DOI: 10.1111/cgf.12466

COMPUTER GRAPHICS *forum*
Volume 00 (2014), number 0 pp. 1–39

Advances in Interaction with 3D Environments

J. Jankowski and M. Hachet

Inria Bordeaux Sud-Ouest, Talence, France
{jacek.jankowsk, martin.hachet}@inria.fr

Abstract
Various interaction techniques have been developed for interactive 3D environments. This paper presents an up-to-date and comprehensive review of the state of the art of non-immersive interaction techniques for Navigation, Selection & Manipulation, and System Control, including a basic introduction to the topic, the challenges and an examination of a number of popular approaches. We also introduce 3D Interaction Testbed (3DIT) to firstly allow a 'hands-on' understanding of 3D interaction principles, and secondly to create an open platform for defining evaluation methods, stimuli as well as representative tasks akin to those found in other disciplines of science. We hope that this survey can aid both researchers and developers of interactive 3D applications in having a clearer overview of the topic and in particular can be useful for practitioners and researchers that are new to the field of interactive 3D graphics.

Keywords: 3D interaction, navigation, selection, manipulation, system control

ACM CCS: [Computer Graphics]: I.3.6 Methodology and Techniques Interaction Techniques; Computer Graphics I.3.7 Three-Dimensional Graphics and Realism Virtual reality

This paper is an extended version of the State-of-the-Art report (STAR): 'A survey of interaction techniques for interactive 3D environments' [JH13] presented at Eurographics'13.

1. Introduction

People spend all their lives in a 3D world and they develop skills for manipulating 3D objects, navigating around the 3D world and interpreting 3D spatial relationships. However, they often find it difficult to interact in interactive 3D environments. Quite often, this is the result of improperly designed user interface (UI) or due to inappropriate selection of interaction techniques. Therefore, in this paper we present a comprehensive review of the state of the art of mouse- and touch-based interaction techniques for interactive 3D environments. We hope that this survey can aid both researchers and developers of interactive 3D applications in having a clearer overview of the topic and in particular can be useful for practitioners and researchers that are new to the field of interactive 3D graphics.

We define interactive 3D environments as computer representations of real world or imaginary spaces through which users can navigate and in which they can interact with objects in real time. Unlike immersive virtual environments [BKLP04], which use specialized displays and interaction devices such as head-mounted displays to create a sense of presence for the users within the virtual world, interactive 3D environments are not limited to immersive context and exploit common, general-purpose hardware for interaction, that is, mouse and keyboard as well as multi-touch input.

The interaction in an interactive 3D environment can be characterized in terms of three universal interaction tasks [Han97, BKLP01, BKLP04]: Navigation—refers to the process of getting around a virtual environment: includes a motor component called travel, and a cognitive one, known as wayfinding; *Selection and Manipulation*—refers to techniques of choosing an object and specifying its position, orientation and scale; *System Control*—refers to communication between user and system which is not part of the virtual environment. The problem of providing a user with an interface for exploring 3D environments is strongly related to all the above listed problems.

The research devoted to the interaction techniques for 3D environments is broad, but it still lacks of a formal structural organization. In light of this consideration, the main contributions of this STAR can be summarized as follows:

- In this state-of-the-art report we cover the important past 16 years of work on interaction techniques for interactive 3D since the last STAR [Han97] on the subject.

- More than 200 interaction techniques for *Navigation*, *Selection&Manipulation*, and *System Control* are thoroughly analysed, formally organized and reviewed in Sections 2–3.2.3.
- We use a task-centric classification of the techniques in this field, in order to help users, in particular practitioners and researchers that are new to the field of 3D interactive graphics, to choose the ones that best suit their needs.
- As we also need practical tools to carry out the research on 3D interaction, in Section 4.2 we introduce 3D Interaction Testbed (3DIT), an open source initiative that aims to allow a '*hands-on*' understanding of 3D interaction principles, and to create an open platform for defining evaluation methods, stimuli as well as representative tasks akin to those found in other disciplines of science.
- As implementation of an interaction technique is usually followed (or at least should be) by its evaluation, in Section 5.3 we describe methodologies that may help in the process of evaluating such techniques.
- As most of our STAR focuses on the specific tasks that occur in 3D applications, which are independent of the input device, in Section 6.4 we describe some Virtual Reality (VR) and 3D User Interface (3DUI) research that can have an impact on interaction techniques for non-immersive 3D environments.
- Finally, based on our experience with developing interaction techniques for the Web-based 3D environments, we summarize what we consider to be the main 3D Web design guidelines in Section 7.5.

1.1. Related work

A great deal of research interest has been devoted to the 3D computer graphics; however, most of this research effort has focused on the technical and application aspects. Our concern, however, is with how users interact with 3D environments and there are only few works related to classification approaches in the field of 3D interaction techniques.

Chris Hand's survey was published 16 years ago [Han97] and is significantly outdated. Much more recent comprehensive overview of interaction techniques is given by Bowman *et al.* [BKLP01, BKLP04]. These works mix immersive and non-immersive techniques. Christie *et al.* [CON08] presented an overview of camera control in computer graphics, focusing mainly on semi- and fully automatic camera systems. Their analysis draws on insights from cinematic and photographic practice. In contrast, our work will be guided by the interaction tasks. Moreover, we provide a review of all three universal 3D interaction tasks. Finally, in [Jan11b] we aimed to clarify some of the foundations of 3D Web UI design. We focused on an understanding of the fundamental tasks users may engage in while exploring Web-based 3D virtual environments and included a review of mouse-based 3D interaction techniques useful in the context of 3D Web. This review formed the foundation for this state-of-the-art report.

2. Navigation

Interactive 3D environments usually represent more space than can be viewed from a single point. Users have to be able to get around within the environment in order to obtain different views of the scene. This process of getting around a virtual environment while keeping track of one's whereabouts and task objectives is the subject of a number of techniques often referred to by such names as wayfinding, locomotion, navigation and camera/viewpoint control.

Developing an effective technique for navigation for interactive 3D environments is difficult. First of all, viewpoint control involves six degrees-of-freedom (DOFs): three dimensions for positional placement (translation) and three for angular placement (rotation). The problem is the number of parameters to be controlled by the user—the major limitation inherent in using 2D devices is that at least one change of state is required to cover all directions of translation and rotation. Another problem is the type of viewpoint control required by a given task that can be as different as simply exploring large-scale 3D environment to high-precision inspection of some 3D object. In addition to the difficulties of controlling the viewpoint, there is a problem of wayfinding, especially in large virtual worlds. It is related to how people build up an understanding (mental model) of a virtual environment and it is significantly affected by technological constraints among which are small field of view and the lack of vestibular information. At last, effective techniques for 3D navigation should satisfy general interface requirements (e.g. is easy to use, prevents user disorientation)

According to Bowman *et al.* [BKLP01] and Tan *et al.* [TRC01], navigation may be broken into three subtasks: *exploration*, navigation with no explicit target, simply to investigate the environment; *search*, moving to a particular target location and *inspection* [TRC01] or maneuvering [BKLP01], to establish and maintain a particular view of an object. Another classification of navigation techniques was proposed by Mackinlay *et al.* [MCR90], who distinguish four types of viewpoint movement for interactive 3D workspaces:

General movement. Exploratory movement, such as walking through a simulation of an architectural design

Targeted movement. Movement with respect to a specific target, for example, to examine a detail of an engineering model

Specified coordinate movement. Movement to a precise position and orientation, such as to a specific viewing position relative to a CAD model: the user has to supply the exact position and orientation of his destination.

Specified trajectory movement. Movement along a position and orientation trajectory, such as a cinematographic camera movement

In recent years, many types of viewpoint movement techniques for interactive 3D environments have been implemented or proposed. In the following we classify them into the four categories proposed by Mackinlay *et al.* [MCR90].

2.1. General movement

We classify general movement techniques into the following groups:

2.1.1. *Rotate-pan-dolly*

Rotating, panning and dollying are the primary camera movements used in almost every 3D modelling environment (from *Jack* [PB88]

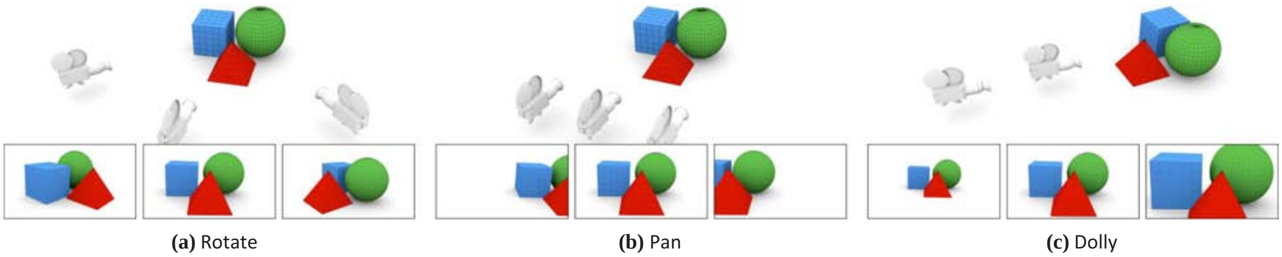

Figure 1: *Rotating, panning and dollying are the primary camera movements used in almost every 3D modelling environment.*

to Autodesk's 3ds Max, Maya or Blender). They are standard ways to inspect objects, and work well with a pointing device such as a mouse since all of them are at most 2D operations.

- *Rotate* (also referred to as *Tumble* or *Sweep*)—refers to orbiting the camera around a central point in any direction—the sweep operation sweeps the camera around horizontally and vertically on a virtual spherical track, keeping it focused at the same reference point (see Figure 1(a)).
- *Pan*—in the context of 3D interaction, *Pan* refers to translation of the camera along *x* and *y* axes (see Figure 1(b));
- *Dolly*—refers to translation of the camera along its line of sight (see Figure 1(c)). In many applications the term 'Zoom' (narrowing the field of view of the camera) is mistakenly applied to this operation.

For example, to navigate in the viewport in Blender, the user needs to drag the mouse while holding the Middle Mouse Button (MMB) pressed to rotate, additionally pressing the Shift button on the keyboard to pan (Shift MMB), and holding the Ctrl button to dolly (Ctrl MMB). It is worth to mention that some applications (including e.g. VRML/X3D viewers) additionally implement *Look Around* technique that changes the orientation of the camera but keeps it at a fixed position.

Current 3D rotation interaction techniques are generally based on Chen *et al.*'s work on Virtual Sphere [CMS88] and Shoemake's ArcBall [Sho92], techniques designed for 3D navigation around 3D objects. Both techniques are based on a concept of a virtual ball that contains the object to manipulate. They utilize the projection of the mouse location onto a sphere to calculate rotation axis and angle. Comparison of mouse-based interaction techniques for 3D rotation can be found in [HTP*97, BRP05].

Rotate-pan-dolly technique requires the user to accomplish a movement by shifting back and forth among simple navigation modes (assigning the mouse to '*Rotate*', '*Pan*' or '*Dolly*' operations) [PB88]. Such an interaction model can be not optimal if the menu has to be used frequently. To solve this problem, Zeleznik and Forsberg [ZF99] proposed gestural interaction for invoking camera functionality. Their approach, called UniCam, requires only a single-button mouse to directly invoke specific camera operations within a single 3D view; remaining mouse buttons can be used for other application functionality.

Zeleznik *et al.* [ZFS97] explored a range of interaction techniques that use two hands to control two independent cursors to perform operations in 3D desktop applications. The authors presented both how to navigate (Rotate-Pan-Dolly and flying techniques) and manipulate (Rotate-Scale-Translate, RST) 3D objects using two pointer input. Balakrishnan and Kurtenbach [BK99] also investigated bimanual camera control; they explored the use of the non-dominant hand to control a virtual camera while the dominant hand performs other tasks in a virtual 3D scene.

2.1.2. *Screen-space methods*

Gleicher and Witkin [GW92] describe a body of techniques for controlling the movement of a camera based on the screen-space projection of an object, where the user indicates the desired position of the object on the screen. In other words, the presented *through-the-lens* techniques permit the user to control the virtual camera by directly manipulating the image as seen through the lens.

Inspired by Gleicher and Witkin's work [GW92] and 3D navigation with multiple inputs [ZFS97, BK99], Reisman *et al.* [RDH09] describe a screen-space method for multi-touch 3D interaction. Just like 2D multi-touch interfaces allow users to directly manipulate 2D contexts with two or more points, their method allows the user to directly manipulate 3D objects with three or more points (see Figure 2). The idea is that each contact point defines a constraint which ensures the screen-space projection of the object-space point

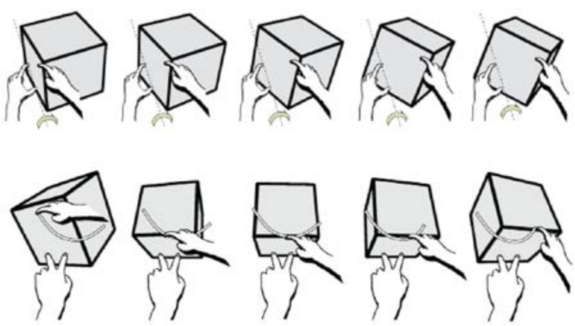

Figure 2: *Two three-finger rotation techniques proposed by Reisman et al. [RDH09]. Top row: defining an axis by connecting the two contacts of the non-dominant hand, and then rotating the object about that axis. Bottom row: interaction not limited by the axis defined by the non-dominant hand. Image courtesy of Jason Reisman.*

'*touched*' always remains underneath the user's fingertip. Walther-Franks *et al.* [WFHM11] addressed the same problem and designed and implemented multi-finger mappings for unimanual manipulation of view, objects and time; they also showed how the technique can be extended for efficient bimanual control [ZFS97, BK99].

Singh *et al.* [SGS04] proposed a screen-space camera widget called IBar for navigation and inspection tasks. It was designed to provide an interface for controlling scene perspective based on the artistic concept of vanishing points. Compared to Rotate-Pan-Dolly, most of the participants of the authors' evaluation found the IBar more intuitive.

2.1.3. *Walking/driving/flying*

The other way of allowing users to navigate in a 3D environment is simply to let them walk along the ground, drive a vehicle or fly a plane [BKH97, WO90]. This technique requires the user to continuously modify the camera position using, for example, arrow keys or controls (button widgets) superimposed on the screen and controlled by the mouse.

In the video game world this technique corresponds to first-person perspective (FPP) game navigation, where a graphical perspective is rendered from the viewpoint of the player character (this may be also the viewpoint from the cockpit of a vehicle). Many different genres have made use of FPP navigation, ranging from adventure games to flight simulators. Perhaps the most notable genre to make use of this technique is the first-person shooter (FPS). Wolfenstein 3D and Doom are video games that are generally regarded as having popularized this genre and helped to shape the overall direction of the gaming industry.

Lécuyer *et al.* [LBHD06] studied the walking metaphor and suggested that oscillating camera motion together with compensation motion which changes the orientation of the camera and simulates oculomotor compensation to keep a constant focal point can improve the sensation of walking in Virtual Environments (VEs). Furthermore, the results of Terziman *et al.* [TLHW09] suggest that oscillating camera motions allow a more accurate distance perception for short travelled distances in VEs.

As the multi-touch technology is becoming more and more available, some research has been made on how to use this new input device also for the '*Walking*' metaphor. Kim *et al.* [KGMQ08] introduce a Finger Walking in Place (FWIP), a scaled-down walking-in-place [SUS95, TDS99] locomotion technique, that allows a user to travel in a virtual world as her/his bare fingers (representing human legs) slide on a multi-touch sensitive surface. The user can move forward and backward, and rotate in a virtual world. While the authors apply the technique to immersive settings, we believe that it is also applicable to non-immersive applications.

2.1.4. *Exploration of joint 2D and 3D data*

There is a specific, increasingly common class of 3D environments produced via computer vision algorithms applied to huge collections of pictures. The process, known as image-based rendering (IBR) [SSS06], allows the creation of interactive, photo-realistic, 3D visualizations of real objects and environments from databases of captured images. Photo Tourism [SSS06], one of the systems representing this approach, enables to browse a specific collection of photographs in a 3D spatial context that gives a sense of the geometry of the underlying scene. It supports full 3D navigation of the set of interlinked images and world geometry as well as indirect navigation based on a complementary thumbnail bar. This work was further extended to automatically compute orbits, panoramas, canonical views and optimal paths between views [SGSS08]. Brivio *et al.* [BBT*13] offers additional interfacing opportunities.

2.2. Targeted movement

A targeted movement, that is, a movement with respect to a specific target, can be classified into the following groups:

2.2.1. *Point of interest movement (Go-To/Fly-To)*

Mackinlay *et al.* [MCR90] describe Point of Interest (POI) Logarithmic Flight, a technique that supports rapid controlled movement through a virtual 3D workspace. POI movement requires the user to simply choose a target '*point of interest*' on an object in the scene by using the mouse, and the virtual camera moves logarithmically towards this target (the camera jumps to the goal position that the user had specified). The viewpoint is automatically oriented during the flight to face the surface being approached by using the surface normal at the POI. Due of the logarithmic nature of this navigation technique, rapid motion is achieved when the distance to the target object is large and controlled movement is achieved as this distance gets shorter. UniCam [ZFS97] implemented a technique called *click-to-focus* that aimed at automatically choosing the endpoint of the camera trajectory according to the proximity of the edges of some object.

In the Depth-modulated Flying (DMF) interface, Ware and Fleet [WF97] combine a flying interface with the idea of making velocity proportional to distance. Their idea is to sample the Z buffer in order to obtain information about the depth distribution in the environment and modulate flying speed accordingly so that it is always comfortable when operating over a range of scales. Their experiment shows that scaling the velocity control by the near point in the scene and by the average point in the scene are equally effective. Drag'n Go [MMG12], another technique that works well for multi-scale VEs, allows the user keep full control of its position relative to the target as well as its travelling speed.

Tan *et al.* [TRC01] describe Speed-coupled Flying with Orbiting—technique that allows users to not only transition between local environment—views and global overviews of the world, but to also orbit around an object of interest in order to examine it. To separate the intention to freely navigate from that of constrained object inspection the technique determines where the user first starts dragging the mouse. If the user drags on free space (i.e. the sky or the ground), the system assumes that the user is trying to freely navigate. If, however, the user starts dragging the mouse on an object, the system assumes that the user is trying to examine it (changing the cursor indicates changes of modes). The technique couples speed

Figure 3: *Navidget can be used on various systems, from small hand-held devices to large interactive displays [HDKG08].*

control to height (position) and tilt (viewpoint) control to give the user the ability to transition seamlessly between and navigate within local as well as global views of the world. The authors suggest that this allows the user to acquire landmark, procedural and survey knowledge and to effectively perform exploration and search tasks. They also showed that this technique was generally superior in performance and preferred by users in comparison to several other techniques.

We have also proposed an extension to the aforementioned work. Firstly, we proposed the *z-goto* technique for mobile devices, where the endpoint is directly selected in depth by means of simple keystrokes [HDG06]. Secondly, partially influenced by UniCam's *region zooming* [ZFS97], we developed a 3D widget called *Navidget* [HDKG08, HDKG09], where the endpoint of a trajectory is selected for smooth camera motions. Compared to the existing POI movement techniques, it does not automatically estimate how the viewpoint is oriented. Instead, it provides visual feedback for fast and easy interactive camera positioning: a 3D widget coupled with a preview window is used in order to let the user control the viewing direction at destination. As our technique is based on 2D inputs, it is appropriate for a wide variety of visualization systems, from small hand-held devices to large interactive displays (see Figure 3). A user study shows that the usability of Navidget is more than satisfactory for both expert and novice users.

2.2.2. *Path drawing*

Igarashi *et al.* [IKMT98] propose a Path Drawing technique for walkthrough in 3D environments, which is an extension of the POI technique. The technique uses user-drawn strokes to control the virtual camera. A stroke is projected onto the walking surface and used as a path for the camera. The path can be updated at any time by drawing a new stroke. The evaluation showed that the technique was slightly slower on average than walking/driving and POI techniques. However, the drawing technique was preferred by most users. The authors also point out that the technique is most suitable for pen-based or touch panel systems. Hagedorn and Döllner [HD08] present another navigation method based on sketching navigation commands (see Figure 4). Their system interprets the sketches according to their geometry, spatial context and temporal context. The authors state that unlike other sketchy navigation techniques, their approach identifies the hit objects of the underlying 3D scene and takes advantage of their semantics and inherent navigation affordances.

Figure 4: *Example of a sketch-based navigation [HD08]: the user draws a curve on the street and combines this command with a circle like gesture; the derived animation will move the camera along the sketched path and rotate for inspecting the target area. Image courtesy of Jürgen Döllner.*

2.2.3. *Hyperlinks and bookmarked views*

Hyperlinks in VEs allow fast (or even instantaneous) direct movement between places that are far apart and therefore can greatly reduce navigation time and allow for the design of flexible layouts compared to conventional techniques. These are the same advantages that hypertext has over conventional texts. Accordingly, the disadvantage of such hyperlinks is that they are likely to cause some cognitive difficulties such as disorientation. Some effects of hyperlinks on navigation in virtual environments were studied by Ruddle *et al.* [RHPJ00]. In order to study linking behaviour in a current 3D environment, Eno *et al.* [EGT10] examined explicit landmark links as well as implicit avatar pick links in Second Life. The authors found that although the virtual world link graph is more sparse than the flat Web, the underlying structure is quite similar. Moreover, they point out that linking is valued by users, and making linking easier would likely result in a richer user experience.

We have also studied the use of hyperlinks for navigation. In our Dual-Mode User Interface (DMUI) [JD12] designed for accessing integrated information spaces, where hypertext and 3D graphics data are simultaneously available and linked, embedded in text hyperlinks may constitute not only a mechanism for navigation between hypertext documents, but also for navigation within 3D scenes.

Some 3D viewing applications (e.g. VRML/X3D browsers) provide a viewpoint menu offering a choice of viewpoints, usually denoted by a brief textual description that helps make clear the intended purpose of each view. Authors of 3D scenes can place several viewpoints (typically for each POI) in order to allow easy navigation for users, who can then easily navigate from viewpoint to viewpoint just by selecting a menu item. The process is analogous to the creation of a set of custom hyperlinks to specific HTML document positions.

The viewpoint menu plays an important role in 3D interaction; it turned out to be an important navigation tool in our recent study on accessing 3D content on the Web [JD12]. Such viewpoints are usually static—a viewpoint is simply a specific camera position and orientation defined by a pair of *x*, *y*, *z* coordinates, that is, a specific view of a 3D scene. While very useful, we believe that static viewpoints often do not show '*3D-ness*' of virtual objects—as Andy van Dam mentioned: '*if it ain't moving, it ain't 3D*'. Therefore, inspired by [Tod04, BKFK06], we performed an evaluation of static versus animated views in 3D WebUIs [Jan12]. We found out that all users clearly preferred navigating in 3D using a menu with animated viewpoints than with static ones (there was not even a single user that disabled animated views during the study).

An interesting way to assist users in navigation was developed by Elvins *et al.* [ENK97, ENSK98]. They introduced a new technique that captures a 3D representation of a virtual environment landmark into a 3D thumbnail, which they call a worldlet. The worldlets are miniature virtual world subparts that may be interactively viewed (one could manipulate them to obtain a variety of views) to enable users getting familiar with a travel destination. In the evaluation conducted by the authors to compare textual, image and worldlet landmark representations within a wayfinding task, subjects who had been shown the worldlets performed significantly better than subjects who had been given pictures of landmarks or verbal instructions.

2.2.4. *Navigation by query*

Van Ballegooij and Eliens [vBE01] propose navigation by query, an interesting navigation technique based on information retrieval. This technique augments interface of virtual environment by allowing users to navigate a virtual world by means of querying its content. The authors' experiments indicate that this type of navigation may help users to find locations and objects that would otherwise be hard to find without prior knowledge of the world. However, the authors make an assumption that the world is sufficiently annotated. Using the same assumption, McNeill *et al.* [MSWM02] report on work on exploiting speech input and natural language processing (NLP) technology to support both general and targeted navigation in virtual environments. According to the authors, spoken dialogue interaction is an effective alternative to mouse and keyboard interaction for many tasks and conclude that multi-modal interaction, combining technologies such as NLP with mouse and keyboard may offer the most effective interaction with Virtual Environments.

2.3. Specified coordinate movement

Specified coordinate movement is a movement to a precise position and orientation, such as to a specific viewing position relative to a car model—the user has to supply the exact position and orientation of his destination. This type of camera positioning is used in CAD and 3D editing software, where the users can simply enter three coordinates for *x*, *y*, *z* for both position and orientation using a keyboard (the ordering of the coordinate axes is typically pre-defined for a given scene). Moreover, parameter specification does not necessarily have to be done via a keyboard. For example, it can be performed through the use of a slider widget, which may contain, for example, either notches for additional precision, or a numeric label to show the value of the parameter as it is changed. Finally, popular 3D modelling environments (like 3ds Max or Maya) and game engine–integrated development environments (like Unity3D) offer tools for translation and rotation (such as the manipulator shown in Figure 18), which allow for specific coordinate movements, since they too can be used for the manipulation of virtual cameras from an alternate view. It is important to stress that this technique can be used efficiently only by the designers who are familiar with the 3D model/environment being under development.

Programmers using game/graphics engines often control a virtual camera directly from code. Such engines typically provide various formalisms to express a transformation in three dimensions. For example, for rotation, formalism alternatives include rotation matrix, axis-angle representation, Euler rotations or Quaternions [Sho85]. For much broader discussion on this topic, please refer to a classic, Computer Graphics: Principles and Practice [FvDFH90] and Rick Parent's Computer Animation: Algorithms and Techniques [Par12].

2.4. Specified trajectory movement

Specified trajectory movement is a movement along a position and orientation trajectory, such as a cinematographic camera movement. Compared to previous techniques, where the users are free to roam and explore, the techniques of this category empower the author to bring structure to the experience. Such viewpoint control limits the user's freedom while travelling through a virtual world. It constrains the audience's movement to (among other things):

- present relevant and interesting locations or objects;
- provide the best overview of the scene;
- create paths that are easy to learn and avoid the disorientation of the user;
- avoids the problem of users getting '*lost-in-cyberspace*'.

2.4.1. *Guided/constrained navigation*

Guided navigation was introduced by Galyean [Gal95], who proposed a new method for navigating virtual environments called '*The River Analogy*'. He envisioned the navigation paths as the rivers flowing through a landscape, and the user as a metaphorical boat floating down the river with some control while also being pushed and pulled by the pre-defined current of the water. Since the boat flows continuously down the river regardless of whether or not there is input, the river metaphor guarantees an uninterrupted flow. Elmqvist *et al.* [ETT08] have presented a similar method that uses motion constraints to guide the user on a pre-defined tour through the environment while still allowing users to control their speed as well as to deviate locally from the tour. The authors also include an

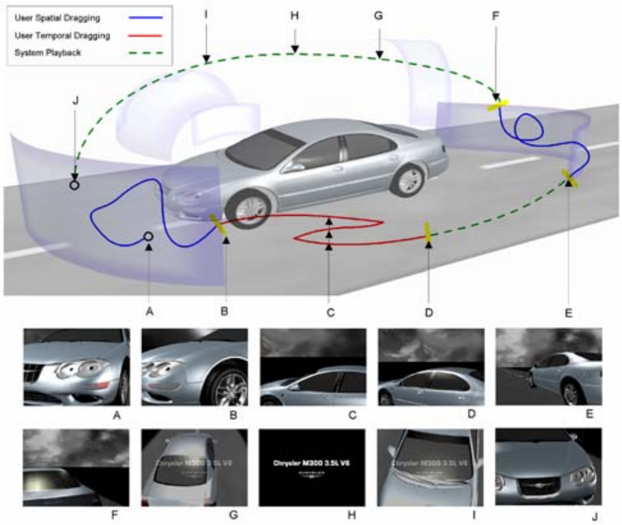

Figure 5: *Example StyleCam experience. Top: system components and their reaction to user input. Bottom: what the user sees [BKF*02]. Image courtesy of Nicolas Burtnyk.*

empirical evaluation of the effectiveness of such a guidance method for real user tasks.

Hanson and Wernert [HW97] propose a navigation technique that, rather than controlling an unconstrained vehicle in 3D space, uses the 2D control device to move the user on a constrained subspace, the '*guide manifold*', a kind of virtual 2D sidewalk. '*Virtual sidewalks*' can be authored by constructing virtual constraint surfaces and specifying gaze direction, vistas and procedural events (e.g. fog and spotlights) along the sidewalk.

Burtnyk *et al.* [BKF*02] describe StyleCam, a system that builds upon the aforementioned navigation ideas, but differs by providing authoring elements that enable a more stylized experience. It is an approach for authoring 3D viewing experiences that seamlessly integrates spatial camera control with the temporal control of animation playback. StyleCam allows the author to significantly tailor what and when the users see. Its aim is to create a viewing experience that can approach the visual richness of sophisticated visual content such as television commercials or feature films and at the same time, allow for a satisfying level of interactivity. StyleCam uses camera surfaces which spatially constrain the viewing camera, animation clips that allow for visually appealing transitions between different camera surfaces and a simple interaction technique that permits the user to seamlessly and continuously move between spatial-control of the camera and temporal-control of the animated transitions (see Figure 5). Burtnyk *et al.* also describe ShowMotion [BKFK06], an interactive system for 3D design review of CAD models. Their system replaces traditional '*bookmarked*' static views with moving '*shots*' interspersed with cinematic visual transitions. As we have already mentioned in Section 2.2.3, we performed an evaluation of static versus animated views in 3D Web UIs [Jan12].

During inspection of an object, the navigation above its surface can become complex. Khan *et al.* [KKS*05] describe HoverCam, a technique for navigating around 3D objects at close proximity. It allows users to move around the object while maintaining a fixed distance from the surface and while keeping the object centred in the field of view. HoverCam integrates tumbling, panning and zooming camera controls so the users can focus on the task at hand instead of continuously managing the camera position and orientation. McCrae *et al.*'s *cubemap* [MMGK09] extends the work on HoverCam with support for multi-scale navigation. The authors present an additional interaction technique for navigation '*look-and-fly*'. They also claim that their approach to the navigation of multi-scale 3D environments offers greater flexibility and ease of use than mainstream applications such as Google Earth and Microsoft Virtual Earth.

2.4.2. *Cinematographic techniques for camera control*

Many problems with camera control have been faced by cinematographers for over a century and over the years filmmakers have developed a good understanding of how to communicate comprehensibly and effectively. Therefore, some researchers exploited established cinematographic techniques for virtual camera control. For example, Christianson *et al.* [CAH*96] describe several of the useful principles of cinematography and show how they can be formalized into a declarative language, called the Declarative Camera Control Language (dccl), for specifying camera shots in terms of the desired positions and movements of actors across the screen. During their work on the virtual cinematographer, He *et al.* [HCS96] introduce the concept of the idiom, a sub-unit of cinematographic expertise, as a means of capturing the essence of a scene; the authors focus on filming dialogues between multiple animated characters. Tomlinson *et al.* [TBN00] created another automatic cinematography system for interactive virtual environments. Their system controls a virtual camera and lights in a 3D virtual world inhabited by both autonomous and user-controlled characters. When deciding where to put the camera, similarly to cinematographers, the authors consider the movements, relationships and emotions of the characters; the arrangement of the set; the ambient light and opportunities for adding or subtracting light. Li and Cheng [LC08] developed a real-time third-person camera control module for navigation in virtual environments, where the automatically generated tracking motion of the camera avoids occlusions from obstacles and follows cinematographic idioms (see Figure 6 for the views acquired from the authors' engine).

For more information about cinematographic techniques for camera control, please refer to the Christie *et al.*'s survey [CON08] that focuses mainly on semi- and fully automatic camera systems and that draws on insights from cinematic and photographic practice.

2.4.3. *Automatic viewpoint/path/motion planning*

Automatic viewpoint selection and path as well as motion planning methods can be useful as a way to augment CAD interaction, automatically create visits of virtual environments representing, for example, historical monuments or create engaging presentations of architectural visualization. They can also be useful in the context of interactive systems to create a first constrained path to help the user navigate a 3D environment.

Figure 6: *Snapshots of the views acquired from Li's and Cheng's third-person camera control module [LC08] in a Stonehenge-like ring. Image courtesy of Tsai-Yen Li.*

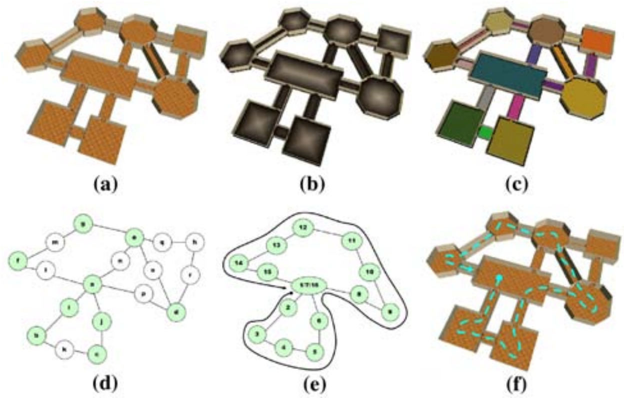

Figure 7: *Overview of Way-finder [AVG04]: (a) input scene; (b) distance-to-geometry field computed over the 3D grid; (c) cells detected with random color; (d) cell-and-portal graph embedded in the model space; cells are labelled according to relevance measure; (e) high-level path computed as a sequence of cells; visited cells is a superset of relevant ones; (f) final path after smoothing. Image courtesy of Carlos Andujar.*

To our knowledge, Phillips *et al.* [PBG92] were first to describe a semi-automatic navigation technique; it was designed for augmenting the process of 3D manipulation by automatically finding an effective placement for the virtual camera. Their human figure modelling system called *Jack* provided automatic features for maintaining smooth visual transitions and avoiding viewing obstructions (based on the hemicube approach).

Building upon the CINEMA testbed system [DGZ92] that provides users with the ability to experiment with various camera movement paradigms, Drucker and Zeltzer focused on an intelligent camera control in a 3D virtual environment [DZ94]. They present a methodology for designing the underlying camera controls based on an analysis of what tasks are to be required in a specific environment. They presented the first approach for automatic navigation in 3D environment using path planning: the plan is computed using a well-known graph searching technique called A*. In their further work, Drucker *et al.* introduced a method of encapsulating camera tasks into well-defined units called '*camera modules*' [DZ95]. The authors argue that through this encapsulation along with constrained optimization techniques, they can aid a user in a wide range of interaction with 3D graphical environments and greatly ease the development of 3D interfaces.

Li *et al.* [LLCY99] present a technique that allows a user to specify locations of interests on a 2D-layout map and let the system automatically generate the animation of guided tours in virtual architectural environments; the technique is based on several path planning algorithms adapted from robotics [Lat91]. Marchand and Courty [MC02] chose to use techniques widely considered in the robotic vision community. The basic tool that they considered is visual servoing [HHC96], which consists of positioning a camera according to the information perceived in the image.

Bares *et al.* [BGL98, BL99, BTM00] studied constraint-based approach to camera planning in virtual environments that is built upon the assumption that camera shots are composed to communicate a specified visual message expressed in the form of constraints on how subjects appear in the frame. The authors present a prototype, where one can specify constraints in a text script which can then be processed by the constraint solver that attempts to find a solution camera shot. Halper *et al.* [HHS01] emphasize guiding camera control by constraints. However, the authors stress that we should also pay particular attention to a trade-off between constraint satisfaction and frame coherence and present an algorithm for dynamic consideration of the visibility of objects that works for arbitrary dynamic scenes and spatial complexities of environments.

Some authors focused on automatic camera control techniques based on information theory. Most notably, Vázquez *et al.* [VFSH01] proposed viewpoint entropy to express the amount of information in a selected view. The authors define their metric as the ratio of the projected area of each surface to the total area of all the surfaces projected to the view sphere. Turkay *et al.* extend this work to crowded scenes [TKB09].

Salomon *et al.* [SGLM03] and Nieuwenhuisen and Overmars [NO03] present similar approaches for interactive navigation in complex 3D environments. Their techniques are based on an algorithm that pre-computes a global roadmap of the environment by using a variant of a randomized motion planning algorithm along with a reachability-based analysis. At runtime, the algorithm can automatically compute a collision-free and constrained path between two user specified locations. Oskam *et al.* [OSTG09] extend the idea and present an approach for computing visibility-aware paths.

Andujar *et al.* [AVG04] present *Way-finder*, another automatic system for generation of walkthroughs in 3D environments that can be segmented using a cell-and-portal approach. The approach splits the environment into cells (e.g. rooms), and portals (e.g. doors or windows) that connect the cells. The decomposition of the scene together with an entropy-based measure of the relevance of a viewpoint can be used to determine a path which traverses all the relevant cells (see Figure 7 for an overview of this approach). The tour generation algorithm presented by Elmqvist *et al.* [ETT07], like the

Wayfinder system's algorithm, is based on a voxelized version of the 3D world. However, Elmqvist *et al.* employ an algorithm that builds disjoint visibility subsets and performs Traveling Salesman Problem (TSP) computations on the resulting connectivity graph. Sokolov and Plemenos [SP08] propose a semi-automatic method for virtual world explorations that use both topological (based on scene geometry) and semantic knowledge (formalized by the scene's author) about the scene.

Xiao and Hubbold [XH98] describes the approach which uses artificial force fields that act upon the user's virtual body such that he is guided around obstacles, rather than penetrating or colliding with them. The authors use a linear force field, in which the repulsive force is inversely proportional to the shortest distance between the user's virtual body and the object in question.

2.4.4. *Assisted navigation*

Assisted navigation, also referred to as the guided tours technique, is based on guided navigation, and was developed to support (especially novice) users in navigating the virtual world, finding points of interests and learning how to interact with them by exploiting an animated character that leads the user on a guided tour of the virtual environment.

Wernert and Hanson [WH99] propose a design that incorporates a personal '*guide*' that serves two important functions: keeping the user oriented in the navigation space, and pointing to interesting objects/places as they are approached. The design has some interesting features:

- If the user is active, these cues may be ignored. However, if the user stops, the gaze shifts automatically towards whatever the guide was interested in;
- This design allows any given viewer to be seen as the guide of one or more viewers following behind;
- The leading virtual guide (the authors selected a guide dog for this avatar) can remind the leading live human guide of interesting sites to point out.

For the 3D Web context, Chittaro *et al.* [CRI03] propose the adoption of a humanoid animated character to lead the user on a guided tour of the world. The authors suggest that the introduction of an animated character has the additional advantage of making the virtual world more lively and attractive for the user. Currently, guided tours are very often used in games (e.g., Star Wars: Knights of the Old Republic) to help players learn the game's UI.

2.4.5. *Adaptive navigation*

Adaptive hypermedia systems build a model of the goals, preferences and knowledge of each individual user, and use this model throughout the interaction with the user, in order to adapt to the needs of that user [Bru01]. Such a design approach is widely used on the Web, where users are recognized through their login/password or cookies, and receive customized versions of web sites. Similarly, some works propose adaptive architectures for generating dynamically personalized 3D worlds [CR02, WC02]. Several works have also focused on adaptive navigation support for virtual environments. Russo Dos Santos *et al.* [RDSGA*00] believe that navigation can be improved by adapting the navigation schemes to the virtual world and to the user's tasks. This belief led the authors to the concept of metaphor-aware navigation, the navigation that is tightly bound to the used visual metaphor and where the way the user moves in the virtual world is determined by the metaphor that the same world is based upon. Russo Dos Santos *et al.* note that navigation may also take different modes in the same world, depending on the hierarchical level of the metaphor in which the user is moving; for example, flying at district level and walking at street level in a metaphoric city. Hughes *et al.* [HBL02] proposed to develop navigation techniques based on established methods from adaptive hypermedia: direct guidance (a strict linear order through the navigation space), hiding (restricting the number of navigation options to a limited subset), sorting (altering the order in which navigation decision are presented to the user) and annotation (displaying additional information on navigation options). Celentano *et al.* [CNP04] have also focused on interaction adaptivity; their agent-based approach is used for monitoring the user activity and for proactively adapting interaction. Much broader discussion of the concepts, issues and techniques of adaptive 3D web sites is presented in [CR07].

In addition to the difficulties of controlling the viewpoint, there is a problem of wayfinding, especially in large virtual worlds. It is related to how people build up an understanding (mental model) of a virtual environment. This problem, also known as a problem of users getting '*lost-in-space*', may manifest itself in a number of ways [DS96]:

- Users may wander without direction when attempting to find a place for the first time.
- They may then have difficulty relocating visited places.
- They are often unable to grasp the overall topological structure of the space.

Efficient wayfinding is based on the navigator's ability to conceptualize the space. This type of knowledge, as defined by Thorndyke [Tho82], who studied the differences in spatial knowledge acquired from maps and exploration, is based on: *survey knowledge* (knowledge about object locations, inter-object distances and spatial relations) and *procedural knowledge* (the sequence of actions required to follow a particular route). Based on the role of spatial knowledge in wayfinding tasks, designers have concerned themselves with developing design methodologies that aid navigation.

2.4.6. *Generalization & stylization of 3D environments*

A generalization of a concept is an extension of the concept to less-specific criteria. It is a foundational element of logic and human reasoning. Generalization has a long history in cartography as an art of creating maps for different scales and purposes. Cartographic generalization is the process of selecting and representing information of a map in a way that adapts to the scale of the display medium of the map. Correctly generalized maps are those that emphasize the most important map elements while still representing the world in the most faithful and recognizable way. The level of detail and importance in what is remaining on the map must outweigh the

Figure 8: *An example generalization: 3D view of Louvre Museum on Google Maps.*

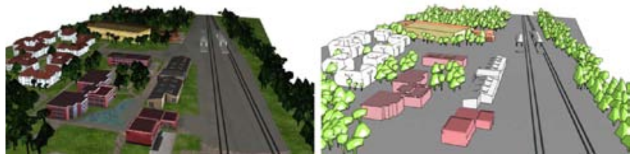

Figure 9: *Photorealistic and non-photorealistic display of a virtual 3D map [DB05]. Image courtesy of Jürgen Döllner.*

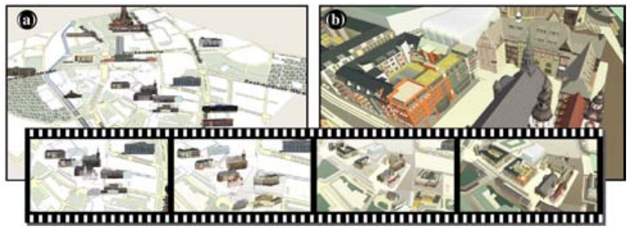

Figure 10: *Seamless transition between abstract graphics (a) and a photorealistic version (b) view-dependently. The sequence below shows frames of transition [STKD12]. Images courtesy of Jürgen Döllner.*

insignificance of items that were generalized, as to preserve the distinguishing characteristics of what makes the map useful and important [BM91, AS01, Mac04].

Quite often virtual 3D environments, such as 3D digital cities, can be characterized by a large number of objects of different types, manifold structures and hierarchies among them, and a high degree of visual detail. These models transport a huge amount of different information, for example, encoded in facade textures, complex building models, vegetation. This frequently leads to perceptional and cognitive problems for the user due to visual noise and information overload and, therefore, impairs tasks and usability, for example, with respect to orientation and navigation in geovirtual environments [GD07, TGBD08].

While for some cases photorealistic rendering is appropriate, other-use cases may need a more abstracted model. Therefore, to facilitate comprehension, interaction and exploration of 3D environments, the principle of generalization has been applied to create an abstract representation of the models. In cartography, the term generalization describes the process of reducing details of the depicted spatial information to a degree that is appropriate to scale, task and viewer [TS04, And05, MF07, For07].

Meng and Forberg [MF07] give a good overview of on-going research efforts in automatic 3D building generalization, including theoretic concepts and assessment of the feasibility of methods. The authors present implementation work, elucidating ideas in automatic segmentation, recognition, simplification and viewer-dependent visualization of 3D building structures. Other good overview of the techniques for generalizing building geometry of complex virtual 3D city models is provided in [GD08b].

With the recent advent of non-photorealistic 3D rendering [GGSC98, SS02], some research has been done on non-photorealistic 3D rendering of virtual 3D maps to reduce their visual complexity (see e.g. Figure 9). Such expressive city models allow for new applications in cartography, such as illustrative maps, artistic maps and informal maps. Döllner *et al.* [DB05, DBNK05] outline main techniques and discusses consequences of non-photorealistic 3D rendering for cartographic information display based on 3D city models. Pan *et al.* [PZG*13] present an expressive rendering approach for visualizing large-scale 3D city scenes with various rendering styles integrated in a seamless way. Each view is a combination of the photorealistic rendering and the non-photorealistic rendering to highlight the information that is interesting for the users and de-emphasize the other that is less important.

Semmo *et al.* [STKD12] present a concept and an implementation of a system that enables different presentation styles, their seamless integration within a single view and parametrized transitions between them, which are defined according to tasks, camera view and image resolution (see Figure 10). The authors also outline potential usage scenarios and application fields together with a performance evaluation of the implementation.

Chittaro and Scagnetto [CS01] investigated semi-transparency (allowing users to see through occluding surfaces) as a navigation aid in Virtual Environments. Their experimental study has shown a positive effect of semi-transparency on user navigation performance and awareness of the VE structure (although not as significant as bird's-eye views). The results of more recent study [EAT07] also show that dynamic transparency results in significantly more efficient object discovery. For a similar purpose, we created a sketch-based system for creating illustrative cutaway renderings [KHG09] (see Figure 11). Semi-transparency as a navigation aid is now a common practice in the mapping engines such as Google Maps (see Figure 8).

2.4.7. *Landmarks*

Darken and Sibert [DS96] present a toolset of techniques based on principles of navigation derived from real-world analogs. Their evaluation shows that subjects' wayfinding strategies and behaviours were strongly influenced by the environmental cues. Vinson [Vin99] proposes design guidelines for landmarks to support navigation in

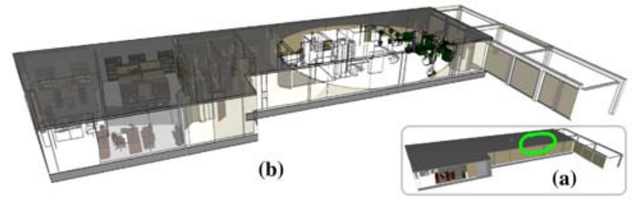

Figure 11: *Exploration of an office: (a) sketching a circular cutaway; (b) final composite illustrative rendering including cutaways, transparency and ghost-views [KHG09].*

Table 1: *The functions of different kinds of landmarks in Virtual Environments (based on [Vin99]).*

Lynch's types	Examples	Functions
Paths	Street, canal	Channel for navigator mvt.
Edges	Fence, river bank	Indicates district limits
Districts	Neighbourhood	Reference region
Nodes	Town square, public building	Focal point for travel
Landmarks	Statue	Reference point

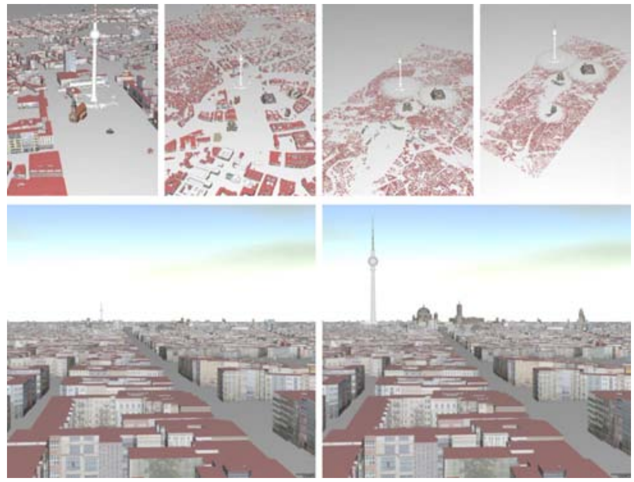

Figure 12: *Top: Behaviour of landmark enhancement with increasing distance to the camera. Bottom: Comparison between standard (left) and enhanced rendering (right) in a perspective view close to the ground [GTD07]. Images courtesy of Jürgen Döllner.*

Virtual Environments; he focuses on the design and placement of such navigation aids. Some of these guidelines are:

- It is essential that the VE contain several landmarks.
- Include different types of landmarks such as paths, fences.
- Landmarks should be distinctive, not abstract, visible at all navigable scales.
- Landmarks should be placed on major paths.

Vinson created a classification of landmarks based on Lynch's classification [Lyn60]. Table 1 summarizes Vinson's design guidelines for the different classes of landmarks.

It is worth mentioning research that has been done towards the automatic detection of landmarks, for example [Eli03, EB05, RW02]. Visible landmarks [PP04] make distant landmakrs visible and allow users to travel relative to those landmarks with a single gesture. Glander *et al.* [GTD07] present a novel concept for the real-time depiction of landmarks that effectively emphasizes these 3D objects by improving their visibility with respect to their surrounding areas and the current 3D viewing settings. The concept is based on scaling landmark geometry according to an importance function while simultaneously adjusting the corresponding surrounding region (see Figure 12).

In the following work, Glander and Döllner [GD08a] present an interesting technique to automatically generalize virtual 3D city models and create a visualization that dynamically exaggerates global landmark objects. The technique uses the infrastructure network to cluster buildings in cell blocks, while preserving local landmarks in their representation. A landmark hierarchy is created based on these initially found landmarks. As the generalization process creates representations of higher levels of abstraction, only landmark buildings from the higher levels of the landmark hierarchy are preserved, effectively reducing their number for clarity. Finally, the technique exaggerates the most important global landmarks dynamically by scaling them.

Delikostidis *et al.* [DER*13] presented the theoretical framework of landmark visibility analysis, applied as a landmark visibility indication function in the development of LandNavin, a mobile navigation system for pedestrians. The results showed that a landmark visibility indication constitutes a useful tool for the majority of the participants, helping them relate reality with the mobile map more easily, which also supports the development and use of their cognitive maps.

Takahashi *et al.* [TYSN06] present occlusion-free representation of routes for the mountain areas for application to car navigation systems. This work has been followed by Möser *et al.* [MDWK08], who distort the terrain surface to be aligned with the predefined base surface to reduce possible occlusions of geographic features. Qu *et al.* [QWC*09] present a related focus-plus-context zooming technique, which allows users to zoom into a route and its associated landmarks in a 3D urban environment. According to presented experiments, the algorithm can handle various routes in different 3D environments. Moreover, the presented focus + context route zooming visualization seems to be very useful for users to navigate in 3D urban environments—the authors report that during the experiment it reduced some time-consuming user interactions such as switching between different views, zooming and panning.

Hirono *et al.* [HWAT13] present a similar approach to disoccluding important geographic features when creating 3D urban maps for enhancing their visual readability based on formulating the design criteria as a constrained optimization problem and solved using the linear programming approach. Figure 14 shows a side-by-side comparison between an original 3D urban map and its optimized version.

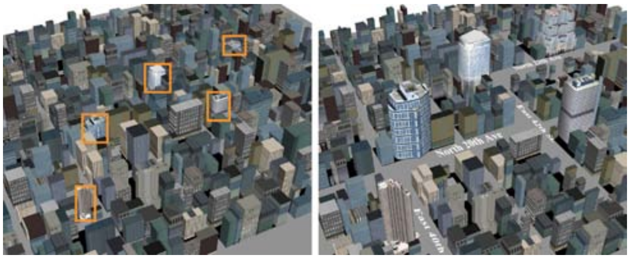

Figure 13: *Focus-plus-Context route zooming and information overlay: (a) A route and its associated five landmarks in a 3D virtual city are occluded by other context buildings; (b) The route and landmarks are enlarged with minimum distortion after applying the route zooming technique [QWC*09]. Images courtesy of Huamin Qu.*

Figure 14: *Side-by-side comparison between an ordinary perspective view (left) and its optimally distorted version (right) [HWAT13]. Images courtesy of Daichi Hirono.*

Trapp *et al.* [TSHD09, TSL*11] present strategies that aim at effectively visualizing 3D points-of-interest in a 3D virtual environment used on mobile devices. In their following work, the authors present a framework that unifies various highlighting techniques and is especially suitable for the interactive rendering 3D digital maps [TBPD11].

2.4.8. *Mini-maps & orientation widgets*

Three-dimensional navigation techniques allow users to explore 3D environments at different levels of detail. However, it is often useful to display more than one level of detail simultaneously. An Overview-plus-Detail interface design can be characterized by the simultaneous display of both an overview and detailed view of an information space, each in a distinct presentation space [PCS95, BGBS02, CKB08]. Such designs have been shown to be effective in navigating large 2D information spaces such as maps [HBP02]. According to some studies [DS96, RM99], [RPJ99], this is also the case in a 3D virtual environment. Mini-maps, an example of Overview-plus-Detail interface, are now very popular interface components in computer games. These miniature maps, typically placed in a corner of a UI, display terrain, important locations and objects and dynamically update the current position of the user with respect to the surrounding environment.

Chittaro and Venkataraman [CV06] proposed and evaluated 2D and 3D maps as navigation aids for multi-floor virtual buildings and found that while the 2D navigation aid outperformed the 3D one for the search task, there were no significant differences between the two aids for the direction estimation task. The study of three wayfinding aids (a view-in-view map, animation guide and human system collaboration) [WZHZ09] shows that although these three aids all can effectively help participants find targets quicker and easier, their usefulness is different, with the view-in-view map being the best and human system collaboration the worst.

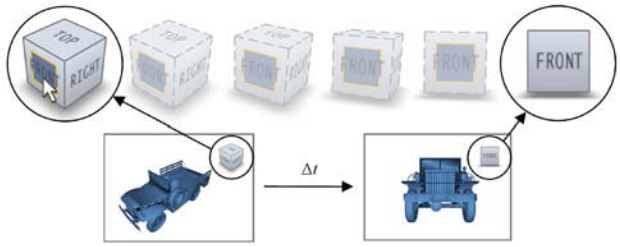

Figure 15: *The ViewCube: clicking on the 'front' face of the cube widget rotates the cube and the 3D scene to the front view [KMF*08]. Image courtesy of Azam Khan.*

Smith and Hart [SH06] presented a user study that evaluated how the different wayfinding aids reduced or increased the cognitive load, measured as usability problems, on the user while they navigated around an environment searching for landmarks. The measured cognitive loads of each wayfinding aid: distance only (simply given the distance to a selected landmark)—Medium, graphical compass—High, dead reckoning (a simple text message to indicate the required movement to the highlighted landmark)—Low, map-based route planning—High, graphical plan following (inserting a path into the virtual environment)—High.

Research in cognitive sciences suggests that orientation and navigation along routes can be improved if the graphical representation is aligned with the user's mental concepts of a route. Glander *et al.* [GPTD09] analyse an existing 2D schematization approach called wayfinding choremes and present an implementation for virtual 3D urban models, transferring the approach to 3D.

Chittaro and Burigat [CB04, BC07] propose 3D arrows as a navigation aid that aims at allowing users to easily locate objects and places inside large-scale VEs. The authors asses the usefulness of the proposed solution, contrasting it with more traditional 2D navigation aids (2D arrows, radar/map). The results of the experiment showed that the proposed 3D navigation aid is at least as effective as 2D navigation aids in helping users during 'walk' mode navigation, while it outperforms them in "fly" mode.

To help address the problem of disorientation of CAD users that occur while developing new 3D scenes, Khan *et al.* [KMF*08] present an in-scene 3D widget called the ViewCube as a 3D orientation indicator and controller (see Figure 15). The ViewCube is a cube-shaped widget placed in a corner of the window. When acting as an orientation indicator, the ViewCube turns to reflect the current view direction as the user re-orients the scene using other tools. When used as an orientation controller, the ViewCube can be dragged, or the faces, edges or corners can be clicked on,

Figure 16: *Effects of our non-linear perspective. Left: bending far distances to the viewer helps to get an overview. Centre: an S-shape deformation adds horizon information. Right: an inverse bend results in a top-down view augmented with the horizon [MDWK08]. Images courtesy of Reinhard Klein.*

to easily orient the scene to the corresponding view. McCrae *et al.* [MGG*10] examines user understanding of position and orientation within multi-scale 3D data sets.

Fitzmaurice *et al.* [FMM*08] have developed a concept called Safe 3D Navigation, where they augment navigation tools with properties to reduce the occurrence of confusing situations and improve the learning experience. They have defined seven properties that work together to achieve a safe 3D navigation experience: (1) cluster and cache tools; (2) create task and skill-based tool sets; (3) provide orientation awareness; (4) enhance tool feedback; (5) offer pre-canned navigation; (6) prevent errors; and (7) recover from errors.

While researching the relation between wayfinding and motion constraints, Elmqvist *et al.* [ETT08] found out that navigation guidance allows users to focus less on the mechanics of navigation, helping them in building a more accurate cognitive map of the 3D world.

2.4.9. *Focus-plus-Context*

Focus-plus-Context information visualization techniques, for example, fisheye views [Fur86], allow users to have the information of interest in the foreground and all the remaining information in the background simultaneously visible. These techniques allow users to view selected data portions in additional detail without requiring a second window, so that split attention is avoided.

Focus-plus-Context information visualization techniques proved to be useful in many situations [CKB08]. As for the 3D interaction context, Robertson *et al.* [RCvD97], while exploring immersion in Desktop VEs, proposed a new navigation aid called Peripheral Lenses as a technique for simulating peripheral vision. However, as their results were not statistically significant, further studies are needed to understand exactly when Peripheral Lenses are effective.

Some works used Focus-plus-Context visualization techniques in 3D urban environments. Carpendale *et al.* [CLP04] introduced the deformation technique into 3D environments to magnify the region of interest. Trapp *et al.* [TGBD08] made use of generalization lenses to gain focus + context visualization in virtual urban environments. Some authors [VC01, MDWK08] propose a perspective that allows blending between the familiar pedestrian perspective and a standard map depiction with reduced occlusion (see Figure 16). Pasewaldt *et al.* [PSTD12] present a rendering technique for a view-dependent

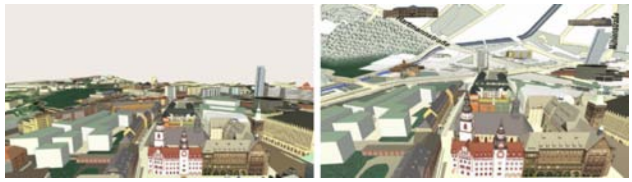

Figure 17: *Exemplary perspective view of a virtual 3D city model rendered photorealistically (left) in comparison to a multi-perspective view (right) [PSTD12]. Images courtesy of Jürgen Döllner.*

parametrization and interpolation of global deformations based on B-Spline curves (see Figure 17).

2.4.10. *Use of personal marks and history keeping*

Another technique for supporting interaction and navigation is augmenting virtual environments with interactive personal marks [GMS06]. Darken and Sibert [DS93] were first to present the concept: users can drop breadcrumbs, which are markers in the form of simple cubes floating just above the ground plane. Edwards and Hand [EH97] describe a similar technique, trailblazing, and the use of maps as examples of planning tools for navigation.

Grammenos *et al.* [GMS06] introduce concept of Virtual Prints (ViPs) as an intuitive metaphor for supporting interaction and navigation, as well as a number of additional tasks in virtual environments. The authors describe three types of ViPs: Virtual Footprints, which are used for tracking user navigation (position, orientation and movement); Virtual Handprints, which are used for tracing user interaction with the VE; and Virtual Markers, which are 'special' marks (usually coupled with information) that can be created upon user request. The results of the studies show strong evidence that an appropriately designed and implemented, fully functional ViPs mechanism can significantly increase the usability of VEs.

Frequent Wayfinding-Sequence (FWS) methodology [SKLS06] uses a modified sequence mining technique to discover a model of routes taken by experienced users of a VE that can be later used to provide navigation assistance to novice users by recommending routes. The study of the FWS approach suggests that it has the potential to improve the user's navigation performance and the quality of the human–computer interaction (HCI).

3. Selection and Manipulation

Another typical task performed in a 3D virtual environment is object selection and its direct manipulation [Han97, BKLP01]. Interaction techniques for the 3D manipulation include three fundamental tasks: object translation (positioning), object rotation and object scaling.

3.1 Object selection

Selection is a process of identifying an object, a set of objects or parts of objects that are targets for subsequent action [Ste06]. The most common way to select objects in Desktop VEs is position a mouse cursor over a given object and clicking a mouse button. One technique to compute the selected object is based on ray casting [Bol80, Min95], [MT97]; it uses the ray from the eye point through the pixel currently selected by the mouse pointer to find the first intersection point with the scene (the surface of the target or to its approximated surface, e.g. bounding box). If the virtual ray intersects an object, it can be selected for further manipulation. Another method for mouse cursor–based object selection, called back-buffer selection [SSKLK13], requires double-buffer mode. When the user picks an object, the entire scene is redrawn in the back buffer, but instead of using the normal colours for objects, each object is given a unique colour which encodes an index for each object. To find out what the user selects, the application simply samples the colour at the pixel that the mouse cursor is positioned at and looks up what object that colour corresponds too.

Looser *et al.* [LCS05] examined target acquisition in 3D first-person computer games. They verified that Fitts' Law, which is widely used for modelling human performance for pointing in 2DUIs [Fit54, Mac92], accurately models this type of target selection. With a similar goal, Teather and Stürzlinger [TS11] investigated means of comparing 2D and 3D pointing techniques. Balakrishnan [Bal04] surveys the research on developing, analysing and evaluating new techniques for artificially improving target acquisition in 2D, but this work can also be useful in the 3D context.

Elmqvist and Fekete [EF08] adapt the concept of semantic pointing to a 3D context for improving object picking performance. The adaptation works by shrinking empty space and expanding target sizes in motor space. The authors also present a user study comparing the performance of object selection in two different game-like 3D environments with and without semantic 3D pointing. The reported study indicates that accuracy is improved by semantic 3D pointing, but not task completion time.

Jáuregui *et al.* [JAL12] focused on improving users' depth perception when interacting with 3D content in desktop virtual environments. The authors proposed and evaluated two interesting techniques: (1) Torch that combines the spot light and the specular highlight which provide additional depth cues, with a 2D cursor (the target point) to increase accuracy and an avatar of a lantern to reinforce the metaphor; and (2) Hand Avatar that displays a hand avatar placed and rotated according to the surface of the intersected 3D object, creating a pseudo-illusion of touch; the size of the projected hand avatar also provide additional depth cues. The authors also explored a pseudo-motion parallax effect as a complementary way to increase depth perception. The reported user study of the proposed techniques showed that both 3D cursors significantly increased the users' depth perception, but at the expense of increase the selection time and decrease the accuracy.

Silk Cursor [ZBM94] is another interesting selection technique, where in contrast to normal practice, the tracking symbol is a volume rather than a point. Moreover, the surface of this volume is semi-transparent, thereby affording occlusion cues during target acquisition. The experiment presented by the authors shows that the volume occlusion cues were effective in both monocular and stereoscopic conditions. IntenSelect [dHKP05] is another selection-by-volume technique that works well with selection of object that are occluded

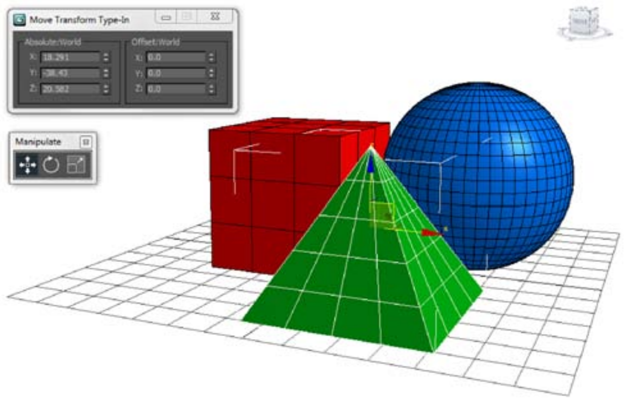

Figure 18: *The same functions presented with tools and with a dialog box (left), and with a manipulator (here controlling the green pyramid).*

or moving. The 3D Bubble Cursor [VGC07], a volumetric extension of the 2D Bubble Cursor [GB05], aims to support the selection of objects in dense target environments, or the selection of objects which are occluded from the user's viewpoint. The cursor uses a bubble that jumps between potentially selectable objects and dynamically resizes such that only the closest target falls within its boundaries. For highly cluttered environments, some researchers [KBB11, CWL12, BKB13] proposed and studied progressive refinement of the set of selectable objects to reduce the required precision of the task.

CAD and 3D modelling applications allow for interaction with many objects in one operation and most of these systems include some variant of rectangle selection and shift-clicking to support group selection [ZHH06, LI05, OSD06]. Another technique, often used in CAD and 3D editing software, is indirect selection of objects from a list. For more information about selection (also in immersive VEs) see the work of Argelaguet and Andujar [AA09, AA13].

3.2. Object manipulation

Manipulating objects is another basic task performed in a 3D virtual environment, especially during scene construction. When designing a scene with multiple objects, users have to repeatedly realign and adjust different parts. It is important to provide an efficient object manipulation technique.

The mouse, an excellent input device for 2DUIs, is commonly used to position objects in 3D environments. This causes the problem of how to map a fundamentally 2D input device into different operations with different DOFs. One solution to the problem is to provide users with a set of tools, dialog boxes, menu items or other kinds of 2DUI elements for changing between manipulation modes, where mode is a mapping of an input to a possible output (see Figure 18 (left)). Much simpler and more efficient solution is to provide users with manipulators (see Figure 18 (the pyramid)) for explicit and more direct manipulation [NO87, Bie87, SC92, CSH*92].

3.2.1. *Manipulators*

Strauss *et al.* [SIS02] define manipulators as visible graphic representations of an operation on, or state of, an object, that is displayed together with that object. This state or operation can be controlled by clicking and dragging on the graphic elements (handles) of the manipulator. Straus *et al.* also list some of the advantages of manipulators:

- Manipulators are located in the scene with the objects they control. When the users edit these objects, their locus of attention stays with the object, not off to the side with the tools. This reduces the amount of mouse traffic, and reduces mode error.
- The users have a number of different controls available at the same time, so they can perform any one of several related operations at any time without an extra click to change tools. This cuts the number of clicks significantly and reduces memory load, since all the possible controls are displayed where they are needed.
- This solution allows the users to separate the desired 3D movement into simpler 1D or 2D components.
- Manipulators can graphically show what they are operating on and how they will work. They also show what operations are possible, in a given context, and can give the users additional feedback about intermediate states.
- Manipulators invite experimentation. They make using the software more enjoyable.

In Straus *et al.*'s notes from SIGGRAPH 2002 course on '*The Design and Implementation of Direct Manipulation in 3D*', there are also useful recommendations for manipulator behaviour and look.

Manipulators, which currently are adopted by most CAD and 3D editing software applications, evolved initially from ideas associated with interactive manipulation using 2D control devices. Positioning manipulators originate from the work carried out by Nielson and Olsen on direct manipulation techniques for 3D objects using 2D locator devices [NO87] and by Bier [Bie87] on skitters and jacks tools for interactive 3D positioning. Bier's technique aimed to reduce the complexity of the traditional scene composition by restricting the user to translate, rotate and scale about one kind of feature, namely an orthogonal Cartesian coordinate frame. Bier also developed a very useful technique for snap-dragging in three diemnsions [Bie90]. Rotation manipulators are also built upon Nielson's and Olsen's [NO87] and by Bier's [Bie87] work, which was later extended by Chen *et al.*'s work on Virtual Sphere [CMS88] and Shoemake's ArcBall [Sho92], the techniques designed for 3D navigation (rotation) around 3D objects. As we have already mentioned, both techniques are based on a concept of a virtual ball that contains the object to manipulate. They utilize the projection of the mouse location onto a sphere to calculate rotation axis and angle. Comparison of mouse-based interaction techniques for 3D rotation can be found in [BRP05].

Since the introduction of Skitters and Jacks [Bie87], 3D transformation widgets used in the manipulation of 3D objects have little evolved. These 3DUI components have been mainly designed for mouse-based systems, where the user benefits from accurate pointing, distant interaction, an unobstructed view of the screen and direct access to numerous buttons and keyboard shortcuts. Touchscreens have none of these qualities [Mos09]. Consequently, 3D transformation widgets need to be reinvented to adapt to the tactile paradigm.

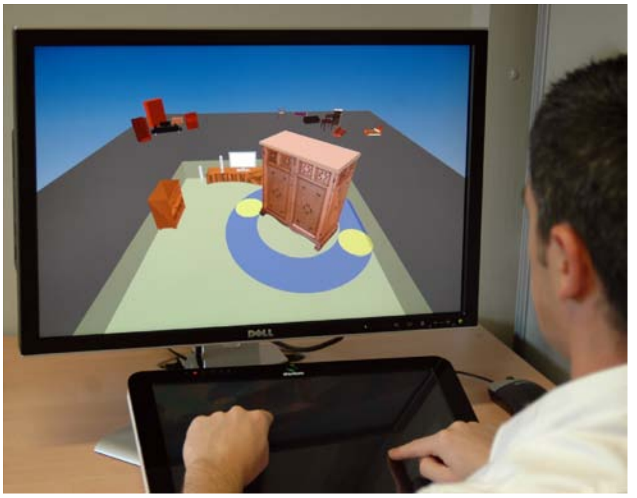

Figure 19: *Indirect control of the RST technique [KH11].*

3.2.2. *Touch-based manipulation*

An example of new 3D transformation widgets has been proposed by Schmidt *et al.* [SSB08], who present a set of sketch-centric techniques for precisely positioning and orienting 3D objects in free-form sketch-based interfaces, such as SKETCH [ZHH06] or ShapeShop [SWSJ05] that supports buttonless, imprecise touch-based input without sacrificing usability. The authors report on a pilot user evaluation that provides preliminary evidence that the techniques are intuitive and easy to learn as well as efficient enough to be competitive with the 'standard' manipulator-based interface.

Others have explored multi-touch controls where several DOFs can be manipulated at the same time. In particular, the already mentioned work of Reisman *et al.* [RDH09] extend the RST multi-touch technique to 3D that can be used for both navigation and manipulation. We conducted a user study to better understand the impact of directness on user performance for an RST docking task, for both 2D and 3D visualization conditions [KH11]. This study showed that direct-touch shortens completion times, but indirect interaction (see Figure 19) improves efficiency and precision, and this is particularly true for 3D visualizations. The study also showed that users' trajectories are comparable for all conditions (2D/3D and direct/indirect). This tends to show that indirect RST control may be valuable for interactive visualization of 3D content.

Hancock *et al.* propose techniques where users manipulate 3D objects with one, two or three fingers in shallow depth [HCC07]. Their study revealed that users are fastest and most accurate when using the three-touch technique and that their preferences were also strongly in favour of the expressive power available from three touch. In their following work, the authors introduce Sticky Tools [HtCC09], virtual 6DOF 3D tools that allow force-based interaction

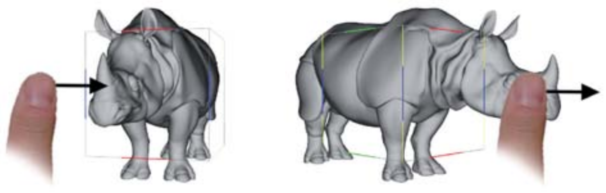

Figure 20: *tBox: dedicated to touch-screens transformation widget [CDH11].*

as force-based effects such as collisions, gravity, mass and inertia can be now easily integrated into 3D environments through the use of physics engines. Martinet *et al.* [MCG10b] studied these techniques, and proposed an extension called DS3—a 3D manipulation technique based on a total separation of the control of position and orientation. Results of a manipulation experiment (3D peg-in-hole task) show the strength of the new technique for both performance and user preference. Martinet *et al.* [MCG10a] also proposed and evaluated two techniques dedicated to 3D positioning: (1) the Z-technique that presents 3D data in full-screen and (2) the technique that splits the screen in four viewports. While the docking task experiment was not able to draw conclusions on performance, the participants clearly preferred the full-screen technique.

In our own work, we studied how manipulators [NO87, Bie87, SC92, CSH*92] can be adapted to the tactile paradigm. We have explored an approach where users apply rotations by means of physically plausible gestures, and we have extended successful 2D tactile principles to the context of 3D interaction. These investigations led to the design of a new 3D transformation widget, tBox [CDH11], that can been operated easily and efficiently from gestures on touch-screens (see Figure 20). Preliminary experiments (assembling a '*Mr. Potato Head*' character) have shown that tBox can be used easily both by 3D expert and novice users for 3D manipulation tasks on a touch screen.

Au *et al.* [ATF12] presents a set of multi-touch gestures which offers a seamless control of manipulation constraints (i.e. axis or plane) and modes (i.e. translation, rotation or scaling). Their technique does not require complex manipulation widgets but uses candidate axes, which are for visualization rather than direct manipulation. The authors state that such design minimizes visual clutter and tolerates imprecise touch-based inputs. The reported evaluation shows that the technique is more effective than a direct adaption of standard transformation widgets to the tactile paradigm. The same research group introduced two-finger gestures for 6DOF manipulation of 3D objects [LAFT12], as single-hand, two-finger operations are often preferred for multi-touch devices, that is, smartphones, to cause less hand occlusion and relieve the other hand for necessary tasks like holding the device. The docking task study revealed that with fewer fingers and fewer hands, the technique outperforms the Screen-Space and DS3 methods and is comparable to the Sticky Tools method.

Herndon *et al.* [HZR*92] propose to use *shadows* to view and to interact with objects in 3D virtual environments. These interactive shadows allow users to translate, rotate and scale objects in a constrained manner. For example, by dragging the shadow of an object, the object itself is translated by the same amount in a plane parallel to the shadow plane.

3.2.3. *Pseudo-physical manipulation*

Some research has been done to exploit knowledge about real world (e.g. gravity) and natural behaviour of objects (e.g. solidity) to provide simple and intuitive interaction techniques for object manipulation [Hou92, BS95, GS99, SSS01].

Bukowski and Séquin [BS95] employ a combination of pseudo-physical behaviour and goal-oriented properties called '*Object Associations*' to position objects in a 3D scene. In their approach objects selected for relocation actively look for nearby objects to associate and align themselves with; an automated implicit grouping mechanism also falls out from this process. They summarize some of the desirable paradigms of this kind of manipulation and some of the behavioural aspects of objects:

- Selected objects should follow the mouse pointer, so that '*point and place*' becomes an integral, intuitive operation.
- Objects typically should not float in mid-air but rest on some supporting surface. If the cursor points to the surface of a desk, it can be implied that the user wants to move the selected object to that particular surface.
- Associations of objects with reference objects should be maintained even when the reference object moves or is changed in other ways; however, they must also be breakable so that objects can be lifted off a surface easily and moved somewhere else.

Bukowski and Séquin have found that mixing the '*magical*' capabilities of 3D editing systems with some simulations of real behaviour reduces the DOFs the user has to deal with explicitly and results in a very attractive and easy-to-use 3D manipulation technique.

Goesele and Stuerzlinger [GS99] built upon the work of Bukowski and Séquin on Object Associations and Bier's technique for snap-dragging and introduced Semantic Constraints. They noticed that in addition to the physical properties people expect objects in a natural environment to behave according to their experience: '*People know that a chair will be standing on a floor and a painting will be hung on a wall. Hanging the chair on the wall and placing the painting on the floor is against common sense*.' The authors' system builds scenes based on a pre-defined library of objects, where each object is given pre-defined offer and binding areas. These areas are used to define surface constraints between objects. Offer areas are places on the object surface where other objects can connect to the object. Binding areas are their counterpart and mark areas on the object surface where it can connect to other objects. (see Figure 21).

Smith *et al.* [SSS01] presented the evaluation of Bukowski and Séquin's Object Associations [BS95] and Goesele and Stuerzlinger's Semantic Constraints [GS99]. They showed that in most cases constraints provide more than a factor of two speed-up combined with a significant increase in accuracy. Oh and Stuerzlinger [OS05] present a technique to manipulate objects in CAD/DVE using 2D input devices that utilizes the fact that people easily recognize the depth-order of shapes based on occlusions.

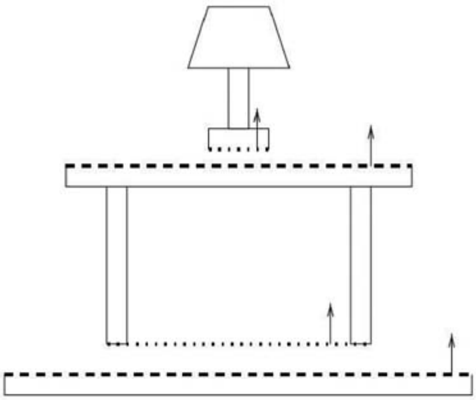

Figure 21: *Table, lamp and floor with offer areas (dashed) and binding areas (dotted). Orientation vectors are perpendicular to the offer or binding areas and define their binding direction [GS99].*

In their technique, the object position follows the mouse cursor position, while the object slides on various surfaces in the scene. The evaluation of the new technique showed that it compares very favourably to the conventional technique of using positioning manipulators.

4. Application Control

As we already mentioned, the interaction in virtual environment systems can be characterized in terms of three universal interaction tasks: Navigation, Selection and Manipulation, along with Application/System control. Application Control describes communication between a user and a system, which is not part of the virtual environment [Han97]. It refers to a task, in which a command is applied to change either the state of the system or the mode of interaction [BKLP01]. Hand and Bowman *et al.* point out that although viewpoint movement and selection/manipulation have been studied extensively, very little research has been done on system control tasks. However, application control techniques have been studied intensively over the past 40 years in 2D '*point-and-click*' WIMP graphical UIs (interfaces based on windows, icons, menus and a pointing device, typically a mouse).

4.1. WIMP application control

The HCI and UI communities agree that a good application control technique should be easy to learn for novice users and efficient to use for experts; it has to also provide the means for the novice users to gradually learn new ways of using the interface [Shn83]. WIMP interfaces have proven to support all these characteristic. People using computer applications have become intimately familiar with a particular set of WIMP UI components: windows, icons, menus and a mouse. Since the introduction of WIMP interfaces in early 1980s, they are still the dominant type of interaction style.

The issuing of command in a WIMP-type interface is simple and it can typically be done through different means, for example, by finding and clicking the command's label in a menu, by finding and clicking the command's icon or recalling and activating a shortcut. Menus and icon bars are easier to learn as they are self-revealing. The set of available commands is readily visible during the use of the software. These interface components were designed specifically to help users learn and remember how to use the interface without any external help (such as online help). Shortcuts address the problem of efficiently accessing the increasing number of functions in applications and limited physical screen space (menu items and toolbar icons must be physically accessible). The most common type of a shortcut is a keyboard shortcut, the sequence of keys which executes a function associated with a menu item or toolbar icon (e.g. Ctrl-C/Ctrl-V—Copy/Paste). To facilitate users learning new keyboard shortcuts, they are usually displayed next to related menu items.

Theoretical models, experimental tools and real applications have resulted in a generally very good understanding of traditional WIMP interfaces for 2D applications (such as word processing and spreadsheets). However, what if the information is represented in 3D space?

Currently, computer and video game industry leads the development of 3D interactive graphics and it is where many cutting-edge interface ideas arise. From what we can observe, most game interfaces are designed in a way, where application control interface components are placed in a screen space, on a 2D plane called HUD (head-up display) that is displayed side-by-side with a 3D scene or overlays the 3D scene.

World of Warcraft is one of the most successful games of all time. From its release in 2004, its interface evolved only a little and we believe that it is a good example of how to design a game interface properly. World of Warcraft's UI provides players with a great amount of useful information, while allowing them to easily and intuitively make actions they want. This balance was achieved through an emphasis on the following:

- The controls are clear: their physical appearance tell how they work [Donald A. Norman in his book '*The Design of Everyday Things*', points out: '*Design must convey the essence of a device's operation*' [Nor90]];
- At the beginning of the adventure in the Warcraft universe there are only few available UI components (health and mana bars, mini-map, few available capabilities or spells, experience bar, menus to change options and settings, text messages, chat text boxes). As the players gain more experience and more abilities, they can gradually enhance their interfaces and add more functionality to their toolbars. Hotkeys and shortcuts are also available for experienced players. The technique is based on an established design concept described by Ben Shneiderman [Shn83] as a '*layered or spiral approach to learning*'. Figure 22 presents a UI of an experienced player;
- Help is embedded throughout the interface. It allows players to click on any UI component for an explanation [SP04].

We found several works that are dedicated to game design, where the authors include some discussion around creating 3D game interfaces [Rou00, SZ03]. Richard Rouse, in his book '*Game Design: Theory & Practice*' [Rou00], while analysing the Sims game, pointed out that the best interface is the one that is difficult to notice:

Figure 22: *World of Warcraft's user interface of an experienced player (courtesy of Cyan).*

> *The best a game's interface can hope to do is to not ruin the players' experience. The interface's job is to communicate to players the state of the world and to receive input from players as to what they want to change in that game-world. The act of using this input/output interface is not meant to be fun in and of itself; it is the players' interaction with the world that should be the compelling experience.*
>
> Richard Rouse [Rou00]

While playing in the game's virtual environment, the most important part of the experience is the sensory immersion in an imaginary 3D space. As the game UI (e.g. the use of options, text-based chat) reminds the player that the game is a structured experience, it might be treated as an unwanted component. Greg Wilson, in his '*Off With Their HUDs!*' article in Gamasutra [Wil06], describes the shift of the game UIs away from a reliance on traditional HUDs.

> *Game developers' goal is to achieve a cinema-quality experience in a video game. One of the key ingredients for such an experience is the successful immersion of the player into the game world. Just as a filmmaker doesn't want a viewer to stop and think, 'This is only a movie' a game developer should strive to avoid moments that cause a gamer to think, 'This is just a game'.*
>
> Greg Wilson [Wil06]

For example, Far Cry 2 has a traditional HUD. However, it only appears in certain situations, and then fades out when it is not needed (the health bar appears when the player is hurt; the ammo shows up when a gun is running low).

Screen-space WIMP application control approach splits the UI into two parts with very different UI metaphors. While the navigation and manipulation functionality is accessible through a 3DUI, the rest of the system's functionality can only be controlled through a conventional 2D GUI. This can lead to a problem with supporting the feeling of immersion and directness. Hand [Han97] points out that this is a general problem with application control, since it may require the user to change from talking directly to the interface objects, to talking about them, thereby stepping outside the frame of reference used when manipulating objects or the viewpoint. If this shift is too great then the engagement of the user may be broken.

4.2. Post-WIMP application control

WIMP GUIs, with their 2D interface components, were designed for 2D applications (such as word processing and spreadsheets) controlled by a 2D pointing device (such as a mouse). However, when the information is represented in 3D space, the mapping between 3D tasks and 2D control widgets can be much less natural and introduce significant indirection and '*cognitive distance*' [vD97]. Andries van Dam argues that new forms of computing necessitate new thinking about UIs. He calls for the development of new, what he calls '*post-WIMP*' UIs that rely on, for example, gesture and speech recognition, eye/head/body tracking. Post-WIMP interfaces are also defined by van Dam as interfaces '*containing at least one interaction technique not dependent on classical 2D widgets such as menus and icons*' [vD97]. In the following, we describe some research works on post-WIMP interfaces that are used for application control in 3D Virtual Environments (see also a discussion on emerging post-WIMP reality-based interaction styles in [JGH*08]).

The ability to specify objects, an operation, and additional parameters with a single intuitive gesture appeals to both novice and experienced users [Rub91]. Such gesture-based system control was implemented in sketch-based 3D modelling interfaces of SKETCH [ZHH06] and ShapeShop [SWSJ05] that rely purely on gestural sketching of geometry but also commands for application control. Gesture-based interfaces, that offer an interesting alternative to traditional keyboard, menu and direct manipulation interfaces, found their way also to games. For example, in the *Black&White* game, to add to the sense of realism, the mouse-controlled hand can perform every function in the game and actions can be invoked by making hand gestures.

There are some system control techniques for Virtual Environments that we believe can be classified between WIMP and post-WIMP paradigms. For example, some UI designers transform WIMP system control techniques used in 2D interfaces, such as buttons and menus, and implement them in 3D environments in object space. Dachselt and Hinz [DH05] propose a classification of existing 3D widgets for system control. The classification is based on interaction purpose/intention of use and it was created for VEs. Dachselt, in his further work on the CONTIGRA project, focused on the menu selection subgroup. Together with Hubner [DH07], they provide a comprehensive survey of graphical 3D menu solutions for all areas of the mixed reality continuum. Compared to the screen-space WIMP techniques the object-space WIMP techniques maintain a UI metaphor supporting the feeling of immersion. However, such adaptation is not straightforward—2D objects that are placed in 3D space may undergo disproportion, disparity, occlusion. Moreover, selection (manipulation of a cursor) in 3D can be difficult to perform; users can have problems with selecting an item in a 3D menu floating in space [Han97].

In the seminal work on information visualization using 3D interactive animation, Robertson *et al.* [RCM93] use Interactive Objects as a basic input/output mechanisms for the Information Visualizer's

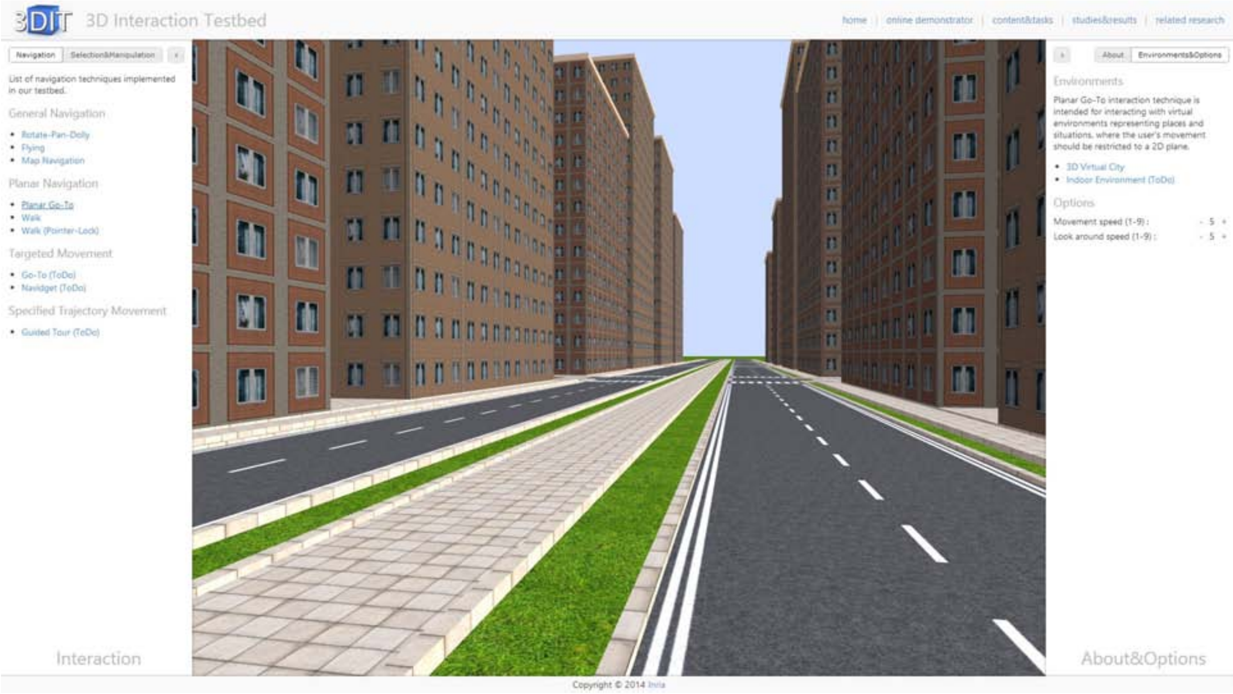

Figure 23: *The user interface of the 3DIT's Online Demonstrator. The left part of the interface is called Interaction and lists navigation as well as selection and manipulation techniques implemented in the testbed. The left part of the interface called About & Options provides description of the technique, suitable virtual environments and options, where different parameters of a given technique, such as movement speed, can be manipulated.*

UI. They are used to form the basis for coupling user interaction with application behaviour and build 3D widgets that represent information or information structure. The Web Forager [CRY96] is an information workspace for the Web built upon the Information Visualizer. Pierce *et al.* [PCvDR99] present toolspaces and glances to provide a mechanism for storing, accessing and retrieving 3D widgets and other objects in interactive 3D worlds. Toolspaces are attached to the user's avatar; objects placed in these spaces are always accessible (yet out of the user's view until needed). Users access these toolspaces through a type of lightweight navigation called glances. Robertson *et al.*, during their further work on information workspaces, developed Task Gallery, a window manager that uses interactive 3D graphics to provide direct support for task management and document comparison [RvDR*00].

Another technique that combines WIMP and post-WIMP to '*get the interface out of your face*' (as Bill Buxton calls his crusade—quoted from [vD97]) is the use of '*marking menus*' [KB94]. Such menu is designed to allow a user to perform a menu selection by either popping-up a radial/pie menu, or by making a straight mark in the direction of the desired menu item without popping-up the menu. Marking menus have been implemented, for example, in Autodesk Maya 3D modelling software.

5. 3DIT—3D Interaction Testbed

In the three previous sections we analysed, organized and reviewed all major interaction techniques for Navigation, Selection & Manipulation and System Control. However, besides the theoretical framework, we also need practical tools to carry out the research on 3D interaction.

Therefore, we have implemented 3DIT environment created to aid both researchers and developers of interactive 3D applications in having a clearer overview of the topic of 3D interaction. The site is intended to be an evaluation platform allowing for easier experimentation and assessment of different 3D interaction techniques. The testbed is available at: http://3dit.bordeaux.inria.frhttp://3dit.bordeaux.inria.fr and consists of three main parts: *Online Demonstrator*, *Content & Tasks* and *Studies & Results*. In the following we will describe these three parts in more detail.

5.1. Online demonstrator

There is no single interaction method can be considered perfect for any task and any 3D scene. The aim of *Online Demonstrator* is to provide a simple classification together with interactive demonstrations of the 3D interaction techniques coupled with suitable types of 3D environments allowing a researcher, a designer or a programmer to choose the best compromise for his/her needs.

Using the demonstrator, the user can choose between navigation as well as selection and manipulation techniques implemented in the testbed and secondly try how the given technique works in different types of virtual environments with different parameters (e.g. movement or look-around speed). Figure 23 presents the UI of 3DIT's *Online Demonstrator*.

The 3D interaction part of the demonstrator was implemented using three.js [Cab14] and WebGL [Khr10]. Adding a new interaction technique is relatively simple: for example, adding a new *X* navigation technique typically requires three steps: copying one of the existing *navigation_Z* html files and renaming it to *navigation_X*, adding *navigation_X* to the list of the techniques stored in the *files* variable in the *testbed.html* file and finally developing a controller class using three.js library.

5.2. Content & tasks

One problem with the research work on 3D interaction is that there are no clearly defined evaluation methods, stimuli as well as representative tasks akin to those found in other disciplines of science. Most often, researchers trying to propose a new 3D interaction technique have to spend considerable time not only on its development, but also on authoring new content and tasks for the technique's evaluation.

The *Content & Tasks* page is intended to list 3D environments that can be easily employed in testing different interaction techniques. At the time of this writing (March, 2014), there is one such environment called AlphaVille—an example of 3D outdoor environment (two other examples of 3D indoor and of 3D information visualization environments are in development). AlphaVille is a small (about 10 km^2) virtual 3D city created to represent the most common real-world street layouts (see Figure 24(a)). The model of the city is very simple—it consists of five parts: Center, East, West, North and South for the streets network and a number of blocks and houses. All of these low-poly models are available in OBJ, FBX and MAX file formats and are available for free as part of the 3D City Building Blocks project [Jan14a] that aims to provide an easy to use set of resources for the creation of simple 3D city environments.

The environments should be accompanied by the tasks that can be reused by evaluators. Currently, for the AlphaVille environment, we have defined the tasks representing various conditions a user is likely to experience using a 3D map application while being in the street level. Specifically for these tasks the layout of the streets was represented as a graph in the OpenStreetMap (OSM) format. Twelve paths of similar complexity, 4×1.5 km for the training and 8×2.5 km for the main part of the evaluation, were defined. For each route, five cars: a red bus, a red firetruck, a brown humvee, a black-and-white police car and a yellow taxi were parked on the way in a random order. All this information is available on the website. The tasks are based on the work of [BKH98] and include:

- *Travel Task (Camera Movement)*—For this task, the participants are asked to simply follow a pre-defined path marked with red arrows projected on streets (see Figure 24(b)). Once they arrive at the end of the path, they are asked to recall positions of the cars they have seen on the way (Figure 24(d)).
- *Travel & Look-Around Task (Camera Movement and Orientation*—This task was similar to the first task. Additionally, the participants are asked to *shoot* green balloons that were deployed on the way (15 balloons for each route). The positions of the balloons are clearly marked in the environment (see Figure 24(c)). To *shoot* a balloon the user has to orient the camera directly towards the target and wait for 1 s.

Both the AlphaVille environment and the street-level navigation tasks have recently been successfully used for a study of street-level navigation techniques in 3D digital cities on mobile touch devices [JH14].

5.3. Studies & results

Replicability, as it is known, is the foundation of modern research in many disciplines of science. Moreover, if consistency between researchers is achieved, more faith is placed in the truth of the findings. The aim of *Studies & Results* section is to list the 3DUI-related publications together with the information about environments used, implementations, tasks and results, the information that can be helpful not only to replicate the research, but also to lower the barrier for the development of new interaction techniques. The 3DIT project's purpose is to allow a '*hands-on*' understanding of 3D interaction principles, but also to create an open platform for defining evaluation methods, stimuli as well as representative tasks akin to those found in other disciplines of science. We are absolutely aware that the goal of creating such a testbed is ambitious and therefore we call for participation. The project is an open-source initiative and any contribution to the project is more than welcome. The testbed is hosted on github [Jan14b]; please refer to [Jan14c] to learn how to contribute to 3DIT.

6. Evaluation Techniques

As we have already mentioned, one of the aims of this paper is to produce a classification of 3D interaction techniques to assist researchers, developers and designers in creating new methods or improving existing ones. Such implementation of a technique is usually followed (or at least should be) by its evaluation. Therefore, in this section we describe methodologies that help in the process of evaluating UIs and interaction techniques. We also categorize all major evaluation techniques used by the authors of the articles referenced in this report.

A growing number of researchers and developers are considering usability as a major focus of interactive system development (e.g. [Nie00, NL06], SP04]). Usability often refers to the elegance and simplicity with which the interaction with applications is designed, and it is often defined as ease of use plus usefulness. Designing for usability from the very beginning of the development process not only gives developers better user acceptance but also keeps interaction designers away from producing interfaces and techniques for interaction that are unreasonable. In the following we present a list of methods that help in the process of evaluating interfaces and interaction techniques, and that focus on usability.

6.1. Usability Inspections

Usability Inspections [NM94] are usually performed in the form of Cognitive Walkthrough [WRLP94] (an approach to evaluating UIs

(a) Layout of the 3D city (b) Travel task (c) Look-Around task (d) Cognitive task

Figure 24: *(a) AlphaVille consists of five parts representing the most common real-world street layouts; (b), (c) and (d) present the tasks defined for the street-level navigation in AlphaVille.*

based on stepping through common tasks that a user would perform and evaluating the interfaces' ability to support each step) and Heuristic Evaluation [NM90] (a method in which usability experts separately evaluate a UI design by verifying its compliance with recognized usability principles, '*the heuristics*'). Usability Inspections, while involve usability specialists, are generally considered to be cheaper and quicker (in terms of generating results), when compared to usability testing.

Sutcliffe and Gault [SG04] present a heuristic method for evaluating virtual environment UIs. Their method, which consists of 12 heuristics addressing usability and presence issues, is based on Nielsen's usability heuristics [NM94], extended by VE-specific principles proposed by Sutcliffe and Kaur [SK00]. Similarly, Gabbard's taxonomy of usability characteristics in Virtual Environments [Gab97] and the Multi-criteria Assessment of Usability for Virtual Environments (MAUVE) system [SMR*03] may be utilized similarly to conduct a heuristic walkthrough of a system. Desurvire *et al.* [DCT04] describe *Heuristic Evaluation for Playability* (HEP), a comprehensive set of heuristics for playability. In the assessment of HEP effectiveness compared to more standard user testing methodologies, HEP identified qualitative similarities and differences with user testing. Another finding was that HEP is best suited for evaluating general issues in the early game development phases. To design, evaluate and understand enjoyment in games, Sweetser and Wyeth [SW05] selected the various heuristics from the literature into a concise model of enjoyment in games that is structured by flow. Their model, *GameFlow*, consists of eight elements: concentration, challenge, skills, control, clear goals, feedback, immersion and social interaction. Each element includes a set of criteria for achieving enjoyment in games. Over time, the model proved to help in understanding of enjoyment in games and in the identification of their strengths and weaknesses.

6.2. User studies

User studies involve measuring how well test subjects perform when interacting with the UI in terms of efficiency, precision/recall and subjective satisfaction. Such study can be made in the form of Formative Evaluation, *Post Hoc* Questionnaire/Interview or Summative or Comparative Evaluation (a statistical comparison of two or more configurations of UI designs, UI components, or interaction techniques). In the following we describe some of the user study techniques used to evaluate the interaction techniques reported in this State-of-the-Art Report; we will pay special attention to the tasks used in the studies.

6.2.1. *Testing viewpoint control*

Most of the movement techniques reported in this survey were evaluated using comparative studies: Navidget [HDKG08] was compared to fly, pan, look around, go-to and orbiting; Z-Goto to go-to [HDG06]; Speed-coupled Flying with Orbiting [TRC01] was studied against driving; Path Drawing [IKMT98] against drive and fly-to; Worldlets [ENK97] against text and image-based bookmarks; FWIP [KGMQ08] against joystick-based travelling technique; iBar [SGS04] against Maya's traditional Rotate-Pan-Zoom camera interface; Ruddle *et al.* [RHPJ00] studied hyperlink-based navigation against walking. With regard to the tasks, the authors typically used a variation of a search task [WF97, ENK97, RHPJ00, TRC01, HDG06, HDKG08, KGMQ08] and navigating while avoiding obstacles [IKMT98, XH98].

Ware and Osborne [WO90] chose different evaluation methodology and used a technique known as intensive '*semi-structured*' interviewing that involves interviewing a small number of subjects under controlled (structured) conditions, while at the same time providing some scope for the subject to provide creative input to the process. As the authors say, '*the goal of intensive interviewing as an evaluation technique is to ask many meaningful questions of a few subjects, rather than asking a single (often trivial) question of many subjects*'.

Evaluation of existing automatic camera planning systems typically involves measuring runtime performance (e.g. time to precompute the tour or memory usage) of the algorithms [SGLM03, AVG04, ETT07]. Li and Cheng [LC08] tested their real-time third-person camera control module using a maze-like environment, where the camera needs to track the avatar closely and avoid occlusions (see Figure 6 for the views acquired from the authors' engine).

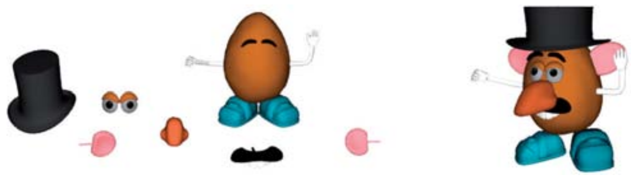

Figure 25: *For testing tBox [CDH11], we asked participants to assemble a cartoon character.*

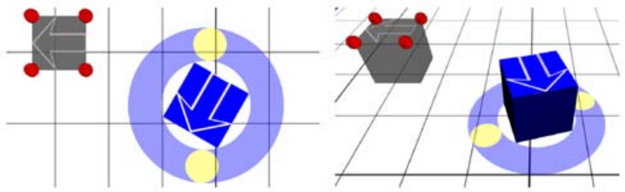

Figure 26: *Docking task. The source object (blue) has to be aligned with target object (grey transparent), in two dimensions (a) and in three dimensions (b). We used this setting in the study of indirect control of the RST technique [KH11].*

6.2.2. *Studies of wayfinding aids*

To study wayfinding aids, the authors of the papers referenced here typically compare the performance of the users on some task with different wayfinding aids. Typically the task involves: searching for a target object [EH97, RCvD97, EF08, WZHZ09, finding a number of targets in a specified order [CB04, BC07], exploring the space, finding the target, and returning to the starting point [DS93], finding a path to a specific object [CS01, CV06], direction estimation [CV06], learning the environment and, for example, sketching its map [DS96, RM99 or placing target objects on the map [BKH98, ETT08].

Elmqvist *et al.* [EAT07], when evaluating transparency as a navigation aid, used four different tasks that had two different purposes: discovery and relation. These tasks include counting the number of targets in the environment (discovery), identification of the patterns formed by the targets (relation), finding unique targets (discovery) and counting the number of targets (discovery, relation).

Smith and Hart [SH06] used a form of the think aloud protocol called cooperative evaluation, where users are encouraged to treat the evaluation as a shared experience with the evaluator and may ask questions at any time. The *interactive personal marks* technique [GMS06] was evaluated using the sequential [GHS99] comprehensive evaluation methodology (see Section 6.4 below for more details). ViewCube [KMF*08] was evaluated using the orientation matching evaluation technique.

6.2.3. *Testing manipulation techniques*

For the rotation task, in the experiments performed by Chen *et al.* [CMS88] and Hinckley *et al.* [HTP*97] subjects were shown a solid-rendered, upright house in colour on the right-hand side of the screen and were asked to match its orientation to a tilted house on the left-hand side of the screen. Bade *et al.* used more complex scan and hit type of aforementioned task [BRP05].

Authors investigating especially pseudo-physical manipulation techniques, studied the performance in a scene construction task, by for example, assembling a chair [OS05, SSB08], furnishing and decorating a room [Hou92, SSS01] or assembling a character, which was the task we used for testing tBox transformation widget [CDH11] (see Figure 25).

Most of the reported manipulation techniques, including [BK99], [HCC07, MCG10a, MCG10b, KH11, LAFT12, ATF12], were evaluated using a variation of the 3D docking task proposed by Zhai and Milgram [ZM98], where participants are required to scale, translate or rotate a source object to a target location and orientation in the virtual 3D space as quickly and accurately as possible (Figure 26 shows the setting we used in the study of indirect RST control [KH11]).

6.2.4. *Studies of 3D web interaction techniques*

Different possible measures could be used to determine the effectiveness and usability of the 3D Web interfaces. While working on the DMUI [JD12, JD13], in choosing tasks for the study, we looked for ones that are both valid (resemble a '*real*' act of browsing 3D content on the Web) and that are recognized for being able to detect significant differences. We decided to adopt tasks that were introduced by Chen *et al.* [CPB04] and were later successfully used by Polys *et al.* [PKB05, PBN11] and Ni *et al.* [NBC06] to evaluate information-rich virtual environments (IRVEs) [BNC*03]. Thus, the participants performed four types of tasks, representing various conditions a user is likely to experience on a 3D website:

1. Search for textual information and then search for visual information (S:H-3D). Task 1 requires the users to first search for text information, and then to find the corresponding visual information in the 3D scene. An example task is: *Find the Horde race that uses Hawkstriders for mounts. What other races are to the left and right of this race?*
2. Search for visual information followed by textual information (S:3D-H). Task 2 is conceptually reversed, in that the users are required to find the visual information on the 3D scene first, and then to answer questions about the related text information. An example task is: *Find the sword which hilt/handle has a yellow dragon eye and ends with dragon claws. What is the price of this weapon?*
3. Compare text information and derive visual information (C:H-3D) (find visual attributes of items with a given text criteria). An example task is: *Find the strongest creature in the museum. What is the colour of the creature's eyes?*
4. Compare visual information and derive textual information (C:3D-H) (search for textual attributes of items with a given visual criteria). An example task is: *There are two races with tails. What are their mounts?*

The example tasks come from the study, where we used a virtual museum featuring heroes, races, creatures and weapons from the fantasy setting of the World of Warcraft.

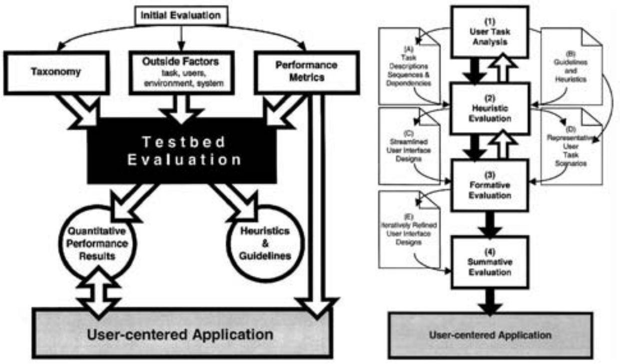

Figure 27: *Testbed [BJH99,BH99] and sequential [GHS99] comprehensive evaluation methodologies. Images courtesy of Doug Bowman.*

6.3. Informal user studies

Informal user studies (Inquiry) is a technique where usability evaluators obtain information about users' likes, dislikes, needs and understanding of the system by talking to them, observing them using the system in real work (not for the purpose of usability testing), or letting them answer questions verbally or in written form. The methods include: Observations, Focus Groups, Interviews, Questionnaires (e.g. QUIS [SP04]) and Logging Actual Use (see e.g. VU-Flow [CI04], [CRI06]).

6.4. Comprehensive approaches

Since interaction styles in virtual environments are radically different from standard UIs, some researchers proposed more comprehensive approaches for design and evaluation of VE user interaction. Bowman *et al.* [BJH99, BH99] propose the *testbed evaluation* approach (see Figure 27 left), the systematic study of the design, evaluation and application of VE interaction techniques. In this methodology, design and evaluation are based on a formal task analysis and categorization of techniques, using multiple performance measures. Gabbard *et al.* [GHS99] present a *sequential methodology* for ensuring the usability of virtual environments through user-centred design and evaluation. This approach, which addresses both design and evaluation of VE UIs, is based on sequentially performing user task analysis, heuristic (or guidelines-based expert) evaluation, formative evaluation and summative evaluation, with iteration as appropriate within and among each type of evaluation (see Figure 27 right). It leverages the results of each individual method by systematically defining and refining the VE UI in a cost-effective progression. Sequential evaluation was designed to be performed in the context of a particular application, while testbed evaluation should be employed in more generic evaluation contexts. Bowman *et al.*, in their survey of usability evaluation in virtual environments [BGH02], describe and compare the two evaluation approaches in detail.

7. Novel Input & Output Devices

As most of our state-of-the-art report focuses on the specific tasks that occur in 3D applications, which are independent of the input device, in the following we describe some VR and 3DUI research that can have an impact on interaction techniques for non-immersive 3D environments. We believe that novel devices will not replace mouse/keyboard and touchscreens in the following decade. However, we are focusing on 3D applications and in fact, this area is the one that can benefit the most from using novel devices.

7.1. Beyond the keyboard, mouse & touch

The keyboard has remained central to HCI from the dawn of the era of personal computing to the present. Even smartphones and tablets adapt the keyboard as an optional virtual, touchscreen-based means of data entry. However, this input device is not really suitable for manipulation in 3D VEs. Even though sometimes the cursor keys are associated to planar navigation, more often the keyboard is used for additional functions and shortcuts for expert users.

The input device that is most often used for interaction in 3D applications is a mouse. The big advantage of this input device is that most users are very familiar with using it and novice users learn its usage in a few minutes. The main disadvantage is that the mouse is a 2D input device and therefore the interaction metaphor must build a relationship between 2D input space and 3D virtual space.

The computer mouse design has remained essentially unchanged for over four decades. This now-familiar device was first demonstrated by Englebart [EE68] to the public in 1968 (the demonstration has become known as '*The Mother of All Demos*') and since then some effort has been made to augment the basic mouse functionality including research on developing new types of mice that can better support 3D interaction. For example, the now familiar scroll wheel, was originally added to support 3D interactions [Ven93] and now, in 3D interaction context, is used mainly for zooming in 3D applications. *Rockin'Mouse* [BBKF97] is a four DOFs input device that has the same shape as a regular mouse except that its bottom is rounded so that it can be tilted. As tilting gives a user control over two extra DOFs, the device is more suitable for interaction in 3D environments. Balakrishnan *et al.*'s results indicate that the Rockin'Mouse is 30% faster than the standard mouse in a typical 3D interaction task (simple 3D object positioning). Other interesting designs include *VideoMouse* [HSH*99], *Cubic Mouse* [FP00] *and Mouse 2.0* [VIR*09] that introduces multi-touch capabilities to the standard computer mouse.

More recently, interactive touch-based surface technologies have appeared. In particular, since the commercialization of the iPhone in 2007, an extremely rapid market penetration of multi-touch surface technologies has occurred. This evolution is redefining the way people interact with digital content.

Touch input favours direct and fast interaction with the underlying manipulated content. On the other hand, it also induces new constraints. In particular, with mouse-based systems, users benefit from accurate pointing, distant interaction, unobstructed views of the screen and direct access to numerous buttons and keyboard shortcuts. Consequently, the standard desktop 3D interaction techniques have been designed from these characteristics. On the contrary, touch -screens have none of these qualities as noted by Moscovitch [Mos09]. Consequently, new interaction techniques have been thought specifically to fit the touch-based paradigm. For example, recently we proposed a system that efficiently combines direct multi-touch interaction and 3D stereoscopic visualization [HBCdlR11].

In recent years, more and more affordable hardware can be found that allows the development of UIs more tailored for 3D applications. *Wiimote*, the device that serves as the wireless input for the *Nintendo Wii* gaming console, can detect motion and rotation through the use of accelerometer technology. Separating the controller from the gaming console, the accelerometer data can be used as input for gesture recognition [SPHB08]. Wiimote can be useful for 3DUI input: Wingrave *et al.*'s [WWV*10] tutorial presents techniques for using this input device in 3DUI and discusses the device's strengths and how to compensate for its limitations.

Some research has shown possible benefits of specialized 3D input devices over traditional desktop settings. For example, Kulik *et al.* [KHKF09] compared the keyboard and mouse interface to bi-manual interfaces using the 3D input devices *SpaceTraveller* and *Globefish* in a spatial orientation task requiring egocentric and exocentric viewpoint navigation. The presented study revealed that both interface configurations performed similarly with respect to task completion times, but the bi-manual techniques resulted in significantly less errors. Seagull *et al.* [SMG*09] compared a 3D two-handed interface *iMedic* to a keyboard-and-mouse interface for medical 3D image manipulation. Their case study revealed that experienced users can benefit when using iMedic interface.

7.2. Head & human pose interaction

Ware *et al.* [ABW93, WAB93] were first to explore the effects of head pose input on the experience in 3D Virtual Environments. The authors adopted the term '*fish tank virtual reality*' to describe the use of a stereoscopic monitor supplemented with a mechanical head-tracking device for adjusting the perspective transformation to a user's eye position to simulate the appearance of 3D objects positioned behind or just in front of the screen. Ware *et al.* used this term because the experience is similar to looking at a fish tank inside which there is a virtual world. Fish tank VR was found to be of significant value compared to standard display techniques, with head coupling being more important than stereo vision [ABW93, WAB93].

Recent advances in face-tracking technology [MCT09] have made it possible to recognize head movements using a commodity web camera. In light of these advances, researchers have begun to study head gestures as input to interact in VEs. For example, Wang *et al.* [WXX*06] studied face tracking as a new axis of control in an FPS game. The authors compared camera-based video games with and without face tracking and demonstrated that using face position information can effectively enhance presence and role-playing. Yim *et al.* [YQG08] had similar results while studying head tracking in a 3D game built upon the popular work of Johnny Lee [Lee08]. Sko and Gardner [SG09] also investigated the potential for head

Figure 28: *A system for producing 3D animations using physical objects (i.e. puppets) as input [HGCA12]. Image courtesy of Maneesh Agrawala.*

gestural input to FPS games and have developed several diverse interaction techniques. Based on their experience derived from the studies, they draw a set of guidelines for the future development of head interaction techniques for computer games.

Silva and Bowman [SB09] investigate the benefits of adding body-based interaction to complement keyboard and mouse interaction in desktop gaming. They claim that body-based interaction (they use leaning to control navigation: leaning forward or backward results in moving in the same direction; leaning to the side rotates the character) in addition to keyboard and mouse can help players perform more tasks at the same time and can be especially attractive and helpful to new players. Terziman *et al.*'s [TME*10] apply the *Walking-In-Place* technique to the context of desktop VR. With their '*Shake-Your-Head*' technique, users can navigate using as sole input the head movements of the user. The experiment conducted to compare the technique with classical input devices used for navigating in desktop VR showed that '*Shake-Your-Head*' could allow for fast navigation when sitting, after a short learning session.

Mine *et al.* [MBS97] present a framework for VE interaction based on proprioception, a person's sense of the position and orientation of his body and limbs. The authors proposed several interaction techniques including scaled-world grab for both manipulation and locomotion as well as head-butt zoom and look-at menus involving head tracking. The following enormous advances in computer vision-based human motion capture and analysis (surveyed in [MG01, MHK06, Pop07]) allowed for novel interaction techniques based on tracking, pose estimation and understanding of human actions and behaviour. One of the results of these works was the development of very sophisticated and affordable motion-sensing input devices such as Microsoft's Kinect [SFC*11] capable of full-body motion sensing that truly set a new standard for motion control and opened new exciting possibilities for 3D interaction (e.g. an interface for 3D animation [HGCA12]—see Figure 28).

Figure 29: *An example of the brain–computer interface: controlling a virtual spaceship [LLR*08]. ©CNRS Photothéque/Hubert Raguet.*

7.3. Gaze-based interaction

Other interesting approaches include gaze-based interaction using specialized eye tracking equipment. Few research works explored the effects of eye-based input on the experience in 3D Virtual Environments. To our knowledge, Tanriverdi and Jacob first proposed eye movement as an active pointing device for 3D object selection in VEs presented in an HMD [TJ00]. Isokoski *et al.* [IM06] used an FPS game to assess the performance of eye gaze as an extra input modality to mouse and keyboard. Smith and Graham [SG06] compared the use of an eye tracker to a desktop mouse for game control. The authors utilized the input in different ways for, among others, an FPS shooter Quake 2, where the user controls orientation with either the mouse or the eyes and a role-playing game Neverwinter Nights, in which an avatar is moved through an interactive environment through pointing. Both Isokoski's and Smith and Graham's study revealed no advantage in performance with gaze. However, the reported subjective results showed that using eye gaze offered a much more immersive experience than using a mouse and keyboard. More recently, Istance *et al.* [IHVC09] used gaze input in different modes for emulating mouse and keyboard events appropriate for interacting with online 3D games. The authors report that it is feasible to carry out tasks representative of game play at a beginner's skill level using gaze alone. It is important to stress that these works on gaze-based game interaction are of huge importance for motor impaired users.

7.4. Brain–computer interaction (BCI)

BCIs are communication systems conveying messages through brain activity only, this activity being generally measured using ElectroEncephaloGraphy (EEG) [WBM*02, TN10]. Recent research results have shown that BCI can be used to interact with 3D virtual environments supporting navigation [LLR*08] as well as selection of 3D objects (simply by paying attention to these objects) and their manipulation [Bay03, LKF*]. Although these works prove that exploring a VE by brain activity is possible, the number of mental commands provided to the user is rather limited (e.g. turning left or right by imagining left or right hand movements, respectively). Nevertheless, in quite recent work, Lotte *et al.* [LvLL*10] proposed an asynchronous BCI that provide three different mental commands to the user and they showed that by using appropriate interaction techniques, only three commands were necessary to explore large and complex 3D VEs. As for our own work, we recently compared the BCI to the walking interface in VR focusing on the impact of motor activity on spatial transfer [LSA*12]; the results indicate that BCI is a promising interface for studying spatial cognition. For a detailed survey on using BCI for 3D interaction, see [LFG*13].

7.5. Display technology

Stereo and autostereoscopic display technology is breaking through in the mass markets of consumer electronics. As a matter of fact, as a result of a big push from the movie industry, currently it is hard to buy a new TV that does not offer some kind of stereo 3D display technology.

With regard to interaction with 3D interactive graphics, compared to traditional 2D displays, stereoscopic displays have shown to enhance performance on a variety of depth-related tasks including navigation as well as selection and manipulation of objects (e.g. [Ros93]). From a cognitive point of view, stereoscopy also proved to improve the spatial understanding of 3D information and help to conceptualize the space (e.g. [WF96]). As for some more recent works, LaViola and his colleagues, while evaluating the benefits of 3D stereo in modern video games, found out that although participants preferred playing in 3D stereo, it did not provide any significant advantage in overall user performance [LL11]. However, in their next study investigating whether user performance is enhanced when using 3D stereo over a traditional 2D monitor coupled with a 3D spatial interaction device in modern 3D stereo games, they reported a positive effect on gaming performance based on stereoscopic vision, although reserved to isolated tasks and depending on game expertise [KSL12].

On the other hand, stereoscopic 3D displays can cause problems including the simulator-sickness-type symptoms of eyestrain, headache, fatigue, disorientation and nausea, which appear to effect large numbers of viewers (perhaps as many as 25% to 50% of the general population) [MHG12]. Therefore, in their great review of human performance on stereoscopic 3D displays, McIntire *et al.* suggest that the technology should be wielded delicately and applied carefully (e.g. when it is necessary to ensure good performance).

8. Accessing 3D Graphics on the Web

The Web evolved from a text-based system to the current rich and interactive medium that supports images, 2D graphics, audio and video. The major media type that is still missing is 3D graphics. As computer graphics technology has reached the point where high quality 3D content can be rendered in real time on commodity desktop and mobile devices, it should be possible to use 3D models rather than 2D images to represent various objects and ultimately improve user experience on the Web.

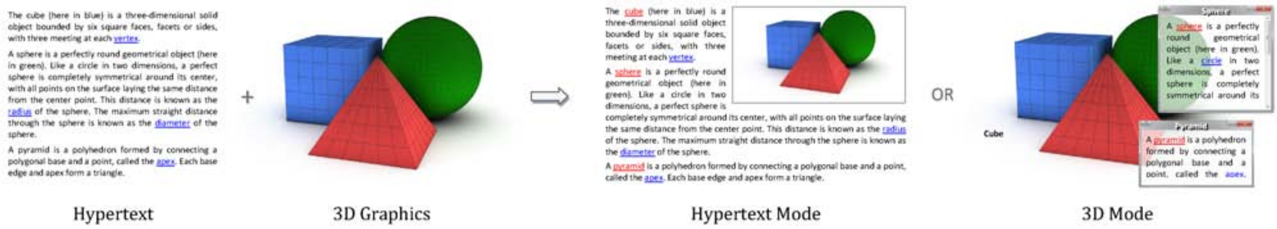

Figure 30: *The same information presented in the hypertext and the 3D mode of the dual-mode user interface [JD12], [JD13].*

There have been a number of approaches over the last years to integrate 3D technologies on the Web and most of these systems and standards disappeared or barely survived. We believe that this is because of the fact that research was focused mostly on 3D graphics and 3D graphics alone. The focus of research did not include the search for user interaction techniques that are optimal for the hypertext-based Web interface. However, what good is a realistic environment if one cannot interact with it? As a result, hypertext (the ultimate product in symbolic communication) and interactive 3D graphics (the ultimate achievement in visual media) are at odds on the Web.

8.1. Hypertextualized virtual environments

We believe that people can gain a lot from using integrated information spaces where hypertext and 3D graphics data are simultaneously available and linked, spaces we call Hypertextualized Virtual Environments (HiVEs) [Jan11a]. We focus on UI design that supports the integrated exploration of HiVEs; the design, where users can browse the text, look through general information and search for more specific information, and where they can also navigate freely through a 3D space, and examine and manipulate virtual 3D objects, to gain a better understanding of the data. Our aim is to pair interactive 3D graphics know-how with well-established UI conventions of the Web to support all these user tasks [JD09, Jan11b].

Most recently, we introduced a DMUI that has two modes between which a user can switch anytime [JD12, JD13]: the driven by simple hypertext-based interactions '*don't-make-me-think*' mode, where a 3D scene is embedded in hypertext and the more immersive 3D '*take-me-to-the-Wonderland*' mode, which immerses the hypertextual annotations into the 3D scene (see Figure 30). The results from the latest competitive user study suggest users performed better with DMUI over alternatives.

8.2. 3D web design guidelines

Based on our previous work, Nielsen's and Krug's work on Web usability [Nie00, NL06, Kru05], as well as observation of interfaces of popular 3D games, several works dedicated to game design (e.g. [Rou00]) and design guidelines for virtual environments [Kau98,Shn03], we summarize what we consider to be the main 3D Web design guidelines:

- *3D websites should explain themselves*: 3D web pages should be self-evident, obvious, self-explanatory. Users should be able to learn what they are and how to use them without expending any effort thinking about it. Remember: much of the Web use is motivated by the desire to save time, so do not waste people's time.
- *Text*: The primary goal of Web design is to convey information. Designers should choose typography that communicates. Keep text readable, give users control over explanatory text and let users select for details on demand.
- *Navigation*: Minimize the number of navigation steps for users to accomplish their tasks, simplify user movement (keep movements planar, use collision detection), allow teleportation (e.g. selecting a destination in an overview).
- *Wayfinding*: Provide overviews so users can see the big picture (e.g. maps), provide history keeping—people are more comfortable when they know that they can undo or alter their actions; Make it easy to go home (so users can start over); Highlight current location in navigational UI components; Allow local search.
- *Manipulation*: Provide users with manipulators, exploit knowledge about real world (e.g. gravity) and natural behaviour of objects (e.g. solidity).
- *Help*: Help should be embedded throughout the interface. It should allow users to click on any new UI component for an explanation.
- *People are creatures of habit*: If they find something that works, they stick to it; they tend not to look for a better way. Designers should use existing Web conventions.

We are convinced that 3D computer graphics and virtual environments have also the potential to support Web UIs. However, this potential requires investigation into the 3D nature of interaction in virtual environments that is suitable for internet users. We hope that the presented 3D Web usability principles may be useful for practitioners and researchers willing to work in the field of 3D Web UIs.

9. Conclusions

Ivan Sutherland, who is acknowledged by many to be the 'grandfather' of interactive Computer Graphics, Graphical UIs and HCI, in his work on the '*Ultimate Display*' [Sut65], demonstrated his vision of the 3D experience. He proposed a 3D interactive environment as the basis of HCI. According to his vision, the ultimate display would act as a looking glass into a mathematical wonderland. It would display information in 3D fashion, represented in an utterly believable virtual world.

The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked.

Ivan Sutherland [Sut65]

Three-dimensional computer graphics has managed to find its way into the lives of all of us. Expressing information through 3D visualization is bringing huge benefits for a variety of applications ranging from video games, through the latest special effects in movies, to serious applications (e.g. engineering or medical visualizations). From Sketchpad [Sut63] we reached the point, where unbelievably realistic high-quality 3D content can be rendered in real time on commodity desktop and mobile devices. In a sense, Sutherland's vision of the ultimate display, the ultimate VR system, is in the making.

Nevertheless, there is still a long way ahead of us and much work needs to be done. As mostly everyone, who has ever developed an application related to computer graphics has faced the problem of defining the appropriate user interaction methods, we developed this State-of-the-Art Report. We hope that this survey of advances in interaction can aid both researchers and developers of interactive 3D applications in having a clearer overview of the topic of Navigation, Selection & Manipulation and System Control, and help them to make the most appropriate choice from the jungle of available interaction techniques.

In this survey, we have focused on the most standard 3D interaction tasks allowing one to explore 3D environments or to manipulate objects. Other tasks may require more specific interaction techniques. Examples are sculpting operations (e.g. *pinch* and *inflate*), exploration of volumetric data or direct control of environmental parameters (e.g. light and material properties). Such interaction tasks are beyond the scope of this work, and numerous interaction techniques are still to be invented in these areas.

In this STAR, we have described a lot of techniques for interaction with 3D environments under various levels of control. For the techniques where the level of control is high, the interaction space is rich. On the other hand, such techniques require a high level of expertise to be used correctly. At the opposite, the techniques with low level of control (e.g. movements computed automatically by the system) fit well with novices users. On the other hand, the level of interactivity with such techniques is much reduced.

A big challenge in the design of interaction techniques is to enhance the level of interactivity, while limiting the level of control that is required. Sketch-based approaches, where users express coarse intent and let the system optimize the resulting actions in the 3D environment are definitively good examples of techniques that try to address this challenge. These approaches may be an answer to the general quest of opening interactive 3D environments to non-expert users.

Another big challenge in 3D interaction is to provide adaptive techniques. Indeed, a novice user will gain in expertise when multiplying the completion of a given interaction task. If the used technique is too simple, he or she will be limited quickly, which will cause frustration. Hence, adaptive 3D interaction techniques where the level of control adapts to the expertise of the user is definitively a promising research direction.

For a long time, the design of 3D interaction techniques has been mainly driven by criteria such as speed, efficiency or precision. Other criteria such as interface appeal and enjoyment arising from the interface should also be considered. Indeed, these criteria have been mostly neglected in academic research whereas we believe they are crucial for users to venture and persevere in the complex process that is interaction with 3D environments. An interface with a strong appeal and enjoyment factor will motivate these users to perform 3D tasks, and will thus stimulate activities such as learning, communication or creation.

Acknowledgements

This work has been supported by SFI under Grant No. SFI/08/CE/I1380 (Lion), by EI under Grant No. PC/2008/353 (Copernicus) and by the Villes Transparentes project.

References

[AA09] ARGELAGUET, F., ANDUJAR, C.: Efficient 3D pointing selection in cluttered virtual environments. *IEEE Computer Graphics and Applications 29* (2009), 34 – 43.

[AA13] ARGELAGUET, F., ANDUJAR, C.: A survey of 3D object selection techniques for virtual environments. *Computers & Graphics 37* (2013), 121–136.

[ABW93] ARTHUR, K.W., BOOTH, K.S., WARE, C.: Evaluating 3D task performance for fish tank virtual worlds. *ACM Transactions on Information Systems 11* (1993), 239–265.

[And05] ANDERS, K.-H.: Level of detail generation of 3D building groups by aggregation and typification. In *Proceedings of the XXII International Cartographic Conference* (La Coruna, Spain, 2005).

[AS01] AGRAWALA, M., STOLTE, C.: Rendering effective route maps: improving usability through generalization. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001). ACM, pp. 241–249.

[ATF12] AU, O.K.-C., TAI, C.-L., FU, H.: Multitouch gestures for constrained transformation of 3D objects. *Computer Graphics Forum 31* (2012), 651–660.

[AVG04] ANDUJAR, C., VAZQUEZ, P.-P., GONZALEZ, M.F.: Way-finder: Guided tours through complex walkthrough models. *Computer Graphics Forum 23* (2004), 499–508.

[Bal04] BALAKRISHNAN, R.: 'beating' fitts' law: Virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies 61* (2004), 857–874.

[Bay03] Bayliss, J.D.: Use of the evoked potential p3 component for control in a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering 11* (2003), 113–166.

[BBKF97] Balakrishnan, R., Baudel, T., Kurtenbach, G., Fitzmaurice, G.: The rockin'mouse: integral 3D manipulation on a plane. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1997), ACM, pp. 311–318.

[BBT*13] Brivio, P., Benedetti, L., Tarini, M., Ponchio, F., Cignoni, P., Scopigno, R.: Photocloud: Interactive remote exploration of large 2d-3d datasets. *IEEE Computer Graphics and Applications 33* (2013), 86–96.

[BC07] Burigat, S., Chittaro, L.: Navigation in 3D virtual environments: Effects of user experience and location-pointing navigation aids. *International Journal of Human-Computer Studies 65* (2007), 945–958.

[BGBS02] Baudisch, P., Good, N., Bellotti, V., Schraedley, P.: Keeping things in context: A comparative evaluation of focus plus context screens, overviews, and zooming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2002), ACM, pp. 259–266.

[BGH02] Bowman, D.A., Gabbard, J.L., Hix, D.: A survey of usability evaluation in virtual environments: Classification and comparison of methods. *Presence 11* (2002), 404–424.

[BGL98] Bares, W. H., Grégoire, J. P., Lester, J. C.: Realtime constraint-based cinematography for complex interactive 3D worlds. In *National Conference on Innovative Applications of Artificial Intelligence* (Madison, WI, USA, 1998), pp. 1101–1106.

[BH99] Bowman, D. A., Hodges, L. F.: Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages and Computing 10* (1999), 37–53.

[Bie87] Bier, E. A.: Skitters and jacks: Interactive 3D positioning tools. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics* (New York, NY, USA, 1987), ACM, pp. 183–196.

[Bie90] Bier, E. A.: Snap-dragging in three dimensions. In *Proceedings of the 1990 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1990), ACM, pp. 193–204.

[BJH99] Bowman, D. A., Johnson, D. B., Hodges, L. F.: Testbed evaluation of virtual environment interaction techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 1999), ACM, pp. 26–33.

[BK99] Balakrishnan, R., Kurtenbach, G.: Exploring bimanual camera control and object manipulation in 3D graphics interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1999), ACM, pp. 56–62.

[BKB13] Bacim, F., Kopper, R., Bowman, D.: Design and evaluation of 3D selection techniques based on progressive refinement. *International Journal of Human-Computer Studies 71* (2013), 785–802.

[BKF*02] Burtnyk, N., Khan, A., Fitzmaurice, G., Balakrishnan, R., Kurtenbach, G.: Stylecam: Interactive stylized 3d navigation using integrated spatial and temporal controls. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2002), ACM, pp. 101–110.

[BKFK06] Burtnyk, N., Khan, A., Fitzmaurice, G., Kurtenbach, G.: Showmotion: Camera motion based 3D design review. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2006), ACM, pp. 167–174.

[BKH97] Bowman, D. A., Koller, D., Hodges, L. F.: Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In *Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)* (Washington, DC, USA, 1997), IEEE Computer Society, p. 45.

[BKH98] Bowman, D. A., Koller, D., Hodges, L. F.: A methodology for the evaluation of travel techniques for immersive virtual environments. *Virtual Reality 3* (1998), 120–131.

[BKLP01] Bowman, D. A., Kruijff, E., LaViola, J. J., Poupyrev, I.: An introduction to 3-D user interface design. *Presence 10* (2001), 96–108.

[BKLP04] Bowman, D. A., Kruijff, E., LaViola, J. J., Poupyrev, I.: *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc, Redwood City, CA, USA, 2004.

[BL99] Bares, W. H., Lester, J. C.: Intelligent multi-shot visualization interfaces for dynamic 3D worlds. In *Proceedings of the 4th International Conference on Intelligent User Interfaces* (New York, NY, USA, 1999), ACM, pp. 119–126.

[BM91] Buttenfield, B. P., McMaster, R. B.: *Map Generalization: Making Rules for Knowledge Representation*. Longman Scientific & Technical, New York, 1991.

[BNC*03] Bowman, D. A., North, C., Chen, J., Polys, N. F., Pyla, P. S., Yilmaz, U.: Information-rich virtual environments: Theory, tools, and research agenda. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2003), ACM, pp. 81–90.

[Bol80] Bolt, R. A.: 'put-that-there': Voice and gesture at the graphics interface. *ACM SIGGRAPH Computer Graphics 14* (1980), 262–270.

[BRP05] Bade, R., Ritter, F., Preim, B.: Usability comparison of mouse-based interaction techniques for predictable 3D rotation. In *Proceedings of the 5th International Conference on Smart Graphics* (Berlin, Heidelberg, 2005), Springer, pp. 138–150.

[Bru01] Brusilovsky, P.: Adaptive Hypermedia. *User Modeling and User-Adapted Interaction 11* (2001), 87–110.

[BS95] Bukowski, R. W., Séquin, C. H.: Object associations: A simple and practical approach to virtual 3D manipulation. In

Proceedings of the 1995 Symposium on Interactive 3D Graphics (New York, NY, USA, 1995), ACM, p. 131.

[BTM00] BARES, W. H., THAINIMIT, S., MCDERMOTT, S.: A model for constraint-based camera planning. In *Papers from the 2000 AAAI Spring Symposium Smart Graphics* (2000), pp. 84–91.

[Cab14] CABELLO, R.: Three.js. http://threejs.org/, 2014. Accessed 25 September 2014.

[CAH*96] CHRISTIANSON, D. B., ANDERSON, S. E., HE, L.-W., SALESIN, D. H., WELD, D. S., COHEN, M. F.: Declarative camera control for automatic cinematography. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence volume 1* (Portland, OR, USA, 1996), AAAI Press, pp. 148–155.

[CB04] CHITTARO, L., BURIGAT, S.: 3D location-pointing as a navigation aid in virtual environments. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA), ACM, pp. 267–274

[CDH11] COHÉ, A., DÈCLE, F., HACHET, M.: tbox: A 3d transformation widget designed for touch-screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), ACM, pp. 3005–3008.

[CI04] Chittaro L., Ieronutti L.: A visual tool for tracing users' behavior in virtual environments. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA), ACM, pp. 40–47.

[CKB08] COCKBURN, A., KARLSON, A., BEDERSON, B. B.: A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys 41* (2008), 2:1–2:31.

[CLP04] CARPENDALE, S., LIGH, J., PATTISON, E.: Achieving higher magnification in context. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2004), ACM, pp. 71–80.

[CMS88] CHEN, M., MOUNTFORD, S. J., SELLEN, A.: A study in interactive 3-D rotation using 2-D control devices. *SIGGRAPH Computer Graphics 22* (1988), 121–129.

[CNP04] CELENTANO, A., NODARI, M., PITTARELLO, F.: Adaptive interaction in web3d virtual worlds. In *Proceedings of the Ninth International Conference on 3D Web Technology* (New York, NY, USA, 2004), ACM, pp. 41–50.

[CON08] CHRISTIE, M., OLIVIER, P., NORMAND, J.-M.: Camera control in computer graphics. *Computer Graphics Forum 27* (2008), 2197–2218.

[CPB04] CHEN, J., PYLA, P. S., BOWMAN, D. A.: Testbed evaluation of navigation and text display techniques in an information-rich virtual environment. In *Virtual Reality* (2004), IEEE, pp. 181–289.

[CR02] CHITTARO, L., RANON, R.: Dynamic generation of personalized vrml content: A general approach and its application to 3D e-commerce. In *Proceedings of the Seventh International Conference on 3D Web Technology* (New York, NY, USA, 2002), ACM, pp. 145–154.

[CR07] CHITTARO, L., RANON, R.: Adaptive 3D web sites. In *The Adaptive Web*. P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.). Springer-Verlag, Berlin, Heidelberg , 2007, pp. 433–462.

[CRI03] CHITTARO, L., RANON, R., IERONUTTI, L.: Guiding visitors of web3D worlds through automatically generated tours. In *Proceedings of the Eighth International Conference on 3D Web Technology* (New York, NY, USA, 2003), ACM, pp. 27–38.

[CRI06] CHITTARO, L., RANON, R., IERONUTTI, L.: Vu-flow: A visualization tool for analyzing navigation in virtual environments. *IEEE Transactions on Visualization and Computer Graphics 12* (2006), 1475–1485.

[CRY96] CARD, S. K., ROBERTSON, G. G., YORK, W.: The webbook and the web forager: An information workspace for the world-wide web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1996), ACM, p. 111.

[CS01] CHITTARO, L., SCAGNETTO, I.: Is semitransparency useful for navigating virtual environments? In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2001), ACM, pp. 159–166.

[CSH*92] CONNER, B. D., SNIBBE, S. S., HERNDON, K. P., ROBBINS, D. C., ZELEZNIK, R. C., VAN DAM, A.: Three-dimensional widgets. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1992), ACM, pp. 183–188.

[CV06] CHITTARO, L., VENKATARAMAN, S.: Navigation aids for multi-floor virtual buildings: A comparative evaluation of two approaches. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2006), ACM, pp. 227–235.

[CWL12] CASHION, J., WINGRAVE, C., LAVIOLA, J.: Dense and dynamic 3d selection for game-based virtual environments. *IEEE Transactions on Visualization and Computer Graphics 18* (2012), 634–642.

[DB05] DÖLLNER, J., BUCHHOLZ, H.: Expressive virtual 3D city models. In *International Cartographic Conference* (2005).

[DBNK05] DÖLLNER, J., BUCHHOLZ, H., NIENHAUS, M., KIRSCH, F.: Illustrative visualization of 3D city models. In *Visualization and Data Analysis* . R. F. Erbacher, J. C. Roberts, M. T. Grohn, K. Borner (Eds.). International Society for Optical Engine (SPIE), San Jose, CA, USA, 2005, pp. 42–51.

[DCT04] DESURVIRE, H., CAPLAN, M., TOTH, J. A.: Using heuristics to evaluate the playability of games. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2004), ACM, pp. 1509–1512.

[DER*13] DELIKOSTIDIS, I., ENGEL, J., RETSIOS, B., VAN ELZAKKER, C. P., KRAAK, M.-J., DÖLLNER, J.: Increasing the usability of pedestrian navigation interfaces by means of landmark visibility analysis. *The Journal of Navigation 66* (2013), 532–537.

[DGZ92] Drucker, S. M., Galyean, T. A., Zeltzer, D.: Cinema: A system for procedural camera movements. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1992), ACM, pp. 67–70.

[DH05] Dachselt, R., Hinz, M.: Three-dimensional widgets revisited: Towards future standardization. In *IEEE VR 2005 Workshop New Directions in 3D User Interfaces* (2005).

[DH07] Dachselt, R., Hübner, A.: Three-dimensional menus: A survey and taxonomy. *Computers & Graphics 31* (2007), 53–65.

[dHKP05] de Haan, G., Koutek, M., Post, F. H.: Intenselect: Using dynamic object rating for assisting 3D object selection. In *Proceeding of the 11th Eurographics Conference on Virtual Environment* (Aire-la-ville, Switzerland, Switzerland, 2005), Eurographics Association.

[DS93] Darken, R. P., Sibert, J. L.: A toolset for navigation in virtual environments. In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1993), ACM, pp. 157–165.

[DS96] Darken, R. P., Sibert, J. L.: Wayfinding strategies and behaviors in large virtual worlds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1993), ACM, pp. 142–149.

[DZ94] Drucker, S. M., Zeltzer, D.: Intelligent camera control in a virtual environment. In *Proceedings of Graphics Interface* 94 (1994), pp. 190–199.

[DZ95] Drucker, S. M., Zeltzer, D.: Camdroid: A system for implementing intelligent camera control. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1995), ACM, pp. 139–144.

[EAT07] Elmqvist, N., Assarsson, U., Tsigas, P.: Employing dynamic transparency for 3d occlusion management: Design issues and evaluation. In *INTERACT* (2007).

[EB05] Elias, B., Brenner, C.: Automatic generation and application of landmarks in navigation data sets. In *Developments in Spatial Data Handling*. Springer, Berlin, Heidelberg, 2005, pp. 469–480.

[EE68] Engelbart, D. C., English, W. K.: A research center for augmenting human intellect. In *Fall Joint Computer Conference* (1968).

[EF08] Elmqvist, N., Fekete, J.: Semantic pointing for object picking in complex 3D environments. In Proceedings of Graphics Interface 2008 (Toronto, ON, Canada, 2008), Canadian Information Processing Society, pp. 243–250.

[EGT10] Eno, J., Gauch, S., Thompson, C. W.: Linking behavior in a virtual world environment. In *Proceedings of the 15th International Conference on Web 3D Technology* (New York, NY, USA, 2010), ACM, pp. 157–164.

[EH97] Edwards, J. D. M., Hand, C.: Maps: Movement and planning support for navigation in an immersive vrml browser. In *Proceedings of the Second Symposium on Virtual Reality Modeling Language* (New York, NY, USA, 1997), ACM, p. 65.

[Eli03] Elias, B.: Extracting landmarks with data mining methods. In *Spatial Information Theory. Foundations of Geographic Information Science*. W. Kuhn, M. F. Worboys, S. Timpf (Eds.). Springer, Berlin, Heidelberg, 2003, pp. 375–389.

[ENK97] Elvins, T. T., Nadeau, D. R., Kirsh, D.: Worldlets: 3D thumbnails for wayfinding in virtual environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1997), ACM, pp. 21–30.

[ENSK98] Elvins, T. T., Nadeau, D. R., Schul, R., Kirsh, D.: Worldlets: 3D thumbnails for 3d browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1998), ACM, pp. 163–170.

[ETT07] Elmqvist, N., Tudoreanu, M. E., Tsigas, P.: Tour generation for exploration of 3D virtual environments. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2007), ACM, pp. 207–210.

[ETT08] Elmqvist, N., Tudoreanu, M. E., Tsigas, P.: Evaluating motion constraints for 3D wayfinding in immersive and desktop virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), ACM/Addison-Wesley Publishing Co., pp. 1769–1778.

[Fit54] Fitts, P.: The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology 47* (1954), 381.

[FMM*08] Fitzmaurice, G., Matejka, J., Mordatch, I., Khan, A., Kurtenbach, G.: Safe 3D navigation. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2008), ACM, pp. 7–15.

[For07] Forberg, A.: Generalization of 3D building data based on a scale-space approach. *ISPRS Journal of Photogrammetry and Remote Sensing 62* (2007), 104 –111.

[FP00] Fröhlich, B., Plate, J.: The cubic mouse: A new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2000), ACM/Addison-Wesley Publishing Co., pp. 526–531.

[Fur86] Furnas, G. W.: Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1986), ACM, pp. 16–23.

[FvDFH90] Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F.: *Computer Graphics: Principles and Practice* (2nd edition). Addison-Wesley, Boston, MA, USA, 1990.

[Gab97] GABBARD, J. L.: *A Taxonomy of Usability Characteristics in Virtual Environments*. Master's thesis, Virginia Polytechnic Institute and State University, 1997.

[Gal95] GALYEAN, T. A.: Guided navigation of virtual environments. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1995). ACM, p. 103.

[GB05] GROSSMAN, T., BALAKRISHNAN, R.: The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2005), ACM, pp. 281–290.

[GD07] GLANDER, T., DÖLLNER, J.: Cell-based generalization of 3D building groups with outlier management. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems* (New York, NY, USA, 2007), ACM, pp. 54:1–54:4.

[GD08a] GLANDER, T., DÖLLNER, J.: Automated cell based generalization of virtual 3D city models with dynamic landmark highlighting. In *11th ICA Workshop on Generalization and Multiple Representation* (2008).

[GD08b] GLANDER, T., DÖLLNER, J.: Techniques for generalizing building geometry of complex virtual 3D city models. In *Advances in 3D Geoinformation Systems*. P. van Oosterom, S. Zlatanova, F. Penninga, E. M. Fendel (Eds.), Springer, pp. 381–400.

[GGSC98] GOOCH, A., GOOCH, B., SHIRLEY, P., COHEN, E.: A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), ACM, pp. 447–452.

[GHS99] GABBARD, J. L., HIX, D., SWAN, J. E.: User-centered design and evaluation of virtual environments. *IEEE Computer Graphics and Applications 19* (1999), 51–59.

[GMS06] GRAMMENOS, D., MOUROUZIS, A., STEPHANIDIS, C.: Virtual prints: Augmenting virtual environments with interactive personal marks. *International Journal of Human-Computer Studies 64* (2006), 221–239.

[GPTD09] GLANDER, T., PETERS, D., TRAPP, M., DÖLLNER, J.: 3d wayfinding choremes: A cognitively motivated representation of route junctions in virtual environments. In *AGILE International Conference on GI Science* (2009).

[GS99] GOESELE, M., STUERZLINGER, W.: Semantic constraints for scene manipulation. In *Spring Conference in Computer Graphics* (1999).

[GTD07] GLANDER, T., TRAPP, M., DÖLLNER, J.: A concept of effective landmark depiction in geovirtual 3D environments by view-dependent deformation. In *4th International Symposium on LBS and Telecartography* (2007).

[GW92] GLEICHER, M., WITKIN, A.: Through-the-lens camera control. *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1992). ACM, pp. 331–340.

[Han97] HAND, C.: A survey of 3D interaction techniques. *Computer Graphics Forum 16* (1997), 269–281.

[HBCdlR11] HACHET, M., BOSSAVIT, B., COHÉ, A., DE LA RIVIÈRE, J.-B.: Toucheo: Multitouch and stereo combined in a seamless workspace. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2011). ACM, pp. 587–592.

[HBL02] HUGHES, S., BRUSILOVSKY, P., LEWIS, M.: Adaptive navigation support in 3D ecommerce activities. In *Workshop on Recommendation and Personalization in eCommerce at AH'2002* (2002), pp. 132–139.

[HBP02] HORNBAEK, K., BEDERSON, B. B., PLAISANT, C.: Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Transactions on Computer-Human Interaction 9* (2002), 362–389.

[HCC07] HANCOCK, M., CARPENDALE, S., COCKBURN, A.: Shallow-depth 3D interaction: Design and evaluation of one-, two- and three-touch techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), ACM, pp. 1147–1156.

[HCS96] HE, L.-w., COHEN, M. F., SALESIN, D. H.: The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1996), ACM, pp. 217–224.

[HD08] HAGEDORN, B., DÖLLNER, J.: Sketch-based navigation in 3D virtual environments. In *Smart Graphics*. A. Butz, B. Fisher, A. Krüger, P. Olivier, M. Christie (Eds.). Springer, Berlin, Heidelberg, 2008, pp. 239–246.

[HDG06] HACHET, M., DECLE, F., GUITTON, P.: Z-goto for efficient navigation in 3D environments from discrete inputs. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2006). ACM, pp. 236–239.

[HDKG08] HACHET, M., DECLE, F., KNODEL, S., GUITTON, P.: Navidget for easy 3D camera positioning from 2D inputs. In *IEEE Symposium on 3D User Interfaces* (*3DUI*) (2008), IEEE, pp. 83–89.

[HDKG09] HACHET, M., DECLE, F., KNÖDEL, S., GUITTON, P.: Navidget for 3D interaction: Camera positioning and further uses. *International Journal of Human-Computer Studies 67* (2009), 225–236.

[HGCA12] HELD, R., GUPTA, A., CURLESS, B., AGRAWALA, M.: 3D puppetry: A kinect-based interface for 3D animation. In *Proceedings of the 25th Annual ACM Symposium on User Interface*

Software and Technology (New York, NY, USA, 2012), ACM, pp. 423–434.

[HHC96] HUTCHINSON, S., HAGER, G., CORKE, P.: A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation 12* (1996), 651–670.

[HHS01] HALPER, N., HELBING, R., STROTHOTTE, T.: A camera engine for computer games: Managing the trade-off between constraint satisfaction and frame coherence. *Computer Graphics Forum 20* (2001), 174–183.

[Hou92] HOUDE, S.: Iterative design of an interface for easy 3-D direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1992), ACM, pp. 135–142.

[HSH*99] HINCKLEY, K., SINCLAIR, M., HANSON, E., SZELISKI, R., CONWAY, M.: The videomouse: A camera-based multi-degree-of-freedom input device. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1999), ACM, pp. 103–112.

[HtCC09] HANCOCK, M., TEN CATE, T., CARPENDALE, S.: Sticky tools: Full 6d of force-based interaction for multi-touch tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2009), ACM, pp. 133–140.

[HTP*97] HINCKLEY, K., TULLIO, J., PAUSCH, R., PROFFITT, D., KASSELL, N.: Usability analysis of 3D rotation techniques. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1997), ACM, pp. 1–10.

[HW97] HANSON, A. J., WERNERT, E. A.: Constrained 3D navigation with 2D controllers. In *Proceedings of the 8th Conference on Visualization '97* (Los Alamitos, CA, USA, 1997), IEEE Computer Society Press, p. 175.

[HWAT13] HIRONO, D., WU, H.-Y., ARIKAWA, M., TAKAHASHI, S.: Constrained optimization for disoccluding geographic landmarks in 3D urban maps. In *Proceedings of PacificVis 2013* (2013), pp. 17–24.

[HZR*92] HERNDON, K. P., ZELEZNIK, R. C., ROBBINS, D. C., CONNER, D. B., SNIBBE, S. S., VAN DAM, A.: Interactive shadows. In *Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1992), ACM, pp. 1–6.

[IHVC09] ISTANCE, H. O., HYRSKYKARI, A., VICKERS, S., CHAVES, T.: For your eyes only: Controlling 3d online games by eye-gaze. In *Human-Computer Interaction-INTERACT* 2009. T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. O. Prates, M. Winckler (Eds.). Springer, Berlin, Heidelberg, 2009, pp. 314–327.

[IKMT98] IGARASHI, T., KADOBAYASHI, R., MASE, K., TANAKA, H.: Path drawing for 3D walkthrough. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1998), ACM, pp. 173–174.

[IM06] ISOKOSKI, P., MARTIN, B.: Eye tracker input in first person shooter games. In *Communication by Gaze Interaction* (2006).

[JAL12] JÁUREGUI, D. A. G., ARGELAGUET, F., LÉCUYER, A.: Design and evaluation of 3D cursors and motion parallax for the exploration of desktop virtual environments. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)* (2012), IEEE, pp. 69–76.

[Jan11a] JANKOWSKI, J., : *Hypertextualized Virtual Environments*. PhD thesis, DERI NUIG, Galway, Ireland, 2011.

[Jan11b] JANKOWSKI, J.: A taskonomy of 3D web use. In *Proceedings of the 16th International Conference on 3D Web Technology* (New York, NY, USA, 2011), ACM, pp. 93–100.

[Jan12] JANKOWSKI, J.: Evaluation of static vs. animated views in 3D web user interfaces. In *Proceedings of the 17th International Conference on 3D Web Technology* (New York, NY, USA, 2012), ACM, p. 182.

[Jan14a] JANKOWSKI, J.: 3D city building blocks. https://github.com/grey-eminence/3DCityBB, 2014. Accessed 25 September 2014.

[Jan14b] JANKOWSKI, J.: 3dit: 3D interaction testbed. https://github.com/grey-eminence/3DIT, 2014. Accessed 25 September 2014.

[Jan14c] JANKOWSKI, J.: How to contribute to 3D interaction testbed.https://github.com/grey-eminence/3DIT/blob/master/CONTRIBUTING.md, 2014. Accessed 25 September 2014.

[JD09] JANKOWSKI, J., DECKER, S.: 2lip: Filling the gap between the current and the three-dimensional web. In *Proceedings of the 14th International Conference on 3D Web Technology* (New York, NY, USA, 2009), ACM, pp. 181–190.

[JD12] JANKOWSKI, J., DECKER, S.: A dual-mode user interface for accessing 3D content on the world wide web. In *Proceedings of the 21st International Conference on 3D Web Technology* (New York, NY, USA, 2012), ACM, pp. 1047–1056.

[JD13] JANKOWSKI, J., DECKER, S.: On the design of a dual-mode user interface for accessing 3D content on the world wide web. *International Journal of Human-Computer Studies 71* (2013), 7–8.

[JGH*08] JACOB, R. J., GIROUARD, A., HIRSHFIELD, L. M., HORN, M. S., SHAER, O., SOLOVEY, E. T., ZIGELBAUM, J.: Reality-based interaction: a framework for post-wimp interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), ACM, pp. 201–210.

[JH13] JANKOWSKI, J., HACHET, M.: A survey of interaction techniques for interactive 3D environments. In *EG STAR Reports* (2013).

[JH14] JANKOWSKI, J., HACHET, M.: A study of street-level navigation techniques in 3D digital cities on mobile touch devices. In *IEEE Symposium on 3D User Interfaces (3DUI)* (2014), pp. 35–38.

[Kau98] Kaur K., : *Designing Virtual Environments for Usability*. PhD thesis, City University London, 1998.

[KB94] Kurtenbach, G., Buxton, W.: User learning and performance with marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1994), ACM, pp. 258–264.

[KBB11] Kopper R., Bacim F., Bowman D.: Rapid and accurate 3D selection by progressive refinement. In *2011 IEEE Symposium on3D User Interfaces (3DUI)* (2011), IEEE, pp. 67–74.

[KGMQ08] Kim, J.-S., Gračanin, D., Matković, K., Quek, F.: Finger walking in place (FWIP): A traveling technique in virtual environments. In *Smart Graphics*. A. Butz, B. Fisher, A. Krüger, P. Olivier, M. Christie (Eds.). Springer, Berlin, Heidelberg, pp. 58–69.

[KH11] Knoedel, S., Hachet, M.: Multi-touch rst in 2D and 3D spaces: Studying the impact of directness on user performance. In *Proceedings of the 2011 IEEE Symposium on 3D User Interfaces* (Washington, DC, USA, 2011), IEEE Computer Society, pp. 75–78.

[KHG09] Knödel, S., Hachet, M., Guitton, P.: Interactive generation and modification of cutaway illustrations for polygonal models. In *Proceedings of the 10th International Symposium on Smart Graphics* (Berlin, Heidelberg, 2009), Springer, pp. 140–151.

[KHKF09] Kulik, A., Hochstrate, J., Kunert, A., Froehlich, B.: The influence of input device characteristics on spatial perception in desktop-based 3D applications. In *IEEE Symposium on 3D User Interfaces (3DUI)* (2009), IEEE, pp. 59–66.

[Khr10] Khronos: Webgl. http://www.khronos.org/webgl/, 2010. Accessed 25 September 2014.

[KKS*05] Khan, A., Komalo, B., Stam, J., Fitzmaurice, G., Kurtenbach, G.: Hovercam: Interactive 3D navigation for proximal object inspection. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2005), ACM, pp. 73–80.

[KMF*08] Khan, A., Mordatch, I., Fitzmaurice, G., Matejka, J., Kurtenbach, G.: Viewcube: A 3d orientation indicator and controller. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2008), ACM, pp. 17–25.

[Kru05] Krug, S.: *Don't Make Me Think: A Common Sense Approach to the Web* (2nd Edition). Pearson Education India, 2005.

[KSL12] Kulshreshth, A., Schild, J., LaViola, Jr. J. J.: Evaluating user performance in 3D stereo and motion enabled video games. In *Proceedings of the International Conference on the Foundations of Digital Games* (New York, NY, USA, 2012), ACM, pp. 33–40.

[LAFT12] Liu, J., Au, O. K.-C., Fu, H., Tai, C.-L.: Two-finger gestures for 6dof manipulation of 3D objects. *Computer Graphics Forum 31* (2012), 2047–2055.

[Lat91] Latombe, J.-C.: *Robot Motion Planning*. 1991.

[LBHD06] Lécuyer, A., Burkhardt, J.-M., Henaff, J.-M., Donikian, S.: Camera motions improve the sensation of walking in virtual environments. In *Virtual Reality Conference* (2006), IEEE, pp. 11–18.

[LC08] Li, T.-Y., Cheng, C.-C.: Real-time camera planning for navigation in virtual environments. In *SG* (2008).

[LCS05] Looser, J., Cockburn, A., Savage, J.: On the validity of using first-person shooters for fitts' law studies. In *People and Computers XIX 2* (2005), 33–36.

[Lee08] Lee, J. C.: Hacking the nintendo wii remote. *IEEE Pervasive Computing 7* (2008), 39–45.

[LFG*13] Lotte, F., Faller, J., Guger, C., Renard, Y., Pfurtscheller, G., Lécuyer, A., Leeb, R.: Combining BCI with virtual reality: Towards new applications and improved BCI. In Towards Practical Brain-Computer Interfaces. B. Z. Allison, S. Dunne, R. Leeb, J. Del R. Millán, José, A. Nijholt (Eds.). Springer, Berlin, Heidelberg, 2013, pp. 197–220.

[LI05] Lucas III J. F., : *Design and Evaluation of 3D Multiple Object Selection Techniques*. PhD thesis, Virginia Polytechnic Institute and State University, 2005.

[LKF*] Lalor, E. C., Kelly, S. P., Finucane, C., Burke, R., Smith, R., Reilly, R. B., Mcdarby, G.: Steady-state vep-based brain-computer interface control in an immersive 3D gaming environment. *EURASIP Journal on Applied Signal Processing 2005* (2005), 3156–3164.

[LL11] LaViola, Jr. J. J., Litwiller, T.: Evaluating the benefits of 3D stereo in modern video games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), ACM, pp. 2345–2354.

[LLCY99] Li, T.-Y., Lien, J.-M., Chiu, S.-Y., Yu, T.-H.: Automatically generating virtual guided tours. In *Proceedings of Computer Animation* (1999), pp. 99–106.

[LLR*08] Lécuyer, A., Lotte, F., Reilly, R. B., Leeb, R., Hirose, M., Slater, M.: Brain-computer interfaces, virtual reality, and videogames. *IEEE Computer 41* (2008), 66–72.

[LSA*12] Larrue, F., Sauzéon, H., Aguilova, L., Lotte, F., Hachet, M., NKaoua, B.: Brain computer interface vs walking interface in VR: The impact of motor activity on spatial transfer. In *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2012), ACM, pp. 113–120.

[LvLL*10] Lotte, F., van Langhenhove, A., Lamarche, F., Ernest, T., Renard, Y., Arnaldi, B., Lécuyer, A.: Exploring large virtual environments by thoughts using a brain–computer interface based on motor imagery and high-level commands. *Presence 19* (2010), 54–70.

[Lyn60] Lynch, K.: *The Image of the City*. MIT press, 1960.

[Mac92] MacKenzie, S.: Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction 7* (1992), 91–139.

[Mac04] MacEachren A. M., : *How Maps Work: Representation, Visualization, and Design*. The Guilford Press, 2004.

[MBS97] Mine, M. R., Brooks, Jr. F. P., Sequin, C. H.: Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 19–26.

[MC02] Marchand, É., Courty, N.: Controlling a camera in a virtual environment. *The Visual Computer 18* (2002), 1–19.

[MCG10a] Martinet, A., Casiez, G., Grisoni, L.: The design and evaluation of 3D positioning techniques for multi-touch displays. In *IEEE Symposium on 3D User Interfaces (3DUI)* (2010), IEEE, pp. 115–118.

[MCG10b] Martinet, A., Casiez, G., Grisoni, L.: The effect of DoF separation in 3D manipulation tasks with multi-touch displays. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, *2010*), ACM, pp. 111–118.

[MCR90] Mackinlay, J. D., Card, S. K., Robertson, G. G.: Rapid controlled movement through a virtual 3D workspace. *SIGGRAPH Computer Graphics 24* (1990), 171–176.

[MCT09] Murphy-Chutorian, E., Trivedi, M. M.: Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31* (2009), 607–626.

[MDWK08] Möser, S., Degener, P., Wahl, R., Klein, R.: Context aware terrain visualization for wayfinding and navigation. *Computer Graphics Forum 27* (2008), 1853–1860.

[MF07] Meng, L., Forberg, A.: 3D building generalisation. *Generalisation of Geographic Information: Cartographic Modelling and Applications*, Elsevier , Amsterdam, 2007, pp. 211–232.

[MG01] Moeslund, T. B., Granum, E.: A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding 81* (2001), 231–268.

[MGG*10] McCrae, J., Glueck, M., Grossman, T., Khan, A., Singh, K.: Exploring the design space of multiscale 3D orientation. In *Proceedings of the International Conference on Advanced Visual Interfaces* (New York, NY, USA, 2010), ACM, pp. 81–88.

[MHG12] McIntire, J. P., Havig, P. R., Geiselman, E. E.: What is 3D good for? A review of human performance on stereoscopic 3D displays. In *SPIE*(2012).

[MHK06] Moeslund, T. B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding 104* (2006), 90–126.

[Min95] Mine M. R., : *Virtual Environment Interaction Techniques*. Tech. rep., University of North Carolina 1995.

[MMG12] Moerman, C., Marchal Damien, C. P. M., Grisoni, L.: Drag'n Go: Simple and fast navigation in virtual environment. In *IEEE Symposium on 3D User Interfaces (3DUI)* (2012), IEEE, pp. 15–18.

[MMGK09] McCrae, J., Mordatch, I., Glueck, M., Khan, A.: Multiscale 3D navigation. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2009), ACM, pp. 7–14.

[Mos09] Moscovich, T.: Contact area interaction with sliding widgets. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2009), ACM, pp. 13–22.

[MSWM02] McNeill, M., Sayers, H., Wilson, S., McKevitt, P.: A spoken dialogue system for navigation in non-immersive virtual environments. *Computer Graphics Forum 21* (2002), 713–722.

[MT97] Möller, T., Trumbore, B.: Fast, minimum storage ray-triangle intersection. *Journal of Graphical Tools 2* (1997), 21–28.

[NBC06] Ni, T., Bowman, D. A., Chen, J.: Increased display size and resolution improve task performance in information-rich virtual environments. In *Proceedings of Graphics Interface* (Toronto, ON, Canada, 2006), Canadian Information Processing Society, pp. 139–146.

[Nie00] Nielsen, J.: *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, 2000.

[NL06] Nielsen, J., Loranger, H.: *Prioritizing Web Usability*. New Riders Publishing, 2006.

[NM90] Nielsen, J., Molich, R.: Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1990), ACM, pp. 249–256.

[NM94] Nielsen, J., Mack, R. L.: *Usability Inspection Methods*. Wiley, 1994.

[NO87] Nielson, G. M., Olsen, Jr. D. R.: Direct manipulation techniques for 3D objects using 2D locator devices. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics* (New York, NY, USA, 1987), ACM, pp. 175–182.

[NO03] Nieuwenhuisen D., Overmars M. H., : *Motion Planning for Camera Movements in Virtual Environments*. Tech. rep., Utrecht University, 2003.

[Nor90] Norman D. A., : *The Design of Everyday Things*. Basic books, New York, 1990.

[OS05] Oh, J.-Y., Stuerzlinger, W.: Moving objects with 2D input devices in CAD systems and desktop virtual environments. In *Proceedings of Graphics Interface* (Waterloo, ON, Canada, 2005), Canadian Human-Computer Communications Society, pp. 195–202.

[OSD06] Oh, J.-Y., Stuerzlinger, W., Dadgari, D.: Group selection techniques for efficient 3D modeling. In *Proceedings of the IEEE Conference on Virtual Reality* (Washington, DC, USA, 2006), IEEE Computer Society, p. 130.

[OSTG09] Oskam, T., Sumner, R. W., Thuerey, N., Gross, M.: Visibility transition planning for dynamic camera control. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2009), ACM, pp. 55–65.

[Par12] Parent R., : *Computer Animation: Algorithms and Techniques*. Newnes, 2012.

[PB88] Phillips, C. B., Badler, N. I.: Jack: A toolkit for manipulating articulated figures. In *Proceedings of the 1st Annual ACM SIGGRAPH Symposium on User Interface Software* (New York, NY, USA, 1988), ACM, pp. 221–229.

[PBG92] Phillips, C. B., Badler, N. I., Granieri, J. P.: Automatic viewing control for 3D direct manipulation. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1992), ACM, pp. 71–74.

[PBN11] Polys, N. F., Bowman, D. A., North, C.: The role of depth and gestalt cues in information-rich virtual environments. *International Journal of Human Computer Studies 69* (2011), 30–51.

[PCS95] Plaisant, C., Carr, D., Shneiderman, B.: Image-browser taxonomy and guidelines for designers. *IEEE Software 12* (1995), 21–32.

[PCvDR99] Pierce, J. S., Conway, M., van Dantzich, M., Robertson, G.: Toolspaces and glances: Storing, accessing, and retrieving objects in 3d desktop applications. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1999), ACM, pp. 163–168.

[PKB05] Polys, N. F., Kim, S., Bowman, D. A.: Effects of information layout, screen size, and field of view on user performance in information-rich virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2005), ACM, pp. 46–55.

[Pop07] Poppe, R.: Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding 108* (2007), 4–18.

[PP04] Pierce, J. S., Pausch, R.: Navigation with place representations and visible landmarks. In *Proceedings of the IEEE Virtual Reality* (Washington, DC, USA, 2004), IEEE Computer Society, p. 173.

[PSTD12] Pasewaldt, S., Semmo, A., Trapp, M., Döllner, J.: Towards comprehensible digital 3D maps. In *Service-Oriented Mapping 2012 (SOMAP2012)*. M. Jobst (Eds.). Jobstmedia Management Verlag, Wien, 2012, pp. 261–276.

[PZG*13] Pan, B., Zhao, Y., Guo, X., Chen, X., Chen, W., Peng, Q.: Perception-motivated visualization for 3D city scenes. *The Visual Computer 29* (2013), 277–286.

[QWC*09] Qu, H., Wang, H., Cui, W., Wu, Y., Chan, M.-Y.: Focus+context route zooming and information overlay in 3D urban environments. *IEEE Transactions on Visualization and Computer Graphics 15* (2009), 1547–1554.

[RCM93] Robertson, G. G., Card, S. K., Mackinlay, J. D.: Information visualization using 3d interactive animation. *Communications ACM 36* (1993), 57–71.

[RCvD97] Robertson, G., Czerwinski, M., van Dantzich, M.: Immersion in desktop virtual reality. In *Proceedings of the 10th Annual ACM SIGGRAPH Symposium on User Interface Software* (New York, NY, USA, 1997), ACM, pp. 11–19.

[RDH09] Reisman, J. L., Davidson, P. L., Han, J. Y.: A screen-space formulation for 2D and 3D direct manipulation. In *Proceedings of the 22nd Annual ACM SIGGRAPH Symposium on User Interface Software* (New York, NY, USA, 2009), ACM, pp. 69–78.

[RDSGA*00] Russo Dos Santos, C., Gros, P., Abel, P., Loisel, D., Trichaud, N., Paris, J.: Metaphor-aware 3D navigation. In *IEEE Symposium on Information Visualization* (2000), IEEE, pp. 155–165.

[RHPJ00] Ruddle, R. A., Howes, A., Payne, S. J., Jones, D. M.: The effects of hyperlinks on navigation in virtual environments. *International Journal of Human-Computer Studies 53* (2000), 551–581.

[RM99] Richardson, A. E., Montello, D. R.: Spatial knowledge acquisition from maps and from navigation in real and virtual environments. *Memory & Cognition 27* (1999), 741–750.

[Ros93] Rosenberg, L. B.: The effect of interocular distance upon operator performance using stereoscopic displays to perform virtual depth tasks. In *Proceedings of the 1993 Virtual Reality Annual International Symposium (VRAIS '97)* (Washington, DC, USA, 1993), IEEE Computer Society, pp. 27–32.

[Rou00] Rouse, R.: *Game Design: Theory and Practice* (2nd edition). Wordware Publishing Inc., Plano, TX, USA, 2000.

[RPJ99] Ruddle, R. A., Payne, S. J., Jones, D. M.: The effects of maps on navigation and search strategies in very-large-scale virtual environments. *Journal of Experimental Psychology 5* (1999), 54.

[Rub91] Rubine, D.: Specifying gestures by example. *SIGGRAPH Computer Graphics 25* (1991), 329–337.

[RvDR*00] Robertson, G., van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., Thiel, D., Gorokhovsky, V.: The task gallery: A 3D window manager. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2000), ACM, pp. 494–501.

[RW02] Raubal, M., Winter, S.: Enriching wayfinding instructions with local landmarks. In *Geographic Information Science*. M. J., Egenhofer, D. M. Mark (Eds.). Springer, Berlin, Heidelberg, 2002, pp. 243–259.

[SB09] Silva, M. G., Bowman, D. A.: Body-based interaction for desktop games. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2009), ACM, pp. 4249–4254.

[SC92] Strauss, P. S., Carey, R.: An object-oriented 3d graphics toolkit. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1992), ACM, pp. 341–349.

[SFC*11] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2011), IEEE Computer Society, pp. 1297–1304.

[SG04] Sutcliffe, A. G., Gault, B.: Heuristic evaluation of virtual reality applications. *Interacting with Computers 16* (2004), 831–849.

[SG06] Smith, J. D., Graham, T. C. N.: Use of eye movements for video game control. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology* (New York, NY, USA, 2006), ACM, article no. 20.

[SG09] Sko, T., Gardner, H. J.: Head tracking in first-person games: Interaction using a web-camera. In *Human-Computer Interaction-INTERACT*. T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. O. Prates, M. Winckler (Eds.). Springer, Berlin, Heidelberg, 2009, pp. 342–355.

[SGLM03] Salomon, B., Garber, M., Lin, M. C., Manocha, D.: Interactive navigation in complex environments using path planning. In *Proceedings of the 2003 symposium on Interactive 3D graphics* ((New York, NY, USA, 2003), ACM, pp. 41–50.

[SGS04] Singh, K., Grimm, C., Sudarsanam, N.: The ibar: A perspective-based camera widget. In *Proceedings of the 17th Annual ACM SIGGRAPH Symposium on User Interface Software* (New York, NY, USA, 2004), ACM, pp. 95–98.

[SGSS08] Snavely, N., Garg, R., Seitz, S. M., Szeliski, R.: Finding paths through the world's photos. *ACM Transactions on Graphics 27* (2008), 15:1–15:11.

[SH06] Smith, S. P., Hart, J.: Evaluating distributed cognitive resources for wayfinding in a desktop virtual environment. In *Proceedings of the IEEE Conference on Virtual Reality* (Washington, DC, USA, 2006), IEEE Computer Society, p. 115.

[Shn83] Shneiderman, B.: Direct manipulation: A step beyond programming languages. *IEEE Computer 16* (1983), 57–69.

[Shn03] Shneiderman, B.: Why not make interfaces better than 3D reality? *IEEE Computer Graphics and Applications 23* (2003), 12–15.

[Sho85] Shoemake, K.: Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1985), ACM, pp. 245–254.

[Sho92] Shoemake, K.: Arcball: A user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the Conference on Graphics Interface '92* (San Francisco, CA, USA, 1992), Morgan Kaufmann Publishers Inc., pp. 151–156.

[SIS02] Strauss, P. S., Issacs, P., Shrag, J.: The design and implementation of direct manipulation in 3D. In *SIGGRAPH Course Notes* (2002).

[SK00] Sutcliffe, A. G., Kaur, K. D.: Evaluating the usability of virtual reality user interfaces. *Behaviour and Information Technology 19* (2000), 415–426.

[SKLS06] Sadeghian, P., Kantardzic, M., Lozitskiy, O., Sheta, W.: The frequent wayfinding-sequence (FWS) methodology: Finding preferred routes in complex virtual environments. *International Journal of Human-Computer Studies 64* (2006), 356–374.

[SMG*09] Seagull, F., Miller, P., George, I., Mlyniec, P., Park, A.: Interacting in 3D space: Comparison of a 3D two-handed interface to a keyboard-and-mouse interface for medical 3D image manipulation. In *Human Factors and Ergonomics Society Annual Meeting* (2009).

[SMR*03] Stanney, K. M., Mollaghasemi, M., Reeves, L., Breaux, R., Graeber, D. A.: Usability engineering of virtual environments (VEs): Identifying multiple criteria that drive effective VE system design. *International Journal of Human-Computer Studies 58* (2003), 447–481.

[SP04] Shneiderman, B., Plaisant, C.: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson Addison Wesley, 2004.

[SP08] Sokolov, D., Plemenos, D.: Virtual world explorations by using topological and semantic knowledge. *Vision Computing 24* (2008), 173–185.

[SPHB08] Schlömer, T., Poppinga, B., Henze, N., Boll, S.: Gesture recognition with a WII controller. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction* (New York, NY, USA, 2008), ACM, pp. 11–14.

[SS02] Strothotte, T., Schlechtweg, S.: *Non-photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[SSB08] Schmidt, R., Singh, K., Balakrishnan, R.: Sketching and composing widgets for 3D manipulation. *Computer Graphics Forum 27* (2008), 301–310.

[SSKLK13] Shreiner, D., Sellers, G., Kessenich, J., Licea-Kane, B.: *OpenGL Programming Guide: The Official Guide to Learning OpenGL* (8th editon). Addison-Wesley Professional, 2013.

[SSS01] SMITH, G., SALZMAN, T., STUERZLINGER, W.: 3D scene manipulation with 2D devices and constraints. In *Proceedings of the Graphics Interface 2001* (Toronto, ON, Canada, 2001), Canadian Information Processing Society, pp. 135–142.

[SSS06] SNAVELY, N., SEITZ, S. M., SZELISKI, R.: Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics 25* (2006), 835–846.

[Ste06] STEED, A.: Towards a general model for selection in virtual environments. In *IEEE Symposium on 3D User Interfaces (3DUI)* (2006), IEEE, pp. 103–110.

[STKD12] SEMMO, A., TRAPP, M., KYPRIANIDIS, J. E., DÖLLNER, J.: Interactive visualization of generalized virtual 3D city models using level-of-abstraction transitions. *Computer Graphics Forum 31* (2012), 885–894.

[SUS95] SLATER, M., USOH, M., STEED, A.: Taking steps: The influence of a walking technique on presence in virtual reality. *TOCHI 2* (1995), 201–219.

[Sut63] SUTHERLAND I., : *Sketchpad: A Man-Machine Graphical Communications System*. PhD thesis, MIT, 1963.

[Sut65] SUTHERLAND, I.: The ultimate display. In *Proceedings of the Congress of the Internation Federation of Information Processing (IFIP)* (1965).

[SW05] SWEETSER, P., WYETH, P.: Gameflow: A model for evaluating player enjoyment in games. *Computer Entertainment 3* (2005), 3.

[SWSJ05] SCHMIDT, R., WYVILL, B., SOUSA, M. C., JORGE, J. A.: Shapeshop: Sketch-based solid modeling with blobtrees. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling* (2005).

[SZ03] SALEN, K., ZIMMERMAN, E.: *Rules of Play: Game Design Fundamentals*. The MIT Press, 2003.

[TBN00] TOMLINSON, B., BLUMBERG, B., NAIN, D.: Expressive autonomous cinematography for interactive virtual environments. In *Proceedings of the Fourth International Conference on Autonomous Agents* (New York, NY, USA, 2000), ACM, pp. 317–324.

[TBPD11] TRAPP, M., BEESK, C., PASEWALDT, S., DÖLLNER, J.: Interactive rendering techniques for highlighting in 3D geovirtual environments. In *Advances in 3D Geo-Information Sciences* (2011).

[TDS99] TEMPLEMAN, J. N., DENBROOK, P. S., SIBERT, L. E.: Virtual locomotion: Walking in place through virtual environments. *Presence 8* (1999), 598–617.

[TGBD08] TRAPP, M., GLANDER, T., BUCHHOLZ, H., DOLNER, J.: 3D generalization lenses for interactive focus+context visualization of virtual city models. In *12th International Conference Information Visualisation* (2008), pp. 356–361.

[Tho82] THORNDYKE, P.: Differences in spatial knowledge acquired from maps and navigation. *Cognitive Psychology 14* (1982), 560–589.

[TJ00] TANRIVERDI, V., JACOB, R. J. K.: Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2000), ACM, pp. 265–272.

[TKB09] TURKAY, C., KOC, E., BALCISOY, S.: An information theoretic approach to camera control for crowded scenes. *Vision Computing 25* (2009), 5–7.

[TLHW09] TERZIMAN, L., LECUYER, A., HILLAIRE, S., WIENER, J. M.: Can camera motions improve the perception of traveled distance in virtual environments? In *Virtual Reality Conference* (2009), IEEE, pp. 131–134.

[TME*10] TERZIMAN, L., MARCHAL, M., EMILY, M., MULTON, F., ARNALDI, B., LÉCUYER, A.: Shake-your-head: Revisiting walking-in-place for desktop virtual reality. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2010), ACM, pp. 27–34.

[TN10] TAN, D. S., NIJHOLT, A.: *Brain-Computer Interfaces: Applying Our Minds to Human-Computer Interaction*. Springer, 2010.

[Tod04] TODD, J. T.: The visual perception of 3D shape. *Trends in Cognitive Science* 8 (2004), 115–121.

[TRC01] TAN, D. S., ROBERTSON, G. G., CZERWINSKI, M.: Exploring 3D navigation: Combining speed-coupled flying with orbiting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2001), ACM, pp. 418–425.

[TS04] THIEMANN, F., SESTER, M.: Segmentation of buildings for 3D-generalisation. In *ICA Workshop on Generalisation and Multiple Representation* (2004).

[TS11] TEATHER, R. J., STÜRZLINGER, W.: Pointing at 3D targets in a stereo head-tracked virtual environment. In *Proceedings of the 2011 IEEE Symposium on 3D User Interfaces* (Washington, DC, USA, 2011), IEEE Computer Society, pp. 87–94.

[TSHD09] TRAPP, M., SCHNEIDER, L., HOLZ, N., DÖLLNER, J.: Strategies for visualizing points-of-interest of 3D virtual environments on mobile devices. In *International Symposium on LBS & TeleCartography* (2009).

[TSL*11] TRAPP, M., SCHNEIDER, L., LEHMANN, C., HOLZ, N., DOLLNER, J.: Strategies for visualising 3D points-of-interest on mobile devices. *Journal on Location-Based Services 5* (2011), 79–99.

[TYSN06] TAKAHASHI, S., YOSHIDA, K., SHIMADA, K., NISHITA, T.: Occlusion-free animation of driving routes for car navigation systems. *IEEE Transactions on Visualization and Computer Graphics 12* (2006), 1141–1148.

[vBE01] van Ballegooij, A., Eliens, A.: Navigation by query in virtual worlds. In *Proceedings of the Sixth International Conference on 3D Web Technology* (New York, NY, USA, 2001), ACM, pp. 77–83.

[VC01] Vallance, S., Calder, P.: Context in 3D planar navigation. In *Proceedings of the 2Nd Australasian Conference on User Interface* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 93–99.

[vD97] van Dam, A.: Post-wimp user interfaces. *Communications ACM 40* (1997), 63–67.

[Ven93] Venolia, D.: Facile 3D direct manipulation. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (New York, NY, USA, 1993), ACM, pp. 31–36.

[VFSH01] Vázquez, P.-P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In *Proceedings of the Vision Modeling and Visualization Conference 2001* (2001), Aka GmbH, pp. 273–280.

[VGC07] Vanacken, L., Grossman, T., Coninx, K.: Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. In *IEEE Symposium on 3D User Interfaces, 3DUI '07* (2007) IEEE.

[Vin99] Vinson, N. G.: Design guidelines for landmarks to support navigation in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1999), ACM, pp. 278–285.

[VIR*09] Villar, N., Izadi, S., Rosenfeld, D., Benko, H., Helmes, J., Westhues, J., Hodges, S., Ofek, E., Butler, A., Cao, X., Chen, B.: Mouse 2.0: Multi-touch meets the mouse. In *Proceedings of the 22nd Annual ACM SIGGRAPH Symposium on User Interface Software* (New York, NY, USA, 2009), ACM, pp. 33–42.

[WAB93] Ware, C., Arthur, K., Booth, K. S.: Fish tank virtual reality. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (New York, NY, USA, 1993), ACM, pp. 37–42.

[WBM*02] Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., Vaughan, T. M.: Brain-computer interfaces for communication and control. *Clinical Neurophysiology* (2002).

[WC02] Walczak, K., Cellary, W.: Building database applications of virtual reality with x-vrml. In *Proceedings of the Seventh International Conference on 3D Web Technology* (New York, NY, USA, 2002), ACM, pp. 111–120.

[WF96] Ware, C., Franck, G.: Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics 15* (1996), 121–140.

[WF97] Ware, C., Fleet, D.: Context sensitive flying interface. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1997), ACM, p. 127.

[WFHM11] Walther-Franks, B., Herrlich, M., Malaka, R.: A multi-touch system for 3D modelling and animation. In *Proceedings of the 11th International Conference on Smart Graphics* (Berlin, Heidelberg, 2011), Springer-Verlag, pp. 48–59.

[WH99] Wernert, E. A., Hanson, A. J.: A framework for assisted exploration with collaboration. In *Visualization'99* (1999), ACM, pp. 241–529.

[Wil06] Wilson, G.: Off with their huds!: Rethinking the heads-up display in console game design, 2006. Accessed 25 September 2014.

[WO90] Ware, C., Osborne, S.: Exploration and virtual camera control in virtual three dimensional environments. In *Proceedings of the 1990 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1990), ACM, pp. 175–183.

[WRLP94] Wharton, C., Rieman, J., Lewis, C., Polson, P.: The Cognitive Walkthrough Method: A Practitioner's Guide. *Usability Inspection Methods*. J. Nielsen, R. L. Mack (Eds.). John Wiley & Sons, Inc., New York, NY, USA, 1994, pp. 105–140.

[WWV*10] Wingrave, C. A., Williamson, B., Varcholik, P. D., Rose, J., Miller, A., Charbonneau, E., Bott, J., LaViola, Jr. J. J.: The wiimote and beyond: Spatially convenient devices for 3D user interfaces. *IEEE Computer Graphics and Applications 30* (2010), 71–85.

[WXX*06] Wang, S., Xiong, X., Xu, Y., Wang, C., Zhang, W., Dai, X., Zhang, D.: Face-tracking as an augmented input in video games: Enhancing presence, role-playing and control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2006), ACM, pp. 1097–1106.

[WZHZ09] Wu, A., Zhang, W., Hu, B., Zhang, X.: Evaluation of wayfinding aids interface in virtual environment. *International Journal of Human-Computer Interaction 25* (2009), 1–21.

[XH98] Xiao, D., Hubbold, R.: Navigation guided by artificial force fields. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1998), ACM Press/Addison-Wesley Publishing Co., pp. 179–186.

[YQG08] Yim, J., Qiu, E., Graham, T. C. N.: Experience in the design and development of a game based on head-tracking input. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share* (New York, NY, USA, 2008), ACM, pp. 236–239.

[ZBM94] Zhai, S., Buxton, W., Milgram, P.: The "silk cursor": Investigating transparency for 3D target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1994), ACM, pp. 459–464.

[ZF99] Zeleznik, R., Forsberg, A.: Unicam—2D gestural camera controls for 3D environments. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics* (New York, NY, USA, 1999), ACM, pp. 169–173.

[ZFS97] Zeleznik, R. C., Forsberg, A. S., Strauss, P. S.: Two pointer input for 3D interaction. In *Proceedings of the 1997*

Symposium on Interactive 3D Graphics (New York, NY, USA, 1997), ACM, pp. 115–120.

[ZHH06] ZELEZNIK, R. C., HERNDON, K. P., HUGHES, J. F.: Sketch: An interface for sketching 3d scenes. In *ACM SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), ACM, article no. 9.

[ZM98] ZHAI, S., MILGRAM, P.: Quantifying coordination in multiple DoF movement and its application to evaluating 6 DoF input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1998), ACM Press/Addison-Wesley Publishing Co., pp. 320–327.