

Title:

An Image-Based Information Representation for 3D Models

Authors:

Qiujiào Wang, wantchoosejoy@163.com, Southwest Jiaotong University Hope College
Zhijie Xie, wanzhuxie@163.com, Shanghai Taiwin Software Technology Co., Ltd.

Keywords:

Model coordinates, Image coordinates, PPT coordinates, CATIA, CAA

DOI: 10.14733/cadconfP.2024.371-375

1. Introduction

The 3D model of a mature product may be relatively large for storage, making it difficult to transmit and review in real time [7]. Data formats such as STL can lighten 3D model data, but it is difficult to express annotation information. In practical industrial applications, there will be multiple phases that need to report the schedule within a project life cycle. Generally, CAD models are not used for reporting, but Microsoft Office software PPT is used as the carrier. The local areas of the 3D model can be captured, after which a summary PPT file can be generated with the captured images and annotation information.

Engineers can take screenshots of specific local positions of 3D models through operations such as movement, rotation, and scaling and add the images and annotation information to the PPT pages. However, in practical projects, there may be thousands of positions that require screenshots to illustrate, and manual adjustments can take up a lot of time. Moreover, sometimes, it is uncertain what effect the model needs to present in the graphic window, which can only be determined after considering the model features and conducting extensive calculations. Therefore, in the project reporting phase, it can greatly improve the efficiency of information summarizing by taking screenshots of local positions of a 3D model and generating a PPT file automatically. This article presents a method for the extraction of 3D model information and the generation of a PPT summary file based on CATIA.

2. Main Idea

2.1 Related Fundamentals

(1) CAA

CAA (Component Application Architecture) is a set of secondary development interfaces provided by CATIA, with rich functional extension APIs (Application Programming Interface) that users can use to develop functions that are suitable for their specific business needs [2, 4].

(2) Window equivalent size

In this article, the term "window equivalent size" is used to refer to the smaller of the height and width of the graphic window. In CAA, the 3D model window has its own specific API object class, which is *CAT3DViewer*.

(3) Bounding sphere

The bounding sphere is a mathematical object that includes two numerical parameters: the centre and the radius. In CAA, the class of the bounding sphere is *CAT3DBoundingSphere*. According to the classification of purposes, bounding spheres can be divided into two types: window bounding sphere and model bounding sphere. The window bounding sphere refers to the sphere with the largest radius

that can be fully displayed in the window, and its diameter is equal to the window's equivalent size. The model bounding sphere refers to the sphere with the minimum radius that can completely surround the model [3, 5]. As shown in Fig. 1, the larger sphere is the window-bounding sphere, and the smaller sphere is the model-bounding sphere.

When only focusing on local models or elements, the *CAT3DViewer::ReframeOn* method can be used to center the model and maximize it within the window range. In this case, the model's bounding sphere parameters, including the center and radius, will be the same as those of the window-bounding sphere.



Fig. 1: The window bounding sphere and the model bounding sphere.

(4) Camera parameters

The graphic window displays a projection image, which is equivalent to setting up a camera to capture the image results of a 3D model at a particular position. Camera parameters refer to a set of parameters for observing the pose of the 3D model, also known as viewpoint, which includes parameters such as Eye Position, Target Position, and Target Direction. In CAA, the camera parameters of a model can be expressed using *CATViewpoint*. Each window object has a main viewpoint, which can be obtained by the *CATViewer::GetMain3DViewpoint* method.

2.2. Relationship between Length and Pixel

When taking a window screenshot of the 3D model, the coordinates of the model elements in the image will change with the camera parameters [1], as shown in Fig. 2. The visual effects presented vary depending on the camera parameters.

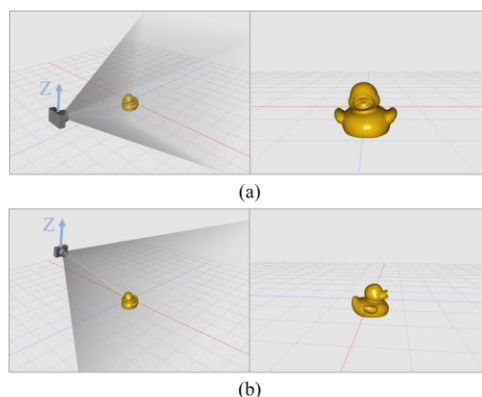


Fig. 2: The different results of the model projection under different camera parameters.

The result is that the pixel length in different images varies with the same actual length in the 3D model. The coordinates of model elements in 3D space are easy to obtain, and CAA provides rich APIs

for calculating the 3D coordinates of model elements and their distances from other elements. Therefore, in the context of 3D models, the relative coordinates between elements are also relatively easy to obtain. At this point, for each image, if we have a correspondence between the actual length in the 3D model and the pixel length in the image, then the pixel coordinates of each element in the image can be further calculated. Here we refer to the correspondence between the actual length and the pixel length as the *Length-Pixel* relationship or the L_p relationship that represents the actual model length of unit pixel length.

When a 3D model is centred in the interactive window of the CAD software, the model bounding sphere fills the entire window, and its parameters are the same as those of the window bounding sphere. Assume that if the radius of the window bounding sphere (actual length) and the size of the window (pixel length) can be obtained, the L_p relationship can be easily calculated, and the image coordinates of any element can be calculated using the L_p relationship. At this point, the problem can be refined as to how to obtain the radius of the window bounding sphere and the size of the window.

In CAA, the size of the CATIA main window can be obtained using the `CATViewer::GetGraphicSize` method, and the window bounding sphere can be obtained using the `CAT3DViewer::GetGlobalBoundingSphere` method. At this point, the L_p relationship can be calculated

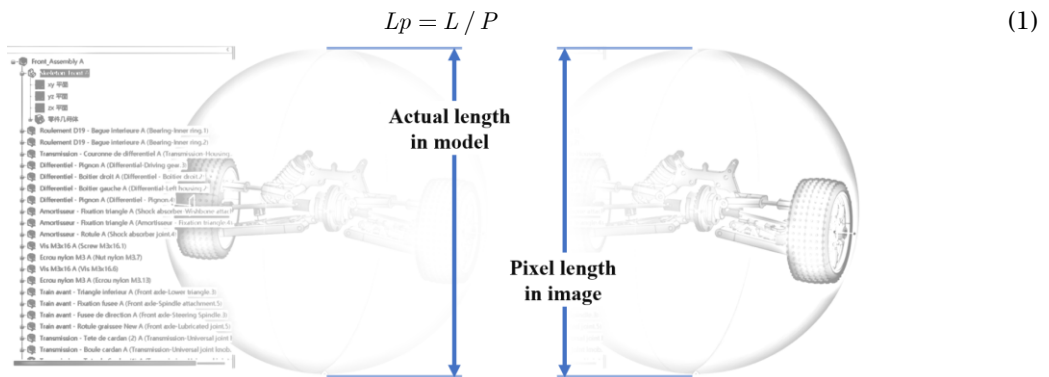


Fig. 3: The L_p relationship of window bounding sphere.

2.3. Calculation of Image Coordinates

The model bounding sphere can be constructed from two values: the centre and the radius, which can be calculated according to the actual situation. For the regular models, such as cubes and vertebrae, their bounding spheres can be calculated through mathematical relationships. For the irregular models, their bounding spheres can be obtained through the following technical paths: first, transfer the model (or some elements) to the `CATI3DGeoVisu` interface; then use its parent class method `CATVisu::GiveRep` to obtain the visual representation `CAT3DRep` of the model; and its method `GetBoundingElement` can be used to obtain the parameters of the bounding sphere.

In fact, when a model is centred using the `CAT3DViewer::ReframeOn` method, the diameter D of the window bounding sphere is slightly smaller than the window equivalent size W , and the relationship is as follows:

$$D = 0.95 * W \quad (2)$$

The point $A(x, y, z)$ in the 3D world coordinate system $OXYZ$ is used as an example in the Fig. 4 to introduce the method of calculating image coordinates. This point can be the centre of gravity of a part or that can represent the position of the 3D model.

Step 1: Centre the model and construct a plane P parallel to the screen with the centre of the sphere as the origin.

Step 2: Project the point A onto the plane P to obtain the projection point $A'(x_L, y_L)$ on the plane. The coordinate system in which the point A' is located is the 2D coordinate system $O'X'Y'$, with the point O' corresponding to the centre point of the graphics window as the origin.

Step 3: Use the Eqn. (1) to convert the coordinate values (x_L, y_L) of the point A' from length to pixel values (x_P, y_P)

$$\begin{cases} x^P = 1 / L_p * x^L \\ y^P = 1 / L_p * y^L \end{cases} \quad (3)$$

Step 4: As the image coordinates have the centre of the upper left corner, the X axis of the rightward direction and the Y axis of the downward direction, it is necessary to calculate the transformation of point A' into the $O''X''Y''$ coordinate system, which is the image coordinates $A''(x''y'')$ of the 3D model point.

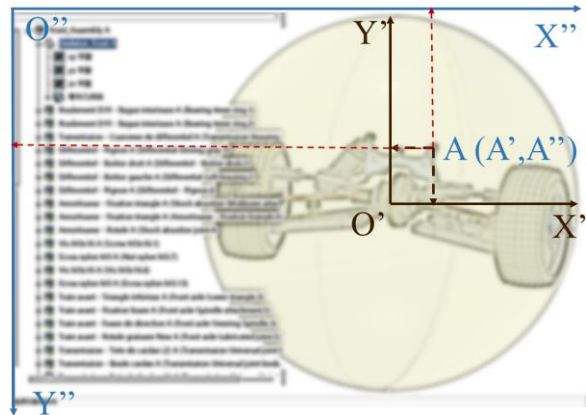


Fig. 4: The image coordinates the calculation process of the point.

2.4. PPT coordinate calculation

The aim of the tool is to annotate information at typical positions corresponding to the 3D model in the PPT text. To perform text annotation in the PPT, it is necessary to obtain the coordinates of the target element in the PPT, and the calculation method is as follows:

$$(X_{Model-PPT}, Y_{Model-PPT}) = (X_{Model-Image}, Y_{Model-Image}) * F_{ImageScale} + (X_{Image-PPT}, Y_{Image-PPT}) \quad (4)$$

Where,

$(X_{Model-PPT}, Y_{Model-PPT})$ is the PPT coordinates of the model element.

$(X_{Model-Image}, Y_{Model-Image})$ is the image coordinates of the model element.

$F_{ImageScale}$ is the image scaling factor.

$(X_{Image-PPT}, Y_{Image-PPT})$ is the PPT coordinates of the image.

The scaling factor refers to the scaling factor at which an image is inserted into a PPT. For example, if the original image size is $600 * 400$ and it becomes $300 * 200$ when inserted into the PPT, the scaling factor is 0.5. The PPT coordinates of an image refer to the position where the image is inserted into the PPT. The size of a PPT page is measured in pixels and is related to the page settings of the PPT software [6]. Fig. 5 shows a page of the final PPT file resulting from the technology mentioned in this article, which can accurately mark the key points of the 3D model.

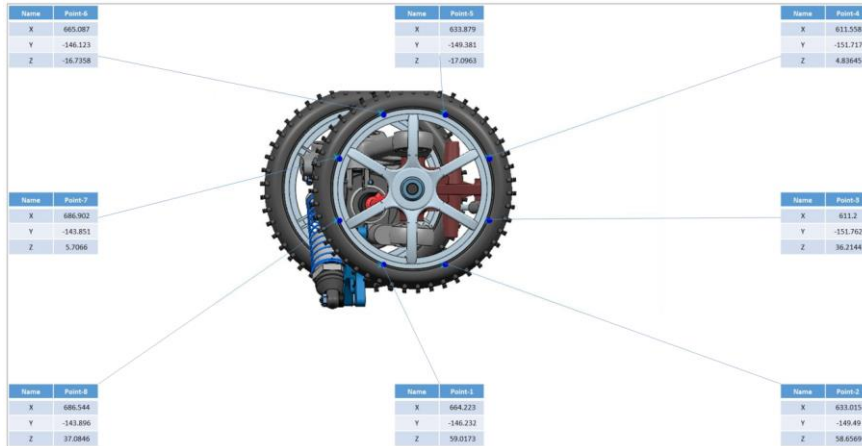


Fig. 5: One page of the final PPT file.

Conclusions:

The information of a 3D model can be described using a PPT document with model screenshots, which can summarize a large amount of information and facilitate transfer and review. When locating elements of a 3D model in images and PPT pages, it is necessary to calculate the correspondence between the actual length in the 3D model and the pixel length in the image or PPT page. Then, the pixel length corresponding to the actual length can be calculated, and the spatial coordinates can be converted to pixel coordinates in the image or PPT page.

Qiuqiao Wang, <https://orcid.org/0000-0003-0346-1198>

Zhijie Xie, <https://orcid.org/0009-0007-0031-2273>

References:

- [1] Ahn, S. H.: https://www.songho.ca/opengl/gl_camera.html, OpenGL Camera.
- [2] Dong, Y.; Su, F.; Sun, G.; Liu, Y.; Zhang, F.: A feature-based method for tire pattern reverse modeling, *Adv Eng Softw*, 124 2018, 73-89. <http://doi.org/10.1016/j.advensoft.2018.08.008>
- [3] Huang, X.; Zhang, S.; Zhou, L.; Huang, L.: A hybrid algorithm for the minimum bounding sphere problem, *Oper Res Lett*, 50(2), 2022, 150-4. <http://doi.org/10.1016/j.orl.2022.01.013>
- [4] Kui, Z.; Lu, Y.; Wei, W.; Xia, A.: IEEE: Parametric Design of Ball-cage Constant Velocity Joint Based on CATIA Customized Development, 2020 3Rd World Conference On Mechanical Engineering and Intelligent Manufacturing (Wcmeim 2020), 2020, 443-8. <http://doi.org/10.1109/WCMEIM52463.2020.00099>
- [5] Meng, F.; Xu, C.; Hao, J.; Xiao, D.: Sphericity evaluation based on minimum circumscribed sphere method, Singapore, Singapore, Trans Tech Publications, 2012, 3146-51. <http://doi.org/10.4028/www.scientific.net/AMR.433-440.3146>
- [6] Microsoft, <https://learn.microsoft.com/en-us/office/vba/api/overview/powerpoint>, PowerPoint VBA reference.
- [7] Roj, R.: A comparison of three design tree-based search algorithms for the detection of engineering parts constructed with CATIA V5 in large databases, *Journal of Computational Design and Engineering*, 1(3), 2014, 161-72. <http://doi.org/https://doi.org/10.7315/ICDE.2014.016>