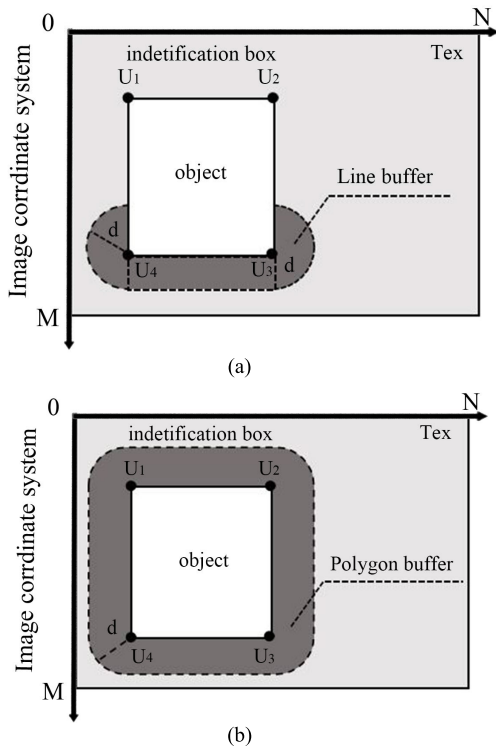


FIGURE 4. Two types buffers. (a) Line buffer. (b) Polygon buffer.

In summary, we refer to the textures as *Tex* collectively. It should be noted that we need to stitch the window, wall, and floor images respectively to produce the corresponding textures.

C. ROUTE STUDIES BASED ON TEXTURE

The route select is performed in the floor texture. All targets which affecting the passage are generate the corresponding identification boxes. According to the type of targets, the buffers are built, and the route is selected by the buffer radius. The main research in this paper is building buffers and selecting route, therefore, it is not elaborated in target recognition.

1) BUFFER ESTABLISHMENT

Set the size of the identification box in *Tex* is $u \times v$, and $u \in [0, M], v \in [0, N]$.

We refer to the corner points coordinate of the identification box as U_1, U_2, U_3, U_4 respectively. In order to avoid the object effectively, it is necessary to judge the class of the object and create the line buffer or polygon buffer on the identification box [24]. The two types buffers is shown in Fig. 4.

In Fig. 4, the white area represents the identification box area, the black points represent the nodes of the identification box, the dark gray area represents the buffer, and d represents the buffer radius.

According to Fig. 4(a), we choose U_3 and U_4 to build the line buffer. Then the $\overline{U_3U_4}$ function can be expressed by the

following equation.

$$v = v_4 \quad \text{and} \quad u \in [u_3, u_4] \quad (15)$$

where (u_3, v_3) and (u_4, v_4) represent the coordinates of U_3 and U_4 , respectively.

The bottom boundary of $\overline{U_3U_4}$ can be expressed as follows.

$$v = v_4 + d \quad \text{and} \quad u \in [u_3, u_4] \quad (16)$$

The buffers at (u_3, v_3) and (u_4, v_4) can be expressed as follows.

$$\begin{aligned} (u - u_3)^2 + (v - v_3)^2 &= d^2 \\ v &\in [v_3 - d] \\ (u - u_4)^2 + (v - v_4)^2 &= d^2 \\ v &\in [v_4 + d] \end{aligned} \quad (17)$$

It is important to note that the line buffer is not required to consider the upper boundary.

According to Fig. 4(b) and combing (15) to (17), the boundary function of the identification box in the polygon buffer can be expressed as follows.

$$\left\{ \begin{aligned} v &= v_4 + d & u &\in [u_3, u_4] \\ v &= v_2 - d & u &\in [u_1, u_2] \\ u &= u_1 - d & v &\in [v_1, v_3] \\ u &= u_4 + d & v &\in [v_2, v_4] \\ (u - u_1)^2 + (v - v_1)^2 &= d & v &\in [v_1 - d, v_1] \\ (u - u_2)^2 + (v - v_2)^2 &= d & v &\in [v_2 - d, v_2] \\ (u - u_3)^2 + (v - v_3)^2 &= d & v &\in [v_3, v_3 + d] \\ (u - u_4)^2 + (v - v_4)^2 &= d & v &\in [v_4, v_4 + d] \end{aligned} \right. \quad (18)$$

where (u_1, v_1) and (u_2, v_2) represent the coordinates of U_1 and U_2 respectively.

2) ROUTE SELECTION

Route selection includes route selection for single object and route selection for multiple objects.

In the process of route selecting for single object: Set the height of the sensor is h and the shoot angle is β . The actual distance s_2 from the shoot blind to the image width center can be expressed as follows.

$$s_2 = \tan \beta \times h \quad (19)$$

Firstly, set the pixel of the *Tex* bottom edge center to be U_0 . Combing Fig. 4(a), the distances from U_0 to U_3 and U_4 can be expressed as follows.

$$\begin{aligned} \overline{U_0U_3} &= \sqrt{(u_0 - u_3)^2 + (v_0 - v_3)^2} \\ \overline{U_0U_4} &= \sqrt{(u_0 - u_4)^2 + (v_0 - v_4)^2} \end{aligned} \quad (20)$$

where $\overline{U_0U_3}$ represents the pixel distance from (u_0, v_0) to (u_3, v_3) and $\overline{U_0U_4}$ represents the pixel distance from (u_0, v_0) to (u_4, v_4) .

Secondly, the path direction is determinate by comparing the magnitude of $\overline{U_0U_3}$ and $\overline{U_0U_4}$. If $\overline{U_0U_3} > \overline{U_0U_4}$,

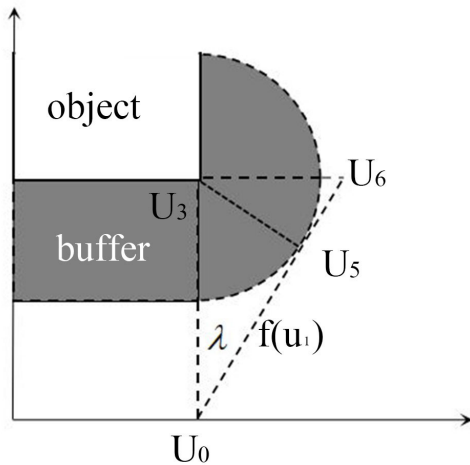


FIGURE 5. The principle of route selection.

the route direction will be U_4 ; if $\overline{U_0U_3} < \overline{U_0U_4}$, the route direction will be U_3 .

Take $\overline{U_0U_3} < \overline{U_0U_4}$ as an example. If $v_3 < v_0 - d$, it can be passed directly by the straight line. The distance of the pixel can be expressed as follows.

$$D = s_2 \times A + M \tag{21}$$

where A is the coefficient obtained from (11), and D represents the distance.

Further, the route function can be expressed as follows.

$$u = u_0 \tag{22}$$

If $v_3 > v_0 - d$, build a tangent line with U_3 as the origin and d as the radius through U_0 and U_5 . As shown in Fig. 5.

In Fig. 5, $f(u)$ is the tangent line function, and it intersects the buffer at U_3 . The distance between U_3 and U_5 is d . Then the following equation is satisfied.

$$\begin{aligned} \sin \lambda &= \frac{d}{\overline{U_0U_3}} \\ \lambda &= \arcsin \frac{d}{\overline{U_0U_3}} \end{aligned} \tag{23}$$

Further, the distance of $\overline{U_3U_6}$ can be expressed as follows.

$$\begin{aligned} \overline{U_3U_6} &= \tan\left(\frac{\pi}{2} - \lambda\right) \times \overline{U_0U_3} \\ \overline{U_3U_6} &= \sqrt{(u_3 - u_6)^2 + (v_3 - v_6)^2} \end{aligned} \tag{24}$$

where (u_6, v_6) represents the coordinate of U_6 .

As shown in Fig. 4 (a), $u_4 = u_6$. Therefore, v_6 can be expressed as follows.

$$v_6 = \left| v_4 - \sqrt{\overline{U_3U_6}^2 - (u_3 - u_6)^2} \right| \tag{25}$$

The route function can be deduced from (u_6, v_6) and (u_0, v_0) . The function can be expressed as follows.

$$\begin{aligned} f(u) &= \left(\frac{v_0 - v_6}{u_0 - u_6}\right)u + v_6 - \left(\frac{v_0 - v_6}{u_0 - u_6}\right)u_6, \\ f(u) &\in [u_6, u_0] \end{aligned} \tag{26}$$

If $v \in [0, v_6]$, $f(u)$ can be expressed as follows.

$$u = u_6 \tag{27}$$

In the case of two objects in *Tex*, it is necessary to calculate the distance from U_0 to the identification box nodes in turn (As shown in Figure 6). Firstly, compare the magnitude of $\overline{U_0U_4}$ and $\overline{U_0U_3}$ sequentially to determine the direction of the route through object 1. Secondly compare the magnitude of $\overline{U_0U_8}$ and $\overline{U_0U_7}$ sequentially to determine the direction of the route through object 2. The method refers to (20). The arrangement of objects is shown in Fig. 6.

In Fig. 6(a), object 1 and object 2 are both to the left of U_0 . Therefore, the route is directed along U_7 and is represented as follows.

$$\begin{aligned} f(u) &= \left(\frac{v_0 - v_6}{u_0 - u_6}\right)u + v_6 - \left(\frac{v_0 - v_6}{u_0 - u_6}\right)u_6, \\ f(u) &\in [v_6, v_0] \\ u &= u_6, \quad v \in [0, v_6] \end{aligned} \tag{28}$$

In Fig. 6(b), the route direction through object 1 is determined based on the magnitude of $\overline{U_0U_4}$ and $\overline{U_0U_3}$. The route direction through object 2 is determined based on the magnitude of $\overline{U_0U_8}$ and $\overline{U_0U_7}$. Finally the route function is expressed as follows.

$$\begin{aligned} f(u) &\in [v_9, v_6] \\ f(u) &= \left(\frac{v_0 - v_6}{u_0 - u_6}\right)u + v_6 - \left(\frac{v_0 - v_6}{u_0 - u_6}\right)u_6 \\ f(u) &\in [v_6, v_0] \\ u &= u_6, \quad v \in [v_9, v_6] \\ f(u) &= \left(\frac{v_6 - v_9}{u_6 - u_9}\right)u + v_9 - \left(\frac{v_6 - v_9}{u_6 - u_9}\right)u_9, \\ u &= u_9, \quad v \in [0, v_9] \end{aligned} \tag{29}$$

where (u_6, v_6) is obtained by (25).

In Fig. 6(c), object 1 and object 2 are on both sides of U_0 , and the interval between object 1 and object 2 is larger than $2d$. Therefore, the function can be expressed as follows.

$$u = u_0, \quad v \in [0, M] \tag{30}$$

In Fig. 6(d), object 1 and object 2 are on both sides of U_0 . However the interval between object 1 and object 2 is less than $2d$. Therefore, it is necessary to compare the magnitudes of $\overline{U_0U_4}$ and $\overline{U_0U_7}$ respectively and obtain route functions which similar to (28).

During *Tex* continuous generate, the direction of texture stitch can be judged by selecting routes.

D. THREE-DIMENSIONAL INTERACTION VIRTUAL REALITY CONSTRUCTION

Based on the method described in the preceding section, the three-dimensional interaction virtual reality is constructed by building cubes model and adding *Tex* to it. *Tex* includes the wall texture (*Tex_wall*), window texture (*Tex_window*), and the floor texture (*Tex_floor*). Therefore, three cubes need to

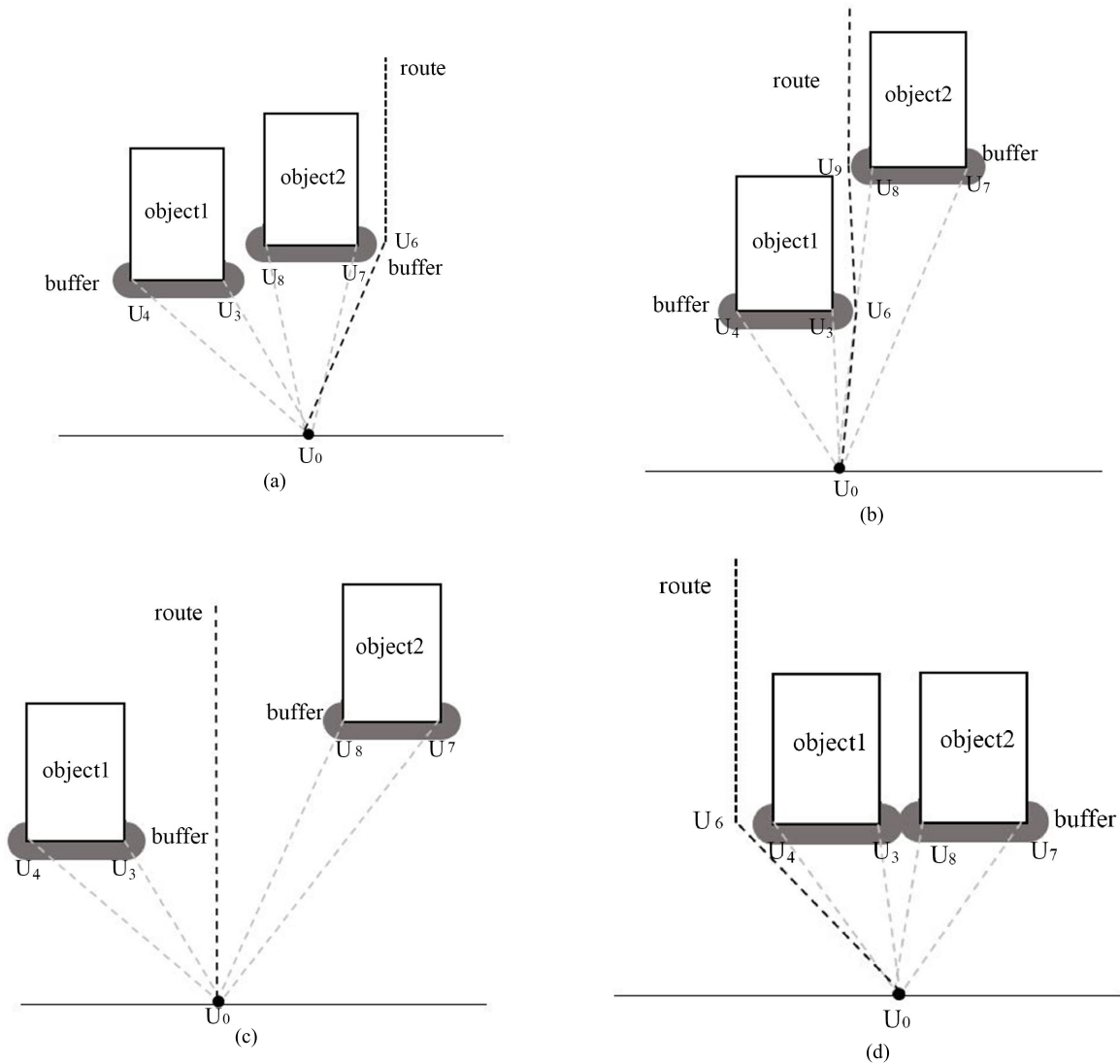


FIGURE 6. The alignment of objects and the route function respectively. (a) Alignment 1. (b) Alignment 2. (c) Alignment 3. (d) Alignment 4.

be constructed, including the wall cube model, the window cube model and the floor cube model.

Firstly, set the size of the Tex_wall to $M_1 \times N_1$, set the size of the Tex_window to $M_2 \times N_2$, set the size of the Tex_floor to $M_3 \times N_3$.

Secondly, according to the actual size of the corridor, the wall cube model, window cube model and floor cube model are constructed. Take the floor cube model as an example, set the actual size of the floor in corridor is $L \times H$. Combining (4), the length and width of the cube model can be expressed as follows.

$$\begin{aligned} l &= \frac{L}{r_1} \\ h &= \frac{H}{r_2} = \frac{Hms_1}{ns_0r_1} \end{aligned} \quad (31)$$

where l represents the length of the cube and h represents the width of the cube. The height of the cube is set to a fixed value w_o . As shown in Fig. 7.

As shown in Fig. 7, according to the Ground Control Points (GCPs), we are aligning the texture to the top surface of floor cube model to paste. The Ground Control Points are the nodes of the floor cube.

In the process of pasting the texture to floor cube model, the following steps need to be followed.

Step 1: Convert the coordinate system of Tex_floor , the coordinate system of the floor cube model and the coordinate system of realistic. Getting the coordinates of Ground Control Points in the three coordinate systems.

Step 2: According to the ground control points, we are aligning the Tex_floor to the top of the floor cube to paste.

Step 3: If the difference between M_3 and L , N_3 and W is large, it is necessary to use the interpolation to make the floor cube satisfy the following conditions.

$$\begin{aligned} L &= M_3 \\ W &= N_3 \end{aligned} \quad (32)$$

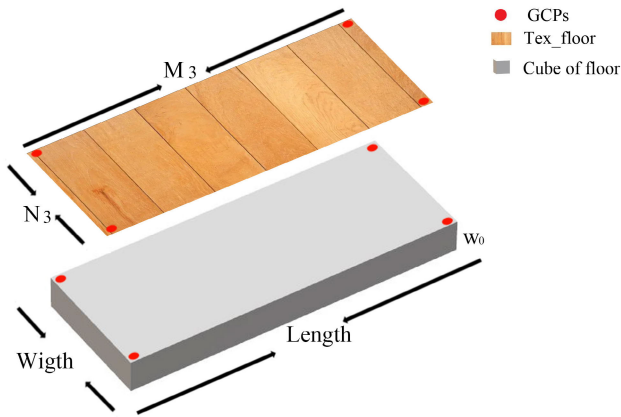


FIGURE 7. The principle of build floor model for corridor three-dimensional interaction virtual reality.

Similarly, the wall cube and the window cube can be built by the above method.

The three-dimensional interaction virtual reality of the corridor is built based on the three cubes. The model as shown in the Fig. 8.

In Fig. 8, cube1 represents the window cube model, cube 2 represents the wall cube model, cube 3 represents the floor cube model.

Place the cube 3 at 0° in the X-axis direction ($X \in [f_0, f_0 + N_3], Y \in [0, M_3], Z = f_0$). Attach the *Tex_floor* to the top of cube 3.

Place the cube 1 at 90° in the Z-axis direction and put it on the left side of the cube 3 ($X = f_0, Y \in [0, M_1], Z \in [f_0, f_0 + N_1]$). Attach the *Tex_window* to the inside of cube3 and connect it to the top of cube1.

Place the cube 2 at 90° in the Z-axis direction and put it on the right side of the cube 3 ($X = f_0 + N_3, Y \in [0, M_2], Z \in [f_0, f_0 + N_2]$). Attach the *Tex_wall* to the outside of cube 2 and connect it to the top of cube 1.

It should be noted that the model to be built should satisfy the following equation.

$$\begin{aligned} M_1 &= M_2 = M_3 \\ N_1 &= N_2 \end{aligned} \quad (33)$$

III. EXPERIMENTS AND RESULTS

Four steps should be followed to construct the three-dimensional interaction virtual reality of the corridor.

Step 1: Measure the sensor height and shoot angle. Shooting the check field. Extracting 4 pairs of coordinates from the image of the check field. Set the coordinates after correcting to invert the parameters $a, b, f_1, f_2, f_3, f_4, r_1$.

Step 2: Keep the sensor at the same height and shoot angle to capture images in the corridor.

Step 3: Identify objects and build buffers in the corridor to select routes. Stitch the images to produce textures.

Step 4: The cubes model are constructed based on the size of the corridor. The textures are aligned with the surface of the model to paste through Ground Control Points.

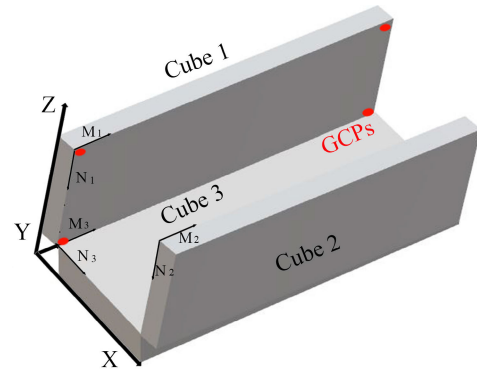


FIGURE 8. The principle of build three-dimensional interaction virtual reality.

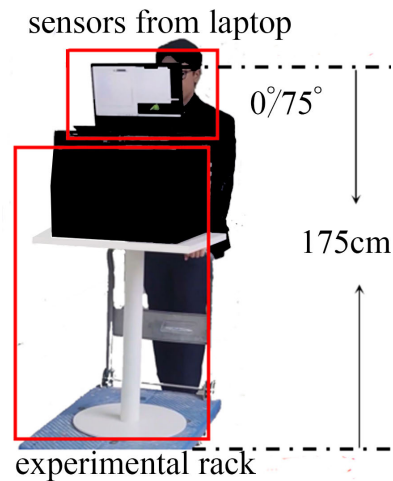


FIGURE 9. The device for experiments.

A. OBTAIN AFFINE PARAMETERS

The three-dimensional interaction virtual reality construction in this paper requires a stable shoot pose and affine parameters. The shoot pose will be determined by measuring shoot height and angle of the sensor. The device, height, and shoot angle are shown in Fig. 9.

In Fig. 9, we use the camera of laptop as the sensor, and the height of the camera is 175 cm, the shoot angle are 0° and 75° . we use this device to collect data of wall, floor, and window respectively. It should be noted that the height and angle when using the device to shoot the check field need to be the same as during the experiment.

The affine parameters are obtained by transforming the 4 pairs of coordinates in check field image. The image of the floor check field is shown in Fig. 10(a). The image of the wall and window check filed are shown in Fig. 10(b).

In Fig. 10, the shoot angle of the floor check filed is 75° . The shoot angle of the wall check filed is 90° .

The Fig. 10 (a) after perspective transforming is shown in Fig. 11(a). The Fig. 10(b) after perspective transforming is shown in Fig. 11(b).

The coordinates after perspective transforming are shown in the following table.

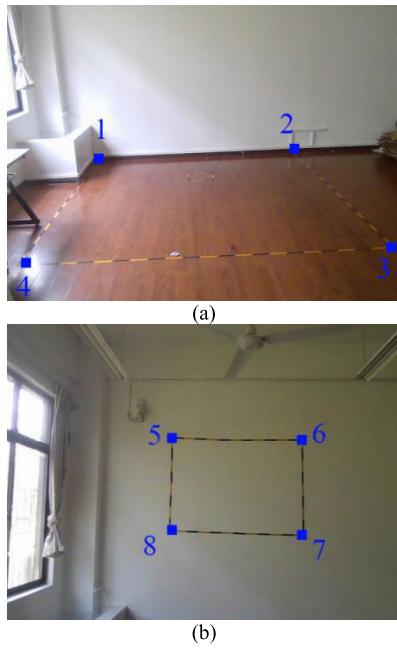


FIGURE 10. The check fields (a) The check field for floor. (b) The check field for wall and window.

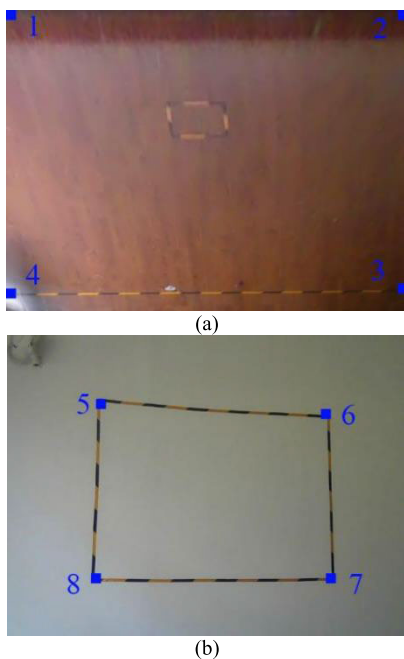


FIGURE 11. The image after perspective transform (a) The image of floor check field after perspective transform. (b) The image of wall and window check field after perspective transform.

The above data can be used to deduce the affine parameters which can perform real-time image correction.

B. CONSTRUCT TEXTURE

According to (6) and combing with Fig. 11, some images are taken and corrected by sensor in corridor as shown below.

In Fig. 12, (a) & (b) represent two successive images taken by the sensor in the direction of the wall, (c) & (d) represent

TABLE 1. Coordinates of points.

Num	Coordinate of Image	Coordinate After Affine Transform
1	(249, 147, 0)	(1, 1, 0)
2	(234, 465, 0)	(1,640,0)
3	(393, 629, 0)	(445, 640, 0)
4	(420, 21, 0)	(453, 1, 0)
5	(0, 180, 268)	(0, 106, 150)
6	(0, 188, 480)	(0, 126, 511)
7	(0, 341, 481)	(0, 386, 518)
8	(0, 332, 265)	(0, 387, 139)

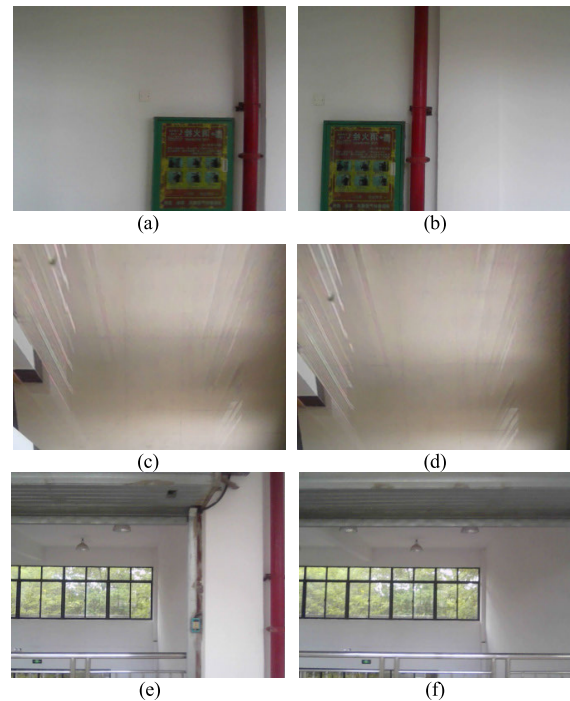


FIGURE 12. Some images are taken by sensor and corrected. (a) & (b) are successive images in wall direction. (c) & (d) are successive images in floor direction. (e) & (f) are successive images in wall direction.

two successive images taken by the sensor in the direction of the floor. (e) & (f) represent two successive images taken by the sensor in the direction of the wall.

The feature points of (a) & (b) and (e) & (f) are sufficient. Feature stitching can be performed. As shown in Fig. 13 (a) & (c), respectively. The feature points of (c) & (d) are insufficient. Direct stitching can be performed. As shown in Fig. 13 (c).

The textures (include $Tex_wallTex_windowTex_floor$) are produced by using direct stitch and feature stitch. The textures after stitching are shown in Fig. 14.

In Fig. 14, due to the limited in image field of the sensor shot and the relatively narrow space, we cannot build a complete wall texture and window texture.

C. SELECT ROUTE

In the Tex_floor , we take the bottom edge of the wall as the object and select the path. The examples of identified objects are shown in the Fig. 15.

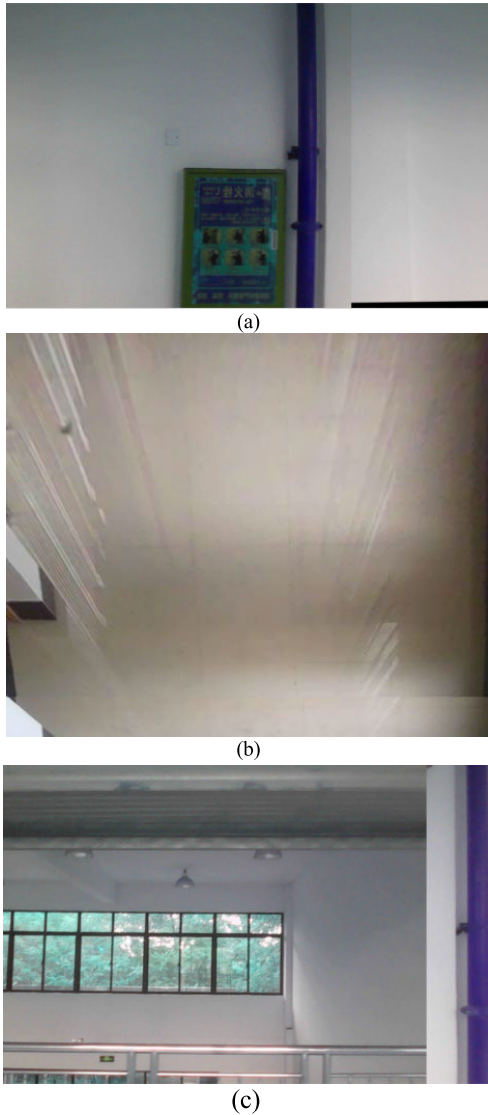


FIGURE 13. The images after stitching. (a)The image after feature stitching. (b) The image after direct stitching. (c) The image after feature stitching.

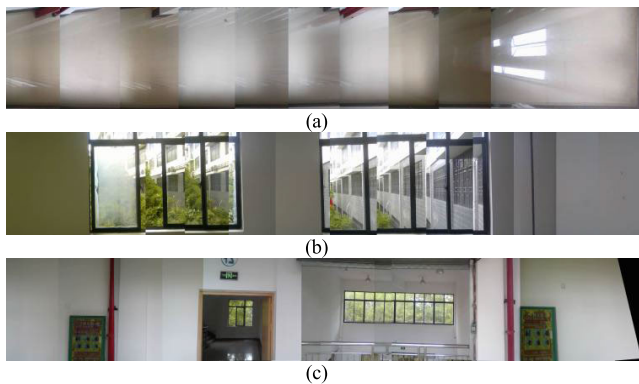


FIGURE 14. The texture produce by stitching. (a) The texture of floor. (b) The texture of window. (c) The texture of wall.

In Fig. 15, the green boxes represent the identification boxes. Consider the size of the experimental rack, we set

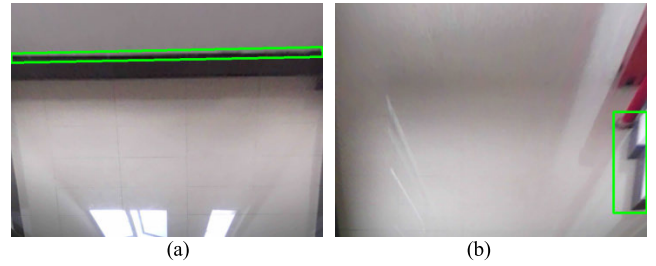


FIGURE 15. The examples of identified objects.

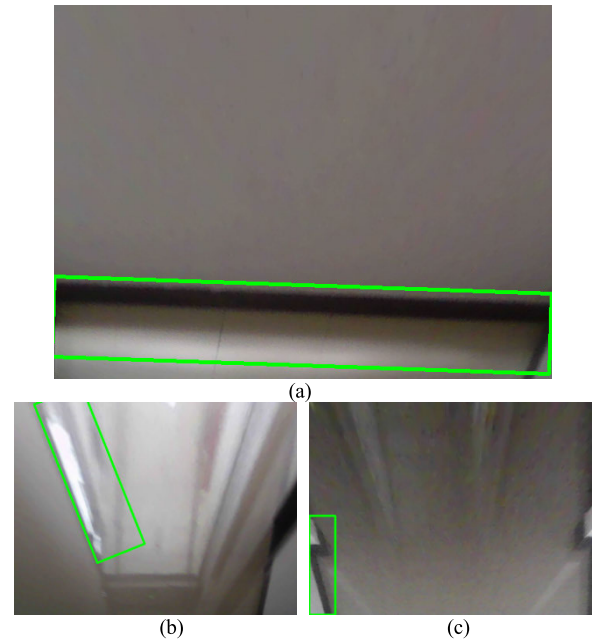


FIGURE 16. The different cases in different angel around the experimental rack. (a) The case in 90°. (b) The case in 180°. (c) The case in 270°.

the distance of 30 pixels as the buffer radius to build the buffer. According to (20), we calculate the distance from identification box to the buffer. When the distance is less than 30 pixels, the path can be passed in a straight line (as shown in (30)). When the distance is large than $N-30$, we will rotate 90° and 270° in clockwise to whole the experimental rack until a new path is found. As shown in Fig. 16.

In Fig. 16, we need to find a new route since the distance is large than $N-30$. After rotating 90° in clockwise direction, since the distance is large than $N-30$ again (as shown in Figure 16(a)), the route is excluded. After rotating 180° clockwise, this is the original route, so it is not referenced (as shown in Figure 16(b)). After rotating 270° in clockwise direction, the distance is less than $N-30$, so it is the new route (as shown in Fig. 16(c)).

In summary, we refer to the data captured on the new route as the texture of Scene 2. And the old one called Scene 1. The texture still includes wall texture, window texture and floor texture ($Tex_{wall} Tex_{window} Tex_{floor}$). According to the *Constrict Texture*, the images are stitched together by using

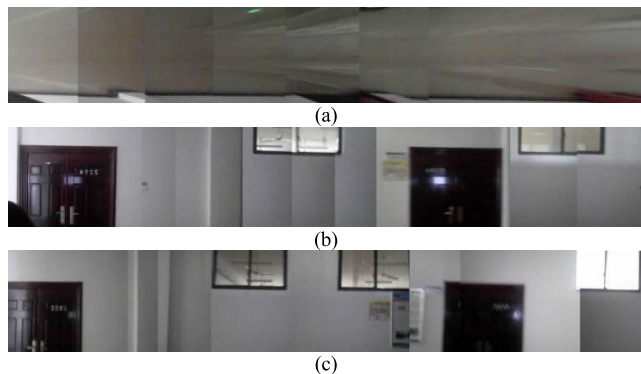


FIGURE 17. The texture of Scene2 produce by stitching. (a) The Scene2 texture of floor. (b) The Scene2 texture of window. (c) The Scene2 texture of wall.

the feature stitch and direct stitch to produce the textures. The texture of Scene 2 as shown in Fig. 17.

In Fig. 17, similar to the Fig. 14, we cannot build the complete wall texture and window texture because of the image field.

D. BUILD MODEL

In the process of building the three-dimensional interaction virtual reality includes modeling different scenes separately. Three cubes are needed to build in different scene respectively. We build two scenes in this paper, and refer two scenes to Scene 1 & Scene 2.

In Scene 1, the size of the corridor is $10m \times 3m \times 1.5m$. We known it by measuring. Firstly, we reference the Fig. 8 to construct the coordinate system, and make each unit represents the actual distance is $0.009m \times 0.004m$. Secondly, we make $w = 1m$ and construct the cubes model. The cubes model of the Scene 1 is shown in Fig. 18(a).

In scene 2, the size of the corridor is $10m \times 3m \times 1.5m$. Similarly, the cubes model of Scene 2 is shown in Fig. 18(b).

Since there is a partial overlapping between the Scene 1 cubes model and Scene 2 cubes model, we merge the models. As shown in Fig. 19.

As shown in Fig. 18 & 19, the coordinates of the 20 nodes contained in Scene 1 and Scene 2 are shown in the following table.

As shown in Fig. 18, the coordinates of the 20 nodes contained in Scene 1 and Scene 2 are shown in the following table.

According to the Fig. 19, we choose the points from 1 to 20 as the control points. According to the previous section, we attach the textures to the model. As shown in the Fig. 20.

In Fig. 20, there are different angle to show the three-dimensional interaction virtual reality of corridor. Due to the limited in image field and the relatively narrow space, we cannot build a complete wall model and window model during the process of construct the model.

E. RESULT ANALYSIS

In this paper, we will analyze the model in terms of visual effect and accuracy. On the one hand, the Three-Dimensional

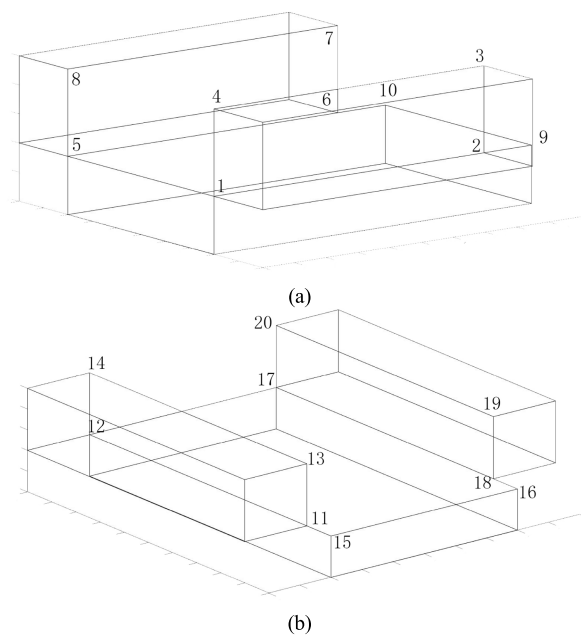


FIGURE 18. The cube model of two scene. (a) The cube model of Scene1. (b) The cube model of Scene 2.

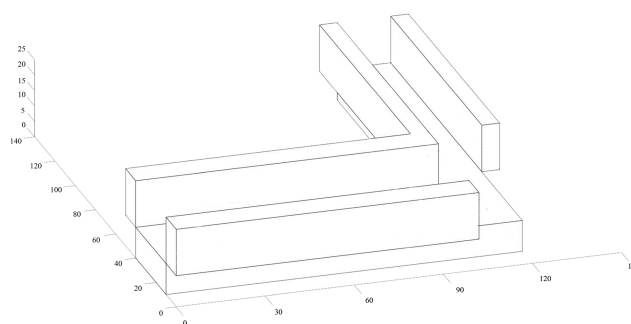


FIGURE 19. The cube model of whole scene.

Model based on photogrammetry can only model the floor and cannot produce a valid corridor model; on the other hand, the model produced by the method proposed in this paper is fundamentally different from the point cloud model and is not comparable. Therefore, this paper is compared with the Building Information Modeling.

1) IN TERMS OF VISUAL EFFECT

Building Information Modeling is achieved by surveying corridor size and making Computer Aided Design (CAD) file; Importing the CAD file into Building Information Modeling software.

The accuracy of Building Information Modeling can be divided into five levels from Level of Detail 100 (LOD100) to Level of Detail 500 (LOD 500). LOD 100 requires the model to be evaluated in terms of area, height, volume, location and sitting direction. LOD 200 requires an increase in the number of components as an evaluation item. These items are consistent with the models generated in this paper. Therefore,

TABLE 2. Coordinates of twenty nodes.

Num	Coordinate of Model	Coordinate of Actual
1	(0, 10, 10)	(0, 0, 0)
2	(100, 10, 10)	(0, 10, 0)
3	(100, 10, 25)	(0, 10, 1.5)
4	(0, 10, 25)	(0, 0, 1.5)
5	(0, 40, 10)	(-3, 0, 0)
6	(100, 40, 10)	(-3, 10, 0)
7	(0, 40, 25)	(-3, 10, 1.5)
8	(100, 40, 25)	(-3, 0, 1.5)
9	(120, 10, 10)	(0, 12, 0)
10	(120, 40, 10)	(-3, 12, 0)
11	(100, 50, 10)	(-5, 10, 0)
12	(100, 140, 10)	(-14, 10, 0)
13	(100, 50, 25)	(-5, 10, 1.5)
14	(100, 140, 25)	(-14, 10, 1.5)
15	(100, 40, 10)	(-14, 10, 0)
16	(120, 40, 10)	(-4, 12, 0)
17	(120, 50, 10)	(-5, 12, 0)
18	(120, 140, 10)	(-14, 12, 0)
19	(120, 140, 25)	(-14, 12, 1.5)
20	(120, 50, 25)	(-5, 12, 1.5)

this paper uses the Building Information Modeling under LOD 100 and LOD 200 levels for comparison. The model by using Building Information Modeling under LOD 100 and LOD 200, CAD file, the model by proposing in this paper are shown in Fig. 21.

According to Fig. 21 the volume, area, height, position and sitting direction of the model at LOD 100 and the model obtained in this paper are the same. and the model in this paper has texture. Compared with the model at LOD 200, the model obtained in this paper can be clearly determined the materials of the components (like windows, floors, etc.) visually. However, Compared with the model obtained by Building Information Modeling, the gap in the model proposed in this paper is due to the implementation of path selection.

2) IN TERMS OF ACCURACY

We use the absolute error and root mean square error to test accuracy of model.

In order to check the accuracy of the three-dimensional interaction virtual reality of the corridor, we select three feature areas to test the model accuracy. As shown in Fig. 22. It should be noted that we are using the actual distance as Coordinate of Actual.

In Fig. 22, there are 12 points in feature areas, and the results of comparison are expressed as follow. It should be noted that, in order to accurately check the point accuracy

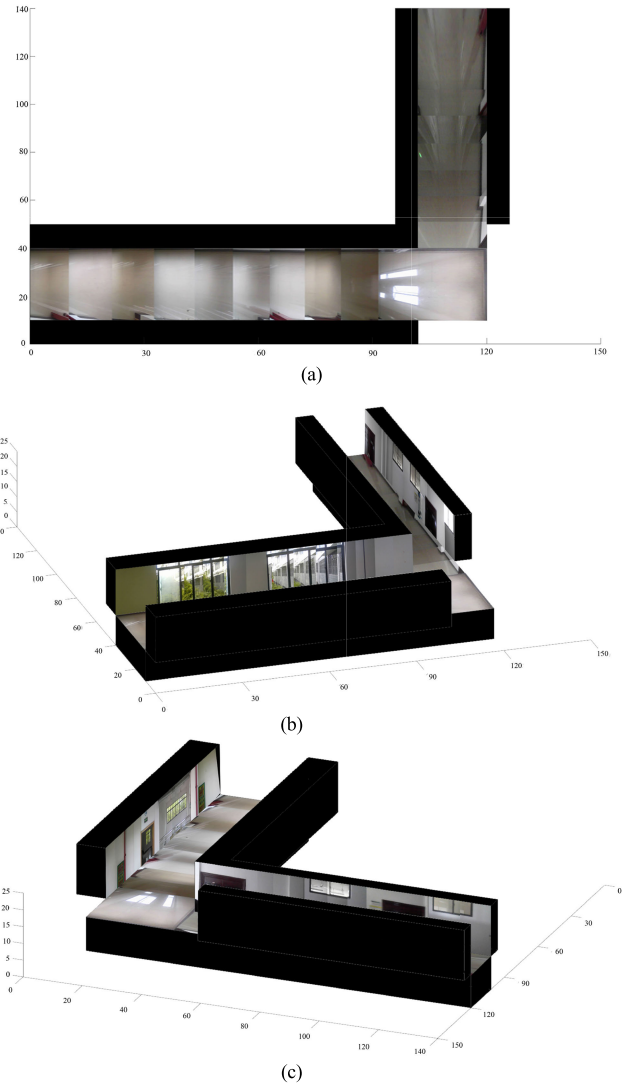


FIGURE 20. The corridor three-dimensional interaction virtual reality in different angle.

of the feature points, firstly, all *w* values in the model were eliminated. Secondly, divide all value of coordinate of three-dimensional interaction virtual reality by 10.

From the above table, the absolute error of the feature points in the feature area is [0.0507,0.1691].

According to the above table, the root mean square error of the feature points can be further derived. The formula for calculating the root mean square error can be expressed as follows.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{num} [(x_i - X_i)^2 + (y_i - Y_i)^2 + (z_i - Z_i)^2]}{num}} \quad (34)$$

where, σ represents the root mean square error of the feature points.

According to the (34), the root mean square error of the feature points in the feature region is 0.1265.

In order to verify the root mean square error can represent the accuracy of the model, we perform accuracy verification

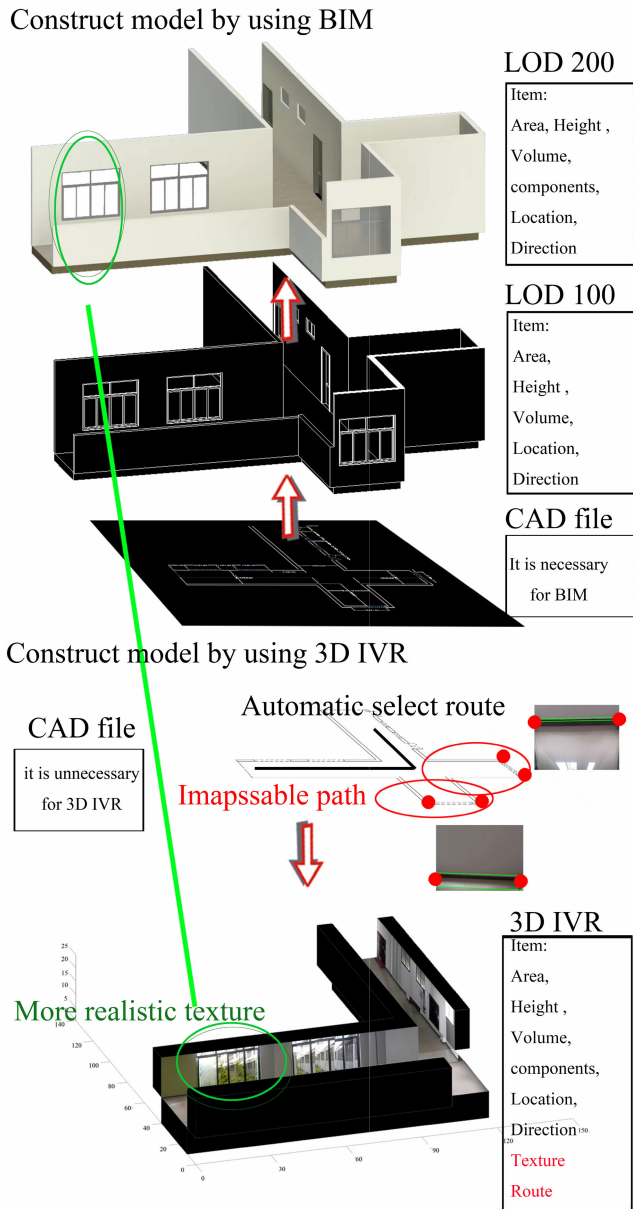


FIGURE 21. The corridor three-dimensional interaction virtual reality under LOD 100 to LOD 200 by using building information modeling.

by the distance difference method. The distance difference method can be expressed as follows.

$$\sigma = \sqrt{\frac{(x_1 - x_2)^2}{D^2} 2m^2 + \frac{(y_1 - y_2)^2}{D^2} 2m^2 + \frac{(z_1 - z_2)^2}{D^2} 2m^2} \quad (35)$$

where,

$$m = \sqrt{\sum_{i=1}^k \frac{\Delta D_i^2}{k-1}} \quad (36)$$

$\Delta D = D_{real} - D_{pixel}$

Further, we select four windows and two doors in the model for accuracy verification. As shown in Fig. 23.

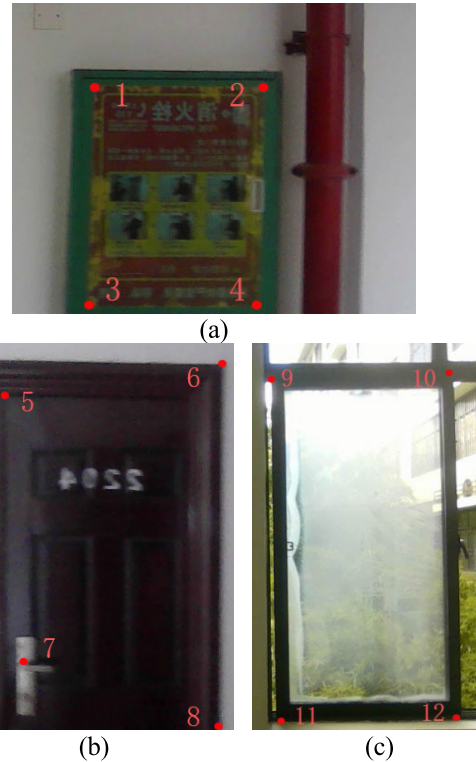


FIGURE 22. Three feature area to test the accuracy of corridor three-dimensional interaction virtual reality. (a) The box in Scene 1. (b) The door in Scene 2. (c) The window in Scene 1.

TABLE 3. Coordinates of points.

Num	Coordinate of Actual	Coordinate of three-dimensional interaction virtual reality	error
1	(0, 1.88, 0.73)	(0, 1.7830, 0.8656)	0.1664
2	(0, 2.55, 0.73)	(0, 2.4490, 0.8656)	0.1691
3	(0, 1.88, 0)	(0, 1.7830, 0)	0.0970
4	(0, 2.55, 0)	(0, 2.4490, 0)	0.1010
5	(-3.12, 10, 0.91)	(-3.1696, 10, 1.0403)	0.1396
6	(-3.96, 10, 1.1)	(-4.0107, 10, 1.2305)	0.1400
7	(-3.15, 10, 0.19)	(-3.1998, 10, 0.3021)	0.1229
8	(-3.96, 10, 0)	(-4.0107, 10, 0)	0.0507
9	(-3, 1.21, 1.38)	(-3, 1.3320, 1.4637)	0.1479
10	(-3, 1.83, 1.38)	(-3, 1.9415, 1.4823)	0.1510
11	(-3, 1.21, 0)	(-3, 1.3320, 0)	0.1220
12	(-3, 1.83, 0)	(-3, 1.9415, 0)	0.1115

As shown in Fig. 23, the 13 to 24 are part of the corner points of the windows and doors.

The size of the windows in the test area are $1.38 \times 0.62(m)$, and the size of the doors in the image field are $0.91 \times 0.88(m)$.

We set the value of the size of the doors and windows are D_{real} . According to the (35) and (36), the root mean square error of four windows and two doors can be expressed as follows.

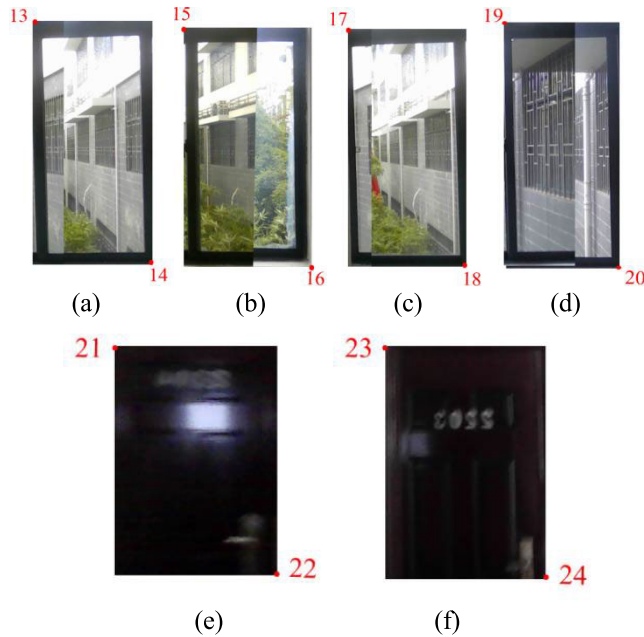


FIGURE 23. Six areas to test the accuracy of corridor three-dimensional interaction virtual reality. (a), (b), (c), and (d) are windows area. (e) and (f) are doors area.

TABLE 4. Root mean square error.

Num	ROOT MEAN SQUARE ERROR
(a)	0.1331
(b)	0.1420
(c)	0.1316
(d)	0.1333
(e)	0.1297
(f)	0.1210

From the above Table, the average root mean square error of Fig.23 is 0.1318. Compared with the root mean square error of the feature point, the difference is 0.0062, And It is similar. Therefore, the accuracy of the model in this paper is about 0.1203 to 0.1318.

IV. CONCLUSION

This paper proposes a method for constructing corridor three-dimensional interaction virtual reality. The method has the following advantages compared with photogrammetry, point cloud modeling and Building Information Modeling.

1. Compared with the photogrammetry to build three-dimensional interaction virtual reality, the method proposed in this paper can build a narrow space model without elevation data. At the same time, the method automatically select route to modeling without planning the route.

2. Compared with point cloud modeling to build three-dimensional interaction virtual reality, the method proposed in this paper has lower cost.

3. Compared with Building Information Modeling to build three-dimensional interaction virtual reality, the texture produced by the our method is more realistic. And, this method

can directly realize the model construction of Level of detail 200 (LOD 200).

However, the method proposed in this paper also has the following disadvantages: Firstly, the error of three-dimensional interaction virtual reality caused by perspective transformation, texture stitching and texture matching is decimeter level. Secondly, stable shooting posture is required. If the shooting posture is unstable, it will further aggravate the error described in 1. Thirdly, the size of the corridor needs to be measured in advance. The cube model is built with that size, which cannot be realized in real-time modeling. Based on the method described in this paper, the research in the future should be conducted in the following accepts.

1. Improve the stability of the equipment.
2. Adding moment recording and speed functions to the device to make direct stitching more accurate and thus reduce errors.
3. Using semantic cutting for buffer construction to make path selection more accurate.

ACKNOWLEDGMENT

The authors would like to thank the editor-in-chief, the associate editor, and the reviewers for their insightful comments and suggestions.

REFERENCES

- [1] A. Jones, M. Lang, G. Fyffe, X. Yu, J. Busch, I. McDowall, M. Bolas, and P. Debevec, "Achieving eye contact in a one-to-many 3D video teleconferencing system," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–8, Jul. 2009.
- [2] H. Guo, H. Li, and M. Skitmore, "Life-cycle management of construction projects based on virtual prototyping technology," *J. Manag. Eng.-ASCE*, vol. 26, no. 1, Jan. 2021, doi: [10.1061/\(ASCE\)0742-597X\(2010\)26:1\(41\)](https://doi.org/10.1061/(ASCE)0742-597X(2010)26:1(41)).
- [3] F. Nex and F. Remondino, "UAV for 3D mapping applications: A review," *Appl. Geomatics*, vol. 6, no. 1, pp. 1–15, Mar. 2014.
- [4] Y. H. S. Jo, "Three-dimensional digital documentation of cultural heritage site based on the convergence of terrestrial laser scanning and unmanned aerial vehicle photogrammetry," *Int. J. Geo-Inf.*, vol. 8, no. 2, p. 53, 2019, doi: [10.3390/IJGI8020053](https://doi.org/10.3390/IJGI8020053).
- [5] Y. Deng, J. C. P. Cheng, and C. Anumba, "Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison," *Autom. Construction*, vol. 67, pp. 1–21, Jul. 2016.
- [6] C. Yang, F. Zhang, Y. Gao, Z. Mao, L. Li, and X. Huang, "Moving car recognition and removal for 3D urban modelling using oblique images," *Remote Sens.*, vol. 13, no. 17, p. 3458, Aug. 2021.
- [7] G. J. Grenzdörffer, M. Guretzki, and I. Friedlander, "Photogrammetric image acquisition and image analysis of oblique imagery," *Photogrammetric Rec.*, vol. 23, no. 124, pp. 372–386, Dec. 2008.
- [8] G. Aguiaro, "Energy planning tools and CityGML-based 3D virtual city models: Experiences from trento (Italy)," *Appl. Geomatics*, vol. 8, no. 1, pp. 41–56, Mar. 2016.
- [9] I. R. Otero. *Anatomy of the SIFT Method*. École Normale Supérieure de Cachan-ENS Cachan. [Online]. Available: <http://demo.ipol.im/demo/82/>
- [10] G. Zhou, X. Liu, T. Yue, Q. Wang, H. Sha, S. Huang, and Q. Pan, "A new graduation algorithm for color balance of remote sensing image," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vols. XLII–3, pp. 2517–2521, May 2018, doi: [10.5194/ISPRS-ARCHIVES-XLII-3-2517-2018](https://doi.org/10.5194/ISPRS-ARCHIVES-XLII-3-2517-2018).
- [11] X. Liu, G. Zhou, W. Zhang, and S. Luo, "Study on local to global radiometric balance for remotely sensed imagery," *Remote Sens.*, vol. 13, no. 11, p. 2068, May 2021.
- [12] R. D. Singh, A. Mittal, and R. K. Bhatia, "3D convolutional neural network for object recognition: A review," *Multimedia Tools Appl.*, vol. 78, no. 12, pp. 15951–15995, Jun. 2019.
- [13] I. Wu and S. Hsieh, "Transformation from IFC data model to GML data model: Methodology and tool development," *J. Chin. Inst. Engineers*, vol. 30, no. 6, pp. 1085–1090, Sep. 2007.

- [14] Y. L. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021, doi: [10.1109/TPAMI.2020.3005434](https://doi.org/10.1109/TPAMI.2020.3005434).
- [15] Y. Wang, S. Li, M. Wang, and Y. Lin, "A simple deep learning network for classification of 3D mobile LiDAR point clouds," *J. Geodesy Geoinf. Sci.*, vol. 4, no. 3, p. 49, Oct. 2021.
- [16] L. E. Carvalho and A. von Wangenheim, "3D object recognition and classification: A systematic literature review," *Pattern Anal. Appl.*, vol. 22, no. 4, pp. 1243–1292, Nov. 2019.
- [17] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 84–100, Jan. 2018.
- [18] M. Malovichko, N. Khokhlov, N. Yavich, and M. Zhdanov, "Acoustic 3D modeling by the method of integral equations," *Comput. Geosci.*, vol. 111, pp. 223–234, Feb. 2018.
- [19] N. Skandhakumar, J. Reid, F. Salim, and E. Dawson, "A policy model for access control using building information models," *Int. J. Crit. Infrastructure Protection*, vol. 23, pp. 1–10, Dec. 2018.
- [20] T. W. Kang and C. H. Hong, "A study on software architecture for effective BIM/GIS-based facility management data integration," *Autom. Construction*, vol. 54, pp. 25–38, Jun. 2015.
- [21] J. Han, O. Heo, M. Park, S. Kee, and M. Sunwoo, "Vehicle distance estimation using a mono-camera for FCW/AEB systems," *Int. J. Automot. Technol.*, vol. 17, no. 3, pp. 483–491, Jun. 2016.
- [22] J. W. Lee, "A machine vision system for lane-departure detection," *Comput. Vis. Image Understand.*, vol. 86, no. 1, pp. 52–78, Apr. 2002.
- [23] S. Kole, C. Agarwal, T. Gupta, and S. Singh, "SURF and RANSAC: A conglomerative approach to object recognition," *Int. J. Comput. Appl.*, vol. 109, no. 4, pp. 7–9, Jan. 2015, doi: [10.5120/19174-0645](https://doi.org/10.5120/19174-0645).
- [24] G. Zhou, R. Zhang, and S. Huang, "Generalized buffering algorithm," *IEEE Access*, vol. 9, pp. 27140–27157, 2021.



JINYE WANG received the Ph.D. degree in science from the Central South University of Forestry and Technology, Changsha, China, in 2006. He was a Visiting Researcher at the Institute of Ecology and Environment, in 2003, and Chinese Academy of Forestry. He was a Visiting Researcher at the Guangxi Forestry Research Institute, in 2010. He is currently the Director and a Researcher of the Institute of Ecological Planning, Guilin University of Technology. He is also a Visiting Professor with the Hezhou College, Hezhou, China. He has authored two books and more than 15 projects.



SHENGXIN HUANG was born in 1994. She received the M.S. degree from the College of Geomatics and Geoinformation, Guilin University of Technology, Guilin, China, in 2020. Her research interest includes spatial analysis.



TINGTING ZHANG was born in 1993. She is currently pursuing the M.S. degree with the College of Arts, Guilin University of Technology, Guilin, China. Her research interest includes building information modeling.



XIAOFAN LIU was born in 1995. He received the M.S. degree from the College of Geomatics and Geoinformation, Guilin University of Technology, Guilin, China, in 2020. His research interest includes three-dimensional interaction virtual reality construct.



JINJIN LU received the M.S. degree from the College of Geomatics and Geoinformation, Guilin University of Technology, Guilin, China. Her research interest includes accuracy analysis of three-dimensional model.

...