# 3D Interaction with Volumetric Medical Data: experiencing the Wiimote

Luigi Gallo[1]
ICAR-CNR
Via Pietro Castellino 111
80131 Napoli, Italy
+390816139269
gallo.l@na.icar.cnr.it

Giuseppe De Pietro
ICAR-CNR
Via Pietro Castellino 111
80131 Napoli, Italy
+390816139503
depietro.g@na.icar.cnr.it

Ivana Marra
ICAR-CNR
Via Pietro Castellino 111
80131 Napoli, Italy
+390816139514
marra.i@na.icar.cnr.it

## ABSTRACT

Three-dimensional virtual environments are becoming more and more important for the inspection of volumetric medical data reconstructed from slices of images coming from Magnetic Resonance Imaging (MRI), Computer Tomography (CT), Positron Emission Tomography (PET) instrumentations. The possibility to visualize and interact with three-dimensional reconstructed organs while being immersed into a virtual environment, provides doctors a very naturalistic way to investigate patient's anatomy. However, most 3D user interfaces for immersive and semi-immersive virtual reality applications lack in usability or make not possible a user-friendly interaction. In this paper we present new flavors of existing 3D interaction techniques specifically designed for interacting with volumetric medical data in a semi-immersive virtual environment by using the Nintendo Wiimote controller as 3D user interface.

## Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques – *User Interfaces*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Input devices and strategies*; I.3.6 [**Computer Graphics**]: Methodology and Techniques – *Interaction techniques*; I.3.7 [**Computer Graphics**]: Three Dimensional Graphics and Realism – *Virtual reality*.

## General Terms

Design, Human Factors.

## Keywords

3D user interface, 3D interaction, Virtual Reality, Usability, Wireless, Manipulation, Pointing.

## 1. INTRODUCTION

Nowadays, imaging technologies are widely used as means of inspecting and interpreting volumetric medical data. However, in the daily clinical practice three-dimensional anatomical objects, coming from slices of images in cross sections, are still mentally reconstructed even if this is not a natural cognitive ability of humans.

Most Medical Imaging software tools assist doctors in the 3D reconstruction process of anatomical structures, since they automatically or semi-automatically reconstruct high quality 3D models from DICOM images coming from CT, MRI, or PET scans.

In the challenge of providing clinicians with a more naturalistic view of patient's anatomy, Virtual Reality plays a primary role. In fact, by providing users with depth perception it is possible to give them a more natural visualization of reconstructed anatomical objects in immersive or semi-immersive environments.

Nevertheless this ongoing transition from the 2D visualization to the 3D one in an immersive environment is not yet complete. An effective 3D interaction should be provided to users. Bowman et al. [1] define 3D interaction as "Human-computer interaction in which the user tasks are performed directly in a 3D spatial context". Usability is the key word in the development of these techniques: the interaction should be user-friendly, natural and effective.

In order to provide users a deep sense of presence into the virtual scene and experiences similar to the real ones, advanced and natural human-computer interfaces are needed. They allow highly interactive visualization and computer generated control over a 3D scene, enabling users to interact in a similar way to how they do in the real life. By giving them an effective perception of the anatomical structures under observation, their and understanding can be enhanced.

In this context, the choice and the integration of input devices, as fundamental components for immersive 3D user interfaces, are very important. Simple and non-obstructive tools, by freeing users from cables and wired devices, can provide intuitive ways of interaction with the data of interest.

In this paper we present 3D interaction techniques developed for manipulating and selecting three-dimensional medical data through the use of the Nintendo Wii controller, to be used in a semi-immersive environment.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
AMBI-SYS 2008, February 11-13, Quebec, Canada
Copyright © 2008 ICST 978-963-9799-16-5
DOI 10.4108/ICST.AMBISYS2008.2880

[1] Luigi Gallo 2nd affiliation: University of Naples 'Parthenope', Via Ammiraglio Fernando Acton 38, 80133 Napoli, Italy. Tel. +390815474893, e-mail: luigi.gallo@uniparthenope.it

These 3D interaction techniques have been implemented by using the Visualization Toolkit (VTK) library, and have been integrated into an open-source and cross-platform Medical Imaging ToolKit (MITO) we have developed [2].

## 2. 3D INTERACTION REQUIREMENTS

The interaction techniques we propose are not suitable for a generic virtual reality application. They have been specifically designed for the application, the task, the domain, the device and the "typical" user of our scenario [1].

In order to improve the usability of the developed techniques, we have primarily considered the clinicians needs with respect to the medical image analysis field.

Our study has outlined some requirements which must be held in due consideration:

- non-obstructive hardware components should be preferred to the obstructive ones. Clinicians prefer to use natural interface [3]. Following this point of view a semi-immersive virtual environment should be preferred to the fully-immersive one, and wireless input devices should be preferred to wired ones;
- intuitive interaction techniques should be preferred to complex ones so to assure an adequate degree of friendliness. Clinicians are not inclined to use new tools if they require time-consuming training and system configuration;
- a near-real-time interactivity [4] should be provided in order to allow a natural interaction;
- a handheld 3D User Interface (UI) must be ergonomic [5]. If clinicians have to use the virtual reality system for a long time, an input device too heavy or difficult to handle can be rejected;
- not all the universal 3D tasks (navigation, selection, manipulation, system control) [6] are required in the interaction with volumetric medical data. There is only one object in the scene, the one reconstructed from the 2D slices. In this context there is no need to grab (select) an object in the environment but only to "point to" a part of it. The navigation task is unnecessary in our application area, whilst the manipulation one is essential to better inspect the data of interest.

## 3. INTERACTION TECHNIQUES

Following the most recent directives aimed at improving the usability of 3D interaction techniques in real-world applications [1], the specificity issue has held a primary role in our design. Four different types of specificity have been considered:

- **Application-specificity**: our killer applications are software tools that apply virtual reality technologies for medical and biological image analysis.
- **Task-specificity**: thanks to continuous feedbacks from clinicians, we have identified the necessary tasks for the application area of interest. For example, the travel (navigation) task has been considered unnecessary, whilst the manipulation one has been considered the most important.
- **Device-specificity**: once we have chosen the 3D User device, interaction techniques have been adapted in order to meet its peculiar features.
- **User-specificity**: clinicians have played a primary role in our design. VE experts use different strategies in the interaction with volumetric data. In order to design a user-friendly interaction, clinicians have to be able to use their real-life skills. From this point of view, a natural interaction should be preferred to a smart but unnatural one.

During the design process, we have asked clinicians, step-by-step, to identify their interaction requirements. The software prototype has played a primary role in this process. Every time clinicians identified a new requirement, they were asked to test its implementation so to validate or modify it.

The interaction features we have identified are the following:

- *Object rotation / translation* – to visualize the model from all possible points of view;
- *Zoom in / out* – to better focus regions of data;
- *Object cropping* – to cut off and view inside the data;
- *Pointing* – to point out a precise point of the visualized data;
- *Control at run-time of the depth perception* – to move inward / outward the object in the virtual space.

Once these features have been identified, we have modeled the user interaction by using a Finite State Machine (in Figure 1).

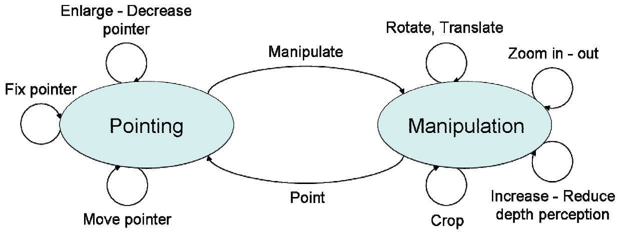

**Figure 1. 3D interaction Finite State machine.**

The system can be in two states: the *Pointing* state, in which the input device is used like a laser pointer-style; and the *Manipulation* state, where the object can be rotated simply by wheeling the input device in the real space. The user can switch between these two states by pushing a button.

### 3.1 Manipulation state

In this state, the 3D object can be rotated, translated or cropped. The interaction technique used to rotate the objects is quite similar to the *arm-extension* one [9]. In the arm-extension technique the object manipulation can be performed through a data glove device, whereas in the proposed technique user has to rotate a handheld input device. The object in the virtual space will rotate along the same axis and with the same angle of the input device.

We have thought up two different versions of the rotation technique:

- *Trackball* style: the 3D object follows exactly the input device rotations when the rotate button is pressed. In order to rotate continuously the model along an axis, the input device must be utilized as a screwdriver (i.e. the user has to press the rotate button, rotate the device along the desired axis, release the button and then repeat the operation).

- *Joystick* style: until the rotate button is pressed on the input device, the 3D object continuously rotates along the device rotation angle.

In the manipulation state user can also zoom in or zoom out the object and increase or reduce the horizontal parallax and the eye angle, so to modify the depth perception. The effects of these two features are really different: if the depth perception is increased, the user will visualize the object closer to him; if the object is zoomed, the user will visualize a bigger object but at the same distance from him.

The cropping functionality is implemented through a cropping box widget that can be rotated, translated and scaled at run-time by using the input device features. More information about the cropping technique can be found in [10].

### 3.2 Pointing state

In this state, the input device is used as a laser pointer-style device. A *flavor* [1] of the classic *ray-casting* technique has been used in order to control the position in the space of a three-dimensional cursor. In the ray-casting technique a virtual light ray is used to select an object, the user has only to move it to the desired position. If this technique is applied to move a cursor in a virtual environment, a problem arises: how to move it inward or outward? In order to solve this problem, we have chosen to adopt the *fishing reel* flavor of the ray-casting technique [9], in which users can move the cursor closer or farther away by using two buttons.

Once a point of interest has been located, the cursor can be fixed by simply pressing a button on the input device. In order to adapt the pointer size to the visualized object, it can be increased or decreased via two buttons.

## 4. THE WIIMOTE DEVICE

Starting from the above considerations, and in accordance with clinicians feedbacks, we have gathered that a suitable 3D UI to interact with volumetric medical data should be:

- Wireless;
- Ergonomic;
- Suitable for a near-real-time interactivity;
- Suitable to implement a pointing feature and to manipulate the object with 6 DOF.

After a brief research among all the off-the-shelf 3D user interfaces, we have chosen a wireless, ergonomic, economical input device: the Wiimote.

The Wiimote, alias Wii Controller or Wii Remote, is the primary controller for the Nintendo's Wii console. Thanks to its motion sensing capability, the Wiimote allows users to interact with and manipulate items on the screen by simply moving it in the space.

The easiness of use of this controller has made it very popular on the web. Many unofficial websites provide accurate technical information obtained by a reverse engineering process [7], [8].

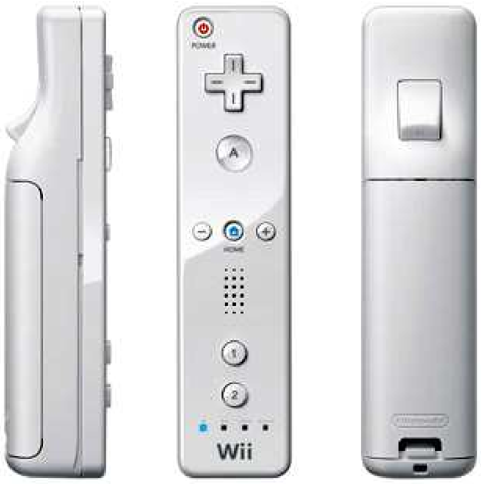

**Figure 2. The Wii Controller.**

In this section we briefly introduce the most important Wiimote features.

### 4.1 Communication

The Wiimote communicates via a Bluetooth wireless link. It follows the Bluetooth *Human Interface Device* (HID) standard, which is directly based upon the USB HID standard. It is able to send reports to the host with a maximum frequency of 100 reports per second. The Wiimote does not require authentication: once put in the discoverable mode, a Bluetooth HID driver on the host can query it and establish the connection.

### 4.2 Inputs

The controller’s movements can be sensed, over a range of +/- 3g with 10% sensivity, thanks to a *3-axis linear accelerometer*. So it is possible to detect the controller orientation in the space. Calibration data are stored in the Wiimote’s flash memory, and can be modified via software.

Another Wiimote’s feature is the *optical sensor* placed in front of it, which is able to track up to four infrared (IR) hotspots. By tracking the position of these points in the 2D camera field of view, accurate pointing information can be derived.

There are 12 buttons on the Wiimote. Four of them are arranged into a directional pad, and the other ones spread over the controller.

### 4.3 Outputs

The Wiimote can send outputs by using three different modalities: switching on/off up to four blue LEDs; vibrating itself and emitting sounds.

## 5. IMPLEMENTATION DETAILS

The proposed interaction techniques and the driver, implemented to use the Wiimote as a 3D UI, have been integrated into the MITO system.

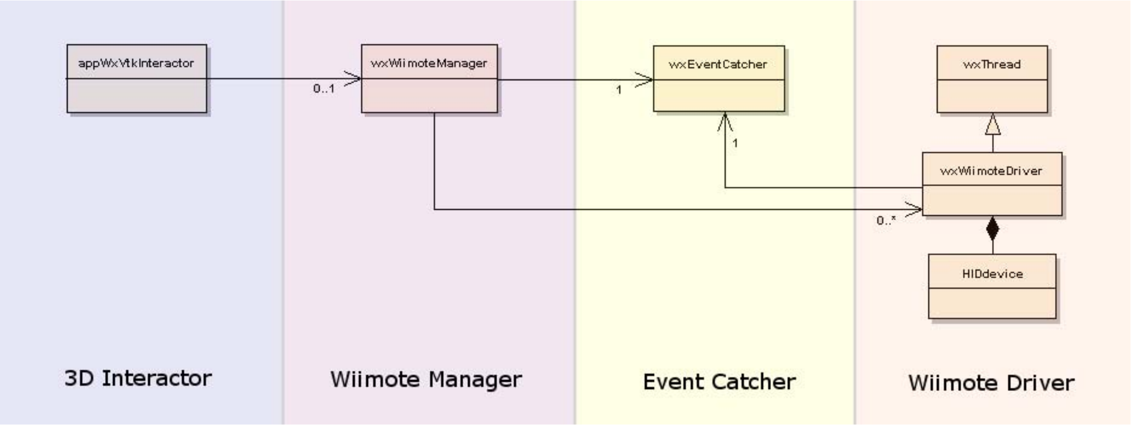

**Figure 3. Class-Diagram of the Wiimote integration in MITO.**

MITO has been written entirely in C++ and built on open-source and cross-platform libraries, which were combined and extended in order to support medical imaging processing, semi-immersive 3D model visualization and interaction functionalities. The used libraries are: OpenGL – for fast 2D and 3D display; VTK (Visualization ToolKit) – for 3D rendering; ITK (Insight segmentation and registration Toolkit) – for image segmentation and registration; WxWidgets (Windows and X widgets) – for graphical user interface.

The Wiimote integration has been performed in two steps; in the first step we have designed and implemented the Wiimote driver, in the second one we have implemented the interaction techniques previously designed.

## 5.1 The Wiimote driver

The Wiimote uses the standard Bluetooth Human Interface Device in order to communicate with the host. Any Bluetooth host is able to detect the Wii controller as a standard input device. But, the Wiimote does not use the HID standard data type, only defines the length of its reports. This is the reason why standard HID drivers cannot be used. In this first step, we have developed a driver for Windows 2000/XP operating systems. In order to do this, we have made use of the Microsoft Device Driver Kit (DDK) library.

The lowest level class we have implemented is *HIDdevice*, which methods allow to fulfil all the operations required to connect and to communicate with a generic HID device. By using this class, it is possible to search for a connected HID device and to read / write data from / to the Wiimote.

The *wxWiimoteDriver* class is built upon the HIDdevice class and inherits from the *wxThread* class. It allows exploiting its Wiimote state knowledge in order to create detailed wxWidgets custom events comprising relevant information such as the state of buttons, the orientation of the device, the led status, the direction of pointing and so on.

Since the Wiimote transmits complex information, we have developed a *wxWiimoteEvent* class in order to define a proper event type for the Wii Controller device. In MITO these events (mouse and keyboard generated events too) are handled by an event catcher, which is defined by the *wxEventCatcher* class.

Whereas all these classes are generic and suitable for any kind of application, the *wxWiimoteManager* class allows to customize the Wiimote events for our system. This class was named “manager” because, as we are planning to develop a multi-user (so multi-Wiimote) interaction, in the new implementation its instance will also handle the multi-user coordination issues.

Finally the *appWxVtkInteractor* class makes it possible to bring the Wiimote events to the 3D scene, allowing the user-machine interaction. From the implementation point of view, this simply means to convert a wxWidgets event in the corresponding VTK event.

## 5.2 Interaction techniques

During the 3D interaction with the Wiimote, the system can be into two states: pointing and manipulation (see Figure 1). According to the interaction state, user actions have different effects (see Table 1).

**Table 1. Wiimote buttons mapping table**

| | **Pointing state** | **Manipulation state** |
|---|---|---|
| **A** | Fix the pointer | Crop the volume |
| **B** | Show/Hide the pointer | Rotate the volume |
| - | Move outward the pointer | Move outward the volume |
| + | Move inward the pointer | Move inward the volume |
| **1** | Enlarge the pointer | Zoom out |
| **2** | Reduce the pointer | Zoom in |
| **PAD** | | Translate the volume |
| **HOME** | Switch to manipulation state | Switch to pointing state |

The Wiimote motion sensing capability is differently interpreted in each state. In the manipulation state, when the user presses the B button, he can rotate the volume by whirling the input device. In this state, we use both the optical sensor and the accelerometer in order to determine the Wiimote position. In the pointing state, instead, only the optical sensor is used.

VTK handles the interaction with a 3D scene by using *interactor styles*. Mouse and keyboard events are handled in different ways depending on the interactor style that is active in the scene. In order to allow the Wiimote integration, we have implemented two new interactor styles, later included in the VTK package. The added classes are:

- *vtkInteractorStyleWiiJoystickStyle* – for a joystick style interaction (i.e. continuous rotation of the volume according to the Wiimote orientation);
- *vtkInteractorStyleWiiTrackballStyle* – for a trackball style interaction (i.e. rotation of the volume proportional to the Wiimote rotation).

The pointing functionality, instead, does not require a new interactor style. The pointer, in fact, is only a new 3D object added to the scene. The 3D cursor, when the volume is rotated, rotates together with it. This allows users to point a particular region, fix there the cursor, and then rotate the volume in order to visualize that region from other viewpoints.

## 6. THE VIRTUAL ENVIRONMENT

The hardware environment we used for preliminary experiments involves:

- a workstation equipped with a processor Intel Pentium 4 3.20 GHz, 1024 Mbytes DDR2 266 MHz (16 KBytes of cache L1, 2048 KBytes of cache L2), graphics card NVIDIA Quadro FX 3500 (RAMDAC frequency 400 MHz, 256 MBytes GDDR3 SDRAM 256-bit) on PCI Express x16a;
- two BenQ PB8250 DLP projectors equipped with linear polarizing filters;
- linear polarized glasses;
- non-depolarizing silver screen.

The Wiimote, in order to be used as a laser pointer-style device, needs to track from one to four fixed source of infrared radiations. In the Wii console, the Nintendo Sensor Bar is used, which is about 20 cm (8 in) in length and features ten infrared LEDs, with five LEDs being arranged at each end of the bar. The Sensor Bar allows to use the Wiimote as an accurate pointing device up to 5 meters (approx. 16 ft) away from the bar. Since the Sensor Bar can only be connected to a Wii console, we have had to build our own sensor bar, equipped with 8 IR LEDs and 2 Blue LEDs. This home-made bar is more powerful of the original one, since it makes possible to use the Wiimote up to 10 meters (approx. 32 ft) away from it.

It is not necessary to point directly at the Sensor Bar, but pointing significantly away from the bar will disrupt position-sensing ability due to the limited viewing angle of the Wii Remote.

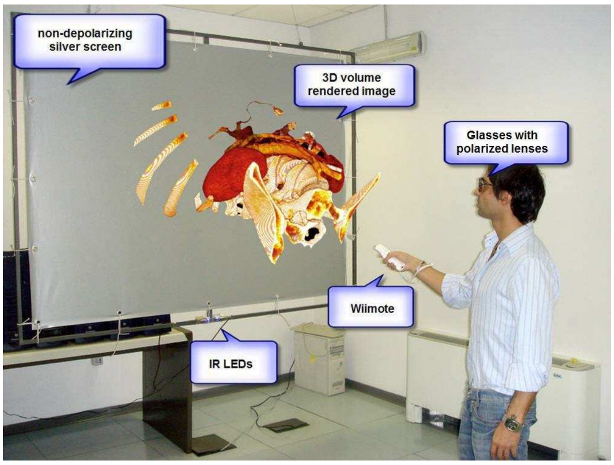

**Figure 4. The semi-immersive 3D interaction.**

## 7. CONCLUSIONS AND FUTURE WORK

In the daily clinical practice, two-dimensional image slices, coming from medical imaging instrumentations, are often used to mentally reconstruct the three-dimensional organs of interest. 3D computer-aided reconstruction provides a great support to this process, exempting clinicians from this time-consuming effort. Immersive virtual reality technologies, by adding a spatial perception to the reconstructed anatomical models, offer the possibility to visualize and inspect them in a more intuitive way. However, in order to enhance the degree of interaction and gain a more and more natural control over the synthetic objects, user-friendly techniques and interfaces are required.

In this paper we have proposed new flavors of existing 3D interaction techniques to immerse clinicians into a virtual environment and let them interact with human anatomy in a more natural manner. To reach this aim we have used a wireless, ergonomic and intuitive device, the Nintendo Wii controller, for manipulation and pointing tasks.

From these first results, we now aim at integrating other advanced and natural methods of interaction, so to develop a multimodal interaction interface for examining volumetric medical data. In particular, we are planning to integrate the speech modality, so to let users issue voice commands to the system. Moreover, in order to provide a suitable tool for computer-assisted education of clinicians, a multi-user interaction approach is under consideration too.

## 8. ACKNOWLEDGMENTS

Our thanks to Prof. Bruno Alfano and Prof. Arturo Brunetti, from the University “Federico II” of Naples, for their precious feedbacks during the system development.

## 9. REFERENCES

[1] Bowman, D. A., Chen, J., Wrave, C. A, Lucas, J., Ray, A., Polys, N. F., Li, Q., Haciahmetoglu, Y., Kim, J. S., Kim, S., Boehringer, R. and Ni, T., “New Directions in 3D User Interfaces”, The International Journal of Virtual Reality, 5(2): 3-14 (2006).

[2] Coronato, A., De Pietro, G., and Marra, I. 2006. An Open-Source Software Architecture for Immersive Medical Imaging. In Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (La Coruna, Spain, July 10 – 12, 2006). VECIMS 2006.

[3] Abaci, T., de Bondeli, R., Cíger, J., Clavien, M., Erol, F., Gutiérrez, M., Noverraz, S., Renault, O., Vexo, F. and Thalmann, D. 2003. The Enigma of the Sphinx. In Proceedings of the 2nd International Conference on Cyberworlds (Singapore, December 3 – 5, 2003). CW 2003.

[4] Myers, B. A., Hudson, S. E. and Pausch, R. F., "Past, present, and future of user interface software tools". ACM Trans. Comput.-Hum. Interact. 7(1): 3-28 (2000).

[5] Patel, H., Stefani, O., Sharples, S., Hoffmann, H., Karaseitanidis, I. and Amditis, A., "Human centred design of 3-D interaction devices to control virtual environments", International Journal of Man-Machine Studies, 64(3): 207-220 (2006).

[6] Bowman, D. A., Kruijff, E., LaViola, J. J. Jr. and Poupyrev, I., "An Introduction to 3D User Interface Design", Presence 10(1): 96-108 (2001).

[7] WiiLi, a GNU/Linux port for the Nintendo Wii, http://www.wiili.org

[8] WiiBrew Wiki, http://wiibrew.org/index.php?title=Wiimote

[9] Bowman, D. A. and Hodges, L. F. 1997. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. In Proceedings of the 1997 Symposium on Interactive 3D Graphics (Providence, RI, USA, April 27 - 30, 1997). SI3D 1997.

[10] De Pietro, G., Gallo, L., Marra, I. and Vanzanella, C. 2007. A New Approach For Handling 3D Medical Data In An Immersive Environment. In Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (Ostuni, Italy, June 25 – 27, 2007). VECIMS 2007.