

inFORM: Dynamic Physical Affordances and Constraints through Shape and Object Actuation

Sean Follmer* Daniel Leithinger* Alex Olwal Akimitsu Hogge Hiroshi Ishii

MIT Media Lab

75 Amherst Street, Cambridge, MA 02139, USA

{sean, daniell, olwal, ishii}@media.mit.edu

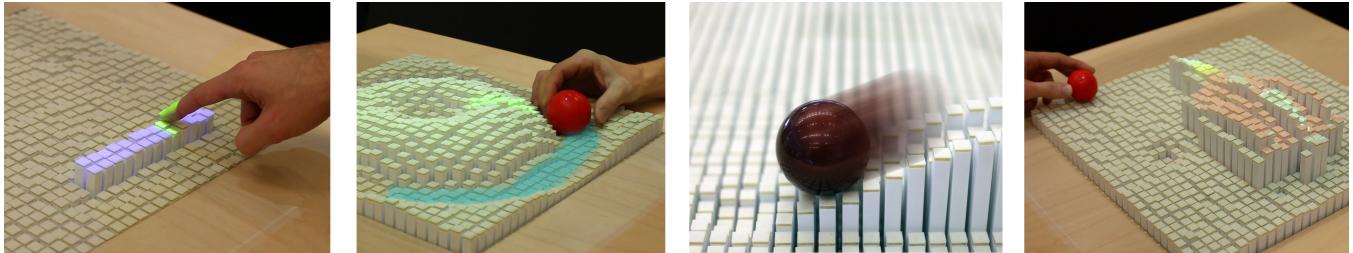


Figure 1: inFORM enables new interaction techniques for shape-changing UIs. *Left to right:* On-demand UI elements through *Dynamic Affordances*; Guiding interaction with *Dynamic Constraints*; Object actuation; Physical rendering of content and UI.

ABSTRACT

Past research on shape displays has primarily focused on rendering content and user interface elements through shape output, with less emphasis on dynamically changing UIs. We propose utilizing shape displays in three different ways to mediate interaction: to *facilitate* by providing dynamic physical affordances through shape change, to *restrict* by guiding users with dynamic physical constraints, and to *manipulate* by actuating physical objects. We outline potential interaction techniques and introduce *Dynamic Physical Affordances and Constraints* with our inFORM system, built on top of a state-of-the-art shape display, which provides for variable stiffness rendering and real-time user input through direct touch and tangible interaction. A set of motivating examples demonstrates how dynamic affordances, constraints and object actuation can create novel interaction possibilities.

Author Keywords

Shape-changing User Interfaces, Shape Displays, Actuated Tangible Interfaces, Token and Constraint Interfaces.

ACM Classification Keywords

H.5.2 User Interfaces: Graphical user interfaces, Input devices and strategies, Interaction styles.

*The first two authors contributed equally to this work.

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.

UIST'13, October 8–11, 2013, St. Andrews, United Kingdom.

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

ACM 978-1-4503-2268-3/13/10...\$15.00.

<http://dx.doi.org/10.1145/2501988.2502032>

INTRODUCTION

The rich variety of physical forms found in everyday life often serve both functional and aesthetic roles. These physical objects have features that not only provide functionality, but also suggest possible uses, or confine the ways we may interact with them; Norman labels these as perceived affordances [30]. This notion of perceived affordances has been long appropriated by the HCI field, particularly in the context of Graphical User Interfaces (GUI) and Tangible User Interfaces (TUI) [17]. While GUIs have the ability to change perceived affordances rapidly to adapt them to different content and context, TUIs primarily exploit the affordances inherent in physical form, as well as their physiological and cognitive advantages [21]. For example, the Token and Constraint framework introduced by Ullmer uses mechanical constraints to provide physical affordances for interacting with tangible controllers, such as tokens [38]. However, TUIs, such as those outlined by Ullmer, are often limited by the static nature of most man-made physical artifacts, and thus cannot easily change their form. Therefore, many projects in this area are commonly single-purpose and do not physically reflect changing program states well [15].

To overcome these limitations, we seek to bring the dynamism of visually perceived affordances of GUIs to physical interaction by utilizing shape-changing UIs (see Figure 1). This paper explores Dynamic Affordances which can transform shape, size, location and orientation, in addition to being able to appear and disappear. They provide appropriate affordances on demand by changing their physical properties based on program states and the context of the user or other objects in the interaction area to *facilitate* interaction. Buttons can, for example, grow in size to ease target acquisition, or move out of the way of an object. We also introduce Dynamic Constraints, which help mediate interaction between

the interface and tangible tokens or tools. Dynamic Constraints not only provide affordances to the user, but also serve to mechanically *restrict* object motion. Constraints limit the degrees of freedom through which users interact with the system, allowing for more precise input in each dimension. These build on and expand Ullmer's Token and Constraint work [38].

In addition to creating affordances and constraints for physical objects and tools, we also show how shape change can be utilized to *manipulate* passive objects. Mechanical forces can push objects, causing them to roll, slide, or tumble in one direction or another. Other constraints can be programmed to rotate or raise passive objects. Passive tangible tokens can be moved to maintain state, and devices, such as phones, can be raised to draw attention to them. Our techniques allow for a wide variety of physical objects to be actuated. This opens up interaction possibilities that point towards tabletop systems that can more easily interact with the world around us.

To explore these techniques and interactions, we introduce the inFORM system, a state-of-the-art 2.5D shape display that enables dynamic affordances, constraints and actuation of passive objects. Shape displays allow for more general-purpose shape change than many other actuated or shape-changing interfaces, and thus are ideal research platforms. The inFORM system supports fast 2.5D actuation, malleable input, and variable stiffness haptic feedback. While shape displays still remain limited in scale and cost, this work is an exploration of the interaction capabilities and is meant to inspire further research in this area. Our belief is that shape-changing interfaces will become increasingly available in the future, and this work tries to push towards creating a vocabulary and design space for more general-purpose interaction for shape displays, including rendering of both content and UI elements.

In this paper, we first review related work in physical affordances and constraints, as well as shape-changing interfaces and dynamic affordances. Next, we discuss the design space of Dynamic Affordances and Dynamic Constraints, and provide methods for using these concepts to also actuate objects. We describe three implemented demonstration applications that highlight different aspects of our concepts, followed by a technical overview of our system. Finally we discuss the implications, limitations, and future potential of the inFORM system.

CONTRIBUTIONS

- An exploration of the design space of dynamic affordances and constraints.
- Actuation of physical objects through shape displays.
- State-of-the-art system for fast, real-time 2.5D shape actuation, co-located projected graphics, object tracking, and direct manipulation.
- Three applications that demonstrate the potential of these interaction techniques for HCI.

RELATED WORK

Our work draws heavily from the rich history of affordances. Gibson introduced the theory of affordances as “what [an object or environment] offers the animal, what it provides or furnishes, either for good or ill.” This can be viewed as the set of action potentials for an object [11]. Norman first applied affordance to design and HCI, focusing on “perceived affordances” that the designer creates to provide interaction clues, or suggestions, to the user [30]. Gaver [9] defines his technology affordances as “properties of the world that are compatible with and relevant for people’s interactions.” He emphasizes the importance of perceptible affordances, since mismatched or hidden affordances interfere with the interface’s legibility, which may confuse the user and result in improper operation. Gaver highlights that these affordances can be perceived visually, tactiley, or aurally. He also expands on sets of affordances: *nested affordances*, which are grouped spatially, and *sequential affordances*, which are grouped temporally. Sequential affordances are often used in GUI based interaction, where graphical perceived affordances can be rendered quickly and then disappear.

Hartson elaborates on Norman and Gaver’s work describing four types of affordances as *cognitive affordance*, “Design features that helps users in knowing something”, *physical affordance*, “Design features that helps users in doing a physical action in the interface”, *sensory affordance*, “Design features that helps users sense something ” and *functional affordance*, “Design features that helps users accomplish work (i.e., the usefulness of a system function)” [13]. Kaptein further splits both cognitive and physical affordances in two parts, describing the *handling affordance*, the affordances a user interacts with, and the *effector affordance*, the affordances a tool manipulates an object with. They suggest that these must be tightly coupled [20]. Feedback and feedforward also play a large role in affordance, and Vermeulen positions feedforward in the context of affordance, and reviews feedback and affordances [40].

In this paper, we focus on rendering both *cognitive affordances* and *physical affordances* through a shape display. These affordances can be perceived both tactiley and visually. Physical form can be used as a cognitive affordance, i.e., rendering a physical play button shaped like a triangle, which could be primarily perceived visually but also tactiley. But, we also render *physical affordances* on demand, which help guide the user’s motion physically or mechanically. These physical affordances can be both handling affordances or effector affordances, which is often the case with dynamic physical constraints. Our dynamic affordances also draw heavily on Gaver’s sequential affordances.

Static Affordances: Tangibles and Graspables

Fitzmaurice et al. [7] demonstrate the benefits of graspable UIs with Bricks, where physical affordances are mapped to the control and representation of virtual objects and abstract actions. SLAP widgets [44] demonstrate how passive controls can provide control and input affordances through optical tracking and sensing.

DataTiles [35] are tangible toolglasses [3], which can be arranged on a screen and act as transparent lenses or props that guide the user's pen interaction with the underlying virtual content. The different tile types embed grooved widgets that physically steer the user's interaction through motion constraints. The Token+constraint framework [38] explores this design space further, highlighting the importance of mechanical constraints, and allows users to build database queries by arranging tangibles where rules and meaning are inferred from spatial relationships.

Haptics

In Haptics [37], a proxy device is typically used to render tactile illusions of physical touch, collisions and constraints in the interface. Electromagnetism can, for example, be used to control a finger or device with an attached magnet [43, 41]. Such haptic interfaces can effectively provide guidance and feedback, but lack perceivable affordances in their static state; the system can only suggest operation during interaction or movement. Tagged handles [27] approach this limitation by having the user experience tangibles with functional form attached to the actuated devices to enhance perceptibility.

Actuated and Dynamically Controlled Objects

A natural extension to user manipulation of TUIs is the ability to computationally control them through actuation. The Actuated Workbench [31] uses electromagnetism for 2D movement of tracked tangibles on an interactive surface, while Madglibs [42] extends the concept to enable height actuation, control of mechanical mechanisms, and to power circuits through induction, in passive tangible assemblies. PICO [32] introduces mechanical constraints as a direct way for the user to specify behavior and rules for actuated tangibles. While the system can communicate system constraints through graphics and object actuation, it is limited in its ability to manifest mechanical constraints; only the user can create those. Other techniques for moving objects on a 2D surface include vibration [36] and robotics [22].

Dynamic, Computationally Controlled Affordances

There is even more interesting potential in the computational control of physical form to adapt and change affordances as a response to both context and user. Coelho and Zigelbaum [5] and Rasmussen et al. [34] review the design spaces for shape-changing interfaces, where actuation actively modifies the shape of an interface or object. Ishii describes his vision of Radical Atoms, and suggests Dynamic Affordances as an interesting area of further research [16]. Most current shape-changing interfaces that address on-demand affordances provide a specific transformation, which limits their use for general purpose UIs and 3D interaction. The haptic chameleon by Michelitsch et al. [29] introduces the concept of shape-changing control devices and reports on experiments with early prototypes. Hemmert et al. [14] manipulate tapering and weight shift in conceptual mobile devices. Bubblewrap [2] embeds electromagnetic actuators in textiles to control form, while MudPad [19] uses magnetorheological fluid to manipulate viscosity. Jamming User Interfaces

can change the stiffness of an input device to change its affordances through particle jamming [8]. Harrison and Hudson [12] employ pneumatics, whereas Tactus Technologies [4] use microfluidics, to inflate predefined physical buttons, for on-demand tactile affordances on touch screens. Madglibs also provide affordances dynamically, by moving them around a surface or mechanically raising or locking elements [42].

Shape displays, which enable more general topologies and greater degrees-of-freedom, tend to primarily focus on content representation through graphics and shape; the generated shapes can respond to the user's touch [18], gestures [25], or other objects' presence [23]. Poupyrev et al. do, however, mention the potential for on-demand UI elements in the description of Lumen, a 13×13 array of actuated illuminated rods [33].

This work builds on the rich literature of affordances and also strongly on the Token+constraint framework [38], but expands on it by considering dynamic changes. The use of a general-purpose 2.5D shape display allows us to support dynamic adaptation of the form, based on user interaction, application context and scenario. The combination of dynamic surface topologies, actuated control of passive tangible objects, user sensing, and object tracking, provides a rich set of capabilities for dynamically controlled perceptible affordances that can optimize user guidance and interaction.

DYNAMIC AFFORDANCES AND CONSTRAINTS

Past research on shape displays has primarily focused on rendering content through shape output, with less emphasis on investigating dynamically changing UI elements. We propose an analysis of dynamically generated physical features with specific affordances that guide the user on how the system can be used and provide passive haptic feedback, enabling interaction at a lower cognitive cost [21]. We believe that shape displays need to provide three types of functionality for creating dynamic UIs: to *facilitate* through Dynamic Affordances, to *restrict* through Dynamic Constraints, and to *manipulate* passive objects through shape change.

Facilitate: Dynamic Physical Affordances

Dynamic Affordances function both as perceived affordances and “real” affordances, as they are rendered physically and provide mechanical support for interaction. We can combine graphical perceived affordances with Dynamic Affordances, or switch between these states.

In the inFORM system, these affordances are physical elements that the user can touch. Depending on how they are rendered by the system, they either directly react to touch, or react to displacement from the user pushing or pulling them. Figure 2 depicts a set of different dynamic affordances rendered on our system. Examples of UI controls with dynamic affordances that our system supports are:

Binary Switches: Buttons

Buttons are formed by raising pins from the surrounding surface. Users activate a button by touching it or by pushing it into the surface, which is registered as a binary input.

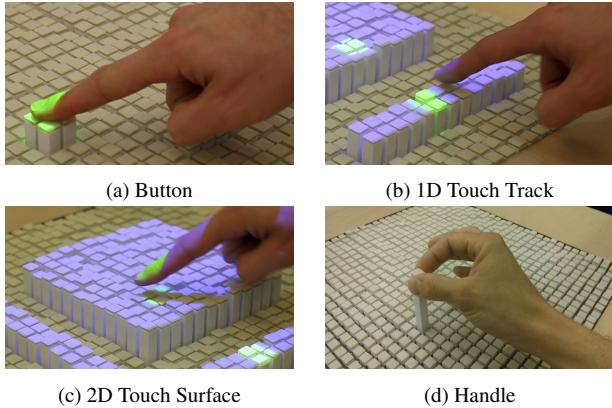


Figure 2: *Dynamic Physical Affordances* transform the UI to facilitate interactions.

1D input: Touch tracks

Touch tracks consist of a line or curve of adjacent raised pins, which the user can touch at different locations, or slide over. These touch points are registered in one input dimension.

2D input: Touch surfaces

Touch surfaces are created using multiple pins, which are aligned to form surfaces. These surfaces, which can be non-planar, map each touch point to two dimensions.

Handles

Handles provide interaction in the Z dimension. These raised pins can be grabbed, and then pulled up or pushed down along one dimension.

Interactions with Dynamic Affordances

Affordances can change shape to reflect a changing program state. For example, when a user presses a play button (triangle shape) it can transform into a stop button (square shape).

Shape-changing affordances can also enable smooth transitions between input dimensions. For example, pressing a button could cause it to transform into a 2D touch panel.

It can be advantageous to let the user's proximity inform shape change. Affordances can, for example, increase in size as a user's hand approaches them, making them easier to acquire according to Fitt's law [28].

Affordances can also move out of the way of physical objects, or rearrange to provide more space for interaction. Besides avoiding physical objects, affordances can compliment them to increase their functionality. As a physical object is moved, the affordances can appear or follow it. For example, as a device is placed on the table, relevant physical UI controls can appear; a phone could be complemented with a large answer button next to it, or a tablet could have buttons to control games appear around it.

The inFORM system is particularly well-suited to guide complex interactions and adapt the affordances, when the user may perform multiple actions using the same controls. Gaver refers to *sequential affordances* [9] in situations where the affordances change based on the interaction, to enable new

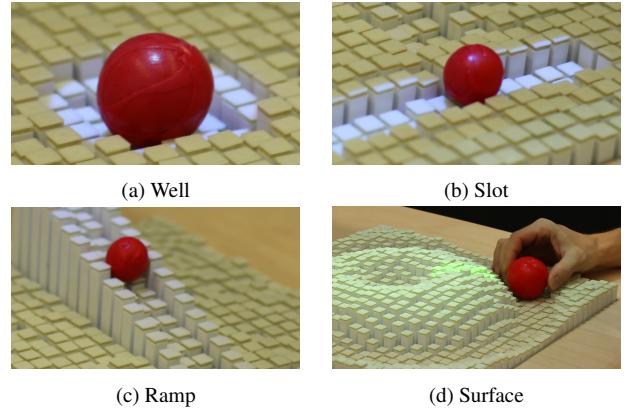


Figure 3: *Dynamic Physical Constraints* guides the user by limiting possible interactions.

possibilities or restrict actions. A UI control can, for example, be rendered with different stiffness to provide further affordances. Two flat surfaces might appear the same, and both afford touching, but once the user touches their surface, a stiff surface affords touch interaction, where a more compliant surface affords pressing. More interestingly, with systems like inFORM, such qualities can be dynamically changed and updated based on interaction and context.

Restrict: Guiding Interaction with Dynamic Constraints

While Dynamic Affordances facilitate user interactions, Dynamic Constraints limit the possibilities, making some interactions difficult or impossible to perform. These Dynamic Constraints make the system more legible, but also guide the user in performing certain interactions through physical interaction with the constraints. They can also help mediate interaction through tangible tokens or tools.

When an object is placed on a shape display, it physically interacts with the shapes generated by the display. In the context of our work, we refer to the physical objects as tokens and the shapes interacting with them as constraints. Figure 3 depicts different types of constraints. Constraints like wells, slots and ramps limit the movement of the token through their shape, thus guiding user interaction, similar to [38].

As our system can sense how tokens interact with constraints, it can dynamically modify their parameters (shape, size, location, orientation) to adapt to user input or to reflect changing program states. Examples of techniques to guide interactions using shape change:

Holding Tokens and Sensing Presence: Wells

Wells act as containers to hold objects. Placing an token inside a well or removing it, is sensed as a binary action. The shape of the well and the shape of the token determine if the token can be rotated in the well, which adds another degree of freedom.

Restricting Movement to 1D: Slots

Slots are grooves which constrain the direction in which a user can move a token. In addition to the same actions that

are supported by the wells, tracked tokens can also be moved in the slots to, for example, control a 1D parameter.

Affecting Movement: Ramps and Curved Surfaces

The surface geometry can be changed to make it easier or harder for the user to move tokens in a certain direction. Ramps can, for example, be used to facilitate or restrict movement of a token due to gravity.

Interaction with Dynamic Physical Constraints

Wells can transform in size and shape to adapt to the size, shape and number of tokens. They can also deepen to move tokens outside the users reach (see Figure 4).

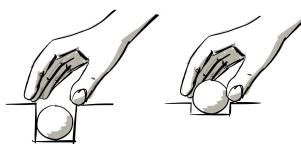


Figure 4: The depth of a well affords if the user can grasp a token contained in it.

Slots change in size, shape and location to reflect a changing program state. A user can, for example, place a token inside a well, which is equivalent to selecting a top level menu item. The well then transforms into a slot, similar to an expanding menu. As the user moves the item inside the well, its shape can transform and branch out to present selectable options.

Slots can also transform their shape to promote the movement of tokens in a certain direction or to hinder it. An example is shown in Figure 5 (right), where a ramp-shaped slot allows users to roll a token with ease in the direction sloping downwards, while requiring deliberate effort to move it in the upwards direction. In addition to ramps, slots can contain further constraints to provide haptic feedback as the user moves the token through the slot. Such feedback can be provided through slots with detents at the bottom (Figure 5, left), ramps with drops, or vibration of the entire slot.



Figure 5: Slots with indentations and ramps can be used to guide the user's interaction or to provide haptic feedback.

Manipulate: Actuating Objects with Shape Displays

Shape displays can appropriate passive objects by independently actuating and manipulating them to create dynamic affordances and constraints. This way, passive objects can be augmented with dynamic capabilities, expanding their possible use as tangible tokens or tools that represent program state or other functionality.

The shape display can apply mechanical force to an object and cause it to move in a variety of ways. This greatly expands opportunities for interaction, and inter-material interaction, as well as solving a problem inherent in passive tangible systems: keeping tokens' physical state synchronized with the digital state. Additionally, it allows the shape display to output greater degrees of freedom (e.g., lateral movement), and enables greater degrees of freedom afforded to the user for input. Our techniques for actuating passive objects do not require an active or special material (such as magnets), but instead manipulate geometrical shapes, with the limitation that certain geometries (such as a ball) are easier to move than others. Other factors to consider include the mass of the object, the force of the motors, and the friction between the shape display surface and the passive object.

Manipulating Objects on the Surface through Actuation

Any object can be lifted vertically by the system as long as the actuation force is sufficient (1.08 N/actuator for inFORM). Objects placed on the table can also be tilted to lean in one direction, with computational control of tilt angle and orientation. This can be used, for example, to orient an object's surface towards a user.

In addition to lifting and tilting, objects can be translated on the X-Y surface through three techniques. Firstly, objects can be lifted and caused to slide or roll down an inclined plane rendered by the surface, essentially using gravity to cause it to move (see Figures 1 and 9). Secondly, given the right shape, the vertical actuator movement can push an object sideways or induce rolling (see Figure 6). This works by applying a force on the object offset from its center of mass, inducing a moment on the object, either causing it to roll or slide out of the way. This actuation method works best for objects with angled or rounded features, like spheres, cones and cylinders. To ensure overlap with at least four pins at a time, our current system actuates spherical objects of at least 25.4 mm diameter. Thirdly, tilt can be used for controlled tumbling of objects about the X- or Y-axis, by alternating tilting and catching, to move the object on the surface. Tumbling works well for cubes or other angular geometry. These different techniques have varying levels of legibility to the user—the inclined plane, for example, makes it clear where the ball will move.

Objects can also be rotated about the surface's Z-axis through similar techniques as used for X-Y translation. Currently only certain objects, with conical or rectangular shapes can be rotated. However, a simple surface feature to allow locking an object at an anchor point could allow for rotation through the aforementioned methods of pushing-induced sliding.

In-air movement through Ballistics

Objects can be projected ballistically from the table into air. The use of multiple pins allows the launch angle to be computationally controlled. With sufficient tracking, the object could also be smoothly caught to dampen impact and avoid subsequent bounces. Currently, we are able to launch a 7 g ball with 20 mm diameter, approximately 80 mm above the maximum pin height of the surface.

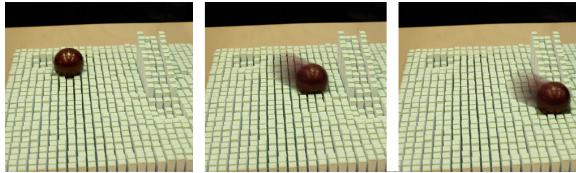


Figure 6: inFORM can lift, translate, tilt, or rotate objects on the surface by changing the surface geometry. Here, a ball is moved on the surface.

Vibrations for Haptic Feedback or Attention

Vibrations by the actuators underneath an object can cause the object to shake. This can provide haptic feedback or draw attention to that object.

User Interaction with Actuated Objects

Similar to PICO [32], users can physically stop the interface from actuating objects by lifting them or holding them in one position. This can be used to control interface behavior of moving parts. In addition, users can place static physical barriers on the table to prevent tokens from moving. These objects can be of arbitrary shape or make use of the space in-between the actuators. An example is a wall barrier made from a cardboard sheet, or a cup that is placed over an object to prevent it from being moved. The constraints defined by the shape display surface could also be user defined, for instance, by deforming the shapes directly with bare hands.

Parameter space

The physical properties of a UI element strongly influence its perceived and real affordances. In his definition of affordances, Gibson [10] lists a number of such properties: “When the constant properties of constant objects are perceived (the shape, size, color, texture, composition, motion, animation, and position relative to other objects), the observer can go on to detect their affordances.” This definition includes dynamic parameters like motion and animation, which static TUIs do not possess as real affordances. Shape-changing interfaces, on the other hand, have the ability to add such parameters, and can be categorized as changes in orientation, form, volume, texture, viscosity, spatiality, adding/subtracting, and permeability [34].

We find it attractive to utilize 2.5D shapes display to render physical affordances, as their hardware capabilities enable simultaneous control over multiple parameters. In the following list, we identify parameters that both contribute to affordances and can be dynamically controlled by inFORM.

Shape: The shape of UI elements can provide multiple affordances, real affordances (how the shape can be touched), and cultural affordances (what the shape represents). Their quality and expressiveness is tightly coupled to the possible resolution and degrees of freedom [25]. This creates interesting interface design challenges that must be considered, in particular, for 2.5D shape display hardware.

Size: The size of a UI element is constrained by the user’s physiology and available space. It has to be sufficiently large for the user to manipulate, while also small enough to fit in

the interface. While static physical UIs have to compromise between these two factors, dynamically resizing UI elements can enable better ergonomics and use of space, given that they can provide sufficiently smooth and continuous transitions at their spatial and temporal resolution.

Position and Orientation: The spatial relationship between objects is an important parameter for TUIs. While these can be dynamically modified, users cannot easily grasp, lift and rearrange objects on 2.5D shape displays. Therefore, we propose to complement them with passive physical tokens that enable these interactions, while they can also be constrained and actuated by the display surface shape.

Color: Color and visual texture can be applied to provide additional graphical perceived affordances, using embedded display [33], projection [25], or augmented reality [24].

Haptic feedback: The material and haptic feedback of an element communicates to the user if it affords actions like deformation. Systems with mechanical actuation can provide haptic feedback by dynamically changing the resistance of a pin when pressed by the user.

Visibility: By rapidly changing the size of a dynamically rendered element, it can appear and disappear. An advantage compared to static systems is that physical objects can be rendered in succession, rather than having to permanently share valuable space. The rate of change is hardware dependent.

Motion: Motion describes the change of the above parameters over time. While motion to switch between predefined static affordances has been explored previously [12], carefully choreographed motion adds a compelling dimension to the interaction. If the UI element transforms continuously as the user interacts with it, the motion itself turns into an expressive affordance. The quality of motion in the context of HCI has previously been described by the path, volume, direction, and velocity of an object [39].

DEMONSTRATION APPLICATIONS

3D Model Manipulation (handles, constraints, context)

The 3D Model Manipulation application demonstrates how the inFORM system’s dynamic capabilities can be used to render physical representations of 3D models that the user can flip through, and then use tokens and tools to transform, edit or paint (see Figure 7).

Users move a token through a slot to browse the different 3D models (Figure 1) and can select it by placing the token in an adjacent well (Figure 7a). When the token is placed in the well, it transforms into a slot. Moving the token to any end of the slot will select the function of the token; rotation, translation or scale. The slot transforms again to represent the degrees of freedom of the current mode. In rotation mode, the slot is a circle circumscribing the model (Figure 7c), scale mode uses a linear slot (Figure 7b), while translation mode has no constraints. As the token is moved in these constraints, the object dynamically transforms. To exit the current mode, the user places the token in the *select well*.

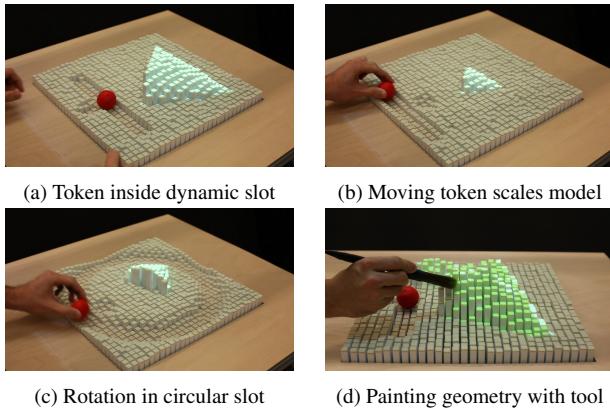


Figure 7: The 3D Model Manipulation application uses tokens and tools to browse, transform and edit physical representations of 3D models.

The user can also use a 3D brush tool to add geometry, erase geometry, or paint on the 3D model, as shown in Figure 7d. To change modes for the brush, the user presses a foot pedal to activate a context menu which is rendered physically as a 2D Dynamic Affordance. The menu appears offset from the current location of the brush in 3D space, allowing for quick selection. The bristles of the brush move smoothly over the shape display surface, while being optically tracked by the system.

Marble Answering Machine

The Marble Answering Machine [6], is a tangible interface to receive, store and play back voice messages that are represented as physical marbles. Its iconic form, sketched by Durrell Bishop, is a continuous surface with a raised hill and a hole in it. Our homage to the original design, implemented with the inFORM system, uses a dynamically changeable form, and demonstrates how Dynamic Physical Constraints transform to reflect changing program states (Figure 8).

New messages are represented by marbles, which are ejected from the hole (Figure 8b) to roll into a newly formed well that stores messages (Figure 8c). To listen to a message, users pick up a marble from the *new message well* and place it inside the *play well* (Figure 8d). The play well transforms into a slot, and as the message is played back, the marble is moved inside the slot, representing its relative playback position. At any given time, users can pick up the marble to stop the message, or scrub it in the playback slot to replay parts of the message. Once the message is played, the machine moves the marble to the *old message well*. Dropping the marble back into the hole on top of the machine erases it.

This system demonstrates the ability to render both aesthetic form and UI elements. In addition, it highlights the ability for the user to directly intervene while an object is being actuated. In the spirit of PICO [32], the user can also introduce mechanical constraints to control computational behavior. By placing a rigid sheet in the path of the playback track, the user can limit message playback to the beginning of the audio file.

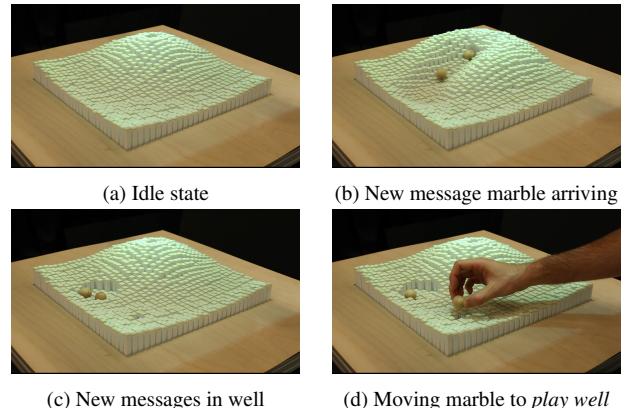


Figure 8: inFORM's version of the Marble Answering Machine uses dynamic shape and constraints to reflect program state.

Actuating Interactive Devices on the Surface

Beyond passive tokens, we can also actuate interactive devices, such as phones or tablets. Such devices can be similarly moved, tilted, rotated, and vibrated. A smartphone could, for example, be tilted towards the user upon a phone call or alert (see Figure 9), or to preserve privacy in a collaborative session when private information should not be seen by others.

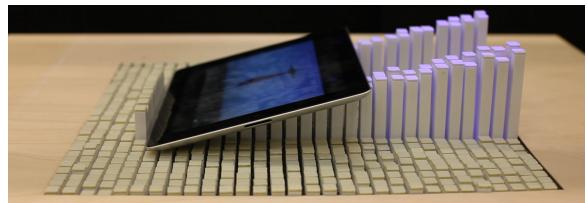


Figure 9: inFORM can actuate devices, for example, by sliding and tilting a tablet towards the user.

The dynamic capabilities of inFORM could be used to render physical UI elements on-demand, for example, by having physical buttons and touch elements emerge when a tablet is placed on the surface. These UI elements provide physical affordances for applications on the tablet. Vibration and haptic feedback could be used to augment gameplay, for instance.

IMPLEMENTATION

Shape Display

The system uses 30×30 motorized white polystyrene pins, in a 381×381 mm area. The pins have a 9.525 mm^2 footprint, with 3.175 mm inter-pin spacing, and can extend up to 100 mm from the surface. Push-Pull rods are used to link each pin with an actuator, to enable a dense pin arrangement independent of actuator size, giving the system a height of 1100 mm. The linkage, a nylon rod inside a plastic housing (Sullivan Gold-N-Rods), transmits bi-directional force from a motorized slide potentiometer (ALPS RSA0N11M9A07), through a bend. Six slide potentiometers are mounted onto a custom-designed PCB, powered by an Atmel ATMega 2560, and TB6612FNGCT-ND motor drivers. The linear positions

are read by the 10-bit A/D converters on the microcontroller, and allow for user input, in addition to servoing their position using PID control.

150 boards are arranged in 15 rows of vertical panels, each with 5×2 boards. The boards communicate with a PC over five RS485 buses bridged to USB. The system has a 60 Hz refresh rate, determined by the 115200 bps RS485 bus speed, the 8 byte control message, and 30 boards on each RS485 bus.

For each pin, we can update both position and PID terms to provide haptic feedback and variable stiffness, for example, to create haptic detents or buttons that are harder to press. This control also allows us to limit power consumption per pin to avoid burning out the motors.

Pin height and stiffness is represented in software as an 8-bit height map, which can be produced by OpenGL shaders, through different shape primitive classes, or by directly writing data. The height map is then sent to the microcontrollers. Similarly, all 900 pin heights can be received over the RS485 bus, and used to track user's surface deformations as input.

Each pin can exert a force of 1.08 Newtons (equivalent to 100 g weight), which was measured using a precision digital scale. The effective average upwards and downwards speeds (0.644 m/s and 0.968 m/s, respectively) were measured using a high speed camera.

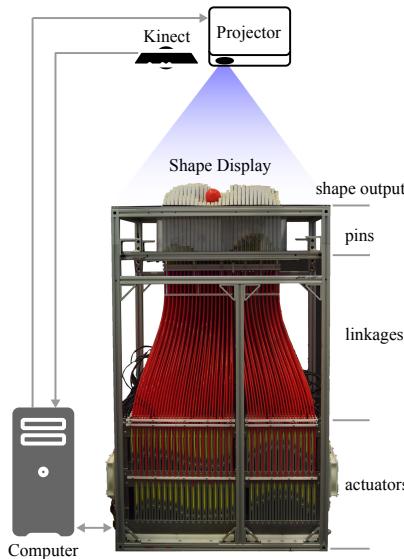


Figure 10: The inFORM system actuates and detects shape change with 900 mechanical actuators, while user interaction and objects are tracked with an overhead depth camera. A projector provides visual feedback.

In theory, the system's 900 pins could consume up to 2700 W due to a peak 3 W power consumption per actuator. In practice, however, we measure the effect to approximately 700 W when in motion. Due to friction, the pins maintain their static position unpowered. Therefore, the power consumption to maintain a shape is less than 300 W (mainly due to the 83 %

efficiency of our 12 V power supplies). Heat dissipation remains a critical design criteria for the actuation assembly and we use two rows of five 120 mm fans to cool the actuators.

User and Object Tracking

Our current system uses an overhead depth camera to track users' hands and surface objects, as shown in Figure 10. A Microsoft Kinect with a 640×480 pixel depth sensor is mounted 1200 mm above the surface and calibrated for extrinsic and intrinsic camera parameters. We combine a static background image of the table surface with the surface's real-time height map to form a dynamic background image that is used for subtraction and segmentation. The height map is scaled and a homography is applied to warp the image into camera space. We find all pixels above the computed dynamic background model, and threshold the image. For hand and fingertip tracking, we use OpenCV to compute contours of the threshold image, followed by convex hull and convexity defects. The 3D finger tip coordinates are transformed to surface space, and an arbitrary number of finger tip contacts can be tracked and detected to trigger touch events. For object tracking, we currently rely on color tracking in the HSV space. However, other approaches to object tracking would be straightforward to implement. Objects and finger positions are tracked at 2 mm resolution in the 2D surface plane, and at 10 mm in height. The touch tracking can be combined with input from the slide potentiometers to, for example, distinguish button presses from light touch.

Graphics and Projection

Graphics are rendered using OpenGL and openFrameworks. The projector's world coordinates were determined after calibration with the Kinect color camera, while distortion coefficients and intrinsic parameters were recovered using ProCamCalib [1]. The projector's 1400×1050 pixel output is then warped to correct for the parameters and projected over an 87×66 cm area.

Limitations

As an alternative to the limited depth camera touch tracking, it may be interesting to embed touch sensors, e.g., capacitive or optical, directly in the pins. More sophisticated object tracking, for example, using multiple cameras, markers, or sensing in the electromagnetic or RF domain, could be used to address user occlusion, object identification, and color interference with projector.

The limited scalability due to the current architecture with one actuator per pin, has significant implications on cost and footprint to move beyond the current relatively low resolution. Also, the current implementation is limited to 1D input and output for each pin, which would be difficult to increase without significantly increasing complexity. While stiffness can be controlled, pin shape, spacing and material choices limit the affordances and constraints that the system can generate and how they interact with external objects. We, however, believe that the system's spacing, resolution and the interactive speeds rendered, are sufficient to allow the prototyping of many interactions that would be challenging on other

existing shape displays. The current cost and scale of our shape display hardware limits its primarily use to research.

DISCUSSION AND FUTURE WORK

Shape displays allow for new ways to create physical interfaces, beyond functionality alone. Aesthetic form is an important part of many of the devices and objects that we interact with on a daily basis. Shape displays begin to let interface designers create radically different physical forms for different applications. The Marble Answering Machine example points towards this type of use, where form is more than functional; it is also evocative and emotional. This introduces an opportunity for physical motion design. It also points towards uses of shape displays for prototyping new physical interfaces.

While we have yet to perform a formal quantitative user evaluation, the system has been used by roughly 50 people and we have collected qualitative feedback from these encounters. Users tried the 3D Model Manipulation application and an example program to move passive objects on the surface autonomously. Initial feedback was generally very positive, with users commenting on the advantages of having physical UI elements appear and transform on demand, as well as expressing general delight with the autonomous movement of passive objects on the table. However, we observed that in the 3D Model Manipulation application, users sometimes struggled with physically overlapping content and UI elements. While we believe the main reasons to be the limited resolution of the shape display hardware and the software not adapting well to content changes, solving the physical overlap of content and interface elements rendered both as shapes is a very interesting new challenge of such interfaces.

We also noticed on multiple occasions how rapid shape transitions were jarring to users, an observation we have made in earlier studies as well [24]. The question remains how to best communicate shape transitions to the user before they occur, to avoid surprise. We see this question as an important next step for research on shape displays. One potential solution is for smooth, slow transitions, that can be explored calmly, as in Lumen [33].

Smooth transitions may not suffice to adequately inform the user; possibilities for shape change may need to be more legible. As Gaver explains: “Affordances are not passively perceived, but explored” [9] and we must find a way for these new affordances to gracefully be explored, potentially by more tightly coupling their motion to the motion of the user. We think this is a rich area for future exploration. For example, it can be viewed as a feedforward problem [40]. Or, one potential direction to explore legibility for potential shape change could be to combine shape change with augmented reality, similar to [24]. More theoretically, considering the shape display as an autonomous agent, may suggest looking towards research in human–robot interaction, where robots may want to convey to the user how they will move through more subtle means.

Along these lines, the legibility of how a passive object will move on the table is of interest. We described two ways to

move a ball on the shape display surface: pushing the ball or rolling it down a slope. Rolling down a slope is much more legible to the user; toy marble runs are very legible, because the marble only has one path and the user can easily follow its trajectory, which is powered only by gravity. A designer, using the inFORM system, can make a ball’s trajectory more legible by creating a slot that it will move in. But, such features for legibility alone may take up space and not scale well. New interaction techniques can be explored to address this legibility.

We believe that the *facilitate*, *restrict* and *manipulate* techniques described here are merely one part of a larger space of Dynamic Physical Affordances, which will emerge as shape-changing UIs mature. The Dynamic Physical Affordances in this paper focus on affordances rendered on a 2.5D surface. However, it is interesting to look at the larger space of possibilities for actuation and shape change: the user, the tool handle, tool, object, and physical surface must be considered. In this work, we have focused on dynamically changing the physical surface, but these other areas and their combinations provide many interesting possibilities for new interactions. For example, a tool can change shape as the interaction surfaces change shape as well. It is at these intersections between different materials and different interaction elements where shape change and actuation begin to open new opportunities for human–computer interaction. This suggests the importance of considering the whole ecology of interaction and interactive devices. We believe that proxemic interaction for shape-changing UIs is another important area to explore, as well as multi-user collaboration, co-located and remote.

CONCLUSIONS

In this work we have explored the design space of Dynamic Physical Affordances and Constraints, and described methods for actuating physical objects on actuated shape displays. Many prior approaches to shape-changing user interfaces have relied on special-purpose or bistable shape change. Instead, we explored dynamic shape change’s more general-purpose role, similar to the flexibility of a bitmap screen for GUIs. This opens possibilities for using shape change both for content and as a UI elements. Dynamic Physical Affordances and Constraints encompass a large space of features that we hope will inspire designers when creating shape-changing interfaces.

ACKNOWLEDGMENTS

We thank Ryan Wistort, Guangtao Zhang, Cheetiri Smith, Alyx Daly, Pat Capaloung and Jason Moran for work on inFORM hardware, and Mark Feldmeier for advice. This research was supported in part by the National Science Foundation Graduate Research Fellowship under Grant No. 1122374. Alex Olwal was supported by a Swedish Research Council Fellowship and a Blanceflor Foundation Scholarship.

REFERENCES

1. Audet, S., and Okutomi, M. A user-friendly method to geometrically calibrate projector-camera systems. In *IEEE CVPR 2009* (2009), 47–54.

2. Bau, O., Petrevski, U., and Mackay, W. Bubblewrap: a textile-based electromagnetic haptic display. In *CHI EA '09*, ACM (2009), 3607–3612.
3. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93*, ACM (1993), 73–80.
4. Ciesla, C. M., and Yairi, M. B. User interface system and method, 07 2009.
5. Coelho, M., and Zigelbaum, J. Shape-changing interfaces. *Personal Ubiquitous Comput.* 15, 2 (Feb. 2011), 161–173.
6. Crampton Smith, G. The hand that rocks the cradle. *ID magazine* (1995), 60–65.
7. Fitzmaurice, G. W., Ishii, H., and Buxton, W. A. S. Bricks: laying the foundations for graspable user interfaces. In *CHI '95*, ACM Press/Addison-Wesley Publishing Co. (1995), 442–449.
8. Follmer, S., Leithinger, D., Olwal, A., Cheng, N., and Ishii, H. Jamming user interfaces: programmable particle stiffness and sensing for malleable and shape-changing devices. In *UIST '12*, ACM (2012), 519–528.
9. Gaver, W. W. Technology affordances. In *CHI '91*, ACM (1991), 79–84.
10. Gibson, J. J. *The Senses Considered as Perceptual Systems*. Houghton Mifflin Co., 1966.
11. Gibson, J. J. *The ecological approach to visual perception*. Psychology Press, 1986.
12. Harrison, C., and Hudson, S. E. Providing dynamically changeable physical buttons on a visual display. In *CHI '09*, ACM (2009), 299–308.
13. Hartson, R. Cognitive, physical, sensory, and functional affordances in interaction design. *Behaviour & Information Technology* 22, 5 (2003), 315–338.
14. Hemmert, F., Hamann, S., Lwe, M., Wohlauf, A., Zeipelt, J., and Joost, G. Take me by the hand: Haptic compasses in mobile devices through shape change and weight shift. In *NordiCHI '10* (2010).
15. Ishii, H. Tangible bits: beyond pixels. In *TEI '08*, ACM (2008), xv–xxv.
16. Ishii, H., Lakatos, D., Bonanni, L., and Labrune, J.-B. Radical atoms: beyond tangible bits, toward transformable materials. *interactions* 19, 1 (Jan. 2012), 38–51.
17. Ishii, H., and Ullmer, B. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97*, ACM (1997), 234–241.
18. Iwata, H., Yano, H., Nakazumi, F., and Kawamura, R. Project feeflex: adding haptic surface to graphics. In *SIGGRAPH '01*, ACM (2001), 469–476.
19. Jansen, Y., Karrer, T., and Borchers, J. Mudpad: tactile feedback and haptic texture overlay for touch surfaces. In *ITS '10*, ACM (2010), 11–14.
20. Kaptelinin, V., and Nardi, B. Affordances in hci: toward a mediated action perspective. In *CHI '12*, ACM (2012), 967–976.
21. Klemmer, S. R., Hartmann, B., and Takayama, L. How bodies matter: five themes for interaction design. In *DIS '06*, ACM (2006), 140–149.
22. KOJIMA, M., SUGIMOTO, M., NAKAMURA, A., TOMITA, M., INAMI, M., and NII, H. Augmented coliseum: An augmented game environment with small vehicles. In *TABLETOP '06*, IEEE Computer Society (2006), 3–8.
23. Lee, N., Kim, J., Lee, J., Shin, M., and Lee, W. Molebot: mole in a table. In *SIGGRAPH '11*, ACM (2011), 9:1–9:1.
24. Leithinger, D., Follmer, S., Olwal, A., Luescher, S., Hogge, A., Lee, J., and Ishii, H. Sublimate: State-changing virtual and physical rendering to augment interaction with shape displays. In *CHI '13* (2013).
25. Leithinger, D., Lakatos, D., DeVincenzi, A., Blackshaw, M., and Ishii, H. Direct and gestural interaction with relief: a 2.5d shape display. In *UIST '11*, ACM (2011), 541–548.
26. Macknik, S. L., King, M., Randi, J., Robbins, A., et al. Attention and awareness in stage magic: turning tricks into research. *Nature Reviews Neuroscience* 9, 11 (2008), 871–879.
27. MacLean, K. E., Snibbe, S. S., and Levin, G. Tagged handles: merging discrete and continuous manual control. In *CHI '00*, ACM (2000), 225–232.
28. McGuffin, M., and Balakrishnan, R. Acquisition of expanding targets. In *CHI '02*, ACM (2002), 57–64.
29. Michelitsch, G., Williams, J., Osen, M., Jimenez, B., and Rapp, S. Haptic chameleon: a new concept of shape-changing user interface controls with force feedback. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, ACM (New York, NY, USA, 2004), 1305–1308.
30. Norman, D. A. *The Psychology of Everyday Things*. Basic Books, Inc., 1988.
31. Pangaro, G., Maynes-Aminzade, D., and Ishii, H. The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. In *UIST '02*, ACM (2002), 181–190.
32. Patten, J., and Ishii, H. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *CHI '07*, ACM (2007), 809–818.
33. Poupyrev, I., Nashida, T., and Okabe, M. Actuation and tangible user interfaces: the vaucanson duck, robots, and shape displays. In *TEI '07*, ACM (2007), 205–212.
34. Rasmussen, M. K., Pedersen, E. W., Petersen, M. G., and Hornbaek, K. Shape-changing interfaces: a review of the design space and open research questions. In *CHI '12*, ACM (2012), 735–744.
35. Rekimoto, J., Ullmer, B., and Oba, H. Datatiles: a modular platform for mixed physical and graphical interactions. In *CHI '01*, ACM (2001), 269–276.
36. Reznik, D., and Canny, J. A flat rigid plate is a universal planar manipulator. In *IEEE ICRA 1998*, vol. 2, IEEE (1998), 1471–1477.
37. Salisbury, K., Conti, F., and Barbagli, F. Haptic rendering: introductory concepts. *Computer Graphics and Applications, IEEE* 24, 2 (2004), 24–32.
38. Ullmer, B., Ishii, H., and Jacob, R. J. K. Token+constraint systems for tangible interaction with digital information. *ACM Trans. Comput.-Hum. Interact.* 12, 1 (Mar. 2005), 81–118.
39. Vaughan, L. C. Understanding movement. In *CHI '97*, ACM (1997), 548–549.
40. Vermeulen, J., Luyten, K., van den Hoven, E., and Coninx, K. Crossing the bridge over norman's gulf of execution: revealing feedforward's true identity. In *CHI '13*, ACM (2013), 1931–1940.
41. Weiss, M., Remy, C., and Borchers, J. Rendering physical effects in tabletop controls. In *CHI '11*, ACM (2011), 3009–3012.
42. Weiss, M., Schwarz, F., Jakubowski, S., and Borchers, J. Madgets: actuating widgets on interactive tabletops. In *UIST '10*, ACM (2010), 293–302.
43. Weiss, M., Wacharamanotham, C., Voelker, S., and Borchers, J. Fingerflux: near-surface haptic feedback on tabletops. In *UIST '11*, ACM (2011), 615–620.
44. Weiss, M., Wagner, J., Jansen, Y., Jennings, R., Khoshabeh, R., Hollan, J. D., and Borchers, J. Slap widgets: bridging the gap between virtual and physical controls on tabletops. In *CHI '09*, ACM (2009), 481–490.