

Twitch Crowdsourcing: Crowd Contributions in Short Bursts of Time

Rajan Vaish¹, Keith Wyngarden², Jingshu Chen², Brandon Cheung², Michael S. Bernstein²

¹ UC Santa Cruz
Santa Cruz, CA
rvaish@cs.ucsc.edu

² Stanford University HCI Group
Stanford, CA
{kwyngard, jingshuc, msb}@cs.stanford.edu

ABSTRACT

To lower the threshold to participation in crowdsourcing, we present *twitch crowdsourcing*: crowdsourcing via quick contributions that can be completed in one or two seconds. We introduce Twitch, a mobile phone application that asks users to make a micro-contribution each time they unlock their phone. Twitch takes advantage of the common habit of turning to the mobile phone in spare moments. Twitch crowdsourcing activities span goals such as authoring a census of local human activity, rating stock photos, and extracting structured data from Wikipedia pages. We report a field deployment of Twitch where 82 users made 11,240 crowdsourcing contributions as they used their phone in the course of everyday life. The median Twitch activity took just 1.6 seconds, incurring no statistically distinguishable costs to unlock speed or cognitive load compared to a standard slide-to-unlock interface.

Author Keywords

Crowdsourcing, microtasking, mobile crowdsourcing

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces. – Graphical User Interfaces.

INTRODUCTION

Mobilizing participation is a central challenge for every crowdsourcing campaign. Campaigns that cannot motivate enough participants will fail [25]. Unfortunately, many interested contributors simply cannot find enough time: lack of time is the top reason that subject experts do not contribute to Wikipedia [29]. Those who do participate in crowdsourcing campaigns often drop out when life becomes busy [21]. Even seemingly small time requirements can dissuade users: psychologists define *channel factors* as the small but critical barriers to action that have a disproportionate effect on whether people complete a goal [27]. Despite this constraint, many crowdsourcing campaigns assume that participants will work for minutes or hours at once, leading to a granularity

problem [3] where task size is poorly matched to contributors' opportunities. We speculate that a great number of crowdsourcing campaigns will struggle to succeed as long as potential contributors are deterred by the time commitment.

To engage a wider set of crowdsourcing contributors, we introduce *twitch crowdsourcing*: interfaces that encourage contributions of a few seconds at a time. Taking advantage of the common habit of turning to mobile phones in spare moments [24], we replace the mobile phone unlock screen with a brief crowdsourcing task, allowing each user to make small, compounded volunteer contributions over time. In contrast, existing mobile crowdsourcing platforms (e.g., [12,16,22]) tend to assume long, focused runs of work. Our design challenge is thus to create crowdsourcing tasks that operate in very short time periods and at low cognitive load.

To demonstrate the opportunities of twitch crowdsourcing, we present *Twitch*, a crowdsourcing platform for Android devices that augments the unlock screen with 1–3 second volunteer crowdsourcing tasks (Figure 1). Rather than a typical slide-to-unlock mechanism, the user unlocks their phone by completing a brief crowdsourcing task. Twitch is publicly deployed and has collected over eleven thousand volunteer contributions to date. The system sits aside any existing security passcodes on the phone.

Twitch crowdsourcing allows designers to tap into local and topical expertise from mobile users. Twitch supports three unlock applications:

1) *Census* envisions a realtime people-centered world census: where people are, what they are doing, and how they are doing it. For example, how busy is the corner café at 2pm on Fridays? Census answers these questions by asking users to share information about their surroundings as they navigate the physical world, for example the size of the crowd or current activities (Figure 1).

2) *Photo Ranking* captures users' opinion between two photographs. In formative work with product designers, we found that they require stock photos for mockups, but stock photo sites have sparse ratings. Likewise, computer vision needs more data to identify high-quality images from the web. Photo Ranking (Figure 1) asks users to swipe to choose the better of two stock photos on a theme, or contribute their own through their cell phone camera.



Figure 1: Twitch crowdsourcing asks for one to two seconds of users' time each time they unlock their phone. From left to right: Census (crowd size), Census (attire), Photo Ranking, and Structuring the Web.

3) *Structuring the Web* helps transform the written web into a format that computers can understand. Users specify an area of expertise — HCI, the Doctor Who television series, or anything else of interest on Wikipedia — and help verify web extractions relevant to that topic. Each unlock involves confirming or rejecting a short extraction. In doing so, users could power a fact-oriented search engine that would directly answer queries like “heuristic evaluation creator”.

After making a selection, Twitch users can see whether their peers agreed with their selection. In addition, they can see how their contribution is contributing to the larger whole, for example aggregate responses on a map (Figure 2) or in a fact database (Figure 5).

We deployed Twitch publicly on the web and attracted 82 users to install Twitch on their primary phones. Over three weeks, the average user unlocked their phone using Twitch 19 times per day. Users contributed over 11,000 items to our crowdsourced database, covering several cities with local census information. The median Census task unlock took 1.6 seconds, compared to 1.4 seconds for a standard slide-to-unlock gesture. Secondary task studies demonstrated that Twitch unlocks added minimal cognitive load to the user.

Our work indicates that it may be possible to engage a broad set of new participants in crowdsourcing campaigns as they go about their day or have a few spare moments. In the following sections, we introduce twitch crowdsourcing in more detail and report on our public deployment and field experiments.

RELATED WORK

Crowdsourcing has demonstrated the potential to support goals ranging from search engine support [6], classifier training [28], and even real-time interaction [5,19]. However, these systems often are limited in the amount of topical and local expertise they can assume. To address this limitation, txtEagle [12], mCrowd [30], and mClerk [16]

have sought to engage new sets of users through mobile crowdsourcing. However, these mobile crowdsourcing applications assume longer stretches of participation from crowd workers; twitch crowdsourcing aims to capture more incidental, ad-hoc opportunities. Bentley et al. characterize micro-moments by short (~10 second) time periods when attention can be taken away from a current task [1,4]. Twitch crowdsourcing demonstrates that these micro-moments are ideal platforms for interaction. Applications such as adaptive flashcards [13] operate on similar insights, but Twitch is unique in its design as an unlock screen.

Twitch: Census

Previous efforts have spread sensor-oriented crowdsourcing tasks through mobile phones [30]. Crowd-sensing applications likewise draw on humans-in-the-loop to trigger sensing actions or review results [26]. Scaling up sensor networks, MetroSense provides a people-centric paradigm for urban sensing [8]. Census also complements sensor technologies to track transportation habits [15] and mobile phone-based remote sensing platforms [11].

Twitch: Photo Ranking

User-generated photo voting platforms often succeed through a motivating goal, such as Kitten War [32]. However, while stock photography portals such as iStockPhoto [33] exist, none provide a vast collection of free to download images. In parallel, computer vision has demonstrated the importance of crowdsourcing large datasets [28]. Rankr [20] aims to rank images, ideas and priorities via a mobile phone application; however, it requires dedicated users and is not integrated into users' general flow. Photo Ranking is also inspired by the Matchin game, which demonstrates that twitch crowdsourcing can complement game strategies [17].

Twitch: Structuring the Web

Campaigns such as FreeBase [34] and dbPedia [35] have demonstrated the opportunities available when crowds help structure the information available on the web. Automatic

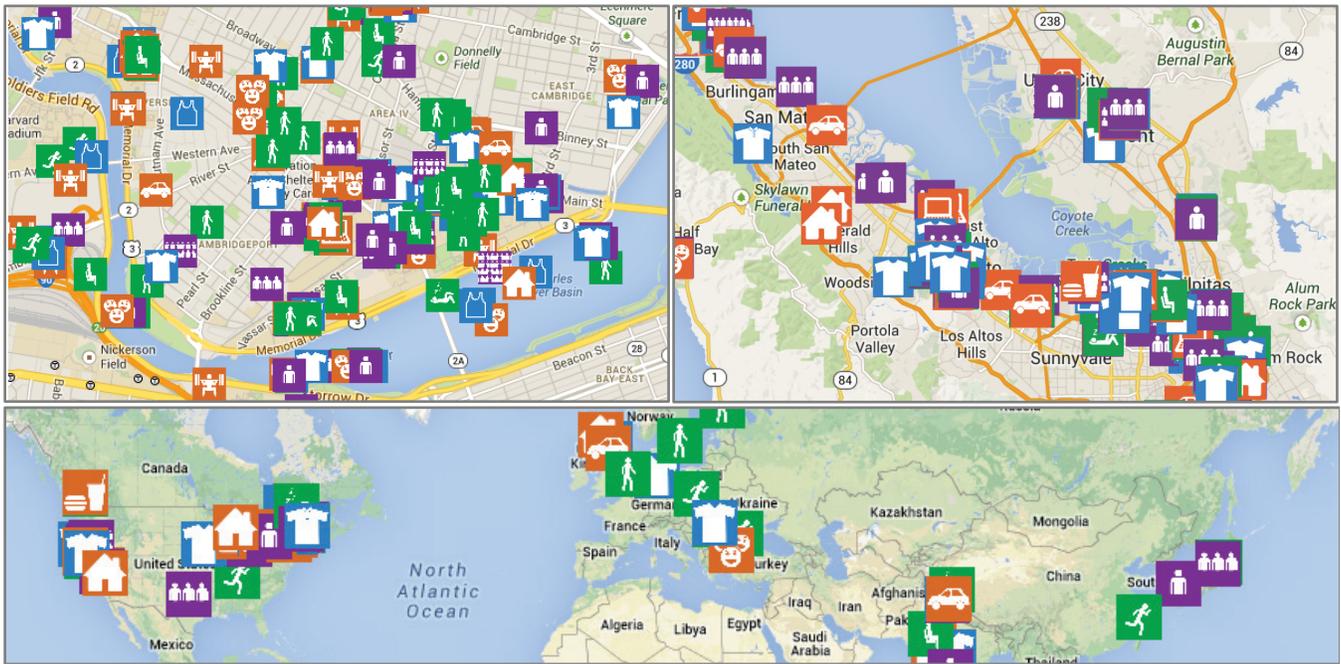


Figure 2: Subsamples from the worldwide Census activity generated by Twitch.
Upper left: Cambridge, Massachusetts. Upper right: Bay Area, California. Bottom: worldwide.

information extraction techniques such as ReVerb [14] can be augmented by crowd verification [18]. Twitch generalizes these interactive techniques and allows users to contribute to essentially any topic of interest.

TWITCH CROWDSOURCING

We address difficulties in mobilizing crowd participation by tapping into brief moments of users' time. While waiting for an elevator, a meeting, a walk signal, or cafeteria service, it can be difficult to resist glancing at the phone [24]. Rather than making this temptation harder or less desirable, we take advantage of these short bursts to encourage people to make small contributions towards solving impactful problems. We call this approach *twitch crowdsourcing*: engaging with participants in a convenient fashion in short bursts.

While there are many possible short bursts of time in users' daily lives, we focus on one that users already expect to be a small task: the mobile phone unlock screen. A slide-to-unlock gesture is useful for preventing accidental activation of the phone, but there is no reason why the unlock needs to be a slide gesture. We replace the unlock screen with a small crowdsourced task that takes roughly the same amount of time to complete. In doing so, the user contributes to crowdsourcing goals they want to support without taking extra time out of their day.

Twitch crowdsourcing interactions are very brief, so it is important that users can complete the tasks extremely quickly. Likewise, crowdsourcing is not the user's primary task, so these tasks must be lightweight and not distracting. We frame the design of our twitch crowdsourcing applications around the following principles:

- Availability: summon the task with a single button press
- Quick completion: contribute within a couple of seconds
- Low cognitive load: do not distract users from their primary task
- Value: volunteer toward a worthwhile goal
- Personal impact: maximize the uniqueness of the user's contributions [2]
- Longevity: keep tasks interesting after many repetitions
- Feedback: situate the user's effort as directly contributing to the larger crowdsourcing goal
- Privacy: do not circumvent standard phone security

To maximize value and personal impact, we focus not on paid crowdsourcing (e.g., Amazon Mechanical Turk) but on activities to which the user might want to volunteer their time. Because twitch crowdsourcing happens while users are mobile, tasks that take advantage of the user's local, physical context are particularly attractive. In addition, by lowering the threshold to contribution, we hypothesize that even busy topical experts might be able to contribute. Thus, we also explore opportunities to capture expertise, much like Wikipedia.

TWITCH

Twitch is an Android application that appears when the user presses the phone's power/lock button (Figures 1 and 3). When the user completes the twitch crowdsourcing task, the phone unlocks normally. Each task involves a choice between two to six options through a single motion such as a tap or swipe.

To motivate continued participation, Twitch provides both instant and aggregated feedback to the user. An instant

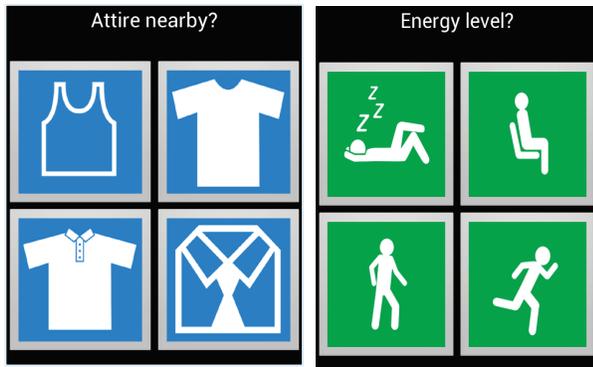


Figure 3: Census tasks, activity and energy level.

feedback display shows how many other users agreed via a fadeout as the lock screen disappears (Figure 4) or how the user's contributions apply to the whole (Figure 5). Aggregated data is also available via a web application, allowing the user to explore all data that the system has collected. For example, Figure 2 shows a human generated map from the Census application.

To address security concerns, users are allowed to either disable or keep their existing Android passcode while using Twitch. If users do not wish to answer a question, they may skip Twitch by selecting 'Exit' via the options menu. This design decision has been made to encourage the user to give Twitch an answer, which is usually faster than exiting. Future designs could make it easier to skip a task, for example through a swipe-up.

Below, we introduce the three main crowdsourcing applications that Twitch supports. The first, Census, attempts to capture local knowledge. The following two, Image Voting and Structuring the Web, draw on creative and topical expertise. These three applications are bundled into one Android package, and each can be accessed interchangeably through Twitch's settings menu.

Census

Despite progress in producing effective understanding of static elements of our physical world — routes, businesses and points of interest — we lack an understanding of human activity. How busy is the corner cafe at 2pm on Fridays? What time of day do businesspeople clear out of the downtown district and get replaced by socializers? Which neighborhoods keep high-energy activities going until 11pm, and which ones become sleepy by 6pm? Users could take advantage of this information to plan their commutes, their social lives and their work.

Existing crowdsourced techniques such as Foursquare are too sparse to answer these kinds of questions: the answers require at-the-moment, distributed human knowledge. We envision that twitch crowdsourcing can help create a human-centered equivalent of Google Street View, where a user could browse typical crowd activity in an area. To do so, we ask users to answer one of several questions about

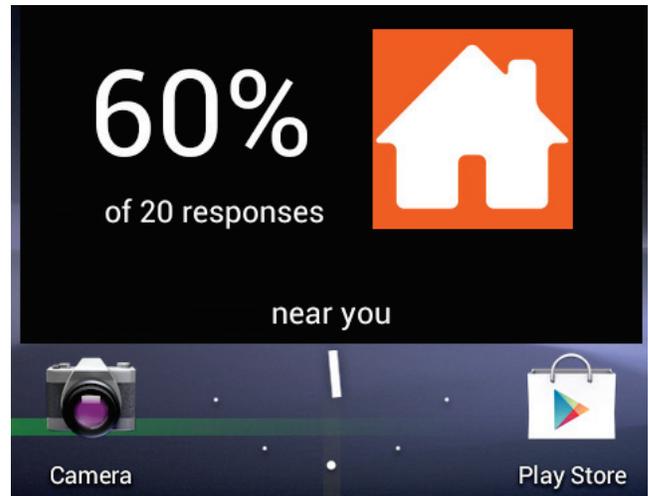


Figure 4: Instant feedback shows people agreeing with the user in her location.

the world around them each time they unlock their phone. Users can then browse the map they are helping create.

Census is the default crowdsourcing task in Twitch. It collects structured information about what people experience around them. Each Census unlock screen consists of four to six tiles (Figures 1 and 3), each task centered around questions such as:

- How many people are around you?
- What kinds of attire are nearby people wearing?
- What are you currently doing?
- How much energy do you have right now?

While not exhaustive, these questions cover several types of information that a local census might seek to provide. Two of the four questions ask users about the people around them, while the other two ask about users themselves; both of which they are uniquely equipped to answer. Each answer is represented graphically; for example, in case of activities, users have icons for working, at home, eating, travelling, socializing, or exercising.

To motivate continued engagement, Census provides two modes of feedback. *Instant feedback* (Figure 4) is a brief Android popup message that appears immediately after the user makes a selection. It reports the percentage of responses in the current time bin and location that agreed with the user, then fades out within two seconds. It is transparent to user input, so the user can begin interacting with the phone even while it is visible. *Aggregated report* allows Twitch users to see the cumulative effect of all users' behavior. The data is bucketed and visualized on a map (Figure 2) on the Twitch homepage. Users can filter the data based on activity type or time of day.

Photo Ranking

Beyond harnessing local observations via Census, we wanted to demonstrate that twitch crowdsourcing could support traditional crowdsourcing tasks such as image

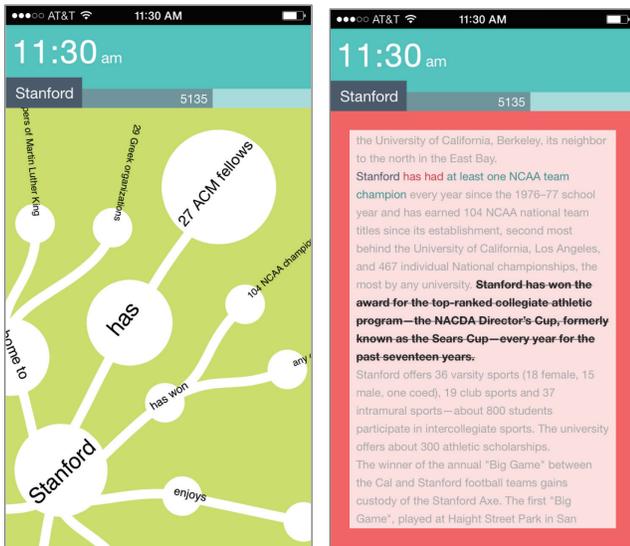


Figure 5: Verifying a web extraction animates a new branch onto the knowledge graph (left). Rejecting the extraction shows how far the user is in the article (right).

ranking (e.g., Matchin [17]). Needfinding interviews and prototyping sessions with ten product design students at Stanford University indicated that product designers not only need photographs for their design mockups, but they also enjoy looking at the photographs. Twitch harnesses this interest to help rank photos and encourage contribution of new photos.

Photo Ranking crowdsources a ranking of stock photos for themes from a Creative Commons-licensed image library. The Twitch task displays two images related to a theme (e.g., Nature Panorama) per unlock and asks the user to slide to select the one they prefer (Figure 1). Pairwise ranking is considered faster and more accurate than rating [17]. The application regularly updates with new photos.

Users can optionally contribute new photos to the database by taking a photo instead of rating one. Contributed photos must be relevant to the day's photo theme, such as Nature Panorama, Soccer, or Beautiful Trash. Contributing a photo takes longer than the average Twitch task, but provides an opportunity for motivated individuals to enter the competition and get their photos rated.

Like with Census, users receive instant feedback through a popup message to display how many other users agreed with their selection. We envision a web interface where all uploaded images can be browsed, downloaded and ranked. This data can also connect to computer vision research by providing high-quality images of object categories and scenes to create better classifiers.

Structuring the Web

Search engines no longer only return documents — they now aim to return direct answers [6,9]. However, despite massive undertakings such as the Google Knowledge Graph [36], Bing Satori [37] and Freebase [7], much of the knowledge on the web remains unstructured and

unavailable for interactive applications. For example, searching for 'Weird Al Yankovic born' in a search engine such as Google returns a direct result '1959' drawn from the knowledge base; however, searching for the equally relevant 'Weird Al Yankovic first song', 'Weird Al Yankovic band members', or 'Weird Al Yankovic bestselling album' returns a long string of documents but no direct answer, even though the answers are readily available on the performer's Wikipedia page.

To enable direct answers, we need *structured* data that is computer-readable. While crowdsourced undertakings such as Freebase and dbpedia have captured much structured data, they tend to only acquire high-level information and do not have enough contributors to achieve significant depth on any single entity. Likewise, while information extraction systems such as ReVerb [14] automatically draw such information from the text of the Wikipedia page, their error rates are currently too high to trust. Crowdsourcing can help such systems identify errors to improve future accuracy [18]. Therefore, we apply twitch crowdsourcing to produce both structured data for interactive applications and training data for information extraction systems.

Contributors to online efforts are drawn to goals that allow them to exhibit their unique expertise [2]. Thus, we allow users to help create structured data for topics of interest. The user can specify any topic on Wikipedia that they are interested in or want to learn about, for example HCI, the Godfather films, or their local city. To do so within a one-to-two second time limit, we draw on mixed-initiative information extraction systems (e.g., [18]) and ask users to help vet automatic extractions.

When a user unlocks his or her phone, Structuring the Web displays a high-confidence extraction generated using ReVerb, and its source statement from the selected Wikipedia page (Figure 1). The user indicates with one swipe whether the extraction is correct with respect to the statement. ReVerb produces an extraction in Subject-Relationship-Object format: for example, if the source statement is "Stanford University was founded in 1885 by Leland Stanford as a memorial to their son", ReVerb returns {Stanford University}, {was founded in}, {1885} and Twitch displays this structure. To minimize cognitive load and time requirements, the application filters only include short source sentences and uses color coding to match extractions with the source text.

In Structuring the Web, the instant feedback upon accepting an extraction shows the user their progress growing a knowledge tree of verified facts (Figure 5). Rejecting an extraction instead scrolls the user down the article as far as their most recent extraction source, demonstrating the user's progress in processing the article. In the future, we envision that search engines can utilize this data to answer a wider range of factual queries.

FIELD DEPLOYMENT AND EVALUATION

This paper hypothesizes that short bursts of crowdsourcing can be embedded in users' everyday activities. We focus our evaluation on the two main components of this claim. First, *is it possible to crowdsource meaningful tasks in short bursts of time?* Can Twitch attract sufficient volunteer usage to achieve its goals via short interaction bursts? Second, *it is possible to embed these tasks in users' everyday activities?* Are the tasks sufficiently quick and easy that users can complete them without excessive burden?

To answer these questions, we pursued two evaluation strategies. First, we publically deployed Twitch and attracted 82 volunteer users over one month. We used this data to investigate naturalistic usage, collect crowdsourced data, and gather qualitative feedback. Second, we performed two field experiments on Twitch users to compare Twitch's time cost to a standard slide-to-unlock gesture, and a study of cognitive load using a working memory instrument called the 2- and 3-back test [23].

Android Deployment

To examine Twitch's ability to attract naturalistic crowdsourced data, we began with a public deployment. In doing so, we focused on unlock statistics, timing data, location demographics, and duration of usage.

Method

We released Twitch on Google Play, a public marketplace for Android applications. We then used social networks and email lists to spread word of the application to individuals who would be willing to try the research prototype. This process resulted in 82 volunteer users installing the application and completing at least one Twitch task. While Twitch contained all of the applications, its focus and default setting was Census.

The application saved a timestamp each time a user pressed the power/unlock button and launched a Twitch unlock, and another timestamp each time the user completed a Twitch task. To address privacy concerns, we tied all data to a randomly generated phone identifier. Finally, to complement our quantitative metrics, we emailed all users of Twitch roughly three weeks after they began using the application with an invitation to complete a brief survey about their experiences.

Results

During the study period, 82 participants completed 11,240 Twitch tasks to unlock their phones. The task completion rate was 37.4%; other times, users pressed the power button but did not complete the task, most likely just checking the time or notifications without unlocking the phone.

Active Twitch users kept using the application for a significant length of time. As expected for free smartphone applications on a public marketplace, many window shoppers dropped off early: about half (46.4%) of those who downloaded Twitch within the study period uninstalled

it within a day. However, among the remaining users who kept the application for at least one day, the average duration of use was 31.9 days (median 30.9 days). One month of usage indicates a significant time and effort investment from these volunteers.

Census, the default Twitch application, received by far the most attention, with 9,717 responses from around the world (Figure 2), including the United States, Europe, India and Asia. The cities with the greatest number of responses were Cambridge, MA; Stanford, CA; and Pittsburgh, PA. Large numbers of responses also came from India and Japan. The resulting data is quite densely populated (Figure 2). The data indicates insights such as when cafes shift from work activities to social life, how many people are in a building early in the morning and late at night, and where people who work in downtown tend to go to exercise.

Roughly one fifth of users switched to try the Structuring the Web application, resulting in 334 responses. Across a range of topics such as "Google", "Android", "Earth", and "United States", Twitch users evaluated 181 unique statements from Wikipedia run through ReVerb. Users marked 41.4% of total extractions as correct.

At the end of the field study, we sent an online survey to 57 users who optionally provided their emails when installing the application. Sixteen participants filled out the survey in three days, with thirteen participants from the United States and three participants from Panama, Canada, and India. Participants generally enjoyed using Census (median Likert score 5 of 7), and found it only slightly distracting (median = 3.5). Participants who self-selected to switch to Image Ranking enjoyed it as well (median = 6).

Twitch discourages skipping tasks — it involves a two-tap exit via a context menu — making it possible that some users would deliberately answer incorrectly just to quickly unlock their phone. However, most participants reported making their selections honestly and accurately (median = 6). Roughly half of participants (56%) never felt the need for a skip button or gesture.

In order to motivate our users to continue using Twitch, and to encourage longevity, we have complemented all the applications with immediate feedback that tells users the percentage of others agreeing. Users moderately enjoyed the popup feedback (median = 5.5). In addition, we provided a web interface where users are able to view their data on a crowdsourced activity map; participants reported that the site had some effect on motivating them to contribute further (median = 5).

Open ended responses reinforced that the application was "very lightweight [...] I'm sure I answered over 100 questions effortlessly." One area of improvement: when users were not moving, Census tasks in particular became repetitive and thus annoying. Some participants wished the immediate feedback to be richer or less precisely tuned to their location so that it reports more data. A few

Field deployment Twitch unlock times (N=11,014)

Activity	Median	Mean
Slide-to-unlock	1421 ms	1956 ms
Census: people	1583 ms	2077 ms
Census: activity	1910 ms	2077 ms
Census: energy	1688 ms	2177 ms
Census: dress	1499 ms	2033 ms
Photo ranking	1579 ms	2438 ms
Structuring the web	2900 ms	3510 ms

Table 1. Answer times for Twitch usage in the wild.

participants were conscious of their location identification. Those users who used Image Ranking enjoyed the experience, but as the application switching was not particularly discoverable (a design decision to make the public application simpler), many weren't aware of its existence. One of the users shared her experience: "The beautiful pictures brought happiness to the moment".

Most participants felt that concerns about privacy were addressed (median = 5); however, some users uninstalled the application due to privacy concerns with location tracking. Redundant tasks and privacy concerns were the core reasons reported for uninstalling the application.

Participants' biggest requests for improving Twitch involved adding variety to the tasks or making tasks appear less frequently. Others wanted more intelligent automatic task selection: we believe this may be possible using the onboard phone sensors.

Field Experiments: Speed and Cognitive Load

For twitch crowdsourcing to succeed, tasks must be quick and low in cognitive load. If a task is too slow, users will opt out or give quick, unthoughtful responses. Likewise, if a task requires too much mental effort, users will have to disengage from whatever they are doing just to unlock their phones, which is undesirable. In a pair of field experiments tied to our deployment, we investigate the impact that Twitch has on the speed of unlocking users' phones and on cognitive load.

Method: Speed Experiment

To compare unlock times to a standard unlock interface; we included a simple slide-to-unlock screen in Twitch when the application was publicly deployed. Slide-to-unlock is similar to the default Android screen lock (called Slider). Thus, slide-to-unlock acts as a speed baseline to compare with rest of the Twitch applications. Once a week, Twitch displayed the slide-to-unlock screen instead of the selected Twitch task, along with a message stating that this temporary unlock interface was part of our field experiment. We compare user speed on the slide-to-unlock test against the rest of Twitch applications, examining whether Twitch tasks are noticeably slower or faster than a standard Android unlock. Speed is calculated in milliseconds from the moment Twitch is visible until the user completes the task.

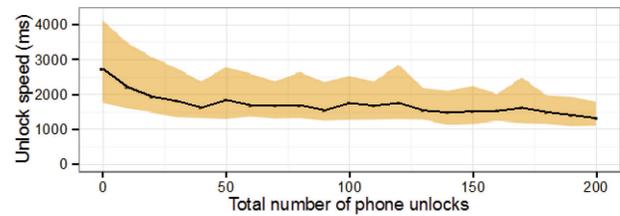


Figure 6. Response times decline over the first forty unlocks. The line indicates the median; band is the interquartile range.

Results

In naturalistic usage via a dataset of 11,014 unlocks (Table 1), slide-to-unlock took users a median of 1.39 seconds to unlock their phone (average 1.89). Median Census unlocks took between 1.49–1.89 seconds (average 2.01–2.49 seconds). Photo Ranking took 1.58 seconds (average 2.44). The median Structuring the Web unlock took 2.09 seconds (average 2.98 seconds), as it requires more reading.

We next tested whether the Twitch unlocks were in fact slower than a standard slide-to-unlock interface. As is common with reaction time data, Twitch duration data are not normally distributed; even after transformation, the data were heteroskedastic and thus not amenable to parametric statistics. Thus, we use the Friedman test on medians, a nonparametric equivalent of the ANOVA with blocked designs. We compared median unlock times between Twitch tasks for users who completed at least 100 unlocks and tried all Census activities, excluding Structuring the Web and Photo Ranking because fewer users had tried it (N=22 users). The Friedman test revealed a significant effect of Twitch activity on unlock time ($\chi^2(4)=24.4$, $p<.001$). Post-hoc paired Wilcoxon signed rank tests using Bonferroni correction between conditions revealed that the "current activity" Census task was significantly slower than all other Census tasks and slide-to-unlock (all $p<.05$), but there were no other significant differences between conditions.

Thus, while Twitch tasks may be slightly slower than a standard slide-to-unlock gesture (100–300ms), the difference is so slight in comparison to the variance that the two are not currently statistically distinguishable. There is clearly variability across Twitch tasks: Census's current activity task is 300–400ms slower than the others, possibly because it is the only categorical (non-scalar) response amongst the Twitch tasks, which impacts visual search.

Users quickly acclimated to Twitch, with median response time dropping from 2.7 seconds in the first ten unlocks to 1.6 seconds by the fortieth unlock (Figure 6). Given that users are unlocking their phones many times each day, they approach optimal performance quickly.

Method: Cognitive Load Experiment

To understand whether twitch might distract or annoy users, we turn to measures of cognitive load. In particular, we adopt a working memory task known as the 2-back and 3-back test [10,23]. The 2-back test displays a stream of

English letters, one at a time, and the participant must indicate for each letter whether it is the same as the letter that was displayed two letters ago. For example, if the previous letter stream were E, U, S, I, and the current letter were Y, the correct answer would be ‘Different’; if the current letter were S, the answer would be ‘Same’. Following previous work [10], we sample so that the correct answer is ‘Same’ with probability 1/7. The 3-back test is identical to the 2-back test except the distance is three letters instead of two, making the task more difficult. To further load working memory, we likewise inject a Twitch Census activity or a Photo Ranking activity with probability 1/7 between letters. Twitch activities add to cognitive load but do not impact the correct answer — they do not count as a letter.

We performed a within-subjects field experiment to test how much each Twitch activity would negatively impact performance on the memory task. For each round of the study, the participant performed a series of 2- or 3-back tasks with random Twitch activity injections until they had completed all of the Twitch activities. One study factor was Twitch activity: we randomized the order of Twitch unlock screens that got injected into the task stream, including a slide-to-unlock control, for each round of the study. The second factor was memory task: participants rotated through rounds of 2-back or 3-back tasks until all Twitch activities had been tested in each condition. We randomized whether 2-back or 3-back tasks appeared first for each participant. The study continued for six rounds, three in each memory task condition. The study typically took fifteen minutes.

We recruited participants from the set of active Twitch users, so they would already be familiar with the applications. In exchange, participants received a raffle ticket for an Android tablet. Drawing on prior work [24,31], we instructed participants to go on a walk outdoors following a predefined path at one of two major universities or (alternatively) following any safe route in their neighborhood. In forcing participants to walk, we made sure that users would need to look around to answer Census questions.

We measured reaction time and task accuracy on the memory task. In particular, we grouped each reaction time and accuracy measurement with the Twitch activity that occurred just before the task. Thus, we measured how much of an impact each unlock screen had on the next 2- or 3-back instance, compared to baselines of no unlock (3-back instance preceded by a 3-back instance) and slide-to-unlock.

Our hypothesis was that Twitch would incur a cost to working memory accuracy and reaction time compared to a slide-to-unlock or no unlock gesture. However, we hypothesized that this cost for most Twitch tasks would be similar to the cost of a simple slide-to-unlock.

3-back task performance (N=5405 tasks)

Previous Screen	Median	Mean	Accuracy
Back task	1194 ms	1802 ms	90%
Slide-to-unlock	1724 ms	2228 ms	87%
Photo ranking	1506 ms	2009 ms	86%
Census: people	1737 ms	2332 ms	87%
Census: activity	1679 ms	2000 ms	93%
Census: energy	1881 ms	2238 ms	93%
Census: dress	2229 ms	2445 ms	95%

Table 2. Median and mean delay times for answering 3-back tasks in the cognitive load study. 2-back results are similar.

Results

Fourteen Twitch users completed the study. The population was worldwide, centered in areas with the most Twitch users. Participants completed 5,405 back tasks.

As with many reaction time measurements, timing data was non-normal after dropping outliers. Unlike the previous study, it was possible to retain homogeneity of variance and approach normality by applying a transformation. Such transformations are common when analyzing reaction time data. We use a generalized power transformation called a Box-Cox transformation, which algorithmically searches for the power λ that best fits a normal distribution. For our data, $\lambda=-.25$, a reciprocal fourth root transformation of the data. We transform timing data for our statistical analysis, but report raw milliseconds to aid interpretability.

The median task took 1.2 seconds to answer for both the 2-back and 3-back conditions when no Twitch activities were interposed (mean 1.68 for 2-back and 1.80 for 3-back, Table 2). Slide-to-unlock caused the task to take a median 1.7 seconds (mean 2.2). All Twitch tasks except Census clothing had similar delays, 1.7–1.8 seconds (mean 2.0–2.3). We performed a two-way ANOVA controlling for participant, using transformed delay as dependent variable, and memory task and Twitch activity as independent variables. The ANOVA found a significant main effect of Twitch activity ($F(6, 5378)=25.8, p<.001$). Post-hoc Tukey comparisons found no significant differences in delay between any Twitch activities and the slide-to-unlock control (all $p>.55$). All activities, including the slide-to-unlock control, were significantly slower than the previous task being another back task (all $p<.001$).

We compared accuracy across conditions using a logistic regression, again controlling for participant. Memory task was significant ($\beta=-0.45, z=-4.3, p<.001$), indicating that 3-back tasks produced less accurate performance than 2-back tasks. However, none of the Twitch activities or the control slide-to-unlock activity had a significant impact on accuracy.

Thus, contrary to our hypothesis, Twitch activities had no measurable impact on accuracy or delay for the working memory task compared to a standard slide-to-unlock gesture. While unexpected, this is good news for Twitch:

Twitch activities may not meaningfully distract the user from their daily activities while using their phone.

DISCUSSION

Twitch crowdsourcing can extend beyond the three applications outlined in this paper: opportunities include citizen science, accessibility, education, and scientific discovery. Users might participate in micro-tutoring sessions by marking an English-language sentence as correct or incorrect to help language learners from the developing world and K-12 education. They might answer short survey questions (e.g., txtEagle and Jana [12]), help power accessible interfaces for other users, collate local news, or direct lost people in their neighborhood. Of course, while we have focused on volunteer crowdsourcing, it may be possible to add a microtask marketplace to Twitch so that users can make money as they go about their day.

Longevity and retention are challenges for every crowdsourcing campaign. In our deployment, the median Twitch user was active for one month. (One month out of a forty-day field deployment period is longer than we expected for users of research software.) Moving forward, we believe that the key to long-term usage is finding design opportunities that match users' intrinsic motivation. Similar to gamification, twitch crowdsourcing cannot make an uninteresting application more attractive, but it can empower an already viable design. Our results indicated that users would be more likely to continue using the application if they see their progress and value toward the larger goal. In response, we have been developing far richer feedback for the Structuring the Web application (Figure 5).

In our vision of twitch crowdsourcing, tasks are fast and impose little cognitive load. However, many worthwhile crowdsourcing goals may not easily fit this definition. Structuring the Web is nearing this limit, taking roughly three seconds per unlock. We believe that it may be possible to break longer task into smaller ones that can be completed serially across multiple screens. This might enable a user to execute a long task through multiple unlocks, with each fragment taking less than 2 seconds.

Reciprocally, some users have slightly longer stretches of time to contribute, for example while waiting for a meeting to start. Twitch crowdsourcing could apply in these situations as well, for example by allowing users to launch a standalone application or ask the unlock screen to keep cycling through new tasks without disappearing. To maintain users' desired level of security, Twitch allows users to add its unlock screen after an Android native unlock. The effect is extremely similar to the passcode unlock on the Apple iPhone: users must both swipe to unlock and enter a passcode. Moving forward, we believe that it would be possible to create Twitch tasks that could function as security locks as well. One possible approach would be to integrate a user-determined security swipe pattern into the motion needed to complete a Twitch task. For example, users could tap to indicate their answer, and

then swipe a gesture password from that starting position to verify their identity.

Our deployment underscored that a stream of fresh tasks is required to keep Twitch interesting. This is possible with applications like Structuring the Web, but Census suffered from boredom in repeat usage. The system could be extended to assign microtasks more intelligently. For example, location-based tasks should be distributed among nearby users and topic-based text extractions and other tasks should be distributed among experts and enthusiasts in each field. Private groups could split up and share their own tasks among friends and coworkers. Likewise, the system could learn to regularly swap stale tasks for fresh ones.

CONCLUSION

This paper presents *twitch crowdsourcing*, an approach to crowdsourcing based on quick microtasks that encourage short bursts of contribution. Our Twitch application augments the lock screen on mobile phones to harness users' spare moments for short tasks. Unlocking a mobile device via a Twitch microtask takes tenths of a second longer than a standard unlock — so little that the two were not statistically distinguishable in our field experiment. Our study, designed to determine the mental workload imposed by Twitch, revealed that our twitch crowdsourcing tasks likewise do not impose a more significant cognitive load than a simple swipe-to-unlock motion. In a public deployment, Twitch collected over eleven thousand data points to help build a census of local activity, rank stock photos, and structure the web.

Reducing concerns about effort, time, and motivation are key steps to greater volunteerism in crowdsourcing. We suggest that small contributions made in short bursts during spare seconds can be aggregated to accomplish significant, customized tasks without placing an undue burden on volunteers. We envision that this approach could bring experts into the crowdsourcing fold, overcoming their historical tendency to stay out because of large time commitments. If we succeed in involving a broader set of participants and topic experts, they could unlock many new opportunities for research and practice in crowdsourcing.

ACKNOWLEDGEMENTS

The authors thank the Stanford HCI Group and MIT UID Group for testing in-progress prototypes. This work was supported by Google Research and HTC.

REFERENCES

1. Anttila, A., Experience, U., Corporation, N., Jung, Y., Specialist, D., and Tower, A. Discovering Design Drivers for Mobile Media Solutions. *Proc. CHI EA '06*, (2006).
2. Beenen, G., Ling, K., Wang, X., et al. Using social psychology to motivate contributions to online communities. *Proc. CSCW '04*, (2004).

3. Benkler, Y. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, 2007.
4. Bentley, F., Tullio, J., Metcalf, C., Harry, D., and Massey, N. A time to glance: Studying the use of mobile ambient information. *Proc. Pervasive '07 Workshop: Ambient Information Systems*, (2007).
5. Bernstein, M.S., Brandt, J., Miller, R.C., and Karger, D.R. Crowds in two seconds: enabling realtime crowd-powered interfaces. *Proc. UIST'11*, (2011).
6. Bernstein, M.S., Teevan, J., Dumais, S., et al. Direct Answers for Search Queries in the Long Tail. *Proc. CHI'12*, (2012).
7. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge. *Proc. SIGMOD '08*, (2008).
8. Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., and Peterson, R.A. People-centric urban sensing. *Proc. WICON '06*, (2006).
9. Chilton, L.B. and Teevan, J. Addressing People's Information Needs Directly in a Web Search Result Page. *Proc. WWW 2011*, (2011).
10. Cohen, J., Forman, S., Braver, T., Casey, B., Servan-Schreiber, D., and Noll, D.C. Activation of the Prefrontal Cortex in a Nonspatial Working Memory Task With Functional MRI. *Human Brain Mapping I*, (1994), 293–304.
11. Das, T. PRISM: Platform for Remote Sensing using Smartphones. *Proc. MobiSys '10*, (2010).
12. Eagle, N. txteagle: Mobile Crowdsourcing. *Proc. IDGD '09*, (2009).
13. Edge, D., Whitney, M., and Charlotte, C. MemReflex: Adaptive Flashcards for Mobile Microlearning. *Proc. MobileHCI '12*, (2012).
14. Fader, A., Soderland, S., and Etzioni, O. Identifying relations for open information extraction. *Proc. EMNLP '11*, (2011).
15. Froehlich, J., Dillahunt, T., Klasnja, P., et al. UbiGreen: Investigating a Mobile Tool for Tracking and Supporting Green Transportation Habits. *Proc. CHI'09*, (2009).
16. Gupta, A., Thies, W., Cutrell, E., and Balakrishnan, R. mClerk: Enabling Mobile Crowdsourcing in Developing Regions. *Proc. CHI'12*, (2012).
17. Hacker, S. and Ahn, L. Von. Matchin: Eliciting User Preferences with an Online Game. *Proc. CHI'09*, (2009).
18. Hoffmann, R., Amershi, S., Patel, K., Wu, F., Fogarty, J., and Weld, D.S. Amplifying community content creation with mixed initiative information extraction. *Proc. CHI '09*, (2009).
19. Lasecki, W., Miller, C., and Bigham, J.P. Warping time for more effective real-time crowdsourcing. *Proc. CHI'13*, (2013).
20. Luon, Y., Aperjis, C., and Huberman, B.A. Rankr: A Mobile System for Crowdsourcing Opinions. *Springer Berlin Heidelberg*, (2012).
21. Luther, K. and Bruckman, A. Leadership in online creative collaboration. *Proc. CSCW '08*, (2008).
22. Narula, P., Gutheim, P., Rolnitzky, D., Kulkarni, A., and Hartmann, B. MobileWorks: A Mobile Crowdsourcing Platform for Workers at the Bottom of the Pyramid. *Proc. HCOMP '11*.
23. Ophir, E., Nass, C., and Wagner, A.D. Cognitive control in media multitaskers. *PNAS 106*, 37 (2009).
24. Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J. Interaction in 4-Second Bursts: The Fragmented Nature of Attentional Resources in Mobile HCI. *Proc. CHI '05*, (2005).
25. Poteete, A.R., Janssen, M., and Ostrom, E. *Working together: collective action, the commons, and multiple methods in practice*. Princeton University Press, 2010.
26. Ra, M., Liu, B., Porta, T. La, and Govindan, R. Medusa: A Programming Framework for Crowd-Sensing Applications Categories and Subject Descriptors. *Proc. MobSys'12*, (2012).
27. Ross, L. and Nisbett, R.E. *The person and the situation: Perspectives of social psychology*. McGraw-Hill, 1991.
28. Su, H., Deng, J., and Fei-fei, L. Crowdsourcing Annotations for Visual Object Detection. *Proc. HCOMP'12*, (2012).
29. Taraborelli, D., Mietchen, D., Alevizou, P., and Gill, A. Expert participation on Wikipedia: barriers and opportunities. *Wikimania 2011*, (2011).
30. Yan, T., Marzilli, M., Holmes, R., Ganesan, D., and Corner, M. mCrowd - A Platform for Mobile Crowdsourcing. *Proc. SenSys '09*, (2009).
31. Yu, C.-H. Mobile continuous reading. *Ext. Abs. CHI '12*, (2012).
32. Kitten War. <http://www.kittenwar.com/>.
33. iStockPhoto. <http://www.istockphoto.com>.
34. Freebase. <http://www.freebase.com/>.
35. dbPedia. <http://dbpedia.org/>.
36. Google Knowledge Graph. <http://goo.gl/6cCKVP>.
37. Microsoft Bing Satori. <http://binged.it/10nFdH0>.