GridDrones: A Self-Levitating Physical Voxel Lattice for Interactive 3D Surface Deformations

Sean Braley, Calvin Rubens, Timothy Merritt[1] and Roel Vertegaal

Human Media Lab
Queen's University
Kingston, Ontario, Canada

[1]Dept. of Computer Science
Human Centered Computing, Aalborg University
Aalborg, Denmark

ABSTRACT
We present GridDrones, a self-levitating programmable matter platform that can be used for representing 2.5D voxel grid relief maps capable of rendering unsupported structures and 3D transformations. GridDrones consists of cube-shaped nanocopters that can be placed in a volumetric 1x*n*x*n* mid-air grid, which is demonstrated here with 15 voxels. The number of voxels and scale is only limited by the size of the room and budget. Grid deformations can be applied interactively to this voxel lattice by manually selecting a set of voxels, then assigning a continuous topological relationship between voxel sets that determines how voxels move in relation to each other and manually drawing out selected voxels from the lattice structure. Using this simple technique, it is possible to create unsupported structures that can be translated and oriented freely in 3D. Shape transformations can also be recorded to allow for simple physical shape morphing animations. This work extends previous work on selection and editing techniques for 3D user interfaces.

Author Keywords
Organic User Interfaces; Claytronics; Radical Atoms; Programmable Matter; Swarm User Interfaces.

INTRODUCTION
The creation of bi-directional tangible interfaces has been an enduring research goal [18]. Sutherland [41] envisioned early on that the ultimate form of Virtual Reality (VR) would entail the rendering of physical matter in lieu of virtual pixels. There are two main reasons for this: 1) Physical matter provides haptic feedback that is difficult to simulate in VR; and 2) to achieve symmetry between the ability for physical objects to control software, and software to control physical representations [19]. Toffoli and Margolus [42] coined the term "programmable matter", refining the concept to pertain to massively parallel arrays of physical cellular

UIST '18, October 14–17, 2018, Berlin, Germany

ACM ISBN 978-1-4503-5948-1/18/10…$15.00
https://doi.org/10.1145/3242587.3242658

Figure 1. GridDrones system with an array of self-levitating physical voxels represented by small quadcopters.

automata capable of rendering 3D geometric shapes that, someday, would be of sufficient resolution to be indistinguishable from actual physical objects. The effort towards interactive programmable matter is continuing today, within user interface paradigms such as *Claytronics* [12], *Organic User Interfaces* [32,43], and *Radical Atoms* [19] and studied in related fields such as reconfigurable [25], modular [24,34] and swarm robotics [14,36]. These interfaces are capable of representing physical 3D objects via synchronous movement of large quantities of miniature robots dubbed *Catoms* (Claytronic Atoms) [23]. However, one of the problems with existing programmable matter prototypes is that it is challenging to position Catoms in 3D, especially in the vertical (z) dimension [13,35]. This is because Catoms need to overcome gravity in order to move in the vertical dimension, and because structures need to always remain structurally stable under gravity during deformation. Indeed, when we examine prior work in programmable matter prototypes, such as Kilobots [36], or Zooids [14], we note that these robot swarms, while relatively high resolution, are only capable of rendering 2D structures.

In recent years, there have been significant advances in the research space of Shape-Changing Displays as well. A notable prototype is inFORM [9], a system that relies on a motorized pin relief map to create a Shape-Changing Display using 2.5D surface normal deformations, again at relatively high resolutions. Structural integrity is not an issue with inFORM, because all surface normals are supported physically along the *z* axis. However, approaches relying on motorized pins are limited in that they can *only* render transformations of the actual *surface* of the device: i.e., they cannot render unsupported structures, as holes along the *z* axis are not possible. 3D spatial transformations such as rotation are also challenging with such systems, which requires rotating the entire display mechanism and thus relationally, the voxels are fixed within the shape display [40].

Different approaches to creating three-dimensional displays that use Catoms as physical representations of voxels can be found in modular robotics. M-Blocks [34] is a system in which individual robot cubes use gyros and magnets to move in three dimensions, by jumping and clasping to one another. However, the accuracy of moving the bots by ballistic means leaves much to be desired. Kinetic Blocks [38] proposed a different approach, combining passive modular building blocks actuated by an underlying shape display. Despite addressing some of the limitations associated with 2.5D relief displays, this approach is generally limited to producing shapes that move linearly through vertical movement, and is limited in its ability to render geometries with unsupported structures. For this reason, researchers are making progress in developing self-levitating tangible voxels. Examples include magnetic [26,34] and ultrasonic levitation [31], floating field mechanisms [46], and flying cubes [11]. Specifically, Gomes et al. [11] introduced BitDrones, a platform that uses drones as self-levitating Catoms. Their work showed a system comprising a small number of quadcopters that was able to physically render sparse voxel-based graphics through seamless movement in all three dimensions. The key benefit cited is that when building blocks are self-levitating, the structural integrity of the resulting model need not be guaranteed at every assembly step. However, the sparseness and limited density of the original BitDrones system posed significant challenges to demonstrate even the most basic shape changing display. Note that while drones are capable of creating free-form 3D structures in mid-air, the confines of an indoor operating volume limits direct hovering of voxels over top of one another, due to turbulence. As such, we generally limited GridDrones to operate as a 2.5D display.

Contribution

We present GridDrones (Figure 1), a self-levitating programmable matter platform that can be used for representing 2.5D voxel grid relief maps capable of rendering unsupported structures and 3D spatial transformations. GridDrones consists of 15 cube-shaped nanocopters that can be placed in a room-scale volumetric 1x*n*x*n* mid-air grid. Grid deformations can be applied interactively to this voxel lattice by first selecting a set of voxels using the hand(s), then programming a continuous topological relationship between voxel sets that determines how they move in relation to each other. Using this simple technique, it is possible to create unsupported structures, such as polyhedra, that can be translated and oriented freely in 3D, provided that voxels do not hover directly over each other. Shape transformations can also be recorded to allow for simple physical shape morphing animations. What makes GridDrones significantly different from other programmable drone swarms is that each drone forms an interactive touchable 3D graphics voxel, rather than a quadcopter with a programmed flight path. Unlike earlier systems, this allows GridDrones to be directly manipulated as a bi-directional Tangible User Interface (TUI). Next, we discuss the background literature, after which we discuss the design rationale and interaction techniques. We will subsequently outline the implementation of the system, and conclude with a discussion of limitations and future applications.

BACKGROUND

Our work draws from the vision of *Organic User Interfaces* and *Radical Atoms* and is related to several research areas, namely: shape displays, data physicalization, swarm robotics and self-levitating user interfaces.

Dynamic Shape Displays

There is a large body of research aimed at developing techniques for controlling physical matter as a means to produce interactive volumetric displays. Such systems support discretized shape control of 2.5D surfaces using 2D arrays of linear actuators, while other systems support continuous shape control using hydraulic or pneumatic actuation [10,45] or shape-memory alloys [32] to produce a 2.5D approximation of an object's shape. Lumen [32] explored individual control of shape and graphics by varying the height of a 5x5 array of light guides using shape-memory alloys. Relief [27] investigated a set of common interactions for viewing and manipulating content on a 12x12 shape display. Recompose [6] implemented a framework that allowed for gestural and direct manipulation of an actuated surface. More recently, inFORM [9] showcased a 30x30 shape display aimed at exploring the design space of dynamic physical affordances and constraints within shape-changing user interfaces. While the shape resolution of the above systems has greatly improved over the years, shape displays can still only render certain types of 2.5D shapes and user interface actions such as translation, rotating or scaling of groups of elements are not generally supported.

Physical Representations of Data

In recent years, there has been a growing interest around data physicalization [21]. While physical representations of information have been around for centuries, recent advances in digital fabrication have led to a new generation of shape-changing interfaces that can be used to explore dynamic physical visualizations of digital information. While recent work has investigated the use of shape displays for data exploration [20], the range of visualization and interaction techniques supported by these platforms is inherently limited by the 2.5D nature of the displays. A promising approach to mitigate this drawback can be found in *swarm user interfaces*, defined by Le Goc et al. [14] as "*interfaces made of independent self-propelled elements that move collectively and react to user input*". Swarm user interfaces provide a promising approach to extend physical representations of data to a three-dimensional space, closely approximating the kind of interactive data visualizations present in 3D graphical information displays.

Swarm User Interfaces

Forming complex displays out of a large number of simple robots was first demonstrated with *Kilobot* by Rubenstein et al. [36]. This low cost scalable robot system served as the basis for a dynamic programmable display comprising of a thousand-robot swarm [35]. However, the robots moved very slowly (~1cm/s) and did not respond to user input. Recently, roboticists have started to develop methods for interacting with large collectives of robots that support either direct or gestural user interaction. Alonso-Mora et al. [29] proposed a display in which each pixel is a robot of controllable color. Their system was recently extended to support interaction through sketching and mid-air gestures [2,3]. Zooids [14] extended this approach further, focusing on direct manipulation of tangible robots. A common drawback with such systems, however, is that they are inherently limited to two-dimensional configurations. While there are systems that are able to self-reconfigure in three dimensions [5,39], these approaches are not able to reconfigure in a manner that closely mimics the movement of digital voxels in, for example, a Virtual Reality interface. To achieve such mimicry by tangible physical representations of digital voxels, independent self-propelled elements that can be both directly manipulated by the user and move autonomously in a coordinated fashion are necessary.

Self-Levitating Tangible User Interfaces

A natural extension to swarm user interfaces is that of self-levitating tangible user interfaces, self-propelled robots that are able to arrange spatially to assemble complex structures on the fly, with examples including *Flight Assembled Architecture* [4] and *Termite Inspired Construction* [44]. While these examples provide exciting technical innovation, there has been little focus on interaction with those systems. Researchers have explored the concept of levitating displays via flying robots equipped with projectors [30,37] and high-resolution displays [11,39]. While these explorations enabled user interaction, they were limited to a single drone that primarily acted as a visual information display. An exception to the above work is Drone 100 [5], a platform comprising 100 quadcopters that act synchronously to display images. However, the large scale of this platform prevents direct user interaction and interaction at room scales. Another example of quadcopters working together to represent 3D structures can be found in BitDrones [11]. BitDrones investigated how small quadcopters that serve as self-levitating building blocks can facilitate human-drone interaction via means of direct touch, bimanual input techniques and gestural interactions. However, BitDrones could only be used to represent sparse 3D voxel models due to its small number of simultaneously flying quadcopters.

The present work attempts to mitigate the core limitations of BitDrones by miniaturizing drones, optimizing communications, and creating a 1xnxn grid of 15 drones capable of displaying shape deformations and spatial transformations.

DESIGN RATIONALE

GridDrones consists of a surface grid of 15 BitDrones, each representing a self-levitating physical voxel or Catom. We used the following design principles when designing GridDrones:

Physical Reality vs. Augmented Reality

While Augmented Reality has been shown to be an effective way of controlling drones [16, 22], our design is different in two ways: 1) GridDrones points towards a future in which programmable matter replaces AR altogether, and b) GridDrones were not to be regarded as traditional individual manually controlled drones, but rather, as clouds of 3D tangible voxels. As such, we did not include an AR interface, but rather, used a smartphone to control non-tangible functions of the grid (see below).

Direct Haptic Rendering

GridDrones was designed as a voxel rendering system that directly provides natural haptic feedback by being entirely physical. Touching the drones provides tactile feedback of the voxel shape. Dragging the drones can provide a kinaesthetic resistive force, depending on the scenario. While modulating the propeller spin could be used as a form of vibrotactile feedback, we decided to rely on naturally existing physical cues only.

Self-levitating Voxels

Each voxel in the grid is self-levitating to allow for unsupported structures and 3D spatial transformations without requiring any structural support.

Size Limitations

One of the issues when miniaturizing drones is that the efficiency of components is reduced at smaller scales. We created the smallest possible drones that are still capable of a payload of a battery, RGB Light Emitting Diodes (LEDs) and a carbon fiber wire frame cube held together with 3D printed nylon parts.

Resolution Limitation

We limited our design to 15 drones in the present iteration. This is mostly due to budget and constraints of the physical space in which the drones are flown and we are working on producing larger grids.

2.5D Surface Transformations

We limited our design to a 2.5D 1xnxn grid because our current design does not allow drones to fly over top of one another. When drones are over top one another, the top drone creates downward thrust for drones flying directly underneath it. This makes 3D structures challenging. We

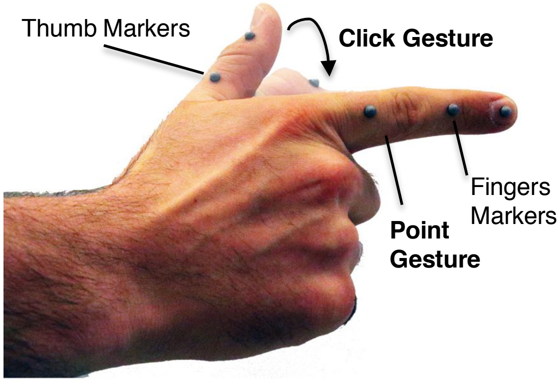

Figure 2. "Point" and "Thumb click" gestures.

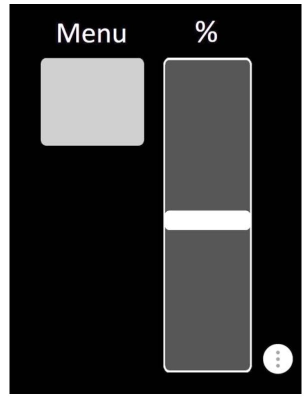

Figure 3. Smartphone app with Topological Programming slider and TUI menu button.

decided to focus on surface deformations to allow for 2.5D self-levitating shape transformations based on those found in relief displays.

3D Spatial Transformations

Self-levitation allows voxels to be suspended without a structural bond to a physical structure. This allows both unsupported structures and spatial transformations to be designed into structurally sound shapes without a need for (intermediate) support materials.

Simple Shape Animation

We designed the system to allow 2.5D transformations to be recorded over time, providing the playback of simple shape animations in mid-air.

User Interface

We designed a simple 3x5 menu grid where the elevation of a voxel indicates the presence of a button, creating the rudiments for a 3D programmable matter-based menu. Since voxels do not convey text, we distinguished menu functions through colour and position only, currently limiting their functionality to the recording and playback of shape animations.

Manual Interactions

BitDrones are tangible and manipulated by hand. An important distinction with other work is that the drones were designed not to be perceived by the user as quadcopters, but rather as physical building blocks that are perfectly safe to touch. There are three styles of input: Uni-manual touch, bi-manual touch and gestural input. Uni-manual touch is used to select single voxels, while bi-manual touch is used to select rectangular arrays and to rotate or translate the entire grid. Figure 2 shows the gestural input recognized by the system. This "Point" gesture projects a 3D ray from the index finger that is used to calculate intersections with voxels. A click to select is performed by rotating the thumb from perpendicular to, to parallel with, the index finger (Figure 2). Manual interactions were based on Graphical User Interface (GUI) interaction techniques: Double tap is similar to selecting words with a double click. Lasso is akin to clicking and drawing a circle around pixels in a GUI. Ray casting was borrowed from Virtual Reality techniques. Bimanual input was inspired by touch screen interactions and used for translation, rotation, and selection, like using two-finger multitouch gestures in maps and photo editors.

Physical Object-Oriented Programming

To allow for shape transformations to be programmed by example, we created a smartphone app that allows topological behaviours to be programmed for sets of selected voxels (Figure 3). This allows voxels to autonomously respond to changes in their topological relationship with another voxel, for example, when it is dragged upwards using the hand. We applied sets of such behaviours such that more complex geometric transformations are scaffolded. In a prior version of this work, the smartphone was tracked and used as a 3D wand that facilitated interactions with remote voxels (see [7] for a description of the wand interactions). In this paper, we decided to focus on the use of manual interactions to emphasize the tangibility of the self-levitating voxels. We still chose to set the topological relationship through a smartphone, as it enables precise numerical control of the strength of the relationship as a percentage. Although this function could be performed by moving voxels directly as a TUI, we note that this interrupts the shape of the voxel grid during the act of programming.

INTERACTION TECHNIQUES

Aside from the operation of 3D menus, all of our interaction techniques were aimed at the efficient selection, translation and orientation of sets of voxels in the grid using uni-manual, bi-manual and gestural input.

Unimanual/Bimanual Select

Touching, then releasing a voxel allows the user to select single voxels (Figure 4a). Touching a first voxel, then simultaneously touching a second voxel selects the rectangular set of voxels that connects touch locations. Double-tapping a voxel selects all voxels in the grid. Double-tapping again deselects all voxels. Touch selection becomes more challenging in larger grids, which is why we developed *Lasso Select* and *Raycast Select* techniques as well.

Lasso Select
The lasso gesture is similar to that found in paint programs. It is performed by pointing the index finger beside the voxels via the "Point" gesture (Figure 2). The user then clicks in empty space using the "Thumb click" gesture, drawing an enclosed figure that ends at the original location and then raising the thumb to close the lasso. Voxels inside this enclosed figure are selected, with their LEDs illuminated for visual feedback. If the vector that extends from the finger happens to intersect with a voxel during a lasso, this voxel is included in the selection. Note that since *Lasso Select* is only performed on a 2D flat grid prior to setting a topological relationship and prior to creating a 3D structure. This means accuracy would not be affected by larger sets of smaller drones.

Raycast Select
Raycast selection of voxels (Figure 4b) is also performed using a "Point" gesture. Here, the user aims her index finger at a (number of) voxel(s), which are selected using a "Thumb click" gesture. Voxels that intersect the raycast by the index finger are selected (Figure 4b). This produces a simple ray casting technique for selecting multiple voxels simultaneously. While ray casting accuracy can be improved via conical shaped rays [15,33], we simply set the selection area for each voxel to a sphere slightly larger than each voxel. This selection area around each voxel has a radius of 15 cm from the voxel centroid. Note that raycast selection of multiple drones can be performed in any 2D or 3D shape.

Topological Program
The topological relationship between sets of voxels is programmed by setting a touch slider on the smartphone app. Voxels are programmed such that the vertical distance of a voxel follows the movement of a connected voxel by a factor (0-100%) that determines the ratio of distance to travel in the *z* dimension (up-down). From 0-50%, this ratio is exponentially related, at 50% it is linearly related, and between 50% and 100% it is inversely exponentially related, with 100% representing a direct connection between voxels.

Drag
Uni-manual touch on a selected set of voxels allows the user to translate (drag) the entire group in the *z* dimension of the grid. Other voxels will follow the set according to their topological program, and this can be used to create compound shape transformations. When dragging, topological programs are scaffolded such that the most rigid connections (100%) are honoured first. This allows users, for example, to create hinged structures that would otherwise not be possible. Entire shapes can be dragged by selecting all voxels using a lasso gesture, then dragging the selection.

Grid Transformations
Translating the entire display in *x,y,z* is accomplished by dragging any pair of (unselected) voxels bi-manually. Rotations of the entire grid are performed by rotating any pair of (unselected) voxels, bi-manually. We also explored using an embodied controller for children to easily interact with the movement and rotation of the grid, which is described in the Flying LEGO® Bricks application scenario.

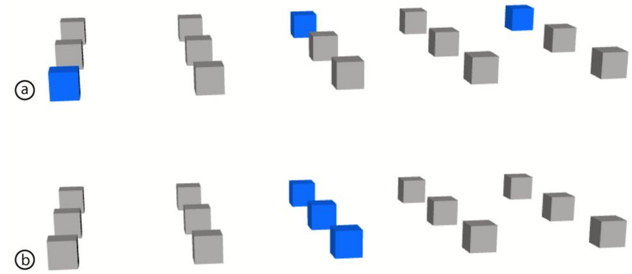

Figure 4. a) Individual voxel selection. b) Ray cast selection of a voxel row.

GridDrones User Interface
Simple tangible user interfaces can be created by selecting a row of voxels, dragging them out, and then selecting each individual voxel and assigning a function to them using the smartphone app. Functions are currently limited to the recording and playback of animations. The TUI is made active by pressing a menu button on the smartphone app.

IMPLEMENTATION
In the following section, we will discuss the implementation of the GridDrones system, including input and drone hardware, flight control software, and the operating system. An overview of the system architecture is shown in Figure 5.

Manual Input
Manual input is tracked via 5 small Vicon markers that are glued to each hand, 3 to the index finger and 2 to the thumb (see Figure 2). If the use of glue is undesirable, gloves with the same pattern can be used instead. The marker configuration is detected as a hand object by the Vicon motion capture software, which transmits marker coordinates to the BitDrones OS. BitDrones OS detects a "Point" gesture when the vectors projected through the thumb markers and index finger markers are at an angle larger than 60 degrees, and a "Thumb click" gesture when this angle is smaller than 10 degrees. Upon a "Point" gesture, BitDrones OS projects an outward vector through the 3 markers on the index fingers. Raycast selection is then implemented by calculating the intersection of this vector with voxels in the grid. When a "Thumb click" gesture is detected, BitDrones OS issues a click to select this intersection of voxels.

BitDrones Hardware
Each BitDrone is a stabilized, standalone quadcopter flight platform with external flight control and two-way telemetry. The propulsion system of a single BitDrone consists of four propellers, each driven by a coreless DC motor. These propellers have been offset and overlapped to maximize the active surface area within the drone footprint to increase overall lift in a smaller area. A custom flight control board with integrated motor controllers running a modified version

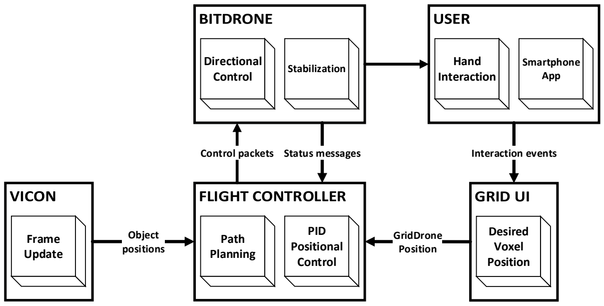

Figure 5. System diagram of GridDrones hardware and software connections.

of MultiWii 2.4.2 stabilizes the drone. It enacts movement commands through the proportional variation of thrust between all four motors. The flight controller is designed around an ATMEGA328P-MU microcontroller (MCU) and an MPU-6050 Inertial Measurement Unit (IMU) that serves to self-stabilize the drones. Self-stabilization is achieved through the use of on-board Proportional, Integral and Derivative (PID) feedback loops that use the orientation data from the IMU. A BitDrone's 3D printed and carbon fiber frame measures $10x10x10$ *cm* and weighs 11 *g*, with a total weight including battery, propellers and motors of 43 *g*. Each BitDrone is powered by a 3.7V, 300mAh lithium-polymer battery. The propellers are 50 *mm* in diameter and are safe for adults when touched. The motors are brushed, coreless 14 *KV* rated (14,000 revolutions per minute per volt). The maximum speed of each BitDrone voxel is limited to 1 *m/s* for precise position control.

Communications
An ESP-8266 wireless module on each drone acts as the intermediary between the flight controller and the off-board flight control system on the computer, over a WiFi connection. The off-board flight control software sends control packets containing the commands for multiple BitDrones. Each ESP-8266 parses the WiFi packets in order to extract commands associated with that drone. The command is then serially transmitted to the flight control board, which relays the thrust commands to each motor.

LED Module
Four RGB LEDs are networked over I2C into the flight control board as slave devices. This allows fast, precise colour control of the LEDs on the drone in order to give the BitDrone the functionality of a coloured voxel.

Structure
Each drone is contained within a carbon fiber and 3D-printed plastic structure. This structure can be made to outline the edges of a three-dimensional shape, in our case a cube. This allows for practical tangible manipulation and further emphasizes the traditional shape of a voxel to the user. This structure also serves as a propeller guard during tangible interactions by the user.

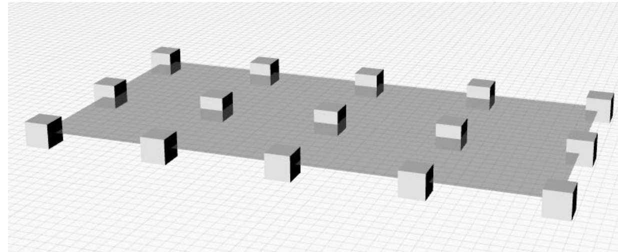

Figure 6. *1x3x5* coplanar arrangement of voxels.

Flight Control
To command a higher level of control and reduce latency, we wrote a new BitDrones OS 2.0 in C++, operating on a Mac Pro running Ubuntu 16.04. The flight control system utilizes reflective markers on each BitDrone, the positions of which are tracked and reported by a Vicon MX Motion Capture System running Tracker 3.1.0 software on a Mac Pro that is networked with BitDrones OS. All devices are on a shared network that is also connected to the individual BitDrones via Wi-Fi, which operates on the b/g/n 2.4GHz band. In order to reduce latency, it is required that the network be unsecured. In BitDrones OS, there is a software object representing each active BitDrone. This enables the flight control computer to run 3D positional PID control loops to determine the control commands to be sent to each BitDrone. This also allows interaction with the UI code, which requests voxels to be directed to positions in 3-space. The UI code runs on the same system and maintains the state of the system in 3-space, requesting and landing BitDrones as necessary. The latency between a Vicon position update and reception of a command by each BitDrone is less than 2 *ms*, while the software sends motor control updates every 10 *ms*.

3D Position and Velocity Control
Each BitDrone has 4 PID-based control loops that are updated at 100 *Hz*. Roll, Pitch and Thrust use dual-layered PIDs to control position and velocity. Yaw has a single layer PID to control angular orientation. These control loops have generic gain settings that govern all drones.

GridDrones Software
Next, we describe the software components that together make up the GridDrones subsystem (Figure 5).

Input
The flight controller takes in the position of the tracking markers on the hand, and presents it to the UI system as a transformation matrix based on the marker closest to the base of the index finger, a rotation matrix that represents the hand's 3-space rotation, and a list of BitDrones that the finger is pointed at. Pointing is detected by checking the distance from the drones to an infinite vector along the index finger's main axis. The system reports when a user's hand is nearing a BitDrone, and when it has made contact, by tracking marker coordinates on the hand and on the drones.

Grid Layout
The grid is a 1xnxn array of BitDrones. When created, it will dynamically add drones until there are none available. In the current system, 15 voxels are typically arranged in a 1x3x5

grid as shown in Figure 6, however, other variations of rectangular grids are supported. The grid can be completely represented by a position in 3-space, which represents its anchor point, a transformation matrix, and a 2D array of vertical offsets for each voxel. Using this data, any grid state can be recorded and played back.

Topological Groupings and Smartphone App

The grid allows for arbitrary groupings within itself. These groups are non-exclusive and can support an arbitrary number of voxels. Groups have a topological relation value, between 100% and 0%, which determines how they will move in relation to each other. When a voxel is moved, changes in its position are communicated to other voxels in the group in real-time at 100 *Hz*, which then follow this behavior according to their topological program. Topological groupings are programmed using a smartphone app (see Figure 3). This app features a slider that sends Open Sound Control (UDP) messages to BitDrones OS, allowing the user to specify the strength of the topological relationship. It also features a menu button that summons the TUI menu for recording animations (see the "GridDrones UI and Animations" section below).

Translation of Groups along the Z-Axis of the Grid

Translation of a BitDrone within the group is done with the hand. At the time of dragging, a distance to the hand is stored, and a projection vector is created extending from the end of the index finger. The difference between the projection vector and the voxel's current position represents an impulse, which is applied to the drone with a topological relationship of 100%. These changes are locked to the z-axis of the grid.

Grid Transformation

Translation and rotation of the entire grid is achieved by applying the hands' relative transformation matrix to the grid. A transformation is calculated from the average rotation and translation matrices of the drones that are being manually manipulated. The transformation matrix is calculated between the voxels' initial position/orientation and final position/orientation, instantly applied and stops when one of the hands is released.

3D CAD TECHNIQUES

GridDrones can be used to model a 2.5D surface through deformations of the grid of voxels. By selecting voxels and setting topological relationships between them, complex shapes with unsupported structures can be made with one gesture. There are various deformation techniques supported by the current system, all of which focus on surface geometry. Below, we discuss deformations of the two-dimensional surface organized by number of control points used, and the curvature acceleration function applied.

Point Controls. To create a pyramidal structure, the user selects all voxels, then programs them with a 50% topological relationship to nearest neighbor voxels. This produces a faceted surface function. The pyramidal structure is created by lifting a center voxel (see Figure 7a) with surrounding voxels displaying a linear slope relative to the center voxel. Any continuous mathematical function can be applied to describe the curvature relationship, known as *curvature acceleration*. For example, applying an exponential function would lead to a 3D hyperbolic structure instead (Figure 7b).

Line Controls. The first type of line control creates a faceted surface that resembles a hinge (also known as a G0 surface continuity [28]). To do this, the user first selects a line of 3 voxels in the middle of the structure using a ray casting "Point" gesture. Using the smartphone, the user then sets the topological relationship between these voxels to 100%. This will ensure all the voxels in the backbone move together as a group. Next, the user selects all voxels in the grid, setting the relationship to 50% using the smartphone app. This creates a triangular structure (see Figure 7c) when the backbone group is lifted upwards with the hand, as each row of voxels moves at only 50% of the distance of neighbouring voxels. The second type of line control allows any continuous mathematical function to be applied to describe the curvature acceleration. E.g., applying an exponential function would lead to a curved arch (Figure 7d). One special case is where a *catenary curve* is applied, which is useful for creating vaulted structures that mimic inverse gravitational paths. This allows for the creation of structurally sound models under gravity, as the curvature follows the minimal energy path [17]. Fig. 8 shows a GridDrones Catenary Arch.

Surface Controls. Complex surfaces are created by selecting all voxels, then setting a topological relationship between all nearest neighbour voxels. Any voxel can now be used as a control point to create surfaces with multiple undulations in both dimensions. When this relationship is linear (e.g., 50%), the surface deformations will appear faceted (Figure 7e). When the relationship is set to a Bezier curve, a complex NURBS surface is produced (Figure 7f).

GridDrones UI and Animations

GridDrones can also be used as a tangible user interface. To do so, users press the on-screen menu button on the smartphone app to switch to the GridDrones UI. This automatically flattens the grid, lifting two voxels as "buttons" up and towards the user, one for recording and one for playback (see Figure 9). Shape animations are created by manually touching and pressing down the record voxel. Each sends a "Button Click" event to the UI subsystem. GridDrones will subsequently record animations as users deform the grid in real time. Upon completion, users return to the UI mode by pressing the menu button on the smartphone app, which stops recording. Users can then play back the shape animation by pressing the green playback button. E.g., to produce a physical animation of a bird in flight, one could alternately lift and push down the backbone of a hinged structure, while moving the backbone through space. While this specific use of both smartphone GUI and TUI appears arbitrary, we included it to demonstrate a TUI menu is possible.

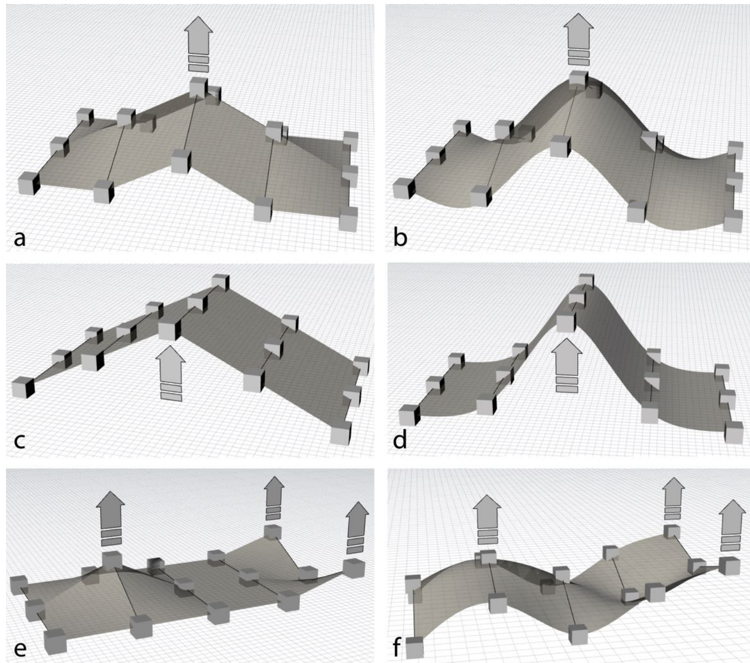

Figure 7. CAD techniques illustrating grid transformations. In all subfigures, arrows indicate user initiated voxel movement in in the *z* dimension. Top: Point controls in which one voxel is moved with *a)* linear relationship among voxels and *b)* curvilinear relationship. Middle: With a line of 3 voxels selected, translation in the *z* dimension resulting in *c)* faceted relationship and *d)* complex curvature. Bottom: Arbitrary voxels selected and translated in the *z* dimension resulting in *e)* faceted relationship and *f)* complex curvature.

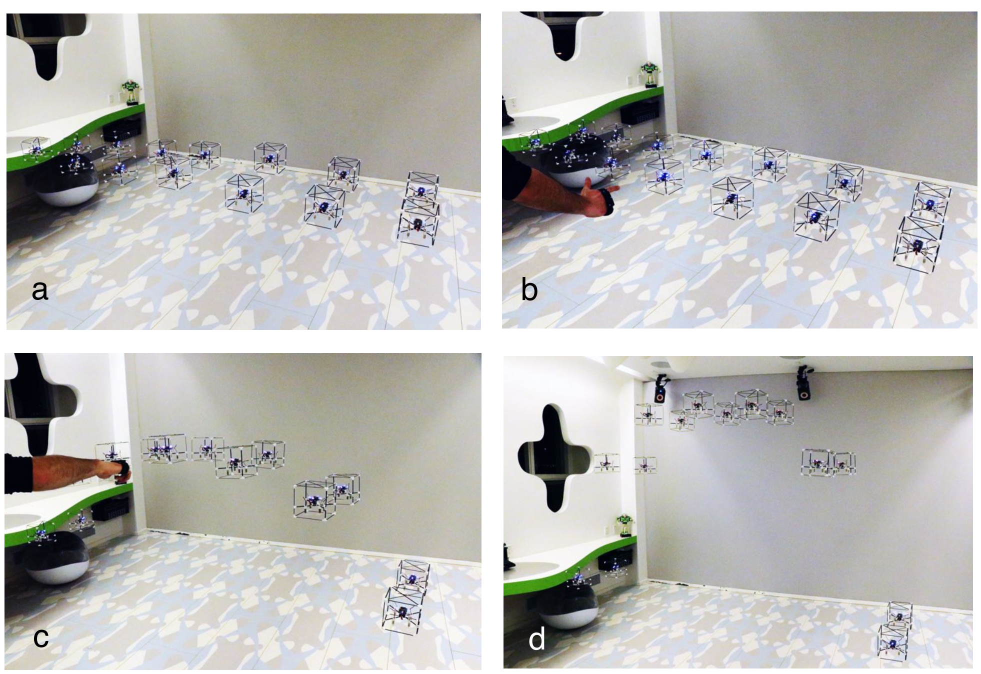

Figure 8. Creating a catenary archway with GridDrones. a) A flat grid of 2x7 drones forms the basis; b) The user uses a "Point" gesture to ray cast and select two key stones in the center of the grid, and sets a 100% rigid topological grouping with the smartphone app. He then selects all drones, programming a catenary curvature relationship between them using the smartphone; c) The user moves the keystones upward. An inverse gravitational curve begins to develop; d) The resulting archway would be structurally sound, if physically built.

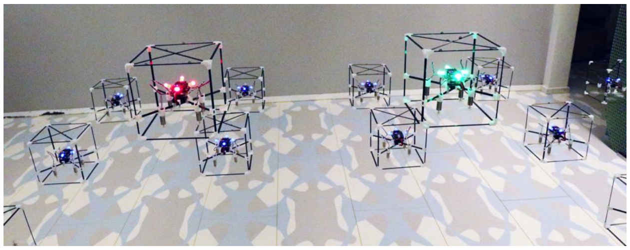

Figure 9. GridDrones user interface with two buttons: Record (red) and Playback (green).

APPLICATION SCENARIOS

The GridDrones platform can be used to support a wide range of physical 3D modeling activities. We highlight two: 1) Providing true to scale prototyping of physical structures for architectural design planning; and 2) Pedagogical tools for constructing and exploring physical movement.

Architectural Planning

Rapid physical prototyping at actual room scale is an area of increased interest in the community [1,47]. With GridDrones, iterations and refinements happen in real time, allowing for live mockups that do not require manual assembly and disassembly. Architects can use GridDrones as a tool to involve the client in a physical design process—where it is common for architects to use wires, cardboard and other artifacts to mock up a proposed building design. With GridDrones, this process expands to 3 dimensions, providing a more realistic approximation of volume and form that can be easily adjusted in life size. The architect can annotate designs using an ensemble of drones, which then feed back into the CAD drawings in real time. For illustration purposes, we describe a scenario of use within an architectural design context in which the design of an arched doorway for a building is developed in full scale using physical simulations of gravitational forces (Figure 8).

"John is an architect who in the past has relied on concept renderings and sketches for his structural designs. He misses the physicality that used to be provided by his cardboard mockups, which allowed him to tinker with the effect of structural forces on his design. Recently, John was tasked with designing a passageway through an existing wall. Instead of relying on traditional CAD tools, he decides to examine the use of a tangible mockup provided by the GridDrones system to model the proposed opening. After entering his office, which is outfitted with a Vicon tracking system in the ceiling, he fires up a grid of 2x7 drones which take flight and hover as a collective co-planar grid of "bricks", as John calls it (see Figure 8a). John likes using his hands to interact directly with the self-levitating "bricks". First, he uses a right-handed Point gesture to raycast through the center two drones in the grid. He snaps his right thumb to select these as keystones of the arch (Figure 8b). Next, he touches the smartphone in his left hand to set a 100% topological relationship between the 2 keystone "bricks", causing them to move together as a group. Then, using another Point gesture, he selects all "bricks" in the grid by lasso. John uses the app on the smartphone to select a parabolic relationship that approximates a catenary curve, which is a model for designing structurally sound arches. John manually drags one of the keystone "bricks". As he moves his hand upwards, the shape of an arch begins to emerge (Figure 8c). The keystone bricks produce a straight line, while the other "bricks" slope down in an accelerated curvature that mimics inverse gravitational forces. This produces an archway that would be structurally sound (Figure 8d). John enjoys the fact that GridDrones models remove the need for structural integrity under gravity during construction of the physical model. In the past, it was difficult for him to glue structures together, and he often required support material to do so. The final model, however, facilitates production of structures that could be manufactured without self-levitation. Finally, John invites his client to come explore the new archway design. He manually adjusts the size of the archway by dragging a cornerstone further up to accommodate the physical size of his client."

Flying LEGO® Bricks

We also developed a pedagogical tool for children interested in creating animated robotic structures. Rather than providing *full tangibility*, for child safety reasons, we employed a *distant embodiment* (see Fishkin [8] for a taxonomy). Figures 10 and 11 show how children can construct physical in-air animations using a simple embodied LEGO® controller and a set of 10 GridDrones serving as flying LEGO® bricks. The embodied controller consists of two wing-shaped LEGO® base plates connected by a flexible hinge (Fig. 10). The movement of each wing is tracked using an IMU and an Arduino, which sends orientation data over WiFi to BitDrones OS. Children decorate the controller using traditional LEGO® bricks. The location and colour of each brick is recorded through computer vision, then mimicked by a set of 10 GridDrones that take flight, each drone representing the location and colour of a physical LEGO® brick (Note that depending on the location of the bricks colours are not always mapped 100% accurately to the drone positions, see Fig. 11). By flapping and tilting the wings of the controller, children animate and play with the voxel flock in real time, simulating the flight of a butterfly with physical 3D voxels. Approximately 250 children tested this design at the LEGO® World Expo 2018 in Copenhagen. Without any exception, each child responded positively to the exhibit. Users quickly grasped the use of the embodied controller, which used direct mapping to the drone flock. The bending of the wings animates the "wings" of the flock, while rotating or tilting the controller resulted in rotation and tilt of the flock. The flock position was limited to remain within the volume controlled by the motion capture system. However, some tried to use the LEGO® bricks as buttons to move the drones forward and backwards. The Flying LEGO® bricks exhibit demonstrates the potential for future exploration of pedagogical and playful applications of our system.

Figure 10. Child playing with Flying LEGO® using the embodied controller at the LEGO® World Expo 2018.

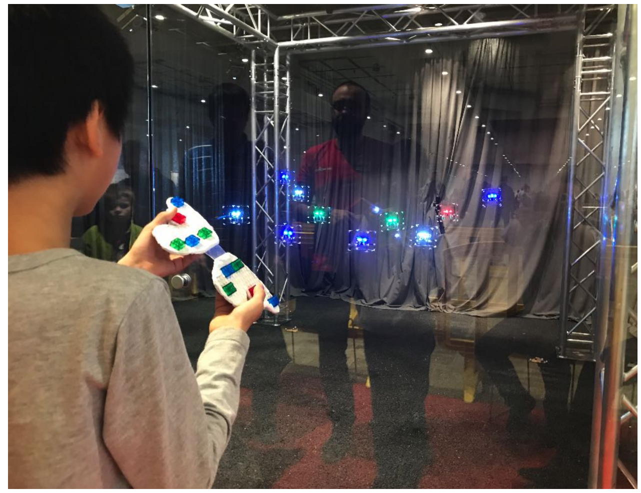

Figure 11. GridDrones interactive animation of the flight of a butterfly at the LEGO® World Expo 2018.

LIMITATIONS AND FUTURE WORK

Empirical evaluation of the GridDrones system is still somewhat beyond the scope of this work. This is because there are a number of limitations to the system, which we discuss below. A significant constraint is the power density of the batteries used to power the BitDrones. Each drone flies for approximately 7 minutes on a single charge, and it is unlikely that this time will increase without further advancement in energy storage technology. During the LEGO exhibit, we manually swapped batteries after every 3 users. An important future direction is to allow self-charging of drones. This would allow drones to be swapped in and out of the model without interruption. However, while the circuitry for self-charging is present in the current generation of drones, fast charge times cause drones to overheat, as heat cannot dissipate in the current design. The inclusion of heat sinks, or the use of the propellers for cooling while charging may address this issue in the near future. A pervasive goal of the BitDrones project is to decrease the size of drones to increase the fidelity of voxel structures. This is challenging at present because propeller efficiencies decrease disproportionally with miniaturization of drones. On the positive side, the system is sufficiently robust to double the number of drones without performance effects. The system could be scaled significantly with access to a larger indoor flying space in which turbulence can disperse, and with more motion capture cameras that increase coverage. Force feedback is currently limited by the strength of the motors, and mostly limited to the *z* dimension. Future versions may utilize stronger motors, or use propellers horizontally. The CAD techniques presented provide initial standard tools to enable users to build forms and set positional relationships within the grid, however, there are various other primitive functions common in CAD systems that have not yet been implemented. Some primitive transformations such as extrusion, revolve and sweep are not yet supported. Extrusion of a planar arrangement of voxels to make a 3D form would necessitate filling in voids with new voxels. Currently, a user can lasso a number of neighboring voxels and move them together on the *z*-axis to build a 3D form, however, the filling of open space is not yet supported. The lasso gesture, however, is limited to the selection of 2D grids of voxels and needs to be expanded to 3D selection. Revolving a line of voxels within the space is possible, yet to create a revolved 3D form, additional drones would need to be introduced into the space. We focused on surface deformations in this system, however, expanding the toolset to include other CAD editing techniques signals important future work.

CONCLUSIONS

We presented GridDrones, a self-levitating programmable matter platform that can be used for representing 2.5D voxel grid relief maps capable of rendering unsupported structures and 3D spatial transformations. GridDrones consists of 15 cube-shaped nanocopters that can be placed in a volumetric 1x3x5 mid-air grid. Grid deformations can be applied interactively to this voxel lattice by first selecting a set of voxels, then assigning a continuous topological relationship between voxel sets that determines how voxels move in relation to each other, then drawing out selected voxels from the lattice structure. Using this simple technique, users can create unsupported structures that can be translated and oriented freely in 3D.

REFERENCES

1. Harshit Agrawal, Udayan Umapathi, Robert Kovacs, Johannes Frohnhofen, Hsiang-Ting Chen, Stefanie Mueller, and Patrick Baudisch. 2015. Protopiper: Physically Sketching Room-Sized Objects at Actual Scale. In *Proc. UIST '15*, 427–436.
2. Javier Alonso-Mora, Andreas Breitenmoser, Martin Rufli, Roland Siegwart, and Paul Beardsley. 2011. Multi-Robot System for Artistic Pattern Formation. *ICRA'11,* 4512–4517.
3. Javier Alonso-Mora, Stefan Lohaus, Phillip Leemann, Roland Siegwart and Paul Beardsley. 2015. Gesture

Based Human-Multi-Robot Swarm Interaction and its Application to an Interactive Display. *ICRA'15,* 5948–5953.

4. Raffaello D'Andrea. 2013. Humans and the coming machine revolution. In *Proc. UIST'13*, 1-2.
5. Ars Electronica Futurelab. 2016. Drone 100. http://tinyurl.com/drone100. Retrieved April 2016.
6. Matthew Blackshaw, Anthony DeVincenzi, David Lakatos, Daniel Leithinger, and Hiroshi Ishii. 2011. Recompose: Direct and Gestural Interaction with an Actuated Surface. In *Proc. CHI EA '11,* 1237-1242.
7. Sean Braley, Calvin Rubens, Timothy R. Merritt, and Roel Vertegaal. 2018. GridDrones: A Self-Levitating Physical Voxel Lattice for 3D Surface Deformations. In *Ext. Abstracts CHI'18*. Paper D200, 4 pages.
8. Kenneth P. Fishkin. 2004. A taxonomy for and analysis of tangible interfaces. Personal Ubiquitous Computing. 8, 5, 347-358.
9. Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: Dynamic Physical Affordances and Constraints Through Shape and Object Actuation. In *Proc. UIST'13,* 417-426.
10. Sean Follmer, Daniel Leithinger, Alex Olwal, Nadia Cheng, and Hiroshi Ishii. 2012. Jamming user interfaces: programmable particle stiffness and sensing for malleable and shape-changing devices. In *Proc. UIST'12*, 519-528.
11. Antonio Gomes, Calvin Rubens, Sean Braley, and Roel Vertegaal. 2016. BitDrones: Towards Using 3D Nanocopter Displays as Interactive Self-Levitating Programmable Matter. In *Proc. CHI '16*, 770-780.
12. Seth C. Goldstein, and Todd C. Mowry. 2004. Claytronics: A Scalable Basis for Future Robots. In *RoboSphere'04*.
13. Seth C. Goldstein, Jason D. Campbell, and Todd C. Mowry. Programmable Matter. 2005. *IEEE Computer, 38, 6*, 99-101.
14. Mathieu Le Goc, Lawrence H. Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building Blocks for Swarm User Interfaces. In *Proc. UIST'16*, 97-109.
15. Tovi Grossman and Ravin Balakrishnan. 2006. The design and evaluation of selection techniques for 3D volumetric displays. In *Proc UIST'06*. ACM, New York, NY, USA, 3-12.
16. Hooman Hedayati, Michael Walker, and Daniel Szafir. 2018. Improving Collocated Robot Teleoperation with Augmented Reality. In *Proc HRI'18*, 78-86.
17. Jacques Heyman. 1966. The stone skeleton. International Journal of Solids and Structures 2, 2: 249–279.
18. Hiroshi Ishii, and Brygg Ullmer. 1997. Tangible Bits: Beyond Pixels. In *Proc. CHI'97*, 234–241.
19. Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. 2012. Radical atoms: Beyond tangible bits, toward transformable materials. Interactions 19, 1, 38-51.
20. Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. 2013. Evaluating the efficiency of physical visualizations. In *Proc. CHI'13*, 2593-2602.
21. Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper Hornbæk. 2015. Opportunities and Challenges for Data Physicalization. In *Proc. CHI '15,* 3227-3236.
22. Shunichi Kasahara, Ryuma Niiyama, Valentin Heun, and Hiroshi Ishii. 2013. exTouch: Spatially-aware embodied manipulation of actuated objects mediated by augmented reality. In *Proc TEI'13*, 223-228.
23. Brian Kirby, Jason Campbell, Burak Aksak, Padmanabhan Pillai, James Hoburg, Todd C. Mowry and Seth C. Goldstein. 2005. Catoms: Moving robots without moving parts. *AAAI Robotics Exhibition'05*, 1730-1731.
24. Haruhisa Kurokawa, Kohji Tomita, Akiya Kamimura, Shigeru Kokaji, Takashi Hasuo, and Satoshi Murata. 2008. Distributed Self-Reconfiguration of M-TRAN III Modular Robotic System. Int. J. Rob. Res. 27, 373-386.
25. Haruhisa Kurokawa, Satoshi Murata, Eiji Yoshida, Kohji Tomita, and Shigeru Kokaji. 1998. A 3-D Self-Reconfigurable Structure and Experiments," in Intelligent Robots and Systems, 860–865.
26. Jinha Lee, Rehmi Post, and Hiroshi Ishii. 2011. ZeroN: Mid-air tangible interaction enabled by computer controlled magnetic levitation. In *Proc. UIST'11*, 327-336.
27. Daniel Leithinger and Hiroshi Ishii. 2010. Relief: A Scalable Actuated Shape Display. In *Proc TEI'10*, 221-222.
28. Henry Packard Moreton. 1992. *Minimum Curvature Variation Curves, Networks, and Surfaces for Fair Free-Form Shape Design*. Ph.D. Dissertation. University of California at Berkeley, Berkeley, CA, USA.
29. Javier Alonso-Mora, Andreas Breitenmoser, Martin Rufli, Roland Siegwart, and Paul Beardsley. 2012. Image and animation display with multiple mobile robots. *Int. J. Rob. Res.* 31, 6, 753-773.
30. Hiroki Nozaki. 2014. Flying display: A movable display pairing projector and screen in the air. In *Proc. CHI EA'14*, 909-914.
31. Yoichi Ochiai, Takayuki Hoshi and Jun Rekimoto. 2014. Pixie dust: Graphics generated by levitated and

animated objects in computational acoustic-potential field. *ACM Trans. Graph.* 33, 4, 85.

32. Ivan Poupyrev, Tatsushi Nashida, Shigeaki Maruyama, Jun Rekimoto, and Yasufumi Yamaji. 2004. Lumen: Interactive visual and shape display for calm computing. In *Proc SIGGRAPH '04*, 17.
33. Gang Ren and Eamonn O'Neill. 2013. 3D selection with freehand gesture. Computers & Graphics 37, 3: 101–120.
34. John W. Romanishin, Kyle Gilpin and Daniela Rus. 2013. M-Blocks: Momentum-driven, Magnetic Modular Robots. In *Intelligent Robots and Systems'13, 4288–4295*
35. Michael Rubenstein, Alejandro Cornejo and Radhika Nagpal. 2014. Programmable self-assembly in a thousand-robot swarm. *Science 345(6198)*, 795-799.
36. Michael Rubenstein, Christian Ahler and Radhika Nagpal. 2012. Kilobot: A low cost scalable robot system for collective behaviors. *ICRA'12*, 3293-98.
37. Juergen Scheible, Achim Hoth, Julian Saal and Haifeng Su. 2013. Displaydrone: A flying robot based interactive display. In *Proc. PerDis'13*, 49-54.
38. Philipp Schoessler, Daniel Windham, Daniel Leithinger, Sean Follmer, and Hiroshi Ishii. 2015. Kinetic Blocks: Actuated Constructive Assembly for Interaction and Display. In *Proc. UIST '15*, 341-349.
39. Stefan Schneegass, Florian Alt, Jürgen Scheible, Albrecht Schmidt, and Haifeng Su. 2014. Midair displays: Exploring the concept of free-floating public displays. In *Proc. CHI EA'14,* 2035-2040.
40. Alexa F. Siu, Eric J. Gonzalez, Shenli Yuan, Jason B. Ginsberg, and Sean Follmer. 2018. shapeShift: 2D Spatial Manipulation and Self-Actuation of Tabletop Shape Displays for Tangible and Haptic Interaction. In *Proc CHI'18.* Paper 291, 13 pages.
41. Ivan Sutherland. 1965. The Ultimate Display. In *Proc. IFIP* 65, 2, 506-508.
42. Tomasso Toffoli and Norman Margolus. 1991. Programmable matter: Concepts and realization. *Physica D: Nonlinear Phenomena* 47, 1, 263-272.
43. Roel Vertegaal and Ivan Poupyrev. 2008. Organic User Interfaces. *Communications of the ACM* 51, 6, 26-30
44. Justin Werfel, Kirstin Petersen, and Radhika Nagpal. 2014. Designing collective behavior in a termite-inspired robot construction team. *Science 343*, 6172, 754–758.
45. Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. 2013. PneUI: pneumatically actuated soft composite materials for shape changing interfaces. In *Proc. UIST '13*, 13-22.
46. Toshiya Yui and Tomoko Hashida. 2016. Floatio: Floating Tangible User Interface Based on Animacy Perception. In *Proc. UIST '16 Adjunct*, 43-45.
47. Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, Alexander Popiak, Thomas Bläsius, and Patrick Baudisch. 2017. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In *Proc. CHI '17*, 2606-2616.