Eurographics Italian Chapter Conference (2006)
G. Gallo and S. Battiato and F. Stanco (Editors)

Multi-Modal Presentation of Work of Arts in Virtual Reality with Simulation of Multi-Mirror Reflection

S. Livatino [1], D. Cuciti [1], A. Wojciechowki [2]

[1] Medialogy Studies, Aalborg University Copenhagen, Denmark
[2] Institute of Computer Science, Technical University of Lodz, Poland

Abstract

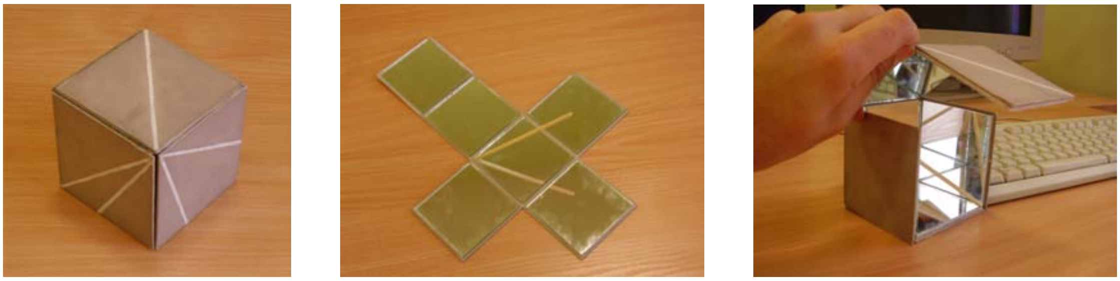
The use of different media such as photography and virtual reality combined with different presentation modalities may provide a user with an extraordinary tool for exploration and appreciation of real work of art. This is especially important in case of time-spatial works-of-art where the problem of functionalities presentation becomes much more demanding. The possibility offered by latest graphics machines has attracted the interest of researchers to investigate this new area. The goal of the presented work is the creation of a multi-modal presentation of a piece of contemporary art, tightening the relation between computer graphics and arts. The focus is on an application that it is capable to cope in real-time with simulation of mirror reflections, including multi-reflections. The results were very encouraging which led to optimistic conclusions with a wide range of options for future works.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction: Visual Arts and Virtual Reality

Contemporary art is very often misunderstood by ordinary people. Compositional tasks go beyond classical harmonization and proportions. Artists make use of time dependent changeability, interactivity and visual illusions in order to evoke certain effect among viewers. Installations constructed based on such elements are called time-spatial since with a passage of time their spatial characteristic is being changed. At the same time lack of space or preservation of the artistic object can make experimenting with pieces-of-art unavailable to people. A kind of remedy to the problem is making a sophisticated multi-modal presentation being a substitute or a complement of the real object and providing a user with wide variety of experiences. These media may comprise: photography, description, animation, film, interactive application, virtual reality, and combination of them. All of the media may co-exist within one complementary presentation. The aesthetical experience gained from the piece-of-art presentation is not only connected with external appearance but with internal functionalities conceptualization as well. In case of time-spatial works-of-art the problem of functionalities presentation becomes much more demanding.

Spatial features can be just imagined by multiplying the number of photos connected with different points of view. Unfortunately such mean of presentation is incomplete due to its discontinuity and lack of thorough object observation possibility. Another methods providing possibility of time-spatial works-of-art presentation are film and computer animation. They can more thoroughly retrieve time and spatial object's features but at the same time limits perspective and duration of the presentation. By means of interactive methods like virtual reality application, user may experiment personally with object's reproduction and discover its functionality. This most advanced manner of time-spatial works-of-art presentation characterizes with similar to real world time perception. None of the time-spatial works-of-art presentation methods are perfect. Presented solutions become either more photo-realistic or more interactive. Unfortunately fully interactive and visually photo-realistic solutions are not available yet due to lack of hardware and software advancement. That is why the main goal of the paper is to provide a user with an extraordinary tool for exploration and appreciation of the real work of art through the use of different media (e.g. photography, virtual reality) and presentation modalities (e.g. observation, interaction).



(a) (b) (c)

Figure 1: *Cube with Mirrors: (a) exterior appearance, (b) flat open, (c) while opening.*

The task of multi-modal presentation of real work-of-art in virtual reality is described basing on six-sided parallelogram, a cube, which contains six mirrors, such that each mirror fully covers the internal part of each cube side (fig. 1). The author of this installation is a second year student of the Graphics and Painting Faculty of the Academy of Fine Art in Lodz in Poland, Malgorzata Filarska. The author of the piece-of-art has constructed transformable installation, and provided a viewer with different operational and interpretational possibilities. Simple elements' joints assure wide spatial transformation possibilities. The usage of the mirrors for object's construction leads to a specific game between real elements of the installations (mirror, wooden stick fixed to the mirror surface), lines painted on the opposite side of mirrors and reflections of the linear objects (real and painted, wooden stick). Composition has both a simple construction and elegant and clever transformation possibilities.

There has been an interest in the recent time among researchers in the thematic related to time-spatial work of Arts [MP02], [WKP02]. Piotr Patyra [Pat05] investigates the Jozef Robakowski "Mirror's Ball" installation currently present at the Museum of Art in Lodz. The object consists of two elements: the spherical mirror and the ball. There could be observed the distort reflection of the little ball in the mirror as well as a reflection of the surrounding, and all this change in a special reflection game while rolling the ball. Earlier the model "Rolls" was created by Marcin Koman, [WKP02]. Even though mentioned authors discussed problem of works-of-art presentation quite thoroughly, none of them tackle the problem of multi-level reflections. None of already written papers solve the problem of interaction with installations comprising mirrors at high satisfactory level. This paper presents complementary attitude towards presentation of the piece-of-art in which reflections play one of the main roles, where reflections considerably influence installation perceiving.

2. The Mirror problem

Mirrors are brilliant and reflective surfaces, where the relation between light and reflection rays can be considered as defined by the laws of reflection by Rene Descartes: "the angle between the incident ray and the normal is equal to the angle between the normal and the reflected ray". Please note that mirrors behave differently depending on the surface (planar, curved convex, curved concave). In Computer Graphics it is often required to render a scene with reflecting surfaces or even mirrors. The Raytracing method qualifies as a perfect solution to the problem of reflection simulation, but it typically is not real-time, so it cannot be considered in applications like computer games. An alternative solution to reflection simulation is Texture Mapping (where the scene is first mirrored and then rendered on a texture). This technique is faster but in case the point of view changes or some object moves in the scene, it is necessary to re-render the texture.

Implementing mirrors in *Maya* is rather simple, (the Alias Maya is a tool for CAD, a very popular high-end solution in Computer Graphics). We use the *Mental Ray* ray-tracing engine because this supports the use of a special material for mirror simulation. It is important to limit the maximum number of reflections among the rendering options when there is more than one mirror in the same scene.

Implementing mirrors in OpenGL can be very suitable for real-time applications. In fact, other than by using Raytracing and Texture Mapping, in OpenGL a different solution may be implemented which employs the Stencil buffer and multi-pass rendering. The Stencil buffer is a component first developed by Silicon Graphics at the end of the '80s [AJ], and it was fully integrated in the mass-market 3D chips ten years after [Kil]. Like the z-buffer, it performs particular tests on a per-pixel basis to determine if the pixel can be drawn or not, thus acting exactly like a stencil. The Stencil buffer allows for tagging pixels in the framebuffer as belonging or not to the mirror surface. When the visible part of the mirror is drawn, a unique value is stored in the positions of the Stencil buffer corresponding to the pixels. That value tags those pixels as "belonging to the mirror". When the flipped scene is being drawn, the Stencil test will prevent the application to write in the pixels that are not marked with the mirror's tag value. A more complete version of the algorithm has to cope with depth buffer, clipping planes, illumination and so on. Please note that a smart use of the Stencil buffer also includes simulating shadows, highlights, etc., which explains why this was included since the earliest specifications of OpenGL. [Arc]

2.1. Multiple Mirrors Reflections

Sometimes in the same scene there is more than one mirror to render. Such cases can be very complicated to deal with. In fact, not only a mirror reflect the other mirrors, but it will also reflect the reflections of the other mirrors and they may reflect each other infinite times! The figure 2 shows an example. If time is not an important issue, Raytracing can be used. However, only a finite number of reflections should be allowed. In fact, no existing technique is able to render mirrors reflecting each others infinite times, and a reasonable limit should be set depending on the technique and hardware used.

In 1996 Diefenbach presented the Stencil buffer algorithm in a recursive fashion, making it feasible to solve the multiple mirrors problem [Die96]. This elegant solution makes a smart use of the Stencil buffer, using it as a counter to track the number of the nested reflections, so that any two reflections will not be "mixed up" in the same mirror. A drawback of this simple version of the algorithm is the excessive load of calculations, (exponential growth of the recursive calls). It is clear that optimization strategies must be adopted.

3. The Proposed Approach

The proposed multi-modal presentation of a real work of art is based on the use of different media (e.g. photography, virtual reality) and different presentation modalities (e.g. observation, interaction). The multi-modal presentation is investigated through the analysis of the type of installation introduced in section 1, the Mirror Cube. The work-of-art presentation modalities can be classified as in the following.

- **Real Observation**. Pictures and movies are the main form of visual documentation used since their introduction and they are something that everybody is familiar with. In this work only still photographs are taken into account and their usage will constitute the basis of the successive work. The Mirror Cube is photographed against a neutral background. High resolution photographs and close-ups are important. A large number of pictures should be taken, but only the most representative ones should be selected for the photographic presentation.
- **Virtual Animation**. The animation should consist of a sequence of views of the virtual work of art changing its configuration. The sequences should be merged in a video with duration of about 1 or 2 minutes. The user is not provided with any kind of interaction in this presentation, only with the basic commands for video reproduction (i.e. *play*, *stop*, *pause* etc.). The video should be graphically appealing, therefore high quality rendering techniques should be used. The Mirror Cube is modeled using the Maya software. This model will be the starting point for both the animation made in Maya and of the interactive application in OpenGL. In the video animation, every scene will be the result of a batch rendering of a number of frames in Maya. The camera should keep continuity in time. A software tool for video editing must be used to join the scenes all together and export them in a video format (e.g. the *.AVI* format).
- **Virtual Interaction**. The interactive application can render less accurate graphics, finding the best trade-offs between image quality and real-time performance on ordinary computers. The user will be given the instruments to change the configuration of the object at his/her will, without breaking the physical constraints of the object. The interactive application is implemented in OpenGL which is also able to simulate mirror reflections by the use of the Stencil buffer. The Mirror Cube modelled with Maya is imported to OpenGL and then interactively rendered. The programming language chosen for the interactive application is C++, since it is powerful, versatile and popular for OpenGL. In fact, even if Java and Java3D libraries support the use of the Stencil buffer (recently added in Java3D version 1.4) they still do not allow for multi-pass rendering (i.e. rendering multiple times with different parameters within a single frame).

The work-of-art presentation concept aims at different results. The video animation certainly goes for the quality of the image. The rendering process is performed only once per frame, therefore an approach based on Raytracing best suits as a solution for the problems of mirror simulation. A short movie will probably be just enough to understand the structure of the object and its functionality, however even if graphically appealing it will not be sufficient to satisfy the curiosity of a normal user. Therefore the need and so the advantage of an interactive presentation. In particular, the user will be able to get answers by manipulating the virtual object in the real-time.

The use of the Stencil buffer approach in OpenGL allows for high speed performance being the Stencil test as well as the depth test implemented in hardware. In fact, nobody seems to propose methods alternative to the depth test to establish object visibility for this type of application. The use of the Stencil buffer approach has anyway the consequence that the illumination is not correctly estimated [Kil]. In fact, the cube mirrors increase the illumination level in the scene as consequence of the mirror reflection. This leads to the fact that some parts of the object are illuminated even though they are not hit directly by light source. If instead than the OpenGL Stencil buffer approach, the scene illumination would be estimated by a Raytracing method, the above problem would not take place. The proposed approach of using recursive mirror reflections is not new. However, a method designed and implemented for rotating mirrors seems to be quite unique (the authors could not find examples of similar

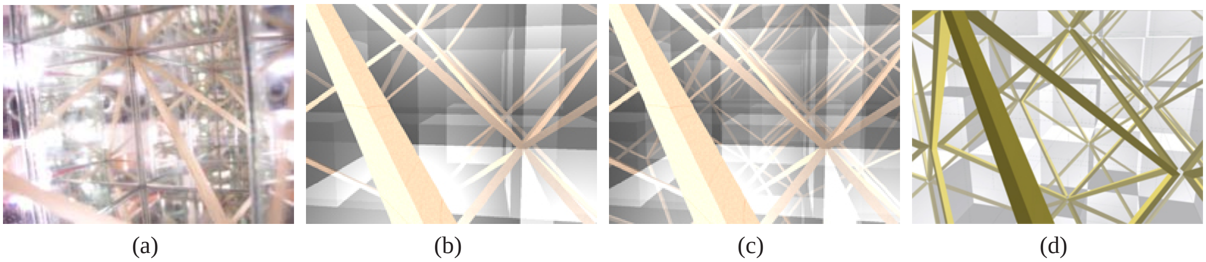


Figure 2: *Multiple reflections: (a) real photograph, (b) raytraced with a reflection limit set to 5, (c) raytraced with maximum 8 reflections, (d) rendered with the OpenGL application and 8 levels of reflections.*

applications). The reader should note that in case of rotating mirrors the appearance of the reflection plane needs to be continuously re-estimated as well as the visibility of a mirror from the other mirrors reflections.

4. Implementation

Real Observation: Photographic Presentation. Over 160 pictures were taken with cameras with different resolutions. The pictures were grouped into four sets according to the following concepts: *exterior appearance* (fig. 1), *Functionality* (fig. 1), *Artistic Nature* (fig. 3), *Infinite Reflections* (fig. 2).

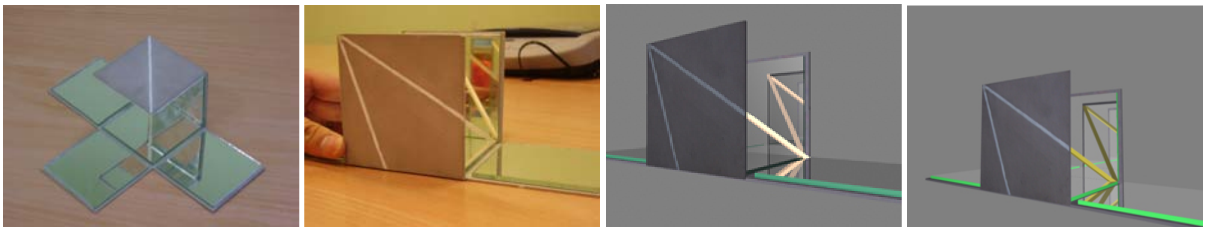
Virtual Animation: Model, Textures and Animation. The animated video sequences of the object are modelled Maya and rendered with *Mental Ray*. The work-of-art is modeled as a collection of boxes (including the mirrors) properly grouped in order to represent the real object transformations. The external surface is texture-mapped with photographs. The pictures are then elaborated in *Adobe Photoshop*. A bump effect is added to better simulate reality. Use of ad hoc material for mirror simulation. Important parameters to be set: number of maximum reflections, reflection color, diffuse color. Ray-traced views differ in either position of the camera or configuration of the object. All the frames in between two *keyframes* are linearly interpolated together with their properties, and then converted with *Adobe Premiere* to movie files. Final sequences presentation should based on a well defined storyboard.

Virtual Interaction: OpenGL and Stencil Buffer. The interactive application can be divided in four functional modules. One is active only during the initialization while the other modules are performed continuously to drive the interaction and the rendering. The “model loading” module manages loading the model previously drawn in Maya into the memory. This part of the program is crucial for performances because it generates the four display lists which simplify the code and improve rendering speed. The cube faces need to be assembled all together before visualization. There are two functions named: *doTransform* and *undoTransform* which perform translations and rotations according to object structure and user required actions. These functions are called during rendering in order to draw the whole model. The following pseudo-code describes the rendering process (the global position for the object in the space is set in transformation 0):

1. doTransform(0)
2. call the display list of the wooden stick
3. draw the face
4. for $n = 1$ to 5
    a. doTransform(n)
    b. draw the face
    c. if $n \neq 4$ undoTransform(n)
    d. if $n = 5$ undoTransform(4)
5. undoTransform(0)

The user should be able to interact with the object with a simple and intuitive use of common input peripherals as the mouse or the keyboard: change the viewpoint, (translation, rotation, zoom in-out), open each face independently from the others, etc. A function with the task of updating those angles is called at every frame, resulting in an animation which can be stopped at any given point.

Managing Multiple Mirrors. The recursive approach previously described is applied for the six mirrors. To reduce the overhead of the recursion we note that: (1) appearance of a mirror not facing the viewer should not be calculated; (2) flat mirrors cannot reflect their self directly. The rendering is implemented through three main functions appositely created for our application: *drawObjects*, *mirrorView*, *DrawGLScene*. The *drawObjects* calls the display lists as designed for this application with the only exception that it does not call the list of the mirror’s surface. The *mirrorView* creates the reflected view from the current mirror. When the current depth is equal to the established maximum depth, the function will only renders a gray polygon with full opacity, so that there will not be any more recursive calls. The *DrawGLScene* first calls the *drawObjects* routine, and then for each mirror in the scene applies the required transformation. If the mirror is visible and it is not the current mirror (which reflection it is being drawn), the call to the *mirrorView* is performed.



(a) (b) (c) (d)

Figure 3: *The artistic nature of the object: a mixture of reality and virtuality where real views merge with the mirrored reflections and lines painted on the exterior case. (a)&(b) real photographs, (c) raytraced with Mental Ray, (d) rendered with OpenGL.*

5. Testing and Discussion

Quality of the Image. The quality of the rendered images is observed by comparing real images with the computer-generated, (fig. 3). Surprisingly, the quality of the images rendered in OpenGL is not far from that of the images generated in Maya. This happens because the illumination in OpenGL is calculated once for each polygon vertex and it is then interpolated for the pixel belonging to the area enclosed among the polygon edges. On the contrary, in the ray-traced solution the illumination is calculated per each pixel. This difference appears to be very small to notice in the obtained result shown in fig. 3. Concerning the rendering of the reflections, these are correctly drawn for both rendering techniques, however a more accurate light calculation makes a greater difference. Interesting, in case of screen-shots taken inside the closed cube, the reflections are identical.

Speed Performance. The responsiveness of the application to input commands is tested for different levels of reflections. The test is done on a an Intel Celeron CPU 2.8GHz, 512MB RAM, 64MB of shared memory graphics card. The framerate during object manipulation is rather constant when the maximum reflection level is set to 3. Above it, the computation speed decreases when closing the faces and depending on viewer viewpoint. A level of 6 makes the system too slow to enjoy the visual effect.

Discussion. The results provided basic insights for future performance optimizations. All mirrors accurately reflect the scene and the other mirrors image. Aspects to improve are: increase the tessellation in the geometry for a more accurate shading, refine the case when the recursion limit is reached to avoid artifacts. More optimization is needed and we look with interest at a recent extension to OpenGL which will allow for discovering if a mirror is drawn in the framebuffer (so on the screen). The Stencil buffer which is now popular in commercial graphics cards, solves the reflection problem up to a relatively small number of nested reflections.

6. Conclusion

This paper described a multi-modal presentation of a time-spatial work of art, the Mirror Cube, allowing for appreciating such objects that are meant to reveal their nature through transformations of their components. The final product consists of a multi-media multi-modal presentation, embodying three forms of exhibitions: (1) photographic documentation; (2) a computer generated movie; (3) an interactive application in virtual reality. Each of these parts was generated trying to fulfill the gaps that the others would leave. A main focus was also the investigation of rendering techniques for scenes with multiple mirrors, (leading to potentially infinite numbers of reflections), so the design and the implementation of such a system. In particular, the approach using OpenGL and the Stencil buffer has been developed. Aspects for further investigation include: support for the mouse and data gloves, 3D stereo visualization, and computer games applications.

References

[AJ] AKELEY K., JERMOLUK T.: High-performance polygon rendering. In *SIGGRAPH '88 Proceedings*.

[Arc] ARCHITECTURE REVIEW BOARD (ARB):. `http://www.opengl.org/documentation/specs/version2.0/glspec20.pdf`.

[Die96] DIEFENBACH P.: *Pipeline Rendering: Interaction and Realism through Hardware Based Multi-Pass Rendering*. PhD thesis, CIS, Uni Pennsylvania, 1996.

[Kil] KILGARD M. J.: *Improving Shadows and Reflections via the Stencil Buffer*. Tech. rep., NVIDIA. `http://developer.nvidia.com/attach/6641`.

[MP02] MISKIEWICZ L., PIETRUSZKA M.: Presentations of time-spatial works-of-art in virtual reality. *The IASTED on Visualization and Image Processing, Malaga* (2002).

[Pat05] PATYRA P.: *Visualization of time-spatial works-of-art whose elements can be moved by a viewer*. Master's thesis, CS. Technical University of Lodz, Poland, 2005.

[WKP02] WOJCIECHOWSKI A., KRYSICKI P., PIETRUSZKA M.: Interaction with time-spatial works-of-arts and their behaviours in a virtual gallery. *ICCVG, Zakopane Poland* (2002).