

# Evaluating the Effect of Feedback from Different Computer Vision Processing Stages

## A Comparative Lab Study

**J. Kittley-Davies**

University of Southampton  
Southampton, UK  
jkd3g11@soton.ac.uk

**A. Alqaraawi**

University College London  
London, UK  
ahmed.alqaraawi.16@ucl.ac.uk

**R. Yang**

University College London  
London, UK  
r.yang@ucl.ac.uk

**E. Costanza**

University College London  
London, UK  
e.costanza@ucl.ac.uk

**A. Rogers**

St Anne's College, Oxford University  
Oxford, UK  
alex.rogers@cs.ox.ac.uk

**S. Stein**

University of Southampton  
Southampton, UK  
ss2@ecs.soton.ac.uk

### ABSTRACT

Computer vision and pattern recognition are increasingly being employed by smartphone and tablet applications targeted at lay-users. An open design challenge is to make such systems intelligible without requiring users to become technical experts. This paper reports a lab study examining the role of visual feedback. Our findings indicate that the stage of processing from which feedback is derived plays an important role in users' ability to develop coherent and correct understandings of a system's operation. Participants in our study showed a tendency to misunderstand the meaning being conveyed by the feedback, relating it to processing outcomes and higher level concepts, when in reality the feedback represented low level features. Drawing on the experimental results and the qualitative data collected, we discuss the challenges of designing interactions around pattern matching algorithms.

### CCS CONCEPTS

• **Human-centered computing** → **User interface design**;  
*Human computer interaction (HCI)*; User studies;

### KEYWORDS

controlled study, stop motion animation, computer vision, keypoints, feedback, processing pipelines

### ACM Reference Format:

J. Kittley-Davies, A. Alqaraawi, R. Yang, E. Costanza, A. Rogers, and S. Stein. 2019. Evaluating the Effect of Feedback from Different Computer Vision Processing Stages: A Comparative Lab Study. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019)*, May 4–9, 2019, Glasgow, Scotland UK. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3290605.3300273>

### 1 INTRODUCTION

As the power of mobile microprocessors increases and algorithms become ever more efficient, users of smartphones and tablet computers are increasingly being exposed to pattern recognition technologies [27]. While these technologies bring about new opportunities for HCI, they also raise new challenges for interaction designers [6]. One such challenge is how best to convey meaningful feedback so that users can understand the input requirements of these systems and so how best to interact with them. The research community has been quick to respond and there is a growing body of work examining how such systems can be made more intelligible [4, 24, 30]. This paper contributes to this space by examining computer vision as a specific application of pattern recognition. Given the growing number of apps which employ computer vision based pattern recognition (for brevity we refer to them as “smart camera apps”), this domain is of increasing importance for HCI research.

Smart camera apps are commonly designed to simplify user interaction. Amazon's mobile app for example, allows users to search for products using images captured with a device's camera. However, technical limitations (e.g. limited training datasets), environmental challenges (e.g. lighting conditions and shadows), image composition (e.g. “noisy backgrounds” and camera focus) and unrealistic user expectations [28] can all negatively impact user experience - making it difficult for users to reason about unexpected outcomes and in general how best to interact. Therefore,

---

*CHI 2019, May 4–9, 2019, Glasgow, Scotland UK*

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019)*, May 4–9, 2019, Glasgow, Scotland UK, <https://doi.org/10.1145/3290605.3300273>.

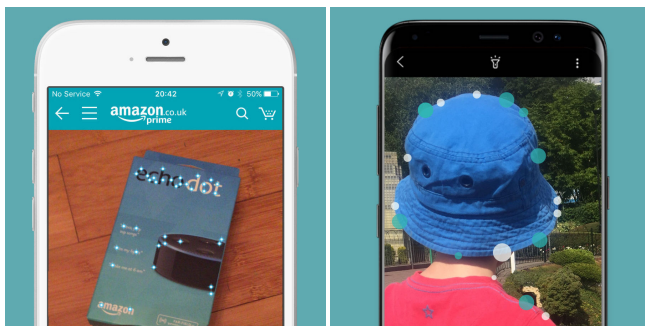
there is a need to support users of smart camera apps in their understanding of system behaviour, so that they can better overcome poor performance and failures. The challenge then is to make the reasons for failures intelligible, without requiring users to become experts in pattern recognition.

Perhaps it is to address these challenges that a number of commercial smart camera apps include visual feedback, overlaying the camera's viewfinder with visual aids. Two notable examples are the aforementioned Amazon app's "search by image" feature and Samsung's Bixby, a camera-based search tool<sup>1</sup> (Figure 1). Both display feedback in the form of "keypoint markers" - coloured dot visualisations which correspond to features of interest identified by an underlying algorithm. While such visualizations have long been popular as a debugging tool for software developers<sup>2</sup>, to date little is known about their effect on end-user interactions. Their inclusion may simply be motivated by a need to convey background activity, however, their presence raises some interesting questions: (i) are they intelligible to lay users? (ii) do they improve usability and aid users' interaction around failures? and conversely (iii) can they mislead users if misunderstood?

Addressing these questions through a controlled yet ecologically valid study is particularly challenging, because they require observing interaction around failures of the pattern recognition system. Such failures need to be controlled and repeatable, but their causes should not be obvious to study participants'. Moreover, the experimental tasks need to be engaging and enjoyable to motivate participants, have a clear goal and provide discussion points. To address these issues, we present a novel experimental lab study design enabled by a novel smart camera app that we developed. By so doing, we aim to make a methodological contribution to HCI.

<sup>1</sup>which tries to find matching images from an internet search

<sup>2</sup>e.g. OpenCV <https://goo.gl/bX4XEM>



**Figure 1: Smart Camera Apps that display keypoint markers feedback to users: left, Amazon and right, Samsung's Bixby.**

Leveraging this experimental design and the novel smart camera app, we conducted a between-groups study comparing keypoint markers with no feedback. Twenty participants with no formal technical training took part. Through a combination of quantitative and qualitative methods the results revealed that participants overwhelmingly misinterpreted the meaning of keypoint marker feedback. Participants interpreted them as indicating high level algorithmic explanations (e.g. about recognized objects), while in reality they refer to low-level features of the image (e.g. pixels). To better understand this finding, 20 new participants were exposed to two additional conditions designed around feedback that is actually related to higher level algorithmic explanations. More formally, this second study addresses a fourth question: does the processing stage (lower level vs higher level) from where the feedback is derived impact user understanding? Taken together, the results of the four experimental conditions indicate that keypoint marker feedback derived from the later stages of processing can be an effective means of informing user understanding. In addition, the studies suggest design tensions and implications for designers of pattern recognition feedback.

## 2 PATTERN RECOGNITION IN APPS

The keypoint marker feedback seen in many consumer applications is most likely derived from a keypoint matching algorithm, an intrinsic part of many smart camera apps, e.g. panorama stitching, object detection, gesture recognition and motion tracking. Most keypoint matching algorithms involve three stages of processing: (i) identify distinctive points of interest in an image (the keypoints), (ii) programmatically describe them, so that the description is resilient to geometric variations e.g. rotation, scale and perspective, and photometric variants e.g. contrast and brightness, and (iii) compare the descriptions with those of another image. How the results of this comparison process are used is application specific. In panorama stitching for example, the closest matching descriptions between images are assumed to represent the same point in the physical world. Using their relative changes in position the images can be transformed such that the keypoints overlap creating a new combined image with a wider field of view.

## 3 RELATED WORK

### The importance of system intelligibility

The significance of how users understand system operation has received considerable attention from the fields of HCI and psychology. Much of the early work in this space centered around the theoretical construct of Mental Models - users' internal representations which allow them to explain

and predict the actions of a system, permitting them to reason about their interactions before committing to an action [19, 20]. This work indicated that the accuracy of a user’s mental model could dramatically impact their capacity to interact effectively with a system [3, 7] and that users with coherent mental models perform tasks more efficiently [11]. More recently, studies have shown that users with more complete mental models are more likely to make systems operate to their satisfaction [5, 16] and that flawed mental models can result in confusion, misconceptions, dissatisfaction and erroneous interactions [15, 26]. When specifically considering systems which employ pattern recognition or machine learning technologies, the overestimation of a system’s intelligence or capabilities has been shown to negatively impact user interaction [1]. This can lead to users becoming over-reliant and so less vigilant to system failures [29]. It can also result in unrealistic expectations [28], the violation of which can impact user trust [12].

User understanding can be affected by many factors, from simple misunderstandings of terminology [2] to complex subconscious biases. A recent study of floor cleaning robots [9] describes how users rated a cleaning robot’s performance to be better when they had witnessed it in motion. The evidence supporting the need to inform users of underlying processes is compelling. However, the means by which this can be achieved is non-trivial when discussing smart systems, with the inherent complexity of these systems potentially inhibiting interaction design [6]. Our work contributes to this space, investigating the importance of user understanding on effective interaction with pattern recognition and computer vision systems. Further to this, we report observations of misconceptions and the consequences for user interaction.

### **Informing users of smart systems**

Investigations of a context aware system [17], which employs machine learning algorithms to make decisions, have demonstrated not only the benefits of making the motivations behind automated decisions salient to novice users (via text notifications), but that explanations of why system behaviour occurred result in better user understanding than explanations of why not, an observation supported in a later study [13] of simulated driver assistants (albeit through a different modality). Researchers found that audible messages which report only the intended actions of the system had little impact on driver safety, whereas messages which conveyed why these actions were necessary had a positive influence. In this paper we expand on this work, examining how effective existing feedback visualisations are at informing user understanding and discuss the implications for designing effective visual feedback for such systems.

*Interaction with pattern matching systems.* Software platforms such as Crayons [8] and Eyepatch [18] were specifically developed to insulate users from the complexities of computer vision and pattern matching technologies. They theorize that by providing users with interfaces that facilitate “rapid trial-and-error” [8], the most effective solutions to classification problems can be found. However, as pattern recognition technologies become increasingly complex, Patel, et al [22] have suggested that successful implementation can only be achieved with a deeper understanding of the inner-workings of the processes. In contrast to Crayons and Eyepatch, DeJaVu [10] was developed to expose domain-expert programmers to computer vision technologies, with the ambition of aiding code debugging. The system allows images passing through the various stages of processing to be inspected and an interactive timeline interface lets users record and examine data flow temporally. Although a small user study was conducted, the focus was system functionality rather than the user experience. Our work builds on this by conducting a lab study specifically designed to evaluate the impact of exposing lay users to pattern matching processes.

Zhao et al. [31] conducted a study examining lay-users’ interactions with an augmented reality pattern recognition system designed to assist users with low vision in a product search task, by highlighting regions of a head mounted display with visual feedback. The feedback in this case was derived from the output of the pattern matching processing pipeline. Our work builds on the work of Zhao et al, exploring the relationship between the origin of the data which informs feedback and users’ interpretation of it.

*Exposing data processing.* Exposing the underlying data processing is an idea which has been explored in the domain of machine-learning. The creators of Gestalt [21] an integrated development environment (IDE) designed specifically to assist programmers creating software which makes use of machine learning technologies, demonstrated through lab studies, that exposing data at various stages of a process significantly improves programmers’ ability to identify and correct errors in their code. Similarly, Prospector [14] which facilitates the probing of the predictive models by data scientists, was shown to help them understand how features affect the overall predictions. They report that by allowing users to adjust input variables and see through visualisations to how the model responds, users gained deeper insights into how the model worked. Our work builds on these findings by demonstrating the capacity of algorithmic feedback to support lay-user understanding, but also how it can lead to misconceptions if designed badly. We expose a number of design implications and discuss the challenges of designing effective visual feedback for such systems.

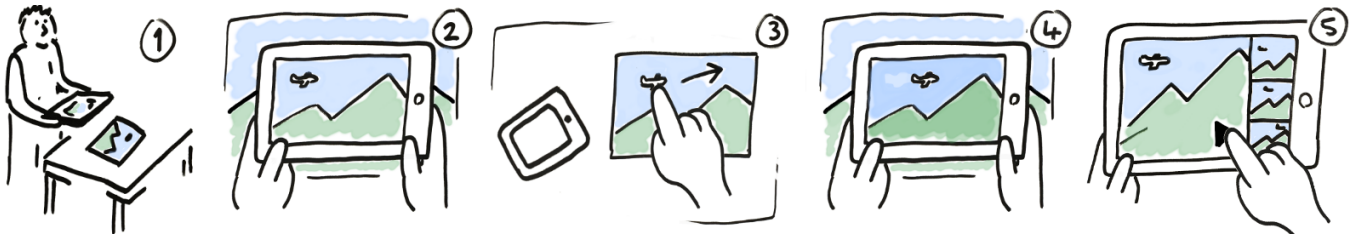


Figure 2: Creating an animation - (1) Set up the background scene with the character in its starting position. (2) Capture an image which encapsulates the scene and the character. (3) Reposition the character. (4) Capture another frame. (5) Preview the captured frames as an animation. Delete and retake if not acceptable, or repeat stages 3 to 5 until the animation is complete.

#### 4 STUDY DESIGN

We designed and conducted a between-groups study with four conditions. The conditions were selected to examine the impact of common visual feedback techniques and assess if the processing pipeline stage from which the feedback is derived has an effect on user understanding. Developing an ecologically valid and testable experimental task which incorporated a keypoint matching algorithm proved non-trivial. The task needed to provide sustained exposure to algorithmic feedback so that participants could observe and reason about the feedback. Further to this, participants must experience instances of failure and success. The task therefore should be controllable, but in a way not obvious to participants. In addition, it would be advantageous for the task to be enjoyable to motivate interaction, have a clear goal and provide discussion points. Through experimentation a task which best satisfied these criteria was developed, the creation of stop-motion animations.

To create a stop-motion animation, an animator must capture a series of still images (frames) of a given scene. By incrementally moving artifacts (characters) between frames the illusion of animation can be achieved i.e. when the frames are played back in order the characters appear to move autonomously in relation to the static elements of the scene (e.g. the background). Figure 2 demonstrates the process. Traditionally stop motion animations are created using cameras where the position and angle are strictly controlled e.g. held in a tripod. To incorporate pattern recognition technologies in to our study design we replaced the controlled camera with a handheld tablet computer and bespoke app (Anim8<sup>3</sup>) which employs a keypoint matching algorithm<sup>4</sup> to align each frame to its predecessor - a process of stabilization. This process makes all frames appear to have been captured from the same physical location even though the camera's position and angle vary. The keypoints with the closest descriptions

are matched and assumed to point to the same physical feature in both frames. The most recently captured image can then be transformed so that its keypoints overlap its predecessors. Characters which have been moved between frames will create erroneous mappings, however if enough matches are found for the elements of the scene which have remained static (e.g. the background) then the matches associated with the moving characters will be treated as outliers and ignored.

In order for the stabilization process to work effectively it is critical that the static elements of the scene are “feature rich”, i.e. the algorithm can identify many keypoints. If there are too few then the transformation process may output an image where the background is distorted and the character remains stationary (Figure 3). Leveraging this limitation, the likelihood of whether the stabilization process will succeed or fail can be controlled - by providing “feature rich” and “feature poor” backgrounds participants of the study can be exposed to situations where the stabilization process succeeds and fails respectively. Factors such as lighting conditions, shadows and camera angle make this form of manipulation not immediately obvious to study participants.

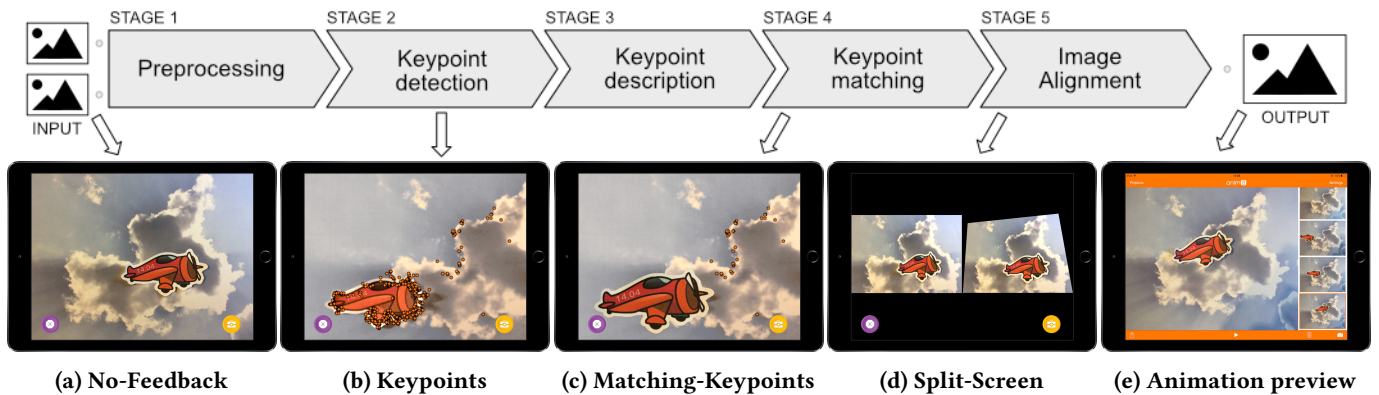
Through pilot studies we concluded that four animation tasks with 4 to 5 frames per task provides sufficient exposure. We designed the tasks to assess whether feedback derived from the stabilization process can help participants develop better understandings of the systems' needs. To create discussion points and elicit user understanding we ask participants to choose one of three background options in the last two



Figure 3: When too few matching keypoints are identified in the background, the stabilization process can result in an image transformed such that the character appears to remain stationary and the background becomes distorted.

<sup>3</sup>For more information about the Anim8 app visit: <http://anim8.space/>

<sup>4</sup>Through experimentation the ORB algorithm [23] proved to offer the best compromise of performance, speed and control for our study.



**Figure 4: Examples of the feedback conditions presented by the Anim8 application and their relationship to the processing pipeline (a, b, c, d). Also the preview interface (e). Note: To see these images animate see supplemental materials.**

animation tasks (3 options per task). The feature richness of the three background options varied and thus the likelihood of the stabilization process succeeding varied (Figure 6).

### Conditions

To explain the study conditions, we describe them in relation to the computer vision pipeline employed by Anim8 (Figure 4). It should be noted that we did not explain the feedback nor point out its presence to participants. This was done to mirror the experiences of current consumer smart camera app users.

*No-Feedback (Figure 4a).* This condition was included as a baseline. The input images to the pipeline were presented back to participants without any additional feedback.

*Keypoints (Figure 4b).* The camera’s viewfinder was augmented with keypoint markers which indicate the locations at which keypoints had been detected in stage 2. It is important to note that not all the identified keypoints will be matched. Matches where the descriptions are considered too dissimilar are deemed outliers and are ignored by the stabilization process. Despite this, the location, distribution and volume of identified keypoints are good indicators for the potential success of the stabilization process.

*Matching-Keypoints (Figure 4c).* Again the viewfinder was augmented with keypoint markers, however in this case only those which have been successfully paired with keypoints in the previous frame were displayed (Stage 4).

*Split-Screen (Figure 4d).* This condition represents the final stage of processing. The viewfinder was divided into two equal halves. On the left: the input image updated in real-time (as per No-Feedback condition). On the right: the image outputted by the processing pipeline (update every ~120ms).

The No-Feedback and Keypoints conditions were compared first, while the Matching-Keypoints and Split-Screen conditions were included at a later stage, as described in the Introduction.

### Procedure

All studies were conducted in the same empty windowless meeting room (so lighting conditions could be controlled) on a university campus. Two experimenters were present at all times - one to conduct the experiment and the other to observe, take notes and make audio recordings.

At the start of the study participants received written instructions detailing: (i) the procedure necessary to create stop-motion animations, (ii) how Anim8 uses computer vision technologies to remove the need for a tripod, and (iii) a high level explanation of the image processing operations - that Anim8 tries to align images “by looking for things in each image which are not supposed to have moved, for example the background”. After reading the instructions participants were asked to stand up while performing the animation tasks.

Participants were tasked with creating 4 stop motion animations. Animating a two dimensional cardboard character (approximately 8cm by 5cm in size) moving across an A3 printed background (see Figure 5 for examples). To ensure that all participants had a good understanding of how to use the Anim8 application, the experimenter demonstrated the capture, playback and delete operations prior to the first task commencing. Whilst demonstrating the capturing of a frame, the participants were advised to ensure the printed background scene was fully encapsulated in the camera’s viewfinder and that the desk should not be visible. This was done to prevent features other than those in the scene impacting the outcome of the experiment (this was not explained to the participant). The participants were also advised that

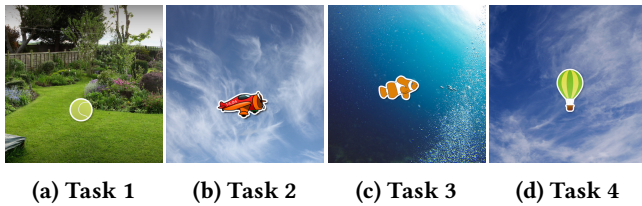


Figure 5: Example frame for each of the animation tasks.

if they needed any assistance regarding the operation of the application during the study, then they could ask at any time.

Prior to each animation task, the experimenter provided each participant with the necessary materials (i.e. a character to animate and static background scene / scenes) and an instruction sheet detailing an example path for the character to follow, along with the number of frames expected (4 to 5). On completion of the task, the participant was asked to play back the animation they had created to the experimenter. The tasks were conducted in the same order for all participants to ensure that they experienced both successful and unsuccessful attempts. The tasks were structured as follows:

*Task 1.* was designed to allow participants to familiarise themselves with the UI and reassure them that the app works as described. To this end, a feature rich background (Figure 5a) which proved in testing to work with almost no failures was provided, making the task easy to succeed. On completion, the experimenter asked how the participants found using the app and if they had any queries.

*Task 2.* was designed to highlight the limitations of the system. The background in this task (Figure 5b) proved in testing to always fail. As it was impossible to complete this task, the experimenter would intervene after a time limit of 2 minutes, if the participant had not already raised concerns. The experimenter would ask the participants to explain what was happening and if they knew why it did not work, before suggesting that they proceed to the next task for brevity.

*Task 3.* was designed to assess users' understanding and create a point of discussion in the interview. Participants were asked to choose the background they felt would work best for the app from a selection of 3 backgrounds (see Figure 6). Participants were advised that they could preview them through the application's viewfinder if they wished. The backgrounds offered had previously been assessed and ranked according to the algorithm's ability to effectively identify features within them. One of the backgrounds consistently failed in testing and the remaining two consistently worked well, although one was more visibly "feature rich" than the other. The motivation for presenting users with this range of background options was to make the different levels of detail between the backgrounds less obvious. Once the

participant completed this animation task, they were asked why they had selected that specific background.

*Task 4.* followed the same structure as Task 3, with a new character and set of 3 backgrounds (see Figure 6). This last task was designed to sustain participant interaction with the application, collect an additional data point and further assess user understanding i.e. what, if anything, had been learned in Task 3.

At the end of the study a semi-structured interview was conducted. The interview began by asking participants if their experience in Task 3 and Task 4 had given them a better understanding of why the animation in Task 2 resulted in failure. Using this as a starting point, the experimenter asked further questions to assess the participants' understanding of the algorithm and their motivations for selecting the backgrounds in Task 3 and Task 4. For the participants of conditions where feedback was presented in the viewfinder, the experimenter also asked what they thought it represented and if they used it in their decision making.

## 5 PARTICIPANTS

We recruited 40 participants (15F, 25M) from the university participant pool which includes university staff, students and the general public. Anyone who expressed interest was allowed to participate in the study, so long as they did not identify as having technical hobbies or interests (e.g. computer programming), were not in technical employment (e.g. lab assistant) and were not technically educated (e.g. no degree in computing or engineering related subjects). Participants were also required to have normal or corrected to normal vision. Each participant received a £10 payment for their participation. Of the 40 participants 29 reported to be in education and 11 in full time employment. Participants' backgrounds were diverse with the most common being Business & Economics (13) followed by Social Sciences (9) Law (5), Languages (5), Art (4), Accountancy (2), Medicine (1) and Geography (1). One participant was aged between 40 and



Figure 6: Background options presented to participants in Tasks 3 (Top row) and Task 4 (Bottom row). Left: Likely to fail, Center and Right: Likely to succeed.

49 years, 6 between 30-39 and 33 between 20-29. For more detailed information please see the supplemental materials.

Ten participants were randomly assigned to each condition. For conciseness, we will refer to participants by condition and subject number, for example, K7 was subject number 7 of the Keypoints condition. Prefixes “N”, “M” and “S” refer to the No-Feedback, Matching-Keypoint and Split Screen conditions respectively.

## 6 QUANTITATIVE FINDINGS

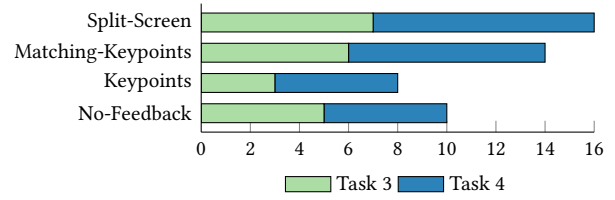
To quantitatively assess the effect of feedback across the conditions, three researchers independently coded participants’ responses to questions pertaining to their background selections (taken from researcher notes and transcripts of audio recordings). This coding process was specifically focussed on the participants’ understanding of how the system works (in contrast, in the next section we report a further analysis of the data through broader, more general coding). In particular, a participant’s response was coded as “correct understanding” if they described how the presence of distinctive shapes and features in the background positively impacted the app’s ability to align frames. For example, the following statements were coded as demonstrating a correct understanding: “I think it picks up the shapes on the picture and it [...] then compares the position of the dots on the other one [...] the next picture? So it can tilt the frame accordingly” (K9) or “because the background is distinct enough” (N6). If a participant reported motives not connected to the requirements of the app or their understanding of what is significant was incorrect they were coded as “incorrect understanding”. For example, the following statements were coded as demonstrating an incorrect understanding: “Because it’s nice and colourful” (N8) or “[...]it looked more homogenous than the other ones. So I thought [...] it would be easier to take the photos like this” (K2).

Table 1 summarizes the background selections made by participants in Task 3 and Task 4 and Figure 7 shows whether their selection was based on a correct understanding of the stabilization processes.

To compare participants’ understanding between the conditions we consider the total number of answers which

**Table 1: No. Participants who selected a “correct background” i.e. suited to the needs of the app.**

	Task 3	Task 4
No-Feedback	10	10
Keypoints	7	10
Matching-Keypoints	10	10
Split-Screen	9	10



**Figure 7: No. Participant responses coded as “correct understanding” when reporting their motivation for background selection in task 3 and task 4.**

demonstrated a correct understanding in Task 3 and Task 4 (Figure 7). For example, 7 of the 10 participants in the Split-Screen condition demonstrated a correct understanding in Task 3 and 9 participants in Task 4, giving a summed value of 16. A chi-square test of the summed values revealed a statistically significant difference (chi-square=8.33,  $p=.040$ ,  $df=3$ , Cramer’s  $V=0.323$ ). To better understand the differences between the conditions, we analysed the chi-squared standardized residuals (presented in Table 2). It can be noticed that the standardized residuals are larger (in absolute value) for the Keypoints and Split-Screen conditions, suggesting that these two conditions explain the significance of the chi-square test. A chi-square test also shows no statistically significant differences for correct background selections (chi-square=6.316,  $p=.097$ ,  $df=3$ ), nor when testing the tasks individually<sup>5</sup>. It should be noted that participants sometimes selected a ‘feature-rich’ background for aesthetic reasons rather than because it would make the app work better (as instructed), failing to demonstrate correct understanding. In the next section we discuss our qualitative findings and the role of background selection further.

## 7 QUALITATIVE FINDINGS

Transcripts of all audio recordings and researchers’ notes collected during the studies were also independently coded by three researchers in a second round of analysis. Codes were

<sup>5</sup>understanding on Task 3: chi-square=3.509,  $p=.320$ ,  $df=3$ , Cramer’s  $V=0.296$ ; understanding on Task 4: chi-square=5.812,  $p=.121$ ,  $df=3$ , Cramer’s  $V=0.381$ ; correct selections on Task 3: chi-square=6.667,  $p=.083$ ,  $df=3$ ; all selections were correct in Task 4, so no statistical test needed

**Table 2: Standard residual results of the No. participants who demonstrated a “correct understanding”.**

	Count	Expected	Std Residual
No-Feedback	10	12	-0.6
Keypoints	8	12	-1.2
Matching-Keypoints	14	12	0.6
Split-Screen	16	12	1.2

initially drawn from research questions and then supplemented with those that emerged from the interviews before being grouped by consensus. In the subsequent subsections we detail these groups and give example quotations. First however, we would like to note that overwhelmingly participants reported the task to be interesting and entertaining. This suggests that the experimental task was sufficiently engaging and participants were invested in creating animations successfully.

### **Participants drew from their existing knowledge**

First we note, that when asked about previous experience with computer vision applications, participants mentioned QR Code scanning, Facebook and Instagram (none of which provide visual feedback). No participants reported using Amazon or Bixby's search by image, or any other application which provides keypoint feedback.

In the No-Feedback condition, half of the participants demonstrated a correct understanding. These participants explained that having elements in the background which were "more detailed" (N1), "most defined" (N7), "distinct" (N6) or "prominent" (N2) would help the app because they were good reference points for alignment. The remaining five participants had an incorrect understanding and in the main focussed on the aesthetics, e.g "I thought the clouds would go really well with [...] the hot air balloon" (N9).

Interestingly, participants in the No-Feedback condition selected a correct background more often than participants in the Keypoints condition (Table 1). Participants K2, K4 and K8 of the Keypoints condition made associations between the keypoint markers and their experience of other applications, suggesting that the keypoint markers functioned in much the same way as the autofocus on digital cameras, in that they highlight regions on which the camera is focussing. Whether these analogies are helpful is not clear. One of the participants who drew such parallels made good choices when selecting backgrounds, while the remaining two were misled by their assumptions - K2 for example, chose a feature poor background for Task 3, expecting that a plain background would make it easier for the app to identify the character.

### **Early stage keypoint marker feedback is not easy to understand**

Participants of the Keypoints condition broadly failed to understand the meaning of keypoint markers and how it related to low-level features of interest to the algorithm (30% demonstrated a correct understanding in Task 3 and 50% Task 4). Participants K1, K2 and K3 incorrectly thought that the keypoint markers were highlighting regions where the algorithm had identified a moving object, something the user intended to animate. These participants theorised that if the algorithm succeeds in finding the objects which are meant to

move, then the algorithm will be able to successfully transform the captured images to create animations e.g. K2 said "these dots might help show that the focus of the photo is the [character] [...] if I have these dots around the [character] then the image will be clearer". K2 and K3 both selected the worst background option for Task 3. They justified their choice by saying that among the three options the plainest background would work best because it would make the identification of the character easier for the algorithm e.g. K3, when asked why they chose a plain background in Task 3, said it was "because [the app] could be confused about the subject of the picture". Both K2 and K3 expressed confusion when keypoint markers appeared in locations which did not fit their understanding of how the system works i.e. on the background instead of the character. K2 remarking: "[keypoint markers] try to capture the [character] in the photo, a balloon, [...], but it's not on the balloon" and K3, "[if keypoint markers] mean the [character] is moving, [...] I don't understand why [keypoint] markers are showing up on the cloud, not the [character]". Despite witnessing evidence to the contrary both participants failed to correct their misunderstanding, a behaviour pattern previously reported in work on intelligent system [26].

### **When keypoint marker feedback was helpful**

The quantity of the keypoint markers was the most commonly reported explanation of how participants took into account Keypoint feedback. For example, K1 explained that if "[...] in background, [I] see a lot of dots. I can tell that background is definite. When I did the [animation of the] plane [for which the app failed], there were only 1 or 2 dots". K6 stated that "if there is nothing [in the background], it's not going to work. [If] something is there it's going to work". However, only four participants demonstrated a better understanding which was consistent with the workings of the stabilization process. These participants noticed how and where the keypoint markers appeared and were able to develop more specific theories of how the algorithm identifies keypoint markers within an image. For example, K10 correctly speculated that the algorithm "pick[s] up the shape" and "areas of heavy contrast".

In the Matching-Keypoint condition, six of the ten participants reported the feedback to be helpful. Of these participants, three described the keypoint markers as indicators, reporting what the algorithm was doing: "I can see what the dots are surrounding. [...] I know what it's doing" (M10), "when I saw [keypoints markers], it was more reassuring [...] saying you're doing it right" (M7), and "the app is trying to match between images [...] things which the app sees in this image which it also saw in the previous image" (M1). The other three participants explained that they saw the keypoint markers as guides, that the keypoint markers were designed



to help them test if the background image would work or not: “the dots showed if the picture would work out” (M6), “I can tell what’s the problem of the image” (M8) and “[the keypoints] might help you pick a background” (M5).

Participants in the Keypoints condition tended to overestimate the meaning of the Keypoint feedback and relate the meaning to higher level concepts, such as the separation of background and foreground objects. In this regard Matching-Keypoints appeared to be more intuitive as its meaning is more inline with user expectation. M1 for example, reported that when the app didn’t work in Task 2 he did not know why. During Task 3, he speculated that the colour might have an effect (lighter or darker colour), but found through experimentation that this was not the case. He then correctly theorized that the app needed distinct features. He explained, “The dots meant like it’s picking distinct points throughout the image. [...] I think [the app is] re-mapping the points that [it had] taken in an image before. I think that’s what it’s trying to do”.

### **Split Screen feedback was helpful, but not in the way we expected**

Seven participants in the Split-Screen condition also reported the feedback to be helpful. Four participants suggested that it acted as a cue, indicating when best to capture a frame e.g. “The preview helped me decide when to take a picture” (S7) or “I [wait] for the preview to stabilize before taking the picture” (S3). An artifact of the stabilization processes implementation is a “flickering effect” which occurs when the system is rapidly toggling between a successful transform and a failure. This strictly speaking is a usability “bug” which participants reappropriated, using it as a means of gauging the likelihood of a successful transform e.g. “If it was flickering I wouldn’t take the picture” (S7), and “I waited for a clear picture [...] then hit capture” (S4).

Another unexpected way of using Split-Screen feedback was described by two participants (S7 and S2). They used the feedback to position the camera in the same place as the previous image, S7 commenting “the preview tells me what angle to take the picture from”. Both participants would keep moving the camera until the left and right images matched in the preview i.e. the alignment transformation was minimal. This approach does in fact help make better quality animations, however it is not how the app was intended to be used and this process of positioning was very time consuming for the participants.

### **When feedback was not helpful**

Five participants in the Split-Screen condition and three in the Matching-Keypoints condition reported the feedback to be distracting or unhelpful. For example, “I found the split screen very distracting and would rather not see it” (S4),

“I found the dots distracting because it ruined the focus at times” (M4), “They were a bit annoying, they get in the way” (M1) and “they could be obstructive” (M6). Interestingly, S6 described the feedback as unhelpful because they preferred to frame the photo from memory, using the viewfinder to align the camera with features they had identified in the background. To this end the preview was unhelpful because the split screen design reduced the size of the viewfinder. These comments illustrate the risk that feedback visualisations can be distracting.

### **Background selection motivation**

Although all participants selected a correct background in Task 4, not all provided a correct explanation. Participants responses when asked why they chose the background image they selected in Task 3 and Task 4 were coded into one of two categories: aesthetic - they were motivated by how the image looked, and detail - where they stated in some way that the level of detail was important (including incorrect understandings). Aesthetics was the primary motivation for 27 selections out of 80 (10 No-Feedback, 9 Keypoints, 5 Matching-Keypoints and 3 Split-Screen), with detail accounting for the remaining 53 selections (10 No Feedback, 11 Keypoints, 15 Matching-Keypoints and 17 Split-Screen). It should be noted that it is by chance that some of our participants considered the correct background to be more aesthetically pleasing.

## **8 DISCUSSION**

In the introduction we set out a series of questions. In this section we discuss the outcomes of our study using these questions as a scaffold.

### **Does the processing stage from which feedback is derived impact user understanding?**

Our results indicate that feedback derived from the later stages of the processing pipeline (Matching-Keypoints and Split-Screen) are more effective at informing users’ understanding. The chi-square test of “user understanding” reveals a significant difference between conditions, with the standard residuals indicating the Keypoints and Split-Screen are responsible. More participants of the Split-Screen condition demonstrated a correct understanding of how the system works than participants of any other condition (Figure 7), with Matching-Keypoints second. In contrast, participants in the Keypoints condition performed worse than participants who received no feedback at all.

Despite users understandings varying between conditions, most participants across all conditions were successful in selecting a correct background (see Figure 1). As mentioned

above, participants sometimes selected the correct background for aesthetic reasons, rather than to make the algorithm work (as requested by the study instructions). As a consequence, instead of using selection as a measure of understanding, we rely only on the participants' explanations of *why* they selected a specific background.

### **Is keypoint marker feedback intelligible to lay-users?**

More participants in the Matching-Keypoints condition were able to correctly describe the input requirements of the system in comparison with those who received no additional information in the form of feedback (No-Feedback). Interview responses indicate that users have a tendency to interpret feedback as an outcome rather than a progress notification of an intermediary stage. In this regard Matching-Keypoints appeared to be more intuitive, as their meaning is more inline with user expectation. We tentatively propose that keypoint markers can be used to inform user understanding, so long as the meaning being conveyed is inline with user expectations.

### **Can keypoint markers mislead if misunderstood?**

Given that the Keypoints and Matching-Keypoints conditions utilise exactly the same feedback visualisation (keypoint markers), the result showing that Keypoints condition participants were least able to understand the needs of the algorithm (Figure 7) suggests that they may have been detrimental to user understanding. While the keypoint markers are a good indicator of the future stabilization processes success, participants commonly understood them to represent the final output, that they represented regions where the stabilization process had identified matches. It is feasible that this misconception could result in users using the markers in ways which inhibit their interactions. Indeed, Keypoints condition participants' interview responses indicate a disconnect between their interpretation of feedback and the actual information conveyed e.g. K3, "[if keypoints] mean the [character] is moving, [...] I don't understand why keypoints are showing up on the cloud, not the [character]".

### **Can keypoint markers improve usability and aid users' interaction?**

The inherently visual nature of computer vision processes, both in their input and also the intermediate stages, makes visual feedback the logical medium through which to deliver feedback [10]. However, participants in our studies, at times reported the feedback to be distracting or obtrusive (e.g. M1 "They were a bit annoying, they get in the way"). This highlights a design tension between attracting attention and causing distraction, and between being informative and not overwhelming. These tensions are well understood in graphic design, particularly around the design of interactive visualizations. However, the situation here is more complex.

Some aspects of algorithm design are conceptually simple and naturally map to visual representations. Keypoints for example, are a concept that lend themselves to being represented pictorially e.g. by marking their physical location with geometric points. It could at first be tempting to see this as an example of "form follows function" [25], however when dealing with the design of feedback for systems which employ pattern matching algorithms, we argue that the "form follows function" principle requires careful interpretation. What is "function" in this case? At first, it may seem to be the "technical" function of the algorithm, but this is not the case. We need to remind ourselves that the "function" is instead the function to help users understand what the system does. One implication then, is that to design feedback, it may be beneficial to distance oneself from the question of how algorithmic steps and internal states map to form, and instead think about the end result of the system and how it will be used. Moreover, in some cases, it may be challenging, or even impossible, to map the function of the algorithm to form.

## **9 CONCLUSIONS**

This paper reported a comparative between-groups lab study examining the role of visual feedback in smart camera apps. Leveraging a novel experimental design centered on the creation of stop-motion animations, 40 participants were exposed to four different levels of feedback. Through a combination of quantitative and qualitative methods, our findings indicate a disconnect between user expectations and the information actually represented by the feedback. Participants exposed to keypoint marker feedback derived from early stages of processing showed a tendency to misunderstand it and overall they performed worse than participants who received no feedback at all. Conversely, participants who received keypoint marker feedback derived from later stages of processing demonstrated an improved understanding of the system operation. We conclude that the stage of processing from which feedback is derived plays an important role in users' ability to develop coherent and correct understandings of a system's operation. We hope that the results presented in this paper will inform the design of feedback in smart camera apps, and other applications of pattern recognition. More generally, we hope that our study method can be used by HCI researchers in future work exploring the design space of feedback and cues.

## **ACKNOWLEDGMENTS**

This work is supported by the Engineering and Physical Sciences Research Council AperiO (EP/L024608/1) and A-IoT (EP/N014243/1) projects. Study approved by the Ethics Committees of UCLIC and of the University of Southampton (ref: 27198). Data URI: <https://doi.org/10.5258/SOTON/D0757>. See supporting materials for image attributions.

## REFERENCES

- [1] Alper T. Alan, Enrico Costanza, Sarvapali D. Ramchurn, Joel Fischer, Tom Rodden, and Nicholas R. Jennings. 2016. Tariff Agent: Interacting with a Future Smart Energy System at Home. *ACM Trans. Comput.-Hum. Interact.* 23, 4, Article 25 (Aug. 2016), 28 pages. <https://doi.org/10.1145/2943770>
- [2] Alper T. Alan, Mike Shann, Enrico Costanza, Sarvapali D. Ramchurn, and Sven Seuken. 2016. It is Too Hot: An In-Situ Study of Three Designs for Heating. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5262–5273. <https://doi.org/10.1145/2858036.2858222>
- [3] Piraye Bayman and Richard E. Mayer. 1984. Instructional manipulation of users' mental models for electronic calculators. *International Journal of Man-Machine Studies* 20, 2 (1984), 189 – 199. [https://doi.org/10.1016/S0020-7373\(84\)80017-6](https://doi.org/10.1016/S0020-7373(84)80017-6)
- [4] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani B. Srivastava, Alun D. Preece, Simon J. Julier, Raghuvveer M. Rao, Troy D. Kelley, Dave Braines, Murat Sensoy, Christopher J. Willis, and Prudhvi Gurram. 2017. Interpretability of deep learning models: A survey of results. *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (2017)*, 1–6.
- [5] Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. 2008. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction* 18, 5 (20 Aug 2008), 455. <https://doi.org/10.1007/s11257-008-9051-3>
- [6] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX Design Innovation: Challenges for Working with Machine Learning As a Design Material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 278–288. <https://doi.org/10.1145/3025453.3025739>
- [7] Benedict du Boulay, Tim O'Shea, and John Monk. 1981. The black box inside the glass box: presenting computing concepts to novices. *International Journal of Man-Machine Studies* 14, 3 (1981), 237 – 249. [https://doi.org/10.1016/S0020-7373\(81\)80056-9](https://doi.org/10.1016/S0020-7373(81)80056-9)
- [8] Jerry Fails and Dan Olsen. 2003. A Design Tool for Camera-based Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 449–456. <https://doi.org/10.1145/642611.642690>
- [9] Pedro Garcia Garcia, Enrico Costanza, Sarvapali D. Ramchurn, and Jhim Kiel M. Verame. 2016. The Potential of Physical Motion Cues: Changing People's Perception of Robots' Performance. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 510–518. <https://doi.org/10.1145/2971648.2971697>
- [10] Jun Kato, Sean McDermid, and Xiang Cao. 2012. DejaVu: Integrated Support for Developing Interactive Camera-based Programs. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 189–196. <https://doi.org/10.1145/2380116.2380142>
- [11] David E. Kieras and Susan Bovair. 1984. The role of a mental model in learning to operate a device. *Cognitive Science* 8, 3 (1984), 255 – 273. [https://doi.org/10.1016/S0364-0213\(84\)80003-8](https://doi.org/10.1016/S0364-0213(84)80003-8)
- [12] René F. Kizilcec. 2016. How Much Information?: Effects of Transparency on Trust in an Algorithmic Interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2390–2395. <https://doi.org/10.1145/2858036.2858402>
- [13] Jeamin Koo, Jungsuk Kwac, Wendy Ju, Martin Steinert, Larry Leifer, and Clifford Nass. 2015. Why did my car just do that? Explaining semi-autonomous driving actions to improve driver understanding, trust, and performance. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 9, 4 (01 Nov 2015), 269–275. <https://doi.org/10.1007/s12008-014-0227-2>
- [14] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5686–5697. <https://doi.org/10.1145/2858036.2858529>
- [15] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI '15)*. ACM, New York, NY, USA, 126–137. <https://doi.org/10.1145/2678025.2701399>
- [16] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. 2012. Tell Me More?: The Effects of Mental Model Soundness on Personalizing an Intelligent Agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1–10. <https://doi.org/10.1145/2207676.2207678>
- [17] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. 2009. Why and Why Not Explanations Improve the Intelligibility of Context-aware Intelligent Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 2119–2128. <https://doi.org/10.1145/1518701.1519023>
- [18] Dan Maynes-Aminzade, Terry Winograd, and Takeo Igarashi. 2007. Eyepatch: Prototyping Camera-based Interaction Through Examples. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, New York, NY, USA, 33–42. <https://doi.org/10.1145/1294211.1294219>
- [19] Neville Moray. 1999. Mental models in theory and practice. *Attention and performance XVII: Cognitive regulation of performance: Interaction of theory and application (1999)*, 223–258.
- [20] Donald Norman. 2014. *On the relationship between conceptual and mental models*. In *Gentner et al (e.d) Mental Models*. Psychology Press.
- [21] Kayur Patel, Naomi Bancroft, Steven M. Drucker, James Fogarty, Andrew J. Ko, and James Landay. 2010. Gestalt: Integrated Support for Implementation and Analysis in Machine Learning. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 37–46. <https://doi.org/10.1145/1866029.1866038>
- [22] Kayur Patel, James Fogarty, James A. Landay, and Beverly Harrison. 2008. Investigating Statistical Machine Learning As a Tool for Software Development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 667–676. <https://doi.org/10.1145/1357054.1357160>
- [23] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An Efficient Alternative to SIFT or SURF. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV '11)*. IEEE Computer Society, Washington, DC, USA, 2564–2571. <https://doi.org/10.1109/ICCV.2011.6126544>
- [24] K. Simonyan, A. Vedaldi, and A. Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Workshop at International Conference on Learning Representations*.
- [25] Louis H Sullivan. 1896. The tall office building artistically considered. *Lippincott's Magazine* 57, 3 (1896), 406.
- [26] Joe Tullio, Anind K. Dey, Jason Chalecki, and James Fogarty. 2007. How It Works: A Field Study of Non-technical Users Interacting with an Intelligent System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 2390–2395. <https://doi.org/10.1145/2858036.2858402>

- Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 31–40. <https://doi.org/10.1145/1240624.1240630>
- [27] Qian Yang, Nikola Banovic, and John Zimmerman. 2018. Mapping Machine Learning Advances from HCI Research to Reveal Starting Places for Design Innovation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 130, 11 pages. <https://doi.org/10.1145/3173574.3173704>
- [28] Rayoung Yang and Mark W. Newman. 2013. Learning from a Learning Thermostat: Lessons for Intelligent Systems for the Home. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 93–102. <https://doi.org/10.1145/2493432.2493489>
- [29] Rayoung Yang, Mark W. Newman, and Jodi Forlizzi. 2014. Making Sustainability Sustainable: Challenges in the Design of Eco-interaction Technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 823–832. <https://doi.org/10.1145/2556288.2557380>
- [30] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *Computer Vision*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 818–833.
- [31] Yuhang Zhao, Sarit Szpiro, Jonathan Knighten, and Shiri Azenkot. 2016. CueSee: Exploring Visual Cues for People with Low Vision to Facilitate a Visual Search Task. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 73–84. <https://doi.org/10.1145/2971648.2971730>