GR6 User Testing and Reflection

Schnap It!

Carl Vondrick, David Hayden, Joseph Lim, David Way

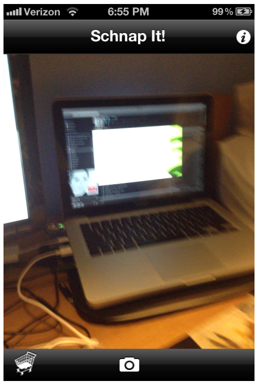
A. Introduction

In this paper, we present Schnap It!, our iPhone shopping assistant that enables consumers to quickly look up reviews and prices of products that they encounter while in a physical retail store (e.g. Best Buy). Users simply take a photo of one or more products of interest and draw coarse bounding boxes around each. Using computer vision, Schnap It! proceeds to automatically retrieves ratings, reviews, and pricing from trusted websites, and aggregates them into a familiar shopping cart-like interface where users can compare and contrast products of interest. Schnap It! distinguishes itself from other mobile visual search applications in that it is optimized to help users compare products that they encounter while in a physical retail space. In the remainder of this document, we present our implementation and show preliminary results from user studies. We also discuss our design decisions, and how they affect the learnability, efficiency, and safety our application.

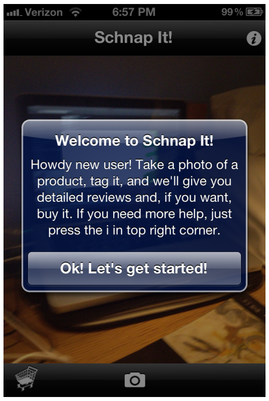
B. Design

In this section, we describe the final design of our interface. We present the screens in an order that a typical user would experience them.

(1) Launch Screen:



Schnap It! opens directly to a live camera view where users can either 1) take a picture of a new product, or skip directly to the Reviews Shopping Cart, which stores products that they've previously looked up.



Welcome to Schnap It!

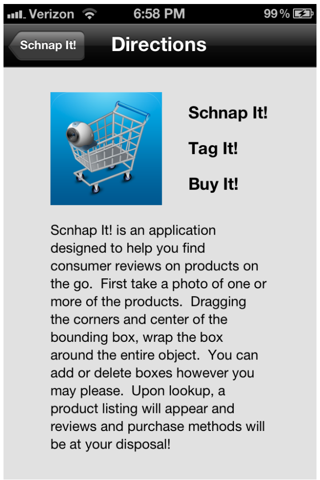
Howdy new user! Take a photo of a product, tag it, and we'll give you detailed reviews and, if you want, buy it. If you need more help, just press the i in top right corner.

When opening Schnap It! for the first time, users will see an informational popup that briefly describes how to use the application and directs them to the help pages (upper-right corner) in case they get stuck. The popup was included because we found that first-time users occasionally faltered between taking a photo and viewing product reviews. We further observed that once users had looked one product up, they had no trouble doing so again. Thus, subsequent application starts do not have this popup.

Throughout user testing, we realized that there were several scenarios where a user would want to open Schnap It! to compare products that they have already imaged. Earlier Schnap It! designs forced them to take a “dummy picture” to get to those products, but they are now able to simply tap the shopping cart icon (bottom right) to go directly to the Reviews Shopping Cart.

Normally, users will center one or more products, and tap the camera icon. Doing so takes them to (3). If they are confused, they may instead tap the *information icon*, which takes them to (2).

(2) Help Screen:



Schnap It!

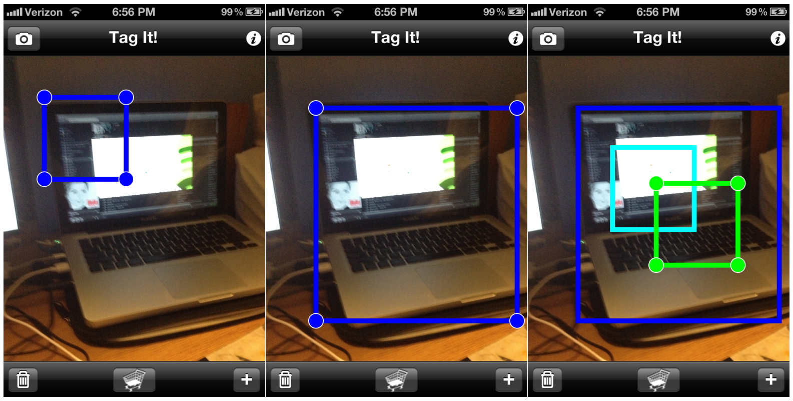
Tag It!

Buy It!

Scnhap It! is an application designed to help you find consumer reviews on products on the go. First take a photo of one or more of the products. Dragging the corners and center of the bounding box, wrap the box around the entire object. You can add or delete boxes however you may please. Upon lookup, a product listing will appear and reviews and purchase methods will be at your disposal!

Users are presented with a single-page help screen if they tap the info icon in the top-right corner in (1). Like the first-time popup, this briefly describes how to use Schnap It! for best results.

(3) Annotation Screen:

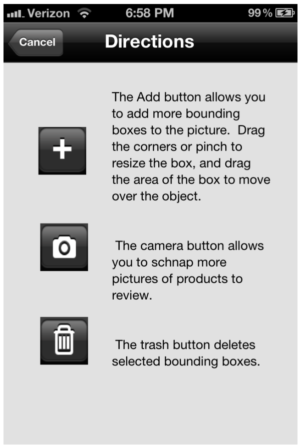


After taking an image containing one or more products of interest, users are presented with the Tag It! view, where they must drag and resize bounding boxes around each product of interest. This view initializes with a single bounding box in the upper-left corner (left image). Additional bounding boxes may be added by tapping the plus icon (right image, lower-right corner), and removed by tapping the trash can icon (lower-left corner). Once users have tagged each object of interest, they may proceed to look up the products by tapping the shopping cart (bottom middle), in which case they proceed to (5). If they are confused, they can tap the information icon for help (4).

It was necessary to ask users to annotate products of interest because current computer vision algorithms struggle with segmenting important content (products) from background clutter. We considered several different implementations. Arguably the most efficient interface would be one in which users draw freeform boundaries around each object with a single stroke. But in testing we found that first-time users were confused unless given a demonstration. While such a demonstration could be provided in an informational popup, we felt that doing so made for a less appealing interface than alternatives.

Among other alternatives, we considered n-sided polygons, and two-finger rectangle pinching where the user controls the diagonal (size) with one hand, and the shape with the other. We ultimately settled on a relatively simpler, movable and draggable bounding box design. This was found to cause the least confusion among users, largely due to its consistency with existing tagging interfaces (e.g. Facebook photo tagging).

(4) Help Screen:



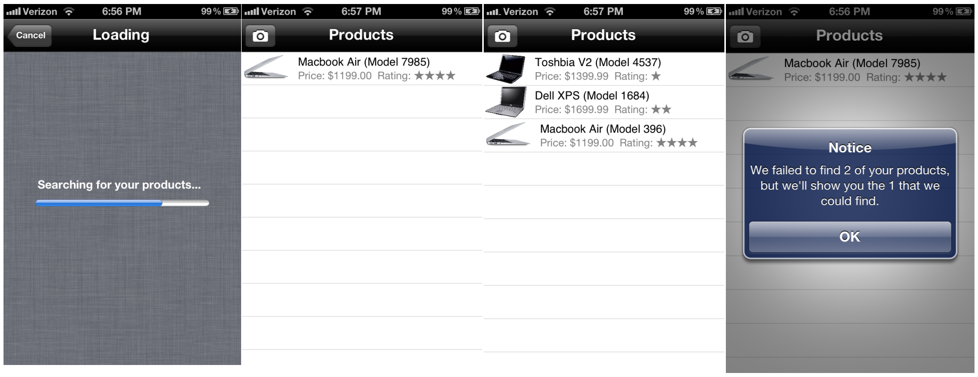
The Add button allows you to add more bounding boxes to the picture. Drag the corners or pinch to resize the box, and drag the area of the box to move over the object.

The camera button allows you to schnap more pictures of products to review.

The trash button deletes selected bounding boxes.

This is the directions screen that pops up when the user hits the info button on the tag screen (see (3)). It gives a concise description of what each button does in case the affordance is not clear. We added this after heuristic evaluations showed that some users did not know that the plus icon added a new bounding box.

(5) Shopping Cart:



Searching for your products...

Macbook Air (Model 7985)
Price: $1199.00 Rating: ★★★★

Toshbia V2 (Model 4537)
Price: $1399.99 Rating: ★

Dell XPS (Model 1684)
Price: $1699.99 Rating: ★★

Macbook Air (Model 396)
Price: $1199.00 Rating: ★★★★

Macbook Air (Model 7985)
Price: $1199.00 Rating: ★★★★

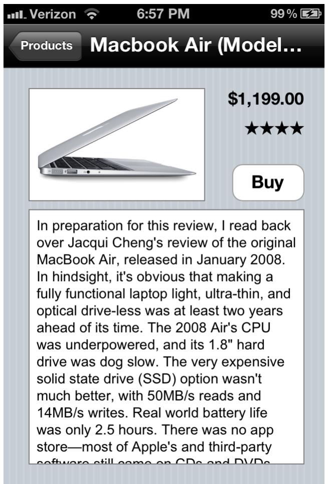
Notice

We failed to find 2 of your products, but we'll show you the 1 that we could find.

Once users have labeled one or more objects in a single image and have tapped the shopping cart icon, they are presented with a loading screen (left) while vision algorithms operate. We found that the loading screen reassured users that the application was, in fact, working and responsive. After loading, users are presented with the Reviews Shopping Cart, which contains stock images for each recognized product, their model, price, and rating (in stars). New products are added to the top of the cart as users take and tag additional images. Users may tap any row to view details about an individual product (6), or they may return to the live camera view (1) to take additional product photos by tapping the camera icon (top-left corner).

Visual product lookups won't always be successful: users may have taken a blurry photo, the product may not be in the database, or the user may not have a network connection. In any of these cases, a notice pops up (right). If there are other products in the shopping cart, it leaves them there. Else, it takes them back to the live camera view (1).

(6) Product Details:



In preparation for this review, I read back over Jacqui Cheng's review of the original MacBook Air, released in January 2008. In hindsight, it's obvious that making a fully functional laptop light, ultra-thin, and optical drive-less was at least two years ahead of its time. The 2008 Air's CPU was underpowered, and its 1.8" hard drive was dog slow. The very expensive solid state drive (SSD) option wasn't much better, with 50MB/s reads and 14MB/s writes. Real world battery life was only 2.5 hours. There was no app store—most of Apple's and third-party software still came on CDs and DVDs

After tapping a product row in (5), the details screen pops up and displays information about the product. We canned the vision backend to only return random laptop products. Reviews are fetched from trusted websites (such as www.cnet.com). One difficulty with doing this is that automatically scoping the length of the text is difficult: too little and the useful parts of a review may be cut off. Too much and the review may be more than the user wishes to see.

Many products are cheaper to buy online than at retail shops. Thus, Schnap It! allows users to buy it directly from a trusted source (e.g. Amazon.com, Apple.com), rather than doing so there in the store. To do so, they tap the buy button (top right), which takes them to the merchant’s web store in the Safari browser.

C. Implementation

In this section, we briefly describe salient implementation details that either contributed to the user experience or impacted our evaluation methods.

(1) Overview: Our application is implemented on an Apple iPhone and written for the iOS platform in Objective C. In most cases, we attempted to use the iOS framework so as to maintain consistency with the rest of the iPhone user experience. Our software is a native iPhone application.

(2) Model View Controller: In order to obtain separation of concerns, we follow the Model-View-Controller pattern. We created product models that managed our database of products. We also created views that only display the relevant information on the screen. The controller manages the application and propagates changes between views and models.

(3) Camera: When implementing the camera functionality, we bypassed the opening shutter animation found in the standard iPhone camera application. This change minimizes the context switch from taking a photo to deciding which object to annotate. We believe this change significantly improved the usability of our application. However, in order to accomplish this, we had to tweak the stock Apple iPhone framework since the shutter animation was difficult to turn off.

(4) Annotation: We spent significant effort designing the annotation screen in order to make it as intuitive and efficient as possible. Since visual image annotation is not provided by existing libraries on the iPhone, we implemented our own. Notably, Schnap It! bounding boxes can be selected and dragged in a single motion.

(5) Reviews Shopping Cart: The shopping cart is implemented using the same data structures that other iPhone applications use. This choice allows the UI to remain consistent (such as allowing the user to delete by swiping a row).

(6) Product Details: When implementing the product details screen, we decided to have the "Buy" button launch the Safari web-browser where the user could proceed to buy the product. We made this choice because we wanted the user to enter a familiar application (i.e., Safari) that they understood well. In other words, we wanted to maintain high learnability during the purchase process. In our experiments, we found mixed opinions about this application switch. While some users were comfortable in Safari, others the application switch to be jarring.

(7) Animations Between Screens: We also decided to use a UINavigationController in our implementation. Rather than stringing together several different UIViewControllers, we have a central controller that maintains a stack of UIViewControllers that allows for easy and consistent forward and back actions. These back and forward actions are responsible for the smooth animations between each screen. We argue that these animations increase the usability of the application since it is clear whether a screen is moving forward or moving backwards now.

(8) Mock Recognition: In our experiments, we did not deploy a computer vision object recognition algorithm due to time constraints. Instead, our object detector was simply a random number generator and we had a predefined database of objects that could be recognized. We maintained realism during our experiments by only using physical products that were in our database.

D. Evaluation

In this section, we evaluate our prototype on real users and discuss their feedback. Although most users found our application useful, they also provided many criticisms of the interface that we would address in later iterations of the development cycle.

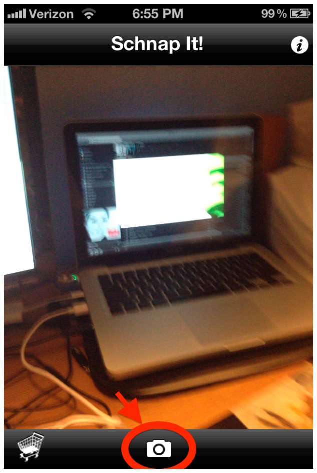
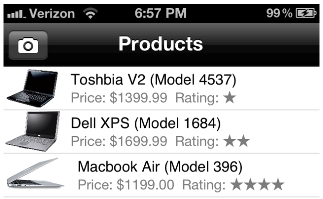
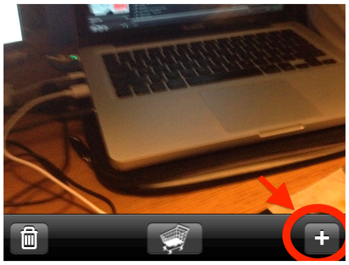
(1) Participants: We randomly selected users from inside Stata Center and asked if they wished to participate in an user interface testing. We report results on 3 users (number of females = 0, median age = 25). Although our sampling was random, we note that participants had strong backgrounds in computer science. One justification for this sampling bias is that we expect such users would be representative of the early adopters who would use Schnap It!.

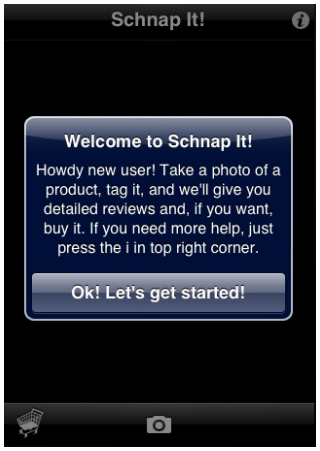
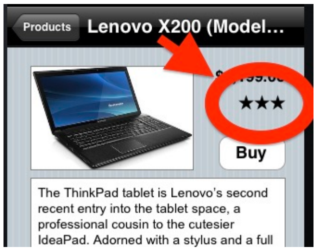
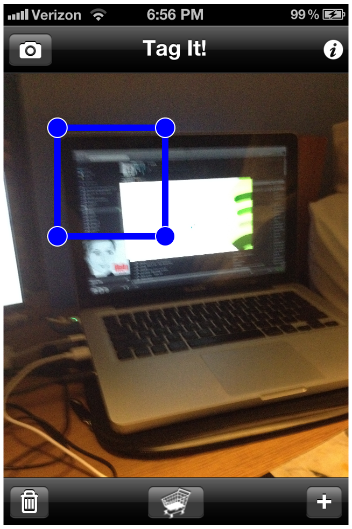
(2) Prompt: After a subject agreed to participate, we read the following prompt to the user:

> *Thank you for participating in our user study! Suppose you are inside Best Buy and you are about to buy a new laptop computer. You have narrowed it down to three options shown in front of you. Using your iPhone and the application SchnapIt!, complete the three tasks: 1) find reviews for each laptop, 2) determine which laptop you wish to buy, and 3) buy the laptop of your choice.*

The subject was seated in a chair in front of a table with three laptops on it. After the subject indicated that they understood the instructions, we silently observed them. At the conclusion of the experiment, we solicited qualitative feedback.

(3) Summary of Results: We found that users generally held favorable opinions of our application and found it useful. It was unclear whether a subject would use the product in practice, but they solved each task with relative ease. Below, we categorize the feedback and insights we gathered in order of severity:

Shopping Cart Confusion (Critical): Users in our study were most critical of the shopping cart. All users thought the shopping cart icon at the bottom of the camera screen would actually buy the product even though it would only bring them to the shopping cart page. This apprehension caused users to push the plus icon (to add a second annotation box) even when they were only looking up a single product! **Solution** In future versions, we plan to change this icon to a magnifying glass in order to better afford that this icon simply initiates product lookup.	
Confusion over New Items in Shopping Cart (Critical): After taking multiple photos, users reported confusion over which products were from the most recent photo and which were from older photos. They correctly assumed they were added to the top of the cart, but still were confused because there are no visual affordances to indicate this, and the stock images do not always look like the image they took. **Solution:** We plan to color-highlight the most recent products in order to indicate that they are the latest products analyzed. We believe this will also mitigate concerns about why the shopping cart kept growing which each photo, a confusion that one user experienced.	 Toshbia V2 (Model 4537) Price: $1399.99 Rating: ★ Dell XPS (Model 1684) Price: $1699.99 Rating: ★★ Macbook Air (Model 396) Price: $1199.00 Rating: ★★★★
Annotation Plus Button (Major): Multiple users were confused by the plus button on the annotation screen. They did not realize that tapping it added a new bounding box for annotation. After pressing it, they quickly learned the interface and recovered by tapping the trash can icon. **Solution** We are pleased to note that this means our interface is both safe and learnable. Nonetheless, we plan to change this icon graphically to better indicate that it is adding a new box and not taking a new photo.	

Informational Popups Annoyance (Major): Although our first round of user testing in GR4 suggested that we should add pop-up boxes to help guide the user through the experience, all our users this round reported being annoyed by the popups. Consequently, users did not read them. As a compromise, we plan to show the help messages unobtrusively as a message on the top of the screen that gracefully fades away.	 **Welcome to Schnap It!** Howdy new user! Take a photo of a product, tag it, and we'll give you detailed reviews and, if you want, buy it. If you need more help, just press the i in top right corner.
Number of Reviews Problem (Minor): One user refused to make a decision among the three products because there was no indication of the number of reviews that contributed to the star rating. **Solution** In future versions, we will display the number of reviews that contributed to the star rating (similar to Amazon.com).	 The ThinkPad tablet is Lenovo's second recent entry into the tablet space, a professional cousin to the cutesier IdeaPad. Adorned with a stylus and a full
Box Centering Efficiency (Minor): All users reported that drawing boxes on our interface was intuitive: all found dragging and resizing to be intuitive. However, new annotation boxes created with the plus icon are not centered. Users complained that this reduced efficiency since products are often centered or nearly so by the user when they take the photo. **Solution** Nonetheless, we believe this remains an active research area on how to best annotate images; hence, we plan to focus future effort in this regime by exploring more advanced annotation techniques (such as computer assisted image and video labeling).	

E. Reflection

Over the course of the semester, this project helped us realize that testing on real users *before and after* an implementation provides invaluable and often unexpected feedback. Initially, we did not expect that silently observing users press buttons on paper would allow us to glean insights into how to improve our interface. To our surprise, we found user feedback to be quite informative. We now firmly believe that users should be involved in UI design as early as possible.

Throughout the iterative process, we felt constrained by the level of usability analysis towards the entire application. There were many decisions and choices that were not independent, so it was difficult to decide on one computer prototype to invest. If we had more time, we would have made several different prototypes, tested many different types of annotation, and compare review methods on the iPhone. The level of analysis towards these specific areas would have improved, rather than being directed toward the details or direction screens.

While our original intention for this project was to study how users can best annotate images and videos on mobile platforms, project requirements encouraged us to approach this interesting HCI problem in a more holistic fashion. This exercise was helpful for us as computer vision researchers because it made us realize that mobile annotation tools will not live or die solely based on the quality of the object recognition algorithms powering them, but, rather, require significant thought about how best to integrate them in a usable interface. Moreover, we realized that the applications for vision are perhaps still very ambitious and not very well defined.

Given the observations noted by the user tests and heuristic evaluations, we had to decide which features to change and how to change them. Often, users had different preferences regarding efficiency and visibility. For instance, users wanted for every image to be saved and have the ability to be accessed in the application. Most users did not immediately understand what the plus annotation icon did, and some were not satisfied with the informational popups that were intended to guide them through their first product lookup. In order to maintain the focused tasks of the application and our own goals in accuracy and efficiency, we occasionally had to decline requests evaluators made.

Overall, this project proved insightful for us and our research. As computer vision researchers, we often envision deploying our work on software used by used by real (non-academic) users. This class and project gave us useful insights and experiences towards designing systems that not only push the boundaries of computer vision, but will also be usable and acceptable to end users.