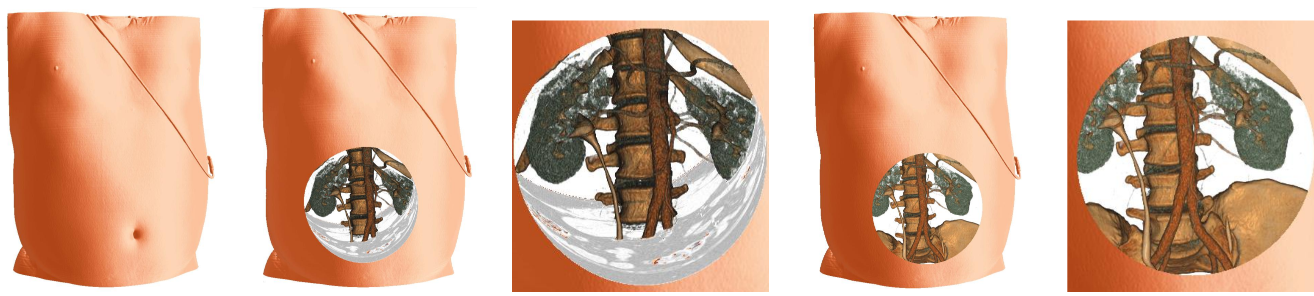
# The Virtual Magic Lantern: An Interaction Metaphor for Enhanced Medical Data Inspection

Eva Monclús, José Díaz, Isabel Navazo and Pere-Pau Vázquez*
MOVING Group, Universitat Politècnica de Catalunya



(a) Original model (b) Virtual Magic Lantern (VML) (c) Zoom-in of the VML region of interest (d) Virtual Magic Window (VMW) (e) Zoom-in of the VMW region of interest

**Figure 1:** *Advanced inspection of a medical dataset using the* Virtual Magic Lantern *(*b *and* c*) and the* Virtual Magic Window *metaphors (*d *and* e*). Note how the first metaphor produces higher amount of contextual information because it displays the boundary of the two differently visualized regions.*

## Abstract

In Volume Rendering, it is difficult to simultaneously visualize interior and exterior structures. Several approaches have been developed to solve this problem, such as cut-away or exploded views. Nevertheless, in most cases, those algorithms usually require either a preprocess of the data, or an accurate determination of the region of interest, previous to data inspection.

In this paper we present the *Virtual Magic Lantern* (VML), an interaction tool tailored to facilitate volumetric data inspection. It behaves like a lantern whose virtual illumination cone provides the focal region which is visualized using a secondary transfer function or different rendering style. This may be used for simple visual inspection, surgery planning, or injure diagnosis. The VML is a particularly friendly and intuitive interaction tool suitable for an immersive Virtual Reality setup with a large screen, where the user moves a Wanda device, like a lantern pointing to the model. We show that this inspection metaphor can be efficiently and easily adapted to a GPU ray casting volume visualization algorithm. We also present the *Virtual Magic Window* (VMW) metaphor as an efficient collateral implementation of the VML, that can be seen as a restricted case where the lantern illuminates following the viewing direction, through a virtual window created as the intersection of the virtual lantern (guided by the Wanda device) and the bounding box of the volume.

**CR Categories:** I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques; J.3 [Life and Medical Sciences]: Medical Information Systems

**Keywords:** Virtual Reality, Medical Models, Interaction

## 1 Introduction

The recent advances in Medical Imaging, Direct Volume Rendering, graphics hardware, and Virtual Reality technologies at affordable prices have empowered the development of Virtual Reality applications tailored to the real time medical data manipulation. Therefore, applications such as interactive inspection, surgery planning and surgeon training have become a reality [Robb 2008]. Clinical use of such applications requires reduced data preprocess times, efficient rendering algorithms, and friendly and intuitive user interfaces.

Direct Volume Rendering (DVR) allows spacial interpretation of medical images but this causes that values from different images are rendered onto a single pixel. This poses problems for the simultaneous visualization of the whole information captured by the volumetric dataset. Recently, different techniques and strategies have been proposed with the objective of facilitating the identification and exploration of features or regions of interest (for space reasons, we only mention some papers): *Cut-away views*([McInerney and Broughton 2006]), *focus+context visualization* ([Bruckner and Gröller 2006]), *Lens and Distortion* ( [Wang et al. 2005]) and *advanced Transfer Functions (TF)* ([Kniss et al. 2001] and [Bruckner and Gröller 2007]). Most of them have been proposed for 2D displays (although some of them could be adapted to stereo), and their focus is to provide the specialist with a view of the feature of interest without occlusions of other neighbor structures.

Complementary to these advanced visualization techniques, some medical applications can benefit from the anatomy's knowledge the specialist has, in order to facilitate the localization and identifica-

*{emonclus,jdiriberri,isabel,pere.pau}@lsi.upc.edu. This work has been partially funded by the Spanish Ministry of Science and Technology under grant TIN2007-67982-C02-01. J. Díaz also is supported by MCI-FPI Grant of the same Ministry.

tion of the region of interest. This requires the use of powerful and intuitive interaction metaphors. In immersive virtual reality environments, human interaction becomes of crucial importance. 3D pointers are commonly used in order to perform tasks of selection and manipulation.

In this paper we present a new interaction metaphor: The *Virtual Magic Lantern* (VML). Its name is inspired by the Magic Lantern [The Magic Lantern Society 2007], the precursor of the projector. It behaves like a lantern whose illumination cone determines the region to be analyzed (see Figure 5.a). In Figure 2 we provide an example of an interactive session in a VR environment where the user is using the VML to inspect a medical dataset. The *Virtual Magic Lantern* is guided by a pointer device that provides 3D position and orientation. The use of such a metaphor has several advantages, the most important one is that most of us have used a lantern many times in order to inspect a low light environments, and therefore its usage is completely familiar to us. Thus, VML is a simple tool that helps to determine a region of interest by providing a simultaneous visualization of different rendering styles for the interior lighted region and its context (see Figure 1.c). The VML can be used either in a Virtual Reality environment or a desktop-based application, although it is especially friendly in a virtual reality setup with large screens, because the interaction becomes very natural and significantly improves the user experience.

The clinical application of an interaction and exploration tool poses some requirements: First, it must maintain a real time framerate. Second, its integration with the rendering algorithms (DVR, Maximum Intensity Projection, expressive visualization . . . ) should be seamless. Our algorithm can be adapted without effort to a volume ray casting algorithm, while maintaining realtime framerates.

Furthermore, we also propose a second interaction metaphor, named *Virtual Magic Window* (VMW) that can be seen as a particular case of the VML. It allows locating a virtual window with the 3D pointer (see Figure 5.b). The region of interest becomes the part of the volume that can be seen through it. The VMW does not provide as much contextual information as the VML, especially on the boundary of the region of interest (see Figure 1.d). Actually, VMW is, in some sense, similar to other established Magic Lens approaches (see for instance [Bier et al. 1993] or [Wang et al. 2005]), although it provides higher flexibility in shading style, or the shape of the analyzed region (and we do not apply volume deformations).

## 2 The Virtual Magic Lantern

The *Virtual Magic Lantern* takes its name from the *Magic Lantern*, a device intended to project images onto a wall through the use of sunlight or candle light, and a convex lens as an objective to focus the images [The Magic Lantern Society 2007]. It is the precursor of modern projectors and its invention is not clear.

We build our *Virtual Magic Lantern* through the use of a Wanda device and a Head Tracker. The Wanda is used as the pointer device that casts a cone onto the model. As an analogy to the change of the *slide* (image) in the Magic Lantern, we change the transfer function or the rendering motif on the region subtended by the virtual cone growing from the pointing device (see Figure 2). The volume not intersected by this cone is rendered using the original transfer function. With our approach, we do not restrict the shading technique to be used: It may be a TF that hides unimportant information for simulating cutaway views, a TF that enhances the model with depth cues for improving perception, or a function that visualizes the intersected geometry using some volume illustration paradigm (see Figure 1 and Figure 6).

Next we introduce the architecture of our application and the Virtual

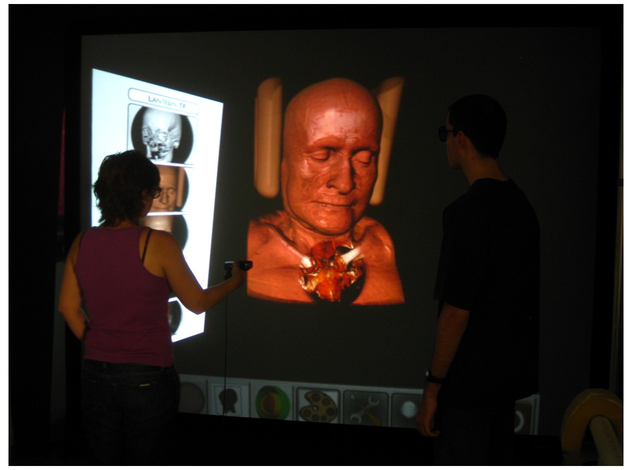


**Figure 2:** *Image showing the users interacting with our system.*

Reality setup, and later we give some details on the implementation and possible applications of our system.

### 2.1 Overview of the system

Our system, coined *VRMedVolVis*, is intended to work in an immersive virtual reality setup, although we also have a desktop version (*MedVolVis*). Concretely, we work on a 2.7 × 2 meters passive stereo powerwall, and we use an Intersense IS-900 Motion Tracking System device consisting on a Head Tracker and a MiniTrax Wanda with joystick Wanda device as a pointer. The Wanda device is used to track the position and orientation of the lantern, and its joystick is used to change the aperture angle, and one of its programmable buttons toggles between the two rendering modes: VML and VMW.

The application window has three main regions, as can be seen in Figure 3: The *rendering window*, placed in the middle, where the model is rendered. The *slide window*, placed on the left, that contains the set of transfer functions or illustration motifs already implemented in our approach. Finally, we have also a *toolbox* placed at the bottom, that contains the remainder of the interaction elements of our applications.

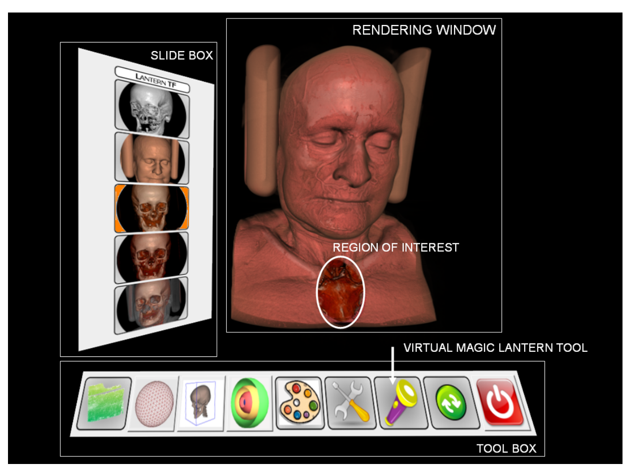
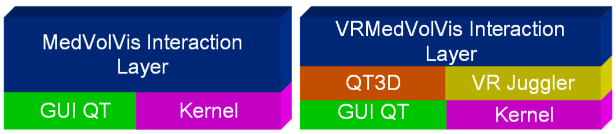



**Figure 3:** *Application layout.*

We have implemented two versions of our application, one intended to work in a desktop PC, and the second one, for an immersive Virtual Reality system. Figure 4 shows the software architecture and technologies used by our application. The whole system is designed in a layered fashion. Both versions (desktop and Virtual Reality) share two modules: The *Kernel* and the *GUI QT* modules. The Kernel module deals with all the operations platform independent such as the model load, all the visualization algorithms and shading techniques, etc., while the *GUI QT*, implements the user interface using Qt. The *Interaction Layer* lies on top of the architecture, and it is different for each application. For the Virtual Reality application, we use two extra modules: VR *Juggler* and Qt3D [Andujar et al. 2006]. The first one is a toolkit for the creation and execution of Virtual Reality applications, independent of the concrete virtual reality environment. The second is used to adapt all the Qt-based interface to a 3D GUI, mandatory for a Virtual Reality application.



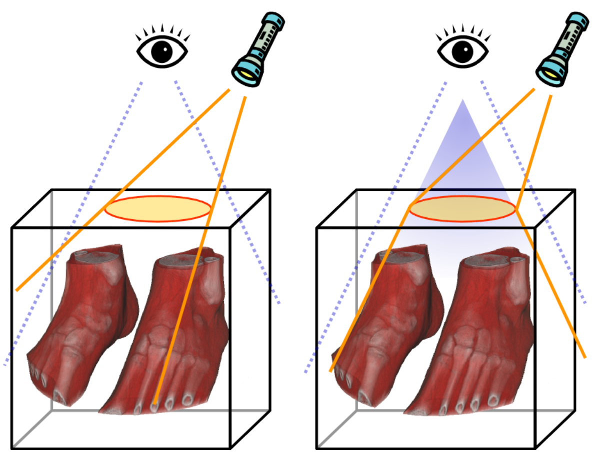

(a) Desktop (b) Virtual Reality

**Figure 4:** *MedVolVis layered software architecture.*

### 2.2 Implementation

In the implementation of both interaction metaphors (VML and VMW), the data is considered as divided in two subvolumes, each of which is rendered using a different transfer function, illustrative motif, or shading style.

As we previously mention, the difference between both approaches is the shape generated through our interaction with the 3D pointer device.These two cases are depicted in Figure 5.



(a) Virtual Magic Lantern (b) Virtual Magic Window

**Figure 5:** *The two cone-based inspection regions.*

In order to obtain a lantern-based inspection using a virtual cone as the region of enhanced inspection, we simply modify the GPU Raycasting [Hadwiger et al. 2006].

The changes we do are the following: We basically add to the fragment shader the code required to determine if a point is inside the cone. The test is executed at each sample point of a ray. The application passes to the GPU the transfer functions and the geometric information that defines the cone: apex, axis, and aperture angle. The resulting shader is simple. For illustration purposes, we sketch the pseudocode of the ray casting implementing this approach.

```
...
vec3 position = calculateOriginRay(gl_FragCoord);
initializations, ...
while (! endrayCasting(...)) do
    float v = texture3D( VolumeTexure, position ).a;
    vec4 material;
    if (isInsideCone(position, v)) then
        material = calculateInsideColor(v)
    else
        material = calculateOutsideColor(v)
    end if
    calculatePhongShading(material, position)
    composite colors
    update position
end while
...
```

This technique generates complex intersection volumes that give the user a lot of information on the surrounding context. Although this implies a rendering penalty, because of the inclusion test, we still maintain real time framerates.

As stated, the *Virtual Magic Window* technique builds an initial cone with origin and direction dictated by the Wanda device and intersects it with the bounding box of the volume. Then, all the rays cast from the observer that pass through this shape, are labeled as lying inside the region of interest. This computation is straightforward. We only change the preproces part for initializing the input ray directions by splitting the texture which stores this information in two. In one of them, the starting position for any ray *outside* the region of interest (and viceversa *in* for the other texture) is initialized as it was outside the volume. Then we perform two classical ray casting passes. For the first rendering we use the secondary TF or shading style in order to display the rays inside the cone and for the last one we use the primary TF (rays lying outside the cone).

## 3 Results and Discussion

So far, we have defined the *Virtual Magic Lantern* metaphor and briefed its implementation. There are some issues that can guide some of the implementation decisions. For inspecting 3D models, our focal region geometry originates and is guided by the Wanda device. If our inspection implies navigating through the model, this may not be the best choice, as then, it is difficult to make the lantern point exactly in front of the user position. If this happens, everything will be rendered using the primary Transfer Function. In order to solve this, we may generate an inspection cone by changing its apex by the observer position, while the direction is still guided by hand. This is a similar approach to the one taken by [Mine 1995] for ray-based object selection.

Our application was tested by about 10 users. All of them agreed the interaction technique was intuitive and behaved in a natural way. However, throughout the experiments some of the users found a shortcoming: the region pointed by the device often changed more abruptly than he or she would expect, probably because the geometry they are pointing at is closer than initially supposed. This could be alleviated by placing the apex of the cone at some distance back of the device. We are preparing a formal usability test to be done in the near future, in which we want to include some medical doctors.

We show the framerates in Table 1. All timings were computed in

| Model | Samples | RC | VML | VMW |
|---|---|---|---|---|
| Feet | 1 | 32 | 17 | 32 |
| | 2 | 18.5 | 9.8 | 18.5 |
| Manix | 1 | 18 | 13 | 18 |
| | 2 | 10 | 6.5 | 10 |
| Head | 1 | 28 | 15.3 | 28 |
| | 2 | 15 | 10 | 15 |
| Rips | 1 | 32 | 14.8 | 32 |
| | 2 | 18 | 8.4 | 18 |

**Table 1:** *Comparison of the VML method and the VMW versus the traditional volume Ray Casting (RC). All the models are* $512 \times 512 \times 512$*. Note that all renderings are in a stereo PowerWall.*

a window size of $900 \times 900$ pixels in our immersive stereo system. The rendering hardware is a 3.00 GHz Intel Core 2 Duo PC with 8GB of RAM memory and equipped with a GeForce 8800 GTX graphics card that has 768MB of RAM. In most cases, the quality obtained with a single sample per voxel is enough, although the timings for two samples per voxel are also provided. In order to have a reference, the first column shows the timings obtained for the original Ray Casting. Note that the VMW obtains the same framerates. Although the VML has a penalty, we still have interactive framerates for most models in the stereo system. When disabling stereo rendering, framerates double. These timings are obtained while rendering a model from outside, thus not covering all the viewport, because it is the typical inspection process we expect. If we get closer to the model, such that the volume covers the whole viewport, the framerates slow down roughly a 40%. Notwithstanding, note that the GPU we use is slightly old (two years), and can be replaced with a more modern one, nowadays relatively cheap.

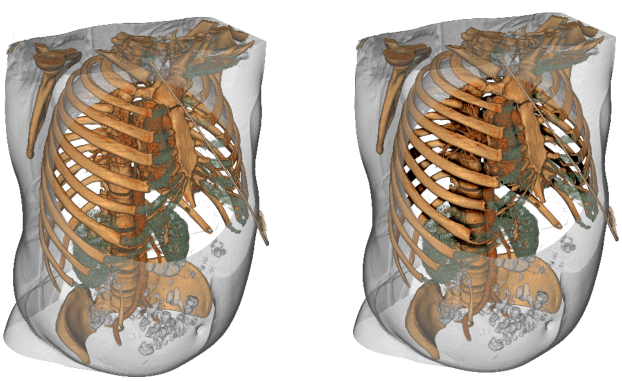


**Figure 6:** *An example of model rendered without (left) and with ambient occlusion (right) on the region of interest.*

## 4 Conclusions and Future Work

We have presented the Virtual Magic Lantern metaphor. This is a simple tool tailored to facilitate volumetric data inspection. It behaves like a lantern whose illumination cone determines the region of interest. The lantern is guided by a Wanda device that provides the axis direction and the apex position of a right circular cone, the aperture angle can be changed using the joystick and the interaction metaphor is toggled between VML and VMW with a button. The region of interest can be rendered using different shading styles that provide a feature rich volume inspection experience. The VML is particularly useful in virtual reality setups with large screens because the interaction becomes very natural and significantly widens the user inspection possibilities. We have shown that the integration of this metaphor to a classical GPU ray casting algorithm is simple and runs in real time. As a collateral implementation of the VML, we have proposed the Virtual Magic Window metaphor that it is similar to the Magic Lens approaches. The user uses the Wanda device to locate a virtual window for seeing the model through it. The VMW provides a very intuitive interaction and its implementation is very simple and efficient, incurring in no penalty with respect to the classical volume ray casting. Although, VMW is very useful for comparing rendering motifs (see Figure 6), it does not provide as much contextual information as the VML, especially in the boundary of the region of interest.

In future, we would like to explore the possible advantages of adding a clipping base to the current "infinite" cone of the lantern. Moreover, we are also working on the combination of different data models (i. CT, MRI and SPECT). We are preparing a usability test in order to get the medical doctors' comments, and deeply examine the interaction issues that may appear when navigating inside models.

## References


ANDUJAR, C., FAIREN, M., AND ARGELAGUET, F. 2006. A cost-effective approach for developing application-control guis for virtual environments. In *IEEE Symposium on 3D User Interfaces*.

BIER, E. A., STONE, M. C., PIER, K., BUXTON, W., AND DEROSE, T. D. 1993. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Annual conference on Computer graphics and interactive techniques*, ACM.

BRUCKNER, S., AND GRÖLLER, M. E. 2006. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*.

BRUCKNER, S., AND GRÖLLER, M. E. 2007. Style transfer functions for illustrative volume rendering. *Computer Graphics Forum*.

HADWIGER, M., KNISS, J. M., REZK-SALAMA, C., WEISKOPF, D., AND ENGEL, K. 2006. *Real-time Volume Graphics*. A. K. Peters, Ltd., Natick, MA, USA.

KNISS, J., KINDLMANN, G., AND HANSEN, C. 2001. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. *Visualization Conference, IEEE*.

MCINERNEY, T., AND BROUGHTON, S. 2006. Hingeslicer: interactive exploration of volume images using extended 3d slice plane widgets. In *GI'06: Proceedings of Graphics Interface*, 171–178.

MINE, M. 1995. Virtual environment interaction techniques. Tech. rep., UNC Chapel Hill CS Dept.

ROBB, R. A. 2008. Medical imaging and virtual reality: a personal perspective. *Virtual Reality 12*, 4, 235–257.

THE MAGIC LANTERN SOCIETY, 2007. A history of the magic lantern. http://www.magiclantern.org.uk/index.html.

WANG, L., ZHAO, Y., MUELLER, K., AND KAUFMAN, A. 2005. The magic volume lens: An interactive focus+context technique for volume rendering. *Visualization Conference, IEEE*.