Hover Pad: Interacting with Autonomous and Self-Actuated Displays in Space

Juilan Seifert[1], Sebastian Boring[2], Christian Winkler[1], Florian Schaub[3]
Fabian Schwab[1], Steffen Herrdum[1], Fabian Maier[1], Daniel Mayer[1], Enrico Rukzio[1]
[1]Institute of Media Informatics, Ulm University, Germany
[2]Department of Computer Science, University of Copenhagen, Denmark
[3]School of Computer Science, Carnegie Mellon University, USA
[1]{firstname.lastname}@uni-ulm.de, [2]sebastian.boring@di.ku.dk, [3]fschaub@cmu.edu

ABSTRACT

Handheld displays enable flexible spatial exploration of information spaces – users can physically navigate through three-dimensional space to access information at specific locations. Having users constantly hold the display, however, has several limitations: (1) inaccuracies due to natural hand tremors; (2) fatigue over time; and (3) limited exploration within arm's reach. We investigate *autonomous, self-actuated displays* that can freely move and hold their position and orientation in space without users having to hold them at all times. We illustrate various stages of such a display's autonomy ranging from *manual* to *fully autonomous*, which – depending on the tasks – facilitate the interaction. Further, we discuss possible motion control mechanisms for these displays and present several interaction techniques enabled by such displays. Our *Hover Pad* toolkit enables exploring five degrees of freedom of self-actuated and autonomous displays and the developed control and interaction techniques. We illustrate the utility of our toolkit with five prototype applications, such as a volumetric medical data explorer.

Author Keywords

Self-actuated displays; volumetric data sets; interaction.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation

General Terms

Design, Human Factors.

INTRODUCTION

With their mobility, handheld displays such as tablets suggest themselves for spatial exploration of information spaces. They can provide a digital window into a much larger three-dimensional information space (see Figure 1a). This approach can help users explore and understand complex volumetric data sets. Information spaces are typically either centered around the user's body [6, 33], or anchored to larger displays in the environment [11, 12, 27]. Previous research

UIST '14, October 05 - 08 2014, Honolulu, HI, USA
Copyright 2014 ACM 978-1-4503-3069-5/14/10…$15.00.
http://dx.doi.org/10.1145/2642918.2647385

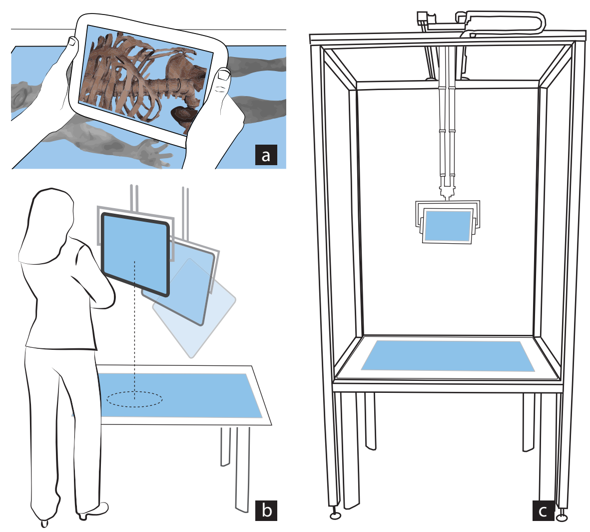

Figure 1. Handheld displays allow exploring volumetric data sets, yet users have to hold them in their hands at all times (*a*). Self-actuated and autonomous displays, on the other hand, jettison this requirement (*b*). Our prototype is a first realization of a self-actuated display that can autonomously move and hold its position (*c*).

has assumed that people move the display manually (may it be a tablet computer or a sheet of paper with projection) using their hands. While in motion, the display content changes continuously according to its position and orientation in space.

Manually controlling the display's position and orientation empowers users to navigate to a desired location in the information space. This approach, however, has its shortcomings: (1) users hold the device continuously (occupying at least one hand) which may increase fatigue; (2) exact positioning becomes difficult due to natural hand tremors [25]; and (3) users have to search for information within the information space which might be time-consuming and error-prone (i.e., missing important aspects in the data as users focus on finding a specific item instead). In summary, handheld displays are tied to the user's physical input (here: moving it in space) in order to change their content.

In this work, we set out to free handheld displays from the user's physical input constraints. That is, displays can autonomously move within the information space of a volumetric data set. Unlike previous systems (e.g., [11, 12, 27, 33]), users do not have to hold the display in their hands; instead the display can move autonomously and maintain its position and orientation (see Figure 1b). This autonomous actuation can further be combined with manual input by users, e.g., a user moving the display to a position where it then remains. To investigate this new class of displays, we built *Hover Pad*– a self-actuated display system mounted to a crane. Our setup allows for controlling five degrees of freedom: moving the display along its *x*-, *y*-, and *z*-axes; and changing both *pitch* (i.e., the displays's horizontal axis) and *yaw* (i.e., the vertical axis) of the display.

With its self-actuated nature, our setup offers three advantages over displays held in hand that are positioned physically and manually by users: (1) the display can *move autonomously* in space without requiring a user's physical effort; (2) it allows for *hands-free interaction* as users do not have to hold the display in their hand continuously – thus reducing fatigue in arms and hands as well as using hands for parallel tasks; and (3), it offers enhanced *visual stability* compared to manually holding a still in a certain position and orientation (i.e., natural hand tremor).

In this paper, we investigate designing interactions with such displays based on our prototype. We focus on how their movement can be controlled either autonomously by the system or by users. We present techniques that, we believe, will benefit from *autonomous and self-actuated* displays. In summary, our work offers two main contributions:

1. A set of **interaction techniques** that allow for controlling the display position – either in a *semi-autonomous* fashion, where the display moves and orients itself on its own following a user's request, or in a *manual* fashion, where users explicitly control the display's motion.
2. A set of **example applications** that were built using our prototype setup of *Hover Pad*. These applications make use of the presented interaction techniques to demonstrate their utility in real-world scenarios.

Further, we present a prototyping toolkit that allows for rapid prototyping of such displays – including a detailed description of how such displays can be constructed. This toolkit enables developers to make use of the presented control mechanisms in a simplified way.

Note that our main contribution lies in the engineering domain to enable the exploration of *autonomous and self-actuated* displays. We do not present a user study of exploring volumetric data using tablet computers, as this has been explored already (e.g., [26, 24]).

RELATED WORK

Our work builds on (1) *spatial exploration of information spaces* with handheld devices, and (2) on *self-actuated objects* both on tabletops and in mid-air.

Spatial Exploration of Information with Handheld Devices

Our *Hover Pad* system combines a tablet and an interactive surface, emphasizing the importance of the spatial relationship between these devices. *Chameleon* and *Boom Chameleon* investigated manually controlled exploration of virtual reality in three-dimensional space [7, 28]. More recently, the combination of mobile devices and large displays has been explored. Schmidt et al.'s *PhoneTouch* locates where a mobile device (and which one) touched a surface [21]. Others explored tracking mobile devices in three dimensions in front of a display [2, 4].

Mobile devices have been used to explore three-dimensional information spaces. When the mobile device is used without another, larger display, these spaces are anchored around the device. In Yee's *Peephole Displays* users move the handheld display in mid-air to reveal content that is virtually located around the device [33]. With *Boom Chameleon*, users can navigate *around* an object in three-dimensional space [7, 28]. The display is attached to a boom, which allows for maintaining the displays's position and orientation after it has been changed *manually*. Unlike *Hover Pad*, its physical constraints prevent movement *through* virtual objects. Chen et al. constructed the information space around the user's body. The handheld display reveals different information based on its location relative to the user's body [6].

In many existing systems, the information space is anchored to a larger display in the environment with that display providing an overview (e.g., a bird's eye view) of the space, which is inspected in detail using a handheld display [26, 24]. Marquardt et al. demonstrate the use of a tablet computer to physically navigate through a pile of photographs [14]. Besides volumetric data, the interaction above or in front of a large display may also extend 2D visualizations. Izadi et al.'s *SecondLight* [11], for example, takes *Magic Lenses* [3] into the third dimension.

With the exception of *Boom Chameleon*, all of these approaches are based on handheld displays and thus require to constantly hold the display in-hand. Exploring the space out of arms' reach becomes impossible and it becomes increasingly difficult to hold the device still at a certain position. In addition, users have to move the device in very small steps in order to explore finer details within that space. This may slow down the interaction [25] and increasing fatigue as well as natural hand tremors make it difficult to examine fine-grained structures. Our design intends to overcome these limitations through self-actuated movement that allows precise positioning, hands-free interaction as well as reaching space out of the user's reach (e.g., exploring information spaces that extend beyond the user's immediate proximity).

Self-Actuated Objects on Interactive Surfaces

Self-actuated objects on interactive surfaces that allow, for instance, to physically animate application state changes (e.g., a slider value or position), have been studied previously. Most prominently, magnets embedded underneath the surface are used to move magnetic objects on top [8]. The *Actuated Workbench* provides feedback by moving tracked objects on an interactive surface [17]. *Pico* works similarly, but adds physical constraints to movable objects [18]. Weiss et al.'s *Madgets* further enable tangibles that can move vertically in a limited range [31]. In addition, they are able to simulate physical properties, such as friction, while people move the tangibles [30].

Others experimented with alternative approaches to self-actuated objects. Rosenfeld et al.'s *Planar Manipulator Display* [20] and Pedersen et al.'s *Tangible Bots* [19] create movable objects by attaching wheels to them. Ultra-sonic air waves [15] or vibration, as in *Touchbugs* [16], have also been explored to control autonomous movement of objects. However, both approaches are constraint in either the objects that are able to move (i.e., lightweight objects through sound) or the level of movement (i.e., one direction with vibration). Also, they can only move in one plane (directly on the top of the surface). Nevertheless, the aspect of physical feedback present in each of these systems also inspired our approach.

Movable Objects and Displays in Mid-Air

More recent work focused on moving objects and displays in three-dimensional space. Alrøe et al.'s *Aerial Tunes* [1] lets multiple balls hover over boxes using a controlled air stream in order to visualize an artistic sound installation. Lee et al.'s *ZeroN* [13] uses electromagnets to position a magnetic sphere in mid-air. When projected upon, this sphere is turned into a display. *ZeroN* can also provide force feedback by changing the magnetic field. In both systems, users are able to reposition objects as a form of input. Hörtner et al.'s *Spaxels* create a large scale volumetric display using small quadcopters, each representing one pixel in space [10]. *Hover Pad* is inspired by these diverse approaches. However, we chose a crane-based implementation for the self-actuated movement in order to achieve a high level of position accuracy and visual stability which allows to investigate diverse interaction techniques.

Besides objects, displays can also be moved in space (without requiring a user's physical effort). Wilson et al.'s *Beamatron* combines a steerable projector with a depth camera to empower users to move digital content in an environment through gestures [32]. In contrast to a projected display, Sinclair et al. mounted a touch-display onto a crane so that it can move forward and backward along one dimension when a user touches the display [23]. This motion allows for exploring physical attributes of virtual objects (e.g., weight). A display moving along one dimension was used by Hashimoto et al. to emphasize movement of a virtual game character moving through a virtual world [9]. With *RoCo*, Breazeal et al. present a robotic computer that features a self-actuated display that can respond to and influence its user's posture [5]. Our work allows for investigating (semi-)autonomous display motion control and interaction in a comparably large *three-dimensional* space.

MOVING SELF-ACTUATED DISPLAYS

The display in our setup can move fully *autonomously*. That is, it can follow predefined paths without requiring any user interaction. However, the setup also allows for two additional types of movement, each of which is triggered by user interaction: (1) the display can move *semi-autonomously* (e.g., in response to an alert or due to artificial intelligence in a game); and (2), its position and orientation can be controlled *manually* by the user. These options can also be combined in flexible ways. In the following we describe these two capabilities in more detail.

Semi-Autonomous Movement

The display can move to a location based on the user's request. However, it still moves to that location on its own without requiring the user to hold it in hand at any time. For example, the surface display could show a top-down view of a human body and the tablet shows the corresponding slice of a CT scan. The user could now request a specific location (say: a slice of the brain) by tapping on the location on the tabletop. This then triggers the display to move to that location in space. In contrast to existing systems, users do not have to physically search for a given area of interest in the information space. In particular, this enables two interaction techniques: *search & inspect* and *bookmark & recall*.

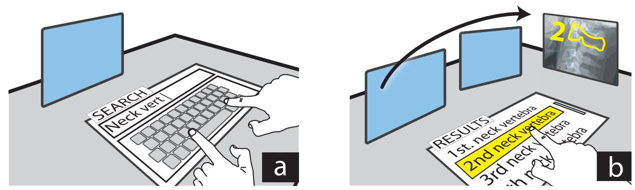

Figure 2. *Search & Inspect* for searching and selecting an item (a). The display then moves autonomously to that item in space (b).

Search & Inspect

Volumetric data sets are often structured and have specific, well-defined areas of interest (i.e., in a CT scan a specific density in a tissue area may indicate a tumor) the system is aware of these areas and knows their location. This allows users to *search* for specific locations and subsequently *inspect* data located there. Figure 2 shows the stages of this interaction technique in more detail: (a) users perform a search (i.e., by entering the name of the area of interest); (b) within the resulting list (neck vertebrae), users can now select the item of interest; (c) upon selection, the display transitions to that location in the volumetric data set for further inspection by the user. The surface display and the display's height can provide context information about this item's position in the volumetric dataset.

Bookmark & Recall

Similar to *Search & Inspect*, users can set *bookmarks* of a location within the volumetric data set – for example, when they come across a detail that they wish to inspect further at a later time. Figure 3 demonstrates the use of this interaction technique: (a) the user presses a button to bookmark a location. The display then stores the location (i.e., its position and orientation); (b) at a later point in time, the user can select a bookmark from a list of bookmarks. This triggers the display to return to the position associated with the bookmark.

Manually Controlling the Display's Motion

When a volumetric data set is more suited for users exploring data, the display's position has to be controlled manually. That is, users should be able to tell the display where to move to. We envision three basic types of interactions.

Directly Moving the Display

The most obvious way of controlling the display's position and orientation is by moving it manually in space. That is, users can grab it and move it directly to the desired location

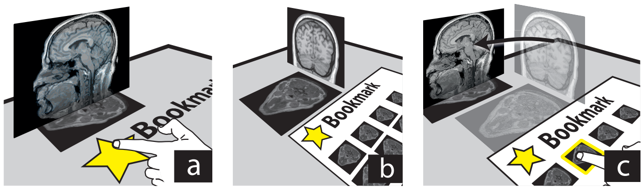

Figure 3. ***Bookmark & Recall*** **allows users to create bookmarks of views (a). Later on, users can select one of these bookmarks from a list (b). This causes the mobile display to return to that bookmark's corresponding position and orientation (c).**

(see Figure 4a). Once the display has been brought to the intended position, the user can let go of the display and it remains in that position, which frees the user's hands (e.g., for secondary tasks such as taking notes or switching between visualizations of the data). Note that – depending on the actual implementation – sensing of the user's input (i.e., force and direction) might be required (e.g., through touch sensors or pressure sensors) which is then translated to movement of the self-actuated display.

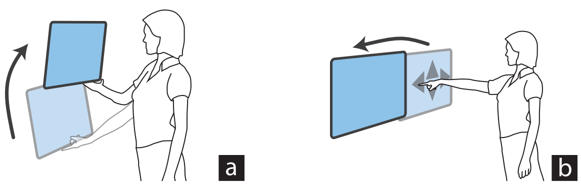

Figure 4. Directly moving the display allows users to directly grab and move the display to change its position and orientation (a). Widget-based interaction on the display's surface enables more fine-grained control of the same movement parameters (b).

Widget-based Motion Control

Widgets represent another opportunity to control a self-actuated display's position and orientation. These widgets can be displayed either on the display itself (see Figure 4b) or on another screen (e.g., the interactive surface below the display). One approach is to provide buttons that correspond to movement of each supported DoF. That is, each possible motion attribute can be manipulated in two directions: *'increase'* and *'decrease.'* A mapping function then determines the granularity of movement – either fine-grained for *slow* or coarse for *fast* movements.

Controlling Motion through Gestures

Another way to control a self-actuated display in space is the use of gestures. We envision that users can apply gestures either (1) in mid-air or (2) on the connected interactive surface. Mid-air gestures allow for controlling the display position while the display is out of reach. Figure 5 shows one possible gesture: (a) the user applies a *picking* gesture to activate motion control; (b) the hand's motion is then mapped either with a *zero order mapping* (i.e., the hand's motion controls the display's position) or with a *first order mapping* (i.e., the hand's motion controls the display's speed).

On the context providing interactive surface, users can perform multi-touch gestures on that device to control the display's motion. For instance, *pinch* gestures for controlling the height, and *swipe* gestures for controlling the *x*- and *y*-position of the tablet.

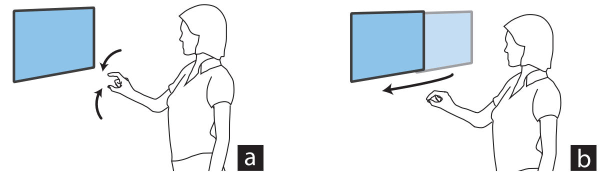

Figure 5. With gesture control, the user first performs a *pinch* gesture (a) to bind the hand's movement to the display's motion. This then allows for continuously controlling the display's position and orientation (b).

IMPLEMENTATION

To explore the aforementioned aspects in more detail, we built *Hover Pad* – a prototype that resembles a *self-actuated and autonomous* display. We further designed a toolkit that implements all the aforementioned motion controls. In this section, we highlight both hardware and software of our prototype. Note that *Hover Pad*'s blueprints as well as the software toolkit are available as open source.[1]

Hardware Setup

The prototype uses a Nexus 10 tablet as *self-actuated display* as well as a Microsoft Surface 2 (Samsung SUR40), which is situated underneath and acts as a secondary display in the environment to provide contextual information. Figure 6a shows the overall setup. To allow for self-actuated movement of a display in three-dimensional space, we designed and built a custom overhead gantry crane. One sliding carriage moves the tablet along the *x*-axis. This carriage holds a second sliding carriage that moves the display along the *y*-axis (see Figure 6b). Sliding carriages are moved by separate step motors (1.8° step angle; 0.5 Nm holding torque; 12V operating voltage) connected to drive belts. The motors driving the carriages as well as the motor controlling vertical movement are connected to a controller unit. This unit provides command messages and power for these motors.

Two parallel telescope bars are connected to the carriage responsible for movement along the *z*-axis (see Figure 6c). Each of these custom engineered telescope bars consists of six elements (made of aluminum; supported through integrated brass rings as gliding means) with one element being 22 cm long.

Attached to them is a mount holding the tablet (see Figure 6d). This mount enables self-actuated rotation along two axes: *pitch* and *yaw*. Accordingly, two motors are integrated in the mount (see Figure 6d-*i* and *iv*). The motors are controlled via an Ioio OTG board (6d-*v*) which is connected to the tablet via Bluetooth. The frame further includes a battery (see 6d-*iii*) for powering the motors and the Ioio OTG board. The frame is equipped with 16 capacitive buttons located on the frame's rim front, side, and back (e.g., 6d-*ii*). These buttons can be mapped freely (by registering event listeners) to user-defined actions or steering commands in order to implement *direct movement of the display* as described in the previous section.

[1] `http://www.uni-ulm.de/?hover-pad`

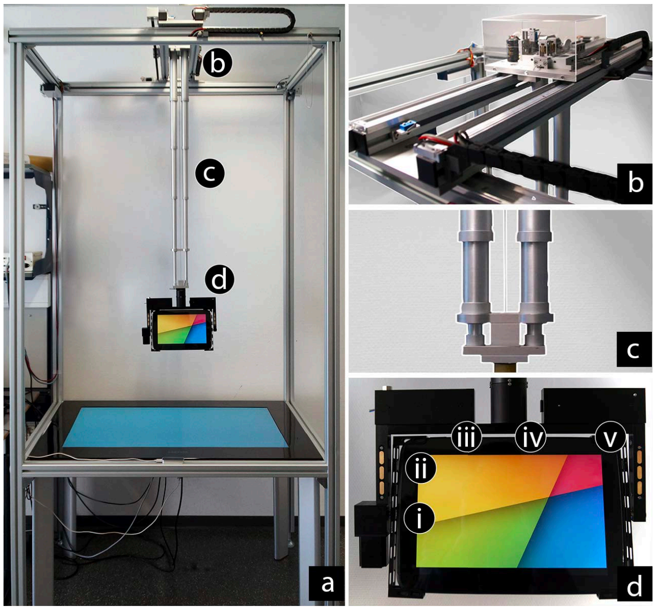

Figure 6. Overview and details of the *Hover Pad* hardware setup (a) with details regarding the sliding carriages for *x,y*-motion (b), the telescope bars for vertical motion (c), and the display's frame for rotation (d), comprising two motors (i, iv), a controller board (iii), a battery pack (v), and 16 capacitive buttons (ii).

The overall setup allows for an operational range of $90 \times 50 \times 107$ cm (*width / x* $\times$ *length / y* $\times$ *height / z*). The base of the operational range is aligned with the interactive surface. In order to limit the movement of the carriages and to prevent them from running out of bounds, five limit switches are installed (two on the *x*- and *y*-axes; one on the *z*-axis). With our current settings on the motor controller unit, the display needs about one second per 10 cm moving distance. Along the *z* and *y*-axis, the smallest possible movement step is about 0.05 mm. The display mount allows for continuous rotation around the *z*-axis (yaw) with a speed of 2.0 sec. per rotation (360°). Due to the mechanical construction, which includes a motor with a very low holding torque, the angle can be determined only with a tolerance of $\pm 10°$.

Software Toolkit

To enable rapid prototyping of applications that make use of *Hover Pad*'s capabilities, we created a software toolkit. This toolkit consists of four main components: the mobile client, the surface server, the crane motion, as well as the control component (see Figure 7).

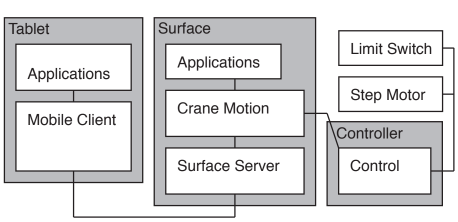

Figure 7. Overview of the *Hover Pad* software toolkit components.

The mobile client is implemented as an Android service running on the tablet and is responsible for managing communication with the surface server which runs on the interactive surface (here: SUR40). The mobile client and the surface server constantly exchange the tablet's position data via *JSON* objects (through HTTP via WiFi). The user's manual rotation of the tablet is sensed by an X-IMU sensor-box that is mounted on the display's back and connected via Bluetooth to the tablet. Also, when the crane motion component is called (i.e., by an application or a user's interaction) the updated position data is sent to the mobile client.

The crane motion component provides an application programming interface (API) that provides simplified methods allowing applications to request position changes of the tablet. Application developers can do so by either providing absolute positions (which requires an initial calibration of the system) or relative position changes. In both cases, it is sufficient to provide a vector with *x*, *y*, and *z* values (all in mm). The crane motion component then determines the number of steps each step motor has to perform in order to reach the requested position. The control component runs on an Arduino board (here: Mega 2560) integrated into the controller unit. It receives calls from the crane motion component (via USB) and sends these commands to the step motors.

In the following, we briefly illustrate how the *Hover Pad* toolkit supports rapid prototyping of applications. In particular, we highlight basic motion control options. Please note, a detailed documentation is available alongside the toolkit sources. Applications for *Hover Pad* are generally distributed, including a component running on the surface side and another component running on the tablet side.

Motion Control Options

Motion control calls can be performed by placing a request to move the tablet to an absolute or relative coordinate (see Listing 1). First, a coordinate is defined (by defining millimeter values, which correspond to exact physical distance). When requesting an absolute position, the tablet moves to this position in relation to the setup coordinate system. In case of relative movement, the given coordinate is used for movement in relation to the current position.

```
coordinate = new Point3D(23, 42, 5); //in cm
_mController.sendAbsolutePoint3D(coordinate);
// alternative relative movement
_mController.sendRelativePoint3D(coordinate);
```

Listing 1. Starting motion to an absolute or relative position.

In order to rotate the tablet in a specific orientation, the movement can be controlled through activating the rotation motors for a desired period of time (e.g., 50 ms). For convenience, a target angle can be set (see Listing 2): First, the auto rotation is activated and a tolerance value is defined. Finally, the angle is defined which triggers the movement instantly.

```
_mController.activateAutomaticOrientation();
_mController.setDeltaAngel(10); //degree
_mController.setYaw2be(90); // degree
```

Listing 2. Automatic rotation management of the tablet.

Connecting Interaction Controllers

On an application level, toolkit users can connect controls that enable interaction options, by simply adding a motion request call to the corresponding callback function. For instance, we used a Leap Motion controller to track mid-air gestures in front of the *Hover Pad* setup. These gestures can be mapped easily to *Hover Pad* movement, as shown in Listing 3.

```
_gestureTracker.gestureChange += new EventHandler<
    GestureEventArgs>(gestureDetected);
void gestureDetected(object sender,
    GestureEventArgs e) {/*trigger movement*/}
```

Listing 3. Mapping a gesture tracker *Hover Pad* movment.

EXAMPLE APPLICATIONS

We envision the usage of self-actuated and autonomous displays in various contexts that involve spatial relations and volumetric data such as education (e.g., in schools or museums), medical application, engineering and mining, surveillance of buildings, or production lines. We realized five example applications for *Hover Pad* that demonstrate how the key advantages support interaction, i.e., *autonomous movement*, *hands free interaction*, and *visual stability*. Examples were selected to showcase how *Hover Pad*'s key characteristics enable novel interaction. Table 1 gives an overview of which aspects are highlighted by the examples. The video figure illustrates the interactions of each of these examples.

	Phys. Object	Map Explorer	Volume Explorer	Anatomy Explorer	Game
Autonomous Movement	+	+	-	+	+
Hands Free Interaction	+	+	-	-	+
Visual Stability	+	-	+	+	+

Table 1. Example applications benefit from *Hover Pad* characteristics.

Physical Object Augmentation and Exploration

The first application allows users to explore a physical object (i.e., a model of the Empire State Building) that is placed on a map (of New York) shown on the interactive surface. The mobile display augments the physical object with virtual annotations (see Figure 8a). On the surface, the user can select points of interest from a list (e.g., a *bookmark* pointing to the 102nd floor), which triggers the tablet to move and show this point (see Figure 8b).

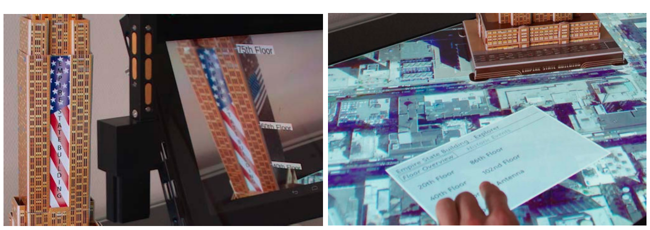

Figure 8. Augmenting physical objects with spatially registered annotations (a) allows users to select and explore diverse points of interest (b) that are than presented through the tablet.

With *Hover Pad* moving autonomously to selected targets, the interaction is predominantly *hands-free*. That is, users do not have to hold and maintain the tablet in one distinct position. This further allows users to interact with the augmented physical objects (e.g., rotate them while the augmentation follows). Through *self-actuated movement*, *Hover Pad* provides the unique possibility to guide users to specific views (based on *Bookmark&Recall*). Physical objects of arbitrary complexity can be augmented through *Hover Pad* with labels, explanations, or hints such as alerts. For instance, an unknown workpiece (e.g., an engine) could be explored by workers where *Hover Pad* would provide a list of changes that could be visited and examined one by one.

As users can freely move physical objects in relation to the *Hover Pad* setup, this example application required the usage of camera based object tracking[2] in addition to the *Hover Pad* framework. This allowed precise spatial correlation between the physical object and the rendered view on the tablet.

Map Explorer

The map explorer application allows users to view a focused 3D visualization of buildings on the tablet (similar to the *Tangible Geospace* by Ullmer and Ishii [29]). The interactive surface below provides context by displaying the surrounding environment (see Figure 9a) and allows users to control the tablet through *widgets-based* position control provided on the surface. The tablet acts as a *magic lens* and allows to switch between different views such as map or satellite view (see Figure 9b). The hight of the tablet is mapped to the zoom level of the provided view.

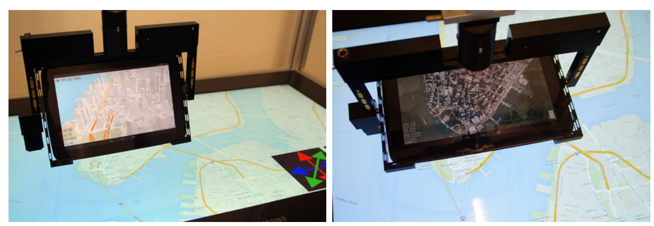

Figure 9. *Hover Pad* supports exploring maps by providing context on the surface and different spatially registered views such as a 3D view (a) or an alternative satelitte view (b).

The spatially registered displays (tablet and surface) allow users to simultaneously observe a focused view and context without any required transitions. Further, the *self-actuated* motion control allows uniquely logging and saving coordinates in a motion history that can be revisited automatically when desired (see *Bookmark&Recall*). Also, the view provided by the tablet enables autonomously controlled tracking shots that could, for instance, visualize flight route of a plane.

Medical Volumetric Data Explorer

A volumetric data viewer allows users to explore volumes such as computer tomography data sets. On the tablet a slice cut of the volume is displayed (see Figure 10a). On the surface the slice position is visualized on a schematic outline in order to support orientation (see Figure 10b).

Users can control the position of the tablet in a *direct and explicit* way by using the touch buttons on the rim of the tablet

[2]Based on Vuforia **http://www.vuforia.com/**

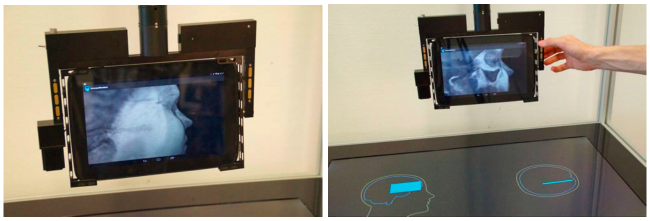

Figure 10. The anatomy explorer allows users to explore, for instance, the human skeleton (a) by selecting bookmarked bones from a list (b) which triggers the tablet to move to that position.

frame. This creates an experience similar to moving the tablet manually (i.e., touching a button on the right to push the tablet to the left). However, the user does not carry the weight of the tablet thus preventing fatigue. Also, the user can release the tablet which remains in its current position (*visual stability*) in order to have time to study and discuss e.g., a complex structure. Requiring explicit input to trigger movement has the advantage over gesture control that positioning can be finely controlled without the risk of accidentally causing the display to move while gesturing towards the point of interest. This application demonstrates how *Hover Pad* allows uniquely to transition *through* a volumetric data set while keeping *visual stability* as the user can release the tablet at any time which remains at this position.

Educational Anatomy Explorer

An anatomy explorer[3] allows users to view a virtual skeleton through the tablet computer (see Figure 11a). On the surface, the user can select, for instance, bookmarked bones that are revealed by the tablet upon selecting a target from a list (see Figure 11b). Alternatively, the user can explicitly move the tablet through space to explore the data set.

Figure 11. The anatomy explorer allows users to explore, for instance, the human skeleton (a) by selecting bookmarked bones from a list (b) which triggers the tablet to move to that position.

Using the *Bookmark&Recall* technique, users can explore human anatomy as *Hover Pad* shows specific body parts in spatial relation upon selecting a bookmark. For instance, in an anatomy class, the application helps users to find specific otherwise unknown details, which is only possible through *self-actuated* movement.

Mixed-Reality Gaming

A mixed-reality game for *Hover Pad* combines *autonomous behavior* and movement in space and interaction with tangible objects on the interactive surface. On the tablet, a virtual character is displayed that follows a virtual path. Accordingly, the tablet moves autonomously to follow the character

[3]Based on 4D Anatomy Explorer; `http://daqri.com/`.

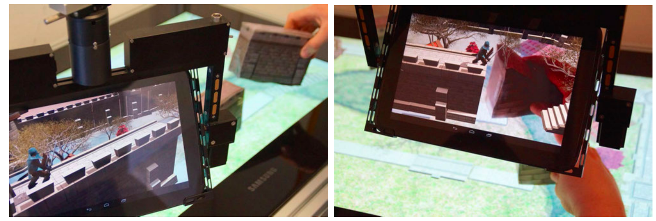

Figure 12. A mixed reality game based on *Hover Pad* requires users to support an autonomously moving virtual character displayed on the tablet (a). The user places physical wall pieces on the surface to prevent the character from falling (b).

in order to keep it in view (see Figure 12a). The user is required to arrange tangible items on the interactive surface below. These objects serve as bridges or staircases, for example, and ensure that the virtual character does not fall down (see Figure 12b).

This application leverages the unique characteristic of *Hover Pad* to combine autonomous behavior with *self-actuated* movement, which cannot be achieved by manually moving spatial displays. This application shows that *direct interaction* can be implemented in various ways: here, tangible objects (tracked through markers attached to their bottom) allow users to react quickly to *Hover Pad*'s autonomous behavior and movement in space.

DISCUSSION

Our example applications implemented with *Hover Pad* highlight possible scenarios in which *autonomous and self-actuated* displays provide advantages over existing, manual systems proposed in related work. *Hover Pad*'s *autonomous and semi-autonomous movement* capabilities enable search & inspect interaction, e.g., in the physical object augmentation application and the educational anatomy explorer, as well as bookmark & recall, e.g., in the medical volumetric data explorer and the map explorer. In contrast to existing systems, users can focus on the information associated with respective points of interest, instead of having to navigate the tablet to those data points manually. Such physical self-actuated movement of a display is in particular of advantage, if an application includes an overview (e.g., a large display or a physical object) that is to be inspected in detail (i.e., using a spatial display). Furthermore, *Hover Pad* provides *hands-free interaction*. In contrast to related work on handheld displays, our *Hover Pad* hardware ensures that the display remains at the desired position and orientation, which enables users to study the displayed content and interact with it (e.g., on the primary display), or with augmented physical objects with both hands and without having to hold the display for prolonged periods of time. This hands-free interaction further has the advantage of providing visual stability, as *Hover Pad* fixates the display's five degrees of freedom when needed. These aspects also result in more fine-grained control of the position of the display, which is particularly important in high resolution data, such as volumetric medical data.

While tablets that are physically moved through an information space by users could theoretically support six *degrees of freedom* (DoF), these 6 DoF would need to be supported by tracking methods as well. Thus, even systems that combine top-projection with paper surfaces (e.g., [27]) likely only support 5.5 DoF, because projection from underneath is hard to achieve. Yet, conceptually, these systems have been shown to be sufficient for exploring the information space. Our current hardware setup provides 5 DoF (i.e., *roll* is not supported), but provides the advantages of hands-free interaction and visual stability noted before. Considering the number of available degrees of freedom of supported movement, it appears that it depends heavily on the supported application how many degrees of freedom are required. For instance, Touch Mover [23] supports movement only along one dimension, yet it allows tactile exploring of volumetric objects. Additional degrees of freedom could even be a disadvantage for some applications. *Hover Pad* however, allowing for max. five degrees of freedom provides a high level of flexibility as all five degrees of freedom can be individually utilized or disabled depending on an application's requirements. This is not possible for approaches that require the user to manually control a handheld display.

Accordingly, accuracy of positioning a display is limited by the operating user in the case of handheld display based approaches. In contrast, systems such as *Hover Pad* that provide self-actuated position control are not limited in their positioning accuracy, as long as the employed hardware components can provide the required movement resolution. An analog coherence exists for the operational space: handheld and user operated approaches cannot be upscaled to larger screen and projection sizes as easily as it is theoretically possible for approaches based on self-actuated movement, because movement in user-operated approaches is restricted by users' body height and arm length. Self-actuated movement also constitutes a fundamental advantage over user-operated approaches: such a system can lead users to points of interest without discarding the spatial relations. For instance, as illustrated with the *physical object augmentation and exploration* application, users can be guided through an information space, which can reduce search time and increase an understanding of spatial relations. In addition to self-actuated movement, system-driven and autonomous position control, for instance, used in games, allows users to interact with *Hover Pad* in a novel way that is a clear distinctive feature regarding user-operated approaches found in related work.

Considering the information fidelity provided, *Hover Pad* provides a high definition display as a spatial display. This level of fidelity has not been achieved by previous approaches in connection with self-actuated movement in space.

The presented implementation of *Hover Pad* constitutes an initial step towards truly *hovering* displays. Our approach enables investigation of relevant interaction patterns, as well as prototyping of applications for such displays. Our prototype has a number of limitations, which we plan to address in future work.

When using a spatial display that is self-actuated, the movement of the display is limited the hardware capabilities (motor speed, stability etc.). Hence, by using handheld spatial displays, users can enjoy a more *responsive* and *immediate* display position control as there is no delay. On the other hand, when using such powerful hardware that is capable in terms of speed to mimic a handheld display, safety aspects have to be considered which is not necessary for *Hover Pad* as the mechanical forces applied by the currently used motors are too weak to be considered dangerous.

In the current prototype, the display swings slightly for a short moment after reaching a desired 3D coordinate with autonomous movement, which is caused by the tension forces acting on the telescope bar. For future iterations, we plan to enhance the rigidness of the telescope bar with advanced materials (e.g., carbon composites) and reduce the weight of the display unit to address this issue. Another approach would be the utilization of a robot arm.

The current prototype, due to its static crane construction, cannot support mobility, e.g., a display device that would follow a user while walking through a museum. Recently, drone- and quadcopter-based displays have been proposed (e.g., [22]). While those approaches may facilitate mobility, they would raise other challenges, e.g., in terms of accuracy and viewing stability. *Hover Pad* could serve as a versatile prototyping platform for applications for such mobile displays, while associated challenges are being addressed.

CONCLUSIONS & FUTURE WORK

We introduced our vision of displays that can move autonomously and semi-autonomously in mid-air to navigate through three-dimensional information spaces. We discussed the potential of this new class of devices and identified relevant interaction patterns enabled by them. We presented *Hover Pad* as a prototype system and framework to explore *hovering* self-actuated and autonomous displays. Our approach enables semi-autonomous control of hovering displays, including hands-free interaction and visual stability.

With our *Hover Pad* system providing rich and diverse interaction and application possibilities, we were interested in understanding how different ways of interaction support diverse classes of tasks. For this purpose, we realized a diverse set of example applications highlighting *Hover Pad*'s capabilities. In future work, we plan to further assess the different interaction techniques and motion control options of *Hover Pad*. Further, we are planning to investigate the potential and impact of *multiple* autonomous and self-actuated displays in an environment on user interaction. Finally, we plan to explore the effect of the presence of additional screens (e.g., the interactive surface) in the vicinity of the self-actuated display on the users' understanding of spatial relations in volumetric data and information spaces.

ACKNOWLEDGMENTS

We would like to thank Alexander Vogel, Johannes Augsburger, and all involved members of the scientific workshop at Ulm University for their support. This work was conducted within the Emmy Noether research group "Mobile Interaction with Pervasive User Interface" and the Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" both funded by the German Research Foundation (DFG).

REFERENCES

1. Alrøe, T., Grann, J., Grönvall, E., Petersen, M. G., and Rasmussen, J. L. Aerial tunes: exploring interaction qualities of mid-air displays. In *NordiCHI '12*, ACM (2012).
2. Baur, D., Boring, S., and Feiner, S. Virtual projection: exploring optical projection as a metaphor for multi-device interaction. In *CHI '12*, ACM (2012).
3. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93*, ACM (1993).
4. Boring, S., Baur, D., Butz, A., Gustafson, S., and Baudisch, P. Touch projector: mobile interaction through video. In *CHI '10*, ACM (2010).
5. Breazeal, C., Wang, A., and Picard, R. Experiments with a robotic computer: Body, affect and cognition interactions. In *HRI '07*, ACM (2007), 153–160.
6. Chen, X. A., Marquardt, N., Tang, A., Boring, S., and Greenberg, S. Extending a mobile device's interaction space through body-centric interaction. In *MobileHCI '12*, ACM (2012).
7. Fitzmaurice, G. W. Situated information spaces and spatially aware palmtop computers. *Commun. ACM 36*, 7 (July 1993).
8. Fitzmaurice, G. W. *Graspable User Interfaces*. PhD thesis, Graduate Department of Computer Science, University of Toronto, 1996.
9. Hashimoto, S., Suzuki, R., Kamiyama, Y., Inami, M., and Igarashi, T. World locomotion. `http://s-hashimoto.org/?page_id=73&lang=en`, 2013. (Last access: 2014/06/30).
10. Hörtner, H., Gardiner, M., Haring, R., Lindinger, C., and Berger, F. Spaxels, pixels in space. `http://www.aec.at/futurelab/files/2012/07/spaxels_revision_final_20120615.pdf`. (Last access: 2014/04/16).
11. Izadi, S., Hodges, S., Taylor, S., Rosenfeld, D., Villar, N., Butler, A., and Westhues, J. Going beyond the display: a surface technology with an electronically switchable diffuser. In *UIST '08*, ACM (2008).
12. Konieczny, J., Shimizu, C., Meyer, G., and Colucci, D. A handheld flexible display system. In *Visualization, 2005. VIS 05. IEEE* (2005).
13. Lee, J., Post, R., and Ishii, H. ZeroN: mid-air tangible interaction enabled by computer controlled magnetic levitation. In *UIST '11*, ACM (2011).
14. Marquardt, N., Jota, R., Greenberg, S., and Jorge, J. A. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *INTERACT'11* (2011).
15. Marshall, M., Carter, T., Alexander, J., and Subramanian, S. Ultra-tangibles: creating movable tangible objects on interactive tables. In *CHI '12*, ACM (2012).
16. Nowacka, D., Ladha, K., Hammerla, N. Y., Jackson, D., Ladha, C., Rukzio, E., and Olivier, P. Touchbugs: actuated tangibles on multi-touch tables. In *CHI '13*, ACM (2013).
17. Pangaro, G., Maynes-Aminzade, D., and Ishii, H. The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. In *UIST '02*, ACM (2002).
18. Patten, J., and Ishii, H. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *CHI '07*, ACM (2007).
19. Pedersen, E. W., and Hornbaek, K. Tangible bots: interaction with active tangibles in tabletop interfaces. In *CHI '11*, ACM (2011).
20. Rosenfeld, D., Zawadzki, M., Sudol, J., and Perlin, K. Physical objects as bidirectional user interface elements. *Computer Graphics and Applications* (2004).
21. Schmidt, D., Seifert, J., Rukzio, E., and Gellersen, H. A cross-device interaction style for mobiles and surfaces. In *DIS '12*, ACM (2012).
22. Schneegaß, S., Alt, F., Scheible, J., Schmidt, A., and Su, H. Midair displays: Exploring the concept of free-floating public displays. In *CHI '14 Extended Abstracts* (2014).
23. Sinclair, M., Pahud, M., and Benko, H. TouchMover: Actuated 3D Touchscreen with Haptic Feedback. In *ITS '13*, ACM (2013).
24. Spindler, M., Büschel, W., and Dachselt, R. Use your head: tangible windows for 3D information spaces in a tabletop environment. In *ITS '12*, ACM (2012).
25. Spindler, M., Martsch, M., and Dachselt, R. Going beyond the surface: studying multi-layer interaction above the tabletop. In *CHI '12*, ACM (2012).
26. Spindler, M., Tominski, C., Schumann, H., and Dachselt, R. Tangible views for information visualization. In *ITS '10*, ACM (2010).
27. Steimle, J., Jordt, A., and Maes, P. Flexpad: highly flexible bending interactions for projected handheld displays. In *CHI '13*, ACM (2013).
28. Tsang, M., Fitzmaurice, G. W., Kurtenbach, G., Khan, A., and Buxton, B. Boom chameleon: simultaneous capture of 3d viewpoint, voice and gesture annotations on a spatially-aware display. In *UIST '02*, ACM (2002).
29. Ullmer, B., and Ishii, H. The metadesk: Models and prototypes for tangible user interfaces. In *UIST '97*, ACM (1997), 223–232.
30. Weiss, M., Remy, C., and Borchers, J. Rendering physical effects in tabletop controls. In *CHI '11*, ACM (2011).
31. Weiss, M., Schwarz, F., Jakubowski, S., and Borchers, J. Madgets: actuating widgets on interactive tabletops. In *UIST '10*, ACM (2010).
32. Wilson, A., Benko, H., Izadi, S., and Hilliges, O. Steerable augmented reality with the beamatron. In *UIST '12*, ACM (2012).
33. Yee, K.-P. Peephole displays: pen interaction on spatially aware handheld computers. In *CHI '03*, ACM (2003).