

Combining Shape-Changing Interfaces and Spatial Augmented Reality Enables Extended Object Appearance

David Lindlbauer¹, Jens Emil Grønbæk², Morten Birk², Kim Halskov³, Marc Alexa¹, Jörg Müller²

¹ TU Berlin, Berlin, Germany

² Department of Computer Science, Aarhus University, Aarhus, Denmark

³ CAVI, Aarhus University, Aarhus, Denmark

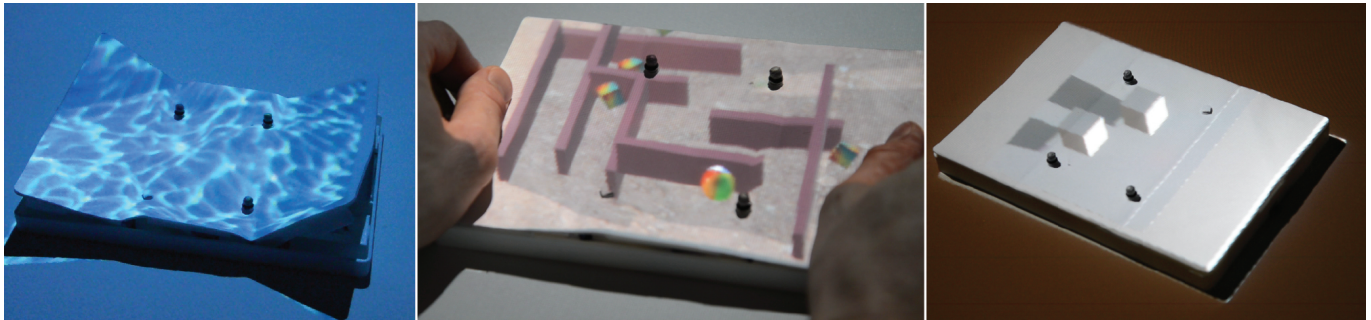


Figure 1. We combine shape-changing interfaces with spatial augmented reality. Shape-changing interfaces physically render coarse shapes and slow movements. Spatial augmented reality is used to display arbitrary textures, objects, and fast motion, directly onto the shape-changing device. This allows us to take advantage of natural depth cues and occlusion. We demonstrate the combination of the two techniques in three applications, i.e. a weather app with waves rendered physically and virtually (*left*) and a game with position of important game elements emphasized through shape change (*center*). We show capabilities of enriching shape-changing interfaces with spatial AR, e.g. the ability to render arbitrary objects and shadows (*right*).

ABSTRACT

We propose combining shape-changing interfaces and spatial augmented reality for extending the space of appearances and interactions of actuated interfaces. While shape-changing interfaces can dynamically alter the *physical appearance* of objects, the integration of spatial augmented reality additionally allows for dynamically changing objects' *optical appearance* with high detail. This way, devices can render currently challenging features such as high frequency texture or fast motion. We frame this combination in the context of computer graphics with analogies to established techniques for increasing the realism of 3D objects such as bump mapping. This extensible framework helps us identify challenges of the two techniques and benefits of their combination. We utilize our prototype shape-changing device enriched with spatial augmented reality through projection mapping to demonstrate the concept. We present a novel mechanical distance-fields algorithm for real-time fitting of mechanically constrained shape-changing devices to arbitrary 3D graphics. Furthermore, we present a technique for increasing effective screen real estate for spatial augmented reality through view-dependent shape change.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords

shape-changing interfaces; spatial augmented reality

INTRODUCTION

Shape-changing interfaces provide benefits with their ability to dynamically alter the physical shape of objects. They allow for dynamic tactile sensation (e.g. [41]), haptic data exploration (e.g. [26]) or in general diverse dynamic affordances (e.g. [23]). However, changing the physical shape of devices to resemble desired target shapes can be challenging. Especially highly detailed shapes and fast motion are hard to achieve for shape-changing interfaces, which oftentimes only feature few actuators (e.g. MorePhone with 4 actuators [18]). Typically, shape-changing devices are limited to their predefined range of possible shapes (i.e. deformations, or *states*), e.g. the faders of 2.5D displays (e.g. [27, 17]), or the predefined air-chambers of pneumatic interfaces (e.g. [41]). Additionally, even if fast physical motion is possible, it can be problematic since it could startle users (cf. [26]).

While research on shape-changing interfaces focused mostly on altering objects *physically*, a wide range of prior work aimed at changing the *optical* appearance of objects and environments, for example through spatial augmented reality (*spatial AR*, e.g. [7, 31, 21, 24]) or fish tank VR (e.g. [36]). Spatial AR is defined as augmentation of real surfaces in a physically co-located way, for example through projection or displays. Users need no additional instrumentation such as

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CHI'16, May 07–12, 2016, San Jose, CA, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3362-7/16/05...\$15.00.

DOI: <http://dx.doi.org/10.1145/2858036.2858457>

see-through glasses (cf. [8]). By enriching static physical objects with visual content, users are presented with *the illusion* of different textures, shapes, or motion. These systems benefit from the high resolution of displays and projectors, typically featuring millions of pixels, whereas creating shape-changing devices with even only several hundred individually controllable elements (e. g. the 900 actuators of Relief [17]) is challenging.

However, with spatial AR, only the optical appearance of objects is dynamic, with the physical shape remaining static. Hence, benefits of shape-changing interfaces such as dynamic tactile sensation are not available. Furthermore, the ability of spatial AR to render 3D graphics is limited by the particular display surface. While it is possible to render virtual objects smaller than the surface, it is not possible to render larger objects. Display performance also depends on the viewing position. If users look at a flat display surface from a steep angle, virtual objects cannot be rendered correctly since they would exceed the bounds of the surface and thus are cut off.

In this work, we propose combining shape-changing interfaces, typically mechanically constrained and low-resolution, with spatial AR, which allows for displaying high resolution graphics and arbitrarily fast motion. Our aim is to bridge the gap between desired (real) objects and their physical and virtual representation. We argue that by incorporating perspective-corrected virtual renderings, shape-changing interfaces can come closer to an accurate visual representation of desired arbitrary objects. We use physical shape change for *dynamic tactile sensation and haptic feedback*, and combine it with spatial AR's virtual graphics for *high-frequency texture and the illusion of fast motion*.

Until now, most research has focused on either physical *or* visual change. In the field of shape-changing interfaces, devices showed co-located 2D information (e. g. MorePhone [19]), were augmented with 2.5D height field data (e. g. Relief [27]), or extended with see-through augmented reality (e. g. Sublimate [26]). None of these approaches extends shape-changing interfaces with perspective-corrected 3D visuals in a physically co-located way. We achieved perspective-correction by tracking a user's head, while physical co-location is achieved using projection mapping. In contrast to prior work, we display 3D virtual content directly on the shape-changing device, which allows us to take advantage of natural depth cues and occlusion. Additionally, this allows for leaving users uninstrumented.

Conceptually, we see the connection between shape-changing interfaces and pixel displays analogous to the connection between (coarse) 3D models and high-detail 2D textures in computer graphics. The geometry of a 3D model is created e. g. through polygonal modeling. Subsequently, low complexity is compensated for through sophisticated textures. Therefore, various techniques such as bump mapping, shadow mapping or animated texture mappings are used to increase the realism of virtual objects. We frame our exploration of shape-changing interfaces and spatial AR in this context, which allows us to identify challenges of both techniques and opportunities of their combination.

We demonstrate the concept of combining shape-changing interfaces and spatial AR with a handheld prototype, more specifically a shape-changing tablet enriched with spatial AR through projection mapping. *However, we envision future shape-changing devices to be fully covered with high-resolution displays*. More specifically, we believe that each surface of a device will incorporate controllable pixels, especially with the rise of flexible and printed displays (e. g. [18, 29]). We use projection for *simulating* such future interfaces. We present three applications demonstrating different interaction modalities: a spatial navigation application, an ambient display, and a labyrinth-style game.

In order to map the deformation of a physically constrained shape-changing device to arbitrary 3D graphics of spatial AR, we present a real-time algorithm for matching the two based on what we call *mechanical distance fields (MDF)*. Our MDF encodes mechanical deformation in a single regular-spaced voxel grid. By matching desired virtual target shapes with MDF, we can determine the necessary deformation to represent any virtual object. This allows us to match the deformation of shape-changing interfaces in real-time to virtual 3D content, even if the device features deformation along 3 dimensions.

Furthermore, we present a technique for enhancing the quality of spatial AR by incorporating a user's perspective, which we call *view-dependent shape-change*. By dynamically altering the display surface, challenges of spatial AR introduced by limited display space can be overcome. This way, we can render virtual 3D objects which would usually be cropped due to the lack of display surface.

Contributions

- An exploration of combining physically changing devices and spatial AR for rendering arbitrary 3D objects and visual effects, framed in the context of computer graphics.
- A technique for extending the design space of shape-changing devices with rendering of complex textures, fast motion, and non-topologically equivalent shape changes.
- A real-time algorithm for matching constrained shape-changing devices to arbitrary 3D models.
- A technique for extending the effective display area for spatial AR based on view-dependent shape change.

RELATED WORK

In the following, we discuss relevant related work from the fields of shape-changing interfaces, spatial AR, and other work which aims at changing the optical appearance of objects.

Shape-changing interfaces

Shape-changing interfaces can change their physical appearance through actuation and serve as input or output devices, or both [23]. They can fulfill a large variety of aims, such as functional (e. g. communication [22], dynamic affordances [17], tactile feedback [41]), hedonic (e. g. aesthetics [41]), conveying emotions (e. g. Thrifty Faucet [38]), or exploration (cf. [23, 33]). To achieve these aims, different strategies are applied, such as different types of shape change or user interaction. 2.5D displays, for example, have been studied as input and

output devices for co-located collaboration, telepresence and interaction (e. g. [26, 17]). Their actuated pins allow for user interaction (input) as well as dynamic physical shape and even for actuation of objects placed on top (e. g. inForm [17]). With PneUI [41], the topology of objects is altered with pneumatics for functional aims (e. g. game controller) or hedonic aims (e. g. shape-changing lamp). Alexander et al. demonstrated interactions with a grid of small actuated (tilt) displays [3], whereas we focus on using perspective-corrected 3D graphics.

Rasmussen et al. [33] and Coelho et al. [14] provide taxonomies of shape-changing interfaces and discuss different types of shape change, interaction, and aims. Roudaut et al. [34] quantify the resolution of shape-changing interfaces based on non-rational uniform basis splines (NURBS). As discussed by Rasmussen et al. [33] and Roudaut et al. [34], most research focused on topologically equivalent changes of shape, such as change in volume (e. g. [22]), or texture (e. g. [27, 41]). Other, non-topologically equivalent changes (e. g. dynamically adding holes to a surface) are less explored, arguably because they are hard to achieve with current technology (cf. [34]). Coelho et al. [13] changed the permeability (i. e. create holes in the surface) of a surface for architectural purposes.

Prior work focused mostly on the combination of shape-changing interfaces with 2D graphics, such as caller information or maps on shape-changing mobile phone (e. g. MorePhone[18]). Leithinger et al. enriched their 2.5D displays with virtual graphics through height field data ([27]) and see-through augmented reality ([26]). We extend this line of research by displaying perspective-corrected 3D graphics directly on shape-changing interfaces.

Spatial augmented reality

Spatial augmented reality typically refers to situations when objects are augmented by displaying virtual content onto them (cf. [8]). The virtual content is thereby *registered* to the real objects, which allows for 3D perception while exploiting natural depth cues. No other instrumentation such as see-through displays is needed, since head-tracking is used. We use an interpretation of the term spatial AR, where also displays mounted on a surface can serve as 3D virtual extension of the object (referred to as fish tank VR, e. g. pCube [36]). With Shader Lamps, Raskar et al. [31] used projection to virtually enhance 3D-scanned objects to alter their appearance dynamically. Techniques for projection mapping (e. g. calibration, compensation for background [20]) are also applied for spatial augmented reality (cf. [8]). These techniques are used for example to support communication (e. g. [15]), urban planning, and media architecture design (e. g. [4]). Furthermore, spatial AR is used for novel interactions (e. g. MirageTable [6]) or increased immersion (e. g. RoomAlive [24]). With Illuminating Clay, Piper et al. [30] augmented a *manually* deformable surface with volume data through projection mapping.

Tangible 3D Tabletop [16, 21] combines tangible tabletop interaction with 3D projection to augment tangible objects with visual content corresponding to its physical shape, position, and orientation on the tabletop. This means that the flat tabletop surface together with the tangible objects becomes a deformable interface.

Among the applications of projection mapping in art we find the works of Valbuena. His installations, for instance Augmented Sculptures [39], are typically composed of very angular and clear-cut geometrical shapes, which, in conjunction with projection mapping, are used to create optical illusions. AntiVJ's installation, Enghien [4], employed projection mapping to create the illusions of the transformation of a building façade. The non-interactive installation first mimics moving light sources to emphasize the 3D effect; then, starts copying and apparently moving the physical, architectural features, such as windows and balconies; finally, it deconstructs these features of the building (cf. [15]).

Bermano et al. [7] augmented an animatronic head with virtual textures. They exploit the dense correspondence between the physical avatar and the virtual shape to increase realism. In our approach, we display *arbitrary* 3D objects and visuals onto shape-changing interfaces.

Appearance control

Besides using projection, researchers changed the optical properties of objects through a variety of other techniques. Alexa and Matusik [2] fabricated objects with different microstructures that yield appearance changes under different viewing angles. Furthermore, a large body of work focused on controlled surface reflectance (e. g. [28]), which allows for objects with dynamic surface properties. All of these works leave the physical shape of objects largely unchanged.

COMBINING SHAPE-CHANGING INTERFACES WITH SPATIAL AUGMENTED REALITY

In this section, we first identify challenges when designing and building shape-changing interfaces taking related work into account. We then address these challenges and frame solutions with analogies to classic techniques of computer graphics.

Challenges of shape-changing interfaces

Looking at work by Roudaut et al. [34] on shape resolution and the taxonomy of shape-changing interfaces by Rasmussen et al. [33], we can identify the main challenges we address in our work:

- *Limited granularity*: The limited number of actuators on many shape-changing interfaces makes rendering more detailed shapes or high frequency textures, such as bumpy surfaces or waves, challenging.
- *Limited speed*: Due to mechanical limitations, actuation speed is limited and high frequency motion is difficult to achieve. Additionally, even if faster motion was possible, it would often not be desirable due to user interaction (e. g. startle users, risk of jamming fingers).
- *Optical appearance relies on physical shape*: Shape-changing interfaces are bound to their physical appearance and environmental influence. Shadows thrown on the device depend largely on (uncontrolled) ambient light. Physical features such as edges of a device depend solely on the current deformation. Current shape-changing interfaces cannot change the ways users perceive them beyond physically possible changes (e. g. perceived levitation or holes).

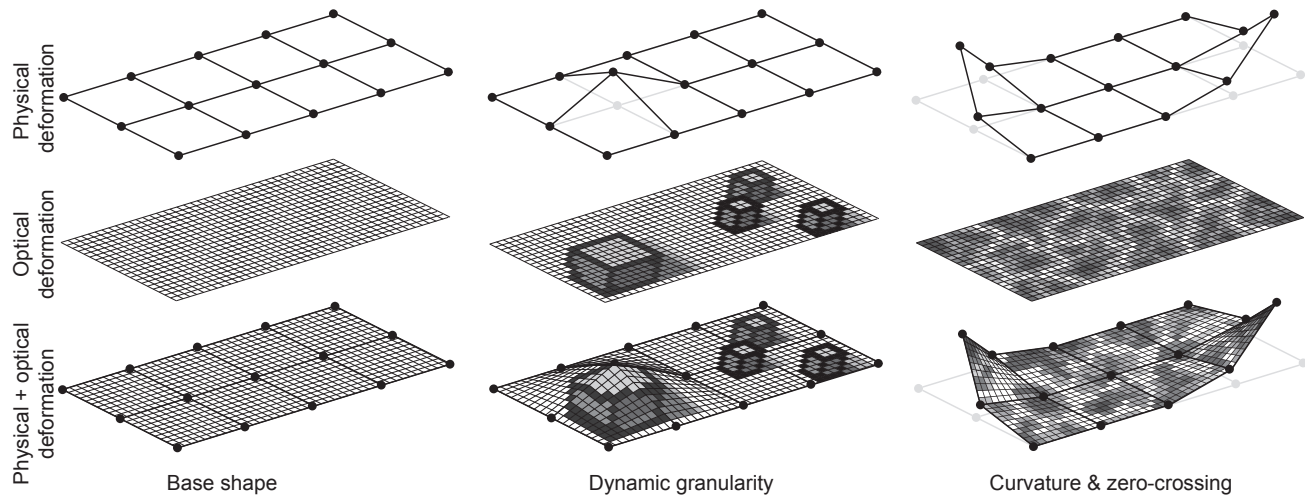


Figure 2. The physical deformation (*top row*) is combined with the optical “deformation” (*center row*), i.e., a shape-changing device with a pixel display (*bottom row*). This allows for higher and changing granularity (*center column*), as well as to render varying curvature and a high number of zero-crossing (*right*). For illustration purposes the display is rather low-resolution (15×30 pixels in this case). We envision the optical component to be a high resolution deformable display (e.g. >300 ppi). Figure adapted from [34].

These limitations reduce the expressiveness of shape-changing interfaces and thus their ability to achieve various goals (cf. [33]). Devices with hedonic aims such as conveying emotion (e.g. Thrifty faucet [38]) could benefit for example by not only triggering emotion through bending but also through a highly dynamic surface texture that moves and shows small spikes. Devices with functional aims such as communicating information (e.g. shape-changing mobiles phones [18, 34]) face challenges when trying to express information beyond the capability of their few actuators, e.g. indicating information about upcoming weather (e.g. strength of wind, cloudiness).

Although spatial AR does not allow for tangible change, it can give users the *illusion* of features such as high detail texture or high frequency motion. We argue that the combination with the tangibility of shape-changing interfaces opens more possibilities for researchers and designers. Additionally, we believe spatial AR benefits from the combination with shape-changing interfaces by dynamically altering the display surface.

Connection to computer graphics: 3D models and 2D textures

We frame our proposed techniques in the context of *computer graphics*, more specifically in the context of polygonal 3D models and 2D texturing. We identified three different techniques which we found to be applicable when using spatial AR to address challenges of shape-changing interfaces: *bump maps*, *animated texture maps*, and *shadow maps*. Furthermore, we show how other optical properties of shape-changing devices such as transparency or area (cf. [33]) can be altered, based on the principle of *environment maps*. We applied those principles to our current prototype.

Bump maps: Increasing perceived resolution

Bump mapping is a technique for the depiction of high detail surfaces with only low resolution 3D models, introduced by Blinn in 1978 [9]. The surface normals are changed, e.g. through perturbation, to render effects such as wrinkles or uneven surfaces, highly contributing to the perceived realism of surfaces. The geometry of the object is not modified. This

allows for high realism, since the perceived intensity of light is mostly determined by the direction of the surface normal, and less by the position on the surface (cf. [9]).

Like for coarse 3D models, the granularity (i.e. the number of actuators with respect to the device area) of shape-changing devices is rather low (cf. [34]). Increasing the granularity can be challenging, since it means adding actuators and their control mechanisms on a limited area (e.g. within a tablet-sized device), and increased power consumption.

In contrast, traditional displays or video projection feature several million controllable pixels. By adding virtual graphics to shape-changing interfaces, their perceived resolution can be increased. Analogous to using bump maps on 3D geometry, shape-changing interfaces provide the coarse, low resolution geometry. 3D graphics are responsible for *simulating* high resolution spatial and temporal variations (cf. high-detail 2D textures). Like 2.5D shape displays, regular displays can change their perceived granularity through simultaneously changing (i.e. grouping) actuators, ranging from changing only 1 pixel to all pixels simultaneously.

We use shape-changing devices to render the coarse (tactile) shape of a desired target, e.g. a wave-like form mimicking a water surface, or an architectural model. We then use spatial AR for more detailed surface features such as bumps on a map for small hills and valleys, ground textures, or objects for games. Those features are displayed perspective-corrected to appear realistic to the observer. Figure 2 (*center*) illustrates changing a device with multiple control points physically, and multiple boxes added by 3D graphics, as well as the combination of the two.

Granularity also influences other features of shape-resolution, such as curvature, zero-crossings (i.e. ability to produce wave-like shapes), and closure (cf. [34]). As noted by Roudaut et al., curve deformation is challenging to achieve due to different material properties. Shape-changing devices need to be manufactured from, or covered with, flexible material

(e. g. foam or Lycra [12, 27]). Dependent on the amplitude and curvature, only a relatively low number of zero-crossings can be achieved. This problem is even more apparent for devices with only very few actuators, e. g. shape-changing mobile phones. In combination with spatial AR, the number of available perceived zero-crossings, i. e., the frequency of wave-like shapes, can be increased drastically. As illustrated in Figure 2 (*right*), low frequency waves are rendered physically, whereas smaller wave-like structures are simulated.

Animated texture maps: Increasing perceived speed

Animated texture maps are used for presenting predefined animations onto other virtual surfaces, without changing the actual surface geometry. The animations can be of arbitrary speed and size.

Roudaut et al. discuss speed and animation, however mostly in terms of a device's ability to perform actions with varying speed. While the speed of most shape-changing interfaces is fixed (i. e. determined by mechanics, cf. [34]), we argue that in combination with spatial AR, perceived speed of shape-changing devices can be varied as well as increased beyond the physically possible motion of the device. Like with *animated texture maps*, which are used to render motion on static or low resolution surfaces, we render fast virtual motion on the rather slow moving shape-changing interface. Additionally, we render motion of more fine-grained textures (e. g. small waves) and objects (e. g. a rolling marble).

The maximum velocity of a shape-changing interface depends on its implementation. Devices such as 2.5D shape displays with motorized pins rely on the actuation speed of the pins, whereas pneumatic interfaces rely on the pumping and vacuuming speed. Dielectric electro active polymers (DEAP) and shape-memory alloys (SMAs) usually rely on rather high voltage or current, respectively, and oftentimes suffer from low (de-)activation speeds. Higher velocity motion would be possible but poses challenges in terms of manufacturing as well as user interaction. When the velocity is increased above a certain threshold, it is challenging to accurately match a desired target deformation due to the mass of the object and resulting oscillation around the target position. Additionally, high speed deformation poses potential risks to users (e. g. jamming fingers or startling users).

By combining shape-changing interfaces and spatial AR, high velocity movements can be performed virtually. Physical deformation can be performed simultaneously, but with lower velocity, overcoming these potential limitations. In our prototypes, physical actuation is included in our shape-matching algorithm with predefined maximum velocity and a damping factor, allowing for smooth motion which does not distract or startle users.

Shadow maps: Adding virtual depth

Shadow mapping, introduced by Williams in 1978 [40], is a technique for efficiently rendering shadows. A shadow map is generated by rendering the scene from the light source's perspective and storing depth information (z-buffer values). Through comparison of the scene part displayed in each pixel, the position of shadows is determined. This allows for greatly

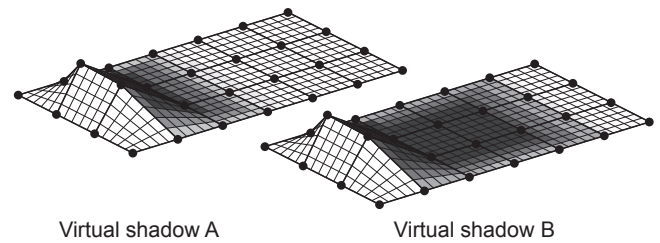


Figure 3. Two different virtual shadows for the same physical shape.

increasing the realism of virtual scenes. Additionally, by moving the virtual light source, different effects such as change in time (e. g. virtual sun clock) and perceived object position (e. g. levitation) can be achieved.

Current shape-changing interfaces are bound to their physical appearance when rendering objects. Occlusion or shadows depend on the current shape of the device and the illumination coming from the environment. Using spatial AR allows for simulating these properties. We change the position of a virtual light source in the "scene" (i. e. the environment where the tracked shape-changing device is in), effectively adding virtual shadows to a physically deformed device (see Figure 3). This allows us to illuminate the device in a way that gives users the impression of different shadows on the device, independently of ambient illumination from the environment. Previously, this was not possible since shape-changing interfaces relied on actual shadows thrown on or through the device. Shadow maps allow designers of shape-changing interfaces to convey different information, e. g. progress of long running processes, or simply time passing, indicated by shadows moving according to the sun (i. e. our virtual light source).

Additionally, *shadow maps* can be used to change optical features of shape-changing interfaces. As illustrated in Figure 4, by emphasizing or blurring physical edges, their appearance can be altered. By emphasizing soft edges with sharp lines, the edges appear more pronounced, and vice versa.

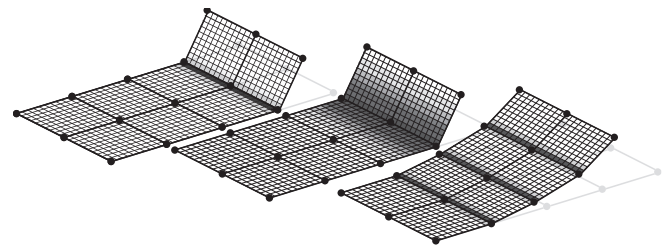


Figure 4. By adding emphasized or blurred virtual edges, perceived shapes such as the sharpness of edges can be altered.

Finally, spatial AR allows for rendering arbitrary objects on shape-changing devices. This makes it possible to render a background texture on the device as well as other objects including their virtual shadow. By exploiting this property, it is possible to create virtual objects on the device that e. g. float above the ground (see Figure 5 and 1), or disappear and reappear on demand. This shadow effect has been used in prior work to make fairies fly in front of a statue as part of culture heritage communication at a historic castle [15].

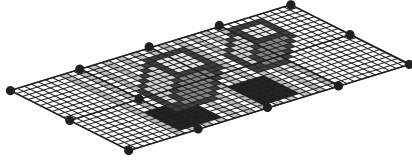


Figure 5. Two virtual cubes are rendered on the shape-changing device. By adding virtual shadows, the illusion of objects hovering over the device can be created.

Extension: Environment maps for optical effects

Besides increasing perceived resolution and speed of shape-changing devices, other optical effects such as transparency can be included in future shape-changing interfaces. By showing the background on top of the device, features such as holes (i. e. partial transparency) can be rendered. This also allows for decreasing the perceived size of a shape-changing device, which is challenging with current technology.

This approach, however, requires knowledge of the environment, such as the surface underneath, beside, or behind the shape-changing device, thus deviates from the other proposed concepts. Therefore, we consider changing these optical properties of shape-changing interfaces as *extensions* of our proposed framework.

In computer graphics, this technique is referred to as environment maps (or reflection maps, [10]). The texture of the environment is captured and stored for later mapping it on a virtual device. This allows for efficient rendering of reflection without having to render the geometry of the environment. Our current implementation uses projection mapping, which allows us to control the appearance of the environment by using it as projection surface (see Figure 6).



Figure 6. By rendering contents on the table, the device blends into the environment and can render apparent holes.

MATCHING PHYSICAL AND VIRTUAL SHAPES

One of our main objectives is to find appropriate physical and optical properties of shape-changing devices to match desired virtual objects. For achieving this goal it is therefore central to match the physical shape of the device to the shape(s) in the virtual environment.

We note that this step was comparatively simple in prior work, as it focused either on 2D information displayed on a shape-changing devices or on height field data displayed on 2.5D shape displays. In our work, however, shapes may cover an arbitrary volume in 3D and they need to be represented by shape change that goes beyond a simple change of height.

Consequently, our goal is to design an algorithm that computes the actuation parameters of the shape changing device so that the virtual model is fit as closely as possible, and in real-time. In the following we develop a model for the problem that makes the idea of a *close-to-optimal fit* precise and facilitates fast computation.

Overview

Our algorithm takes a model of an actuated device (e. g. as a set of 3D models, one per deformation state) as input. The input, representing individual actuators and states, is then encoded into multiple 3D voxel grids. Each voxel holds information on how the device needs to be deformed to cover it. The individual actuators of a device are encoded separately and later combined into a single voxel grid. This representation allows us to store actuation efficiently, and enables real-time matching.

For matching a the shape-changing device to a arbitrary target shape (e. g. given as 3D model), we first decode the target into a voxel grid as well. We then overlay the initially computed voxel grid with the target voxel grid and take the overlapping voxels into account for fitting. Subsequently, we calculate the optimal fit between the initial and the target voxel grid (see below). Figure 7 illustrates the complete algorithm.

Representation

The most important part of the algorithm lies in the representation of the shape changing device. Typically, a shape-changing device is composed of multiple actuators (i. e. degrees of freedom), which, when activated, deform a specific part of the device. The resulting motion need not be linear, which is typical for devices such as shape-changing mobile phones (e. g. [34, 18]). We assume the actuation level of each actuator to be discretized. Then we represent the *state* of the shape-changing device as $\mathbf{s} \in \mathbb{Z}_m^n$, where n is the number of actuators and m is the number of actuation levels per actuator. Note that there is a total of n^m distinct states for the device.

We represent the shape of the device as a discrete grid of voxels above the zero-state $\mathbf{s} = (0, \dots, 0)$ of the device (i. e. all actuators in state 0). Each voxel is either *covered* or *uncovered* by the device, where covered means the surface has been pushed far enough such that the voxel is below the surface. The status of each grid point is represented by 0 for uncovered and 1 for covered. The shape is a binary vector depending on the state, i. e. $V(\mathbf{s}) \in \{0, 1\}^{x \times y \times z}$, where x, y, z are the dimensions of the voxel grid in space. We note that the discretization need not be rectilinear, however, this is clearly the most convenient choice. We require that the voxel representation can be computed for any state \mathbf{s} , i. e. that the shape-changing device is known a priori.

The virtual target shape to be represented by the device is likewise represented in the volumetric grid and denoted $V_T \in \{0, 1\}^{x \times y \times z}$. In a typical scenario the target shape is given as manifold, intersection-free mesh without boundary (sometimes referred to as 'watertight'), which can be scan-converted into the voxel representation. In principle, however, our approach works with every shape representation that can be converted into a voxel grid.

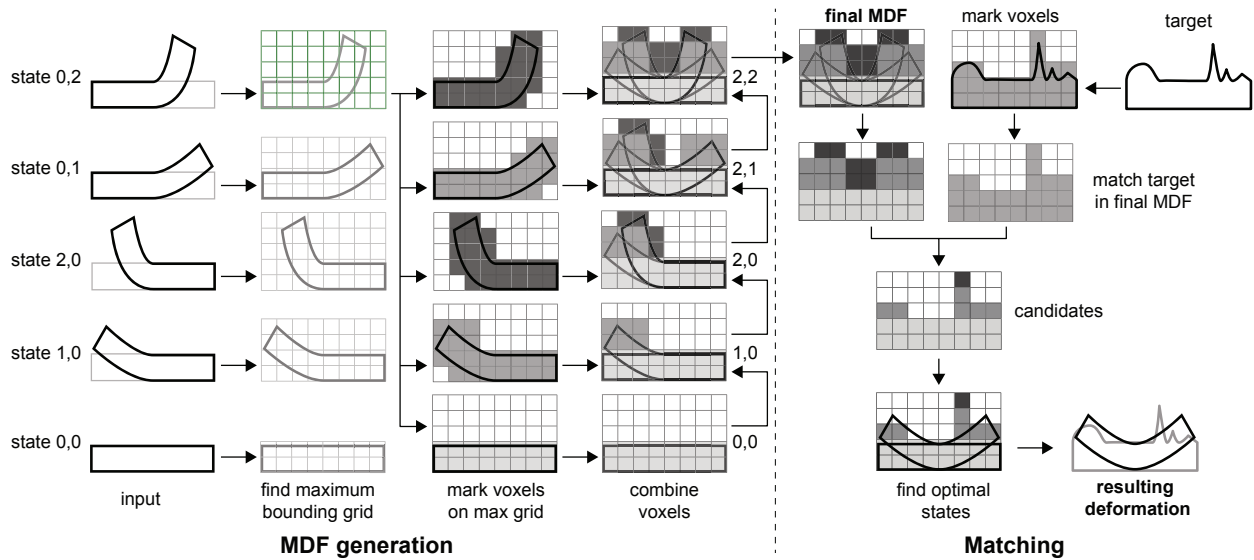


Figure 7. Computation of the initial mechanical distance field (left) and matching to a target shape (right) in a 2D example. Each state of the device (left) is encoded individually, and combined to a single MDF (center). The target object is also encoded into a voxel grid and then matched with the initial voxel grid. This gives us candidate voxels (encoding device actuation). We then find the close-to-optimal fit in the MDF, which gives us the resulting deformation for any given target (bottom right).

Reduction and simplification

In general it is not feasible to store the grid for each of the n^m possible states, as this would require $xyz \cdot n^m$ bits storage, which is cumbersome for realistic dimensions. In particular, real-time computation would be difficult on this data volume.

Our solution is to *decouple* the individual actuators. This means we will consider the effect of each actuator on the shape changing device *independent* of the state of the remaining actuators. This is justified as only few voxels are actually influenced by more than one actuator. Moreover, we do model the interaction of actuators, albeit probabilistically: we consider the expected value for a voxel depending on the state of an actuator, considering the likelihood of the actuation levels of the other actuators being equal. For example, if a voxel is covered at state j of actuator i for half of the possible states of the other actuators, we consider the value of the voxel to be 0.5. This means we can now store the voxel grid for each actuator i and each actuation level j independently in a real valued vector $\tilde{V}_{i,j} \in [0, 1]^{x \times y \times z}$. Note that now the complexity has reduced from using n^m to using nm grids.

We further simplify this representation, making a *monotonicity* assumption: as the actuation levels of the actuators are increased, the enclosed volume never shrinks. This means, if a voxel is covered for actuator i at level j , it will also be covered for any level $j' > j$. We notice that most voxels are affected only by a single actuator. Such voxels will at some actuation level change their value from 0 to 1. Rather than storing all 0's and 1's it suffices to store the actuation level for the change. Furthermore, voxels that depend on more than one actuator usually behave linearly, and we decided to store the values of any voxel for actuator i as a function that is equal 0, then linearly increases to 1, and then stays constant. This means we can store values of each voxel for a given actuator as two actuation levels: the first actuation level at which the value is non-zero, and the first level for which the value is 1. For

voxels that are dependent on a single actuator, the two values will be identical. We store this representation as two vectors $V_i^0, V_i^1 \in \mathbb{Z}_m^{x \times y \times z}$. When needed, this information can be used to quickly generate the values $\tilde{V}_{i,j}$.

The representation of the shape for a single actuator has a nice intuitive visualization: the actuation levels V_i^0, V_i^1 symbolize the 'mechanical work' that has to be exerted by the actuator to 'reach' the voxel. Based on this idea, and in analogy to distance fields (cf. [25, 11, 35]) we nickname the representation *mechanical distance field*. The generation of the MDF is visualized in Figure 7 and the result for the set of all actuators combined in our prototype is shown in Figure 8.

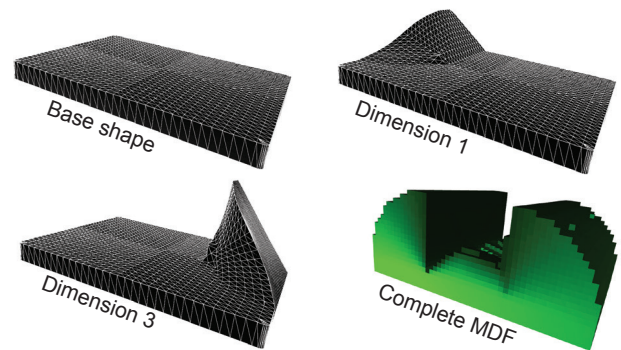


Figure 8. Example of a fully computed MDF (bottom right) for a device (base state 0 top left). Two of six dimensions of deformation (2 actuated states for dimensions 1 and 3) are illustrated. Darker green indicates larger state.

Real-time matching

Based on the precomputed grids $\tilde{V}_{i,j}$ we use a voting approach to decide on the best state s_T : for each actuator i we check all actuation levels j and try to maximize the fit between the volume represented by the shape changing device and the target shape.

We assess the error of fit for level j for actuator i as the symmetric difference between $|\tilde{V}_{i,j} - V_T|$ (where the absolute value $|\cdot|$ is performed element-wise). Note that this evaluates to zero whenever the values are identical; and to one whenever either the voxel should be covered but is not, or should be uncovered but is. Naturally, we want to choose j such that this value is small.

On the other hand, we cannot expect to cover the shape with a single actuator so the error will usually be large. Among states with large error we would prefer those that represent most of the volume of the target shape, i. e. we would like to maximize the intersection of the target shape and the volume covered by the device. This volume is given by $\tilde{V}_{i,j} \cdot V_T$.

Based on these measures we defined the optimal state for actuator i as

$$s_i = \operatorname{argmin}_j \frac{|\tilde{V}_{i,j} - V_T|}{\tilde{V}_{i,j} \cdot V_T}. \quad (1)$$

For numerical reasons we only consider values j for which the common interior volume $\tilde{V}_{i,j} \cdot V_T$ is bounded away from zero. Together these values form the optimal state vector \mathbf{s}_T .

Computational considerations

We first determine size of the voxel grid by computing the bounding box of the shape for all states. The smallest side length of the bounding box divided by the resolution r defines the side length of the voxels. In the case of our tablet prototype, as an example, we sample the deformation of each actuator $r = 33$ times, resulting in an initial voxel grid of $x = 83 \times y = 33 \times z = 59$ (~160K total) voxels, which gives us a good balance between resolution of the mechanical deformation and computational effort.

The computation of all best fits can be performed in a single iteration over all voxels, collecting the products and differences for each actuator, and calculating the best fit. For the above mentioned voxel resolution our algorithm runs in real-time, with the most expensive part being the intersection and difference computations. The complexity of the algorithm grows linearly with the number of voxels, actuators, and actuation levels.

VIEW-DEPENDENT SHAPE CHANGE

Typically, the ability of spatial AR to present 3D contents to users is limited by the display surface. If a virtual object extends beyond the bounds of the projection surface, it is cropped. This is true for projection mapping as well as for other techniques such as stereoscopic displays. We make use of the shape-changing devices' ability to extend its surface to avoid cropping without the need to wear glasses. The surface of the device is deformed depending on the angle between the user and the device. In our prototype, this means that an edge is bent upwards to increase the effective projection surface area, as shown in Figure 9.

Figure 10 illustrates the computation of the deformation. The angle α between the user and the device is calculated as the dot product between the viewing direction \vec{u} and the normal of the device \vec{n} . We then calculate the deformation simply as $\beta = \theta - \alpha \times \frac{\alpha}{\theta}$, where θ denotes the maximum deformation of

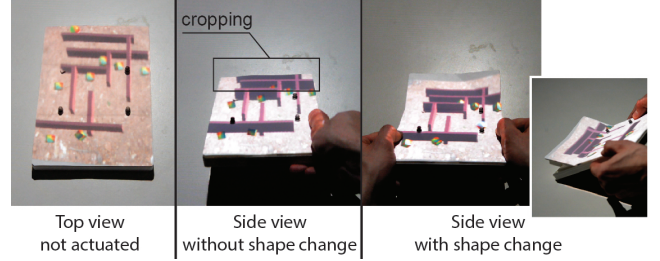


Figure 9. The virtual world displayed on the shape-changing device (left). Without shape-change, the display area of the device is not sufficient for users to see the virtual world fully (i. e. walls are cropped, center). With view-dependent shape-change, the virtual world can be seen fully (right, deformation state *inlay*).

the device (in our case, 90°). We deform the actuators which are furthest away from the user.

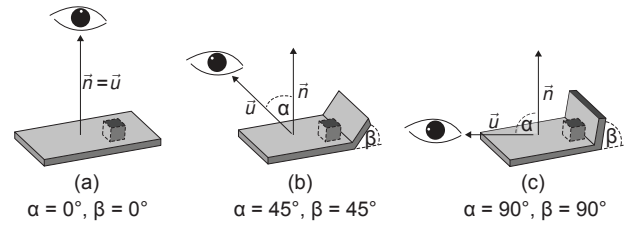


Figure 10. The shape-changing device is deformed to enable the best viewing angle on virtual objects displayed on the device.

IMPLEMENTATION

We built a prototypical shape-changing device enriched with spatial AR through projection mapping to illustrate the concept of combining the two. Software and hardware is released as open-source at <http://www.cg.tu-berlin.de/research/projects/sci-and-ar/>.

Shape-changing tablet

We built a shape-changing tablet ($21 \text{ cm} \times 15 \text{ cm} \times 2 \text{ cm}$) from a custom 3D printed housing with 6 servo motors (Flug51 SG90) as actuators. The design of the device was inspired by Rasmussen et al. [32], who were using a smaller, phone-sized and wired version with 4 actuators for their user study. The device is powered by two 2000mAh LiPo batteries. The servo motors are controlled through an Arduino Pro Mini, connected wirelessly via Bluetooth (HC-06 module) to a computer running our custom software. The top of the device is 3D printed from flexible filament (NinjaFlex), which allows for bending according to the servo position. For increased projection quality, the top is additionally covered with a sheet of paper (see Figure 11).

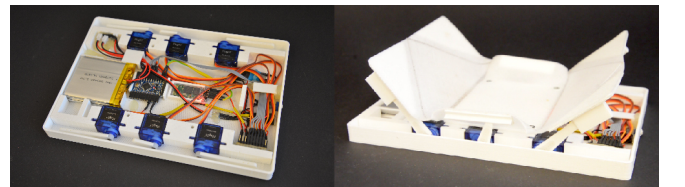


Figure 11. Shape-changing tablet containing 6 servo motors. Each corner can be bent, as well as the center of the tablet.

Projection mapping

3D projection mapping is based on having a 3D model of the physical object(s). This allows us to incorporate the device's geometry into our projection mapping system. We use a rigged 3D model of the tablet within our projection mapping application. By positioning and calibrating the projection system so that the relationship of the projection to the physical object corresponds to the virtual camera's relation to the 3D model, we can project the digital model onto the physical elements of the installation, thereby augmenting the physical object. We track the device as well as the user's head position using an OptiTrack IR tracking system. Projection is done using a single projector positioned ~2 m above the surface.

Software

The control software for our sample applications is written in Unity, with rigged models of our prototypes created in Blender. We use a proprietary software for calibrating our projection mapping system, written in Java. The MDF algorithm is developed in C++ using openFrameworks and CGAL [37]. For voxelization of shapes, we use a simple box-triangle intersection algorithm as described in [1]. The C++ application handles generation and persistence of the MDF. The Unity software handles real-time shape matching.

APPLICATIONS

We demonstrate three different applications showcasing the potential benefits of combining shape-changing interfaces with co-located 3D graphics.

Spatial navigation

Users navigate in a virtual 3D environment, i. e. a 3D map. The shape-changing device deforms according to the geometry, as depicted in Figure 12. Details of buildings and places are rendered onto the device using 3D co-located graphics. Users can rotate the device to explore the map freely. This application uses our view-dependent algorithm. When the user look from an angle, the deformation adapts, making it possible to render features extending the bounds of the projection surface (e. g. by tilting the device 90°).

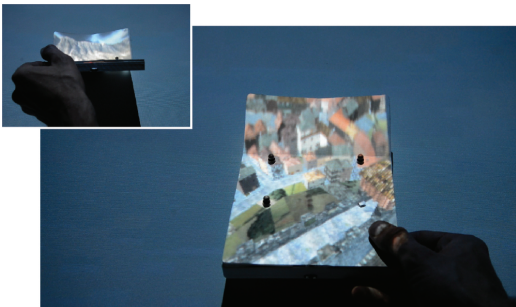


Figure 12. Users navigate on a 3D map. Tilting of the tablet is compensated for using view-dependent shape-change (*inlay*).

Ambient display

Information of current and upcoming weather and the time are displayed on and with the shape-changing tablet (Figure 13). Virtual shadows of 3D objects convey information about time of the day, following a virtually moving sun. Additionally, a wave simulation is rendered physically through deformation

and virtually through spatial AR. The spatial and temporal frequency depends on the strength of wind. The shape-changing tablet deforms according to the wave simulation (i. e. stronger waves result in faster and larger deformation). spatial AR is used for rendering more fine grained details of the simulation. Furthermore, we added virtual fog to convey information of wind strength. The fog seems to float on top of the tablet.

A problem with shape-changing interfaces is that sudden deformations are not always suitable. To solve this, we implemented the concept of *anticipating shape change*. First, only the virtual waves move on top of the device indicating an upcoming shape change. With a short delay, the physical movement starts.

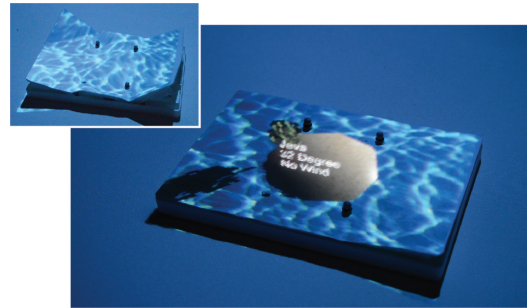


Figure 13. An ambient display conveys information about time and weather. Waves are rendered physically and virtually (*inlay*).

Labyrinth game

We implemented a simple labyrinth game (Figure 14). Users control a (virtual) rolling ball and maneuver it around to pick up colorful objects to complete a level. 3D information of the environment is rendered virtually on the device, including perspective-corrected walls, elevated game objects and high frequency ground textures. The ball also serves as input for our shape-matching algorithm and triggers deformation. Especially for regions covered by player's hands, this seeks to increase the feeling of a physical game object. Additionally, game effects such as stairs are rendered as a combination of shape-changing and virtual graphics.

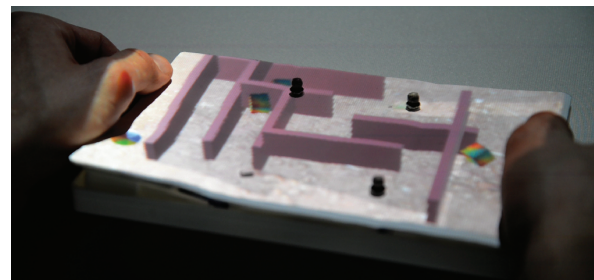


Figure 14. In the labyrinth game, the position of the ball (*bottom left*) is also rendered physically for greater immersion.

Additional application using dynamic porosity

By using knowledge of the background, we can render holes into the shape. In our current implementation, the background is created by our application, therefore no camera is needed. We implemented this in a simple physical simulation (see Figure 6). The virtual ice first has holes, then breaks apart, thus revealing the underlying surface.

FIRST USER EXPERIENCES

We conducted a small informal user study with 5 participants (24 - 36 years), all students and staff from a local university, for getting first insights into our proposed technique. The study was performed as a think-aloud session. Participants, all unexperienced with both techniques, were first introduced to the applications described earlier. Thereafter, they experimented with the applications themselves.

For the ambient display, participants reacted positively to the combination of the two techniques. Participants immediately understood that physical shape-change corresponds to the virtually rendered waves. One participant noticed: "Oh, now it is clearly getting windy!" after the shape change started on the weather simulation and virtual fog was moving faster. Another participant suggested that the device should spray water, indicating that the device becomes more organic when it is enriched with 3D content. For the labyrinth game, participants felt like the virtual game world was moving whenever the ball was at a position featuring actuation.

This makes us believe that participants considered the device to resemble the objects they saw (i. e. the device *becomes* the water or the game world) and not just serve as a container for the displayed elements. This is different to other work, where virtual content was rendered and controlled through a "container" device, such as pCubee [36]. Therefore, the combination of the two techniques increased the "realism" of the device and, consequently, immersion. Also, the correspondence between the physical and virtual deformation guided users' perception. Designers and researcher can take this into consideration when designing shape-changing interfaces.

DISCUSSION AND FUTURE WORK

The combination of shape-changing interfaces and spatial AR allows for a wide variety of applications and extends the range of perceived appearances of shape-changing interfaces. It enables shape-changing interfaces (with inherently limited spatial resolution) to feature high-resolution 3D content that can overcome physical constraints. Furthermore, shape-changing interfaces only have to render slow and coarse motion, since arbitrarily fast motion can be rendered with spatial AR. Lastly, the combination can be used for rich haptic exploration of 3D visual information, allowing to better convey information.

We also believe by increasing the resolution of shape-changing interfaces, devices can not only serve as a "container" for digital content, they can also come closer to actually representing virtual objects. This would allow them to serve as *avatars* of digital objects, currently only present as on-screen content. By enabling shape-changing interfaces to effectively transition between representations, the space of possible usages can be drastically increased. Incorporating high detail 3D graphics is a step in this direction.

Tangible qualities

Spatial AR allows for changing the optical appearance of a shape-changing device, however these changes are intangible. As soon as users touch the surface which is displaying a high frequency texture, the illusion breaks. Furthermore, only the perceived physical shape of an object can be altered. Other

properties such as stretchability, viscosity or strength remain unchanged. They rely completely on the resolution of the shape-changing device. Therefore, it is desirable to increase the range of shape-changing interfaces as well as the resolution of incorporated spatial AR. Including tactile technologies such as TeslaTouch [5] in future devices would help overcome restrictions of combining shape-changing interfaces with pixel displays.

Furthermore, the visual appearance of an object is tightly coupled to its physical shape. While our concept allows for altering an object's perceived appearance, large-scale changes (e. g. doubling its size) are challenging to achieve. Thus, there are limitations in terms of possible optical illusions.

Display quality

Typically, the field-of-view of spatial AR is limited. If users look at a surface from a very steep angle, graphics disappear or are cropped (like with bump maps). We partly overcome this problem with our view-dependent shape-change approach, however, this solves the problem only to a certain extent. Furthermore, spatial AR relies on perspective correction with respect to user perspective. However, we believe that our approach is superior to currently used see-through AR technologies because it features natural depth cues and occlusion.

Our vision is that spatial AR is achieved by OLED-style displays wrapping around shape-changing interfaces, or any other display technology featuring high-resolution flexible displays. Currently, our system only features displaying digital content through projection mapping with a single projector, which gave us the ability to develop and test our concepts. However, we argue that our approach is agnostic to the way the image is generated.

In the future, we plan to extend our prototype with flexible and autostereoscopic displays and an increased number of actuators.

CONCLUSION

The combination of shape-changing interfaces with spatial AR allows for a more accurate and realistic representation of content. The technique features natural depth cues and leaves users uninstrumented, which we believe is important. It allows for displaying high frequency textures, fast motion and other optical effects such as shadows on a low resolution shape-changing device. The shape-changing device complements 3D graphics with tactile sensation and physical deformation. We believe that future shape-changing interfaces should incorporate pixel displays to increase the space of possible interactions and usages.

ACKNOWLEDGEMENTS

We want to thank Alexandra Ion and Christian Lessig for feedback and discussion, and Andreas Fender for support with the projection mapping. This work has been supported by the ERC through grant ERC-2010-StG 259550 ("XSHAPE").

REFERENCES

1. Tomas Akenine-Möller. 2005. Fast 3D Triangle-box Overlap Testing. In *ACM SIGGRAPH 2005 Courses*

- (SIGGRAPH '05). ACM, New York, NY, USA, Article 8. DOI:<http://dx.doi.org/10.1145/1198555.1198747>
2. Marc Alexa and Wojciech Matusik. 2010. Reliefs As Images. *ACM Trans. Graph.* 29, 4 (July 2010), 60:1–60:7. DOI:<http://dx.doi.org/10.1145/1778765.1778797>
 3. Jason Alexander, Andrés Lucero, and Sriram Subramanian. 2012. Tilt Displays: Designing Display Surfaces with Multi-axis Tilting and Actuation. In *Proceedings of the International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 161–170. DOI:
<http://dx.doi.org/10.1145/2371574.2371600>
 4. AntiVJ. 2009. Enghien. (2009). Retrieved September 17, 2015 from <http://antivj.com/enghien/>.
 5. Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. 2010. TeslaTouch: Electro-vibration for Touch Surfaces. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 283–292. DOI:
<http://dx.doi.org/10.1145/1866029.1866074>
 6. Hrvoje Benko, Ricardo Jota, and Andrew Wilson. 2012. MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 199–208. DOI:<http://dx.doi.org/10.1145/2207676.2207704>
 7. Amit Bermano, Philipp Brüsweiler, Anselm Grundhöfer, Daisuke Iwai, Bernd Bickel, and Markus Gross. 2013. Augmenting Physical Avatars Using Projector-based Illumination. *ACM Trans. Graph.* 32, 6, Article 189 (Nov. 2013), 10 pages. DOI:
<http://dx.doi.org/10.1145/2508363.2508416>
 8. Oliver Bimber and Ramesh Raskar. 2005. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A K Peters.
 9. James F. Blinn. 1978. Simulation of Wrinkled Surfaces. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '78)*. ACM, New York, NY, USA, 286–292. DOI:
<http://dx.doi.org/10.1145/800248.507101>
 10. James F. Blinn and Martin E. Newell. 1976. Texture and Reflection in Computer Generated Images. *Commun. ACM* 19, 10 (Oct. 1976), 542–547. DOI:
<http://dx.doi.org/10.1145/360349.360353>
 11. Harry Blum. 1967. A Transformation for Extracting New Descriptors of Shape. *Models for the Perception of Speech and Visual Form* (1967).
 12. Marcelo Coelho, Hiroshi Ishii, and Pattie Maes. 2008. Surfex: A Programmable Surface for the Design of Tangible Interfaces. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems (CHI EA '08)*. ACM, New York, NY, USA, 3429–3434. DOI:
<http://dx.doi.org/10.1145/1358628.1358869>
 13. Marcelo Coelho and Pattie Maes. 2009. Shutters: A Permeable Surface for Environmental Control and Communication. In *Proceedings of the International Conference on Tangible and Embedded Interaction (TEI '09)*. ACM, New York, NY, USA, 13–18. DOI:
<http://dx.doi.org/10.1145/1517664.1517671>
 14. Marcelo Coelho and Jamie Zigelbaum. 2011. Shape-changing Interfaces. *Personal Ubiquitous Comput.* 15, 2 (Feb. 2011), 161–173. DOI:
<http://dx.doi.org/10.1007/s00779-010-0311-y>
 15. Peter Dalsgaard and Kim Halskov. 2011. 3D Projection on Physical Objects: Design Insights from Five Real Life Cases. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1041–1050. DOI:
<http://dx.doi.org/10.1145/1978942.1979097>
 16. Peter Dalsgaard and Kim Halskov. 2014. Tangible 3D Tabletops. *interactions* 5, 21 (2014). DOI:
<http://dx.doi.org/10.1145/2658859>
 17. Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hoge, and Hiroshi Ishii. 2013. inFORM: Dynamic Physical Affordances and Constraints Through Shape and Object Actuation. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 417–426. DOI:
<http://dx.doi.org/10.1145/2501988.2502032>
 18. Antonio Gomes, Andrea Nesbitt, and Roel Vertegaal. 2013. MorePhone: A Study of Actuated Shape Deformations for Flexible Thin-film Smartphone Notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 583–592. DOI:
<http://dx.doi.org/10.1145/2470654.2470737>
 19. Antonio Gomes and Roel Vertegaal. 2015. PaperFold: Evaluating Shape Changes for Viewport Transformations in Foldable Thin-Film Display Devices. In *Proceedings of the International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15)*. ACM, New York, NY, USA, 153–160. DOI:
<http://dx.doi.org/10.1145/2677199.2680572>
 20. Michael D. Grossberg, Harish Peri, Shree K. Nayar, and Peter N. Belhumeur. 2004. Making one object look like another: controlling appearance using a projector-camera system. In *Proceedings of the IEEE Computer Society Conference Computer Vision and Pattern Recognition*. DOI:<http://dx.doi.org/10.1109/CVPR.2004.1315067>
 21. Kim Halskov, Peter Dalsgaard, and Louise Bak Stolze. 2014. Analysing Engaging Experiences with a Tangible 3D Tabletop. In *Proceedings of the Conference on Advances in Computer Entertainment Technology (ACE '14)*. ACM, New York, NY, USA, Article 29, 10 pages. DOI:<http://dx.doi.org/10.1145/2663806.2663831>
 22. Oren Horev. 2006. 'Talking to the Hand' An exploration into shape shifting objects and morphing interfaces. (2006). Master's thesis.

23. Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. 2012. Radical Atoms: Beyond Tangible Bits, Toward Transformable Materials. *interactions* 19, 1 (Jan. 2012), 38–51. DOI: <http://dx.doi.org/10.1145/2065327.2065337>
24. Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-camera Units. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 637–644. DOI: <http://dx.doi.org/10.1145/2642918.2647383>
25. Mark W. Jones, J. Andreas Baerentzen, and Milos Sramek. 2006. 3D Distance Fields: A Survey of Techniques and Applications. *IEEE Transactions on Visualization and Computer Graphics* 12, 4 (July 2006), 581–599. DOI: <http://dx.doi.org/10.1109/TVCG.2006.56>
26. Daniel Leithinger, Sean Follmer, Alex Olwal, Samuel Luescher, Akimitsu Hogge, Jinha Lee, and Hiroshi Ishii. 2013. Sublimate: State-changing Virtual and Physical Rendering to Augment Interaction with Shape Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1441–1450. DOI: <http://dx.doi.org/10.1145/2470654.2466191>
27. Daniel Leithinger, David Lakatos, Anthony DeVincenzi, Matthew Blackshaw, and Hiroshi Ishii. 2011. Direct and Gestural Interaction with Relief: A 2.5D Shape Display. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 541–548. DOI: <http://dx.doi.org/10.1145/2047196.2047268>
28. Wojciech Matusik, Boris Ajdin, Jinwei Gu, Jason Lawrence, Hendrik P. A. Lensch, Fabio Pellacini, and Szymon Rusinkiewicz. 2009. Printing Spatially-varying Reflectance. *ACM Trans. Graph.* 28, 5, Article 128 (Dec. 2009), 9 pages. DOI: <http://dx.doi.org/10.1145/1618452.1618474>
29. Simon Olberding, Michael Wessely, and Jürgen Steimle. 2014. PrintScreen: Fabricating Highly Customizable Thin-film Touch-displays. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 281–290. DOI: <http://dx.doi.org/10.1145/2642918.2647413>
30. Ben Piper, Carlo Ratti, and Hiroshi Ishii. 2002. Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 355–362. DOI: <http://dx.doi.org/10.1145/503376.503439>
31. Ramesh Raskar, Greg Welch, Kok-Lim Low, and Deepak Bandyopadhyay. 2001. Shader Lamps: Animating Real Objects With Image-Based Illumination. In *Proceedings of the Eurographics Workshop on Rendering Techniques*. Springer-Verlag, London, UK, UK, 89–102. <http://dl.acm.org/citation.cfm?id=647653.732300>
32. Majken K. Rasmussen, Timothy Merrit, Miguel Bruns Alonso, and Marianne G. Petersen. 2016. Balancing User and System Control in Shape-Changing Interfaces: a Designerly Exploration. In *Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '16)*. ACM, New York, NY, USA.
33. Majken K. Rasmussen, Esben W. Pedersen, Marianne G. Petersen, and Kasper Hornbæk. 2012. Shape-changing Interfaces: A Review of the Design Space and Open Research Questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 735–744. DOI: <http://dx.doi.org/10.1145/2207676.2207781>
34. Anne Roudaut, Abhijit Karnik, Markus Löchtefeld, and Sriram Subramanian. 2013. Morphees: Toward High "Shape Resolution" in Self-actuated Flexible Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 593–602. DOI: <http://dx.doi.org/10.1145/2470654.2470738>
35. Jean Serra. 1983. *Image Analysis and Mathematical Morphology*. Academic Press, Inc.
36. Ian Stavness, Billy Lam, and Sidney Fels. 2010. pCubee: A Perspective-corrected Handheld Cubic Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1381–1390. DOI: <http://dx.doi.org/10.1145/1753326.1753535>
37. The CGAL Project. 2015. *CGAL User and Reference Manual* (4.6.3 ed.). CGAL Editorial Board. <http://doc.cgal.org/4.6.3/Manual/packages.html>
38. Jonas Togler, Fabian Hemmert, and Reto Wettach. 2009. Living Interfaces: The Thrifty Faucet. In *Proceedings of the International Conference on Tangible and Embedded Interaction (TEI '09)*. ACM, New York, NY, USA, 43–44. DOI: <http://dx.doi.org/10.1145/1517664.1517680>
39. Valbuena, Pablo. 2008. Augmented Sculpture. (2008). Retrieved September 17, 2015 from <http://www.pablovalbuena.com/selectedwork/augmented-sculpture-v1/>.
40. Lance Williams. 1978. Casting Curved Shadows on Curved Surfaces. *SIGGRAPH Comput. Graph.* 12, 3 (Aug. 1978), 270–274. DOI: <http://dx.doi.org/10.1145/965139.807402>
41. Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. 2013. PneuUI: Pneumatically Actuated Soft Composite Materials for Shape Changing Interfaces. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 13–22. DOI: <http://dx.doi.org/10.1145/2501988.2502037>