http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (5): 672-680, 2007
ISSN 1812-5638

Simulation of the Handling of Real Objects with a Complete Control of Rotation

[1,2]Brahim Nini and [2]Mohamed Batouche
[1]Larbi Ben M'hidi University, 04000 Oum El-bouaghi, Algeria
[2]Faculty of Engineering, LIRE Laboratory, Vision and Computer Graphics Group,
Mentouri University, 25000 Constantine, Algeria

Abstract: This study describes a complete movie-based solution for simulating the manipulation of a real object in three rotational degrees of freedom. The solution uses a linear structure to store the frames corresponding to different viewing directions of the object. Simulating rotations is then equivalent to orbiting a virtual camera around the object and projecting what it is supposed to grab. The method links these rotations with those of the real camera while grabbing in order to extract the corresponding frames. This correspondence is geometry-based and expressed in an analytical way. The solution is entirely parameterised and independent of the number of taken images. Consequently, this study is an assessment of the mathematical background of an image-based manipulation using a specific organization both in the grab and storage of the images. In fact, the research provides a complement matter to some works which detail explicit mathematical concepts related to scene navigation but, in general, they do not emphasize the ones related to object manipulation.

Key words: Image-based, manipulation, storage, 3D geometry

INTRODUCTION

The manipulation of virtual objects is an important requirement for many applications. The rendering may use either synthetic or real objects. Typically, there are some particular situations where real objects, whose rendering is image-based, are more suitable than synthetic ones.

Despite recent advances, the modeling of complex real objects is still difficult and rendering textured models usually does not result in realistic images. Clearly, rendered images do not look as realistic as photographs which easily capture complex geometric texture and global illumination. In addition, the more complex the virtual environment is, the more the simulation of navigation or object movements hardly appear natural. The difficulty is that all differently positioned and orientated views with related details should be computed from geometric models. The impact is that the considerable associated time of the rendering would negatively affect the length of a process made up of many steps. Furthermore, despite the use of special hardware 3D rendering engines in some works to reach real-time rendering of 3D entities, many difficulties are still encountered. In fact, the 3D modeling and rendering approach presents three main problems in spite of hardware acceleration technologies advance. First, creating 3D geometrical entities relies on a difficult manual process. Second, the rendering engine usually sets a limit to object complexity and rendering quality while they have really no upper bound. Third, although its price is accessible, professional hardware is still not easily affordable. One solution to bypass these difficulties is to create a model of the object by capturing its shape and surface reflectance properties. Obviously, photographs are the high-quality renderings of surfaces.

Over the past few years, Image-based Rendering (IBR) techniques have been investigated by many researchers as an alternative to some conventional 3D model-based rendering. Instead of modeling scenes and/or objects, these are photographed from reality. The aim of IBR systems is the use of input pictures as the basic primitive to synthesize virtual images. It has the advantages of higher photorealism potential and rendering speed being independent of scene complexity. For instance, the automatic change of the level of detail in IBR systems can be easily achieved through multiple level resolutions.

Other systems, extending from the previous ones, with the principle of Image-space-based rendering, have been used to navigate in virtual environments modelled from real-world digitized images. These systems belong particularly to virtual reality domain. Enough photographs of a scene or object should be taken so that every

Corresponding Author: Brahim Nini, Faculty of Engineering, LIRE Laboratory, Vision and Computer Graphics Group, Mentouri University, 25000 Constantine, Algeria Tel: +21372874681 Fax: +21332421036

potentially visible surface to a user is captured at least once. Most of the proposed solutions for photograph storage the user is intended to walkthrough are movie-based. During the navigation, s/he is restricted to move through either current images or interpolated ones. Nevertheless, in both situations, the user is restricted to move through existent connections or branches that link photographs or movies together.

Several works have been achieved through this way. An early example based on other previous works was Quick time VR (Chen, 1995) which suggests that the traditional modeling/rendering process can be skipped. Instead, a series of captured environment maps allow a user to look around a scene from fixed points in space. The work applies an approach which uses 360-degree cylindrical panoramic images to compose a virtual environment. For the specific case of an object rotation, the movie contains a two-dimensional array of frames which correspond to the viewing directions of all the allowable orientations of the object. Those of Gortler *et al.* (1996) and Levoy and Hanrahan (1996) are other works which do not rely on geometric representations; however, they require a huge number of images. They sample and reconstruct a 4D function that generates new images of the object independent of the geometric or illumination complexity. Pulli *et al.* (1997) describe a method for displaying scanned real objects. It lies in between purely model-based and image-based methods. Another important method presented by Wang *et al.* (2002) allows objects to be visualized with continuous and interactive changing viewpoints. Its limit, however, is a viewpoint restriction. Note that most of works deal with the view interpolation method which stores only a few key frames and synthesizes the missing frames by interpolation.

Most of studies in this field are oriented scenes-representation. For instance, Darsa *et al.* (1997) give a presentation of a z-buffered image-space-based rendering technique that allows navigation in complex static environments. Popescu *et al.* (2000) designed WarpEngine, which is a 3D graphic hardware architecture for real-time image-based rendering of natural scenes from arbitrary viewpoints by warping images with depth. In this work, objects are rendered with the scene but not apart; they cannot be isolated. Lately, hybrid methods have been setup for the purpose of reaching realistic views with less human cost. The study of Ono *et al.* (2005) discusses one way of how to mix between IBR and GBR (Geometry Based Rendering) in a driving simulation system for traffic experiment space. On another side, Bakstein and Pajdla (2006) present an overview of the practical implementation of an IBR system for real complex photorealistic representation scenes.

This study adds a new brick to the previous ones so that it focuses on the manipulation of an object in the context of IBR systems. Whereas the manipulation is entirely mastered for synthetic objects, there are still some difficulties in connecting between limited number of images and all infinite orientations that can be inquired. This work gives a general solution to this connection. The solution covers all possible and natural orientations which are independent from the number of images taken. Moreover, the work takes advantage of photorealism and rendering speeds. Although it does not interpolate new images, the user may feel natural when manipulating the object.

To provide the manipulation, projected images should agree with user's requests in a given relationship. Primarily, the object is photographed following a given trajectory. When a user requests to simulate the same trajectory, this is simply reached through the projection of the images without any transformation. However, this ideal situation is seldom encountered and a general solution should be established. Unfortunately, many works detail explicit mathematical concepts in relation to scene navigation, but, usually, they do not emphasize the ones related to object manipulation though the concepts are important to both scenes and objects. This is what this research establishes for objects. A storage method using a linear structure for generated images from the photographed object is stated first. Then, the sufficiency of the method to cover either simple or multiple resolutions is elucidated. Finally, the analytical expressions of the geometrical rotations are given to make a mathematical relationship between required simulations of infinite virtual object orientations and the stored images.

In addition, this study extends the one presented by Nini and Batouche (2005). Because of the high overall time spent in augmenting a scene due to complex computation, the rendering time of complex virtual objects, as a part of a whole process, should be minimized. It is one among many other features which one can act on to recover this overall-time. Furthermore, the manipulation of virtual objects is amid some actions the system should allow the user to do interactively in real time. This is why this solution involves high improvement to augmented reality systems. It is so relevant to many applications that carry out this mixture between real scenes and virtual objects. In some study, for instance, virtual objects constitute a tool for the purpose of making the achievement of certain tasks easier (Braun, 2003). On another side, Zhong *et al.* (2002) describes an industrial maintenance where the virtual object is a guide for the repair operation and to improve the augmentation process, Nini and Batouche (2005) present an approach where virtual objects are image-based.

Based on the previous principles, the research shows how it is possible to simulate the manipulation of an object through its images in three rotational degrees of freedom. However, like almost all pieces of research, some conditions of use should accompany some results. Therefore, to give details about these results and their relative constraints, the linear structure of the storage is presented in the next section. It states the order in which images should be organized as a movie. All the geometric aspects and their relative analytical expressions are detailed in the succeeding section. It establishes the mathematical basis of how the linking between images and orientations is done. The relation of these results to the previous ones and some features are outlined in the following section and, finally, a conclusion summarises this work and gives future orientations.

IMAGES ORGANIZATION

An object rotation simulation requires a set of images. The way they are organized is important for their storage and retrieval. The chosen solution is a linear structure which is stored in a movie file.

Visualizing an object in different orientations requires photographing it primarily by a camera. Typically, the camera's motion has six degrees of freedom grouped into three classes (Chen, 1995). First, the camera rotation refers to rotating the camera's view direction while keeping the viewpoint stationary. Second, an object rotation refers to orbiting the camera while keeping its view direction centered at the object. This is equivalent to rotating the object itself. Third, the camera movement refers to the free motion of the camera in a given environment. In this case, both the viewpoint and viewing direction change. The first and third classes are related essentially to scenes though they can combine to object motion via some ways. Particularly, this study leads with the second class of motions. Practically, the camera orbits around the object in two directions at a constant increment and points to its center (Fig. 1) though photographing physical objects in this way is very challenging without special apparatuses. Each generated image is linked to its specific viewpoint. Therefore, in order to visualize a virtual rotation of the object, a particular navigation through this set of images should be made so that it reflects a specific user requirement.

Besides the quality of photographs, a jump from one image to another plays an important role in rotation realism. The similarities of two walkthrough images, which depend on the grab step, influence the apparent natural movement. Hence, to provide a smooth rotation, the increment should be as small as possible. Unfortunately, this leads to the production of a great number of images.

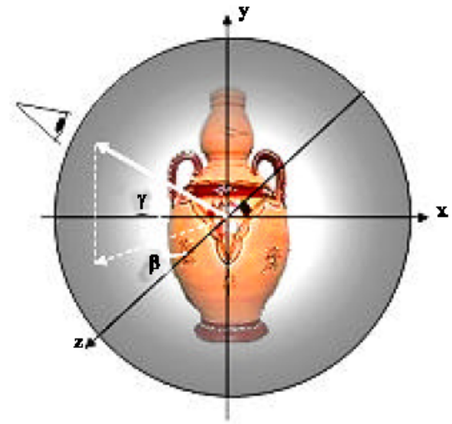

Fig. 1: The object is photographed at equal distances by a camera viewpoint directed to its center. Each position is referenced by two angles: For each value of γ varying in $[-\pi, +\pi]$, β varies in $[0, 2\pi[$

The object is photographed with respect of two criterions. First, the object frames are created with a constant colour background to facilitate compositing onto other backgrounds. Normally, each image is processed before being stored so the object's region becomes easy to identify. Second, each image is linked to a camera viewpoint referenced by its spherical coordinates (γ, β, d) in the object's reference frame during the grab operation. Both angles γ and β should be multiples of the increment which is a chosen constant st. The numerical value d is important for the purpose of providing a multiple resolution system. It represents the distance separating the camera from the object center and it should be the same for all taken photographs of a given resolution. Note that one image of the object should be the reference view having $\gamma = 0$ and $\beta = 0$. For this objective, the value of st should be of the form $\pi/2.a$, where a is an integer which values are positive; otherwise, this position cannot be reached.

Taken photographs are stored in a movie AVI file type as a linear structure which has an organization directly dependent on the way the object is grabbed. The order of images is chosen so that it makes their access as fast as real time simulation requirement is. Knowing that the simulation of a rotation selects always one of the surrounding grabbed frames of the current viewpoint, images belonging to a same plane, i.e., γ has the same value, are stored consecutively following the order of their grabbing progression from the back to the top. Namely, having $\beta = n.st$ and $\gamma = k.st$ the angles of the camera position with reference to yz and xz planes, respectively, it is possible to compute the position p where the generated image should be stored in the video stream as:

$$p = n + \left(2k + \frac{\pi}{st}\right)\frac{\pi}{st} \quad (1)$$

Expression (1) is obtained empirically for the storage of images corresponding to just one resolution. Particularly, the γ and β angles should take their values in [-π,+π] and [0, 2π], respectively and consequently, the values of k are in [-π/2.st, π/2.st] and those of n are in [0, 2π/st]. As a result, the image of the reference view is stored at position $p = \pi^2/st^2$ where n = 0 and k = 0. Furthermore, the values of k in [0,π/2.*st*] are for the images of top views and those in [-π/2.st, 0] are for bottom ones.

Although there are several issues for the storage of images belonging to multiple resolution systems (Chen, 1995; Gortler *et al*., 1996; Levoy and Hanrahan, 1996), this work simply extends the linear structure organization to provide this multi-resolution. We take advantage of expression (1) by gathering, in the same file, all the sets of images related to different resolutions. Simply, each image is stored at position r.s+p. That is, r is the resolution level which values are greater than 0. When only one level is used, it should be 0. Furthermore, it is easy to establish that the quantities 2π/st and 1+π/st are the numbers of all viewpoints when grabbing the object in longitude and latitude directions, respectively. Thus, the value of

$$s = \frac{2\pi}{st}\left(1 + \frac{\pi}{st}\right)$$

is used in order to go throughout all images of each level. Consequently, expression (1) becomes:

$$p = n + \left(2k + 2r\left(1 + \frac{\pi}{st}\right) + \frac{\pi}{st}\right)\frac{\pi}{st} \quad (2)$$

GEOMETRIC ASPECTS OF IMAGE-BASED MANIPULATION

The simulation of an object manipulation is achieved through navigating into the set of its images and rendering the one that reflects current orientation of the user's desire. Indeed, user's requests should be transformed to a positioning of a virtual camera. In response to these playback queries, the system should link between taken images and the virtual camera's viewpoints. Since the images have been generated by the real camera from particular orientations, the chosen one is simply projected after having been oriented in agreement with the current virtual viewing direction.

In order to link grabbed images to users' requirements, we use two particular orthonormal right-handed 3D coordinate systems (Fig. 2). When the object rotates, this goes together with the rotation of these two reference frames but in opposite directions. One

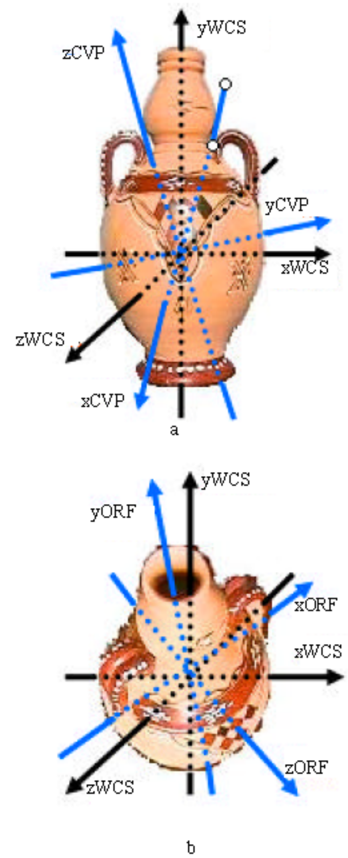

Fig. 2: The two system frames used in manipulation control. (a) reflects the rotation of CVP that corresponds to the view of (b) where ORF is oriented with the object

of the two is associated to the object itself. Typically, it is the Object's Reference Frame (ORF) which makes the same rotations of the object. It reflects exactly any current object's orientation. Its z axis is always oriented to the reference image. The second coordinate system simulates the virtual camera direction so that its z axis is always facing the virtual camera viewpoint (CVP) when orbiting around the object. This direction provides the image that reflects the current view.

When simulating object rotations, CVP rotates inside the object which is assumed to be motionless. In fact, rotating the object around its center is equivalent to orbiting the camera around it. More precisely, CVP rotations are equivalent to making the environment move around the object whilst having an eye on it. It is why CVP rotates in the opposite direction of that of ORF. Thus, the spherical coordinates of its *z* axis give required information to retrieve the grabbed image at such position.

From a geometrical point of view, there are three coordinate systems linked together into a relative rotational relationship that should be expressed analytically. Explicitly, the two previous ones are rotating in the space of the World's Coordinate System (WCS) which is assumed fix. Hence, the analytical expression of the relationship should enable us to locate the image which corresponds to a user's request and project it in a correct manner. This means that we ought to express combined rotations of the object around arbitrary axes because the aim of the system is to allow the object to be viewed at any direction. Namely, two rotation matrices are associated to CVP and ORF called, respectively R_v and R_0. They are updated in the same way in which CVP and ORF rotate. So, in order to assess these principles and establish a general analytical relationship, we begin by explaining the particular case of the first grab operation and then present an analysis of how these systems work together.

First grab operation: It is possible to state two different but equivalent thoughts depending on the fact that either the camera or the object is rotating during the first grab operation. If the object is fixed, it is evident that the camera rotates about the y axis of WCS for the longitudinal grab and the x axis of CVP, which is rotating too, for the latitude grab. If the camera is fixed, the object should rotate about the y and x axes of, respectively ORF and WCS in order to be grabbed. In both cases, ORF rotates at the same time with the object. Hence, projecting successive images in the order of their grabbing in a longitudinal direction is equivalent to rotating the object about its y axis, i.e., ORF rotates about its own *y* axis and CVP about the y axis of WCS. In the same way, when projecting images in latitude direction, this makes it rotating about the x axis of WCS and CVP about its own x axis. For both cases, ORF rotates in opposite direction comparatively to CVP and provides an object rotating in the opposite direction of the camera. Finally, rotating the object about its z axis is simply the rotation of the projected image. We can conclude that when CVP rotates about WCS's axes, ORF rotates in opposite direction about its own axes and vice versa. These are the followed rotations to project the object in the same way it was grabbed.

Rotating the object in the previous way is reached by giving different values to n or k parameters; although, it remains insufficient. These rotations, with that about the z axis of ORF, were the only ones allowed in the previous study presented by Nini and Batouche (2005); unfortunately, they do not cover the three rotational degrees of freedom because they are limited to the previous specific rotations. Specifically, they are the *y* and x axes of ORF and CVP, respectively. There are many situations where changing the values of n or k does not provide expected views. A problem arises when the object is already rotated somehow with a combination of some particular rotations. This is why the established conclusion should be generalised to the other axes.

Object rotations about ORF's axes: As ORF rotates with the object, rotations about its axes referenced in WCS should be assessed. They can be considered made about arbitrary axes where each one lies simply along one of the ORF's. Since ORF is represented by the R_0 matrix, the orientation of the object can be evaluated while rotating ORF about the requested axis of R_0. In other terms, rotating the object about one ORF's axis is considered to be about a particular axis which is one of the R_0 columns. Specifically, knowing that the columns of rotation matrices represent the coordinates in the rotated space of unit vectors along the axes of the original space, R_0 is simply rotated about $R_{0_{axis}}$ which is either the first, second, or third column of R_0 depending on the value of axis, which may be, respectively x, y, or z, namely:

$$R_0 = R_{0_{axis}}^{\alpha} \cdot R_0 \tag{3}$$

The notation $R_{0_{axis}}^{\alpha}$ means $R_v^{-1} R_y^{-1} R_z R_y R_x$. The expression lies the z axis of WCS along o_{axis} ($R_y R_x$), i.e., the column in R_0 corresponding to axis, rotates it by α (R_z) and restore the system to its initial position ($R_v^{-1} R_y^{-1}$).

Furthermore, when ORF rotates, CVP should do so. Although the thought about ORF's axes is so easy because of its natural behaviour related to the visual aspect of the object, some difficult viewing directions of CVP have to be found in relation to ORF rotations. The most evident is, based on the previous subsection, the rotation of CVP about the y axis of WCS when the object rotates about its y axis. Consequently, rotations about the x and z axes remain to be understood.

To find the relation that links rotations of ORF about its axes to those of CVP, one can verify empirically that CVP rotates about the same WCS's axes for the purpose of pointing to expected camera viewpoint direction. Typically, like ORF, CVP rotates around the same axis but after having been transformed by R_v which is the inverse of R_0 in this case. This way of thinking can also be based on the established conclusion in the previous subsection because the rotation is about of the ORF's axes. Specifically, when the object is required to rotate α degrees about one of the ORF's axes, CVP should make a rotation by $-\alpha$ about the same WCS's axis. In matrix form, rotating CVP about WCS's axes is equivalent to multiply the R_v matrix by a rotation matrix about one axis, namely:

$$R_v = R_{axis}^{-\alpha} \cdot R_v \quad (4)$$

$R_{axis}^{-\alpha}$ is a rotation matrix about axis which is one of the x, y, or z axes of WCS by-α angle.

Object rotation about WCS's axes: When the object is rotating about WCS's axes, ORF does the same. So each rotation about any axis of WCS by α angle is directly expressed, as:

$$R_o = R_{axis}^{\alpha} \cdot R_o \quad (5)$$

Based on the previous two sub-sections, it is easy to deduce that CVP rotates about its own axes in the inverse direction in which ORF does about WCS's. Dealing with the same reflection as previously, the rotation of CVP is equivalent to rotate WCS by R_v and then rotate CVP about the concerned axis. Indeed, this kind of rotation is exactly the opposite of that about ORF's axes. It is why R_v is rotated about $R_{v_{axis}}$ by -α which is one of its columns depending on the value of axis, which can be one of the three axes x, y, or z.

$$R_v = R_{v_{axis}}^{-\alpha} \cdot R_v \quad (6)$$

Object rotations about arbitrary axes: We should generalise the rotation so that it could be done around arbitrary axes. The previous rotations then become particular cases. Note that the principle of CVP rotating by -α when the object rotates by α remains always true.

Throughout the presentation of object's rotations about axes belonging to different systems, we can now establish with experimental basis that ORF rotates directly around arbitrary axes. Indeed, any axis that the object is requested to rotate about is referenced in WCS. Consequently, this is similar to the case of rotating about the ORF's axes when considering them as particular cases. The difference is that any other arbitrary axis lies along no one of ORF's. In matrix form we obtain:

$$R_o = R_{W_{AnyAxis}}^{\alpha} \cdot R_o \quad (7)$$

$R_{W_{AnyAxys}}^{\alpha}$ is a rotation matrix about AnyAxis referenced in WCS by α angle.

Within the same way of reflection, it is effortless to deduce that CVP rotates around the corresponding transformed axis by R_v. This is obvious since we know that at the same time when ORF rotates around one of either WCS's or ORF's axes, CVP rotates around its own ones or WCS's, respectively. This can be interpreted as if the axis is one of either WCS or ORF axes and then transformed by R_v. Consequently, if the axis is at an arbitrary position referenced in WCS, it should be transformed by R_v before applying the rotation. Mathematically, the operation consists in transforming the axis by the rotation R_v and then evaluating the requested rotation; namely the matrix R_v becomes:

$$R_v = R_{W_{TAnyAxis}}^{-\alpha} \cdot R_v \text{ where } TAnyAxis^T = R_v \cdot AnyAxis^T \quad (8)$$

Simulation achievement: The simulation of the object's movement is achieved through linking between grabbed images and current orientations of ORF and CVP. The user actions affect the axes' orientations being used to extract the image to be projected. Accordingly, looping on actions should simulate the rotation.

Linking between the orientation of these systems and stored images is made of two parts. The first one is to locate the real image related to the current viewpoint. This can be provided by the orientation of the z axis of CVP, i.e., the third column of R_v. The second part is to estimate the amount of rotation to be applied to the image so it reflects effectively the user's wish. Usually, this is not a direct query. It could result from particular orientations and should be determined. This is merely the angle that separates the y axis of ORF from the yz plane of WCS.

Seeking the expected image goes through n and k parameters estimation. Seeing that the z axis of CVP is oriented towards the virtual camera direction, its simultaneous relative orientation with yz and xz WCS's planes is used to get, respectively the values of n and k. More precisely, spherical coordinates of the third column ($R_v(:,3)$) of the R_v matrix are evaluated so as to estimate γ and β. Namely, assuming that $R_v(:,3)$ makes ϕ radians with yz plane and θ radians with xz plane, the values of n and k are, respectively:

$$n = \mathrm{mod}\left(\mathrm{round}\left(\frac{\phi}{st}\right), \frac{2\pi}{st}\right) \text{ and } k = \mathrm{round}\left(\frac{\theta}{st}\right) \quad (9)$$

where ϕ and θ should agree with some constrains. Especially, estimated values of θ are modified to π/2-θ and those of ϕ, if they are negative, to 2π + ϕ.

The rotation of the image in the z direction is decided upon the leaning study of the *y* axis of CVP in reference to yz plane of WCS. In fact, the z axis of WCS is oriented towards the use's viewing direction. So, the object should appear oriented if its vertical axis leans from the vertical plane yz of WCS. Consequently, the second column ($R_v(:,2)$) of the R_v matrix is used to estimate the inclination.

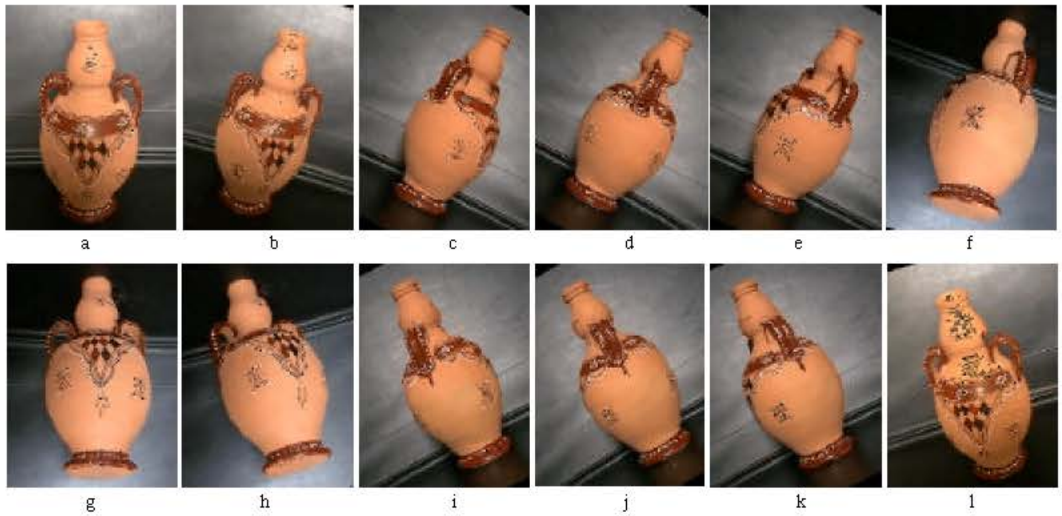

Fig. 3: A complete rotation of an object (from a to l) around the y-axis of WCS with a step st = p/6, after having been rotated about x axis of ORF by st (image a)

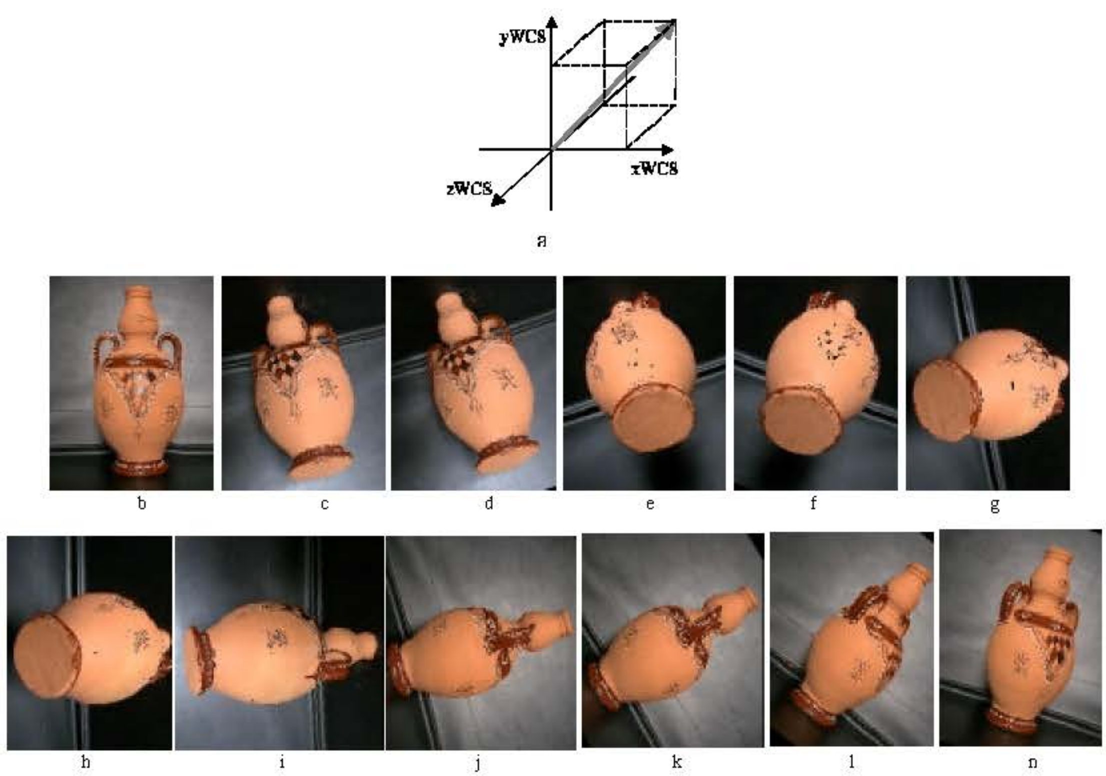

Fig. 4: A complete rotation (from b to m) about the simulated axis presented in a using a step st = p /6

Figure 3 and 4 depict an object rotating around different axes. By using the images in these figures in their original form, we intend to show the inclination induced by a particular rotation deduced from that of ORF.

DISCUSSION

This study provides two main results. The proposed mathematical solution may be considered as new data added to the previous results dealing with Image-space-based rendering systems. The linear storage structure is another result which easily supports the domain's requirements than the previous ones.

Especially, most of studies predefine paths the user is constrained to walkthrough during the simulation of the object manipulation which are generally the ones of the grab operation. Moreover, the only manner images are projected is in the way they have been grabbed. They can be just rotated in response to explicit user's requests, but this is not for the purpose to simulate a natural movement. The images are rotated without constraints applied to. This work does not only give the possibility to satisfy the user's requests, but assesses a basis for natural rotations independent from the way of the first grab and the number of images.

Furthermore, the proposed linear structure provides an advantage over the others. It gives an access to the frames in both linear and matrix forms because the images are organized in multi layers map structure. Through the n, k and r parameters of expression (2), each layer r can be manipulated as a set of images in relation to each other. This is significant for extending this work using interpolation principles for the purpose of generating smooth motion. The surrounding images of a given one are simply accessed through n and k. For this reason, this structure may be suitable for the other works. Consequently, despite the use of an AVI file, the structure is independent from any particular file type. It allows the storage in the same movie of multiple resolution image zooming. Hence, most compression algorithms can be applied whereas only specific directions may be so for other works. This is the case, for instance, for inter-frame within each viewing direction (Chen, 1995) or the 4D array (Gortler *et al.*, 1996).

Despite the previous advantages, there are some features that affect negatively the quality of obtained results. Among them, the vibration of the object can be seen during its rotation. Generally, this problem is due to the inaccuracy of the used mechanical apparatus during the grab operation. Typically, this leads to an object's center not at the same position in all frames. Another feature is the noise in data. So, in order to compensate it and due to the accumulated errors in matrix multiplications, we applied the method presented by Zhang (2000) to always estimate the best rotation matrices.

Finally, we outline the manner the object is manipulated. The emulated virtual sphere controller is used since it seems natural and lets the user feels like s/he is actually rotating the object (Chen *et al.*, 1988). On the display screen, the user can imagine viewing an object encased in a glass sphere. Rotation is then a matter of rolling the sphere and therefore, the object with the mouse cursor. Also, using non usual or expensive means goes against our goal which is the use of the most common tools. Moreover, as the principle of the virtual sphere is based on the computation of the axis of rotation from the corresponding used 2-D control device, the presented generalization of the rotation of the object around any arbitrary axis helps greatly in this realization. The computed axis is the one around which the object should rotate. For instance, Fig. 4 is a special case where the object rotates around an arbitrary axis that lies along none of our previously defined systems' axes.

CONCLUSION

In this study, we have described a general solution that links a set of images of an object stored in a movie file to specific viewing directions. This allows a user to simulate the object rotations with respect to the three degrees of freedom via its images. We have shown the way in which they could be organized in a linear structure. This organization presents many advantages than others because it allows at least an easy access. Moreover, it has been extended to be capable of supporting a multiple level resolution. We have also given the geometrical interpretation of an object's rotation around arbitrary axes that allows linking images to current viewpoints of a virtual camera. We have defined two different reference frame systems; each one following a specific way of rotation. The related analytical expressions have been established based on geometrical reflections and supported by experimental results demonstrating their efficiency.

One extension of this research, which is not necessary, is the possibility to expand it using image interpolation results. The purpose in this case would be to add more smoothness to the realism of movements and to face the problem of storage space. In fact, as the method does not entail any specific step during the grab operation, this can be adapted to reach any requirement of softness. Consequently, the amount of associated data would be too big particularly when high resolution images are used. It will be then important to develop powerful compression methods even with or without method expansion.

The most important extension to this work is to generalise it so that it covers the six degrees of freedom. It might be more attractive if a user is able to control the object in a whole 3D scene as if it was real. This goes through the manipulation of WCS in reference to other systems.

REFERENCES

Bakstein, H. and T. Pajdla, 2006. Omnidirectional image-based rendering. In: Computer vision winter workshop, February 6-8 2006.

Braun, N., 2003. Storytelling in collaborative augmented reality environments. WSGS' (2003), February 3-7 2003. Pizen. CZech Republic.

Chen, M., S.J. Mountfurd and A. Sellen, 1988. A study in interactive 3-D rotation using 2-D control devices. In: Proceedings of Computer Graphics SIGGRAPH 1988, pp: 121-130.

Chen, S.E., 1995. Quick time VR-an image-based approach to virtual environment navigation. In: Proceedings of Computer Graphics, Annual Conference Series, 1995, pp: 29-38.

Darsa, L., B. Costa and A. Varshney, 1997. Navigating Static Environments Using Image-space Simplification and Morphing. In: Cohen, M. and D. Zeltzer (Eds.), Symposium on Interactive 3D Graphics, April 1997. ACM SIGGRAPH, pp: 25-34.

Gortler, S.J., R. Grzeszczuk, R. Szeliski and M.F. Cohen, 1996. The lumigraph. In: Computer Graphics Proceedings SIGGRAPH'96, August 1996, New Orleans, pp: 43-54.

Levoy, M. and P. Hanrahan, 1996. Light field rendering. In: Computer Graphics Proceedings SIGGRAPH'96, August 1996. New Orleans, pp: 31-42.

Nini, B. and M. Batouche, 2005. Virtualized real object integration and manipulation in an augmented scene. In: Proceedings of the 11th International Conference on Computer Analysis of Images and Patterns, September 5-9 2005, Paris, France, LNCS 3691. Springer-Verlag, pp: 248-255.

Ono, S., K. Ogawara, M. Kagesawa and K. Ikeuchi *et al*., 2005. A photo-realistic driving simulation system for mixed-reality traffic experiment space. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV 2005), June 2005.

Popescu, V., J. Eyles, A. Lastra, J. Steinhurst, N. England and L. Nyland, 2000. The WarpEngine: An Architecture for the Post-Polygonal Age. In: Proceedings of the 27th Annual Conference on Computer Graphics, ACM SIGGRAPH, July 23-28, 2000, pp: 443-454.

Pulli, K., M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro and W. Stuetzle, 1997. View-based rendering: Visualizing real objects from scanned range and color data. In: Proceedings of the 8th Eurographics Workshop on Rendering, pp: 23-34.

Wang, L., H.-Y. Shum and S.B. Kang, 2002. Object representation and rendering using inverse concentric mosaics. In: Proceedings of The 5th Asian Conference on Computer Vision, ACCV, January 2002, Melbourne, Australia, pp: 23-25.

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell., 22: 1330-1334.

Zhong, X.W., P. Boulanger and N.D. Georganas, 2002. Collaborative augmented reality: A prototype for industrial training. In: Proceedings of the 21st Biennial Symposium on Communications, June 2002, Canada.