

**ESDA2012-82592**

## VIRTUAL REALITY INTERFACES APPLIED TO WEB-BASED 3D E-COMMERCE

**Unai Elordi\***  
**Alvaro Segura**  
**Jon Goenetxea**  
**Aitor Moreno**

Vicomtech-IK4  
Mikeletegi 57, 20009 San Sebastian  
Spain  
Email: uelordi@vicomtech.org  
asegura@vicomtech.org  
jgoenetxea@vicomtech.org  
amoreno@vicomtech.org

**Jon Arambarri**

Virtualware  
Usausuaga 7, 48970 Basauri  
Spain  
Email: jarambarri@virtualware.es

### ABSTRACT

*E-commerce is well established on today's Internet and has created its own set of techniques for performing and optimizing sales. Traditional shops on the other hand provide a physical space in which decoration and space are relevant. In this paper we present innovative interfaces that join the concepts of real shops and on-line e-commerce in a Web-based virtual reality platform.*

*The idea involves making the e-shopping experience more intuitive, closer to the experience of visiting a real shop, and providing on-line 3D interfaces for the entire process from shop design and product placement to customer experience analysis. We researched traditional marketing and sales techniques and the way they can be transferred to 3D virtual reality environments. Web 3D technologies allow us to implement interactive virtual worlds on the Web, accessible for everyone over different platforms and without special applications or plug-ins.*

---

\* Address all correspondence to this author.

### INTRODUCTION

In the last years there have been several attempts at uniting electronic commerce and virtual reality. While conventional e-commerce experienced a spectacular growth overcoming initial trustworthiness problems [1] and general reluctance, virtual 3D shopping was not very successful.

Virtual Reality strictly refers to complete hardware-software systems that immerse users in a virtual world by means of motion/orientation tracking devices and immersive 3D displays. In this paper we use Virtual Reality as a traslation of that concept into the Web with no specialized interaction devices beyond a conventional PC. It is thus a lower complexity substitute of full-fledged Virtual Reality that brings a comparable 3D immersive first-person experience to Web browsing within its constraints.

3D e-shopping, as e-commerce in general, provides consumers with convenience and economy [2]. Consumers are used to finding and buying products online based on photographs and descriptions. They can even compare prices and features and e-commerce sites become an enhanced form of *catalogue merchant*.

But shopping is more than examining a catalogue and ordering. There is an emotional side in visiting real shops. Merchants

pay a lot of attention to decoration, atmosphere and good use of the space. Marketing strategies are obviously different in e-commerce and real shops where spatial placement is relevant. In shops, for example, important goods are placed at eye's height, essential products are placed in distant spots so that the user must walk through the shop and might be attracted by other products along the way.

In theory virtual 3D stores can also provide a user experience that is closer to that of real shopping, compared to conventional e-commerce. We consider the limited success of virtual shopping to be related to two kinds of factors: those affecting the consumer and those affecting the retailer. VR usually comes in the form of standalone applications (such as games) or Web-based applications that require the installation of a specific plug-in for rendering 3D graphics. Both need easier tools without the hassle of installing more software in their PCs.

Now the new standards for rich internet applications and plug-in-less 3D in the Web enable the creation of more accessible Web platforms where retailers can create and manage their virtual stores and consumers can visit, browse and buy goods. This medium has its own limitations compared to high-end virtual reality systems that will be discussed in the paper.

In our research, we tried to find innovative ways to build an integrated online 3D VR e-shopping platform. Our platform is a centralised Web application that both retailers/designers and consumers access in different ways. In fact, it is novel virtual user interface to e-commerce divided in two parts: a shop creation and management interface and a virtual shop interface for visitors. The management part includes tools to analyse the behaviour of visitors, information that can be exploited to enhance sales. All data is stored in the server and all the users need is a modern Web browser supporting the WebGL standard.

We implemented different navigation and interaction methods for designers and visitors: first person walking with collision detection, fly-over, object selection by picking, object placement and translation by dragging and dropping, etc. The concept seamlessly integrates 3D virtual reality elements with classical Web interface elements such as product catalogues with thumbnails and the shopping cart mechanism. The interfaces proposed are intuitive enough not to demand shop administrators high 3D modelling skills to create and configure the virtual shop. The 3D environment allows us to research about the virtualization of traditional marketing techniques.

The following sections begin with a brief state of the art on the subject and present our concept of a VR interface for virtual electronic commerce in the Web. The medium chosen is convenient for its accessibility and ubiquity but faces important limitations with respect to the virtual reality techniques that can be applied. The technology that all the interface is based on, is standard Web technology, not depending on any proprietary technology or add-ons: HTML5, CSS, JavaScript and WebGL. This paper discusses how these can be used to create VR-like 3D

interfaces for e-commerce.

## STATE OF THE ART

Conventional electronic commerce is well established nowadays and generates a considerable amount of Internet traffic and business. E-commerce sites provide the means for customers to find products of their interest and actually purchase them. These sites have overcome initial human factors problems of *trust* in online payments and user experience as discussed in [1]. Concepts such as the virtual shopping cart make the process easier introducing a metaphor of real life commerce but the process is still different. The Web site acts as an online electronic catalogue where users can look up a product, read a description and see a picture of it, and finally buy it. The experience is completely different from that of visiting a local modern shop designed with visitor comfort in mind, walking around, browsing and examining objects exhibited, and buying them.

Virtual Reality interfaces to e-commerce have seen some attempts with little success. As explained in [3], a VR store has the advantage that it can be closer to the real-world shopping experience, more familiar to the customer. That research presents guidelines for the design of effective Virtual Reality stores. VR shopping also brings opportunities to implement traditional marketing and sales optimization techniques from the real world into online commerce.

Kinset [4], the product of a 2007 Boston start-up, attempted to provide new tools to create and publish functional 3D virtual stores. Kinset's tools were applications that users had to install and run: an editing application for creators and a 3D browsing and purchasing application for customers in a SecondLife style. It is accepted [5] that most users, especially novice or non-technical users, are bothered by the requirement of installing specific software or browser plug-ins and the current trend is to provide desktop-like functionality in standard Web applications.

More recently, in 2008, *VirtuyMall* [6] was presented in 2008 as an innovative e-commerce platform. *Virtuy* relies on the *Unity Web Player* browser plug-in to present interactive virtual stores in a 3D mall which display their products with information and allow actual purchasing. The user is presented a third person view that shows his/her 3D avatar from behind as it moves around the environment. The shops in *Virtuy* are 3D models, but products for sale in them are not, being simple flat images rendered as billboards (i.e. sprites) on the shelves. Representing the user's own avatar helps in the perception of sizes and enables social functionalities when other users are displayed as well.

The *Virtuy* platform includes a control panel for store owners to manage the products displayed and information on pricing and availability. But the core of the virtual reality interaction is on the customer's side. Interaction is limited to navigation around the scene (third person walk with collision detection) and clicking on products to add them to the cart.

Several solutions have been presented with the intent of enhancing interactivity within the virtual world bringing more realism by adding external control devices (Nintendo's Wiimote, home-made devices, motion trackers) and implementing methods to navigate the shop [7]. In fact, proper virtual reality is usually that which provides an immersive experience through the use of special input and output devices. For example, the well-known interaction techniques described in [8] assume the presence of 6-DOF interaction hardware or other devices. The work described in this paper, targeted at the Web, brings as much as possible said interaction and visualization techniques to a comparatively limited medium.

Part of the platform presented deals with the creation and decoration of virtual shops. This concept has similarities with online home design tools such as Autodesk's *HomeStyler* [9]. This application is very complete and user-friendly in terms of interaction and usability, but is lacking as a virtual reality tool. Its 3D view is in fact a pseudo-3D orthographic rendering supporting eight possible viewpoints, and it is based on Adobe's proprietary Flash technology, something we intend to avoid.

Aside from other elements, virtual reality shops on the Web require some kind of Web 3D technology, one that enables the integration of real-time 3D interactive computer graphics in Web pages on the Internet. Several standards have been presented since the mid 1990s, such as VRML and X3D, as well as proprietary systems like Quest3D and the now popular Unity3D. They are all high level declarative (i.e. 3D scene description-based) systems and their use require users to install specific browser plug-ins. In contrast, the recently approved WebGL standard [10] proposes a low level imperative API for 3D graphics that works natively in modern Web browsers. WebGL itself does not provide any navigation or interaction controls so all of that has to be implemented almost from scratch.

## METHODOLOGY

Our concept brings together several technology areas: modern Web standards and technologies, Web 3D technology, virtual reality, and e-commerce. The resulting integrated platform uses them to provide retailers with an online tool to design virtual stores and make them available for operation. Users of this system must be logged in and have different profiles as managers or users that affect the interfaces that will be presented to them.

There is no standard way of creating 3D applications on the Web. Our approach combines the classical 2D web interfaces, such as form controls, lists with icons and drag and drop, with 3D elements such as virtual object viewers and immersive scenes. The virtual world presented is common among administrators and users, but each profile has its own interaction and navigation modes. For example, the administrator can modify decoration and products on display, and regular users can navigate the scene, request information on products and add them to

their cart.

The following subsections describe the system and the interaction modes between the different kinds of users and the application. First, the architecture of the system is presented. Then, the interface for shop creation and product placement is explained, followed by the interface for visiting users. The system includes functionality to analyze the behavior of visitors which is presented in the last subsection.

## Architecture and data model

Classical Web applications are based on the transmission of formatted pages from the server for every request of information and every change in content. Modern applications use dynamic changes to the page elements to create a more fluid experience closer to desktop applications. The architecture of our platform is distributed and relies heavily on dynamic content through HTML DOM manipulation and Ajax requests [11].

The Web page and 3D content on the client side provide a user interface and the logic and data storage residing in the server are accessed via asynchronous Ajax calls (see figure 1) that exchange data in JSON format. Some information requested by the client might be held by a third party (Remote Web Service). In that case the request is forwarded to an external source by a SOAP Web services client.

A virtual shop is not just a 3D polygonal model. In order for it to work as expected it needs semantics not supported by generic 3D object formats. We designed a *data model* as part of the architecture. A virtual shop is basically a structure that contains (i) one definition of the shape of the environment, (ii) a set of 3D static objects for decoration and (iii) a set of objects representative of products for sale. Decoration objects and products include positioning data that will be interactively editable in the design interface, and products have additional properties (reference ID, description, price, product class-dependent properties). The reference ID of each product is what links the virtual environment with the actual e-commerce system enabling stock querying and purchasing.

## Shop creation

Shop creation is the process of designing a new shop's shape and appearance and placing virtual objects that will represent products for sale. This is the part that managers will use to create new shops and also to update their contents and distribution.

We consider that managers do not need to be specialists in 3D modelling and graphical design so the platform has to provide simple enough tools. As explained before, data model description has a shape definition, which is a base where all other objects are placed. This base environment can be interactively created by the designer or imported from a third party full fledged 3D modelling tool. The idea is that anyone can easily design a simple shop with our parametric builder, the possibility is open for au-

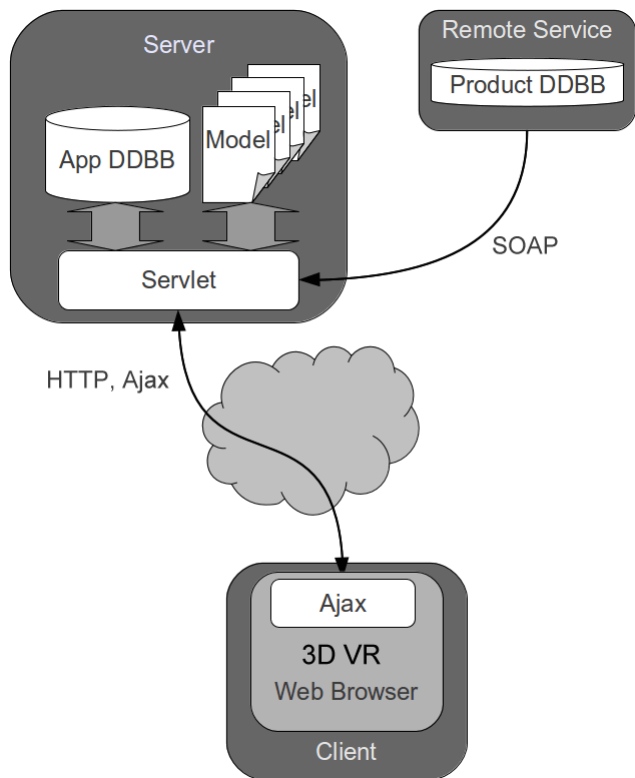


FIGURE 1. System architecture

thors to use or hire modelling resources able to design detailed environments with tools such as 3Ds Max or Blender.

We designed two ways to create a new shop. On the one hand there are parametric shop templates based on simple shapes like rectangles, L-shapes or free-form polygons. For any of those contour shapes, the designer can specify materials for ground, ceiling and walls and a wall height, and the system will create a shop geometry by extruding the contour vertically. On the other hand, if the designer is skilled in a 3D modelling tool or wants to subcontract this task the system allows him to import a pre-designed scene model by a simple drag and drop of a Zip archive into the Web interface.

To choose the desired creation method, when starting a new shop, the designer is presented with the following four options:

**Rectangular** : This option creates a rectangular layout based on 3 editable parameters: width, depth and height.

**L-shaped** : This creates a layout with the shape of an 'L' with selectable full length, full depth, short length, short height and height.

**Free-form layout** : In this case a 2D grid is presented representing a top view of the space. The user can draw a polygon for the shop's layout by clicking points on it with the mouse. A point clicked close to the first point closes the polygon. To

simplify editing there is a snap mode that only allows horizontal and vertical lines to be drawn.

**Imported scene** : This option allows uploading a 3D model of an empty shop. The user must pack a model in COLLADA format along with its textures in a Zip archive and drag it into the Web interface. The design of this model, as well as all models used for decoration or products, must use a common scale and up-axis orientation so that all elements fit. The uploaded model will be stored and registered in the shops database, but only for this administrator.

### Shop, product and decoration model management

The process so far creates an empty space for a shop. Now the administrator can add furniture and other decorative objects to the scene, and finally place the products that will be available for purchase.

All these objects will be represented as 3D models in the scene, and they must be available in the server when the virtual shop is rendered in the browser. The system then needs functionality to upload new objects because of this objects can be used in the scenes.

For administrators, there are some forms to define the models they need, and add them to the system. Those models are stored in the server as COLLADA definition XML files and they are accessible from the Internet for all the customers and administrators. They are published in the server as other kind of web resources (web pages, images, etc.) and a browser can take each of them by requesting its URL.

Administrators can easily upload new models to the platform (empty shop scenes, decoration objects or products), and define their properties (name, description, reference number, etc.) which will be stored in the database.

In case of products, it is necessary to associate the object with the real product it represents so that it can be effectively purchased. These models have a *reference* property that the system will use to request current data about it such as price or available stock when necessary.

All products need an associated 3D model to appear in the virtual shop. Since their creation can be too complicated it can be helpful to provide simpler methods for some special cases. With this intention the system can generate models for products based on an image. It produces a flat object and applies the image to it as a texture map. This type of generation is useful for semi-flat products, for example T-shirts, ties or pictures. These objects can be represented by an image with a transparency channel defining their contour.

The stored models are associated with the administrator's account and with his shop group. A shop group is formed by all the shops a manager can access. So, when a manager is editing a shop, he can access all his objects uploaded for that shop or for other shops in the same group. The id, reference, position and

orientation of the objects will be stored in the shop's data model file when the user saves the scene.

### Interaction and navigation

Both administrators and regular users are presented with the virtual 3D scene of the shop. However, as their goals are different, the interaction and navigation modes are different for each kind of user. On the one hand, administrators must be able to modify the environment, both altering decoration and replacing products. On the other hand, visitors must be able to move around the shop as they would in a real one, browse products and request information, add them to their shopping cart and finally buy them.

In both cases most interaction takes place in the 3D world, but sometimes a 2D interface or a combination between 2D and 3D elements are better suited. For example, the shopping cart contents with partial and total price is displayed in a table. And adding objects to the scene is made by dragging thumbnails from a list on the panel and dropping them in the 3D world.

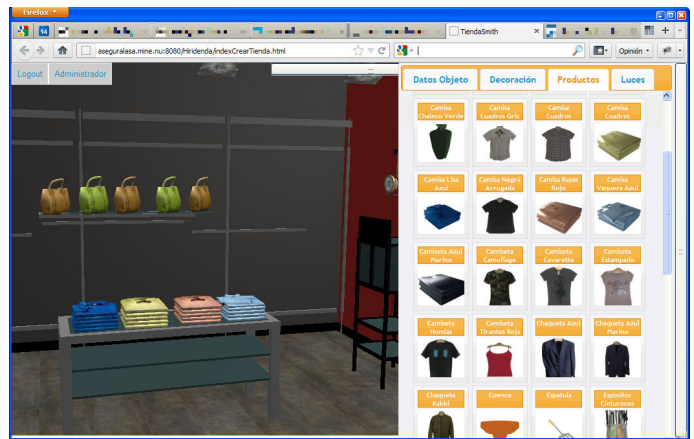
**Administrator profile** The administrator profile is targeted at design and maintenance of the 3D shop and its contents. For editing, a first person navigation mode would be cumbersome even if it would better mimic real life shop decoration and product placement. Instead, we decided that more freedom of movement would be better suited, as if manipulating a miniature model of the shop. An administrator must be able to add and precisely place objects in the scene, both decorative and products, as well as move, rotate or remove them.

Navigation in this profile is a fly over mode with no collision detection. The designer should be able to reach all parts of the scene easily and does not need a realistic walking experience. The user can use the keyboard to move around and the mouse to rotate the view. Mouse-based look can apply different mappings of mouse movements and button presses into rotations around the axes. We selected one based on mouse press and drag which maps horizontal mouse displacement to horizontal rotation speed and vertical mouse displacement to vertical rotation angle. Mixing of speed and position control can seem confusing but it is quite common and relatively natural.

Object manipulation is the most demanding aspect for the manager with respect to interaction. Most 3D modelling applications have quite standard interactive manipulation controls to move, rotate and scale objects in all three axes freely. This platform is definitely not a full 3D editor which would be unnecessarily complicated for non experts. We prefer to provide a more natural interface with these premises: (i) in real life one does not scale objects, their dimensions are fixed; (ii) in real life one can't place objects floating in the air, they always rest on some surface; (iii) in real life furniture and most objects have an up-axis and one does not bend them or turn them upside down, but

one may rotate furniture around its vertical axis to find a good orientation.

Based on these rules, we designed controls to naturally add and place objects on the scene (see figure 2). A pair of catalogue-like lists on the right panel contain thumbnail images (right side of figure 2) of decoration objects and products. The user can click and drag those small images from the panel and drop them on the 3D scene. The corresponding 3D object will be added to the scene in the precise point where it was dropped. In order to find that spot, a ray is shot from the camera to the pixel where the mouse button was released and the intersection point with the scene is computed. That makes the object stay on an existing surface.

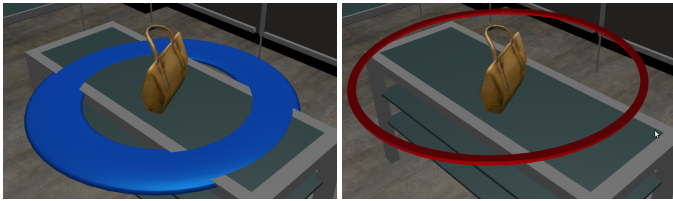


**FIGURE 2.** A snapshot of the administrator interface, showing the list of products which the current manager can add to the scene by a mouse drag&drop action

An object added to the scene once selected can be moved to a new location and rotated around its vertical axis. A simple click with the middle button selects the object which is highlighted by a blue ring shaped manipulator around its base (see figure 3 (left)). Now the user can drag the object around the environment with the mouse. As before, on every new mouse position while dragging an intersecting ray is shot against the scene in order to keep the object always resting on a surface. In order to rotate or remove the object, when the user right-clicks on it, an animated contextual menu pops up on it showing several action icons. Selecting the rotation option a red ring appears around the object (see figure 3 (right)) which, when dragged with the mouse, makes the object rotate.

Selection of objects and dragging them is based on a picking function (see figure 4) that also intersects a ray from the camera location through the mouse position against all objects of the scene to find the closest intersecting mesh. Fortunately these intersection computations can be hardware accelerated as

is common in real time computer graphics by performing a special render cycle of one pixel.



**FIGURE 3.** Two different interactive manipulators that user can see in the application, the first (left picture) as selection marker, and the second (right picture) as rotation manipulator

**User profile** The user profile is applied to all visitors of the shops and is targeted at navigating the shop as in real life, browsing, selecting products and buying. The interaction model is simpler than that of the administrator because the user must not be able to modify the environment.

Navigation here is a walk-type mode in first person view. The user can only walk around on the ground and look 360° around the vertical axis and 180° around a horizontal axis to look up and down to simulate real human motion.

Customers must not be able to go through walls, products or decoration so a collision detection mechanism is needed. By shooting a ray from the camera position to the user’s forward direction and finding its intersection with the scene the navigation system can detect obstacles that would prevent walking forward. If we want to go upstairs, we need a gravity vector. Similarly, for floor following a vertical ray is shot downwards. A complete physics simulation engine could be added to improve the behavior of objects but since objects are static it is not considered essential and it would be computationally expensive.

As we are dealing with Web technologies the only input devices are keyboard and mouse. We designed the following methods to navigate the shop.

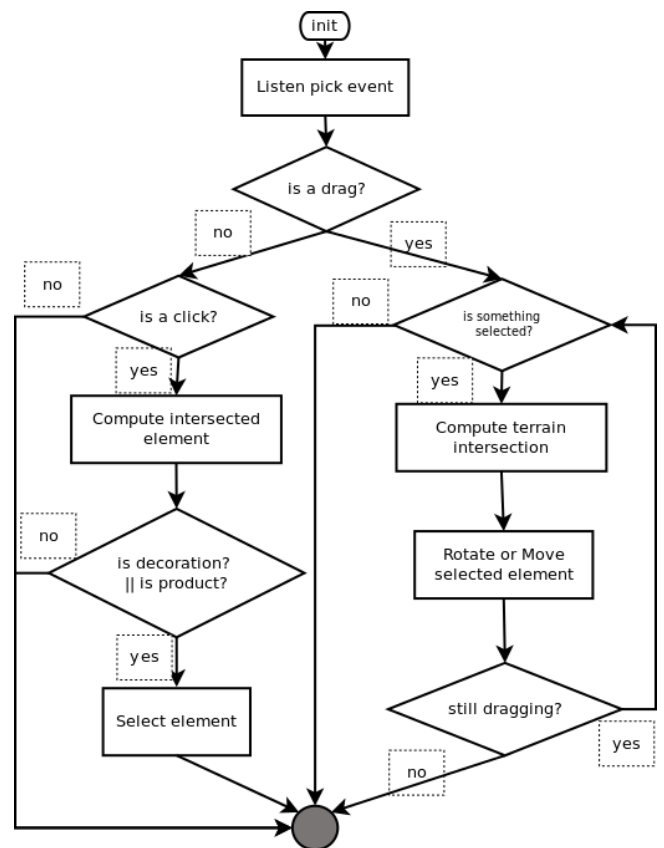
**Navigation based on arrow keys** This is the simplest option to navigate the virtual world and is intuitive for regular users. The keyboard arrow keys are the only way to walk around the shop. The user can press the *up* and *down* keys to move forward and backwards. To look left or right he can press the *left* or *right* arrow keys and to look up or down use the *page up* or *page down* keys.

**Navigation mixing keyboard and mouse** This method is common in first person shooter games. It uses the keyboard to move forward, backward and sideways and the mouse to look around.

**Navigation based on HUD panel with directional buttons**

This navigation model is designed for customers not used to video games or virtual worlds. Similar to typical graphical user interfaces a small panel is superimposed at the bottom of the 3D view containing a set of buttons. Each button generates motion events when pressed (step forward, step backwards, turn left, turn right, move view up and move view down).

Customers can also interact with products in the shop. When a user clicks on an object in the scene the interface displays associated information in a pop up panel such as a picture of it, price, available stock, description and other properties as defined by the administrator. This panel includes controls to add the selected product to the shopping cart in whatever quantity. An additional panel lists the current contents of the cart and allow making the actual purchase operation.



**FIGURE 4.** Interaction logic



## User behaviour

In [3], the authors mention acquiring customer models from recorded *clickstream* data from previous visits to the VR environment. They store all the mouse clicks performed by users on the scene to analyse their behaviour. We consider that in virtual and real shops not only clicks (i.e. interaction with objects) are relevant but also all movements of users around the space. A good distribution of space and products must make certain products easy to find, or must place products in places where most customers move or where they look at. Knowing about customer's movements and actions in the shop can help enhance the space and placement of goods to improve user experience and sales. Our concept platform includes functionality for this purpose.

**Recording user actions** The application continually tracks the movements of the customer as he moves around the virtual shop. This collected information is sent to the server for storage and later analysis. The variables tracked are position, orientation, product selection actions and purchase actions, always associated to a time stamp. Instead of periodically storing all variables, the system only considers changes in them, so that a user standing still does not flood the system with redundant data. When the client has collected a certain amount of samples they are sent to the server via Ajax requests where they will be stored and associated with the client and his session.

**Representation of logged user actions** Obviously only storing this data would be rather useless. The platform needs to provide a means to present this data to the administrator in a way that is comprehensible. While there are different possible ways to represent collections of variables, we selected two different kinds of visual representations and integrated them in the administration part of the VR shops Web interface.

As the system has collected a continuous stream of position and orientation samples with time stamps, it is able to fully reproduce a client's session as if it was recorded by a surveillance camera. We implemented an interactive VCR-like session player that replays a user's own view as he browsed the VR shop. In this case, the administrator has to select one session and one user in order to replay it and watch how he moved or what he looked at. The player includes fast-forward functionality so the user can skip to a desired point in time by dragging a slider bar control.

But presenting data for one single person in one specific visit to the shop is not statistically significant. More useful representations should take into account full sets of visits of different customers to see the overall behaviour and points of interest. This is achieved through visualizations made on a top view of the scene in two different forms.

First the system can draw a map of trajectories followed by users on the top view, either for individual users or for all of them together. Then, and more importantly, it can render what we call

a *heat map* of the virtual environment. By superimposing the trajectories of multiple users in multiple sessions, the system paints the floor image with a colour map resulting in an image that depicts in hotter (i.e. red) colours the areas where most people have wandered. The idea comes from the attention maps generated by eye trackers in marketing studies. As a base for these maps an orthographic top view of the environment is first rendered into an image.

## IMPLEMENTATION AND RESULTS

In contrast to higher end virtual reality systems, the concept described in this paper is highly constrained by the medium in which it has to be deployed, the standards-based Web, and thus implementation faced special problems. It cannot obviously use special VR devices such as HMDs, stereo displays or tracking sensors. Our implementation relies heavily on WebGL, the new standard for imperative real-time 3D graphics rendering on the Web. WebGL is an API exposed to JavaScript scripts and all interaction, scene creation and management and user interface behaviour has to be implemented in said scripting language.

### Implementation and deployment issues

The server side was implemented in Java and deployed on the GlassFish application server along with a MySQL database. The client side consists of some HTML, CSS and a lot of JavaScript, both specific of this application and third party libraries. External libraries account for 379 KiB while specific application code is spread over 42 files weighting 456 KiB. These JavaScript files, 835 KiB total, have to be transmitted to the clients in source form, which produce a relatively slow start-up. The situation is common in many modern rich HTML5 applications but is mitigated by today Internet connection speeds. Nevertheless, our tests have shown acceptably low delays even in moderately slow connections. In fact, the original source files can be optimized and compacted so that the number of HTTP requests is reduced and code size is also shrunken. We applied Google's Closure Compiler to the code with default settings and it reduced the code to less than half, and not all is needed in all profiles (the administrator profiles needs code that the visitor profile does not need).

One design decision was to base all 3D model files for products, decoration objects and pre-built scenes on the COLLADA format. It was chosen as it is a widely available standard with growing support in modelling and conversion tools. The idea is that retailers would find it easy to get 3D models of their products in a well known format to populate their virtual shop. COLLADA is an XML-based format and is known to produce larger files than most other formats which is a drawback for Internet applications. Also, we found that COLLADA support in current tools is still limited, sometimes flawed and not equally generated

or treated by different software. We found problems with models exported from Blender, for example.

WebGL support is currently not available in all major browsers. The browsers used for implementation and validation were *Mozilla Firefox* (>5.0) and *Google Chrome*. We found the behaviour not to be identical in both applications. Mouse events are managed differently in *Firefox* and *Google Chrome* and we had to make extensive adaptations to achieve a smooth interaction with objects. Both show good performance with our application but the latter feels smoother. The development and test platform is a common mid-range PC with 3D hardware acceleration (3GHz Intel Core2 Duo with Nvidia GeForce 9800). Quick tests were also performed on a Samsung Galaxy Tab tablet which showed too slow performance, achieving 1 to 3 fps. This suggests that Web 3D applications like this one need a specially adapted low complexity version to be usable on mobile devices.

At the moment, Microsoft *Internet Explorer* does not support WebGL natively but adding Google's *Chrome Frame* plug-in makes this browser compatible. A special HTML tag had to be added to make this work correctly. An alternative is the use of a plug-in called *IEWebGL* that adds support for WebGL to IE but this is not desirable as it defeats the purpose of being plugin-less. In this case the required changes in the code are larger and have not been tested yet. Anyway, the need for specific plug-ins in current versions of Internet Explorer somewhat hinders the intended ubiquity of our approach.

**Use cases** The system was tested in two scenarios, a fictional shop created with the parametric editor, and a real shop externally modelled and imported into the system. The first one was totally modelled with the design interface for administrators of our system, which is seen as very convenient, while the real shop was modelled by expert designers using Maya. The first model is just a few kilobytes in size and the second takes about 1.5 MB, not counting the decoration and product objects. File transfers show a noticeable delay of several seconds in slow connections but it is tolerable time and can still be optimized via standard compression.

Several people from our department, but unrelated to the project, were presented a demo of the platform prototype. While they see the value of being able to create the whole VR environment in the same Web platform, they admit that the visual quality achievable is relatively low. The real shop model imported is much more visually appealing than the one parametrically generated. Performance (i.e. frame rate and feeling of smoothness) is good but suffers slightly in the case of the detailed model due to the ray intersections computations needed for floor following and collision detection against a large number of meshes and polygons.

A few people from the e-commerce sector also tried the prototype. Upon testing the system, they found the concept in-

teresting but expressed concerns about the process of updating the catalogue with newly available products. Products need associated 3D models and currently some skilled designer should create them. This may enable a business model in which a studio is hired every time new objects must be displayed but that may be a problem for very small businesses. Simpler methods of 3D model creation show up as one of the major limitations to the success of these systems.

**Navigation and interaction** In the implementation of the navigation and the interaction models some problems were identified and addressed.

In keyboard and mouse navigation mode for the regular user profile, the left mouse button is used for two different actions: object selection and view rotation. In order to distinguish them the system needs to identify short button presses as opposed to long presses and drag. The solution was to add a timer. One mouse button press is interpreted as a drag action (i.e. look around) if it lasts longer than a certain time (0.5 seconds currently) but it is interpreted as a click action when press duration is shorter (which will select the object and request information about it from the server).

Another problem arises when a manager needs to move an object from a place in front of the camera to another out of the field of view. In this case it was necessary to combine the object movement with a camera rotation so that the destination point is brought into view.

One last problem is related to navigation performance. As explained above, the client navigation uses two intersecting rays for collision detection and floor following. These ray intersections are quite computationally expensive, depending on the scene complexity and may result in very slightly choppy movements. The administration profile, on the other hand, implements a fly-mode with no collision detection so navigation for this profile is significantly smoother.

**User actions representation** In this work a heat map image is presented as a representation of logged users' behaviour. To compose this image, data from all past visitors must be processed. Considering that the number of users can be high and each user can have more than one session in the same shop, it appears that a large and growing amount of data is required.

In the current approach, the creation of the map from all that data takes place in the browser which renders the result in an HTML5 canvas. This process is relatively expensive and Javascript spends a fair amount of time performing it (6-7 seconds with only 30 individual user sessions). An example of a resulting map can be seen in figure 5.

To reduce the computation time, it would be possible to generate these representations in the server, using faster or more optimized programming languages such as Java or C++.





**FIGURE 5.** Image of resulting heat map from the motion of a set of users

## CONCLUSION AND FUTURE WORK

We have presented a concept for the creation of virtual reality shops on the Internet. The concept integrates in a single online platform the creation and management of virtual shops as well as the virtual spaces that users would visit to browse and buy products in a way that mimics real urban shops. The concept thus makes Web-based virtual reality a user interface or front-end to an e-commerce infrastructure back-end. Actually for practical purposes our application makes the user interface a combination of 2D and 3D VR elements.

The implementation tries to mitigate some of the problems that hindered the adoption of 3D virtual reality e-shopping. First, it provides an integrated centralised approach that doesn't require users or merchants to install any specific applications or browser plug-ins. And second, it implements simple and intuitive interfaces to 3D world editing (targeted at editing VR shops, not at generic 3D modelling). The editing part is still probably too simple for real world usage but can be improved for better results.

This work presents the feasibility of creating and deploying 3D VR applications based on standards such as HTML5 and WebGL acting as front ends for e-commerce. In fact, other applications are possible in this way with a suitable back-end: the VR world would just need to call services exposed by the back-end upon user actions on the scene's elements through Ajax requests.

The limitations of these technologies compared to full fledged VR systems sets some limits to the amount of detail, visual quality or functionality but what can be achieved is acceptable for the target application. The whole environment and all 3D objects must be simplified to avoid lengthy transfer times and low rendering performance.

What our work does not solve yet is the long standing problem of making easier for merchants to update the catalogue of 3D objects as new products become available.

There is obviously room for improvement. It is not clear that COLLADA, however popular for 3D asset exchange, is a good choice for Internet applications. Its support in modelling tools is not universal and files tend to be too large for effective transfer of complex models. WebGL is still quite new and has not been fully exploited in this implementation. The standard leaves room for better materials and lighting effects that might be implemented in future versions of our platform.

Future additions may include testing different 3D asset formats, more compact and efficient for use on the Internet, improvements in the integrated 3D design functionality to generate better quality environments without external tools, and improvements in rendering quality with better illumination (for example by implementing precomputed light maps). Another interesting advance would be the support of mobile devices such as tablets and the exploitation of their motion sensors as an interaction mode. Some tests were already made which worked but resulted in extremely slow performance of the 3D rendering. In parallel we will seek ideas for tools to assist in the creation of new models for objects for sale, via 3D scanning, reconstruction from multiple views, or interactive methods.

## REFERENCES

- [1] Egger, F. N., 2001. "Affective design of e-commerce user interfaces: How to maximise perceived trustworthiness". *Proceedings of The International Conference on Affective Human Factors Design Asean Academic Press, London.*
- [2] Burke, R., 1998. *Sense and Respond: Capturing Value in the Network Era*. Harvard Business Press, New York, ch. 11. Real Shopping in a Virtual Store.
- [3] Chittaro, L., and Ranon, R., 2002. "New directions for the design of virtual reality interfaces to e-commerce sites". *5th International Conference on Advanced Visual Interfaces, ACM Press, New York*, p. 5.
- [4] Catone, J., 2007. "Kinset: the second life for shopping". Retrieved on 18/10/2011 from [http://www.readwriteweb.com/archives/kinset\\_like\\_second\\_life\\_for\\_shopping.php](http://www.readwriteweb.com/archives/kinset_like_second_life_for_shopping.php).
- [5] Aoki, Y., 2002. Building a collaborative web environment for supporting end users. Tech. rep., IBM Research, Tokyo Research Laboratory.

- [6] “Virtuymall”. Retrieved on 9/10/2011 from <http://www.virtuy.com/>.
- [7] Bowman, D. A., Chen, J., Wingrave, C. A., Lucas, J., Ray, A., Polys, N. F., Li, Q., Haciahmetoglu, Y., Kim, J.-S., Kim, S., Boehringer, R., and Ni, T., 2006. “New directions in 3d user interfaces”. *The International Journal of Virtual Reality*.
- [8] Bowman, D. A., Kruijff, E., LaViola, J. J., and Poupyrev, I., 2005. *3D User Interfaces: Theory and Practice*. Addison-Wesley.
- [9] Autodesk, 2011. “Homestyler”. Retrieved on 15/04/2011 from <http://www.homestyler.com>.
- [10] Marrin, C. *WebGL Specification*. Khronos WebGL Working Group.
- [11] Garrett, J. J., 2005. Ajax: A new approach to web applications. <http://adaptivepath.com/ideas/essays/archives/000385.php>, February. [Online; Stand 18.03.2008].

#### **ACKNOWLEDGMENT**

The work here described was part of the HIRIDENDA 2020 project, partly funded by the Basque Government’s ETORGAI programme.