AirConstellations: In-Air Device Formations for Cross-Device Interaction via Multiple Spatially-Aware Armatures

Nicolai Marquardt
Microsoft Research, UCL

Hugo Romat
ETH Zürich, Microsoft

David Ledo
Autodesk Research

Teddy Seyed
Microsoft Research

Nathalie Henry Riche
Microsoft Research

Michel Pahud
Microsoft Research

Chunjong Park
University of Washington

Eyal Ofek
Microsoft Research

Christian Holz
ETH Zürich

Frederik Brudy
Autodesk Research

Molly Jane Nicholas
University of California, Berkeley

Bongshin Lee
Microsoft Research

William A. S. Buxton
Microsoft Research

Ken Hinckley†
Microsoft Research

Examples: Multi-Device Apps & Techniques

Reconfigurable workspaces with tiled virtual windows and tool palettes in productivity applications

Video conferencing with registered person and task space, and proxmic-aware side conversations

Layered cross-device window management for context switching

Ad-hoc horizontal tabletop-like flat surface for data analysis

AirConstellations Platform & Sensing

Fine-grained position and orientation of device formations

2–5 semi-fixed, spatially-aware armatures for dynamic in-air device formations

IMU sensing: tilt position, analog rotation sensors (avoid sun glare)

Off-the-shelf device mount

2–5 semi-fixed, spatially-aware armatures for dynamic in-air device formations

Design space

16 key dimensions of spatially-aware dynamic in-air device formations:

- Arrangement
- Orientation
- Structure
- Scaling
- Form
- Modality
- Dynamics
- Size
- Users
- Distance
- Motion
- Direction
- Spatial sensing
- Transfer function
- Behavior
- Actuation

Figure 1: AirConstellations enables flexible cross-device interactions where users can move, re-orient, and pose devices together to form dynamic in-air device formations.

ABSTRACT

AirConstellations supports a unique semi-fixed style of cross-device interactions via multiple self-spatially-aware armatures to which users can easily attach (or detach) tablets and other devices. In particular, AirConstellations affords highly flexible and dynamic device formations where the users can bring multiple devices together in-air — with 2–5 armatures poseable in 7DoF within the same workspace — to suit the demands of their current task, social situation, app scenario, or mobility needs. This affords an interaction metaphor where relative orientation, proximity, attaching (or detaching) devices, and continuous movement into and out of ad-hoc ensembles can drive context-sensitive interactions. Yet all devices remain self-stable in useful configurations even when released in mid-air.

We explore flexible physical arrangement, feedforward of transition options, and layering of devices in-air across a variety of multi-device app scenarios. These include video conferencing with flexible arrangement of the person-space of multiple remote participants around a shared task-space, layered and tiled device formations with overview+detail and shared-to-personal transitions, and flexible composition of UI panels and tool palettes across devices for productivity applications. A preliminary interview study highlights user reactions to AirConstellations, such as for minimally disruptive device formations, easier physical transitions, and balancing "seeing and being seen" in remote work.
1 INTRODUCTION

People increasingly own and use multiple devices in their personal workspaces, a long-term trend (e.g., [81]) accelerated by remote work during the COVID-19 pandemic [90]. In the course of a single day, users may go from a teleconference to data analysis to editing spreadsheets and back again—with unpredictable interruptions of home life (e.g., partners, children, pets, deliveries) driving task shifts or spur-of-the-moment reconfgurations. Within these personal device ecologies, cross-device computing leverages the complementary roles and micro-mobility [62] afforded by individual devices to support a wide range of application scenarios, use cases, and communication [8].

In these emerging cross-device ecologies, most existing work either uses fixed setups with combinations of interactive surfaces embedded in the environment for individual and small-group use (e.g., [7, 39, 102, 103]), or mobile cross-device setups that link devices such as tablets, phones, or other digital surfaces into ad-hoc, lightweight federations (e.g., [13, 32, 60]). Physical reconfgurations to better adapt to changes in task, users, or other context [56, 78, 90] remain diffcult in fxed spaces—and while easier in mobile setups, there are often only limited options to spatially arrange devices around a workspace, so they often remain either hand-held or placed f at a desk surface. Furthermore, for most mobile setups, sensing where devices are in space (which drives many cross-device interaction techniques [8]) remains challenging (e.g., [29, 74]).

In between these two classes of fxed vs. mobile multi-device use, we can position a third category of semi-fxed cross-device setups, akin to Hall’s semi-fxed architectural features [31]. The goal is to preserve the modularity and rapid reconfgurability of mobile setups, while at the same time allowing physical rearrangement of diverse configurations as the task and situation demand, facilitating people’s rich practices of spatially organizing information [46]. There have been several investigations into more fexible device configurations, such as through actuated furniture [2, 27, 28, 58] or re-connectable devices [64, 73, 87, 88]. Yet, preserving high degrees of modular reconfgurability, affording fuid and lightweight changes of physical workspace device confgurations, remains a challenge. This becomes even more diffcult for interactions on a desktop-scale, with fast-changing working practices [90] and due to limited screen real-estate in the presence of multi-tasking, with varied applications side-by-side with one another.

AirConstellations contributes a new technical approach, platform, and techniques for in-air, poseable multi-device assemblages—devices not only f at on a table, but floating in front, at varying angles, or huddled together (Figure 1). AirConstellations workspaces are composed of 2–5 fexible armatures with attached devices (such as tablets), allowing easy physical movement and positioning of devices around a workspace. In particular, we retrofit multiple off-the-shelf device mounts with (low-cost and robust) sensors to determine the spatial location and orientation of all devices, which we then combine into a 7DoF model of device assemblages around the workspace. This sensor augmentation opens up new possibilities for sensor-driven application behaviors. To inform the design of this new category of mid-air, spatially-aware cross-device setups, we synthesized a design space (cf. Figure 3)—along 16 dimensions of hardware conguration (arrangement, structure/orientation, layering, scale, fxture, form, modality, dynamics, and size) and interaction (users, distance, motion/direction, spatial sensing, transfer function, behavior, and actuation).

We then contribute a set of applications and interaction behaviors, illustrating techniques across different segments of this device conguration design space. In particular, we use the sensed information of 3D position and orientation of all connected devices to drive applications (with a low transaction cost) responding to any changes of orientation or proximity between devices. Our applications include a video conferencing app facilitating liquid separation of task- and person-space [10], the use of layered and tiled device formations with techniques such as overview+detail and shared-to-personal transitions, and fexible workspaces and tool palettes in productivity applications. We also illustrate the use of ad-hoc operations for managing windows and fles across devices, and explore feedforward and feedback to continuously reveal interaction possibilities, so that the user can discover, select between different options, and act on (or choose to ignore) AirConstellations’ spatially-aware behaviors. A preliminary (remote) interview study with eight participants suggests that AirConstellations could afford rapid transitions or adjustment of people’s workspaces while minimizing disruption, allow balancing of “seeing and being seen” during remote collaborations, and afford meaningful physical proximity-based techniques that facilitate sharing.

In summary, we contribute:

- Air Constellations system: Semi-ffexible cross-device interaction with dynamic in-air device formations via multiple self-ffable, spatially-aware armatures;
- Design space that articulates 16 key dimensions of (physical & spatial) hardware conguration and interaction for semi-ffexible cross-device setups;
- Application scenarios & interaction techniques for the dynamic device formations afforded by AirConstellations, including layering, tiling, adjacency, and feedforward animations that surface capabilities to users; and
- Initial user insights (from a remote interview study) for the highly reconfgurable workspaces, applications, and techniques thus afforded.
Our work aims to show how multiple AirConstellations devices used together in the same operating volume can afford some aspects from mobile interaction (and micro-mobility concepts from social theory [62]) while simultaneously offering some properties of semi-fixed device arrangements in a way that is flexible and delightful, allowing the effort of forming a particular device arrangement to repay itself over time because it stays put in mid-air once released. Such devices, in combination with appropriate applications, feedback, interaction techniques, and distributed systems foundations such as SurfaceFleet [9], may open up an intriguing new design space of highly dynamic and easily reconfigurable personal workspaces with dynamic in-air device formations.

2 BACKGROUND

We synthesize work in cross-device computing, in particular for fixed, instrumented environments and mobiles, ad-hoc collaborative interactions, challenges of spatial tracking, and work towards reconfigurable workspaces.

2.1 Multi-Surface Workspaces

Large interactive surfaces can form parts of effective individual or collaborative workspaces (often inspired by Wellner’s seminal DigitalDesk [102]). Varying physical form factors and orientation can afford collaborative work from horizontal [17, 20], tilted [16, 38, 69], to vertical [103] surfaces. Non-planar, bent interaction surfaces [101, 105] can facilitate reach and visibility of digital information.

To better support collaborative work, multiple interactive surfaces can be combined in interactive environments [87], such as for collaborative astrophysics explorations [103], work in design studios [21], facilitating video-mediated communication [51], or data analytics [15, 85]. Most combine horizontal and vertical displays, some use projectors [104], or other forms of output (cf. taxonomies [6, 8]). Following Hall’s terminology [31], these interactive environments could be categorized as fixed features (Figure 2), parts of a space that are not or only rarely reconfigured or changed. To make interactive spaces adapt to changes in context, some follow a hybrid approach [3, 4, 37, 40], for example, by including mobile devices and laptops in a “walk up and share environment” [103].

2.2 Ad-hoc Interactions Across Mobile Devices

In contrast to fixed interactive spaces, mobile setups are at the opposite end of the mobility spectrum (Figure 2) — built to afford high degrees of flexibility and dynamic configurations for different contexts. Some of these setups allow users to combine mobile devices (e.g., phones, tablets, e-readers, smartwatches) in unified interactions spanning across multiple devices [9, 61]. Such setups can then support a wide range of user activities & tasks, including active reading [13, 34], sharing personal media [60], sensingmaking [106], and ad-hoc data exploration [52].

Interaction techniques (e.g., pen/touch input, hand gestures, or device motion) can facilitate exploration, transfer, and other content engagement across devices (cf. [8, Table 3]). Rekimoto’s seminal work [76] introduced pick-and-drop for transfer. Many other techniques followed using, for example, stitching [36], 3D hand motion [75], tapping and dragging [32], tilting and portals [66], redirecting content [108], eye-tracking [99], or conduit gestures [12] for fluid data transfer in cross-device ecologies. Related techniques include spatial information lenses [19] and merged displays [60, 82].

To build techniques spanning across devices requires flexible software infrastructure. Therefore, previous work explored approaches for developing dynamic cross-device applications, for example, by introducing shareable dynamic media [47], using scripting for cross-device behaviors [14], generating semi-automatic adaptations based on capabilities [70], and debugging cross-device applications [71].

2.3 Reconfigurability

Reconfiguration of a workspace or device setup can be beneficial for various use cases: better supporting a particular social interaction, adapting small workspaces or spaces with multiple uses [90], or changing a setup for a different activity. We apply Hall’s [31] term of semi-fixed features to describe such reconfigurable device formations. One approach for such reconfigurability is the integration of actuated, shape-changing mechanisms — e.g., surfaces changing from horizontal to vertical [27], drafting-table orientation [58], or shape changes of the surface area depending on tasks [27, 28]. On a smaller scale, HoverPad [83] uses an actuated screen above a tabletop, and Living Desktop [2] introduces mechanically actuated monitors and keyboards that reconfigure depending on a task or social context.

In some cases, the boundaries between technology and interior architecture blur, when devices or displays become furniture-like modular blocks. For example, AdapTable [50] uses smaller tabletop-blocks, UbiWall [45] allows modular display cubes to be reassembled when needed, and Foxels [73] integrates various input and output capabilities in furniture cubes. Similarly, i-LAND [87] explores such device-furniture hybrids, for example, with ConnecTables [88] where display spaces merge when moving tables in direct proximity. u-Texture [48] introduced grid-like structures and links that allow surfaces to be assembled in different horizontal and vertical arrangements, and SurfaceConstellations [64] uses 3D-printed brackets to form multi-device workspaces. Shape-changing [1, 92] and modular [24] mobile devices allow reconfigurability at the smallest scale.
2.4 Sensing Location and Orientation

For most cross-device setups, one particular challenge is to infer the position and orientation of the devices to enable any spatially aware cross-device interaction techniques (e.g., easily re-directing input to devices, directional gestures transferring content). This is why cross-device research [8] investigates strategies to sense inter-device proximity and orientation. Some use computer vision of RGB [29], depth cameras [75], marker-based motion capture [65, 107], marker recognition [59], polarization filters [74], or eye tracking [99]. Other techniques apply short-range infrared sensing [68], radio-based sensing such as Bluetooth [22, 42], or near-field communication (NFC, RFID) [77]. Hybrid approaches are promising, for example, fusing radio- and camera-based sensing [66], or Bluetooth-radio and acoustic stereo signal positioning [41]. In addition, aside from automated tracking, synchronous gestures [33] and stitching [36] are effective techniques for quickly forming device federations. Reliable and robust sensing, however, remains a “major challenge” [8]: vision-based sensing achieves relatively high spatial resolution but faces difficulties with occlusion, lighting conditions, and potentially undesirable always-on camera tracking; and radio-based sensing struggles with achieving high spatial resolution or accuracy. Furthermore, many of these setups require outside-in-tracking [8], which can limit their use and deployability.

2.5 Using Spatial Changes as Sensor Input

We can leverage information about any changes of the spatial configuration of a device (or devices) to function as input, affecting application behavior. For example, Codex [35] (and similar approaches that followed [80]) senses the opening angle of the connecting hinge to inform context switching and other adaptations of dual-surface applications. BEXHI [72], as one example of deformable interfaces, uses stretching and bending as direct input. For stationary setups, sensing input of a display’s tilting angle can drive context-appropriate transitions [79], and tracked vertical up and down movements of a tablet can facilitate 3D drawing [53].

Beyond the sensing of a single degree-of-freedom movement (e.g., up/down, tilting angle), tracking the full position (3DoF for x/y/z position) and orientation (additional 3DoF of yaw/pitch/roll orientation) of a device in space allows for the use for rich 3D spatial interaction. Most notable is activeLENS [96], a spatially tracked arm-mounted display, functioning as a window into the hybrid digital-tangible geospace of metaDESK (sensed through a 6DoF magnetic tracker). BoomChameleon [21, 93, 94] introduced a similar setup of a display mounted on a tracked mechanical boom, for rich 3D viewpoint control and spatial annotations.

We apply this particular combination of spatial awareness and high degrees of reconfigurability of spatially-aware, arm-mounted displays to multi-device interactions. As summarized in Figure 2, our work is positioned in the space between fixed and mobile device setups, where we contribute a new approach and platform for dynamic in-air device formations. Through this modularity and high degree of reconfigurability, such semi-fixed setups have the potential to effectively support changing working practices, such as the need for increased work at home, with possibly smaller or shared workspaces [90], or demands for more flexible device and furniture configurations in offices for collaboration [56].

3 DESIGN SPACE

To better understand the parameters for creating dynamic in-air device formations, we synthesized a design space along dimensions that represent varying interaction techniques, sensing fidelity, physical characteristics, and spatial arrangements (Figure 3). We incorporated other classifications, in particular the dimensions of flexible dual-device configurations [35, Fig. 2], articulated orientation angles of tilted drawing boards [79, Fig. 3], mappings of the field of cross-device interaction [8, Fig. 2 and Table 3], modular assemblies of multi-device workspaces [64, Fig. 3], and the disparate but related dimensions of graspable user interfaces [20, Table 1]. We identified the following key dimensions relevant for the design of spatially-aware dynamic in-air device formations:

- **P1 Arrangement** — The spatial configuration of devices: separated [3, 103], continuous surface [75, 82] (e.g., edge to edge), overlapping [23, 37], and/or occluding.
- **P2 Structure/Orientation** — The physical orientation of devices in space: horizontal [17, 18, 88], vertical [64], tilted [79], concave [101, 105] vs. convex or both [35, 44], or hybrid [28, 64].
- **P3 Layering** — Spatial layering of devices around a person’s workspace, from one to n layers [23].
- **P4 Scale** — The number of devices combined in one cross-device formation.
- **P5 Fixture** — Are devices configured as a stand-alone [64] setup, for example, attached to a desk, or do they function as a companion [87] for another device (e.g., smaller devices attached to a large interactive whiteboard).
- **P6 Form** — From heterogeneous setups (using similar devices) [75, 108] to heterogeneous configurations (devices of varying size or different capabilities) [3, 37, 103].
- **P7 Modality** — Which input or output devices form part of the multi-device setup. Input modalities can be multi-touch (e.g., many of our examples use multi-touch tablets), physical keyboards, or cameras. Output modalities can be screens, projectors, or other hardware.
- **P8 Dynamics** — How often is the spatial configuration of devices changed: from permanent setups, to changes every few weeks/days/hours/minutes, to continuous changes (cf. Figure 2).
- **P9 Size** — The size of devices included in the formation: small (e.g., phone), medium (e.g., tablet, laptop, monitor), and large (e.g., digital whiteboard).
- **P10 Users** — Workplace for single user interactions, multi-user collaboration, or multi-user co-located interactions.
- **P11 Distance** — How close are the devices physically positioned in relationship to the user: directly in front, in arms reach, further away but reachable, or far.
- **P12 Motion/Direction** — Different directions of approach when moving one device closer to another: from left or right side; top or bottom of the device; approaching from the front or back; or perpendicular.
- **P13 Spatial sensing** — The fidelity of spatial sensing available: from no sensing to sensing of presence, 1DoF [58, 69, 79], 2DoF [57], 3DoF, 6DoF [83, 93], and 7+DoF (including overall formation).
**Physical and Spatial Configuration**

<table>
<thead>
<tr>
<th>Form</th>
<th>Layering</th>
<th>Scale</th>
<th>Arrangement</th>
<th>Orientation</th>
<th>Structure/Fixture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separate</td>
<td>Single layer</td>
<td>1</td>
<td>Vertical</td>
<td>Horizontal</td>
<td>Homogeneous</td>
</tr>
<tr>
<td>Continuous</td>
<td>Two layers</td>
<td>2</td>
<td>Tilted</td>
<td>Vertical</td>
<td>Heterogeneous</td>
</tr>
<tr>
<td>Overlapping/Occluding</td>
<td>Three layers</td>
<td>3</td>
<td>Convex</td>
<td>Horizontal</td>
<td>Hybrid</td>
</tr>
<tr>
<td>Stacked</td>
<td>n layers</td>
<td>4</td>
<td>Hybrid</td>
<td>Horizontal</td>
<td></td>
</tr>
</tbody>
</table>

**Modality**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>— multi-touch screens</td>
<td>— screen surfaces (e.g., monitors)</td>
</tr>
<tr>
<td>— physical keyboards</td>
<td>— projectors and other output hardware</td>
</tr>
</tbody>
</table>

**Dynamics**

- Permanent
- Change every few weeks
- Change every few days
- Change every few hours
- Change every few minutes
- Constant changes

**Size**

- Small (e.g., phones)
- Medium (e.g., tablet computer, laptops, monitors)
- Large (e.g., digital whiteboard)

---

Figure 3: Design space of dynamic in-air device formations: (left) physical and spatial configuration and (right) interaction and behavior (gray background indicates design dimensions we illustrate through examples in this paper).

15 **Transfer function** — How is the sensor input mapped to any interactive application behavior: no mapping, binary threshold (e.g., crossing a distance threshold between two devices triggers action), gradual [63], linear continuous [3], or non-linear continuous [79].

16 **Behavior** — The way applications implement different action behaviors in response to sensed input: manual selection (e.g., if there is no application opened), seeing suggested options that require confirmation (e.g., when there is no clear preference in the system of a default choice), showing options but pro-actively triggering the default option, and pro-actively automatically triggering an action (for example when only one option is available).

17 **Actuation** — Is any actuation mechanism used to reconfigure the position of devices in a formation. In the scope of this paper, we illustrate examples where users manually reconfigure formation. Other approaches are feasible, such as semi-automated (e.g., assisted repositioning) or automated (autonomous position changes using robotic arms) formations (similar to approaches in [2, 27, 28, 58, 83, 86]).

## 4 AirConstellations Platform

To enable easy reconfiguration of device setups, we designed a hardware platform that consists of off-the-shelf mount armatures (to hold tablets, phones, keyboards, and any other devices) that are retrofitted with sensors to continuously track the location and orientation of each attached device. We then used the tracking data from these sensors to drive a number of novel interaction techniques (introduced shortly). In this section, we explain the technical implementation of our platform enabling dynamic in-air device formations for cross-device computing.

### 4.1 Concept: Self-stable Armatures for In-Air Device Formations

For the flexible positioning of devices around a user, we needed a mechanical hardware construction that physically holds devices in place. To achieve this we use an armature design of a flexible articulated arm — similar to activeLENS [96] and BoomChamelon [93, 94] — but extend this concept to multi-device use and composition of device formations. In particular, we optimized our platform for easy reconfigurability in small-scale, multi-device setups — multiple displays and other devices that can be freely positioned in the same working volume. Due to size constraints of using this in desk-sized workspaces, we decided not to use a counter-balanced mechanical boom (which are often used for holding and positioning heavier cameras in movie productions), but instead use a 3-joint flexible tablet mounting arm [95] as the basis of our design. The devices we use in our setups (tablets, keyboards) are lighter and don’t require a fully counter-balanced mechanical construction.
4.2 Adding Spatial Awareness: Retrofitting Armatures with Sensors

We retrofit these off-the-shelf armatures with sensors to continuously track the position and orientation of each device. The technical setup of a single AirConstellations arm is illustrated in Figure 4.

In each AirConstellations arm we use an Arduino MKR 1010 Wi-Fi (SAMD21 Cortex-M0+) microcontroller, with a 1-to-8 I2C multiplexer (TCA9548A), connecting to four IMUs (Inertial Measurement Units), all either MPU6050 or MPU9250. We use one IMU for each of the two main joints of the arm, one for the base (so we can sense the placement of the base) and the other for the joint that connects to the tablet at the end of the mount (Figure 4). To eliminate yaw-axis drift (a challenge with most IMU-based sensing), we do not use any yaw-axis measurement from the IMUs but added mechanical rotation sensors. We use high-precision 10-turn rotation sensors (Bourns 3510) with a connected 1:5 gear translation in the base, and an additional rotation sensor connected to the top joint where the device gets connected. Our algorithm then combines the yaw-axis rotation measures (from the base and head rotation axis) with the pitch-axis measures provided by the IMUs for determining the 3DoF position of the device in space, together with 3DoF orientation of the device around the head of the mounting arm. Using off-the-shelf armature mounts and low-cost sensing hardware makes this approach scalable and easy to replicate. We release the STL designs for the 3D-printed sensor enclosures, attachments, and other fixtures (shown at the bottom of Figure 4).\(^1\)

We built five AirConstellations hardware prototypes, each with one device attached (using tablet devices such as the Microsoft Surface Go, Surface Pro, and Surface Book, all connected through local WiFi, and Bluetooth peripherals such as keyboards). The armatures can be attached to any surface, such as a desk or a small table. On first use, the position of the location where each of the armatures is attached to needs to be configured in our software (i.e., the relative location and distance from one base to the next). After this one-time configuration, all AirConstellations armatures continuously track the position and orientation of the attached device when moved, and send their position information via UDP packets to a PC merging all the position information and running our prototyping software (Figure 5).

4.3 From Mobile to Semi-fixed

We integrated a simple detection for when devices are added or removed from a formation. For example, users can take their mobile devices and join them to a semi-fixed AirConstellations formation, like docking their phone or tablet into one of the armatures. The device is recognized by the armature mount, where we compare the sensed orientation of the mount to the orientation sensor of the device (i.e., the internal IMU) to recognize docking or removing, and the mobile device then becomes part of or is removed from the current setup. Overall this orientation matching approach works well, but could lead to false recognition due to identical device/mount orientation — other techniques could be used instead, such as an RFID-based detection of devices attached to the armatures (e.g., RFID reader in the armature, tags attached to the back of devices).

\(^1\)https://github.com/nicmarquardt/airconstellations

Figure 4: AirConstellations technical implementation: adding spatial sensing to a 3-joint flexible armature.

This docking behavior is similar to Codex [35], where devices can be disconnected from the connecting joint when needed. Mobile devices can join semi-fixed device formations when needed (moving along the design space dimension \(d\) from left to right), and disconnected when again needed as mobile, portable devices.

4.4 Stabilization and Software Filtering

We use multiple levels of software filtering to stabilize the sensor signal and minimize jitter when using the sensed position of the devices to drive application responses. On the microcontroller, we do 12-bit sampling of the analog rotation sensors, oversampling at 110Hz, and use an exponential filter as a first-level filtering for all measured orientation angles — from the two rotation sensors and the four IMUs. The base then connects to a PC over WiFi (using the microcontroller’s NINA-W10 radio of the n-blox chip), and streams all sensor values at 60 Hz via UDP to a C# software tool (Figure 5). In the software (Figure 5b), we use an additional filter level over all sensor values using the \(1\)\(\text{\$}\) filter \([11]\) \((\beta = 0.007, \text{dcutoff} = 1.1, f_{\text{min}} = 0.02)\).
We built a rapid prototyping tool (C# Windows software) to explore weight experimentation with dynamic application behaviors — that is, exploring concepts of how applications running on the devices connected to AirConstellations could dynamically respond to users’ interactions with the formation, when they move devices in and out of formations, change the orientation of devices, and so on. Our prototyping tool (summarized in Figure 5c-d) created a scaffolding that empowered us to create interaction-driven animations by (1) connecting multiple devices together, (2) processing the sensor data, and (3) enabling translating the sensor data into rich continuous animations as the interactions are taking place.

Designing animations of application behaviors. To visually author these kinds of real-time behaviors — without the overhead of full application implementations — we designed our tool so it uses interactive animations of native application behaviors that can be implemented with a wide variety of prototyping tools, such as PowerPoint animations, or video mock-ups from AfterEffects. We built on existing approaches, such as Astral [55] and the one by Romat et al. [79], and applied them to cross-device prototyping. Consequently, our tool facilitates quick exploration, testing of alternatives, and fine-tuning dynamic behaviors. After designing a new application behavior (e.g., a PowerPoint animation showing feedback of devices approaching each other), we import these as video sequences into our prototyping tool (Figure 5c).

Sensor-driven animations. Our tool segments all prototype video animations into sequences of still frames, and — most importantly — provides detailed settings for specifying how the live sensor tracking data is mapped to the behaviors of the different prototype animations. To do this, the designer (1) selects the sensor input to control the animation (such as sensed position, movement along a path, orientation changes, etc.), (2) chooses a transfer function (e.g., linear, non-linear, ease-in/out) that controls the mapping of sensor input to animation frame, and (3) specifies the animation sequence to show on individual devices. The prototyping tool also provides a number of optional settings to fine-tune the processing of the incoming sensor data (e.g., filter thresholds, cutoff values) or playback of the animation (e.g., reverse playback, select subset of the animation sequence). This prototyping approach allows exploring different facets of user flow (combining animations and states) within a single application. For example, one application sequence might be mapping the proximity of two devices of a formation to show the animation, another sequence might respond to changes in device orientation, and so on.

Interactive playback of animations. The prototyping viewer client software (Figure 5d) displays the different animated application sequences on the devices attached to AirConstellations armatures. Each of the viewer clients receives the real-time sensing data (as part of a UDP packet stream) of the AirConstellations platform. The client software then dynamically animates the application interface in response to the physical movement of the devices and physical changes of the formations (Figure 5).

Prototyping and refinement. We designed 52 prototype application behaviors, which demonstrate different applications and operating system functions in a cross-device setting. From these behaviors we selected, refined, and combined a representative sample of our design space which we discuss in the next section. This prototyping process propelled a rapid exploration of many techniques and application behaviors while enabling iteratively refining the designs as we increased our prototype fidelity. Because all of the resulting applications use real-time sensing input (moving devices, proximity, orientation, touch input, etc.), it is possible to experience the different nuances of how the spatially-aware applications can respond to dynamic changes of formations of people and devices.
5 AIRCONSTELLATIONS APPLICATIONS AND INTERACTION TECHNIQUES

To illustrate the dimensions of the design space for dynamic in-air device formations, we now present a number of concrete application scenarios and interaction techniques. In particular, we demonstrate how AirConstellations seamlessly manages the locations and orientations of devices to enable applications to easily respond to spatial (re)configurations and inter-device proximity.\(^2\)

5.1 Multi-Device Video Conferencing

Our multi-device video conferencing application allows dynamic reconfiguration of the task and person space. Thus, it facilitates transitions between group chats and face-to-face side conversations, incorporates light-weight use of tools such as whiteboards during video calls, and enables document sharing through touch and tilt gestures.

Device formations for separated person and task space. This example illustrates the separation of person and task space [10] during video calls, related to the Hydra four-way video conferencing [84], and techniques on large surfaces [51, 89], blended interaction [40], and tilting single-device screens [79]. We translate this technique to dynamic in-air multi-device workspaces, offering more degrees of freedom for possible reconfigurations and adjustments. When moving a second device of our AirConstellations setup closer to a tablet running a video conference call (Figure 6a), the video feed of individual participants can be transferred from the horizontal screen to any nearby vertically positioned tablet using a sliding gesture into the direction of the device (Figure 6b-c). Once all three remote participants have been moved to a separate screen (recreating the experience of four people facing each other in a physical meeting), the fourth device in the center transitions into task space mode and opens the collaborative whiteboard sketching app of the team (Figure 6d). Thanks to the flexibility of the armature, one can position the camera optimally (with each of the three person space devices) independently from the ergonomic placement of the task space device (i.e., placing the whiteboard sketching app horizontally in front for sketching or writing on the surface).

Ad-hoc proxemic-aware transitions to individual side conversations. Changing from a group conversation to a two-person side conversation can be difficult to do in digital video conferencing compared to how easily and naturally we can initiate a side conversation with someone in a physical meeting. As shown in Figure 6, AirConstellations supports a lightweight technique to transition from group to individual conversations, inspired by proximically-aware physical surrogates [25] and leveraging proxemic interactions [26, 49].

---

\(^2\)We focus on demonstrating how users might work with the AirConstellations platform, primarily through the exploration of the design space, and a combination of novel and replicated examples [54].
of the three speakers’ proxy view, separating them from the device formation [P1] and bring the screen closer towards themselves (moving along spectrum of [32] decreasing proxemic distance, moving that individual speaker from Hall’s social to the personal space [31]), to initiate a private side conversation (Figure 6e). When doing so, the audio channel to the other person remains open, but gets muted for all other participants, allowing for a conversation in private. All other meeting participants can continue the ongoing conversation (with audio/video). As the user moves the device back to the group of other devices, the side conversation ends.

**Feedforward vs. manual confirmation of application behavior.** For revealing multiple options for what should happen in response to spatially-aware device movements, we introduce feedforward techniques (illustrating different techniques of [32]) and depending on sensor input [5]. As Vermeulen et al. state in their paper [98], “feedforward [...] tells users what the result of their action will be.” In this variation of the video conferencing app, when a user brings a second tablet closer (Figure 6f) a side-panel fades in with preview-icons of all four remote participants of the conversation (Figure 6g). This semi-transparent panel provides a feedforward preview of the options to move one of the remote participants’ video stream to the other device. The user can simply click & drag one of the camera preview icons from the sidebar (Figure 6h) to move the participant to the new device. If no option is selected, the panel fades out after five seconds.

**Light-weight, transient use of collaborative tools.** AirConstellations also supports ad-hoc, transient application behaviors during video conversations. For example, when separating the tablet away from the device formation and moving it closer, while tilted like a drawing board [P2], we fade in a shared Whiteboard from the bottom half of the screen (Figure 6h), giving easy access for shared writing and drawing. Moving the tablet back as part of the device formation makes the whiteboard canvas disappear.

**Sharing gestures for individuals and groups.** Our app facilitates sharing of documents during video conferencing. A user can share documents with other individual participants by using directional sliding gestures (Figure 6j) into the direction of the proxy-screen of another participant of the conversation, which creates a transient shared portfolio [9] (Figure 6k). A fade-in animation indicates this document sharing, where the shared portfolio slowly fades in when performing the sliding gesture — and the sharing can be revoked easily by reversing the gesture. Sharing documents with all participants during video conferencing can be done through a more expressive gesture — such as tilting devices. A user can select the document to share on the screen (Figure 6l), followed by tilting the device into the direction of the device formation with all remote participants (Figure 6m), which shares the document with everyone. By design this gesture requires more effort compared to the sliding gesture for sharing with only one individual — we match the effort of the gesture with its impact (i.e., minimal gesture for sharing with one person, more effortful gesture when sharing with all participants).

### 5.2 Productivity Applications

Proximity-dependent cross-device behaviors can facilitate managing multiple application windows, tool palettes, as well as help redirect input devices when using productivity applications (e.g., presentation software, email clients, or development tools).

**Cross-device hand-over of tool palettes.** When a person moves a tablet into the proximity of another device running an application (Figure 7a), tool palettes that can be transferred onto the second device are shown in a semi-transparent overlay on the side of the screen (Figure 7b). The person can then select and expand the tool palettes by dragging, spreading them out on screen (Figure 7c-d). For the tool palettes transfer technique, we again use feedforward to indicate action possibilities: we map the sliding-in/out of the palette window (Figure 7b) directly to the continuous movement [5] of the approaching device (and a touch gesture confirms the action, otherwise the preview slides away with no action).

**Multiple options and default behaviors.** When implementing proximity- and orientation-dependent application behaviors for device formations, there can be cases where multiple options are available to choose from (e.g., different actions resulting from bringing two devices closer together — we describe these cases in the behavior dimension [5]). For example, when a user moves a tablet next to a laptop running a development tool, a sidebar suggests three possible outcomes: moving the code editor to the tablet, moving tool palettes, or no action (Figure 7e). In this case, a default option (moving code editor, shown in green) is automatically selected when continuing the movement of the tablet in the direction of the laptop (Figure 7f). When the developer later moves a second tablet closer, options are again shown (Figure 7g), but no default action is available — the user decides to manually move the tool palettes to the second screen (Figure 7h).

**Proximity-dependent input redirection.** Aside from digital tablets, we can use other input/output devices with AirConstellations [P7]. When attaching a physical keyboard to an AirConstellations arm, the input of that keyboard can be re-directed to different devices based on proximity. For example, when moving the physical keyboard closer to a tablet, the onscreen soft keyboard fades out and the device instead uses the physical keyboard (Figure 7i-j). In this case, keyboard input can always be redirected to the closest tablet — or alternatively we can use spatial gestures like swiping in the direction of the tablet the keyboard input should be directed to.

**Spatially-aware soft-keyboard extensions.** When moving another screen close to the physical keyboard, this device will then function as an extended keyboard to show pre-sets of emojis (Figure 7k). This kind of virtual keyboard extension can depend on the application currently in focus and other context: for example, with a chat messenger app the keyboard would show emojis and GIF images, for a video editor pre-sets of command functions, and for a vector drawing application it could show a visual clipboard.

**Transfer portals with whiteboard sketching applications.** Close proximity between devices can open ad-hoc file transfer portals (similar to [66, 100]). When moving a tablet perpendicular to the large display (Figure 7l), a transfer portal proxy is shown on both on the tablet and the large display — establishing a direct
virtual transfer link, which allows to move sketches from the tablet to the wall display with a sliding gesture (Figure 7m-o).

5.3 Workspace Formations for Data Analysis
We illustrate the use of ad-hoc device federations in the context of data analysis and information visualization applications (e.g., similar to the distributed and coordinated visualization views in VisPorter [15], VisTiles [32], or ReticularSpaces [4]). In particular, we show how to use feedforward animations to facilitate forming device formations, and how to enable ad-hoc transitions from joint to personal, individual views.

Fast, reconfigurable multi-display workspace extensions. Similar to traditional multi-monitor desktop setups [30], users can reposition any of the AirConstellations tablets so that they function as an additional, ad-hoc screen extension [PT]. Placing tablets next to each other (Figure 8a) adds the display to the current formation arrangement — and updates the spatial configuration of the desktop layout. Users can then use touch gestures to extend content across the displays (like the high-pixel astrophysics image in Figure 8b). In our example, this enables multiple overview+detail views of different magnification levels (Figure 8c). The spatial awareness of AirConstellations means that physically rearranging devices does not require any manual reconfiguration of display arrangement in the system settings — and this lowers the barrier for more frequent physical reconconfigurations, even when needed for only a short amount of time.

Ad-hoc collaborative tabletop-like device formation. We use another feedforward technique — which we combine with slow motion feedback [97] techniques for previews and emphasizing change — to indicate opportunities for joint device formations (Figure 8d). In particular, we show a fluid wave/ripple animation (Figure 8e) when a user brings horizontally oriented tablets close to another to indicate...
Fast, reconfigurable multi-display workspace extensions

(a) Moving devices immediately updates multi-desktop environments. (b-d) allowing spatio-tactile gestures to distribute content across devices.

Ad-hoc collaborative tabletop-like device formations

(d) configuring ad-hoc tabletop formations with multiple tablets. (e-f) feedforward wave animation indicates possibility of joining tablets. (g) new formation allows cross-device touch gestures.

From joint to individual views of interactive data sets

(h-i) devices can transition from joint to individual views of data set by pulling away from formation. (j-k) allowing overview-detail interactions.

Figure 8: Multi-device formations workspaces for data analysis: (a-c) easily reconfigurable multi-display arrangements, (d-g) ad-hoc tabletop-like interactive surface with feedforward indicating possible merging of multiple tablet devices into a continuous surface, and (h-k) moving from shared to personal data views.

the possibility to merge these devices into a tabletop-like formation (moving from separate to continuous formation [P1]). The intensity of the wave animation increases when moving closer (in a non-linear mapping [S]). When both devices are very close, we fade-in a second animation showing an expanding arrow indicating the new formed merged view of the devices (Figure 8f). Both animations fade out once the surfaces are physically touching, and this new formation can then be used like a continuous interactive surface, for example, using cross-device pinch-to-zoom [66] (Figure 8g).

From joint to individual views of interactive data sets. Devices can transition from a joint device formation (similar to our previous example) to an individual, decoupled representation of the dataset. When analyzing a dataset on a horizontal multi-tablet formation (Figure 8h), a user can move one of the tablets apart from the formation, creating a separate, decoupled viewing window on the device (Figure 8i). Manually selecting parts of the dataset through direct touch on the shared view shows a magnified view of the data selection on the separate device (Figure 8j). The user can then directly interact with the dataset on the break-out device (Figure 8k).

Light-weight see-through portals across desktops. Layering devices [P3] on top of or in front of each other can facilitate OS-level operations across devices. The see-through desktop portal is activated when one device is directly placed on top of another device. When a person slides the device on top of another screen, the part of the occluded desktop is shown on the device in front (Figure 9a-b). The user can then drag&drop files or windows from one desktop to another (Figure 9c-d). We use the fact that devices are physically overlapping (instead of being positioned side-by-side, cf. [P1] and [P5]) as a unique trigger for this OS-level function.

Layering for cross-device window management and application migration. We can use proximity and orientation of tablets to facilitate window management across devices. Our system recognizes when a person moves a tablet in front of another (Figure 9e), occluding the device in the back (P3-occluding and [S]-front). This initiates the window-transfer view (Figure 9f): all application windows from the tablet in the back are shown semi-transparently on the tablet in front. A user can select one of the windows (Figure 9g) and pull the device closer to transfer the selected application to the tablet in front (Figure 9h). This example illustrates how we can use the fine-grained inter-device proximity and occlusion information to trigger a specific device operation (the window view), which is then confirmed through explicit user input (direct touch).

5.4 System-Wide Cross-Device Behaviors

We designed two additional techniques for system-wide cross-device application and window management: a window manager hand-over function and a see-through desktop portal for easy transfer of applications or files.
6 PRELIMINARY INTERVIEW STUDY

To gather feedback on AirConstellations, we conducted 1-hour interviews with eight participants (6 female, 2 male). We asked about their multi-device use and daily tasks, showed them a video with AirConstellations applications (similar to the video figure), collected their immediate impression on a scale from 1-negative to 5-positive, and followed up by gathering insights on the most and least compelling aspects.

Most valued scenarios and techniques. Participants tended to favor scenarios that reflected their most frequent activities. For example, P1, P6, and P7 spend a large portion of their day for data analysis, programming, and document editing, rating workspace and desktop management scenarios the highest. On the other hand, P2, P3, P4, P5, and P8 who use video meetings frequently rated remote collaboration scenarios as most compelling, relating to some of their needs when conducting remote collaboration. P3 commented that the multi-device video conferencing scenario “is really where the innovation is” and that the ad-hoc whiteboard technique during video calls as “super useful to share and express ideas with other designers.” P4 commented that assigning video feeds to different devices was “taking the physical [meeting configuration] and turning it into the digital” and that being able to look at participants seated at specific physical location was invaluable.

Balancing seeing vs. being seen. P2, P4, and P5 described tradeoffs they often made during video calls, having to optimize their setup for either better content viewing (seeing) or better webcam capture (being seen). P2 commented on the time spent to (re)adjusting her devices, angling some to avoid glare for content viewing, or for gaining a better angle for the webcam depending on lighting conditions throughout the day. P4 commented on the importance of eye contact during video calls but did not always switch webcams when switching devices (e.g., for screen sharing) for short periods of time as it was disruptive. Both noted that AirConstellations would directly facilitate these adjustments, and suggested that additional sensors could perceive lighting conditions or devices being interacted on, to provide fluid webcam switch: “posture, environment, ergonomics, orientation, source of lighting, all of that matters” (P2).

Organic setups and physical reconfigurability. Participants described AirConstellations as “mobile yet stable” (P1), and they see the potential of the proposed dynamic in-air device formations reducing the cost of making physical adjustments to their workspaces, thus enabling “more organic setups” (P6) evolving even over short periods of time. Related, P1 mentioned that AirConstellations afford “fine motor movements [screen angle adjustments] and broader types of movements [moving screens closer together].” P7 commented that such fluidity could transform the way she works with multiple devices: “It would make my life 100% more productive.” For his use of video conferencing, P1 commented that with AirConstellations he could transition to a more optimal multi-device configuration (migrating his chat window, the video feed, or the webcam), even for a short period of time, as it appears “effortless, [enabling] moving content without having to think.”

Replacing ‘clicks’ through physical device movement to facilitate transfers. All participants commented on the hurdles of migrating data or applications between their currently used devices: “today it requires a manual transfer of context [...] a bunch of clicks and a concerted effort to transfer content.” (P1). Bringing devices closer to share data and apps via desktop portal or transparent layer was compelling to all participants: “Extremely easy to use. It would feel like magic.” (P7). P1 highlighted that “a [single] physical interaction is more appealing than using a bunch of mouse and keyboard interactions,” and that proximity sensing “feels natural to move content [between devices].”

Unintentional interactions. A few participants raised issues with regard to interactions launched from context sensing. “There is a fine line to make it easy vs enabling them [the interactions] all the time. It should be part of the setup. I never want motion of my

---

*Backgrounds: 3 researchers, 4 designers, 1 data scientist.*
device unless I say it.” (P6). P2 echoed this concern by explaining that making some interactions “too easy” (by replacing a series of UI interaction with a single physical motion) would perhaps lead to unintentional application behaviors. For example, sharing a wrong document during a meeting or migrating a wrong app. These comments reinforce the importance of making interactions easily reversible, and of supporting effective visual feedforward and feedback.

Consistency of interaction behaviors. P7 pondered if the different behaviors are too numerous and prove somewhat inconsistent or hard to predict what would happen when moving a device. The concern that it could be disruptive led her to suggest that it might be better to select a small coherent and consistent subset.

Directness & immediate feedback. All participants externalized their delight, mentioning “super cool” (P5), “pretty neat” (P6), and “magical” (P7). They pointed out the fluid animations and direct feedback associated with physical interactions (e.g., making a window from device 1 pop up as device 2 approaches) as what made it delightful. They also responded enthusiastically to see-through/AR metaphors, such as the desktop portal and the flight simulator example: P5 commented that “Cool, I probably said that a lot, but it’s like the VR headset, without wearing the headset!”

Tradeoffs with armature design & cost. P2 commented that fixed external monitors on a desk were likely to be less “wobbly” but saw value in creating temporary setups, noting that it would be important for them to be lightweight and easily attachable on any surface. Other participants commented on the need to own multiple of these arms to unlock the power of the multi-device scenarios they saw: “Pretty neat, but the device arm device would have to be pretty cheap. A very low incremental expense.” (P6). P1 also commented on the need to have many arms in multiple places, to avoid having to move them too frequently: “If they were reasonably cheap and I would put them everywhere.”

7 DISCUSSION
Here we reflect on the potential of dynamic in-air device formations, discuss limitations of the approach (e.g., tradeoffs with spatial-awareness or proactive application behaviors), and suggest directions of future work.

Spatially-aware tracking vs. manual configuration. Beyond the affordance of our armatures for plug & play use as well as quick rearrangement, the integrated tracking system affords mutual spatial awareness between multiple armatures. This allows our interaction techniques to leverage the spatial setup of devices as an implicit parameter of actions that users take. Users can put displays where they want them, in a direct manner in-air, rather than indirectly configuring display layouts through control panels and so forth. This directness and simplicity, and the corresponding reduction in time and effort, may lower the threshold for users to reconfigure or adjust displays more frequently, whether for ergonomic, social, or task-driven needs. At the same time, spatially-aware tracking might not always be needed or desired, such as for use cases that do not require knowledge of the precise locations [36, 36]. The design space accommodates setups without integrated tracking, and a user could use instead a software-based setup similar to configuring monitor setups in system preferences today. Alternatively, techniques from previous work could serve to accomplish this in a usable manner (e.g., stitching [36] or in world-in-miniature configuration view that support drag&drop device placement [7]).

Transitions between fixed, semi-fixed, and mobile. For quick transitions with the fluid use of devices in either semi-fixed or fully mobile roles, the simple clamp-based attachment of AirConstellations facilitates devices that come and go to varying degrees. In the semi-fixed case, users can easily pull a device partially away from a formation to focus on particular details in a visualization (Figure 8) or pull aside someone in a small-group video conference (Figure 6e), for example. But users can “grab and go” (detach a device) to go fully mobile (as shown in our supplementary video) at a moment’s notice. As such, a tablet, netbook, or smartphone could serve as a logical hub to migrate device, task, and interactive state between different work or home-office locations, including other AirConstellations setups.

From single user to collaborative AirConstellations. For future work, exploring the use of ad-hoc dynamic device formations for co-located collaborative applications will be interesting. New device formations could support collaborative tasks — using shared devices and users’ own personal devices in concert — merging device formations and people formations (cf. Kendon’s f-formations [43, 67]). In addition, in-air device formations have the potential to best support people’s dynamic and highly flexible use of interpersonal space for interaction [31], akin to use in Proxemic Interaction [26] or spatially configured displays affecting people’s behavior [91].

Ecologies of devices and sustainability. AirConstellations provide an opportunity to contribute to sustainable computing due to the modular nature of their assembly. Since AirConstellations separates affordance of a particular device, tracking, and the actual interaction with the display into separate components, our design can easily incorporate older devices into formations. For example, one could upcycle older tablets or touch monitors to be used as part of a formation, yet retain all the functionality we have demonstrated in our scenarios to create individual ecologies optimized for compartmentalized tasks.

Tradeoffs of proactive vs. reactive application behaviors. Sensing the proximity, relative orientation, or adjacency of displays affords certain automatically sensed transitions in application states or behaviors. However, people’s natural socio-spatial literacy [49] implies that such device motions and configurations are also already part of the vocabulary of human-human interaction. This represents a potential design tension in AirConstellations: should a particular transition happen automatically, or not? Several of our interaction techniques show that hybrid designs may be possible, where proximity of devices triggers feedforward of possible connection states and other relationships between devices. Users then have options to react to this feedback, to act on suggested transitions suitable to the current application and device arrangement, or to default to “no change” if they simply ignore such suggestions. Alternatively, users could manually configure their preferences for proactive, proximity-dependent behaviors, or use non-spatially responsive AirConstellations designs.
Further use cases and hardware form factors. There is also an opportunity to explore other armature form factors (e.g., similar to custom haptic feedback arms [5]). In our approach, we re-purposed existing armatures as much as possible, but custom-built designs may increase stability, reach, or accuracy of tracking. Further exploration could include servo-driven actuated changes of device formations (e.g., extending actuated desktops [2] or interactive surfaces [83, 86]). Dynamic in-air device formations could also be applied to many other use cases, such as in virtual- or augmented-reality applications (e.g., as a form of flexible, dynamic interaction surfaces), education, or specialized domain expert workspaces.

**Extending design space.** Our design space is a first-order approximation of the spectrum of design parameters relevant for dynamic in-air device formations. But, as with any such mapping, there are possible extensions and additional dimensions to consider (similar to how we incorporated dimensions of earlier design spaces [8, 20, 35, 64, 79]). Such extensions could be, among many others, a fine-grained categorization of collaborative use cases, differentiating levels of actuation (e.g., with links to the field of robotics), or an in-depth unpacking of the roles and functions of devices as part of formations.

8 CONCLUSION

Using the AirConstellations platform we presented in this paper, we have explored the potential of dynamic in-air device formations for the next generation of multi-device workspace setups in a variety of use-cases. A core characteristic of all AirConstellations setups is their ability to support and retain the affordances of mobile interaction, while at the same time allowing semi-fixed devices to be (re-)configured in different physical arrangements. Our applications showcase the potential of such highly dynamic workspace configurations, with spatially-aware behaviors, feedforward, and other interaction techniques for flexible and delightful use of technology best supporting tasks at hand. The scenarios we demonstrated represent only a subset of the overall design space of dynamic in-air device formations, and we hope our work will inspire many promising research avenues to explore.

ACKNOWLEDGMENTS

We thank the Swiss Joint Research Center with Microsoft Research for funding Hugo Romat at ETH Zurich.

REFERENCES


