Appeared in the Proceedings of the ACM Symposium on User Interface Software and Technology (UIST) 2006

Interactive Environment-Aware Display Bubbles

Daniel Cotting, Markus Gross
Computer Graphics Laboratory
ETH Zurich

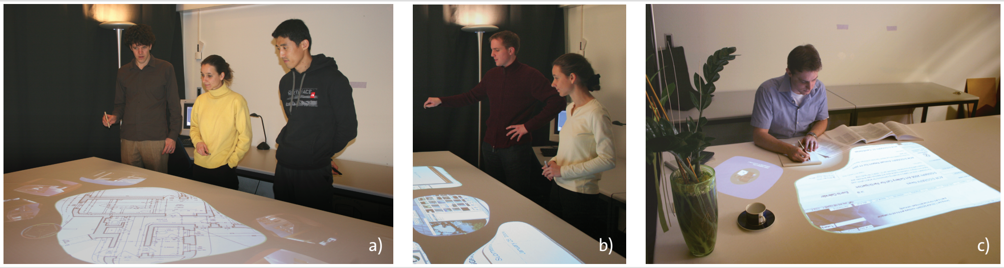

Figure 1: Interactive environment-aware display bubbles can be instantiated anywhere on demand. Their freeform shapes, the displayed content and the illumination can freely be redefined using a laser pointer. The novel display metaphor is well suited for a wide range of use cases and applications including: a) Collaborative design, engineering, modeling and visualization. b) Presentation meetings, edutainment and brainstorming sessions. c) Single-user large scale workspaces in office environments.

ABSTRACT

We present a novel display metaphor which extends traditional tabletop projections in collaborative environments by introducing freeform, environment-aware display representations and a matching set of interaction schemes. For that purpose, we map personalized widgets or ordinary computer applications that have been designed for a conventional, rectangular layout into space-efficient bubbles whose warping is performed with a potential-based physics approach. With a set of interaction operators based on laser pointer tracking, these freeform displays can be transformed and elastically deformed using focus and context visualization techniques. We also provide operations for intuitive instantiation of bubbles, cloning, cut & pasting, deletion and grouping in an interactive way, and we allow for user-drawn annotations and text entry using a projected keyboard. Additionally, an optional environment-aware adaptivity of the displays is achieved by imperceptible, realtime scanning of the projection geometry. Subsequently, collision-responses of the bubbles with non-optimal surface parts are computed in a rigid body simulation. The extraction of the projection surface properties runs concurrently with the main application of the system. Our approach is entirely based on off-the-shelf, low-cost hardware including DLP-projectors and FireWire cameras.

ACM Classification: H.5.2 [User Interfaces], I.3.2 [Graphics Systems], I.3.6 [Methodology and Techniques], I.3.1 [Hardware Architecture], I.4.1 [Digitization and Image Capture], I.4.8 [Scene Analysis]

General Terms: Design, algorithms.

Keywords: Tabletop, adaptive displays, focus and context, interaction, imperceptible structured light, projectors.

1 INTRODUCTION

Computer technology is increasingly migrating from traditional desktops to novel forms of ubiquitous displays on tabletops and walls of our environments. This process is mainly driven by the desire to lift the inherent limitations of classical computer and home entertainment screens, which are generally restricted in size, position, shape and interaction possibilities. There, users are required to adapt to given setups, instead of the display systems continuously accommodating the users' needs and wishes. Recent research has succeeded in alleviating some of the restrictions [37, 7, 16, 12], but the resulting displays are still confined to rectangular screens, do not tailor the displayed information to specific desires of users, and generally do not provide a matching set of multi-modal interaction techniques.

By introducing so called interactive environment-aware display bubbles, we aim at significantly enhancing the flexibility, interactivity and adaptivity of displays: Our new metaphor allows users to freely and independently set their displays anywhere on demand, and additionally define their content and the surrounding illumination on the working space. We focus on projected tabletop displays, which lend themselves very well to implement natural user interfaces [41], since desks and tables are used extensively in everyday life to work with physical items such as paper, books and pens. We address both information presentation and interaction issues in a complete framework for smart, space-efficient, freeform displays, which facilitate rich interpersonal communication and collaboration. In addition to providing effective single-user large scale workspaces, our framework is very well suited for interactive multi-user presentation, visualization, design and modeling tasks on tabletops in highly collaborative conference rooms and office settings (cf. Figure 1). The underlying system is entirely based on low-cost, off-the-shelf hardware.

Our core contributions include novel bubble-based focus and context display techniques, a matching set of intuitive interaction schemes, and an environment-aware adaptivity of the bubbles using per-pixel light control (we prefer the expression *environment-aware* over *context-aware* to prevent confusion with focus and *context* techniques).

The realization of our novel techniques draws upon many different technologies in a large variety of research areas, whose most relevant previous work is summarized in Section 2. A brief overview of the hardware system and its setup procedure is given in Section 3. Subsequently, our novel algorithms for bubble generation are presented in Section 4, our display adaptivity concept in Section 5, followed by the matching interaction techniques in Section 6. We conclude the paper with a presentation of current results, a discussion, and an outlook on possible future extensions.

2 RELATED WORK

Our interactive display bubbles rely on projection technology, since currently no other technology provides a way nearly as competitive and effective to build flexible, large and high-resolution screens. Therefore, our work draws upon research from the large tiled displays community, where multiple projections are aligned and blended to create wide-area screens [48, 10, 39, 20]. Once these systems are set up, they all focus on achieving a static display, without considering the projection of dynamic on-demand screens in flexible user-defined areas.

User-Defined Displays. Even though the PixelFlex system [39] is able to automatically rearrange its projection layout, it was not designed to provide on-demand screens on arbitrary surfaces. The IBM Research steerable projections [37] and subsequent systems inspired by them [51, 6, 7, 16] are notable exceptions and allow dynamic, spatially reconfigurable displays. However, they are restricted to rectangular displays and single projectors, require motorized equipment, and generally environment-aware adaptivity and scalability are not considered.

Tabletop Systems. Following the inspiring tabletop environment presented in Wellner's DigitalDesk system [55], many novel tabletop systems have been developed recently. Lumisight [24] introduces a view-dependent, space-efficient tabletop setup, which allows personalized contents to be displayed for up to four users simultaneously. However, the display geometry remains static and the installation is rather cumbersome and fairly expensive. Providing a larger working area, the Escritoire system [2] implements an interactive foveal display for a single user per working site, which consists of a rectangular desk. In a comparable setup, DTLens [18] allows multiple users to zoom into certain parts of the display using rectangular stretching regions. Also using restricted rectangular working spaces, Watson et al. [54] explore collaborative group interaction in an educational classroom setting, and the EDC system [1] studies shared understanding and informed participation. In a similar setting, the Caretta system [47] integrates personal and shared spaces to support face-to-face collaboration, while BUILD-IT [17] provides a planning tool allowing a group of people to interact with objects in a virtual scene using real bricks. Considerably enlarging the available working space, Augmented Surfaces [44] allow users to project displays onto tables or walls as a spatially continuous extension of their portable computers. As a major limitation, all aforementioned approaches rely on custom-designed applications. By contrast, Topos [35] displays arbitrary applications at desired locations by wrapping content into editable, rectangular objects of a 3D window manager. Abandoning such rectangular settings, Vernier et al. present novel visualization techniques and layout schemes on circular tabletop displays [52], however these are restricted to dedicated applications and still do not permit arbitrary shapes.

Interactivity in tabletop systems is generally achieved using cameras [24, 43], tracked devices [2, 52] or sensors integrated into the table surface [15, 42].

Focus and Context. Focus and context techniques aim at optimally presenting large amounts of information in a given restricted viewing area by focussing on relevant data while only hinting at the context information of the remaining data. Various techniques have been published for stretching and distorting spaces to produce effective visualizations. Research spans a wide range encompassing polyfocal projection [23], fisheye views [19], distortion oriented presentation [32], focus and context [31], detail-in-context [25] and nonlinear magnification [26]. Despite the many publications, to the best of our knowledge no approach for mapping rectangular content to an arbitrary freeform shape has been presented yet.

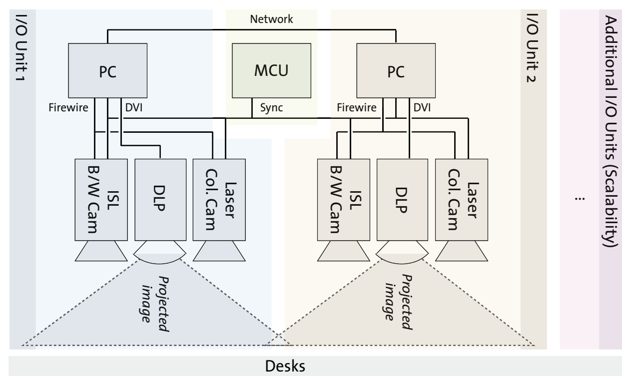

Figure 2: Configuration of the tabletop setup for interactive environment-aware display bubbles.

3 HARDWARE SETUP

Our setup for creating interactive environment-aware display bubbles consists of scalable, networked units (see Figure 2), which can be individually oriented to cover the desired working space, usually consisting of several desks. In our implementation, we rely on two modular I/O units each including a conventional DLP projector (Infocus X3) connected to a PC and a total of two color cameras for laser pointer tracking and two grayscale cameras (Point Grey Dragonfly) for the extraction of display surface properties using imperceptible structured light (ISL, see Section 5.1). Additionally, we use a microcontroller unit (MCU, Toshiba TMP92FD54) as a synchronization source for the cameras and the graphics boards (NVIDIA Quadro FX3000G). The total cost of our prototype amounts to less than US$ 7,000 without ISL functionality and to approximately US$ 13,000 with the ISL-based environment-aware display adaptivity. Due to the modular hardware and software design and an optimized distributed display approach [12], the system is easily scalable to additional units without imposing any additional network load.

To achieve a seamless alignment of our display projections, all devices must be calibrated intrinsically and extrinsically with relation to each other. For this purpose, we calibrate both cameras and projectors by an approach based on a propagation of Euclidean structure [4] using point correspondences embedded into binary patterns [12]. Using the same patterns, initially presented by Vuylsteke and Oosterlinck [53], we reconstruct the projection surface.

4 DISPLAY BUBBLES

The following sections present our techniques to generate and deform display bubbles using a warping approach based on focus and context visualization techniques and hyperbolic projection.

4.1 Warping

Projection surfaces, especially in tabletop settings, are not always guaranteed to provide an adequately large, uniform and continuous display area. A typical situation in a meeting room or office environment consists of cluttered desks, which are covered with many objects, such as books, coffee cups, notepads and a variety of electronic devices. There are several strategies for a projected display to deal with objects on a desk: First, ignore them and therefore get distorted images. Second, integrate the objects into the display scene as part of the projection surface in an intelligent way [11], unfortunately often resulting in varying reflection properties. Or third, be aware of the clutter and do not project imagery onto it.

To achieve the best possible control of projection quality, we opt for the third solution and maximize surface usage by allowing displays to smoothly wind around obstacles in a freeform manner. As opposed to distorted projections resulting from ignoring objects on the desks, the deformation is entirely controllable and modifiable by the user, providing her maximum flexibility over the display appearance. In presence of clutter and obstacles, free-form bubbles permit a much tighter geometric packing than rectangular displays and thus permit to use limited display space more effectively.

However, even on empty tabletops our techniques can be used to tile and arrange a very large number of displays in a space-efficient manner while taking into account a user-defined information focus, therefore maximizing tabletop real-estate. Contrary to the usage of virtual desktops [13], or the cascading, tiling or overlapping of traditional rectangular windows, more relevant information can be visualized in parallel while additionally keeping helpful context data. Furthermore, the flexibility of the resulting bubbles allows for user-defined information and display compositing in the spirit of WinCuts [49] and the User Interface Façades [46].

Ideally, in the future the design of graphical user interfaces should be adapted to the requirements of non-rectangular displays. However, in the meantime, most common applications remain rectangular, and we therefore propose our focus and context warping approach to make our bubbles generally applicable to the wide range of existing applications. Our bubble warping is, to the best of our knowledge, the first approach to map rectangular displays to arbitrary shapes. In conjunction with an intuitive focus and context interaction technique, which provides enhanced virtual screen space, our mapping represents a powerful new display paradigm. In our experiments, it has provided very satisfactory experiences with a wide range of traditional rectangular applications.

Problem Definition. Given an arbitrary closed shape S, where display content is optimally placed on the projection geometry (cf. Figure 3), we want to compute a display mapping $M\colon (x, y) \rightarrow (u, v)$ of the original rectangular screen content R, such that: a) The defined focus shape S displays enclosed content with maximum fidelity, i.e. least-possible distortion and quality loss. b) The remaining content is smoothly arranged around the shape S in a controllable context area C. The boundaries of R, S, and C are given by ∂R, ∂S, and $\partial C_1 \cup \partial C_2$ respectively.

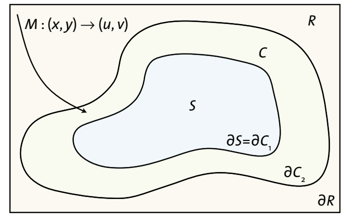

Figure 3: Given a focus shape S, the content of the original rectangle R is warped into the area composed of S and a surrounding context area C using a mapping $M\colon (x, y) \rightarrow (u, v)$.

In our case, for each pixel $P(x, y)$ of the original screen content R its final position (u, v) under the aforementioned constraints has to be found. This problem corresponds to image warping, which has been studied very well in computer graphics (e.g. [56, 5, 34]). Most traditional approaches to image warping utilize smooth geometric deformations guided by interactively set landmarks. Instead, we have developed a physically-based method which guarantees a smooth deformation while allowing us to elegantly preserve the specific boundary conditions imposed by our application, including potentially concave regions of the focus shape S. Although other mappings are possible as well, our potential-based approach seems to be a conceptually clean and sufficiently simple approach providing a smooth, accurate and stable warping without ad-hoc empirical methods.

Potential Field. Our method constrains the mapping M to follow field lines in a charge-free potential field defined on the projection surface by two electrostatic conductors set to fixed, but different potentials V_S and V_R, where one of the conductors encompasses the area enclosed by S and the other one corresponds to the border of R. Without loss of generality, we assume that $V_S > V_R$.

The first step in computing the desired mapping involves the computation of the 2-dimensional potential field $V(x, y)$ of the projection surface parametrization, which is given as the solution of the Laplacian equation

$$\Delta V(x, y) \equiv \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0, \tag{1}$$

with the inhomogeneous boundary conditions

$$V(\partial S) = V_S \text{ and } V(\partial R) = V_R. \tag{2}$$

Since analytical methods for computing the potentials in regions of arbitrary shapes do not exist, we resort to a numerical method and compute the potential using a finite difference discretization of the Laplacian on a regular, discrete $M \times N$ grid of fixed size. We employ iterative successive overrelaxation with Chebyshev acceleration [38]. Figure 4 a) shows the resulting potential field for a given freeform shape. Note that the Laplacian equation can be solved very efficiently on regular grids and that the computational grid can be chosen smaller than the screen resolution, typically 100 × 75 in our examples.

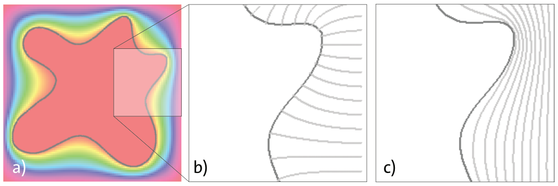

Figure 4: a) Computed discrete potential field (color coded). b) Computed field lines in a subarea. c) Changes in the border ∂C_2 of the context area C in a subarea for varying V_Δ.

Field Lines. When determining the position (u, v), where a certain pixel $P(x, y)$ of the original rectangle R should be warped to, we follow the corresponding field lines computed from the discrete potential values towards the area S. We use a simple Euler integration method to trace the field lines. These exhibit many desired properties, such as absence of intersections, smoothness and continuity except at singularities such as point charges, which fortunately cannot occur in our charge-free region. Figure 4 b) illustrates a selected set of field lines.

Context Handling. To determine the exact location, where each pixel of the original rectangular display will be warped to, we draw upon research from focus and context visualization techniques, in particular from the area of hyperbolic projection [28, 31]. Every pixel inside S keeps its location and is thus part of the focus area, which displays the enclosed content with maximum fidelity and least-possible quality loss. For every pixel $P(x, y) \in R \setminus S$, we determine its potential V_P, and given a user-defined parameter $V_\Delta \in (0,\infty)$, the pixel $P(x, y)$ is moved along its field line to the position (u, v) with potential

$$V_M = V_S - \frac{V_S - V_P}{\sqrt{\left(\frac{V_S - V_P}{V_\Delta}\right)^2 + 1}}, \quad (3)$$

which corresponds to a hyperbolic projection of the potential difference $V_S - V_P$ between the point P and the focus area S, as shown in Figure 5.

The hyperbolic projection has some interesting properties, in that pixels near S are focused, while an infinite amount of space can be displayed within an arbitrary range C defined by V_Δ. Note that Equation 3 guarantees that no seams are visible between the focus and the context area, and thus ensures visual continuity.

Shape Continuum. The resulting mapping provides a smooth arrangement of $R \setminus S$ around the focus shape S in an intuitive context area C, which can be controlled by a user-defined parameter V_Δ influencing its border ∂C_2 (see Figure 4 c)). In order to warrant maximum flexibility, our displays can gradually transition from traditional rectangular screens over focus and context freeform bubbles to simple clipped shapes: If $V_\Delta \to \infty$, the original rectangular shape is maintained, and when the user parameter V_Δ is converging to 0, the context area disappears and the warping corresponds to a clipping with S as a mask. We thus do not restrict the user to a specific type of display, but allow her to freely explore the entire space of possible configurations. We are not aware of any other display metaphor supporting ordinary rectangular computer applications and offering as many potential degrees of freedom to the users.

Constrained Width. If a constrained width of the context area C is required, geometric distance along the field line can be used instead of the potential difference during the hyperbolic projection. Here, the distances are computed by adding the spatial differences while tracing the field lines using Euler integration. Each pixel $P(x, y) \in R \setminus S$ with distance D_{PS} to S along its field line is therefore mapped to the point on the line at distance

$$D_M = \frac{D_{PS}}{\sqrt{\left(\frac{D_{PS}}{D_\Delta}\right)^2 + 1}}, \quad (4)$$

where the user-defined parameter D_Δ specifies the width of the context area C along the field lines. Figure 6 compares resulting context areas of the potential and distance based approaches.

Optimization. In order to achieve interactive rates, not every pixel's mapping is calculated, but rather discrete locations of a warping grid are evaluated and the remaining pixels interpolated through hardware-accelerated texture mapping (see Table 1 for warping performance measurements). This allows for interactive recomputation of the warping such as being needed for bubble deformation, and also performs high-quality antialiasing. This feature is of great help to attenuate aliasing artifacts arising when rescaling bubbles with fineprint text. Figure 7 illustrates various bubbles resulting from our warping approach.

Special Cases. For convex shapes S^{cv} we have successfully applied a simpler distance based warping approach, where the field lines are replaced by rays through the center of mass of S^{cv}.

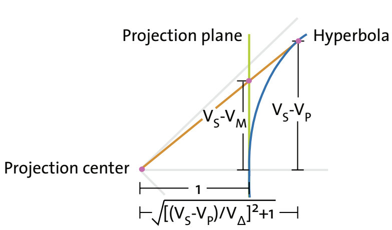

Figure 5: Hyperbolic projection of the potential difference $V_S - V_P$ between a given point *P* and the focus area *S*.

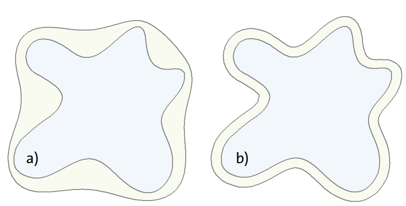

Figure 6: Shape of the context area using potential difference (a) and geometric distance (b) for hyperbolic projection.

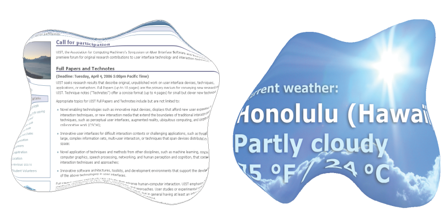

Figure 7: Bubbles warped with our approach using hyperbolic projection and potential differences ($V_\Delta = 0.3$).

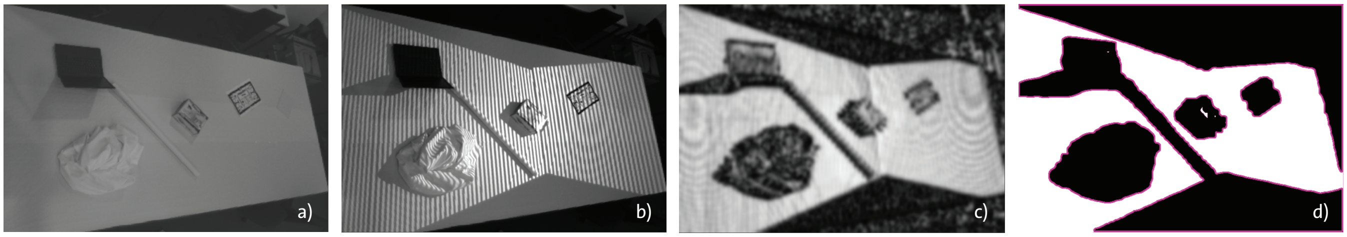

Figure 9: Analysis of the projection surface properties for the environment-aware display adaptivity: a) Snapshot of the scene taken from a camera. b) Captured pattern image *I*. Note that in this setting the projector frustums do not cover the entire table. c) Magnitude of the Gabor filter response $G \otimes I$. d) Computed parts of the environment not admissible for display (in black).

4.2 Distributed Display Content

In addition to a warping approach of the bubbles, a method for display content generation is required in our framework, which supports multiple projectors in a distributed environment.

We extend a previous approach for distributed display [12], which relies on an efficient and scalable transmission based on the Microsoft RDP protocol, with support for the cross-platform VNC protocol [45], user-defined widgets, and lighting components. First, the RDP and VNC protocols allow content of any source computer to be visualized remotely without requiring a transfer of applications or their data to the nodes of the display bubble system. As a major advantage, this allows us to include any laptop as a display input in our collaborative meeting room environment. Second, widgets represent small self-contained applications giving the user continuous, fast and easy access to a large variety of information, such as timetables, communication tools, forecast or planning information. Third, as a complement to the protocols generating the actual display content, our lighting components allow users to steer and command bubble-shaped light sources as a virtual illumination in their tabletop augmented reality environments.

Each content stream, consisting of one of the aforementioned protocols, can be replicated to an arbitrary number of bubbles, which can concurrently be displayed by multiple of our I/O units (see Section 3), as illustrated in Figure 8. This versatility easily allows multiple users to collaborate on the same display content simultaneously.

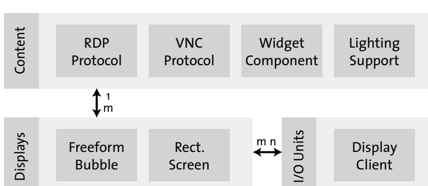

Figure 8: Content of various protocols can be replicated to several displays distributed among multiple I/O units.

In a multi-projector system, corresponding images from different projectors need to be blended to create appealing displays at frustum intersections. For that purpose, we use an OpenGL-based GPU-procedure resulting in pixel-level alpha masks, which can directly be applied to projections onto arbitrary surfaces [12].

5 ENVIRONMENT-AWARE DISPLAY ADAPTIVITY

Given the projection geometry of our setup (cf. Section 3), we optionally check the suitability of the surface for display by continuously analyzing its reflection properties and its depth discontinuities, which have possibly been introduced by new objects in the environment. Subsequently, the bubbles are moved into adequate display areas by computing collision responses with the surface parts, which have been classified as not admissible for display.

5.1 Projection Surface Properties

In order to determine the display surface properties of a scene as shown in Figure 9 a), we continuously project a static stripe pattern using an imperceptible embedding which controls the appearance of the projection surface during a triggered camera exposure [40, 11, 12]. We thus actively include the projector into the determination of suitable surfaces, unlike previous approaches [51]. Since the pattern can be considered a spatially periodic signal with a specific frequency, its detection can be performed by applying an appropriately designed Gabor filter G to the captured image I of the reflected stripes, illustrated in Figure 9 b) [27]. As shown in Figure 9 c), the magnitude of the filter response $G \otimes I$ will be large in continuous surfaces with optimal reflection properties, whereas poor or non-uniform reflection and depth discontinuities will result in smaller filter responses due to distortions in the captured patterns. Refer to Kim et al. [27] for a detailed analysis of the detectable range of depth images. After applying an erosion filter to the Gabor response and thresholding the resulting values, the non-optimal surface parts of the environment can be determined, as illustrated in Figure 9 d).

5.2 Collision Handling

The display bubbles are continuously animated using a simple, 2D rigid body simulation [3]. The non-optimal surface parts computed in Section 5.1 are used as collision areas during collision detection computations of the bubbles. Colliding bubbles are repelled by the areas until no more collisions occur. During displacement of the bubbles, inter-bubble collision detection and response is performed continuously in an analog way.

5.3 Shadow Avoidance

Since shadows result in a removal of the projected stripe pattern and therefore in a low Gabor filter response, shadow areas are classified as collision areas. Thus, bubbles continuously perform a shadow avoidance procedure in an automatic way, resulting in constantly visible screen content.

6 INTERACTION

To accommodate multiple users and to allow for encumbrance-free interaction, our novel display bubbles can no longer rely on traditional input devices like keyboards and mice. For this reason, we have developed a comprehensive set of interaction operations relying on a laser pointer tracking approach, where Kalman-filtered 3D laser pointer paths are reconstructed from real-time camera streams and the resulting coordinates are mapped to the appropriate bubbles. In contrast to sensor-based surfaces [15, 42] and pen-based tracking [2], we do not require any invasive or expensive equipment. Furthermore, unlike bare fingers as input pointers, laser dots can efficiently and reliably be tracked, do not interfere with our environment-aware display adaptivity (see Section 5), and have a very large range of operation. This allows users to interact with the displays and their content, both in a remote fashion and in the users' vicinity.

At any time, users can intuitively toggle between the available operation modes using a hierarchical on-screen pie menu [21], which can be activated by triggering the pointer at empty tabletop locations (see Figure 15 a)). Circular popup menus are well suited for tabletop settings, since their visual appearance provides an orientation-independent view of the selectable options. Besides noting an increase in efficiency (see e.g. [9]), test users have found the menus to be very intuitive and flexible: Pie menus offer visual feedback for inexperienced users, while at the same time providing gesture-like shortcuts for advanced users [30].

In the following sections, we present a selected set of bubble operations to efficiently work with the displays. Note that new operations can be added easily and are partly subject to future work. We refer to the accompanying video for further explanations about the navigation using our pie menu and a live presentation of various bubble operations.

6.1 Warping Operations

Warping parameters of a currently selected bubble can be changed dynamically. We allow the curve defining the focus area S to be deformed, the potential V_Δ to be modified, the rectangle R to be realigned with respect to S, and the content which appears in focus to be interactively changed.

Freeform Editing. The self-intersection free curves, which define the focus areas of the screen bubbles, can be manipulated by the user in a smooth, direct, elastic way (cf. Figure 10 a)). Given a pointer position $\boldsymbol{L}_0 = (\tilde{u}_0, \tilde{v}_0)$ in the screen geometry parametrization at the beginning of a freeform editing step and a position $\boldsymbol{L}_t = (\tilde{u}_t, \tilde{v}_t)$ at time $t > 0$, the deformed positions of the curve points $\boldsymbol{P}_i$ are given by

$$\boldsymbol{P}_i'(t) = \boldsymbol{P}_i + \mathrm{e}^{-\frac{(\boldsymbol{P}_i - \boldsymbol{L}_0)^2}{2\sigma(t)^2}} \cdot (\boldsymbol{L}_t - \boldsymbol{L}_0), \tag{5}$$

where $\sigma(t)$ specifies the Gaussian falloff of the smooth displacement kernel and is defined as $\sigma(t) = \|\boldsymbol{L}_t - \boldsymbol{L}_0\|$. This variable factor provides a simple form of adaptivity of the edit support with respect to the magnitude of displacement of an editing step at time t, similar in spirit to the influence region growing of the as-rigid-as-possible curve editing presented by Igarashi et al. [22]. The user can dynamically move the pointer and preview the new shape of the focus area in real-time until she is satisfied with its appearance. After the user acknowledges an editing step at a certain time t' by releasing the laser pointer, the coordinates $\boldsymbol{P}_i'(t')$ are applied and the curve is resampled if required. Subsequently, the new warping parameters are computed for the newly specified focus shape (cf. Figure 13). It is needless to say that other curve editing schemes, such as control points, could be accommodated easily.

Rectangle Alignment. If the position of a bubble has to remain constant, but the content should be scaled, translated and rotated, then the rectangle R can be zoomed, moved or spun around the shape S as shown in Figure 10 b). If required, we continuously adapt the rectangle's size so that it entirely contains S.

Focus Change. We allow the user to dynamically redefine the contents of the focus area in real-time by letting her move the texture of the original screen content R by a displacement vector

$$\boldsymbol{v} \sim \boldsymbol{L}_t - \boldsymbol{L}_0, \tag{6}$$

where $\boldsymbol{L}_0 = (\tilde{u}_0, \tilde{v}_0)$ represents the laser pointer position in the screen geometry parametrization at the beginning of a focus change operation step and the position $\boldsymbol{L}_t = (\tilde{u}_t, \tilde{v}_t)$ corresponds to the position at time $t > 0$ (cf. Figure 10 c)). This allows the user to freely navigate around extensive content, as shown in Figure 14, and due to the inherent scrolling functionality also facilitates the exploration of large desktops where unused information or inactive applications can be parked in the context area. Switching from one information or application to another is then as easy as changing focus.

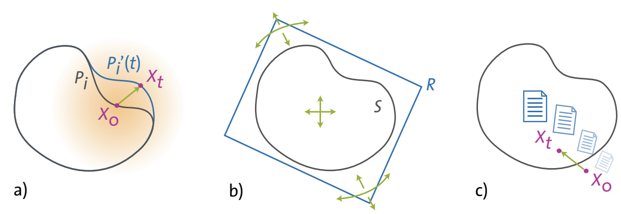

Figure 10: Warping operations. a) Bubble freeform editing step. b) Rectangle alignment. c) Focus change on content.

Potential Variation. As shown in Figure 4 c), the resulting shape of the context area of a bubble can be changed by adapting the user-defined potential parameter V_Δ, allowing a continuous change in bubble shape from the unwarped rectangular screen to the shape of the focus area (cf. Figure 12). This allows the user to continuously choose her favored representation according to her current tasks and preferences.

6.2 Bubble Arrangement

At the user's discretion, the bubbles can be transformed and arranged in various ways.

Affine Transformations. With the help of the laser pointer, the bubbles can be scaled, changed in aspect-ratio, rotated and translated to any new location on the projection surface. Additionally, we allow the bubbles to be pushed in our rigid body simulation framework by assigning them a velocity vector proportional to the magnitude of a laser pointer gesture.

Grouping. As a more elaborate arrangement operation, we allow multiple bubbles to be marked for grouping by elastic bonds [3], allowing the users to treat semantically related displays in a coupled way. After grouping, the linked bubbles are immediately gathering due to mutual spring forces (cf. Figure 15 b)).

6.3 Cardinality Change

The cardinality of the set of currently displayed bubbles can be changed in multiple ways as described in the following.

Instantiation. New bubbles can be created with the laser pointer by tracing a curve defining a new focus shape S (cf. Section 4.1). The rectangle R, which is required for the warping computation, is automatically mapped around this curve as a slightly enlarged bounding box. It can subsequently be aligned with the *Rectangle alignment* operation presented in Section 6.1, and the displayed content can be chosen with the *Content cycling* operation from Section 6.4. The *Instantiation* operation mode is automatically activated when no more bubbles are defined.

Cloning. A bubble can be cloned by dragging a copy to a new location. Examples of clones are shown in Figure 1 a) and b).

Deletion. Multiple bubbles can be marked for deletion by subsequently pointing at them. If no more bubbles are available, we automatically switch to the *Instantiation* operation mode. Note that we do not provide an explicit merging operation of multiple bubbles, since the *Deletion* operation renders this operation redundant.

Cut & Pasting. By pointing at one or multiple bubbles in a sequence, the user can mark a set of displays for a cut operation, which stores the affected bubbles into a persistent buffer, which can be pasted onto the projection surface an arbitrary number of times at any desired location.

6.4 Application Interface

Our framework also includes operation modes allowing to interact with the underlying display content (see Figure 16).

Mouse Navigation. Mouse events can be dispatched to the protocols being used for display content generation (see Section 4.2). For that purpose, the laser pointer location $\boldsymbol{L}_t = (\tilde{u}_t, \tilde{v}_t)$ in the screen geometry parametrization is transformed to bubble coordinates (u, v), then unwarped by $M^{-1}: (u, v) \rightarrow (x, y)$, while focus parameters are accounted for in order to recover the correct corresponding application or widget screen coordinates. Mouse locations at the border of the screens automatically initiate a scrolling of the image contents by dynamically adjusting the focus. To trigger events, we use a second laser modulation provided by our off-the-shelf two button dual mode pointer (see Figure 11).

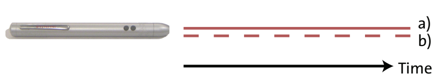

Figure 11: Mouse navigation using dual mode laser modulation: a) Pointing operation. b) Triggering operation.

Keyboard Tracing. For textual input in multi-user collaborative environments, we introduce shorthand-aided rapid keyboarding [29] into tabletop settings. Trajectories of words traced by the user on a configurable, optimized keyboard layout, which is overlaid on the bubble (cf. Figure 16), are recognized and matched to a large internal database using elastic matching with zero look-ahead not requiring any training [50]. Both shape and location information are considered, and if multiple word candidates remain, the user is given the option to select one from a list of most probable candidates. Due to the intuitive and deterministic nature of the input method, the user can gradually transition from visually-guided tracing to recall-driven gesturing. Therefore, after only a short training period, the approach requires very low visual and cognitive attention and offers high input rate compared to alternative approaches [8]. Additionally, in contrast to previous methods [33], it does not require any cumbersome input device like a glove. As further advantages, it provides a degree of error resilience suited for the limited precision of the laser pointer based remote interaction. Note that it is possible to use conventional (wireless) keyboards within our framework as well.

Annotation. Using the laser pointer, users can draw on the contents of bubbles to apply annotations, which are mirrored to all bubbles displaying the same content.

Content Cycling. The content of each bubble can be changed by cycling through a set of concurrently running protocols (cf. Section 4.2). This allows users to switch from one content to the next on the fly depending on the upcoming tasks, and also permits to swap contents between bubbles.

7 RESULTS

Based on the display metaphor presented in this paper, we have implemented interactive environment-aware display bubbles using off-the-shelf components in a standard meeting room environment (see Section 3). Both single-user settings and multi-user scenarios have been explored as shown in Figure 1 and illustrated in the accompanying video.

Multiple interaction steps of warping operations such as potential variation, freeform editing and focus change are illustrated in Figures 12, 13 and 14. We refer to the accompanying video for a better impression of the dynamics of these operations.

Timing measurements of the warping computation steps are summarized in Table 1. In our current implementation, we use a 100×75 potential field and a 40×30 warping grid for all examples. These settings provide adequate visual quality and allow all warping parameters to be updated in less than 120 ms. Note that a complete recomputation is only necessary as a final step of a freeform editing, potential variation, rectangle alignment and instantiation operation. All bubble operations have been carefully designed to give the user immediate realtime feedback on her interactions, e.g. interactive potential variation is achieved by continuously previewing the resulting warping shape using a reduced 20×15 grid, whose computation can be performed in 18 ms. Replacement of the simple Euler integration scheme with a more elaborate approach might further speed up warping computations.

A snapshot of the pie menu used for selecting bubble operations and the effect of grouping using elastic bonds are

Table 1: Timing measurements for the potential field relaxation and the field line tracing for varying grid sizes. Computation times are given in milliseconds for the first bubble in Figure 7 on a single Intel Pentium 4 3.0 GHz processor.

	Warping grid 20×15	*Warping grid* 40×30	*Warping grid* 60×45
Potential field 80×60	Relaxation: 35 Tracing: 15	Relaxation: 35 Tracing: 56	Relaxation: 35 Tracing: 121
Potential field 100×75	Relaxation: 51 Tracing: 18	Relaxation: 51 Tracing: 65	Relaxation: 51 Tracing: 141
Potential field 120×90	Relaxation: 74 Tracing: 20	Relaxation: 74 Tracing: 75	Relaxation: 74 Tracing: 162
Potential field 160×120	Relaxation: 123 Tracing: 26	Relaxation: 123 Tracing: 91	Relaxation: 123 Tracing: 201

shown in Figure 15. The coupled bubbles are depicted in the final resting state. A selected set of application interface operations, including annotation mode and keyboard tracing are presented in Figure 16, while our optional environment-aware adaptivity is illustrated in Figure 17. The extraction of the projection surface properties in our straightforward implementation, which has not yet been optimized, runs at interactive rates of 5 frames per second. We have tested our display metaphor with scenarios containing at least as many obstacles as shown in Figure 9 and have achieved very satisfactory results.

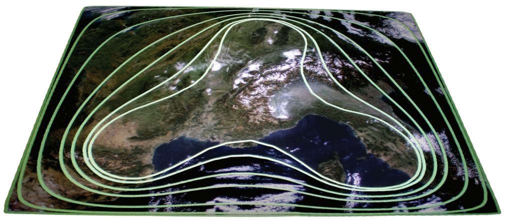

Figure 12: Changing the user-defined potential parameter V_Δ of a bubble (see Section 4.1) smoothly adapts the shape in a range from the unwarped rectangular screen to the shape of the focus area. To better convey the resulting warping, multiple snapshots of a display with varying values V_Δ have been composited in this illustration.

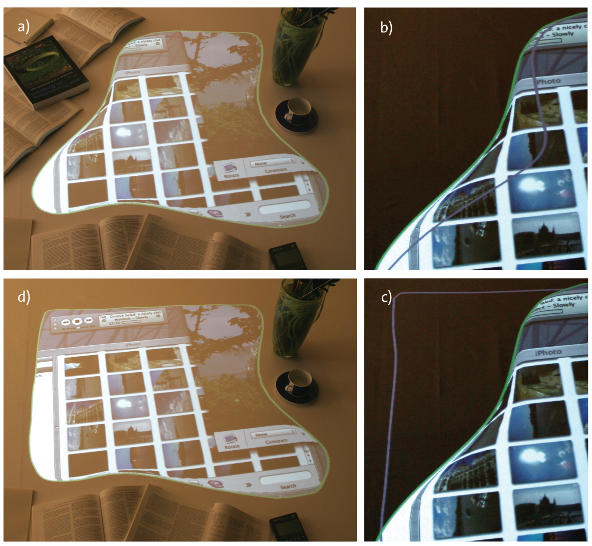

Figure 13: Freeform editing: a) Initial setting. b) Purple curve specifying the focus area of the initial setting. c) Curve after freeform editing. d) Resulting bubble after freeform editing.

Multiple use cases of our interactive environment-aware display bubbles are shown in Figure 1 and the accompanying video. Our display metaphor has already proven to be effective in a wide range of application scenarios. With support for content replication, multiple views, dynamic collision-aware display reconfiguration and multi-user remote interaction, our displays lend themselves very well for collaborative engineering, design and modeling, as some of the examples show. Additionally, the simple inclusion of any personal laptop as a display source makes our system suitable for brainstorming meetings or presentations. Furthermore, due to the integrated focus and context techniques the use for visualization of large-scale datasets for intelligence analysis or edutainment is straightforward. Last but not least, everyday single-user office workspaces, which often are cluttered as we all know from experience, greatly benefit from the enhanced display and illumination flexibility. In all use cases mentioned above, the optional environment-aware adaptivity and shadow avoidance can continuously guarantee optimal visibility of projected displays at any time.

Note that this paper intentionally focuses on the core idea of our display metaphor, its technical realization, a comprehensive set of matching interaction methods, as well as a sketch of possible application scenarios. A detailed evaluation and assessment of the suggested interaction metaphor and an extensive user study are still subject of ongoing work. However, while systematic user experiments have not yet been performed, informal feedback from users with diverse backgrounds has been very positive and encouraging: Users found the additional degrees of freedom of the display bubbles to be very pleasing and they soon appreciated the new flexibility in collaborative tasks. Due to its deterministic and foreseeable behavior, the proposed bubble implementation was favored by all participants over a setting where bubbles autonomously flow to different places. Surprisingly, both the laser-pointer interface and the pie menu navigation could be used effectively even by inexperienced users not

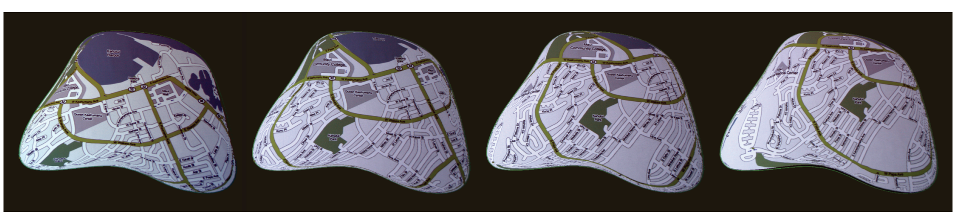

Figure 14: Series of changes of the focus parameters influencing the display warping.

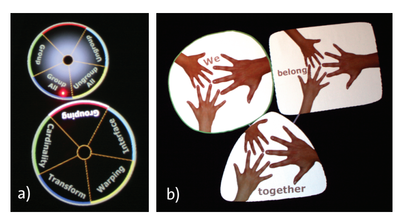

Figure 15: a) Menu. b) Grouping.

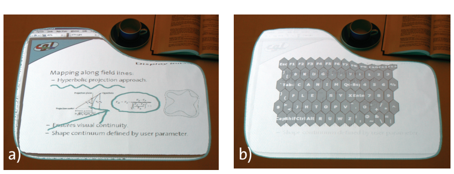

Figure 16: Snapshots of two application interface modes: a) Applying annotations. b) Textual input using keyboard tracing.

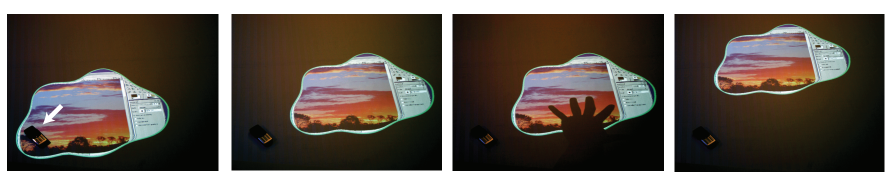

Figure 17: Environment-aware display adaptivity. The bubble reacts to the colliding cell phone and subsequently avoids the shadow cast by the hand by dynamically relocating to an adequate area determined by the scanning of the projection surface.

familiar at all with tabletop settings. For input of longer text passages, an additional finger-based interaction was desired, a feature we are currently adding to the system. For such hand-based interaction scenarios, we will disable the environment-aware adaptivity on demand.

8 CONCLUSIONS

In this paper, we have presented a novel display metaphor, which extends traditional tabletop projections by introducing novel forms of environment-aware display representations and a matching set of interaction schemes. Using a warping approach, arbitrary display content is mapped into multiple freeform, bubble-based representations, which can be elastically deformed, moved, cloned, deleted and grouped. The presented interaction schemes allow users to efficiently work with freeform displays and can easily be extended with new operations in the future. All operations are performed in an interactive and intuitive way by relying on focus and context techniques driven by laser pointer tracking. By scanning the projection geometry in an imperceptible way, we can optionally provide a smooth adaptivity of the information bubbles to changes in the environment.

We strongly believe that our novel framework for realizing interactive display bubbles is an appealing solution to lift the shape, placement, collaboration and interaction restrictions imposed by current computer and home entertainment systems. We also hope that our metaphor will trigger inspiring future work by inciting researchers and designers to rethink the rigid rectangular layout of today's displays and by encouraging them to pursue novel designs and applications with projected displays.

Limitations. In our current implementation, the system is restricted to a single interaction operation at a time, a limitation we want to remove in the near future by including simultaneous laser pointer tracking capabilities [14, 36]. Furthermore, despite hardware accelerated texture filtering, the rendering of fineprint text on small bubbles still leaves room for improvement in our prototype system. This problem can easily be alleviated by including additional I/O units into the scalable setup in order to increase the display resolution and extend the working space.

Future Work. In addition to including the aforementioned extensions, we would like to explore additional bubble operations such as display deformation, fusion and splitting upon collision. Direct hand and finger gestures could complement the current laser pointer tracking and allow for bi-manual interaction using dominant and non-dominant hands. Furthermore, we plan to investigate more elaborate automatic display initialization, adaptivity and rearrangement schemes. As an interesting challenge, it remains to be determined, which components the user is inclined to delegate to an automatic control mechanism and which ones she would like to control manually and explicitly.

ACKNOWLEDGMENTS

We thank R. Brunner, N. Cotting, D.-Y. Kwon, S. Lang, N. Lefebvre, C. Sigg, C. Voegeli and D. Wehbrink for using our system and for experimenting with our proposed display metaphor. Their feedback and ideas has significantly contributed to improving the quality of our tabletop system. We also thank our anonymous reviewers for their valuable comments and suggestions for improvement.

References

1. E. Arias, H. Eden, G. Fischer, A. Gorman, and E. Scharff. Transcending the individual human mind - creating shared understanding through collaborative design. ACM Trans. Comput.-Hum. Interact., 7(1):84–113, 2000.
2. M. Ashdown and P. Robinson. Experience with the Escritoire: A personal projected display. IEEE Multimedia, 12(1):34–42, January-March 2005.
3. D. Baraff and A. Witkin. Physically-based Modeling, Principles and Practice. In SIGGRAPH'97 Course Notes, Course 34, 1997.
4. J. Barreto and K. Daniilidis. Wide area multiple camera calibration and estimation of radial distortion. In Proc. of OMNIVIS'04, 2004.
5. T. Beier and S. Neely. Feature-based image metamorphosis. In Proc. of ACM SIGGRAPH'92, pages 35–42, 1992.
6. S. Borkowski, O. Riff, and J. Crowley. Projecting rectified images in an augmented environment. In Proc. of PROCAMS'03, 2003.
7. A. Butz, M. Schneider, and M. Spassova. SearchLight - A Lightweight Search Function for Pervasive Environments. In Proc. of PERVASIVE'04, pages 351–356, 2004.
8. W. Buxton. Human Input to Computer Systems: Theories, Techniques and Technology (book manuscript). http://www.billbuxton.com/inputManuscript.html, 2002. Viewed: November 1, 2005.
9. J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparison of pie vs. linear menus. In Proc. of CHI'88, pages 95–100, 1988.
10. Y. Chen. Building a Scalable High-Resolution Display Wall. PhD thesis, Princeton University, 2001.
11. D. Cotting, M. Naef, M. Gross, and H. Fuchs. Embedding Imperceptible Patterns into Projected Images for Simultaneous Acquisition and Display. In Proc. of ISMAR'04, pages 100–109, 2004.
12. D. Cotting, R. Ziegler, M. Gross, and H. Fuchs. Adaptive Instant Displays: Continuously Calibrated Projections Using Per-Pixel Light Control. In Proc. of Eurographics 2005, pages 705–714, 2005.
13. J. D. Austin Henderson and S. Card. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. ACM Trans. Graph., 5(3):211–243, 1986.
14. J. Davis and X. Chen. LumiPoint: Multi-User Laser-Based Interaction on Large Tiled Displays. In Displays 23:5, pages 205–211. Elsevier Science, 2002.
15. P. Dietz and D. Leigh. DiamondTouch: a multi-user touch technology. In Proc. of UIST'01, pages 219–226, 2001.
16. J. Ehnes, K. Hirota, and M. Hirose. Projected Augmentation - Augmented Reality using Rotatable Video Projectors. In Proc. of ISMAR'04, pages 26–35, 2004.
17. M. Fjeld, F. Voorhorst, M. Bichsel, H. Krueger, and M. Rauterberg. Navigation methods for an augmented reality system. In Proc. of CHI'00, pages 8–9, 2000.

18. C. Forlines and C. Shen. DTLens: multi-user tabletop spatial data exploration. In Proc. of UIST'05, pages 119–122, 2005.
19. G. W. Furnas. Generalized fisheye views. In Proc. of CHI'86, pages 16 –23, 1986.
20. D. Germans, H. Spoelder, T. Schaaf, and H. Bal. Realizing a Stereo Tiled Display Using Commodity Components. In Proc. of ASCI'03, 2003.
21. D. Hopkins. The Design and Implementation of Pie Menus. Dr. Dobb's Journal, 16:16–26, 1991.
22. T. Igarashi, T. Moscovich, and J. F. Hughes. As-Rigid-As-Possible Shape Manipulation. In Proc. of ACM SIGGRAPH'05, pages 1134–1141, 2005.
23. N. Kadmon and E. Shlomi. A polyfocal projection for statistical surfaces. The Cartographic Journal, 15(1):36 – 41, June 1978.
24. Y. Kakehi, M. Iida, T. Naemura, Y. Shirai, M. Matsushita, and T. Ohguro. Lumisight Table: An Interactive View-Dependent Tabletop Display. IEEE CG&A, 25(1):48–53, January/February 2005.
25. A. Keahey. The Generalized Detail-In-Context Problem. In Proc. of INFOVIS'98, pages 44–51, 1998.
26. T. A. Keahey and E. L. Robertson. Techniques for non-linear magnification transformations. In Proc. of INFOVIS'96, pages 38 –45, 1996.
27. C. Kim, J. Park, J. Yi, and M. Turk. Structured light based depth edge detection for object shape recovery. In Proc. of PROCAMS'05, 2005.
28. A. Kolliopoulos. Visualizing trees with a hyperbolic projection in one dimension. J. Comput. Small Coll., 18(4):133–138, 2003.
29. P.-O. Kristensson and S. Zhai. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In Proc. of UIST'04, pages 43–52, 2004.
30. G. Kurtenbach and W. Buxton. Issues in combining marking and direct manipulation techniques. In Proc. of UIST'91, pages 137–144, 1991.
31. J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In Proc. of CHI'95, pages 401–408, 1995.
32. Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. ACM Trans. Comput.-Hum. Interact., 1(2):126–160, 1994.
33. B. McCaul and A. Sutherland. Predictive Text Entry in Immersive Environments. In Proc. of IEEE VR'04, pages 241–242, 2004.
34. T. Milliron, R. J. Jensen, R. Barzel, and A. Finkelstein. A framework for geometric warps and deformations. ACM Trans. Graph., 21(1):20–51, 2002.
35. P. Ørbæk. A Multi-sided Tabletop Interface. Unpublished techreport, April 2004.
36. A. Pavlovych and W. Stuerzlinger. Laser pointers as interaction devices for collaborative pervasive computing. In Advances in Pervasive Computing, pages 315–320, April 2004.
37. C. Pinhanez. Using a Steerable Projector and a Camera to Transform Surfaces into Interactive Displays. In Proc. of CHI'01, pages 369–370, 2001.
38. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, 1992.
39. A. Raij, G. Gill, A. Majumder, H. Towles, and H. Fuchs. PixelFlex2: A Comprehensive, Automatic, Casually-Aligned Multi-Projector Display. In Proc. of PROCAMS'03, 2003.
40. R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. Proc. of ACM SIGGRAPH'98, pages 179–188, 1998.
41. M. Rauterberg, T. Mauch, and R. Stebler. How to improve the quality of human performance with natural user interfaces as a case study for augmented reality. In Advances in Occupational Ergonomics and Safety I, pages 150–153, 1996.
42. J. Rekimoto. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In Proc. of CHI'02, pages 113–120, 2002.
43. J. Rekimoto and N. Matsushita. Perceptual Surfaces: Towards a Human and Object Sensitive Interactive Display. In Proc. of PUI'97, 1997.
44. J. Rekimoto and M. Saitoh. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In Proc. of CHI'99, pages 378–385, 1999.
45. T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual Network Computing. IEEE Internet Computing, 2(1):33–38, 1998.
46. W. Stuerzlinger, O. Chapuis, D. Phillips, and N. Roussel. User Interface Facades: Towards Fully Adaptable User Interfaces. In Proc. of UIST'06, to appear, October 2006.
47. M. Sugimoto, K. Hosoi, and H. Hashizume. Caretta: a system for supporting face-to-face collaboration by integrating personal and shared spaces. In Proc. of CHI'04, pages 41–48, 2004.
48. R. J. Surati. Scalable Self-Calibration Display Technology for Seamless Large-Scale Displays. PhD thesis, Massachusetts Institute of Technology, 1999.
49. D. S. Tan, B. Meyers, and M. Czerwinski. WinCuts: manipulating arbitrary window regions for more effective use of screen space. In Proc. of CHI'04, pages 1525–1528, 2004.
50. C. Tappert. Cursive script recognition by elastic matching. IBM Journal of Research & Development, 26(6):756–771, 1982.
51. Y. Tokuda, S. Iwasaki, Y. Sato, Y. Nakanishi, and H. Koike. Ubiquitous display for dynamically changing environment. In Proc. of CHI'03, pages 976–977, 2003.
52. F. Vernier, N. Lesh, and C. Shen. Visualization techniques for circular tabletop interfaces. In Proc. of Advanced Visual Interfaces'02, 2002.
53. P. Vuylsteke and A. Oosterlinck. Range Image Acquisition with a Single Binary-Encoded Light Pattern. IEEE TPAMI, 12(2):148–164, Feb 1990.
54. B. Watson, J. Kim, T. McEneany, T. Moher, C. Hindo, L. Gomez, and S. Fransen. StorySpace: technology supporting reflection, expression and discourse in classroom narrative. IEEE CG&A, pages 2–4, Jan-Feb 2004.
55. P. Wellner. Interacting with paper on the DigitalDesk. Communications of the ACM, 36(7):86–97, 1993.
56. G. Wolberg and T. E. Boult. Separable image warping with spatial lookup tables. In Proc. of ACM SIGGRAPH'89, pages 369–378, 1989.