

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

S. Livatino<sup>1</sup>, D. Cuciti<sup>1</sup>, A. Wojciechowki<sup>2</sup>

<sup>1</sup> Medialogy Studies, Aalborg University Copenhagen, Denmark

<sup>2</sup> 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

---

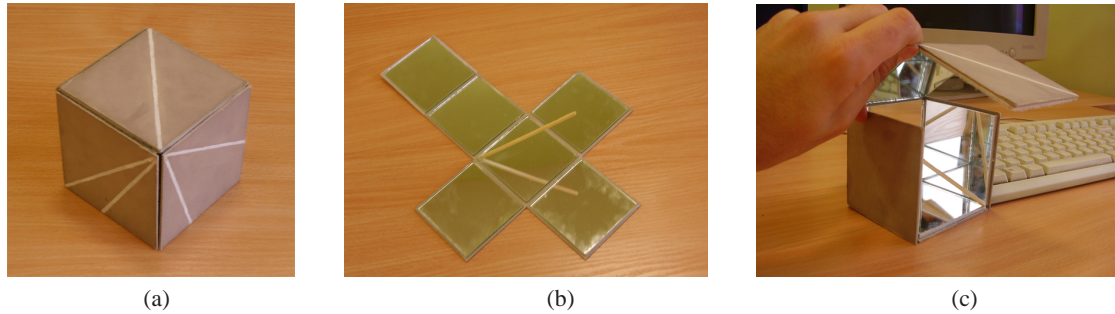
## 1. Introduction: Visual Arts and Virtual Reality

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 basing 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 sufficient exhibitions' conditions (e.g. lack of space for presenting all of pieces, installations which can not be touched in the museum) make experimenting with pieces-of-art unavailable for people. A kind of remedy to the problem is making a sophisticated multi-modal presentation being a substitute of a real object and providing user with wide variety of experiences. These media comprise: photographs, description, animation, film, multimedia, interactive application and virtual reality. All of the media may co-exist within one complementary presentation. The aesthetical experience which makes sense of 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.

Non-interactive techniques let user follow predefined mean of presentation. The most characteristic compositional features of certain pieces of art might be revealed automatically so that the viewer is just a witness of subsequent changes in a piece-of-art appearance and functionality, e.g. pictures made of light illuminations, computer animations. For example photographs which are flat reproductions of the installation reduce its third dimension. 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. The user can not much influence presentation chronology. He can just stop it, rewind or play slower. Computer animation provides much more possibilities but in comparison with a film has still not acceptable image sterility.



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

By means of interactive methods like virtual reality application, user may experiment personally with object's reproduction and discover its functionality. The piece of art might be revealed due to viewer activity, e.g. the observer modifies properly designed components of the installations and gradually discovers their features, i.e.: installations reminding books, installations which components can be modified in a different way where elements are joined by means of hinges, articulated and flexible joints or rollers, etc. This most advanced manner of time-spatial works-of-art presentation characterizes with similar to real world time perception. Especially, it concerns situations when events are triggered by user and their's duration depends on him. Virtual world presentation can be treated as object's simulation rather than object's reproduction. Simulation loop controls user's reactions, their influence on object's behavior and point of view orientation. Finally the most recent image corresponding to an actual point of view position and orientation is rendered.

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).

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 motivation for the object making was to compose a piece of art having theoretically unconstrained time-spatial construction. The author aim was to construct time-spatial piece-of-art providing interesting, from a plastic point of view, subsequent (in time)

spatial events (by means of elements' picking, method of joining them to each other, determining changes in elements' appearance and their origin). 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.

## 2. Related Work

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. In the mirror there is reflected a surrounding as well. The ball is covered by the photos. The viewer can move the ball. After changing the position of the ball, the reflection in the mirror change. The ball can rolls at the mirror's surface on the specific elliptic path. Earlier the model "Rolls" was created by Marcin Koman (master works by Andrzej Jakubowski [Jak02] and Adam Ulacha [Ula02] [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.

### 3. The Mirror problem

Mirrors are brilliant and reflective surfaces, meaning that when they are hit by the light this is not absorbed (for the most) but, it bounces in a different direction. The new direction of every ray of light mainly depends on three factors: angle of incidence, point of incidence (*poi*) and vector normal to the surface in the *poi*. In particular their relationship is 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, (the Raytracing method simulates the behavior of rays of light when they bounce among objects in a scene), but Raytracing 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.

#### 3.1. Mirrors in Maya

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. In order to render mirror reflections we use an algorithm which main steps are: (1) select the surface that is going to be the reflective surface of the mirror; (2) assign to this a material and a color; (3) adapt reflected color and perform some small tuning to improve appealing of the reflection. 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.

#### 3.2. Mirrors in OpenGL: The Stencil Buffer Approach

Other than using Raytracing and Texture Mapping, a different solution 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 [AJ88], and it was fully integrated in the mass-market 3D chips ten years after [Kil99]. 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. A simple algorithm would include: (1) draw all the objects in the scene, mirrors excluded; (2) enable stencil test and disable the colors; (3) draw the surface of the mirror in a way that updates the Stencil buffer; (4) flip the scene through the mirror's plane and enable the colors;

(5) draw again all the objects, mirrors included, only where the stencil buffer has been updated (i.e. inside the mirror); (6) disable stencil test.

When the visible part of the mirror is drawn in (3), 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 in step (5) 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. The solution above is a simplified version of the real one, that has to cope with depth buffer, clipping planes, illumination and so on. Nevertheless, it is sufficient to understand the basics of this approach. 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]

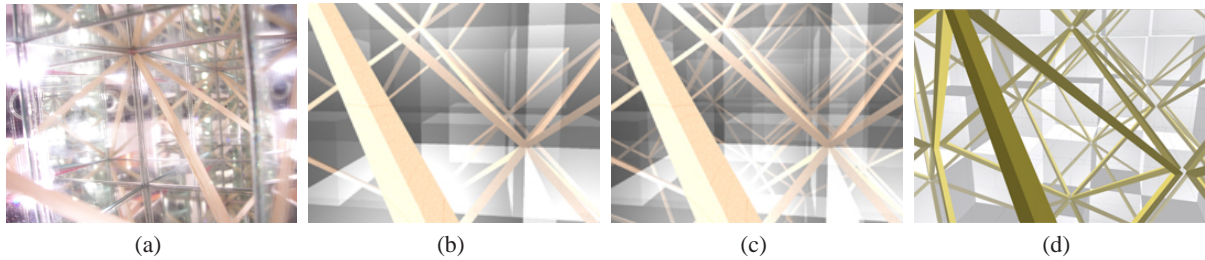
#### 3.3. Multiple Mirrors Reflections

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. Let us consider a simple scenario where two mirrors are placed one in front of the other, 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 but in a recursive fashion, making it feasible to solve the multiple mirrors problem [Die96]. It follows a description of the Diefenbach's solution through a pseudo-code labeled as the *Draw\_Scene\_Recursively* function.

1. enable stencil test: passes only if the reference value equals the current\_depth;
2. draw all the objects in the scene, mirrors excluded;
3. disable the colors;
4. for each mirror in the scene:
  - a. draw the surface of the mirror: update the stencil buffer by incrementing the value;
  - b. flip the scene through the mirror's plane and enable the colors;
  - c. if  $\text{current\_depth} \neq \text{maximum\_depth}$  call *Draw\_Scene\_Recursively*(current\_depth+1);
  - d. disable colors and draw again the surface of the mirror: update the stencil buffer decrementing the reference value;
5. enable colors and return.

This elegant solution makes a smart use of the Stencil



**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.

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, given by an exponential growth of the recursive calls. In a scene with two mirrors, if  $n$  is the "maximum\_depth", the function above will be called  $2^n$  times. It is clear that optimization strategies must be adopted.

#### 4. The Proposed Approach

The main goal of the presented work is to investigate the realization of a multi-modal presentation of a real work of art 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. Different aspects, as well as the concept itself of the work-of-art of interest, can be appreciated by means of: photographic documentation, animated virtual demos, and interactive demo applications. In particular, the work-of-art presentation modalities can be classified as in the following. Please note that the three presentation modalities can be performed independently of one another.

- **Real Observation:** observation of the real object through still photographs and movie sequences. Pictures and movies are the main form of visual documentation used since their introduction and they are something that everybody is familiar with. This is the reason here they are referred as "real observation" in spite of their two dimensional nature. In the scope of this project only still photographs are taken into account and their usage will constitute the basis of the successive work. Hence the more pictures are taken the more material will be later available.

The Mirror Cube is photographed against a neutral background. The camera should have high resolution and support the focus for close-up pictures, so that every picture is very detailed. A large number of pictures should be taken, but only few of them should be selected for the photographic presentation, i.e. the most representative ones.

- **Virtual Animation:** computer generated animation of virtual objects. 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. From scene to scene the object does not change configuration, only the position of the camera is different in order to 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:** computer generated interactive rendering of virtual objects. 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 such as tearing its components apart or consuming the object material. The interactive application is expected to import the geometry of the Mirror Cube which has been built using the Maya software and then interactively render this object. The interactive application is implemented in OpenGL which is also able to simulate mirror reflections by the use of the Stencil buffer.

The programming language chosen for the interactive application is C++, since it is powerful, versatile and uncountable tutorials are available on the internet about the combined use C++/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 al-

low for multi-pass rendering (i.e. rendering multiple times with different parameters within a single frame).

#### 4.1. Motivation

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 in this case. 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, who may wonder: “*what if I look at it from this point of view, in this particular configuration?*”. The above question is just an example of the limits of a presentation based only on animations or photographic observations. Therefore the need and so the advantage of an interactive presentation. In particular, the user will be able to get answers to questions like the above one by manipulating the virtual object in the real-time. The quality of the graphics is expected lower when compared with the video animation. Nevertheless, the reflections will be correctly rendered up to a predefined level of depth.

The use of the Stencil buffer approach in OpenGL allows for high speed performance being that the Stencil test as well as the depth test are 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 [Kil99]. 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 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.

### 5. Implementation

#### 5.1. Real Observation: Photographic Presentation

Over 160 pictures were taken in three photographic sessions. Two digital cameras with different resolutions were used. Furthermore, for practical reasons and for a more suitable

presentation, the pictures were grouped into four sets according to the following concepts: exterior appearance, functionality, artistic nature, infinite reflections effects. In particular:

- *Exterior Appearance*. The Cube is shown from outside, closed. The work of art is the only visible object. The view is prospective, so that it is possible to see different faces at the same time and appreciate the composition made by the lines. Samples are shown in fig. 1.

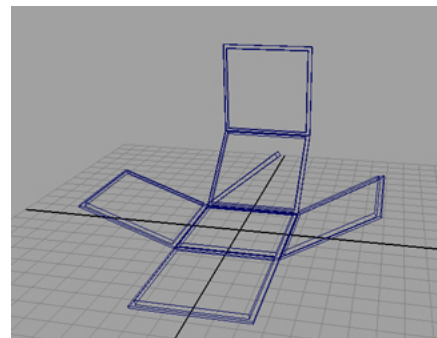
- *Functionality*. It describes the possibilities of opening the Cube. The faces are shown half opened and no relevance is given to other features of the object, like the reflections or the wooden stick. Samples are given in fig. 1.

- *Artistic Nature*. It gives the idea of what this object is all about: a mixture of reality and virtuality, with the real object views (e.g. the wooden stick) that merge with the reflections on the mirrors and also with the lines painted on the exterior case. Samples are given in fig. 3.

- *Infinite Reflections Effects*. A collection of pictures interesting merely from a photographic point of view. They provide an interesting view of the optical effects achievable with six mirrors. The greatest challenge is to reproduce these effects in the virtual representations of the work of art. Examples of this challenge are depicted in fig. 2.

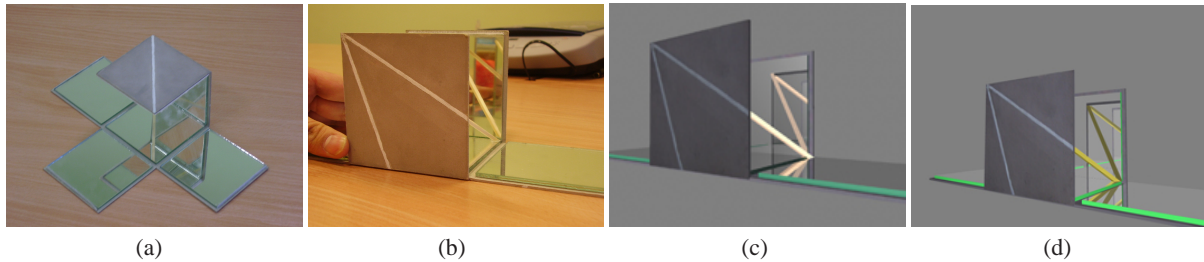
#### 5.2. Virtual Animation: Model, Textures and Animation

The Maya software was used to create the model of the Mirror Cube, and to produce an animated video sequence of the object, 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, so that the same transformation would affect objects which are actually connected together. A hierarchy is set connecting rotating actions.



**Figure 4:** The model of the Cube with Mirrors just created in Maya.

The external surface is textured with six pictures of the real object to correctly reproduce each orthogonal face. The pictures were then elaborated in *Adobe Photoshop*, (a popular software for image processing). A bump effect is added



**Figure 3:** The artistic nature of the object: (a)&(b) real photographs, (c) raytraced with Mental Ray, (d) rendered with the OpenGL application.

to all these textures, to give the impression of a rough surface as it is in reality. The mirror texture is simulated using ad hoc material in Maya and only some parameters need to be set. These are: number of maximum reflections, reflection color, diffuse color and few others. The Raytracing routines will take care of all the calculations. The wooden stick is textured with predefined material.

Once all the materials have been applied, the model is ready to be rendered. Every rendered image should differ from the previous ones in either position of the camera or configuration of the object or both. In Maya this is easily done with the use of keyframing. A finite number of frames is selected as a “key” frame for that object. All the frames in between two keyframes are linearly interpolated together with their properties. The same keyframe technique is applied to many objects at the same time and to the camera as well.

All frames are then rendered in a batch job, saved as images, and imported in a program like *Adobe Premiere*, (a video editing tool allowing for exporting the sequence in a movie file). Samples from this sequence are shown in fig. 3. In order to make a complete video presentation many sequences occur, with a well defined storyboard.

### 5.3. Virtual Interaction: OpenGL and Stencil Buffer

The interactive application is written in C++ programming language using OpenGL libraries. It 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 display lists. At the end of the model loading routine, four display lists are ready to be used, simplifying the rendering code and improving its 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:

1. *doTransform*(0)
2. call the display list of the wooden stick
3. draw the face
4. for  $n = 1$  to 3
  - a. *doTransform*( $n$ )
  - b. draw the face
  - c. *undoTransform*( $n$ )
5. *doTransform*(4)
6. draw the face
7. *doTransform*(5)
8. draw the face
9. *undoTransform*(5)
10. *undoTransform*(4)
11. *undoTransform*(0)

Notice that the starting point for the transformation 5 is transformation number 4. While in transformation 0 the global position for the object in the space is set.

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. A requirement is also the possibility to change the point of view, (rotating the object and performing zoom in-out). The user should be able to open each face independently from the others in respect of the physical constraint of the joints. A function with the task of updating those angles is called at every frame, resulting in an animation. The function increases/decreases the angle according to the status of the face. The user is allowed for stopping the animation at any given point. Each face along with the rotational angle has then got a status variable which can be either “opening”, “closing” or “stop”.

### 5.4. Managing Multiple Mirrors

The recursive approach previously described is believed to be straightforward for rendering the six mirrors that are present in the scene. But first, a few considerations to reduce the overhead of the recursion have to be made: (1) a mirror

might not be facing the viewer, thus its reflective surface is not visible; (2) flat mirrors cannot reflect itself directly.

The rendering is implemented through three main functions appositely created for our application:

- *drawObjects()*
- *mirror\_view(int depth, int current\_mirror)*
- *DrawGLScene(int depth, int current\_mirror)*

The *drawObjects* function 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 *mirror\_view* function creates the reflected view from the current mirror by performing the following operations:

- a. increasing the Stencil buffer where the surface is visible;
- b. reset z-buffer in visible areas;
- c. fill visible areas with background color;
- d. create and set the clipping plane;
- e. set the transformation matrix for the mirrored scene;
- f. call the "DrawGLScene" function;
- g. resume the transformation matrix;
- h. disable the clipping plane;
- i. draw mirror's polygon with gray light (blending enabled);
- l. increase Stencil buffer values where the surface is visible.

When the current depth is equal to the established maximum depth, the function will skip those steps and will only renders a gray polygon with full opacity, so that there will not be any more recursive calls.

The *DrawGLScene* is the main rendering function. It first calls the *drawObjects* routine. Then for each mirror in the scene, it applies the required transformation and, if the mirror is visible and it is not the current mirror (which reflection it is being drawn), the call to the *mirror\_view* is performed.

## 6. Testing and Discussion

**Test Design.** A testing phase is performed for the computer animation and the interactive application. The focus is on: (1) the quality of the rendered images; (2) the speed performance of the interactive application. The quality of the rendered images is observed by comparing the real pictures with screen-shots taken from the two means of presentation. The speed performance (responsiveness of the application to input commands) is tested for different levels of reflections. The level of photo-realism in the generated images is expected higher for the virtual animation, but it is important to find out what exactly makes the Maya application better than the OpenGL one. The speed performance is expected to exponentially decrease with the increasing of the depth of recursion. Though, it is interesting to investigate the circumstances in which the simple mechanism of pruning the recursion improves the responsiveness. The results are expected to provide basic insights for future optimizations.

**Test Results: Quality of the Image.** The figures 3.c and 3.d are screen-shots taken from rendering sessions in

Maya and in OpenGL. Surprisingly, the quality of the images rendered in OpenGL is not far from that of the images generated in Maya. In particular, if we only consider the external views of the object, it is the illumination of the surface a major contributor. 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. Nevertheless, when it comes to the internal views, the wooden stick appears better textured when using Maya. 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. Nevertheless, the reflections generated by Maya may produce "strange" transparencies, whereas the ones generated by OpenGL using the Stencil buffer may appear with a "strange" background with gray cubes.

**Test Results: Speed Performance.** The "speed" performance test is done on a laptop with an Intel Celeron CPU of 2.8GHz, 512MB of RAM and an integrated Intel video card of 64MB of shared memory. Starting from a maximum depth of 1 reflection, the program is run. The framerate is rather constant during the manipulation of the object. Same observations are made for the maximum depth of 2 and 3 reflections, but in the latter the movements are less smooth when some faces are closed and visible. When the number of reflections is set to 4 the problems come. In particular, in cases when the cube is completely opened, the computation speed is still acceptable, however, while closing one by one the faces the system performs slower and slower. If the cube is rotated in a way that the closed faces do not point toward the viewer, then an increase of framerate is noted. With all faces closed, even if no mirror is visible, the calculations are still made for the mirrors which surfaces are facing the viewer. A level of depth set to 6 makes the system too slow to enjoy the graphic effect. Still, when the faces are almost all opened the framerate is like it was with only 1 reflection.

**Discussion.** The results of the "quality" test were encouraging. All mirrors accurately reflect the scene and the other mirrors image. Aspects to improve are:

- increase the tessellation in the geometry, so that more accurate shading calculation will be performed, resulting in a surfaced lightened more realistically in the interactive application;
- improve the texture of the wooden stick within the interactive application;
- fix the transparencies problem spotted in fig. 2;
- find an alternative solution for the algorithm when it reaches the limit of the recursion, to avoid the gray cubes background artifact.

The result of the “speed” performance test showed that there is still a long way to go before the algorithm can be said properly optimized. In fact, most of the recursive calls are called even though they do not yield any contribution to the frame (e.g. in case the view point is placed outside the cube). A small success in the meantime is achieved by pruning the tree of the recursive calls. There still is a large room for improvement and optimization. The most relevant help in this direction comes from an extension to OpenGL, recently added among the official extensions approved by ARB, that in this case would allow for discovering whether a mirror is drawn in the framebuffer (i.e. it will be displayed on the screen) or not. With this extra information it is possible to calculate only the reflections that are contributing to the image, reducing the computational complexity of the algorithm.

## 7. Conclusion

In a clear perspective of considering Virtual Reality a valid instrument that can provide aid to Visual Arts, this paper successfully produced a multi-modal presentation of a time-spatial work of art, the Mirror Cube. The contribution of virtual reality consists of making this piece of contemporary art remotely available, overcoming the need of physical presence to appreciate 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, e.g. the interactive application alone would not give the graphic details that a high quality animation and the real pictures can provide.

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. Among the various approaches analyzed, the one using OpenGL libraries and the Stencil buffer were preferred. In fact, the Stencil buffer is very common in the commercial video cards, and it solves the problem up to a relatively small number of nested reflections. When “too many” nested reflections are involved a slow response to user commands may be experienced, however, there is large room for optimizations.

The presented work provides many aspects open to further development. For example, adding the support for the mouse and data gloves could be an interesting challenge. Another aspect worth to investigate would be to develop one or more solutions for 3D stereo visualization, aiming at increasing the “sense of presence” during the presentation. A further very promising aspect to investigate which is related to the interactive application and the implementation of a recursive algorithm, it is the re-use of such an approach in computer

games applications. In case of a first person shooter game it is the main character who performs actions, it will consequently be of great interest and very engaging, the case when the character would enter a room full of mirrors, where the reflection would respond accordingly to character movements. Let us imagine for example the case of two or more mirrors sitting one in front of each other (and even in different but communicating rooms). This type of graphical effects, despite very challenging to implement, is expected to encounter great success among game users.

## References

- [AJ88] AKELEY K., JERMOLUK T.: High-performance polygon rendering. In *SIGGRAPH '88 Proceedings* (Aug. 1988), pp. 239–246.
- [Arc] ARCHITECTURE REVIEW BOARD (ARB): *OpenGL 2.0 Specification*. available at <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, Computer Information Science, University of Pennsylvania, 1996.
- [Jak02] JAKUBOWSKI A.: *Dokumentacja cyfrowa wybranych dzieł sztuki z kolekcji Muzeum Sztuki w Łodzi*. Master’s thesis, Institute of Computer Science Technical University of Lodz Poland, 2002.
- [Kil99] KILGARD M. J.: *Improving Shadows and Reflections via the Stencil Buffer*. Tech. rep., NVIDIA Corporation, Nov. 1999. available at <http://developer.nvidia.com/attach/6641>.
- [MP02] MISKIEWICZ L., PIETRUSZKA M.: Presentations of time-spatial works-of-art in virtual reality. *The IASTED International Conference on Visualization and Image Processing Malaga Spain* (2002).
- [Pat05] PATYRA P.: *Visualization of time-spatial works-of-art whose elements can be moved by a viewer*. Master’s thesis, Institute of Computer Science Technical University of Lodz Poland, 2005.
- [Ula02] ULACHA A.: *Interakcja: odbiorca - dzieło sztuki w rodowisku wirtualnym z zanurzeniem*. Master’s thesis, Institute of Computer Science Technical University of Lodz Poland, 2002.
- [WKP02] WOJCIECHOWSKI A., KRYSICKI P., PIETRUSZKA M.: Interaction with time-spatial works-of-arts and their behaviours in a virtual gallery. *International Conference on Computer Vision and Graphics (ICCVG) Zakopane Poland* (Sept. 2002).