

Interactive Object Manipulation Strategies for Browsing Object-Centered Image Set

Chih-Han Yu^{1,2}(游直翰), Yu-Pao Tsai¹(蔡玉寶), Chu-Song Chen¹(陳祝嵩)
and Yi-Ping Hung^{1,2}(洪一平)

¹Institute of Information Science, Academia Sinica, Taipei, Taiwan

²Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan.

Tel: +886-2-27883799 ext 1310 Fax: +886-2-26518660

E-mail: song@iis.sinica.edu.tw

Abstract

In image-based approaches, a 3D object is usually represented by an Object-Centered Image Set (OCIS), which is a set of images taken from different viewpoints centered on the 3D object. Our objective is to browse such an image-based object by controlling it with a 2D mouse. Traditionally, the movement of the 2D mouse is converted to the 3D rotation angles of Pitch and Yaw only (i.e., not considering the Roll angle); most of the images in the OCIS are acquired with the camera's up-vector pointing to the North Pole of the object. In this paper, we propose a new method for browsing an image-based object relying on the augmented "yaw-pitch-roll" strategy. With this strategy, the system first converts the 2D mouse movement to three rotation angles: Pitch, Roll, and Yaw, and then rotates the corresponding image in the OCIS in an appropriate way before displaying it. To evaluate the performance of this strategy, we have conducted a series of experiments to compare the efficiency and precision of this strategy with two other popular strategies. The experimental result shows that the proposed strategy is easier to use for browsing image-based objects and can lead to more efficient and precise results.

1. Introduction

The interactive browsing of virtual objects plays a significant role in both computer graphics and virtual reality (VR). Two common existing techniques developed for constructing a VR world are the geometry-based approach and the image-based approach. The geometry-based approach first constructs 3D models of real world objects and then generates a scene by rendering the 3D models. This approach provides good interactivity but usually requires a tedious way to build the 3D models. On the contrary, the image-based approach uses real

images for interactive displaying and browsing. It provides photo-realistic visual effects and its rendering speed is independent of the complexity of the scenes or objects. In this paper, we focus on the techniques for browsing an object-centered image set (OCIS), which is a collection of images taken from different viewpoints centered on the 3D object and is a popular image-based representation for browsing that object.

The work of browsing an OCIS is highly correlated to that of browsing a geometry-based object model, which will be briefly reviewed here. Chen et al. [2] presented four approaches, the "Sliders", the "Overlapping Sliders", the "XY+Z", and the "Virtual Sphere", respectively, to manipulate the 3D object with a 2D control device, e.g. mouse. The "Sliders" approach lets the user drag any knobs on the sliders to control the orientation of an object by rotating about pre-defined axes. The "Overlapping Sliders" [3] gives a modification of the traditional sliders approach by providing vertical, horizontal, and circular sliders to control the rotations about the x, y, and z axes respectively. The "XY+Z" approach provides two modes of controlling the rotation of an object: 1) rotate the 3D object about the axis which is on the xy-plane of the user's viewer frame and perpendicular to the direction of the mouse movement, 2) rotate the 3D object about the optical axis, z-axis, of the viewer frame while the mouse is depressed outside of the circular slider. The "Virtual Sphere" approach is used to simulate the mechanics of a physical 3D control device, the trackball, which allows the user to rotate the 3D object about all three axes by clicking and dragging inside a circular slider.

Similar to the Virtual Sphere, Shoemaker [10] proposed the "Arcball" approach giving a consistent mapping from a 2D mouse movement to a 3D arc on a pseudo-sphere for specifying the 3D orientation of an object. The advantages of the "Arcball" approach

are that it provides kinesthetic consistency between the mouse motion and the object rotation while avoiding problems due to the *gimbal lock* or any noisy data. Thus users are able to readily and smoothly manipulate the object. Hinckley et al. have made comparisons between the “Virtual Sphere” approach and the “Arcball” approach [5]. The comparison of these manipulation techniques is based on a formal study with a large subject pool.

In essence, most of the existing approaches are developed for the manipulation of geometry-based 3D models. However, very few papers have studied the interactive techniques for manipulating an image-based 3D object, or equivalently, an OCIS. Most of the popular approaches are analogous to sliders-like approaches, which allow users to browse an image-based object by rotating about the x-axis or about x- and y- axes. In this paper, we introduce a different browsing method, called the augmented “yaw-pitch-roll” method, which is similar to the “XY+Z” approach method in [2]. However, our method focuses on the browsing of an OCIS while the “XY+Z” method deals with the rotation of a 3D graphical object. In addition to introducing this new method, we also designed an “orientation-match” problem to evaluate the performance of different object manipulation strategies when used for integrating an OCIS with real or virtual background for augmented reality or augmented virtuality [1][6][7][10].

The remainder of this paper is organized as follows. Section 2 gives a problem description for the interactive browsing of an OCIS. Section 3 discusses some of the basic rotation strategies for browsing an OCIS, that is, the “yaw-pitch” strategy and the “yaw-pitch-roll” strategy. In section 4, we introduce to augmented “yaw-pitch-roll” strategy by adding to the basic “yaw-pitch-roll” strategy, a circular slider for performing the roll method. Section 5 shows the experimental results of comparing different rotation strategies. Finally, a conclusion is drawn in section 6.

2. Problem description

In this work, the images contained in an OCIS are captured with a digital camera mounted on Megallen M-2500. By using this camera mounting device (or other similar devices), the orientation (i.e. the pan and tilt angles) of each viewpoint is roughly known. In order to obtain smooth browsing, we let the center of the 3D object (C_o) be the intersection of the pan- and the tilt- axes of the camera mounting device and let the optical axis of the camera pass through the object center. Then, we can assume that these views are on the surface of a pseudo sphere centered at the C_o , called the *virtual viewing sphere*

(VVS), as illustrated in Figure 1. By choosing an object reference frame (ORF) associated with the VVS, we can fetch any image from the OCIS with the captured direction. For example, if the user specifies a desired viewing direction, as shown in Figure 2(a), the image corresponding to the specified viewing direction will be fetched from the OCIS and then displayed, as shown in Figure 2(b). This process is quite straightforward in showing the desired view when the corresponding viewing direction is given, as illustrated in Figure 2. This process is not quite as simple when the user needs to rotate the image-based object to the desired pose by controlling a 2D input device, and thus presenting the problem we want to solve.

The following four steps are usually required for the browsing of an OCIS:

- Step1: Receive the user input (e.g. a mouse drag).
- Step2: Select the corresponding image to be displayed.
- Step3: Rotate the selected image accordingly.
- Step4: Display the final image.

Before any user operations occur, our system arbitrarily defines an initial viewer reference frame (VRF). Step 1 changes the VRF to where the user wants to observe the 3D object based on the user input. Then, in Step 2, the correct image is selected and shown on the screen as in Step 4, (depicted in Figure 2), or it is rotated appropriately in the following step. Notice that Step 3 is usually neglected in the existing tools, e.g., in QuickTime VR, developed for browsing image-based objects. These tools allow the user to browse a 3D object by rotating it about two fixed axes, pan (yaw) and tilt (pitch), but do not allow the use to rotate it about any arbitrary axis. This approach will be referred to as the “yaw-pitch” approach in this paper. To overcome this drawback, we adopted another rotation strategy, called the “yaw-pitch-roll” approach, which performs a roll motion by rotating the 2D image. Step 3 will prove to be necessary in the following.

A usual practice in acquiring images for an OCIS is to let the up-vector of the camera to always point toward the North Pole of the 3D object. Hence, the indexing of the OCIS allows only a rotation of two degrees of freedom (DOF): yaw and pitch. That is, the indexing of the OCIS itself is not sufficient enough to provide rotations of three DOFs to allow users to observe the 3D object in an arbitrary way. Fortunately, we can perform a 2D image rotation to extend the two DOFs of rotation to three. An example illustrates the difference of the “yaw-pitch” approach and the “yaw-pitch-roll” approach in Figure3. Figure 3(a) shows the initial view, with the mouse starting at position p_s and ending at position p_e . Figure 3(b) shows the intermediate views of the rotation of the 3D object with the “yaw-pitch”

approach and Figure 3(c) shows the rotation with the “yaw-pitch-roll” approach. More details will be described in the next section. Notice that two different modes of the “yaw-pitch” approaches are considered in this paper: the “X and Y” mode and the “X or Y” mode. The example shown in Figure 3 adopts the “X and Y” mode.

3. Basic rotation strategies for browsing an OCIS

In this section, we will discuss two rotation strategies for browsing the OCISs. The first one is a simple rotation approach based on the “yaw-pitch” strategy, which will be described in Section 3.1. The “yaw-pitch” strategy only allows the user to rotate the objects about some fixed axes and them immediately shows the user the image selected from the OCISs. The other is the proposed strategy, named the “yaw-pitch-roll” strategy, which will be described in Section 3.2. The second approach allows the user to rotate an image-based object about an arbitrary axis and is hence more natural for the user. In the following subsections, we shall describe the details of both approaches by following the four steps mentioned in the previous section.

3.1. The “yaw-pitch” approach

For our observations, the “yaw-pitch” approach is the most common approach for the interactive browsing of object movies. It lets the ORF rotate along the “yaw-” and “pitch-” directions. In the following, we imagine rotating the object along these two directions. Since the VVS is imagined to be bound outside the object, it will also rotate correspondingly.

When initializing an OCIS, the object is given an initial orientation. However, the initial orientation of the object is irrelevant because it is simply the starting point for browsing the object. In practice, the initial orientation of the ORF is usually the first shot taken of the object, and this corresponding image is thus shown on the screen.

In the following, we describe each step of this approach.

1) Based on the last image shown on the screen, the user can rotate the object to the desired orientation he/she wishes to inspect. In this step, the manipulating process waits for the user input and then appropriately rotates the object to the new orientation. In most cases, the user input is a two-dimensional movement caused by dragging the mouse on the screen. Traditionally, two different methods are used to interpret the two-dimensional mouse movement. One is the “X and Y” mode that provides the rotation of the object about the X -

axis and then the Y -axis. The other mode is called the “X or Y” mode that provides the rotation about either the X -axis or the Y -axis, but not both simultaneously. It is based on a predefined decision rule. We illustrate these two modes in the following examples, as shown in Figure 4. Consider the mouse movement \vec{m}_1 and \vec{m}_2 dragged by the user, as shown in Figure 4(a) and Figure 4(d). Figures 4(b) and 4(e) show the rotation with “X or Y” mode, and Figures 4(c) and 4(f) show the results of rotation with “X and Y” mode.

I. “X or Y” mode: First, the mouse movement \vec{m}_i is decomposed into two vector components of pitch direction and yaw direction, i.e. \vec{x}_i and \vec{y}_i respectively. Second, the object is rotated about in either the yaw or pitch direction, at a rotation angle proportional to the lengths of \vec{x}_i or \vec{y}_i depending on the rotation axis. In practice, the decision rule compares the lengths of \vec{x}_i and \vec{y}_i , because the longer of the two will become the rotation direction and the proportional ratio of the lengths will become the rotation angle. For example, Figure 4(b) shows that the horizontal component of the mouse drag is more dominant, so the object is rotated in the pitch direction.

II. “X and Y” mode: Similar to the “X or Y” mode, the mouse movement is firstly decomposed into the \vec{x}_i and \vec{y}_i vectors. But this new mode rotates the object in the pitch direction and then in the yaw direction, at a rotation angle proportional to the lengths of \vec{x} and \vec{y} .

2) According to the process stated in step 1, the object should have reached a new orientation after the rotation. We assume that the viewpoint of the viewer frame is located at P on the surface of the VVS so that its viewing direction is the unit vector \overrightarrow{PC}_o . Now, the correct image that presents the 3D object should be in the same viewing direction as \overrightarrow{PC}_o . However, we did not gather all the images, which is continuously distributed on the surface of the VVS, so that the selected image’s viewing direction is closest to \overrightarrow{PC}_o . This is the common method used to approximate a continuous situation.

3) The “yaw-pitch” approach usually does not perform anything in this step. To repeat the above 1 and 2 steps, the users can browse the 3D object by using the acquisition points, as the lattice on the surface of the VVS, by rotating the object with respect to the yaw and pitch direction. This method does not allow the user to browse the 3D object with an arbitrary traveling path. Furthermore, jumping effect occurs when the

VVS's North Pole or South Pole has rotated to the VRF's position shown in Figure 5. Figure 5(a) shows the current view and the corresponding mouse drag. In this example, the object rotates from the front to the back and the North Pole of the VVS rotates across the center of the view. Figures 5(b), 5(c), 5(d), and 5(e) show the intermediate views with the "yaw-pitch" approach ("X or Y" mode), and the unexpected flip occurs between Figures 5(c) and 5(d). Figures 5(f), 5(g), 5(h), and 5(i) show the intermediate depictions of the object's rotation which the user may need. Observably, a popular browser for OCIS must process the unexpected flip automatically.

- 4) Since there is no further processing of the image in step 3, in step 4, we merely show the selected image on the screen.

From our observations, the products on the market almost always use the "yaw-pitch" approach. For example, the Apple QuickTime VR applies the "X and Y" mode and handles the image flip automatically but doesn't support an arbitrary rotation of the image.

3.2. The "yaw-pitch-roll" approach

Although the "yaw-pitch" approach introduced in Section 3.1 is easy to implement, it does not allow the user to browse the object freely because it requires the user to rotate the 3D object along the yaw and pitch directions. For this problem, we introduce a method of rotating the image to an arbitrary orientation so that it allows the users to browse the 3D object freely. We named this approach the "yaw-pitch-roll" because we perform image rotation before displaying the image in Step 4. With this approach, the whole three degrees of freedom is provided so that the user can feel that the rotations are more "natural" when browsing the object. We will follow the previous proposed steps to introduce the process of this approach.

- 1) In this step, when the user drags a mouse from the starting position M_S to the ending position M_E , as shown in Figure 6(a), the rotation axis \vec{R} can be obtained by the following procedure. First, we are only concerned with the mouse drag vector, not the precise starting point and ending point of the drag; we shift the mouse drag by moving M_S to the image center C_I , and the M_E has also shifted to M_E' where the C_I is the image center of the observation viewer frame. Furthermore, the vectors $\overrightarrow{C_O C_I}$ and $\overrightarrow{C_O M_E'}$, which begin with the center of the VVS pointing to the starting and ending points of the mouse drag respectively, can be calculated, as shown in Figure 6(b). Second, the rotation axis \vec{R} is a unit vector that has the same direction as the

vector of the cross product of $\overrightarrow{C_O C_I}$ and $\overrightarrow{C_O M_E'}$.

- In addition, the rotation angle θ should be a scaled proportion of the length of the mouse drag. The rotation axis \vec{R} is parallel to the viewing frame and perpendicular to the mouse drag, such that it is the most intuitive axis for the user's mouse input.
- 2) In this step, the "yaw-pitch-roll" approach select the desired image of the object movie, whose acquisition point is nearest to the viewpoint of the view frame. However, the image browsing may seem to be discontinuous if the displacement between the acquisition points is too large. In order to balance the amount of captured images and the deficiency of discontinuity, the novel view generation can be considered. Light Field Rendering [8] and Lumigraphs [4] are two famous examples, but their large memory requirements make them impractical for real applications, especially for those requiring internet transmission. In addition, Hung et al. also have proposed the disparity-based view morphing to generate the novel views among the three captured images [5]. Though the novel view generation provides the user with more continual browsing, it requires more computation time. From experience, we suggest a) applying the novel view generation for a smaller image scale of the OCISs or b) selecting the closest image to the VRF. By doing so, the image selected also becomes suitable for a larger scale of images.
 - 3) Unlike the "yaw-pitch" approach, where the object is rotated merely with respect to the yaw or pitch directions, the "yaw-pitch-roll" approach provides rotation about any arbitrary axis. Consequently, the up-vectors of the viewer frame and the capturing view are not the same during browsing, i.e. the up-vector of the viewer frame may not always point toward the North Pole of the VVS. Figure 7 shows an example to illustrate this situation. Figure 7(a) shows the orientation of the viewer frame after rotation. Notice that its up-vector is not pointing toward the North Pole of the VVS. Figure 7(b) shows the selected image from the previous step but it is not the correct image, the image that corresponds to the current orientation of the viewer frame. Assume that the up-vector of the capturing view corresponding to the selected image is \vec{u}_c , and the up-vector of the current viewer frame is \vec{u}_r . Therefore, we need to perform an image rotation to get the correct image, as shown in Figure 7(c). In addition, the unexpected flip effect, described in the "yaw-pitch" approach, easily allows the user to rotate the object about the optical axis of the viewer frame with a circular slider. Similar to the above sequence, along with this additional device, the "yaw-pitch-roll" approach allows the user to rotate the object rather

easily. This approach will be further discussed in section 5. In addition to this advantage, this approach can reduce the number of images required of the object. The images located on the North or South Pole can be reduced to merely one image because the other orientation images can be generated by this one image through image rotation.

- 4) After rolling the image appropriately, we display the rolled image on the screen.

4. Augmented rotation strategies

In the previous section, we have introduced the “yaw-pitch-roll” strategy that allows the user to browse an image-based object in an arbitrary way, rather than rotating it about the fixed axes in the yaw- and the pitch- directions. This rotation strategy is very useful in augmented reality [6][7][10] for inserting an OCIS into a virtual/real environment with a desired 3D object pose. Consider the example shown in Figure 8, where the user wants to rotate the 3D object about the roll axis only. In this case, the user may have to drag the mouse many times before achieving the goal if he/she adopts the “yaw-pitch-roll” strategy. It should be noted that it is impossible to accomplish the task if the user adopts the “yaw-pitch” strategy. To improve the efficiency of 3D object rotation, we add a circular slider into our browser so that the user can perform merely the roll motion by dragging the mouse outside the circular slider. If the mouse drag is inside the circular slider, the rotation is performed by using the basic “yaw-pitch-roll” strategy. The combination of the “yaw-pitch-roll” strategy with the mechanism of the circular slider will now be referred to as the *augmented “yaw-pitch-roll” strategy*.

It is also possible to combine the circular slider with the “yaw-pitch” strategy, which will then lead to the augmented “yaw-pitch” strategy. Notice that when using the augmented “yaw-pitch” strategy, the 2D image rotation will have to be performed once a pure roll motion is performed.

Figure 9 shows how the circular slider is used to increase the performance of browsing an image-based object. In this example, Figure 9(a) shows the target pose, and 9(b) shows the initial pose. After starting to browse, the user may drag the mouse insides the circular slider just along the horizontal direction, as shown in 9(c), and the rotation result corresponding to the mouse movement is shown in 9(d). Notice that the mouse movement only has a x component in this example so that all the rotation strategies of the “yaw-pitch” approach and the “yaw-pitch-roll” approach produce the same results. After rotating the object along with horizontal direction, the user can drag the mouse outside the circular

slider to rotate the object about the roll axis, as shown in Figure 9(e). Therefore, the user can make this an easy task, as shown in Figure 9(f).

5. Experiments

In this section, we present the results from evaluating the rotation strategies introduced in this paper. To evaluate the rotation strategies we have conducted a series of experiments to compare the efficiency and precision achieved by the different rotation strategies. In these experiments, each user was asked to solve a series of “orientation-match” problems by rotating an image-based object, as the example shows in Figure 9. For each problem, an image associated with a particular pose (i.e. the “target” pose) was selected from the OCIS and presented to a user. The user was then asked to rotate the object from the initial pose to the target pose by using a pre-specified rotation strategy, which include any of the following three strategies: the two “yaw-pitch” approaches and the “yaw-pitch-roll” approach. After the user rotates the object, and she/he feels that the newly generated object pose and the particular object pose are correctly matched, the user can then submit the results. After the user submits his/her results, the time taken by the user and the accuracy of the match are recorded. These results can then be used to compare the efficiency and precision achieved by these different strategies. Such an experimental design is based on the assumption that a good rotation strategy should allow the user to rotate the object from an initial orientation to a different orientation as fast and as precise as possible. In fact, such an orientation-matching capability is also important when the user needs to insert an OCIS into an image or a panorama for mixed or augmented reality applications [1][6][9].

5.1. Selection of problems

Since different orientation-match problems may have different levels of difficulty, we have classified the problems into three categories: easy, average, and difficult. Different passing criteria and different time limits are set for problems of varying levels. To classify and select the problems, sixteen problems were randomly generated at first. In order to decide the difficulty level of each problem and to eliminate problems that were too difficult or too easy, we invited four expert users to take the test and aid in the elimination process. The following equation is used to compute a difficulty level factor, D_i , which is the summation of both the manipulation time T_i and the manipulation error E_i :

$$D_i = T_i + E_i, i = 1, \dots, 16$$

Table 1. Passing criteria and time limits set for problems of different level of difficulties

	Easy Level	Average Level	Difficult Level
Passing Criterion	We set a relatively loose threshold for the error tolerance.	No error threshold is set, a user can submit his/her results when he/she felt that the result was good enough.	We set a relatively stricter threshold for the error tolerance.
Time Limit	N/A	N/A	60 seconds.

After computing D_i from each of the four users' manipulation results, we eliminated two problems with the maximally average D_i and two others with the minimally average D_i . The remaining twelve problems were then indexed in an ascending order with respect to the D_i . Problems indexed with 1-4, 5-8, and 9-12 were then classified as those of difficult, average, and easy levels, respectively. Then, the twelve problems were used to test other users. In particular, the passing criterion and time limit of each level are set differently, as shown in Table 1.

We introduce a subject pool containing eight experienced users and twenty-four ordinary users. To eliminate outside factors, we chose the subjects from both genders and from all ages. In addition, the order of the three browsing strategies presented to users is randomly generated. After answering all the questions, the users were also asked to express their subjective preferences of the browsing strategies for further subject-opinion analyses.

5.2. Evaluation results

From Figure 10, we can see that the distribution for augmented the "yaw-pitch-roll" approach is obviously more condense and closer to the original point. This means that the augmented "yaw-pitch-roll" approach is more efficient and accurate than the other two strategies. Figure 11 and Figure 12 show that for both expert users and ordinary users, the average manipulation time of the augmented "yaw-pitch-roll" approach is faster than the other two strategies. The standard deviation of the manipulation time is also smaller than those of the other two strategies. Figure 13 and Figure 14 show that the precision achieved by the augmented "yaw-pitch-roll" approach is also better than those of the other two strategies for both expert and ordinary users. For the ordinary users, the standard deviation of the "X or Y" mode is obviously larger than the standard deviation of the "X and Y" mode. This therefore means that many users are inept at using the "X or Y" mode.

We set sixty seconds as the time limit in the third step of our experiment. Figure 15 shows that fewer users failed in answering the questions within sixty seconds while using the augmented "yaw-pitch-roll"

approach. Figure 16 shows that when asked for their subjective opinions, most users think that the usability of the augmented "yaw-pitch-roll" approach is better than the other two strategies. When choosing between "X and Y" mode and "X or Y" mode, slightly more users prefer the "X and Y" mode to the "X or Y" mode.

6. Conclusion

In this paper, we have proposed a new method for browsing an Object-Centered Image Set based on the augmented "yaw-pitch-roll" approach analogous to the "XY+Z" approach that has been used to manipulate 3D geometry-based objects mentioned in [2]. Notice that for a geometry-based object, we have the full 3D information of the object for rendering the image. However, this is not the case for image-based objects, and thus we must approximate the desired view by first selecting a corresponding image from the OCIS and then rotating it accordingly. To compare our method with other popular "yaw-pitch" methods, we have designed a set of "orientation-matching" tests to evaluate the performance of different browsing method. The experimental results have shown that the augmented "yaw-pitch-roll" method is clearly better than the other two methods in terms of manipulation speed and precision. Additionally, more users prefer the augmented "yaw-pitch-roll" method, which is consistent with the experimental results

References

- [1]. C. S. Chen, C. K. Yu, and Y. P. Hung, "New Calibration-free Approach for Augmented Reality Based on Parameterized Cuboid Structure," *Proc. 7th Int'l Conf. on Computer Vision (ICCV 99)*, IEEE CS, Corfu, Greece, 1999, pp. 23-37.
- [2]. M. Chen, S. J. Mountford, and A. Sellen, "A Study in Interactive 3-D Rotation Using 2-D Control Devices," *Proceedings of Computer Graphics (SIGGRAPH '88)*, 1988, pp. 121-129.
- [3]. M. Chen, "A Technique for Specifying Rotations in Three Dimensions Using a 2D

- Input Device,” *Proceedings of IEEE Montech’87 – Compint’87*, 1987, pp. 118-120.
- [4]. S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, “The Lumigraph,” *Computer Graphics (SIGGRAPH 96)*, ACM Press, New Orleans, Louisiana, 1996, pp. 43-54.
- [5]. K. Hinkley, J. Tullio, R. Pausch, D. Proffitt, and N. Kassell. “Usability Analysis of 3D Rotation Techniques,” *Proceedings of ACM Symposium on User Interface Software and Technology (UIST ’97)*, 1997, pp. 1-10.
- [6]. Y.-P. Hung, C.-S. Chen, Y.-P. Tsai, S.-W. Lin, “Augmenting Panorama with Object Movies by Generating Novel View with Disparity-Based View Morphing,” to appear in *The Journal of Visualization and Computer Animation* 2002.
- [7]. A. Katayama, Y. Sakagawa, H. Yamamoto, and H. Tamura, “Shading and Shadow-Casting in Image-Based Rendering without Geometric Models,” *Conference Abstracts and Applications (SIGGRAPH ’99)*, 1999, pp. 275.
- [8]. M. Levoy and P. Hanrahan, “Light Field Rendering,” *Proc. Computer Graphics (SIGGRAPH 96)*, ACM Press, New Orleans, Louisiana, 1996, pp.31-42.
- [9]. Y. Seo, and K. S. Hong, “Calibration-Free Augmented Reality in Perspective”, *IEEE Trans. on Visualization and Computer Graphics*, Vol. 6, No. 4, pp. 346-358, Oct.-Dec., 2000.
- [10]. K. Shoemake "ARCBALL: A User Interface for Specifying Three-Dimensional Orientation Using a Mouse," *Proceedings of Graphics Interface (GI '92)*, 1992, pp. 151-156.
- [11]. S. Uchiyama, H. Yamamoto, A. Katayama, and H. Tamura, “Presentation and Interaction of Virtual 3D Objects without Geometric Model,” *Proceedings of International Conference on Human-Computer Interaction (HCI '97)*, Vol. 2, 1997, pp. 869-872.

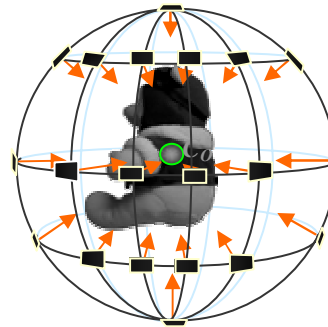


Figure 1: An illustration of the 3D configuration of multi-perspective views. These views can be thought of as being on the surface of a pseudo sphere, called the virtual view sphere (VVS).

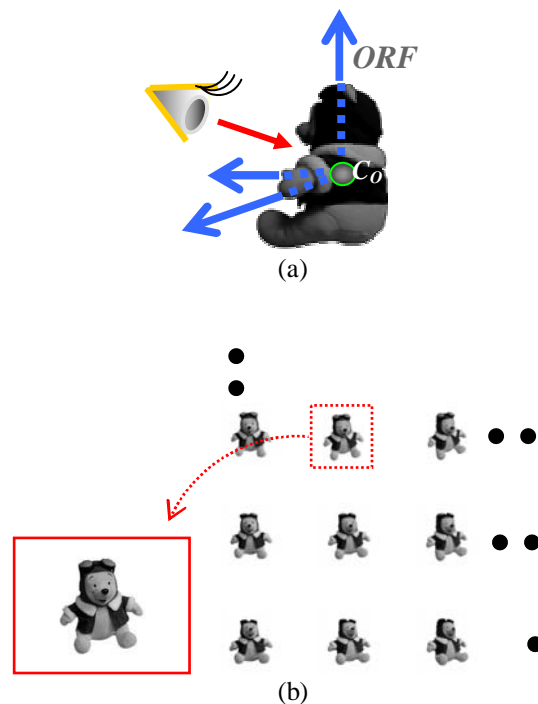


Figure 2: When the user specifies a viewing direction, as shown in (a), the correct image corresponding to the viewing direction can be fetched and displayed on the screen, as shown in (b).

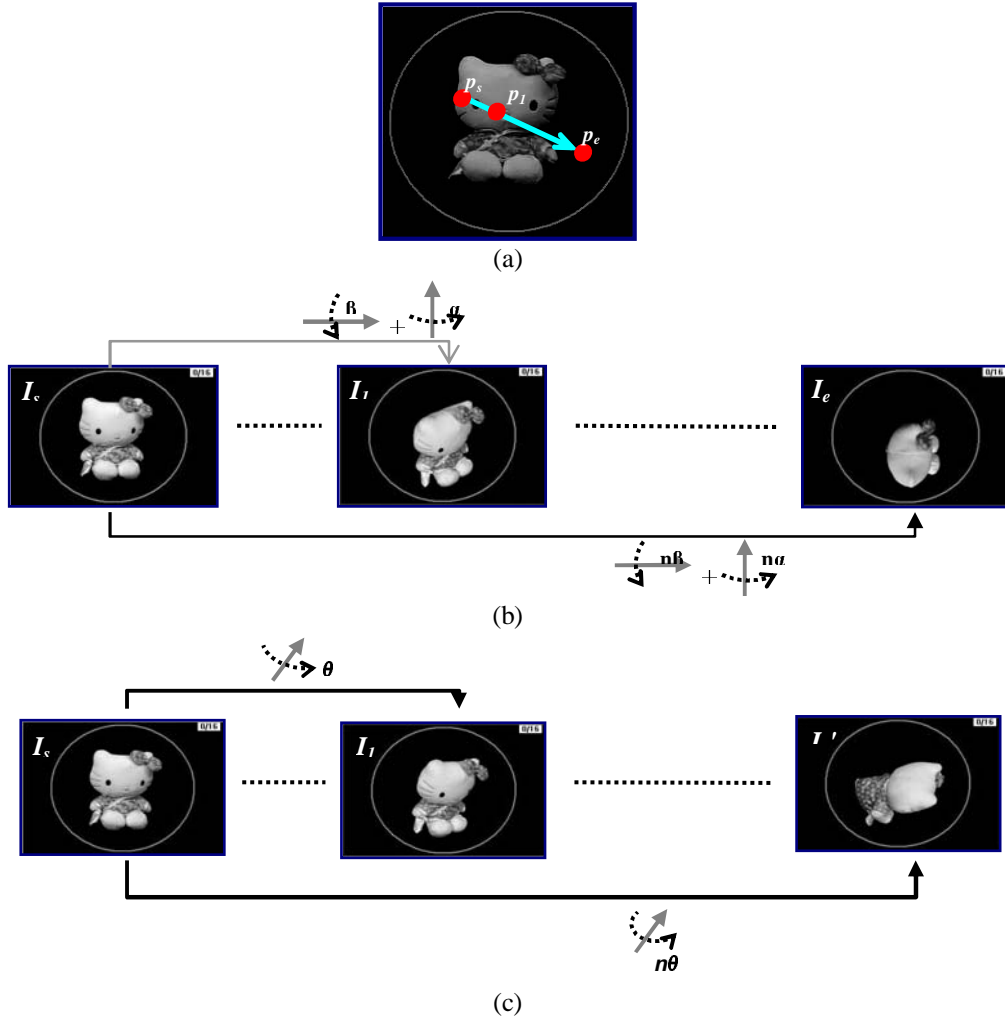


Figure 3: An example of browsing an OCSI by applying different manipulation strategies. (a) shows current view of the object and mouse motion which starts from position p_s and ends at position p_e . (b) shows few views according to the mouse position p_s , p_1 , and p_e using the “yaw-pitch” approach with “X and Y” mode. (c) shows the intermediate views using the “yaw-pitch-roll” approach. Compare with (b), the augmented “yaw-pitch-roll” seems more “nature” for browsing the object.

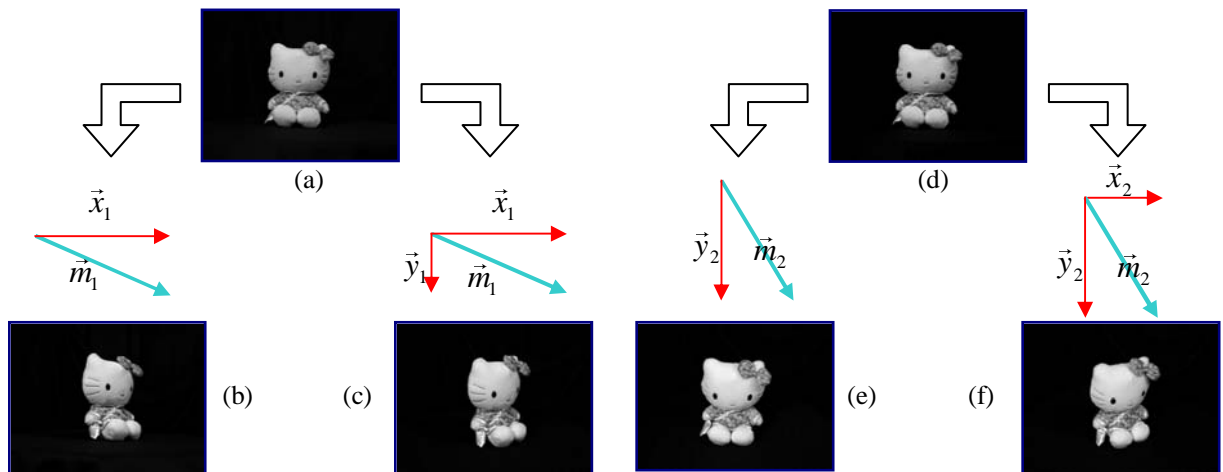


Figure 4: An example of comparison between “X or Y” mode and “X and Y” mode. (a) and (d) show the current view and two mouse movements, \vec{m}_1 and \vec{m}_2 . (b) and (e) show the rotation result with “X or Y” mode, respectively. (c) and (f) show the rotation result with “X and Y” mode. In (b), the length of \vec{x}_1 is longer than \vec{y}_1 so that the object was just rotated in pitch direction by the angle that is proportional of the length of \vec{x}_1 .

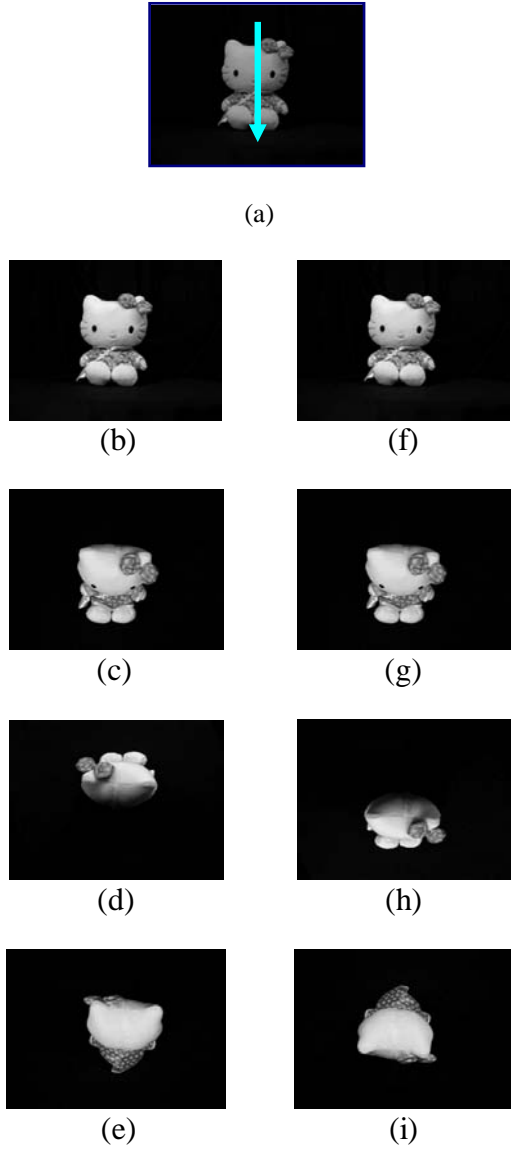


Figure5: An example of showing the drawback of “yaw-pitch” approach while the user browses the 3D object from the front to back and across the North Pole of the object. (a) show the current view and a mouse movement. (b), (c), (d), and (e) show the intermediate views of object’s rotation with “yaw-pitch” approach, (f), (g), (h), and (i) show the intermediate views of rotation which one may expect. Obviously the unexpected flip occurs between (c) and (d).

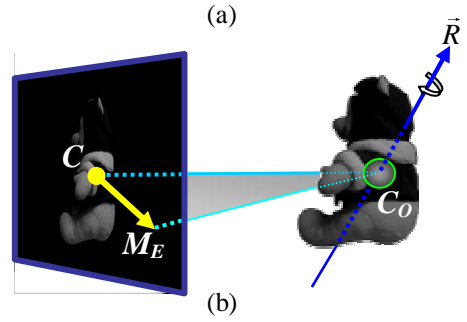
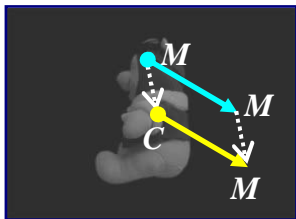


Figure 6: Illustration of calculating rotation axis \vec{R} used in “Nature Mode” approach. (a) shows the current view and mouse motion from M_S to M_E and first step to perform calculation, which shifts the mouse motion to $\overline{C_I M_E'}$. (b) shows the 3D configuration of current viewer frame and the 3D object. Second step of calculating \vec{R} is to compute the directions, $\overline{C_O C_I}$ and $\overline{C_O M_E'}$, and then \vec{R} will be the unit vector of $\overline{C_O C_I} \times \overline{C_O M_E'}$.

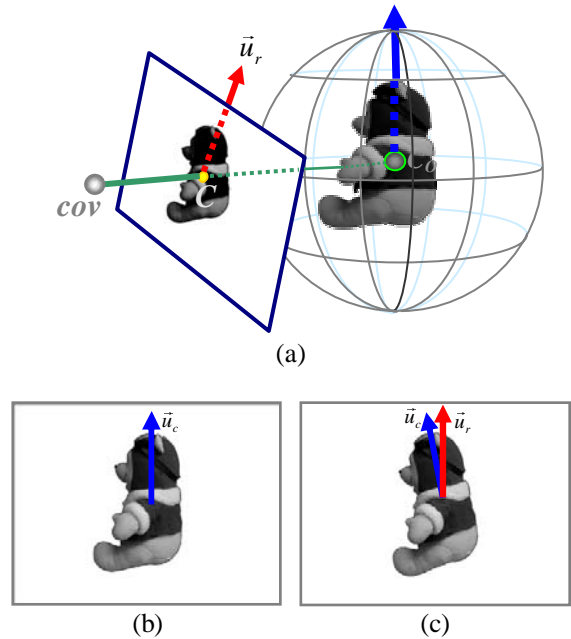


Figure 7: An example of showing image rotation process in “Nature Mode” approach. (a) shows the 3D configuration of viewer frame and VVS. The cov is the center of view. \vec{u}_c and \vec{u}_r are the up-vector of capturing view and current view, respectively. (b) shows the selected image, and (c) shows the result after image rotation by the included angle of \vec{u}_c and \vec{u}_r .

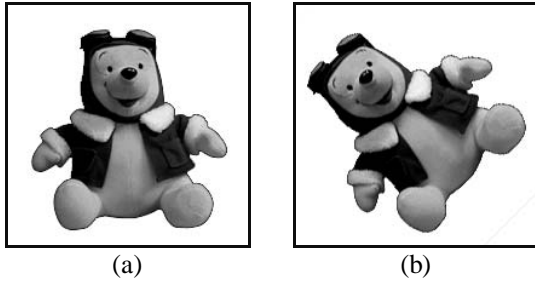


Figure 8: An example of object's rotation about roll axis only. (a) shows the initial pose. (b) shows the target which the user desires.

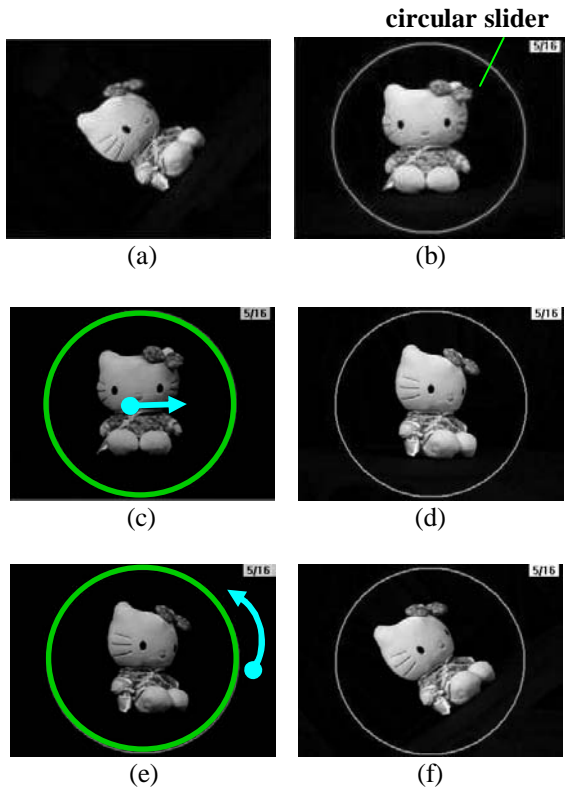


Figure 9: An example using the circular slider to perform the roll motion. (a) shows the target pose and (b) shows the initial pose. (d) and (f) shows the rotation of the object according to the user input shown in (c) and (e).

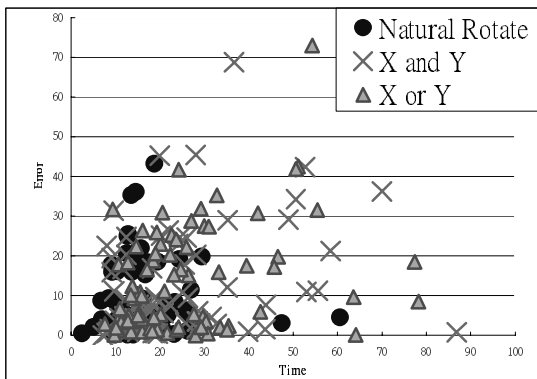


Figure 10

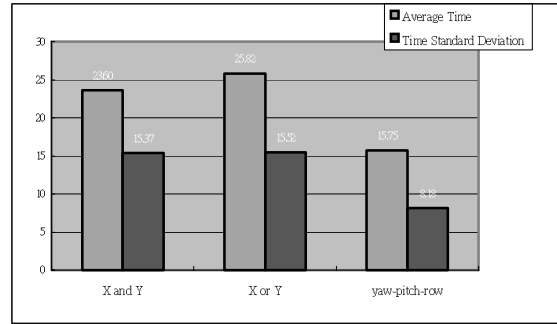


Figure 11

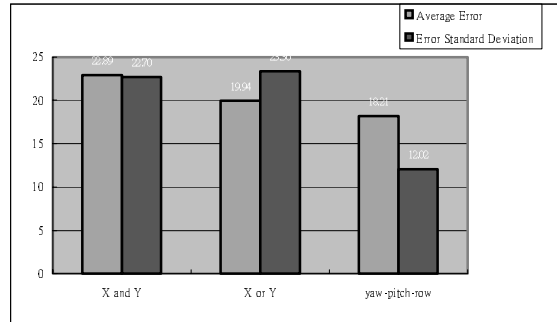


Figure 12

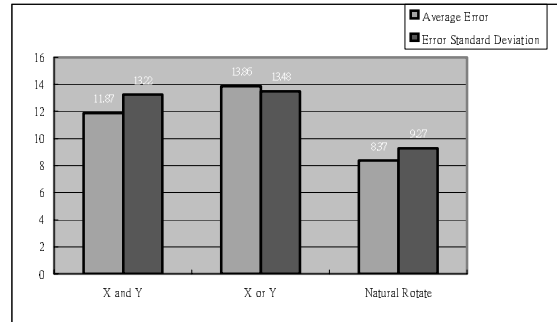


Figure 13

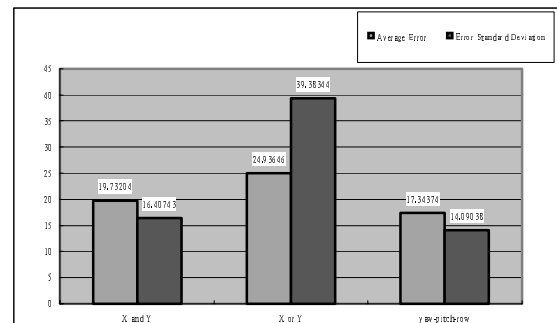


Figure 14

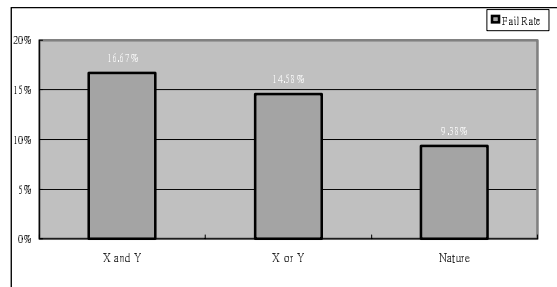


Figure 15