Human Computation and Crowdsourcing: Works in Progress and Demonstration Abstracts
AAAI Technical Report CR-13-01

In-HIT Example-Guided Annotation Aid for Crowdsourcing UI Components

Yi-Ching Huang, Chun-I Wang, Shih-Yuan Yu, Jane Yung-jen Hsu
Graduate Institute of Networking and Multimedia
Department of Computer Science and Information Engineering
National Taiwan University
{d00944010, r01922132, r01922040, yjhsu}@csie.ntu.edu.tw

Abstract

This paper presents an approach to crowdsourcing annotations of UI components from images. Using the "Find-Draw-Verify" task design, an in-HIT example-guided annotation aid is proposed to facilitate workers thereby improving the result quality.

Introduction

User interface design involves selecting proper UI components and their layout on each screen. To facilitate data-driven design, Webzeitgeist offers large-scale design mining from a repository of over 100,000 Web pages and 100 million design elements (Kumar et al. 2013). Building the repository for mobile UI design is challenging as mobile design samples[1] are mostly in mockup images without source codes.

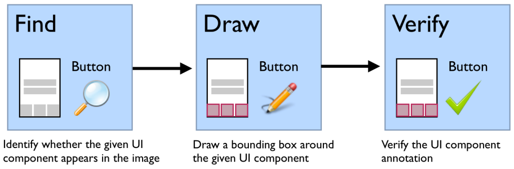
This paper proposes a novel approach to crowdsourcing annotations of UI components from such mockup images. Annotating a UI component consists of 1) identifying the presence of UI component in the image, and 2) drawing a bounding box around the UI component. In what follows, we start by explaining the "Find-Draw-Verify" workflow pattern and the problems encountered in our pilot test. We then introduce "in-HIT example-guided annotation aid" to facilitate the drawing task for workers without the domain knowledge or experience in UI design.

Workflow Design

The crowdsourcing workflow pattern "Find-Fix-Verify" was first introduced in (Bernstein et al. 2010) to break down a task into appropriate sub-tasks. In (Little et al. 2010), a toolkit named Turkit was proposed to perform iterative human computation algorithm on Mechanical Turk. To annotate UI components on mockup images, the workers should perform the following sub-tasks.

- *Find*: A worker identifies the label of a UI component that appears in the mockup image.
- *Draw*: A worker draws a bounding box of the UI component corresponding to the label.
- *Verify*: A worker verifies the annotated UI component or merges similarly annotated components.

Figure 1 shows the overall workflow. Simple subtask helped improve the quality of drawing the bounding box iteratively in (Su, Deng, and Fei-Fei 2012). Similarly, multiple HITs may be created for each stage of the workflow so workers can improve the quality iteratively by modifying the result from previous workers.



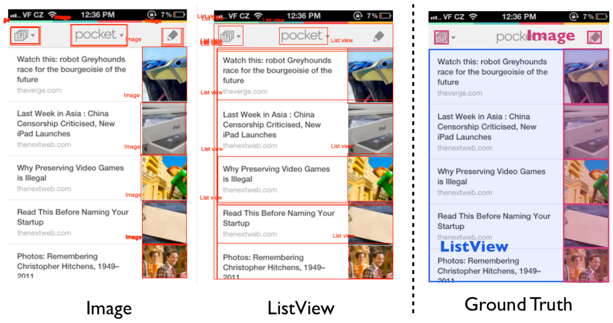
Identify whether the given UI component appears in the image

Draw a bounding box around the given UI component

Verify the UI component annotation

Figure 1: The Find-Draw-Verify workflow for annotating UI components in mockup images.

Figure 2 shows the result of iterative subskeleton task in the drawing stage. Three different UI components are annotated on a given mockup image. The quality is low, as the bounding boxes are not tightly around the UI components. Five list items are identified as the ListView component, while the ground truth is one bounding box containing five list items.



Watch this: robot Greyhounds race for the bourgeoisie of the future
theverge.com

Last Week in Asia : China Censorship Criticised, New iPad Launches
thenextweb.com

Why Preserving Video Games is Illegal
thenextweb.com

Read This Before Naming Your Startup
thenextweb.com

Photos: Remembering Christopher Hitchens, 1949–2011

Figure 2: The results of our method. The rightmost is ground truth of UI annotation.

In our pilot test, identifying UI components such as Label, TextField, GridView, and ListView turns out to be too

[1] Mobile design gallery http://pttrns.com.

difficult for workers domain knowledge on interface design. Besides breaking down the workflow into simple subtasks, a better annotation interface is in order.

In-Hit Annotation Aid

We introduce how in-HIT annotation aid can help workers improve task quality.

Qualification Test

UI components are classified into 4 categories: view, input, display and bar. The worker only have to pass the qualification of a single UI category to perform the annotation task. The drawing task is provided with an external tutorial and guiding examples.

However, we found only few people pass the qualification, which limits the throughput of the task. Also the annotation made by qualified workers were not precise. The results show that even the qualified workers was not clear about the definition of each UI component type. Also, the provided external tutorial link in the hit was ignored. The results indicate the workers needs aid on the spot.

In-HIT Example-Guided Aid

To provide just-in-time learning, we show some mockup images with ground truth to help workers correctly identify the target UI component. Figure 3 shows our interface with in-HIT example guided aid. The guiding examples were placed right next to annotating area, providing direct visual cue. Workers can learn by examples on the spot and perform the task at the same time. The results are shown in Figure 4. With in-HIT example guided aid, workers achieve more output agreements. We can see that workers annotate the correct ListView component, compared with previous result in Figure 2.

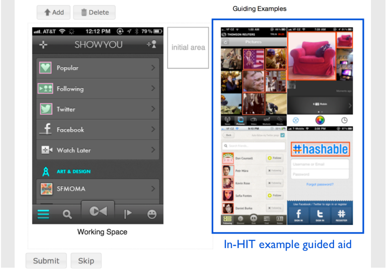


Figure 3: In-HIT example-guided aid in interface

BoxSnap: In-HIT Drawing Aid

"Drawing bounding box is difficult and time-consuming(Su, Deng, and Fei-Fei 2012)", while workers hope to complete HIT as soon as possible to earn more money. This way leads to bad annotation. To lower the difficulty, we design an embedded intelligent assistant *BoxSnap* in interface. Instead of drawing the regions manually, *BoxSnap* can auto-detect those candidate regions while user pointing the UI components with mouse. Workers can select the UI component in just one click and modify the boundary slightly if necessary.

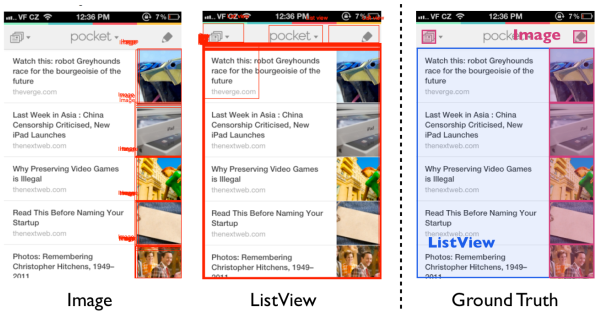


Figure 4: The results of using In-HIT example-guided aid.

We utilize digital image processing techniques to preprocess those mockup images. Our image preprocess follows standard procedure: edge detection, dilation, erosion and contour finding for connected component.

For each mockup, the regions of UI components are segmented and stored in the database. When user hover the mouse over the mockup, we calculate the distance of the mouse position to the centroid of each the bounding area, and choose the one with shortest distance as candidate. A threshold distance was set to determine whether user has intention to annotate the area.

Discussion and Conclusion

Our study showed that annotation interface significantly impacts the quality of crowdsourced data. Selecting workers using qualification tests limits the size of candidate pool for hard tasks. This research proposed in-HIT crowdsourcing aids to help workers produce better results.

BoxSnap facilitates drawing bounding boxes via image preprocessing. To detect all UI components automatically, the image processing algorithms should be improved in terms of accuracy. The proposed method will be evaluated by further experiments in mobile interface design.

References

Bernstein, M. S.; Little, G.; Miller, R. C.; Hartmann, B.; Ackerman, M. S.; Karger, D. R.; Crowell, D.; and Panovich, K. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, 313–322. New York, NY, USA: ACM.

Kumar, R.; Satyanarayan, A.; Torres, C.; Lim, M.; Ahmad, S.; Klemmer, S. R.; and Talton, J. O. 2013. Webzeitgeist: Design mining the web. In *Proceedings of the 2013 annual conference on Human factors in computing systems*.

Little, G.; Chilton, L. B.; Goldman, M.; and Miller, R. C. 2010. Turkit: human computation algorithms on mechanical turk. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, 57–66. New York, NY, USA: ACM.

Su, H.; Deng, J.; and Fei-Fei, L. 2012. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.