PhotoCal - Mediating the Divide Between Digital Calendars and Physical Event Advertisements

Lucas Ochoa
Carnegie Mellon University
Pittsburgh, Pennsylvannia
lbo@andrew.cmu.edu

Gautam Bose
Carnegie Mellon University
Pittsburgh, Pennsylvania
gbose@andrew.cmu.edu

Juho Kim
KAIST
Daejeon, South Korea
juho.kim@kaist.ac.kr

ABSTRACT
Digital calendars provide great functionality in terms of collaboration, working across many devices, and synchronization with digital event services and social media platforms like Facebook. However, many events choose to advertise through posters which afford advantages of low cost, a high degree of customizability, and a low technical barrier to creation and distribution. In our iOS prototype, we propose new crowdsourced based workflow that allows users to quickly have a new event ready to add to their calendar in a single tap. We also demonstrate how we can use a database of posters to intrinsically motivate users to label posters. Finally, we show a novel interface that drastically lowers the friction that comes with entering in calendar information from a poster. During a small scale testing period of 3 days we saw a total of 13 posters uploaded and labeled by users.

Author Keywords
Crowdsourcing; Events; Calendars; Interaction Design;

INTRODUCTION
Online calendars have become ubiquitous for both modern and personal use. They provide flexibility and integration with collaborative tools and social media event services, the largest of which being Facebook [3]. However, people trying to attract others to their event often still create physical artifacts like posters in addition to their digital counterpart on Facebook.

These posters are fantastic for those trying to attract people to an event because they can easily and cheaply be placed in high traffic areas. Additionally, with basic image editing skills, many people can create high quality and striking posters that grab peoples attention. This is difficult to do on most web based event platforms as they all require a certain format, while a poster can have anything you like on it. This leads to a poor user experience, where users are stuck filling out the many fields required for an event while having to stop in front of the poster.

To prevent this, some users simply take a picture of the poster with the goal to enter it into their calendar later, but current calendar interfaces are not designed for this kind of input and require a lot of switching apps, which has been shown to be an activity that requires high cognitive load [1].

In this work, we present our iOS prototype called PhotoCal - The Photo Calendar. This application uses crowdsourcing to provide labeled posters to users so they can add them to their calendar in a single tap.

To illustrate the utility of our system, we'll take the example of two college students "Grace" and "Soonho". Soonho is walking to class when a poster catches his eye in the hallway. He has just enough time to get to class, so he quickly takes out his phone and snaps a picture with PhotoCal. From there the photo is uploaded to our "Public Feed" of posters.

Grace is in her dorm room and wants to find a fun event that she can attend with her friend that is coming in from out of town. She opens PhotoCal to browse the public feed; the poster that Soonho uploaded draws her attention. She presses the 'Add' button next to the photo, as if to add it to her calendar. Due to the fact that nobody has labeled this poster, instead of adding it to her calendar she is presented with a modal instructing her to label the poster before she can add it. She quickly enters in the posters information, which is a low friction experience due to the fact that she can zoom and pan on the poster while typing. After she clicks confirm and the event goes into her calendar.

Soonho checks back into the app to see that his poster now has event information labeled. He simply clicks 'Add' and now the appropriate calendar information for the event has been added to his calendar.

We have tested this using a small scale roll out to friends and family users, distributed using Apple's TestFlight program. Over the course of 3 days, we saw a total of 13 posters uploaded by users.

BACKGROUND AND RELATED WORK
The primary way that others have attempted to tackle this problem is through computer vision. Zhang Y. [2] attempted a similar project using computer vision, but ran into severe difficulty on all but the clearest posters. This points to one of the biggest challenges we identified for a computer

vision or OCR system: the combination of highly variable as well as highly unstructured information that could be present on a poster. Zhang Y.'s results showed that CV and OCR systems are not equipped to deal with the kind of variability that occurs on posters. This was apparent from their findings that posters with complex backgrounds or handwritten text consistently failed to be recognized by their system.

Model of Motivation

One of the main references for the design of our worker motivation system was the website Glassdoor [4]. The website's primary functionality is to provide a platform for users to write reviews and salary information about their employers. This lets other users read firsthand accounts about what working at different companies is actually like.

Glassdoor prohibits users from seeing more than a certain number of reviews until they themselves upload a review. Essentially they hook a users attention by presenting a tantalizing piece of information, but then make the user contribute before giving out that information. We theorize that this system coerces people to contribute in ways that they might otherwise not due to the fact that they want information that has been so teased by the website.

SYSTEM

Our system consists of an iOS app which connects to Firebase for a real time database as well as online storage of images.

The application can be broken into 3 main parts: a feed which doubles as doubles as motivation for crowd workers, a novel poster labeling interface, and an uploading screen. In this section we will discuss the details of each part of the interface, as well as as an overview of how our back end implementation works.

Feed

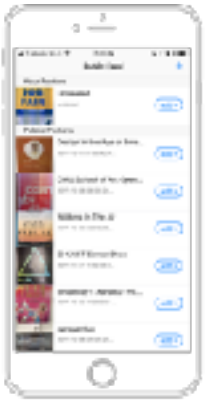

Figure 1. A depiction of our main feed.

The main feed shown in Fig 1. serves as the home landing page of the application. Clicking on a poster will allow them to view a larger version of the image to see the event details if it is labeled. Clicking the add button will either simply add the event to the users calendar or bring to the labelling screen, depending on whether the poster is labeled or not. There are two key features that allow this part of the interface to perform all of its necessary functions.

First, we separate the users personal photos from the "Public Feed" section of the app. This allows people to quickly determine when their poster has been labeled. This divider is also key to separate posters that are meant to be 'browsed' and those that the user is waiting on to label.

The Glassdoor Model of Motivation

We wanted to implement a similar model of motivation as described earlier in the section on the website Glassdoor. To do so, we display the same 'Add +" button next to every post in the list. This is a small but critical detail, as it is what affords our model of motivation.

We theorize that users may not actually want to label the posters if we outright ask them to. Initial versions of the application displayed a 'Label +' button if the poster was unlabeled and only displayed the 'Add +' button when they were truly ready to be added to the calendar. This early iteration failed to take into account the anticipation that comes with hitting the 'Add +' button. If users are interested in a poster and then see the 'Label +' button, they may begin to weigh their interest in having the event in their calendar against the work required for labelling.

However, by displaying the 'Add +' button, if a user is interested, they will click it automatically. Then if they are prompted with the labelling screen, they will label the poster because of the anticipation and interest they have in the poster.

Additionally, we feel that this actually reduces the cognitive load on the user. This way they are not browsing the feed and making a decision based on two factors (the event itself and whether it is labeled or not) but instead one, only the event. Should they click 'Add +' on an unlabeled poster, they are brought to the labelling screen without any thought to the other issue.

Quality Control

If a user adds a wrongly labeled event to their calendar, they could end up showing up at the wrong place at the wrong time, when nothing was going on. In order to prevent this, we allow users to edit posters that have incorrectly labeled information by tapping on the image of the poster rather than the 'Add +' button.

Labeling Interface

Figure 2. An illustration of our labeling interface and how a user can pinch to zoom in while a poster is being labeled.

We propose a new interface that makes labeling a lot easier once users get to this phase. Figure 2 displays this UI, showing our 3 fields that we request from users: event title, event start date and time, and the event end date and time. When those three fields are filled, the user clicks save and the information is uploaded to our database.

One of the primary difficulties we saw in the existing ways that users tried to solve this problem is that there is no way to look at the photo for the poster as well as enter information in the fields that calendar apps require to create a new event. If a user has a photo of a poster, they are forced to app switch constantly back and forth between the calendar application and the photo viewing application.

Application switching can cause increase cognitive load, as shown by Leiva et al. [1]. Otherwise users only option is to try to quickly memorize the even title, location and start and end times so that they can enter in their calendar app, but this also increases cognitive load. We wanted to provide a solution where a user could quickly take a snapshot of poster that catches their eye and still have a calendar event created even if the user was in a hurry or otherwise engaged in a different activity.

With our interface, instead of constantly switching between applications, users view the photo of the poster they are trying to label right beneath the text and date fields they are trying to fill out. Not only is the information very proximate, but the photo can be easily pinched in to zoom and panned around with one finger. We chose to have these actions not dismiss the keyboard so as not to break the flow from one field in the form to the next while they were labelling.

Constraining User Input

We chose to give the users a blank text field for the even title, but constrain their input of the start and end dates/ times using the iOS date picker. This gives us data that can easily be transferred to the Calendar Store should a user decide to add a photo to their calendar.

Camera Upload Screen

Figure 3. Camera Upload Screen

The Upload screen is a simple screen which allows users to upload photos of posters they want to have labelled. Here you can take a photo and review it in the grey square depicted in Figure 3. After that, click Upload and the photo will appear on the 'Your Photos' section of the feed on your device, and the public feed section of all other users.

Server Implementation

We chose Firebase for our back end because of its realtime database operations. This allowed all are users to receive real time updates for poster labeling, meaning they would get information pushed to their phone without having to pull to refresh the feed.

Additionally, we use Firebase's anonymous per device authentication. This means that you can simply download the app and get started without making an account. However, we still retain unique fingerprints for each device so we can load their specific user photos as well as ban problematic users.

We use Kingfisher for downloading and caching poster images and thumbnails on the device.

EVALUATION

We distributed our iOS prototype using Apple's TestFlight program to students from the CMU campus. We viewed college students as one of the main audiences of this projects because they are constantly being advertised to by posters from various groups on campus. Additionally, large poster walls concentrate the availability of event information spatially.

Figure 4. Amount of users for the end of December when we deployed.

User Count

We were able to recruit about 10 users from our school, but many people didn't use the app consistently. The extra 5 devices on the chart are development devices and simulators. However, we did have a small contingent of a few users who did use the product reasonably frequently and we maintained constant contact with them to get feedback on the app.

User Engagement

Our biggest concern is that nobody would spend time on the feed browsing the posters that other users had uploaded. If everyone only uses the app for the posters that they are interested in, our system would not work due to the fact that

people aren't even looking at other posters so labeling is not possible. However, we got reassuring results that people were at least browsing the feed. We saw that people spent an average of around 7 minutes and 8 seconds on the feed screen. Keep in mind that we probably haven't ran the study long enough to see how this metric changes over time, and perhaps much of this time was spent figuring out how the application worked because it was totally new to our testers.

We had 13 posted labeled during the time that the beta application was active.

DISCUSSION

Motivating the crowd to label posters voluntarily was a very interesting task. We didn't have anything like altruism on our side, and we knew that it was critical that the 'snap a photo of a poster and have it in your calendar' interaction did occur on our platform. We didn't want to bog down that interaction in points or rewards or modals. Instead we centered on the idea that when someone uploads a photo to our platform, it isn't just beneficial to them to have it there, but it is actually potentially beneficial to everyone on the platform. Much of our design decisions surrounding the feed really came out of this understanding, that we did actually have something that users were interested in and we could use it to get work done on our platform.

There are some clear limitations with PhotoCal, the first and foremost is quality control. Right now there is no flagging system, and we believe that this would be key for making sure that there is no offensive or harmful content perpetuated on the application. Another slightly smaller limitation is that we cannot sort out duplicate entries.

We also feel that the UI might change and improve as we continue to work on the application and system. We are unsure if relying solely on intrinsically motivated crowd workers is appropriate, and are exploring hybrid options between intrinsic and paid worker workflows.

Now at least as of right now, computer vision cannot solve this problem on its own, but I do believe there will come a day when that is no longer the case. We theorize that this points to a larger trend in which event advertisement, attendance, and management become increasingly hosted on digital platforms and the relevance of print posters lessens.

Even right now, its hard to justify the fact that people should come to our feed to see events when they can just go on Facebook and see an incredible range of events from all across campus or their city. However, we do think that the core interaction of taking a picture of something and then asynchronously turning that image into a structured piece of data is actually quite sound, and maybe in the future this application won't feature a feed quite as prominently but instead feature a camera. If computer vision can handle many cases, maybe we can simply hand off the few remaining cases to paid crowd-workers for a nominal fee.

FUTURE WORK

We hope to continue exploring the notion of providing quick on demand interactions like poster labeling by offloading the work to the crowd. Our next steps will be to integrate a failsafe mechanical turk mechanism that will make up for any insufficiencies or imbalances in our motivation structure. Integrating mechanical turk would allow us to confidently promise users that the core reason they're using the system is for the novel interaction of snapping a photo and having it labeled.

REFERENCES

1. Luis Leiva, Matthias Böhmer, Sven Gehring, and Antonio Krüger. 2012. Back to the app: the costs of mobile application interruptions. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services (MobileHCI '12). ACM, New York, NY, USA, 291-294. DOI: https://doi.org/10.1145/2371574.2371617
2. Zhang, Y. Event Info Extraction from Flyers. Retrieved December 17, 2017 from https://pdfs.semanticscholar.org/8f42/945bf093d0d61076f3fc17022d3301cb61cb.pdf
3. Josh Constine. Jul 29, 2015. With 450M Users, Facebook Events Is Primed For A Standalone App. Retrieved December 17, 2017 from https://techcrunch.com/2015/07/29/will-facebook-launch-an-events-app/
4. Glassdoor Inc. Retrieved December 17, 2017 glassdoor.com

VIDEO DEMO

https://www.youtube.com/watch?v=2vRMfqZnybg