Affinity Lens

Data-Assisted Affinity Diagramming with Augmented Reality

HARIHARAN SUBRAMONYAM, University of Michigan
STEVEN M. DRUCKER, Microsoft Research
EYTAN ADAR, University of Michigan

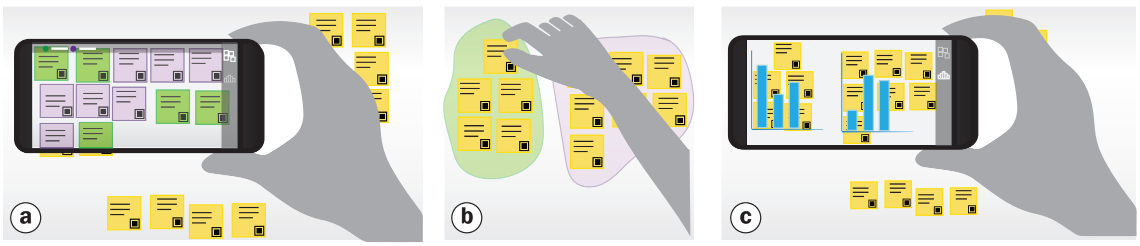

Fig. 1. Affinity Lens used to split a larger affinity cluster based on income level. (a) The user applies a heatmap lens to an existing cluster which shows two sub-groups. (b) The designer regroups the notes. (c) A histogram lens compares sleeping schedules for the two sub-clusters found in (a).

Despite the availability of software to support Affinity Diagramming (AD), practitioners still largely favor physical sticky-notes. Physical notes are easy to set-up, can be moved around in space and offer flexibility when clustering unstructured data. However, when working with mixed data sources such as surveys, designers often trade off the physicality of notes for analytical power. We propose Affinity Lens, a mobile-based augmented reality (AR) application for *Data-Assisted Affinity Diagramming (DAAD)*. Our application provides just-in-time quantitative insights overlaid on physical notes. Affinity Lens uses several different types of AR overlays (called lenses) to help users find specific notes, cluster information, and summarize insights from clusters. Through a formative study of AD users, we developed design principles for data-assisted AD and an initial collection of lenses. Based on our prototype, we find that Affinity Lens supports easy switching between qualitative and quantitative 'views' of data, without surrendering the lightweight benefits of existing AD practice.

CCS Concepts: • **Human-centered computing → HCI design and evaluation methods**; **Visual analytics**; *Mixed / augmented reality*; *Visualization systems and tools.*

Additional Key Words and Phrases: affinity diagrams; visual analytics; augmented reality

ACM Reference Format:

Hariharan Subramonyam, Steven M. Drucker, and Eytan Adar. 2019. Affinity Lens: Data-Assisted Affinity Diagramming with Augmented Reality. 1, 1 (April 2019), 18 pages. https://doi.org/10.1145/3290605.3300628

Authors' addresses: Hariharan Subramonyam, University of Michigan, Ann Arbor, MI, harihars@umich.edu; Steven M. Drucker, Microsoft Research, Redmond, WA, sdrucker@microsoft.com; Eytan Adar, University of Michigan, Ann Arbor, MI, eadar@umich.edu.

---

1 INTRODUCTION

Affinity Diagrams (AD) and related approaches are the method of choice for many designers and UX researchers. AD supports analysis and synthesis of interview notes, brainstorming, creating user personas, and evaluating interactive prototypes [24]. Notes can be placed on walls or surfaces in a way that leverages spatial cognition, offers flexibility in grouping and clustering, and then physically *persists*. Both individuals and groups can participate on large shared surfaces. AD users work to derive structure from inherently fuzzy and seemingly unstructured input. Though software tools have been implemented to emulate and significantly extend the AD experience [19, 46], many designers still favor the traditional, physical, 'sticky-note-on-wall' methodology [20].

While there are numerous advantages to the physical approach, it prevents the adaptation of AD practice for understanding data that is increasingly complex and mixed. By conducting an extensive literature search on past use of AD within HCI research, we found that in many cases (28 out of 47 papers) analysis also involved data from surveys [6, 12, 23, 26], sensor data [25], and interaction logs [10, 21, 31, 48]. In addition, our pilot interviews with industry practitioners revealed that they often bring their laptops to AD sessions in order to access quantitative data from spreadsheets or summary reports. In their current practice, designers look up quantitative insights that correspond to interview notes (e.g., interaction log data corresponding to "problem controlling music using voice") and make a note of them on the affinity wall (AD notes serve as "magnets for more details"). This approach is not only time consuming, but also problematic in that coherence between the analysis on the wall and the analysis on the screen is hard to maintain. Thus, the motivating question for our work is how we could expand AD for this new type of design process while at the same time supporting the physicality of the movable sticky-note?

By conducting a design probe with affinity diagramming users, we identified three main concerns: (1) the affordances of physical notes should be maintained, (2) additional data and insights should be easy to retrieve, and (3) data should be available just-in-time, without disrupting the primary diagramming practice. On this basis, we propose *Affinity Lens*, an augmented reality (AR) based tool for *Data-Assisted Affinity Diagramming (DAAD)*. Affinity Lens addresses these three concerns by leaving the physical notes in place while using the phone's camera and software to understand the note layout and to 'project' quantitative insights or overlay information on top of the notes and wall surface.

As a simple example, take a designer analyzing comments on a new IoT-based clock radio to determine which features to add. In addition to the text of the comments, the designer also has associated demographic information for each participant. The designer may begin with the comments as affinity notes, ending up with three clusters. The benefit of Affinity Lens becomes apparent when the designer starts looking for deeper patterns. For example, the designer decides to explore the implication of higher level incomes on the kinds of comments from users. By pointing the phone towards a cluster, the designer can easily identify notes from people with high and low incomes and separate them into two different clusters (Figure 1a). Once the new clusters are formed (Figure 1b), the designer can use the phone to look at distributions of sleeping schedules for each cluster (Figure 1c).

Affinity Lens is designed to play an *assistive* role. It allows the designer to maintain their existing (favored) work practice while at the same time offering on-demand analysis. In this sense, the process is backward compatible, both as documentation of an analysis effort and as a usable 'analysis artifact' that can be manipulated beyond the AR. Our key contributions are: identifying where data-assistance can augment AD; implementing a DAAD-focused system, Affinity Lens, which provides an array of extensible AR lenses; and validating, through two studies, that rather than disrupting AD, DAAD and Affinity Lens enriches the practice.

2 RELATED WORK

Affinity diagramming (also known as the KJ Method) has been used extensively for over 50 years [42]. AD supports organizing and making sense of unstructured qualitative data through a bottom-up process. A *schema* is developed by individuals, or groups, who arrange and cluster paper notes based on similarity of content, i.e., affinity. Because of its wide use, several projects have worked to address the shortcomings of the basic, 'pen-and-paper' use. These have centered around several areas including remote collaboration, clusters creation assistance, explicit and implicit search mechanisms, general visual analytics systems, and systems to bridge digital and paper documents. We briefly touch upon each area to set the context for the Affinity Lens project.

Collaboration: A number of studies worked to enhance the collaborative nature of affinity diagramming. Though some efforts focused on better-shared spaces (e.g., digital tables [27, 45]), others tackled the individual's role in a shared space by creating different private and shared views (e.g., [46]). These projects seek to enhance the collaborative experience and isolate areas where individual work can happen (likely leading to more diverse observations [14]). With Affinity Lens, we preserve the shared space by maintaining the majority work in the physical space. However, each participant can use their own device to support private analysis (currently we do not synchronize analyses). Affinity Lens can also track changes in the display (indicating what changed since last time) to support both the individual's work over a long period or for asynchronous collaboration.

Cluster creation: Exploration of how people organize information goes back several decades. Malone's early observations on physical organization [37] have been extended and adapted for digital interfaces. Tools for assisting in the creation of clusters have used everything from UI to ML techniques (e.g., [2, 13, 15, 32]). The general idea is that a user should be able to ask what cluster an individual item belongs to, or conversely, what items belong to a chosen cluster. The iVisClustering [35] work provides summaries of clusters including representative keywords and a cluster similarity view. While these have proven useful, the transformation of these object from paper to digital form has limited their widespread use. Though we do offer support for automatic clustering, our focus is enabling the end-user to drive this process. Put another way, Affinity Lens aids the sensemaking process [41] rather than attempting to automate it.

Explicit and Implicit Search: Several projects have explored simple aids for search. These include iCluster [15] and Note Finder [20] which support keyword-based search for matching cards. This capability has been implemented almost directly within Affinity Lens. However, as noted in this past work, this capability is insufficient to be useful on its own. Other efforts have used visual cards as jumping off points for pulling in additional information. Notably, the implicit search work of Dumais and colleagues (e.g., [16]), and the Scatter/Gather work [11] help take affinity diagramming from schematization into additional information gathering.

Visual Analytics Systems: Some prior work explored the notion of a spatial environment for more formal analytical tasks [47]. While completely digital, the notion was that notes could be linked with other notes and augmented with rapid grouping technique and analytical visualizations. The Jigsaw system extends these actions with a greater variety of support for quantitative analytics [43]. We incorporate lightweight, analytic summarizations in a similar style to both of these systems through specific summary lenses. Affinity Lens builds on other, related, visual analytic techniques including the set visualization techniques of [1], where set membership summary information is important to understand overall concepts and the interactive word clouds for summarizing coded text in grounded theory analysis [7].

Paper to digital transformation: Even with these many different directions of work, affinity diagramming in its classic form remains in frequent use due to the extremely low barrier for entry (i.e., sticky notes, pen, and a work surface). In Harboe et al.'s in-depth review of many of these tools [20], they arrive at the same conclusion that we do: instead of trying to replicate paper on screen, tools should offer ways to augment paper notes and support seamless integration between paper and digital worlds (e.g., [28, 29, 33, 34, 39]). The Affinity Note Finder prototype [22] explores one aspect: search. Issues of implementation (slow, heavy device, delay in responsiveness) were an issue, but the biggest concern was that keyword search alone was not sufficient for finding notes. This makes it clear that any single augmentation to the affinity diagramming process must work in conjunction with a constellation of desired activities. Affinity Lens expands that support to include other significant activities in the overall analytics process.

Other projects have explored the paper-digital divide in ways that seek to emulate the large-surface experience of AD. Some sought to bridge the gap by using touch-based interaction on tables and screen. For example, Affinity Table [19] attempts to replicate the look and feel of paper notes by providing natural inking and gestures on a digital display. The iCluster [15] system was implemented on top of a large interactive digital whiteboard. 'The Designer's Outpost' [33] of Klemmer et al. also uses sticky notes and an interactive whiteboard to support the transformation of physical to digital. When a sticky note is placed on to the whiteboard, it is scanned through cameras and subsequently manipulated digitally. The model for Affinity Lens is to preserve the note as a tangible object and virtually augment the information with overlays. That said, to support a number of lenses, Affinity Lens recognizes notes and tracks them in a virtual model.

There are a few additional UI interface metaphors that we build upon. The basic interaction metaphor, that of overlaying additional information and different representations on top of the existing material, draws heavily on the concept of the seminal Toolglass and Magic lens work of Bier et al. [5], as do many other augmented reality experiences. We heavily borrow on overlays and augmentation throughout the Affinity Lens user experience. We also use the concepts from Baudisch et al. [4] for helping give cues to the locations of notes that are currently off-screen.

3 A DESIGN PROBE FOR DAAD

To better understand the design space for data-assisted affinity diagramming we initiated an affinity diagramming exercise. The probe had participants work on an artificial task that contained textual comments augmented by associated quantitative data. Participants could also request analyses (in the form of printed visualizations) based on quantitative questions. These were produced by a study administrator who was present in the room with laptop and printer.

We recruited 10 participants who were either UX professionals or HCI-focused graduate students. They all had prior experience with AD, statistics, and data visualization. To encourage participants to think aloud and simulate a more realistic collaborative diagramming session, we had participants work in pairs (5 sessions). Each session lasted 75-90 minutes, and participants were compensated with $20 for their time. The high-level task had participants construct affinity clusters to answer a clustering task. After the subsequent implementation of Affinity Lens, we returned to this task with other groups using the working tool (Section 8).

Task and Dataset: We asked participants to analyze a dataset consisting of college students' food choices and cooking preferences using AD. The dataset included: descriptive summaries of a student's current diet, along with other behavioral and demographic attributes including how often they cooked, how often they ate outside, living arrangement, employment, family income, grade point average (GPA), body mass index (BMI), grade level, how often they exercised, marital status, and a self-rated health score on a scale of 1-10 (total of 11 variables) [40]. We selected sixty observations (rows) from the dataset, ensuring that there were plausible clusters in the set that were not too skewed (e.g., 55 people in one, five people in the other). We also ensured that the data varied on different dimensions to encourage the use of a combined analysis approach to form clusters. Each row was printed on a separate note and included an identifier, the text summary, and a table with responses to the 11 variables.

At the start of the study, participants were briefed about AD (though all were familiar with it) and introduced to the dataset and its attributes. They were instructed to cluster the students into six groups (with a maximum of 12 students in each group) such that each group could be assigned to one of six advertisements about food-related services based on their current diet. In addition, participants were provided with summary visualizations for all of the data attributes and were told that they could request additional visualizations on-the-fly based on note IDs. Although visualizations were produced as-requested, the study coordinator kept track of clusters being produced physically on the wall. This ensured that we could quickly generate requested visualizations for notes or clusters. Thus, participants could focus on AD rather than inputting clusters or learning a visualization package.

All sessions were video recorded, and the study coordinator made observational notes and prompted participants with clarifying questions about their clustering choices. At the end of the session, participants provided feedback through interviews. We analyzed the recordings, interviews, and final clusters from all five sessions. Broadly, we found that data-driven insights (i.e., quantitative analysis) supported decisions at all stages of the affinity diagramming workflow. More specifically, data informed a number of task-specific *decision points* for AD. These decision points can be grouped into four main 'assistance' categories: (1) detail access, (2) search, (3) clustering, and (4) summarization. Common AD tasks, such as identifying outliers, were often approached using multiple assistance categories. We provide details and examples for each below.

Detail assistance: A common task in AD is text *interpretation*. From this, topics can be extracted through heuristics to determine affinity. In cases where the text did not provide sufficient details (i.e., lacked clarity) or when interpreting text was hard, participants referred to data attributes to make inferences. For instance, one of the responses in the dataset was *"I eat 3000 - 4000 calories per day and … "*. Here, participants referred to BMI and exercise levels to disambiguate between an athlete with high caloric needs and someone who might be obese. As a consequence of accessing the quantitative data in relation to clustered subsets, participants began to find novel associations (e.g., responses that mentioned being busy were associated with employment or a living situation; and those who mentioned eating a high protein diet were associated with low BMI and exercise routines).

Search assistance: When a combination of data attributes was *perceived* as anomalous (e.g., a 4th-year student living on campus, or someone who eats healthy but has a low health score, etc.) participants attempted to look for other individuals with similar profiles. In cases where the combination was common, participants were able to generate new clusters. Alternatively, if no matches were found, the note was labeled as an outlier and set aside for later discussion. More specific to the text itself, participants regularly engaged in search and scan tasks to find notes that contained certain words or phrases (e.g., 'try,' 'high-protein,' 'diet').

Clustering assistance: Because text was 'primary' for AD, and thus more salient for the participants, many of the initial clusters were based on text. However, participants consulted data attributes for working with these starting clusters. A commonly observed pattern was using data to *split* larger clusters into smaller ones. Specifically, participants used the cluster level visualizations to determine if the cluster could be split along attribute values (e.g., 'always cooks' vs. 'never cooks'). For a smaller number of instances, participants used data similarity for *combining* smaller clusters. Visualizations were also used to *detect outliers* in clusters and notes were moved or marked for further analysis.

Summarization assistance: Participants used data in a number of ways to *validate* their clusters. This included simple visualizations to test the 'purity' of clusters. Participants often hypothesized, and would test, the idea that people with similar themes to their quotes would share other similar properties. The data-derived similarity 'assessments' would often be captured as cluster labels. Participants also used data to develop a narrative across different clusters. For example, participants utilized their cluster summaries to find that *"... freshmen who live on campus and tend to eat unhealthily, then they become sophomores and juniors and start cooking, seniors live off campus... [but] this one group of seniors live on campus and do not eat healthy... they never moved on"*.

4 DESIGN GUIDELINES

The probe sessions allowed us to identify key tasks for data assistance. These were used to drive many of Affinity Lens features. Additionally, we determined a set of guidelines both from observing the AD process and from feedback.

D1: Text first, then data. Affinity diagramming is at its most powerful when used for unstructured data, such as text. Datasets that are entirely structured are most often analyzed using other tools. AD, on the other hand, is suited to the bottom-up construction of clusters that requires human interpretation and input for clustering. Even in our probe, the two of five sessions that *began* clustering using data were less successful in completing tasks. They took a lot longer to analyze text within each cluster and to interpret how the text and data made sense as a whole. Because of this, Affinity Lens encourages end-users to start clusters based on analysis of text or other unstructured data. Though it would be relatively easy to implement, Affinity Lens does not, for example, suggest initial clusters.

D2: Support just-in-time insights. The type of data insights participants referred to during our study were highly context-driven and based on immediate decision support. Interactions to acquire such insights should be fast, expressive (support a variety of query and visualization needs), and low-effort, i.e., not distract from the primary task.

D3: Leverage spatial interactions for data access. Observing our participants we noticed extensive physicality to the AD process. Participants would move away and towards the wall to get different views. To understand the relationship between clusters (the broad view) they would often step away from the wall. To focus they would approach the wall and stand still (or seat themselves near the wall) to study individual clusters. A design guideline for Affinity Lens, and in part what motivated our use of AR through portable devices, was that the data could move with the AD practitioner and adapt to their spatial position and context. This is different, for example, from a large touchscreen that requires physical proximity for use.

D4: Offer automatic visual insights when possible. Though we encourage the text-first (D1) approach, this has the risk that practitioners over-focus and forget that other data is available. In our study, for example, we would occasionally 'probe' the participants to inquire if they required visualizations. It was rare in our experience that participants would remember to initiate a data request, but were responsive when probed. When presented with the data, participants found the information helpful and in most cases performed actions based on the data. Affinity Lens must balance a 'background' role with active help. To achieve this, Affinity Lens is designed to keep track of the state of the AD process (as much as possible) and to be ready with a set of automatically generated visualizations when called upon.

5 USER EXPERIENCE

Affinity Lens was built as a mobile (phone and tablet) application, with a companion desktop utility for note creation and for viewing captured analyses. As opposed to an always-on display such as a projector or screen, mobile devices can be turned off when not needed (D1) and can be easily moved around in space to support near and far interactions (D4). Figure 2 captures the four main regions of the mobile interface: the largest, is dedicated to the camera and visualization augmentation (a), a contextual menu occupies the right edge of the display (b) and dynamically changes depending on what is present in the camera's field of view, a data attribute menu at the bottom edge manages the configuration of the

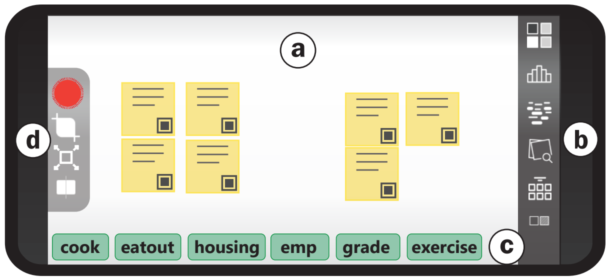

Fig. 2. Affinity Lens User Interface. (a) main camera view, (b) contextual lens selector, (c) lens configuration options, (d) lens modes

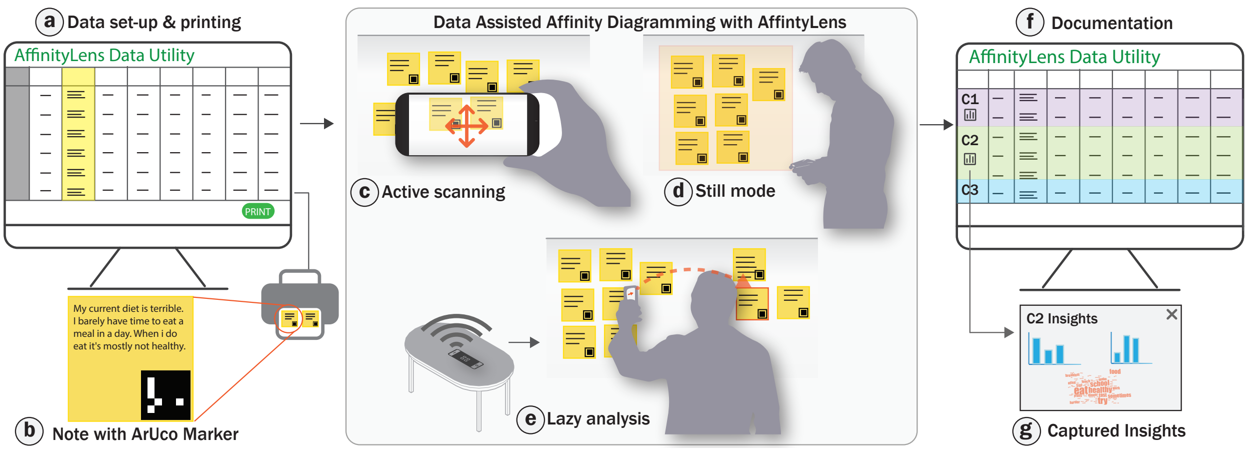

Fig. 3. Affinity Lens workflow. Data is acquired (a) and automatically tagged for a Marker (b) for printing. Various forms of DAAD (c, d, e) can be documented (f) along with associated insights (g).

current analysis tool (c), and dedicated controls allow setting modes of operation (d). In Affinity Lens, lenses are the collection of AR overlays available to the user. These include anything from visualization (e.g., bar charts based on what's in the display) to search (e.g., highlighting similar notes in the field of view). To better understand Affinity Lens' workflow (Figure 3) we follow a designer, Dave, as he uses DAAD to analyze the food choice dataset (this example is based on a combination of real use cases from our user studies).

5.1 Data and Notes Set-Up

Dave begins his analysis by loading survey responses he's collected into our desktop utility (Figure 3a). Each row corresponds to a different individual's response and each column is a question. From here, Dave selects the 'descriptive summary' column and issues a print command. Affinity Lens generates a unique AR marker for each row in the table which is printed along with the selected column value as affinity notes (Figure 3b). This 'binds' the printed note to the specific row. When using other data sources, such as interviews, Dave can import transcribed and coded notes from services such as nVivo, or even generate blank notes with markers and bind labels later using our lenses.

5.2 Clustering

Once the notes are printed, Dave lays them all out to begin the bottom-up clustering. He starts with a note that captures his attention: *"I try to eat healthy, but it doesn't always work out..."* He hypothesizes that this person may be unable to maintain a healthy diet, with planned, home-cooked meals, because they are busy. Dave picks up his phone with Affinity Lens, and *points* it at the note. Affinity Lens recognizes that only one note is in view, and augments the note using a lens that shows all attribute values (i.e., columns in the original CSV) associated with it (Figure 4 a). Here Dave sees that the student *eats out* most of the time, and also *works* a part-time job. He taps on those attributes to mark them as important to that text. Dave thinks that there may be other students with similar habits. He brings up the search lens and types in the keyword 'try' and then *pans* the phone over all notes (Figure 4 b). In the camera view of Affinity Lens, notes with the search term are highlighted in a different color. Dave gathers these notes as he finds them and piles them together for further clustering.

After forming a cluster of people which he labels 'tries but fails [to eat healthy],' Dave is interested in breaking it into smaller clusters. He brings up Affinity Lens and points it at the cluster. The view changes to offer a set of lenses that apply to note *clusters*. Dave is focused on this particular cluster, so he turns on the *still mode* (Figure 3 d) so he can continue working without pointing at the physical notes (D2, D3). Still mode captures a snapshot which persists in the display. He applies the heatmap lens by configuring different attributes, and sees that the cluster is split almost evenly by people who live on- and off-campus. Using this view Dave splits the cluster into two.

He sets the phone aside and continues working on clustering. Affinity Lens continues analysis in the background (Figure 3 e) and alerts him that all but one student in the on-campus sub-cluster are first years (D4). By tapping on the notification, and pointing it at the notes (guided by Affinity Lens' navigation augmentation), he sees a heatmap augmentation in which one student is a senior. He marks the student as an outlier and places the note away from that cluster.

5.3 Pruning and Sensemaking

After clustering all notes, Dave sees that there are two clusters which are labeled "healthy eaters," and "healthy eaters + specific diet." He hypothesizes that those with a specific diet are more physically active. To validate this, he places both clusters in Affinity Lens' frame. From the lenses menu, he selects the histogram lens and configures it for the exercise attribute. Affinity Lens overlays individual histograms on top of each cluster, where he can see that those with specific diets tend to exercise more than the other group. He also looks at the distribution of health scores and finds that both groups have a similar distribution of self-reported health scores. To look for other text-based differences, Dave augments the two clusters with word cloud visualizations. He sees that the most used word in the healthy eaters is 'balanced,' while the other cluster includes words such as *high protein* and *paleo*. He saves these insights with their associated note cluster through the Affinity Lens interface.

5.4 Documentation

Finally, Dave assigns labels to each clusters by using the label lens (Figure 4 f). Affinity Lens automatically updates the dataset with corresponding labels which can be viewed in real-time in the data utility tool (a web service viewable by Dave or others). Dave can use the labeled dataset for further analysis, or for recording the affinity outcomes. This feature also ensures that Dave has access to the saved visualizations he generated for each cluster.

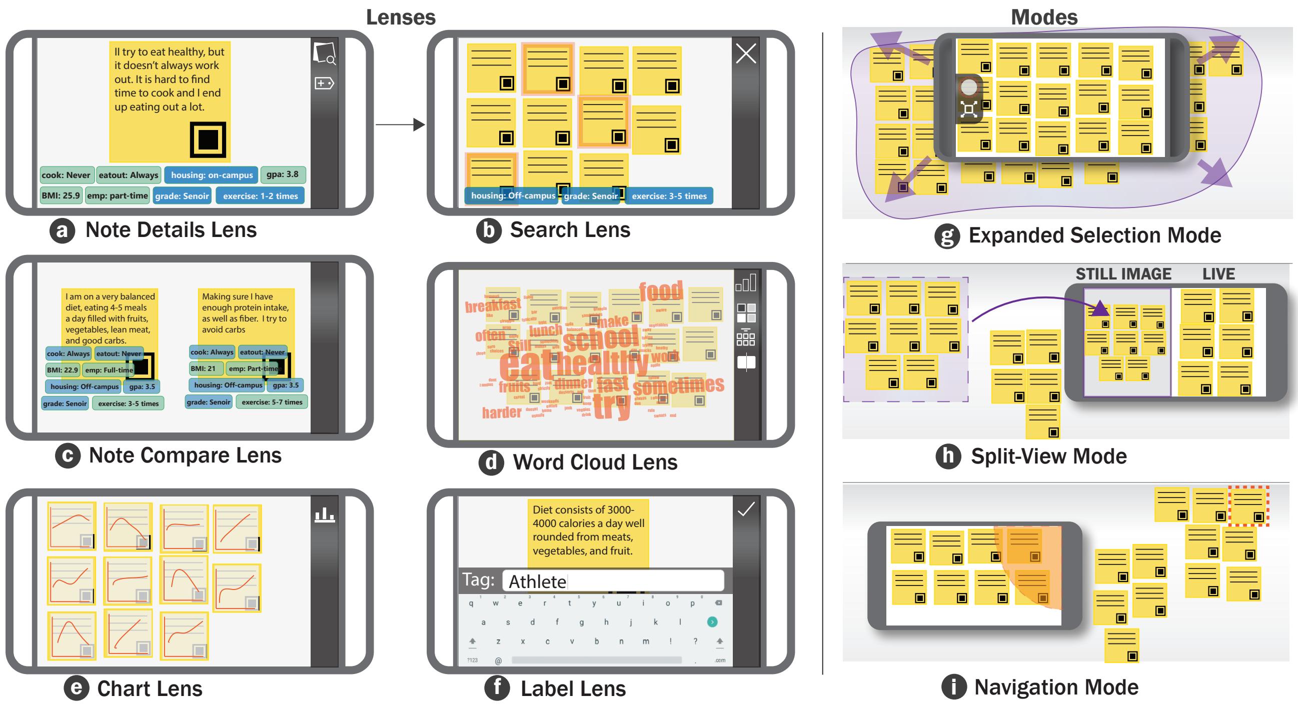

Fig. 4. A sampling of Affinity Lens AR Lenses

6 AFFINITY LENS(ES)

Affinity Lens allows users to choose among different lenses to overlay AR content on top of affinity notes. Here we describe the main categories and specific instances of lenses.

6.1 Lenses

For our prototype, we have implemented a number of lenses (examples in Figure 4) to support common tasks. These directly map to the four assistance types identified in our probe: details, search, clustering, and summarization. Affinity Lens is designed for extension so that new lenses can be added. In a practical scenario, users switch between different lenses as they engage in 'foraging' and sensemaking tasks.

Detail Lenses: In the context of mixed data, information contained on the physical note (i.e., the text) is only a *partial* view of the data. *Detail lenses* support understanding/interpreting the text by augmenting it with additional relevant information from the underlying data. In our implementation, when the end-user points at a single note, we augment that note with data values for that note (e.g., the row in the database). Other detail lenses, such as overlays of images [19] or videos, are possible with our architecture but not implemented in the prototype.

Search and Navigation Lenses: AD can have a large number of notes (as many as 200 – 500 [20]). An advantage of using a digital aid such as Affinity Lens is that it allows users to find notes based on user-defined queries. We have implemented two search lenses that allow *searching by text* phrases, and *searching by data* attribute values. In our pilot study, we found that designers did not seem to want 'generalized' search queries. Rather they wanted to find 'similar' notes based on what they were doing. Put another way, they wanted 'search-by-example.' To support this, our *search lens* can be launched from notes viewed through a *detail lens* (D2). For example, when the designer points at the note, they see the associated data for that note through the detail lens. From this view, they can select *values* as search criteria (thus launching the search lens). Query results are displayed by the search lens by highlighting matching notes. The mobile device can be panned over the wall's surface and the lenses will automatically adjust the AR overlays to match the current view. Because not all matches may be in the field of view (D4), 'hints' are offered to indicate matching offscreen notes in the style of Halo [4] (Figure 4i).

Clustering Lenses: The Affinity Lens prototype supports grouping and clustering through three lenses: (1) the *heatmap lens*, (2) the *note comparison lens*, and (3) the *cluster label lens*. The *heatmap lens* places an overlay on notes that uses color to encode a selected attribute and its values (Figure 1a). For example, we might select 'weight' as an attribute and all notes will be color coded from light to dark based on the weight value associated with that note. This form of augmentation acts to summarize but also directly supports decisions around splitting and grouping multiple clusters. For a pair of notes, the *note comparison lens* (Figure 4c) displays those data values that are the same and those that are different (a weak representation of affinity). Finally, the *cluster label lens* is used to 'tag' all notes in a cluster with a persistent label.

Summarization Lenses: The final set of lenses allow end-users to summarize insights about clusters. This is done largely through the use of visualization overlays. In addition to the heatmap lens, our prototype also provides a *histogram lens*, a *wordcloud lens*, and a *radar plot lens*. The histogram lens will generate a histogram bar chart based on some selected attribute (e.g., the number or fraction of people who said 'yes' to dieting in a cluster versus 'no'). Clusters can be explicit (i.e., the designer tagged a cluster) or can be dynamic and contextual based on the notes in the field of view. The resulting histogram is placed over the entire field of view. When looking at text, a wordcloud lens (Figure 4d) will generate an overlay of common words (sized by frequency) on top of the notes. A radar lens will produce a radar plot to summarize multiple quantitative variables simultaneously. When multiple clusters are in view, or the designer uses a split view to see two clusters side by side, summarization lenses will be applied to each cluster separately (e.g., two or more histograms will be overlayed).

6.2 Interactive Querying through Scene Specification

In Affinity Lens, the primary mode of interaction is by first selecting the lens (and potential parameters on the mobile device's screen) and then viewing the physical notes through the phone's display. The subset of notes in the view provides a natural scope for the query (D3). The user can either use Affinity Lens in *live mode*, where the display updates based on the camera's field of view, or in *still mode* which uses a static snapshot. In live mode lenses dynamically adapt as the user pans across the surface. In still mode, the user can easily switch between multiple lenses and apply them to the notes captured in the view. This can be significantly more comfortable than continuously holding the phone in mid-air and also allows for 'private' analysis in a shared setting. To support analysis of large clusters, we provide an expanded selection mode. The mode will cause Affinity Lens to include off-screen notes, that were labeled as belonging to the cluster, in any analysis (e.g., a histogram) (Figure 4g).

In either *live* or *still* mode, the user has the option to 'split' the view (Figure 4h). This permits comparison between different clusters that are physically distant. It also allows for an overview-plus-context view where one side of the screen can be used to drill down into details for notes or clusters contained on the other side of the screen.

Finally, Affinity Lens supports what we call *lazy interactions*. Affinity Lens leverages periods of inactivity to analyze data and generate potential clusters and other insight recommendations such as outliers. When a new insight is available, the phone displays a notification to the user about the insight along with details about the target set of notes. The user can then tap on the insight and use guided navigation to find the physical notes on the wall. For example, if Affinity Lens detects an outlier in a particular cluster when the notification is selected, arrows will lead the user in *live mode* first to the cluster and then to the highlighted outlier.

7 SYSTEM ARCHITECTURE

While complete details of our implementation are beyond the scope of this paper, we provide a high-level view of the architecture. As shown in Figure 5, Affinity Lens is comprised of five main components: (1) Scene Analyzer, (2) Lens Controller, (3) Dynamic View Configurator, (4) lenses, and (5) the Data Access and Analytics Module.

The *Scene Analyzer* detects notes from the incoming camera feed (i.e., the scene) by using computer vision based processing. Note information including the number of notes and positions are relayed to the *Lens Controller*. This module determines candidate lenses based on notes and updates the phone interface through the *Dynamic View Configurator*. Once a lens is selected and applied (either the system default or by end-user selection), the system generates a database query for the notes in view for execution by the *Analytics Module*. Finally, query results are rendered on top of the scene by the View Configurator. This process happens continuously and in-sync with the camera feed. The system itself is implemented using JavaScript and is executed (and displayed) in the browser on the phone or tablet device.

7.1 Scene Analyzer

Our current prototype uses ArUco Markers [18] for detecting notes along the x-y plane. Using computer vision libraries [9, 38], this module determines marker positions and builds spatial relationships between notes. The scene analyzer overlays a grid structure on top of the markers, and each marker is assigned a row and column position relative to the scene. This information is also used to detect clusters in which individual clusters are separated by areas of empty grid cells. In each refresh cycle of the scene, notes are updated with revised x and y positions along with marker IDs for eight adjacent markers (to support navigation), and cluster ID. This information is used by other modules in the system pipeline.

7.2 Lens Controller

This module consists of a collection of lenses, along with a look-up table containing prerequisites and configuration parameters. Depending on the number of notes or clusters in the scene (single, pair, multiple, etc.), the lens controller will select all applicable lenses and send configuration information to the Dynamic View Controller. If the mode corresponds to a single lens, the controller also instantiates the detail lens. This module also coordinates different lenses by passing relevant setting and parameters between them (e.g., maintaining attribute selection between lenses, setting selected attribute values such as search parameters, etc.).

7.3 Dynamic View Configurator

The Configurator updates the Affinity Lens interface in real time based on input from the lens controller. Candidate lenses are presented as items on the right contextual menu. When a lens is selected, appropriate configuration attributes are rendered at the bottom of the screen. When the end-user interacts with these menu options, this module also relays events and selections back to the lens controller. Once a lens is selected, this module applies the output of the lens and displays the augmented view on the screen.

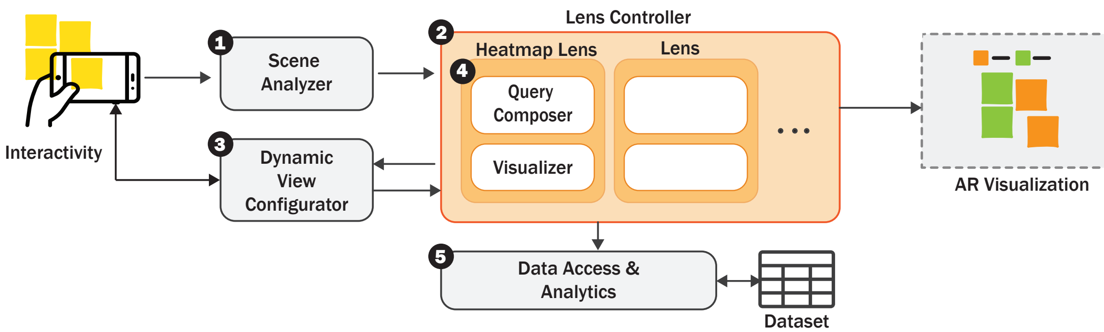

Fig. 5. System Architecture. (1) Scene analyzer extracts notes from camera feed, (2) lens controller determines set of lenses applicable to notes in view, (3) dynamic view configurator updates the interface with available lenses, (4) lens queries for data from the (5) Data access and analytics module, and renders the augmented visualization.

7.4 Lens Design

Each lens is made up of two sub-components: a *query-builder* and the *visualizer*. The query builder constructs a query for the notes in the view along with other lens specific configurations (e.g., selected attribute). For example, the histogram lens will identify that a cluster of notes is currently in view and query the database for the values for *those notes* based on the attribute the end-user has selected. This query is processed by the Data Access Module. For example, when a histogram is requested over a set of ten notes, with 'living preference' as the data attribute, the query builder fires a query by passing note IDs and living preference as conditional clauses. The results are rendered by the visualizer sub-component. This module makes use of positioning information made available by the scene analyzer to determine the placement of the rendered visualization. This abstraction allows us to easily build new lenses through a standard API.

7.5 Data Access and Analytics

This module supports two types of data operations. It executes query requests issued by the lenses over the dataset and updates the dataset based on real-world actions (e.g., if a new cluster is formed and detected, the associated rows in the database are labeled with a cluster ID).

The module also supports lazy-analysis interaction. Based on note adjacency and clustering information provided by the Scene Analyzer, background clustering and analysis are executed and results are surfaced back to various lenses. For example, to support clustering, we use the techniques developed in the iCluster work [3]. Existing cluster information is used to create a metric space by which clusters are formed. Distances between notes are based on a combination of attributes and keywords. Weights on attributes are adjusted such that notes belonging to the same cluster are deemed closer together while notes in different clusters are further apart. If there are sufficient notes in each cluster, a classifier can be trained to help decide which cluster a note belongs to. Using this information, possible clusters can be highlighted for a selected note. Alternatively, if a cluster is selected, matching unclustered notes can be highlighted.

7.6 Implementation

We implemented Affinity Lens as a mobile web application that runs on the phone browser. A Node.js server handles data analytics and image storage, and a HTML/JavaScript client uses OpenCV.js and js-ArUco libraries for camera and image processing and D3.js for visualization.

8 EVALUATION

To evaluate Affinity Lens, we conducted two different in-lab AD studies. The first was a controlled study (using the same protocol as in section 3) in which we determined whether end-users could effectively generate data insights using Affinity Lens. In the second study, which was open-ended, we aimed to evaluate Affinity Lens in a realistic AD workflow.

8.1 Study 1: Controlled Evaluation

For this study, we conducted three 90-minute sessions (two participants per session) with four HCI design student (P1-P4) and two UX professionals (P5-P6). We used the same task and study protocol as in section 3, but instead of having the data directly printed on the notes, we added an ArUco marker to bind the note to a data row. To encourage discussion between participants (for think-aloud), we only provided a single Android mobile device (5.5. inches,1440 x 2560 pixels) with Affinity Lens running on the Chrome browser.

At the start of the session, participants were given a hands-on demo of the system including the use of different lenses. Once participants indicated familiarity with the system, they proceeded to analyze and cluster the notes for the advertising task. Sessions were video recorded for analysis, and a study coordinator took observational notes. At the end of the session, participants did a verbal walk-through of the distinct characteristics of each cluster and finally took part in an informal interview to report their experience.

Findings

Data assistance for clustering notes: Across all sessions, we observed that participants effectively invoked different lenses to generate data overlays for single, and group of notes (D2). While reading a note, if participants noticed an interesting phrase, or when there was disagreement about which cluster to place the note in, they would invoke the details overlay on that note. Beyond note level details, participants also made use of data overlays to revise initial clusters generated from text. A repeated pattern we observed was that participants cycled through different data attributes using the heatmap lens to split a clusters, *update* cluster labels, or make distinctions across different clusters.

A common practice in AD is to set aside notes that do not fit into any clusters for further analysis. For such notes, participants took a trial-and-error approach by placing the note being discussed next to notes in other clusters to test for "affinity" using the note-compare overlay. Once clusters were generated, participants used both the histogram and heatmap overlays for validating cluster affinity and outlier detection (D4). They often expressed delight when their text-based interpretations matched what the data showed. However, participants reported that they did not find the wordcloud lens very useful. We suspect this is because of the smaller number of notes used in this study. Further, we only observed a few instances of multiple-cluster comparison. This may be attributed to the fact that data level bins were already determined when clustering.

In all sessions, while the clusters aligned with our artificial grouping, we observed that overall engagement with Affinity Lens was *higher* than we had intended (i.e., somewhat a violation of D1). This may be due to the nature of the clustering task which required data insights, but more likely the novelty of the system. As reported by P2: *"I was relying* *too much on the app . . . not using the notes as much"*, and P1:*"it (system) is fun . . . especially when you don't know how to group something (using text)"*.

User Experience with Affinity Lens: The portable nature of our solution made it easy to blend spatial interactions with our lenses interface (D3). In one of the sessions (P1-P2), participants spread the notes on the table, and sorted the notes by using the heatmap lens. When discussing cluster level insights, participants found the *still-mode* extremely useful. We observed that one of the participants would capture cluster insights and engage in rich discussion with the other participant by trying out different lenses (D3). Participants also found the *split-view* mode helpful when comparing distant clusters, and appreciated that they did not have to move clusters around to make comparisons.

During the feedback session, all participants reported that the concept of lenses, and Affinity Lens' interface was easy to understand and use. When explicitly asked about the ArUco markers, participants indicated familiarity with QR codes, and that the markers did not interfere with AD. We note that in some instances, Affinity Lens did not recognize the markers. For example, distance was an issue when multiple clusters were in view. This issue can likely be remedied by implementing image enhancement techniques (e.g., [44]).

Finally, in comparison to our probe session, in which data persisted on notes along with text, the AR features of Affinity Lens made it possible to make salient (bring to view) specific types of details, on demand. Participants were able to easily toggle between text and data views, and compare insights across clusters in a fast and fluid manner. A drawback is that data insights are not persistent, which can be problematic when working with larger datasets. As mentioned by one participant (P5), persisting data-specific insights on paper might be useful. They even recommended having colored markers corresponding to the heatmap color palette, and adding indicators on physical notes (they referred to clusters by colors: "these are the reds, add them to the purple cluster").

8.2 Study 2: Open-ended AD Workflow Evaluation

To emulate a realistic workflow as described in section 5, we gave participants the results of a survey we conducted about Facebook Usage and Privacy using Qualtrics. The survey consisted of closed-ended questions about Facebook Usage, Ads on Facebook, types of data shared (posts, pictures, profile information, etc.), concerns about privacy and data sharing, and an open-ended question requesting examples of privacy violation on Facebook. All questions were required, and we set a minimum limit of 250 characters for the open-ended question. We collected 100 responses using Amazon's Mechanical Turk and generated the notes by exporting the data as a text (CSV) file from Qualtrics.

We recruited six participants with prior experience in conducting AD: three UX professionals (P7-P9), one design-science researcher (P10), and two privacy researchers (P11-P12). We conducted three sessions with pairs of participants, and each session lasted 2-hours. Participants were paid $30 for their time. In each session, we described the survey questions to the participants and asked them to generate sources for privacy violation using AD. We then provided a guided tutorial of the system. We concluded each session with a walkthrough of the clusters and an informal interview. In this study, we provided participants with multiple device options (phone, and tablets with 12.3-inch screen, 2736 x 1824 pixels) all running Affinity Lens on the Chrome browser.

Findings

Data-assisted, not data-driven clustering: In all our sessions, we observed participants trying to navigate when to use data versus text views. At the start of each session, one of the participants wanted to start with the data view, while the other preferred generating an initial set of clusters based on text (P11: *"open-ended responses are more reliable . . . we can use our judgment to categorize them first and then use [Affinity Lens] to double check"*). The rationale for data-first was that being able to quickly try out different groupings with data would help ask more questions earlier on in the process, as P9 mentioned *"rather than using the lenses to drill-down, I wanted to use it as a way to bubble-up questions."*

While data overlays offered a quicker alternative to generate clusters (P7: *"we started with the obvious and it was massive… we realized we need to get out of reading the content and look at the data"*, P8: *"… with all the ad tracking we wanted to hack for a trend,"*), participants realized that over-reliance on data could make it hard to make sense of text content within individual clusters. The switch from data-view back to content occurred when participants became aware that they devalued content, or when there were no discernible patterns from data. In summary, participants saw value in having both views, and being able to switch between them (e.g., P11: *"[Affinity Lens] enhanced the depth of analysis and helped us figure out what is going on, the nuances… "*).

Time costs for DAAD: When using DAAD, we hypothesized that Affinity Lens would speed up the AD process. Across all sessions, we observed variations in when, and for how long, participants engaged with Affinity Lens. In session 1, the use of Affinity Lens (i.e., data view) was more evenly spaced out. The first use was at 14.5 minutes into the session, followed by switching between text and data views every 10-12 minutes. In sessions 2 and 3, participants first used Affinity Lens after around 40 minutes of clustering by note content but extensively used Affinity Lens for pruning and sensemaking during the second half of the session.

Some participants felt that they spent *more* time on AD because the insights from data were interesting (e.g., P7: *"If I had just words I would have been like, yeah, that is all we are going to get … [with Affinity Lens] I could keep going on looking for new patterns"*). In this study, because participants were not directly involved in survey design, some participants found the range of attributes overwhelming (we utilized a very broad survey instrument). P8 suggested different tabs to categorize the attributes (e.g., demographics tab, Facebook usage tab, etc.) but added that if they were using in their own work, this may not be a problem.

DAAD in existing design workflows: In discussing applicability of DAAD in their own design process, several participants were keen on using Affinity Lens as a way of getting "buy-in" from managers and data analysts. For example P7:*"not everybody buys into AD and Affinity Lens is a nice vis bank … "*, P9: *"I could advocate for the realness of my work… "*, etc. While all participants agreed that quantitative data was not the place to start AD clustering (confirming D1), participants mentioned that data insights from AD could generate an initial set of hypothesis for data analysts. During feedback, participants also recounted examples from their own experiences of working with mixed methods approaches, and how Affinity Lens could have helped in those situations. For example, P4 mentioned conducting AD exercise with data collected from a photo diary, and that having Affinity Lens could have helped augment pre- and post-study information and metadata (e.g., timestamp).

In summary, the results from our study demonstrate the usefulness of Affinity Lens in the AD workflow. Though we expect that testing Affinity Lens in additional contexts will lead to more features and improvements, the feedback we received from our participants, and their interactions with Affinity Lens, is highly encouraging.

9 DISCUSSION AND FUTURE WORK

There is clearly a need for integrated sensemaking from qualitative and quantitative data when conducting mixed-methods research. Through Affinity Lens's AR overlays, we demonstrated how DAAD can enrich the analysis experience of survey data, a typical use-case within HCI research. Beyond surveys, HCI work also uses interaction logs, sensor streams, and multimedia content (photos/videos) to inform system design and end-user behavior. Current workflows for analyzing such data typically follow a unidirectional pipeline (e.g., video footage –> transcripts –> grounded theory coding), making it hard to flexibly combine raw data with qualitative insights in a just-in-time manner. Future work can look at ways to incorporate DAAD into existing workflows by linking lenses with rich data sources (e.g., [36]). For example, one can augment the text from think-aloud transcripts with interaction logs showing mouse clicks data, or overlay raw video footage of actual task execution for multiple participants (affinity notes) in parallel.

In our current implementation of DAAD, we do not focus on the collaborative nature of AD, or potential collaboration between qualitative and quantitative analysts. However, we believe there is an opportunity for more collaboration-focused lenses. For example, we can imagine sharing configured lenses between devices to enable different users to study different parts of the wall with the same lens. Further, in Affinity Lens we primarily support just-in-time insights with minimal query specification (D2). To emphasize the assistive role of data, and given the form factor, we did not explore all features of a data analytics tool such as Tableau or R in DAAD. However, based on participant feedback it may be desirable to have richer support for data analysis within DAAD to enable collaboration between designers and analysts. Building on prior work on spatial [2], and tangible visualizations [17, 30], we are exploring ways to leverage sticky-notes for querying and visualization specification.

In our studies, we printed notes on plain paper. This requires the added effort of cutting and adding adhesive. In real world deployment, this limitation can be easily overcome by either using a template based printing technique (i.e., pasting sticky notes on letter size paper template before printing) or by using special portable printers such as [8]. Lastly, camera resolution and field-of-view (FoV) constrain scalability when there are a large number of notes. This creates a challenge for using the phone for maintaining the system's internal models of the physical AD. Affinity Lens currently updates note positions by surreptitiously capturing frames when the user pans the phone during use. Future work can explore other active interactions to maintain this representation (e.g., prompting the end-user to explicitly capture "current state" by scanning across the wall). By open sourcing our implementation, we hope that we can better understand how these features are used and enhanced.

10 CONCLUSION

Affinity diagrams are used throughout academic and business communities as part of the design process. However, as designers are increasingly working with sources of information that consist of both qualitative and quantitative data, they often desire analytical power beyond physical sticky notes. Prior research to address these shortcoming have posed barriers including prohibitive costs of large, interactive whiteboard systems or disruptions of current workflow practices. With Affinity Lens, we have demonstrated how data-assisted affinity diagrams can be implemented with low-cost, mobile devices while maintaining the lightweight benefits of existing AD practice. To date, we have only lightly explored the space of lenses, but already, users of the current system were enthusiastic about using Affinity Lens in their current AD-related work tasks.

11 ACKNOWLEDGMENTS

We thank the anonymous reviewers and our study participants for their time and helpful feedback. We also thank Linfeng Li and Xiaochuan Kou for their help with the video.

REFERENCES

[1] Bilal Alsallakh, Luana Micallef, Wolfgang Aigner, Helwig Hauser, Silvia Miksch, and Peter Rodgers. 2016. The State-of-the-Art of Set Visualization. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 234–260.

[2] Christopher Andrews, Alex Endert, Beth Yost, and Chris North. 2011. Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. *Information Visualization* 10, 4 (2011), 341–355.

[3] Sumit Basu, Danyel Fisher, Steven M Drucker, and Hao Lu. 2010. Assisting Users with Clustering Tasks by Combining Metric Learning and Classification.. In *AAAI*.

[4] Patrick Baudisch and Ruth Rosenholtz. 2003. Halo: a technique for visualizing off-screen objects. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 481–488.

[5] Eric A Bier, Maureen C Stone, Ken Pier, William Buxton, and Tony D DeRose. 1993. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, 73–80.

[6] Erin Brady, Meredith Ringel Morris, Yu Zhong, Samuel White, and Jeffrey P Bigham. 2013. Visual challenges in the everyday lives of blind people. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2117–2126.

[7] Senthil Chandrasegaran, Sriram Karthik Badam, Lorraine Kisselburgh, Karthik Ramani, and Niklas Elmqvist. 2017. Integrating visual analytics support for grounded theory practice in qualitative text analysis. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 201–212.

[8] Mangoslab Co. 2018. Nemonic Mini Printer. http://www.mangoslab.com/n/nemonic/?lang=en

[9] Intel Corporation. 2018. Open CV Library. https://docs.opencv.org/3.4.1/index.html

[10] Yanqing Cui, Jari Kangas, Jukka Holm, and Guido Grassel. 2013. Front-camera video recordings as emotion responses to mobile photos shared within close-knit groups. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 981–990.

[11] Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. 2017. Scatter/gather: A cluster-based approach to browsing large document collections. In *ACM SIGIR Forum*, Vol. 51. ACM, 148–159.

[12] David Dearman and Khai N Truong. 2010. Why users of yahoo!: answers do not answer questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 329–332.

[13] Marie Desjardins, James MacGlashan, and Julia Ferraioli. 2007. Interactive visual clustering. In *Proceedings of the 12th international conference on Intelligent user interfaces*. ACM, 361–364.

[14] Steven P. Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. 2012. *Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased Self-efficacy.* Springer Berlin Heidelberg, Berlin, Heidelberg, 127–153. https://doi.org/10.1007/978-3-642-21643-5_8

[15] Steven M Drucker, Danyel Fisher, and Sumit Basu. 2011. Helping users sort faster with adaptive machine learning recommendations. In *IFIP Conference on Human-Computer Interaction*. Springer, 187–203.

[16] Susan Dumais, Edward Cutrell, Raman Sarin, and Eric Horvitz. 2004. Implicit Queries (IQ) for Contextualized Search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '04)*. ACM, New York, NY, USA, 594–594. https://doi.org/10.1145/1008992.1009137

[17] Johannes Fuchs, Roman Rädle, Dominik Sacha, Fabian Fischer, and Andreas Stoffel. 2013. Collaborative data analysis with smart tangible devices. In *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 90170C–90170C.

[18] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292.

[19] Florian Geyer, Ulrike Pfeil, Jochen Budzinski, Anita Höchtl, and Harald Reiterer. 2011. Affinitytable-a hybrid surface for supporting affinity diagramming. In *IFIP Conference on Human-Computer Interaction*. Springer, 477–484.

[20] Gunnar Harboe and Elaine M Huang. 2015. Real-world affinity diagramming practices: Bridging the paper-digital gap. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 95–104.

[21] Gunnar Harboe, Crysta J Metcalf, Frank Bentley, Joe Tullio, Noel Massey, and Guy Romano. 2008. Ambient social tv: drawing people into a shared experience. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1–10.

[22] Gunnar Harboe, Jonas Minke, Ioana Ilea, and Elaine M. Huang. 2012. Computer Support for Collaborative Data Analysis: Augmenting Paper Affinity Diagrams. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12)*. ACM, New York, NY, USA, 1179–1182. https://doi.org/10.1145/2145204.2145379

[23] Chris Harrison, John Horstman, Gary Hsieh, and Scott Hudson. 2012. Unlocking the expressivity of point lights. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1683–1692.

[24] Rex Hartson and Pardha S Pyla. 2012. *The UX Book: Process and guidelines for ensuring a quality user experience.* Elsevier.

[25] Elaine M Huang, Gunnar Harboe, Joe Tullio, Ashley Novak, Noel Massey, Crysta J Metcalf, and Guy Romano. 2009. Of social television comes home: a field study of communication choices and practices in tv-based text and voice chat. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 585–594.

[26] Elaine M Huang and Khai N Truong. 2008. Breaking the disposable technology paradigm: opportunities for sustainable interaction design for mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 323–332.

[27] Petra Isenberg and Danyel Fisher. 2009. Collaborative Brushing and Linking for Co-located Visual Analytics of Document Collections. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 1031–1038.

[28] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, 234–241.

[29] Robert JK Jacob, Hiroshi Ishii, Gian Pangaro, and James Patten. 2002. A tangible interface for organizing information using a grid. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 339–346.

[30] Seokhee Jeon, Jane Hwang, Gerard J Kim, and Mark Billinghurst. 2006. Interaction techniques in large display environments using hand-held devices. In *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM, 100–103.

[31] Tero Jokela and Andrés Lucero. 2013. A comparative evaluation of touch-based methods to bind mobile devices for collaborative interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3355–3364.

[32] William P Jones and Susan T Dumais. 1986. The spatial metaphor for user interfaces: experimental tests of reference by location versus name. *ACM Transactions on Information Systems (TOIS)* 4, 1 (1986), 42–63.

[33] Scott Klemmer, Mark W Newman, and Raecine Sapien. 2000. The designer's outpost: a task-centered tangible interface for web site information design. In *CHI'00 extended abstracts on Human factors in computing systems*. ACM, 333–334.

[34] Beth M Lange, Mark A Jones, and James L Meyers. 1998. Insight lab: an immersive team environment linking paper, displays, and data. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 550–557.

[35] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. 2012. iVisClustering: An interactive visual document clustering via topic modeling. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1155–1164.

[36] Zhicheng Liu, Bernard Kerr, Mira Dontcheva, Justin Grover, Matthew Hoffman, and Alan Wilson. 2017. CoreFlow: Extracting and Visualizing Branching Patterns from Event Sequences. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 527–538.

[37] Thomas W. Malone. 1983. How Do People Organize Their Desks?: Implications for the Design of Office Information Systems. *ACM Trans. Inf. Syst.* 1, 1 (Jan. 1983), 99–112. https://doi.org/10.1145/357423.357430

[38] Juan Mellado. 2018. ArUco JavaScript. https://github.com/jcmellado/js-aruco

[39] Thomas P Moran, Eric Saund, William Van Melle, Anuj U Gujar, Kenneth P Fishkin, and Beverly L Harrison. 1999. Design and technology for Collaborage: collaborative collages of information on physical walls. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*. ACM, 197–206.

[40] Bora Pajo. 2017. Food choices: College students' food and cooking preferences. hhttps://www.kaggle.com/borapajo/food-choices.

[41] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, Vol. 5. 2–4.

[42] Raymond Scupin. 1997. The KJ method: A technique for analyzing data derived from Japanese ethnology. *Human organization* 56, 2 (1997), 233–237.

[43] John Stasko, Carsten Görg, and Zhicheng Liu. 2008. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization* 7, 2 (2008), 118–132.

[44] Drew Steedly, Chris Pal, and Richard Szeliski. 2005. Efficiently Registering Video into Panoramic Mosaics. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2 (ICCV '05)*. IEEE Computer Society, Washington, DC, USA, 1300–1307. https://doi.org/10.1109/ICCV.2005.86

[45] Edward Tse, Saul Greenberg, Chia Shen, Clifton Forlines, and Ryo Kodama. 2008. Exploring true multi-user multimodal interaction over a digital table. In *Proceedings of the 7th ACM conference on Designing interactive systems*. ACM, 109–118.

[46] William Widjaja, Keito Yoshii, Kiyokazu Haga, and Makoto Takahashi. 2013. Discusys: Multiple user real-time digital sticky-note affinity-diagram brainstorming system. *Procedia Computer Science* 22 (2013), 113–122.

[47] William Wright, David Schroh, Pascale Proulx, Alex Skaburskis, and Brian Cort. 2006. The Sandbox for analysis: concepts and methods. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 801–810.

[48] Jun Xiao and Jian Fan. 2009. PrintMarmoset: redesigning the print button for sustainability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 109–112.