BitDrones: Towards Using 3D Nanocopter Displays as Interactive Self-Levitating Programmable Matter

Antonio Gomes, Calvin Rubens, Sean Braley and Roel Vertegaal
Human Media Lab, Queen's University,
Kingston, Ontario, K7L 3N6, Canada
{gomes, braley, roel}@cs.queensu.ca, j.rubens@queensu.ca

ABSTRACT
We present BitDrones, a toolbox for building interactive real reality 3D displays that use nano-quadcopters as self-levitating tangible building blocks. Our prototype is a first step towards interactive self-levitating programmable matter, in which the user interface is represented using Catomic structures. We discuss three types of BitDrones: PixelDrones, equipped with an RGB LED and a small OLED display; ShapeDrones, augmented with an acrylic mesh spun over a 3D printed frame in a larger geometric shape; and DisplayDrones, outfitted with a flexible thin-film 720p touchscreen. We present a number of unimanual and bimanual input techniques, including *touch*, *drag*, *throw* and *resize* of individual drones and compound models, as well as user interface elements such as self-levitating cone trees, 3D canvases and alert boxes. We describe application scenarios and depict future directions towards creating high-resolution self-levitating programmable matter.

Author Keywords
Organic User Interfaces; Tangible User Interfaces; Claytronics; Radical Atoms; Real Reality Interfaces; Programmable Matter.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
The thought that computer interfaces might someday *physically* embody user interactions with digital data has been around for a long time. In 1965, Sutherland envisioned the "Ultimate Display" as a room in which the computer controlled the existence of matter [42]. According to Toffoli and Margolus [45], such programmable matter would consist of small, parallel, cellular automata nodes capable of geometrically shaping themselves in 3D space to create any kind of material structure. Since then, there has been a significant amount of research conducted towards this goal under various monikers, such as Claytronics [13], Organic User Interfaces [48], and Radical Atoms [18]. All of these seek, at least in part, to utilize programmable matter for user interface purposes to allow for a full two-way synchronization of bits with atoms — something the first generation of Tangible User Interfaces was unable to achieve [17]. While there has been progress towards building hardware modules capable of various forms of self-actuation (known as Claytronic atoms or *Catoms*) [12,38], much of the work on programmable matter has been theoretical in nature. How to create a massively parallel system of Catoms capable of displaying two-way immersive physical user experiences is an enduring research goal. This would promise Augmented Reality (AR) systems physically integrated with real objects, creating Real Reality Interfaces (RRI) that render interactive digital experiences using real matter. The problem we address in this paper is that Catoms need to overcome gravity, typically via structural support by other Catoms, when building larger structures. While there has been some prior work in this area, the movement of individual Catoms in 3 dimensions is generally limited [31]. We propose to address the levitation problem using nano-quadcopters. While there have been explorations of swarms of quadcopters for visualization applications [4,26], there has been little work on interactive, real-time user interface applications of 3D drone displays.

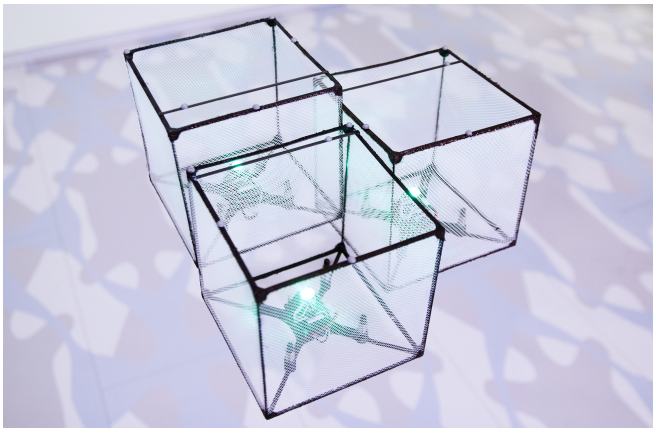


Figure 1. BitDrones hovering in a tight formation.

Contribution

In this paper, we present BitDrones, an interactive, glasses-free, 3D tangible display that uses nano-quadcopters as self-levitating Catoms that serve as real reality voxels (see Figure 1). Our prototype is a first step towards interactive self-levitating tangible user interfaces with multiple building blocks that are capable of physically representing 3D data on the fly. We discuss three types of BitDrones, each representing Catoms of distinct resolutions: 1) *PixelDrones*, which are equipped with an RGB LED and an OLED display. PixelDrones represent a single illuminated voxel or, optionally, a text label; 2) *ShapeDrones*, which are BitDrones augmented with a lightweight acrylic mesh spun over a 3D printed frame in a larger geometric shape, such as a cube or a sphere. ShapeDrones only have one RGB LED, used to illuminate a lightweight mesh diffuser that covers the 3D frame. ShapeDrones allow users to build higher granularity 3D models without requiring drones to stack on top of each other in mid-flight; 3) *DisplayDrones*, which carry a flexible touchscreen display, molded in an arc, with an Android 5.1 smartphone board. DisplayDrones are able to render high-resolution images, such as contextual menus, pictures and videos. We present both unimanual and bimanual input techniques, including touching, dragging, throwing and resizing of BitDrones and compound models, as well as user interface elements, including flying cone trees, video displays and alert boxes. We conclude with application scenarios in 3D editing, molecular modeling, real reality information visualization and telepresence. While the BitDrones system at present only allows for sparse models of up to 12 Catoms, unlike VR or AR, it promises to fully preserve natural depth and tactile-kinesthetic cues without requiring headsets or simulated haptics.

BACKGROUND

We will first discuss prior work in the areas of Tangible User Interfaces, Augmented and Mixed Reality Interfaces, Actuated Shape Displays, Programmable Matter and Self-Levitating Interfaces, as they relate to the design of our system. Note that we purposefully omitted a discussion of VR interfaces as these systems are designed to be fully immersive, i.e., do not allow the user to interact with the real world. We also discuss ongoing work in the emerging field of Human-Drone Interaction.

Tangible User Interfaces

Tangible User Interfaces (TUIs) leverage our ability to sense and manipulate the material world by providing physical form to digital information, facilitating direct engagement with the physical world. TUIs were introduced by Ishii and Ullmer [17] as an effort to augment the real world by coupling digital information to everyday physical objects and environments. Over the years, tangible interfaces have been extensively explored in electronic music platforms [20], urban planning [34], constructive assembly [35], as well as graspable interfaces [9]. However, TUIs have limited ability to display changes applied to the physical objects by software processes or remote users. To overcome the limitations of tangibles, Radical Atoms [17] described a hypothetical physical material that could be coupled with an underlying digital model, allowing for human interactions with computationally transformable physical materials.

Augmented Reality Interfaces

There is a large body of work aimed at interacting with virtual elements in 3D environments, which we cannot possibly do justice in this review. Robertson et al. [37] designed cone trees as a means of visualizing file hierarchies in a 3D user interface. In cone trees, folders and files are arranged in a circular manner, with each level of the hierarchy containing a horizontal "wheel" of folder or file nodes. Robertson et al. also investigated other forms of navigating 3D Graphical User Interfaces (GUIs), but was limited to a 2D display. By contrast, Augmented Reality systems use partially transparent displays to superimpose graphics onto real objects or environments [15,24]. They can be combined with passive tangible input in the real world, for example, using AR Toolkit markers [16]. HoloLens [15] introduced a semi-transparent head-mounted display that maps the space around the user using a fully integrated depth camera, allowing the system to simultaneously model its environment as well as sense marker-less gestural input. Shader Lamps [36] introduced a method for rendering AR *directly* onto physical objects, thus merging the display with the real world. Researchers have deployed such 3D projection-mapping techniques to create interactive 3D physical objects [1,14] and to simulate textures and material properties of designs [14,19]. 3D projection mapping typically requires expensive capturing of the 3D topography of the projection space. LightSpace [47] showed how cheap arrays of Kinect depth cameras could be combined with multiple projectors to augment any surface in an entire room with rich interactive graphics.

Mixed Reality Interfaces

Mixed Reality [27,28] is a special class of augmented and virtual reality technologies for creating environments wherein real and virtual world objects are presented together in a single display. Milgram and Colquhoun Jr. [28] formulated a global taxonomy to describe how the virtual and real aspects of Mixed Reality environments are combined and experienced, with the ultimate objective of clarifying conceptual boundaries existing among noted research. Benford et al. [5] experimented with projecting a virtual poet into a real theatre as part of a live performance. They suggested that their "early experience of social interaction between physical and synthetic spaces implies the need for a more systematic approach to joining them together". According to them, real-world interactions combine the local with the physical, VR combines the remote with the synthetic, while AR combines the local with the synthetic. They proposed *mixed reality* as a new form of shared space that integrates the local with the remote and the physical with the synthetic. An example mixed reality system, MirageTable [6], allowed users to mix real and virtually projected 3D objects on a curved tabletop. It used a

depth camera to map objects and users around the tabletop, then project these back onto a remote table such that remote and local users could interact with one another. This allowed users to virtually share physical objects in a way that was fully spatially congruous. In this paper, we suggest working towards merging the local, remote, physical and synthetic in such a way that they become one and the same, at least when it comes to the sharing of objects.

Actuated Shape Displays

Actuated shape displays provide novel opportunities for experiencing and interacting with digital content in the physical world, by dimensioning the actual displays through actuation. Lumen [33] explored the design of an actuated pixelated relief display. Lumen used light guides actuated through shape memory alloys to display low-resolution images with some *z* dimension. Leithinger and Ishii's Relief system [25] featured an actuated tabletop display that rendered interactive 3D shapes with a wider range of *z* actuation. Their system used a series of motorized pins actuating a fabric display that was top projected with images. More recently, inFORM [11] explored the use of high-resolution dynamic relief to display digital information. inFORM's shape display provided for variable stiffness rendering and real-time user input through direct touch and tangible interaction. Although shape displays can provide a bi-directional interface between bits and matter, including visual, haptic and force feedback experiences, they are limited in the kind of shapes they can render. E.g., the aforementioned explorations cannot render shapes with holes inside their structures.

Programmable Matter

Recently, there has been a significant research effort towards developing a new generation of computer interface capable of displaying physical 3D structures via programmed movement of large quantities of self-assembling nanorobots [38,46]. Goldstein et al. [12] alluded to the creation of physical artifacts using a collection of such Catoms [13], which would eventually be able to mimic an analog object's shape, movement, visual appearance, and tactile qualities. Alonso-Mora et al. [2] presented a display in which each pixel is a mobile robot of controllable color. Their system took images or animations as input and produced a multi-robot display as output. Similarly, Rubenstein et al. proposed a thousand-robot swarm capable of self-assembling into larger, potentially interactive, shapes [39]. Practical attempts at creating claytronic atoms, however, have proven difficult. While the above systems successfully demonstrated programmable self-assembly of complex shapes, thus mitigating some of the limitations associated with shape displays, they are limited to rendering shapes in two dimensions.

Self-Levitating Interfaces

An important problem in the creation of 3D programmable matter is that the structural integrity of the object needs to be preserved while it is changing shape [39]. This can be difficult to achieve with robotic motes as they have to rest on top of one another to overcome gravity. To address this issue, researchers have investigated the use of magnetic [23] and ultrasonic self-levitation [31]. However, these methods pose distinct limitations to the independent motion of multiple Catoms. Karagozler et al. [21] proposed the use of robotic helium balloons, or Giant Helium Catoms. The problem with these structures is that they are inherently difficult to miniaturize.

Drones and User Interaction

While there is a body of work that investigated non-interactive swarms of quadcopters acting together as a scalable production means [10,26], flying displays [30,40,41], augmented sports [29] and feedback mechanisms [42], further research is necessary to ascertain what are the most effective methods to facilitate human-drone interaction [4,8]. Pfeil et al. [32] explored 3D spatial interaction metaphors for interacting with drones. Their approach aimed at generalizing interaction metaphors that could potentially be applied to future user interfaces. This concept was further explored by Cauchard et al. [8], who conducted a Wizard-of-Oz elicitation study to evaluate how users naturally interact with drones. Their results suggested that users interact with drones in a similar way they would interact with a person or a pet. Nitta et al.'s [29] Hoverball demonstrated a flying controllable ball to integrate imaginary dynamics into ball sports. The authors investigated new ball-playing vocabularies, such as hovering, anti-gravity, proximity, or remote manipulation, as a method to extend the way people experience ball-based sports. Researchers have also explored the concept of free-floating midair displays via flying robots equipped with projectors [30,40] and high-resolution displays [41]. In contrast to traditional static displays, free-floating displays have the potential to change their position to appear at any given point in space and approach the user to communicate information. While these explorations facilitated human-drone interaction, they lacked support for multimodal I/O between the drone and the user. Additionally, they were limited to a single drone that primarily acted as a visual information display. Instead, we investigate how users can interact with swarms of nano-quadcopters via direct touch, unimanual and bimanual input techniques and gestural interactions. Our goal is to create the first system that uses multiple drones that act together to create interactive tangible interfaces, and generate real reality 3D displays using nano-quadcopters as self-levitating tangible building blocks.

DESIGN RATIONALE

To inform the design of BitDrones, we considered the following design parameters:

Modularity and Granularity

We designed the system in a modular fashion, like Lego™, allowing users to build structures out of individual, self-levitating building blocks. Structures can have varying granularity, i.e., consisting of a few to a dozen drones, depending on the complexity of the compound object. We compensated for a lack of density in building blocks by

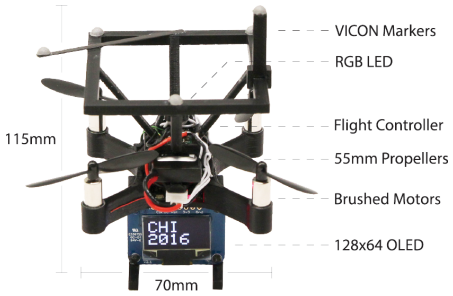
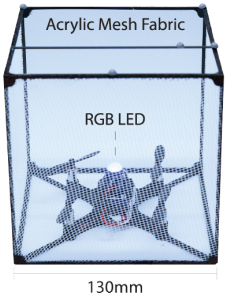
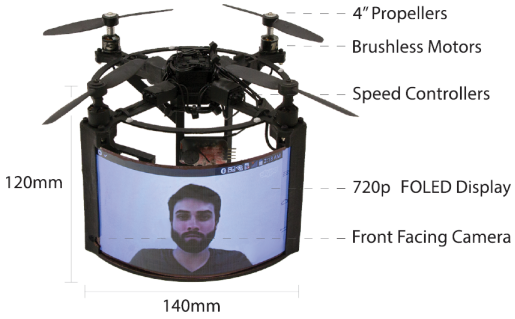


Figure 2.1. PixelDrone with RGB LED and OLED Display.



2.2. ShapeDrone with RGB LED and Cube-shaped Diffuser.



2.3. DisplayDrone with curved 720p FOLED Android display.

allowing the use of higher resolution, sub-voxel imaging via drone-mounted displays.

Drone-Mounted Displays

The original vision for single voxel *Catoms* would require thousands of microscopic drones flying in very close proximity. While this may be feasible in the future, we decided to use sub-voxel imaging techniques to enhance imaging resolution of a sparser drone cloud. We designed 3 types of drones, each with a different sub-voxel imaging resolution, and different form factors: PixelDrones that display a single color voxel; 2) ShapeDrones that display larger shapes and 3) DisplayDrones that carry a high resolution display.

Haptics: Tangible vs. Gestural Interaction

While BitDrones have the ability to follow gestures as well as user's body movements within a tracked space, an important design criterion was for each drone to respond to direct touch. This allows users to directly interact with the real reality interface, without any mapping or prior knowledge of predefined gestural interactions. BitDrones are simply grasped and moved around freespace at will. Groups of BitDrones are manipulated through the use of PixelDrones that serve as handles, allowing for intuitive bimanual 3D translation, rotation and resizing operations. The ability to directly touch and move drones provides haptic experiences not available in VR or AR systems. BitDrones do not require simulation of tactile kinesthetic feedback, as this is naturally provided.

Multiple Form Factors

Due to their sparseness, we designed BitDrones in a multitude of form factors, allowing users to efficiently build larger objects. While BitDrones can carry any type of 3D printed structure, we limited ourselves to constructing basic 3D geometries using very lightweight, wire mesh materials.

Physically Immersive User Interface Elements

The design of our graphical user interface elements was inspired by prior work in the 3D interface space, with the distinction that users *physically* immerse themselves in the interface, without the aid of virtual reality headsets. Borrowing from Robertson et al. [37], e.g., users surround themselves with physical, circular displays showing filenames or content that rotate automatically to stay visible to users at all times.

Physics Engine

We designed the BitDrones system with a built-in physics engine that allows drones to act under the constraint of real-world physics. Single drones, as well as groups of drones, can simulate physical momentum that corresponds to objects moving under friction, to allow greater predictability of their movements.

Other Physical Constraints

Nanodrones are generally limited in payload. The use of specific, lightweight hardware was required to achieve our design goals. For this reason, we used thin-film flexible OLED (FOLED) displays in the design of DisplayDrones. Another physical constraint is the presence of turbulence, mainly caused by the rotors of the drones. This means BitDrones cannot fly directly above or below one another, and need some clearance in order to achieve stable flight. In general, the smaller the drones, the tighter formations they can form.

IMPLEMENTATION

In BitDrones, each nano-quadcopter represents a Catom that can hover anywhere inside a volume of 4m *x* 4m *x* 3m in size. Our system currently supports up to 12 drones in simultaneous flight. BitDrones can hover 15cm from one another and their individual accuracy is 5cm per axis. As an exception, however, when placed against one another, they can fly as a group in tight formations (see Fig. 1). Users can walk around the interaction volume and interact with drones by touching and dragging them. BitDrones can be used for input, output, or for both at the same time. Simple atomic information can be displayed by a few drones, while more complex 3D data displays can be constructed using up to a dozen drones, providing basic elements for a voxel-based 3D modeling system capable of representing sparse 3D graphics in real reality.

Figure 3. Spherical, Cylindrical and Rhombicuboctahedral ShapeDrones.

Hardware

We built three different types of BitDrones, each capable of displaying information at different resolutions: PixelDrones, ShapeDrones and DisplayDrones (Figure 2).

PixelDrone Hardware

Figure 2.1 shows PixelDrone, a nano-quadcopter of our own design. Each PixelDrone is equipped with 4 brushed motors, a Micro MWC flight controller board with an IMU, and an Xbee Wi-Fi radio. PixelDrones run a customized version of MultiWii 2.3.3 as a flight control operating system. An RGB LED provides a colored voxel to the user. PixelDrones are optionally equipped with an Adafruit 128*x*64 OLED display that allows them to display text labels and simple graphics. This display is driven by an ATmega 328p chip which can be programmed wirelessly through the flight controller board. PixelDrones carry a 300mAh battery, providing them with approximately 7 min of flight time. The flight time is reduced to 3 min if an OLED display is included.

ShapeDrone Hardware

ShapeDrones consist of the same hardware as BitDrones without the OLED display. Instead, they carry a lightweight acrylic mesh fabric spun over a 3D printed frame mounted around their propellers (see Figure 2.2). The mesh acts as a diffuser for the RGB LED, allowing the entire shape to be lit up in color to act as a larger pre-shaped voxel. This frame can be printed in any shape. Similar to pre-shaped Lego™ bricks, ShapeDrones allow for the use of larger, more complex shapes that would normally be difficult to produce using a sparsely spaced set of drones. All shapes have holes in the top and bottom for airflow. We printed spherical and cubed shapes, but other shapes are possible (see Figure 3).

DisplayDrone Hardware

Figure 2.3 shows a DisplayDrone, a quadcopter 17.5 cm in size diagonally and capable of lifting heavier loads. DisplayDrone uses the same flight controller and Xbee Wi-Fi radio. MultiWii firmware was modified to drive brushless motor controllers. Each brushless motor is outfitted with a 6A electronic speed controller. DisplayDrones carry a lightweight 1280 *x* 720 LG Display FOLED touchscreen display, with a processor board running Android 5.1 mounted below the propellers. DisplayDrones allow for a high resolution, flying Graphical User Interface (GUI) that can run any Android app. Embedded light sensors measure ambient light and dynamically adjust screen brightness. The display is mounted in a 3D printed frame, curved 90º around

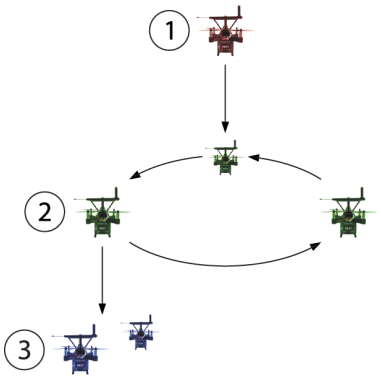


Figure 4.1. Cone tree node; 4.2. Browsing of submenu (rotational wheel); 4.3. Expanded 2nd submenu.

the drone, and 15 cm in size diagonally. The Android board runs the Android 5.1 operating system, rendering this drone with full smartphone functionality. A small 2.1MP, 30 fps camera is mounted on the front of the DisplayDrone, to the bottom left of the display, allowing the drone to capture real-time video. Each drone is also outfitted with a small stereo speaker for sound. DisplayDrones carry a 1000 mAh battery, providing them with approximately 7 minutes of flight time.

Flight Control

Each BitDrone has a set of reflective markers in a unique configuration, allowing its *x,y,z* position and orientation to be individually tracked by a Vicon MX Motion Capture System running Tracker 3.0 software [49]. A Mac Pro connected to the Vicon system over Ethernet provides location-based flight control information to a second Mac Pro running Windows 8. The BitDrones OS is a C# application running on this computer. It maintains each drone's state in a C# object. Objects representing all user interface elements and application functionality orchestrate the overall BitDrones system's behavior. BitDrones OS also tracks user input to the drones via the Vicon, processing input events as outlined below. For each drone, the BitDrones OS wirelessly sends motor signals to the MultiWii software over Wi-Fi to direct it to a particular location. A set of PID loops controls each drone's movements towards end positions based on user input and interface or application behaviors.

PID Loops

Each drone object in BitDrones OS has its own set of Proportional, Integral and Differential (PID) control loops [44]. These equations model each drone's current location, destination and velocity errors, calculating multipliers that affect the drone's thrust along the *x,y* and *z* axis and about the *z* axis. Each PID loop is tuned to the weight and balance of each individual class of drone. PID loops govern a drone's behavior as it flies to a destination. Once a drone has reached its destination, the MultiWii flight control firmware embedded on the drone controls a hover-in-place. This firmware has its own set of PID loops, one per motor, which controls the stability of the drone. The firmware does not rely on the Vicon system for this purpose, but on the IMU hardware in each drone.

User Input

The Vicon also tracks small reflective markers on the user's hands, allowing for detection of touch and move events by BitDrone OS. By estimating the relative positions between markers on the user's hands and the markers on the drones, the system detects interaction primitives such as touching or dragging of individual drones across the Vicon space. When a user picks up and moves a drone, this allows the BitDrone OS to respond by telling a drone to hover in the new destination. Input primitives can be combined for more complex interactions with 3D compound objects, as outlined below.

INTERACTION TECHNIQUES

Groups of drones were used to construct a number of user interface elements borrowed from 3D user interfaces:

User Interface Elements

3D Conetree Menus, Folders and Files

Figure 4 shows 6 PixelDrones flying in a group that forms a menu structure shaped as a cone tree. Our cone trees begin with a top PixelDrone node representing the root folder (see Figure 4.1). When touched, this expands into a submenu of PixelDrones flying underneath it in a circular arrangement (Figure 4.2). The name of the folder or file is displayed on the corresponding PixelDrone's display. Submenus can be browsed by gentle unimanual horizontal throws of a single drone inside the submenu. This causes the entire submenu wheel to rotate with simulated momentum and friction (Figure 4.2). When a folder in a submenu is touched, a third level of submenus can be displayed (Figure 4.3). Other drones automatically make space for this layer, while drones are automatically recruited to display its contents. When a file node is touched, drones are recruited to show the object inside the file. Currently, file contents are limited to 3D models. Due to the limited number of drones available, files are browsed in place, reusing drones from the cone tree menu to display the file. When a file is open, one BitDrone representing the menu continues to hover above the 3D contents of the file. Users can save the file and return to the menu by touching this drone.

3D Canvas and Handles

When a 3D model file is opened, its 3D boundaries are indicated by 4 PixelDrones, which act as handles to the compound object inside this 3D canvas. When drones are placed inside the boundaries of this 3D canvas, they become part of a compound object, maintaining their relative location within the 3D canvas even when the 3D canvas is moved. Handles also provide a convenient way to resize, move or rotate an entire 3D Canvas using bimanual input techniques.

3D Compound Geometries

Compound objects typically consist of ShapeDrones or PixelDrones acting as sparse voxels representing some physical 3D data structure. The advantage of using a ShapeDrone is that larger, pre-shaped 3D content can be added to the canvas without requiring PixelDrones to render that shape. While this requires a priori knowledge of the rendered shape, it partly addresses the concern that it would require many PixelDrones to create continuous large shapes. Resulting designs are automatically communicated to a Unity 3D engine, from where they can be texture mapped and exported, or 3D printed.

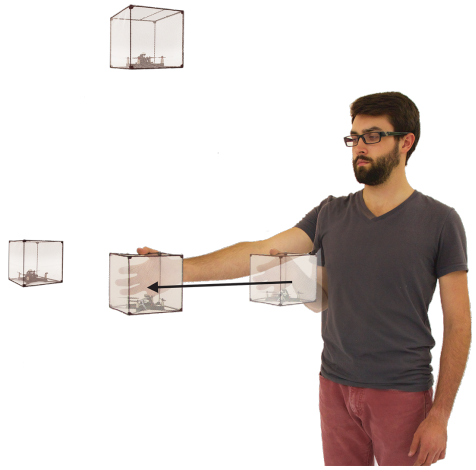


Figure 5. User positioning ShapeDrones within a compound object and releasing them in mid-air.

2D Contextual Flying Menus, Alerts and Video Windows

DisplayDrones are used to represent contextual menus capable of tracking the user, staying within arms length distance as they walk around the Vicon space. Contextual menus can be used to perform menu commands, like copy and paste, on selected drones. DisplayDrones can also be used as a means of alerting the user through a flying notification dialog box, to show images, or to represent remote users via a video window.

Unimanual Input Techniques

We designed a number of unimanual, bimanual and gestural input techniques that implement tangible, direct touch interactions with single drones or compound 3D objects.

Touch

Given the presence of a Vicon system, we sense touch by augmenting the user's hands with small Vicon markers. Touch is triggered when the distance of the user's fingers to a drone is smaller than 0.5 cm. Touch operates similarly to a click in a GUI. E.g., if a drone represents a folder in a menu, touching the drone reveals its content. Touching the drone again closes the content. A touch on a drone in a compound object selects the drone. A touch on two different drones in a compound object selects the entire object. DisplayDrones also sense capacitive touch within their display.

Drag

Drones can be dragged by holding them, then physically moving them to another location within the Vicon space. Upon detecting a drag, drones will hover at their new position. When drones are dragged to within the boundaries of a 3D canvas, they automatically become part of a compound object, maintaining relative position within that object.

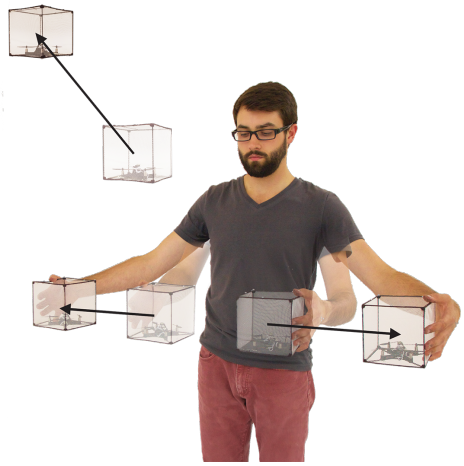


Figure 6. User resizing a compound object using a bimanual pinch gesture, by moving 2 of the total 3 drones.

Throw

Drones can also be moved to out-of-reach destinations by holding them and physically throwing them in the direction of that destination. Upon detecting a throw event, drones will fly in the direction of the throw with simulated momentum and friction that is based on their acceleration when released by the user. The simulated physics allows users to estimate the throw force required to allow a drone to move to the desired destination.

Pick Up/Remove

When not in use, drones are placed on a table in their home location. They can be activated either programmatically when summoned to display content, or by the user picking them up and then releasing them in mid-air (see Figure 5). Upon detecting a pick-up event, drones activate their propellers, hovering in position when released. Conversely, drones can be removed from a model by moving or tossing them towards the home location. Once they arrive within threshold distance of their home location, drones automatically land and spin down.

Bimanual Input Techniques

Drag Group

Holding two drones with separate hands inside a compound object and moving the hands allows users to drag an entire compound object. 3D Canvases can be dragged by moving two of their handles.

Resize Group

Compound objects can be resized by holding two drones inside the compound object, then performing a bimanual pinch gesture (Figure 6). Similarly, 3D Canvases can be resized by holding two handles– one with each hand – and performing a bimanual pinch gesture.

Rotate Group

Compound objects can be rotated by holding two drones inside a compound object and moving the hands in a joint arc, thus performing a bimanual rotation. Other drones in the model automatically adjust their relative position to the orientation of the first two drones. 3D Canvases can be rotated by holding two handles – one with each hand – and performing a bimanual rotation gesture.

While we discussed dragging, resizing and rotation of entire compound objects, our system also supports translation of both compound objects and single elements.

Gestural Input Techniques

Follow the User

Any drone has the capacity to follow the user's movement within the Vicon space. Typically, however, drones stay in their location to provide the user with spatial coherence. DisplayDrones offer a follow-me function that allows them to stay within arm's reach as the user moves around the space. This allows, e.g., contextual menu palettes to hover around the user. Although we could have implemented other gestural input techniques, such as summoning a drone from a distance [7,32], we wanted to emphasize tangibility, and focus on the use of direct touch input in this paper.

Remote Interactions

If two users are working on the same object using the BitDrones system in two different locations, drone actions are automatically synchronized between sites. That is, when a user edits the location of a BitDrone inside a compound object, its movement is automatically replicated in the corresponding remote canvas. This allows users to remotely collaborate on 3D designs with fully synchronized remote Catoms.

APPLICATION SCENARIOS

While the BitDrones OS can be used for a variety of applications, we developed a number of specific scenarios of use that demonstrate the versatility of the system. Since at present, BitDrones OS can only deploy up to 12 drones at a time, most applications were designed with sparse 3D models in mind.

3D Design

Figure 7 shows how the BitDrones OS can be used as a toolbox for 3D real reality modeling applications. In this example, users are building an architectural model out of cubed ShapeDrones. ShapeDrones can be added, removed, and moved around the model freely. PixelDrones form a bounding box that defines a 3D canvas. Users can resize the model within the canvas by holding two ShapeDrones and performing a bimanual pinch gesture. They can rotate the model by holding two ShapeDrones and moving them in an arc. We chose to model the "Cube house project" by Zafari architects [3] as it represents a sparse architecture in which drones need not stack or fly in very close proximity. More traditional architectural models requiring stacking of building blocks are more challenging to represent with the current system. The 3D design application is synchronized with a 3D visualization in Unity 3D, which allows users to render the model with texture maps and shading, as well as 3D print the model after real reality edits are completed (see Figure 8).

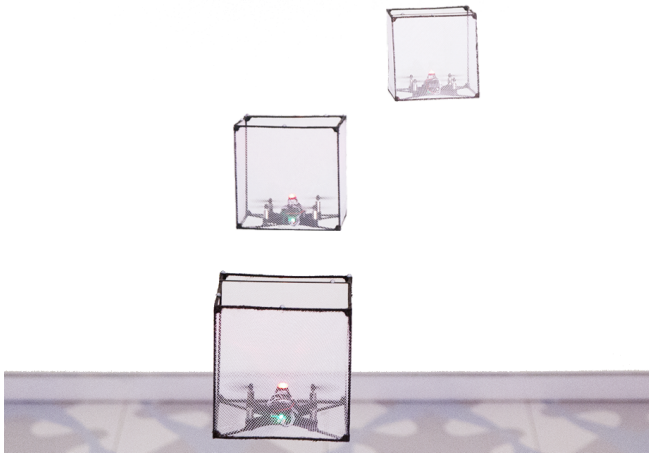


Figure 7. 3D editor with architectural model composed out of 3 cubed ShapeDrones.

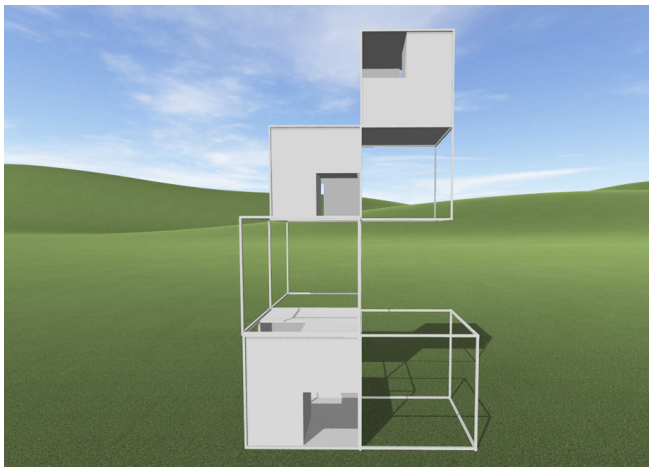


Figure 8. Unity 3D rendering of resulting "Cube house" model [3].

Molecular Modeling

BitDrones OS allows the exploration of bonds between pairings of atoms in 3D. BitDrones can represent molecular structures in 3D in mid-air using PixelDrones or ShapeDrones, allowing users to interact with these structures in real reality. Spherical ShapeDrones are particularly useful to simulate atoms. Multiple ShapeDrones can be placed in a spatial arrangement that simulates a more complex molecule. To simulate chemical reactions, a user can manipulate individual atoms by dragging drones in or out of a molecular bond. These chemical reactions can be recorded for automated use, for example, to instruct chemistry students about molecular bonds. The drones can also simulate repulsion and attraction forces present in molecular bonding.

Interactive Real Reality InfoVis

The BitDrones system can also be used to represent points in interactive data visualizations. For instance, the position of drones can be determined by a mathematical expression, creating a physical representation of that expression. Manipulating one of the drones modifies some parameters of the expression, such as the curvature of a parabolic function. Other drones adjust their position accordingly, preserving the spatial relations as defined by the mathematical expression. PixelDrones can be used to represent live cloud-based information, such as stock market data or twitter feeds. The location of drones would, in this example, be tied to some parameter of the data. The value of stock or frequency of a retweet can, e.g., be represented by the *y* coordinate of a PixelDrone.

Figure 9. User inspecting a remote facility using a DisplayDrone with telepresence functionality.

Remote TelePresence

We also developed a remote telepresence application in which users embody themselves via a remote DisplayDrone that displays a Skype videoconference (Figure 9). When reciprocated, this gives users the ability to fly around a model and interact with remote participants at eye level without requiring a robotic structure. The front facing camera on the DisplayDrone shows parallax-free images of the remote user on the local DisplayDrone, and vice versa. Multiple DisplayDrones can be used for multiparty conferences, serving as a self-levitating Hydra system that preserves eye contact and head orientation cues [7].

5D Printing

When polyhedral ShapeDrones are landed on top of one another, they can create complex compound structures that resemble 3D prints. Here, ShapeDrones are connected by magnetic bonds, allowing them to remain in place even when their propellers are spun down. While extremely low-resolution at present, this allows for 5D printing technology with full synchronization between hardware and software editing of prints. We use the term 5D to indicate the real-time bi-directional nature of such edits: Future versions will be capable of real-time 3D animation of parts of the print.

LIMITATIONS

Our current implementation is limited to 12 simultaneously flying BitDrones. One limiting factor is downdraft, which restricts the ability of drones to fly directly over top of each other. A second limiting factor is drift due to turbulence when many drones are flying in a confined space. BitDrones are, however, sufficiently stable to hover within 15 cm from one another. Because of the above limitations, our current

design can only be used to represent relatively sparse 3D voxel models.

The lack of pixel density and resolution achievable with our system poses considerable challenges to support tasks that require a lot of detailed information. We considered combining floating voxels with a contextual large graphic display to complement the lack of resolution and density, however, we focused our efforts on developing a proof of principle for a tangible three-dimensional floating interface rather than detailed graphic renderings.

While we claim that BitDrones are immersive without requiring user augmentation, as evidenced by the DisplayDrone touchscreen, our current implementation *does* rely on the use of a Vicon to track user's interactions with ShapeDrones and PixelDrones. Another limitation of our system is that drones produce a clearly audible noise, a considerable challenge for video telepresence.

Finally, the system as presented lacks robustness for a large number of user trials. Specifically, the power requirements to run these trials as well more robust PID loops to correct for air draft and turbulence generated by BitDrones in close proximity are technical challenges that need to be addressed in order to perform a meaning and thorough evaluation of our system.

FUTURE DIRECTIONS

We aim to increase the number of simultaneously flying BitDrones by allowing drones to connect with one another magnetically, and spin down when stacking on top of one another to create vertical structures. Another solution is to use more robust motors to compensate for downdrafts caused by other drones.

That said, we do expect to be able to scale up our architecture to include hundreds of smaller drones. One of the benefits of smaller drones is that they generate significantly less turbulence, allowing them to fly in tighter formations.

To eliminate the need for user augmentation, future BitDrones prototypes will be outfitted with capacitive sensors for all touch and drag operations. Tracking of a remote user's head movements in the video conferencing scenario will be performed using a 3D depth camera, leading to a completely marker-less system that requires no user augmentation to experience immersive 3D tangible interactions.

While we used DisplayDrones to combat the lack of pixel density and resolution, we believe when miniaturized, and with more drones, our system will more fully embody the discussed interaction scenarios.

Currently, the battery life of BitDrones is limited to about 7 minutes. To address this limitation, we propose self-docking drones that use inductive recharge stations when not active. Furthermore, the audible noise produced by BitDrones can be improved by reducing mechanical imbalances associated with off-the-shelf hardware. Although ShapeDrones are currently safe to touch due to their protective mesh, PixelDrones and DisplayDrones have exposed propellers. We will include guards in future designs, thus ensuring user safety. Our next step is to devise and evaluate our interaction language for future self-levitating tangible user interfaces.

CONCLUSIONS

We presented BitDrones, a toolbox for building interactive real reality 3D displays with nano-quadcopters as self-levitating tangible building blocks. Our prototype is a first step towards interactive self-levitating programmable matter, where the user interface is represented using Catomic structures. We discussed 3 types of BitDrones: PixelDrones, with a single color RGB LED and a small OLED display; ShapeDrones, which are BitDrones that carry a pre-molded 3D printed frame in larger geometric shapes; and DisplayDrones, with a curved thin-film 720p FOLED touchscreen and Android functionality. We presented a number of unimanual and bimanual input techniques: touching, dragging, throwing and resizing of individual drones as well as compound models, and self-levitating user interface elements, including cone trees, 3D canvases and alert boxes. Finally, we discussed applications of BitDrones with 3D design, InfoVis and telepresence scenarios that exemplify the potential functionality of this new category of real reality interfaces.

ACKNOWLEDGEMENTS

This work was funded by the generous support of NSERC of Canada and Immersion, Inc. We thank Jordan van der Kroon for his work on the first version of BitDrones OS.

REFERENCES

1. Eric Akaoka, Tim Ginn, and Roel Vertegaal. 2010. DisplayObjects: Prototyping functional physical interfaces on 3D styrofoam, paper or cardboard models. In *Proc. TEI '10*, 49-56.
2. 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 (May 2012), 753-773.
3. Atelier Zafari, Cube House Project. *http://atelier-zafari.com/cube-house/*
4. Federico Augugliaro, and Raffaello D'Andrea. 2013. Admittance control for physical human-quadrocopter interaction. *Control Conference (ECC)*, IEEE, 1805-1810.
5. Steve Benford, Chris Greenhalgh, Gail Reynard, Chris Brown, and Boriana Koleva. 1998. Understanding and constructing shared spaces with mixed reality boundaries. *ACM Trans. Comput.-Hum. Interact.* 5, 3 (September 1998), 185-223.
6. Hrvoje Benko, Ricardo Jota, and Andrew Wilson. 2012. MirageTable: Freehand interaction on a projected augmented reality tabletop. In *Proc. CHI'12*, 199-208.

7. William Buxton. 1992. Telepresence: Integrating shared task and person spaces. In *Proc. of Graphics Interface'92*, 123-129.
8. Jessica R. Cauchard, Jane, L. E., Kevin Y. Zhai, and James A. Landay. 2015. Drone & me: An exploration into natural human-drone interaction, In *Proc. UbiComp'15*, 361–365.
9. George W. Fitzmaurice, Hiroshi Ishii, and William Buxton. 1995. Bricks: Laying the foundations for graspable user interfaces. In *Proc. CHI'95*, 442-449.
10. Flight Assembled Architecture. *www.gramaziokohler.com/web/e/projekte/209.html*
11. 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.
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. 2005. Programmable matter. *IEEE Computer, 38, 6*, 99-101.
14. David Holman, Roel Vertegaal, Mark Altosaar, Nico Troje, and Derek Johns. 2005. PaperWindows: Interaction techniques for digital paper. In *Proc. CHI '05*, 591-599.
15. HoloLens. 2015. *www.microsoft.com/microsoft-hololens*
16. Eva Hornecker and Thomas Psik. 2005. Using ARToolKit markers to build tangible prototypes and simulate other technologies. In *Proc. INTERACT'05*, 30-42.
17. Hiroshi Ishii, and Brygg Ullmer. 1997. Tangible bits: Beyond pixels. In *Proc. CHI'97*, 234–241.
18. Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. 2012. Radical atoms: Beyond tangible bits, toward transformable materials. *Interactions* 19, 1 (January 2012), 38-51.
19. Doug L. James and Dinesh K. Pai. 2002. DyRT: Dynamic response textures for real time deformation simulation with graphics hardware. *ACM Trans. Graph.* 21, 3 (July 2002), 582-585.
20. Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable: Exploring the synergy between live music performance and tabletop tangible interfaces. In *Proc. TEI '07*, 139-146.
21. Mustafa Karagozler, Brian Kirby, Wei Jie Lee, Eugene Marinelli, Tze Chang Ng. 2006. Ultralight modular robotic building blocks for the rapid deployment of planetary outposts. *RASC-AL Forum*.
22. 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.
23. 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.
24. Jinha Lee and Cati Boulanger. 2012. Direct, spatial, and dexterous interaction with see-through 3D desktop. In *ACM SIGGRAPH 2012 Posters*, 69.
25. Daniel Leithinger and Hiroshi Ishii. 2010. Relief: A scalable actuated shape display. In *Proc TEI'10*, 221-222.
26. MIT Flyfire. 2014 - *http://senseable.mit.edu/flyfire/*
27. Paul Milgram and Fumio Kishino. 1994. A Taxonomy of Mixed Reality Visual Display. In *IEICE Trans. Information and Systems,* vol. E77-D, no. 12, 1321-1329.
28. Paul Milgram and Herman Colquhoun Jr. 1999. A Taxonomy of Real and Virtual World Display Integration. *Mixed Reality - Merging Real and Virtual Worlds.* Springer Verlag, p. 5-30.
29. Kei Nitta, Keita Higuchi, and Jun Rekimoto. 2014. HoverBall: Augmented sports with a flying ball. In *Proc. AH'14*, Article 13.
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. Kevin Pfeil, Seng Lee Koh, and Joseph LaViola. 2013. Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles. In *Proc. IUI'13*, 257-266.
33. 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.
34. John Underkoffler and Hiroshi Ishii. 1999. Urp: A luminous-tangible workbench for urban planning and design. In *Proc. CHI '99*, 386–393.
35. Hayes Solos Raffle, Amanda J. Parkes, and Hiroshi Ishii. 2004. Topobo. In *Proc. CHI '04*, 647–654.
36. Ramesh Raskar, Greg Welch, Kok-Lim Low, and Deepak Bandyopadhyay. 2001. Shader Lamps: Animating real objects with image-based illumination. In *Proc. of the 12th Eurographics Workshop on Rendering Techniques*, Springer-Verlag, 89-102.
37. George. G. Robertson, Jock. D. Mackinlay and Stuart K. Card. 1991. Cone trees. *In Proc. CHI '91*, 189–194.

38. Michael Rubenstein, Christian Ahler and Radhika Nagpal. 2012. Kilobot: A low cost scalable robot system for collective behaviors. *ICRA'12*, 3293-98.
39. Michael Rubenstein, Alejandro Cornejo and Radhika Nagpal, R. 2014. Programmable self-assembly in a thousand-robot swarm. *Science 345(6198)*, 795-799.
40. Juergen Scheible, Achim Hoth, Julian Saal and Haifeng Su. 2013. Displaydrone: A flying robot based interactive display. In *Proc. PerDis'13*, 49-54.
41. 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.
42. Ivan Sutherland. 1965. The Ultimate Display. In *Proc. IFIP* 65, 2, 506-508.
43. Daniel Szafir, Bilge Mutlu and Terrence Fong. 2015. Communicating directionality in flying robots. In *Proc. HRI'15*, 19-26.
44. Grzegorz Szafranski, and Roman Czyba. 2011. Different approaches of PID control UAV type quadrotor. *IMAV'11*, 70-75.
45. Tomasso Toffoli and Norman Margolus. 1991. Programmable matter: Concepts and realization. *Physica D: Nonlinear Phenomena* 47, 1, 263-272.
46. George Whitesides and Bartosz Grzybowski. 2002. Self-assembly at all scales, *Science, 295(5564)*, 2418-21.
47. Andrew D. Wilson and Hrovje Benko. 2010. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proc. UIST '10*, 273-282.
48. Roel Vertegaal and Ivan Poupyrev. 2008. Organic user interfaces. *Communications of the ACM* 51, 6, 26-30.
49. Vicon. 2015. *http://www.vicon.com/*