Ubi-cook: Automated Recipe Finder

Caitlin Cagampan, HCI
cmcagamp@ucsd.edu

Justin Li, HCI
jkli@ucsd.edu

Hannah Chen, CSE
hechen@eng.ucsd.edu

Steven Rick, HCI
srick@ucsd.edu

Narine Cholakyan, CSE
ncholaky@ucsd.edu

Abstract

Ubi-cook is a Kinect application for the chef inside all of us, enabling people everywhere to make sense of the supplies they have on hand. Ubi-cook solves the manual input process of locating recipes by visually identifying food items and automatically querying recipe sites to save time and money.

Author Keywords

ubiquitous computing, Microsoft Kinect, computer vision, smart kitchen, ambient interaction

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

Introduction

Cooking is not only the process of making food, but also includes planning, buying and preparing the necessary ingredients. All cooking requires an investment of time which can often overwhelm or deter a beginner or novice cook. This may explain why Americans eat out 5 or more times a week [10], a convenient yet expensive alternative. There are many advantages to preparing one's own meal, such as potential to reduce costs and increased awareness of caloric and dietary consumption. Most importantly, an individual can identify what food he or she is putting into

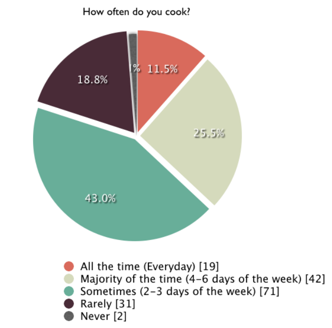
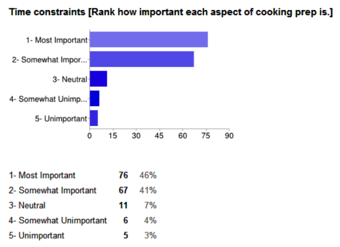
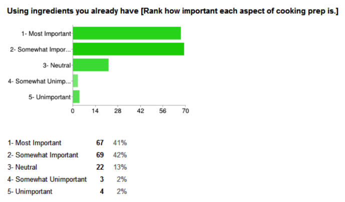


Figure 1: How often do you cook?



1- Most Important	76	46%
2- Somewhat Important	67	41%
3- Neutral	11	7%
4- Somewhat Unimportant	6	4%
5- Unimportant	5	3%

Figure 2: Importance of time constraint factor



1- Most Important	67	41%
2- Somewhat Important	69	42%
3- Neutral	22	13%
4- Somewhat Unimportant	3	2%
5- Unimportant	4	2%

Figure 3: Importance of using ingredients you already have

his or her body. The common saying: "You are what you eat" continues to hold true, so in order to shift the statistics, there must be a simpler and easier way to cook a delicious meal.

Recipes are often used in order to create new or complex dishes. Primarily found online or in cookbooks, they instruct and simplify the cooking process for people who may have not cooked the dish before. Although they are incredibly handy, it is difficult to find the right recipe due to the nature of having to shop for certain ingredients to make each dish. This leaves a very impersonal experience for the user and provides a lot of room for improvement.

Ubi-cook is a Kinect application that provides a time-efficient, instructive guide utilizing resources and ingredients found in the kitchen to assist with the cooking process. It expedites and personalizes the process by suggesting recipes based on ingredients the cook already has. Traditionally, preparing a meal would require foresight and planning. By scanning the ingredients in front of the camera, the application would provide recipes with cook time and instructions. Regardless of your skill level, with Ubi-cook, 'you can be the cook'.

Finding a recipe is as simple as a quick search online, but identifying one that takes into account ingredients you already have is not so easy. Reverse recipe look-up services do exist online; however these catered results require users to manually input each and every ingredient constraint. With the decreasing cost and increasing functionality of scanners and other devices, the concept of a "Smart Kitchen" is finally coming to fruition. Ubi-cook is a starting step to bring an efficient and ubiquitous user experience to cooking by scanning available ingredients and providing recipes based on this constraint, and ultimately producing a more cost effective and time efficient cooking process.

Motivation

When it comes to preparing meals, people are faced with the daily dilemma of what to eat with the random array of ingredients found in their kitchens. On top of that, deciding and preparing what to eat often takes place within a hectic and time-constrained schedule.

Survey Results

To explore this issue further, we created and distributed a short online survey to acquire more generalized, user-focused data surrounding the meal preparation process. The survey was distributed through the personal Facebook profiles of 5 college students and answered by a total of 165 participants. Overall, the survey results revealed that the current process of cooking with already available ingredients required external resources, such as recipe websites.

Overall, 80% of participants reported cooking regularly (See Figure 1), with the largest percentage of participants (43%) cooking "Sometimes (2-3 days of the week)". 56% of participants confirmed that they made use of recipes "Most of the time" or "Some of the time" while cooking. This helped us identify the area in need of improvement: the recipe search process.

Users further described that they used recipes because they did not want to waste time or ingredients, with 87% and 83% of participants classifying "Time Constraints" and "Using ingredients you already have", respectively, as important components of their cooking experience (See Figure 2 and 3).

Participants also self-reported that recipes could be improved by providing potential ingredient substitutes that make use of available ingredients. It was reported that they primarily use the internet to acquire recipes, with 40% of participants citing the use of Google search to constrain recipe results by ingredients. Ubi-cook strives to build upon this idea by seamlessly integrating ingredient input with recipe querying by placing a smart device in the kitchen, bringing us one step closer to the realization of a "Smart Kitchen". Altogether, the user input from the survey resulted in the final design insights that motivated our development of Ubi-cook.

1. Users need a recipe in order to give them some structure and direction.
2. Users need recipes based on available ingredients.

With these user needs in mind, Ubi-cook offers an intuitive, hassle-free way of preparing meals and a more practical, less wasteful food preparation process.

Related Work

There are a number of existing applications, tools, and research attempting to provide users knowledge about the ingredients they have on hand in their kitchen. These various related works helped define the scope of Ubi-cook as well as motivated the direction that our development took.

Visual Inventory

Insider [7], a product prototype, sets out to capture a visual inventory of a user's groceries. This device acts as a window into your fridge, providing access to a camera wirelessly. Insider emphasizes the importance of visual information for creating an item inventory, but stops there. The system does not have any intelligence to make sense of the items that it sees. By pushing all of the processing to the human, Insider severely limits its usefulness to whatever you can see on your smartphone through the device. This drove the need for object recognition in Ubi-cook.

Object Recognition

Looking towards Google, the search giant offers Search by Image [6] functionality to ease searching in the visual domain. Exploring this, we found out that Search by Image largely uses color histograms. This enables the search engine to easily match up visual content that already exists in its databases, but when given novel content it instead produces results that are 'visually similar'. A freshly taken photo can return results that share colors and hues with the query image, but has no promise of matching the content of the image. For Ubi-cook to work in a real environment, it needs to function effectively with classification of novel images.

Google has also developed Google Goggles [5], an application that allows you to take live visual content and identify it. Looking into this application, we found that Google was using a number of different engines for trying to make sense of the content it was receiving, such as OCR for text, readers for barcodes, and SURF/SIFT for image processing. However, because Google aims to have a very extensive library of things that it can identify with Goggles, the SURF/SIFT models have trouble telling more common and similar things apart without extremely extensive training data. For this reason, Goggles can identify landmarks, like the Eiffel Tower, but cannot make sense of fruit such as an orange. Ubi-cook could use such a system, but would require finding a method to optimize object recognition for use with common groceries and not just text and landmarks.

As we browsed around further into object recognition, we came across this video [13]. This individual was using Kinect and OpenCV for image processing to train his system on a set of objects, which enabled recognition after the fact. We found the application to be very impressive, but realized that he was testing his system with unique items. This showed us that OpenCV might allow us to train Ubi-cook to recognize groceries but left us unsure how it would handle similar looking (but unique) objects.

Food Recognition

We found two systems that attempted to make sense of groceries explicitly, one being the MIT Media Lab Food Cam [11] and the other being a Vegetable Recognition system by Studio Diip [12].

The MIT system utilized a webcam and a BOVW (bag of visual words) method for identification of the food objects that it saw. By using opponent-color SURF features and a SVM (support vector machine), it could differentiate between food categories, such as cookies, Indian, Italian, fruits/vegetables, and sandwiches, with reasonable accuracy. Because the Food Cam attempted to make distinctions between completed food dishes rather than just ingredients, it suffered from low accuracy due to a wide range of trained items fitting into each category. The MIT system did make it clear to us, however, that by focusing on ingredients and categorizing accordingly, we could very well accomplish our goal with a SIFT/SURF + BOVW + SVM system.

The Vegetable Recognition system, in contrast, focuses on identifying ingredients and providing recipe-based feedback to the user. With Studio Diip disclosing little more than a brief overview of the product, we were left with nothing surrounding implementation method or product effectiveness. This inspired us to focus on creating an easy to use ingredient recognition system that not only added value to the user through recipe lookup, but might also allow customization based on dietary needs or preferences as well as mobile tracking of their inventory.

Ubi-cook: You can be the cook

Ubi-cook is an application that synthesizes the Kinect camera feed with a recipe look-up site [2] to provide recipes tailored to the user's available ingredients. This automated process tackles the tediousness of manual input and experiments with the concept of a smart environment. While our prototype is built with the Kinect in mind, its current capabilities (namely object recognition via images rather than scanned models) can easily be adapted for webcams and other video/camera feeds.

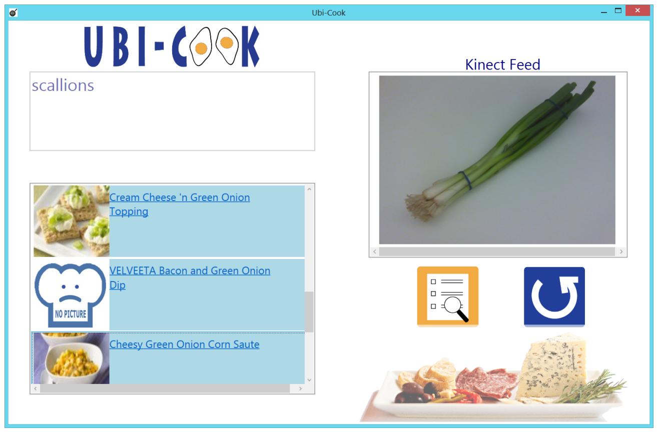

Ubi-Cook
UBI-COOK
scallions
Kinect Feed
Cream Cheese 'n Green Onion Topping
VELVEETA Bacon and Green Onion Dip
NO PICTURE
Cheesy Green Onion Corn Saute

Figure 4: Ubi-cook user interface

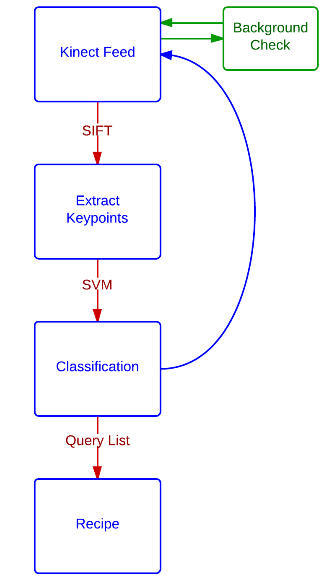


Figure 5: Ubi-cook Pipeline

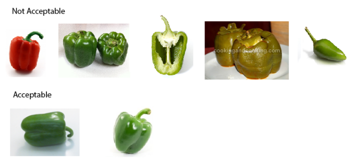


Figure 6: Examples of Good and Bad Images for our classifier

Implementation
Our pipeline involves three main processes: obtaining data via Kinect feed, object classification based on the Bag-of-Visual-Words [8] computer vision model, and finally querying and outputting the results of a recipe site. The automation is achieved by capturing a data frame every 7 seconds and classifying the given frame with an existing item library. To eliminate unwarranted computation, a low-pass comparison between the data frame and the background frame (i.e. default frame with no item in view) is done to determine if there is an item present for classification.

Prior to classification, an image library was created containing ten different ingredients (k=10). Each image is represented by feature descriptor vectors generated with the scale-invariant feature transform (SIFT) [9] algorithm. The descriptors are quantized into k clusters and a codebook, or image dictionary, is created. With the support vector machine learning model [3], each ingredient is then classified based on a histogram of codewords generated by its descriptors.

The Ubi-cook interface seamlessly translates the Kinect camera feed to an ingredient contained in our food library on a loop, and finally returns the name, picture and link to the top ten recipes based on results from Recipe Puppy.

Limitations
While SIFT has many benefits, such as identifying keypoints regardless of image orientation and scale, the original algorithm is designed for grayscale images. To improve ingredient classification, alternate algorithms, such as CSIFT [1], can be applied that take into account color gradients alongside intensity gradients. Another limitation is that Ubi-cook presently cannot differentiate between learned and unlearned items. Any item provided will ultimately be mapped to a specific codeword within our library. This obvious misclassification can be eliminated by setting a threshold for the codeword histogram values. Determining a significant threshold level for our demo would have been arbitrary due to our limited training data, and we simply tested items known to exist within our library. Finally, all training must be done prior to testing, so incorporating new ingredients into the library cannot be done in real time. While some computational time can be eliminated by saving the SIFT descriptors, the calculation for k-means clustering must take place each time training is done to generate the codebook.

The success of our application ultimately relies on our recipe database. Many alternative recipe sites were considered, but due to the closed nature of those sites and lack of APIs, we relied on Recipe Puppy because queries are made through the URL string. Queries can contain both ingredients (e.g. garlic, broccoli) and keywords (e.g. salsa, soup); the current version of Ubi-cook only takes advantage of ingredient querying. We found certain instances of recipe queries to cause crashes or incorrect recipe generation due to inconsistencies and unknown factors in Recipe Puppy's backend. For instance, a "pepper" query was automatically converted into "black pepper". We also found that certain recipe titles contained characters that could not initially be parsed by our web scraper. While these issues can be easily addressed on a case by case basis, that does not provide a viable solution for a library with thousands of items.

Setup Environment
The same setup environment featuring the Kinect situated with a top-down view over a white tabletop was used for both learning and testing. Our training data consisted of fifty images per ingredient, with half of these images

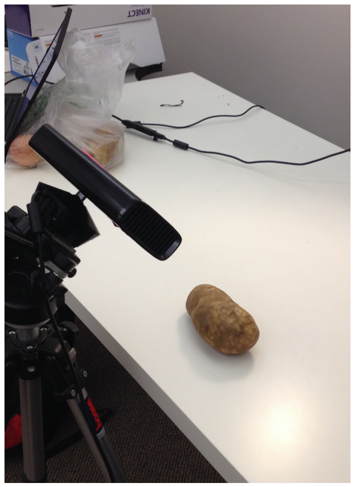


Figure 7: Environment Setup for Ubi-cook

coming from Google images and the other half through the Kinect feed. Images were restricted to: single items, item in full, items in same view (e.g. top versus side view of a pineapple), raw ingredients, same type of ingredient (e.g. red onion vs. yellow onion), untouched ingredient, and containing no watermarks. Testing the classification library with an image from Google and an image taken with the Kinect yielded correct classification in 9 out of 10 and 6 out of 10 ingredients, respectively. This was found from an extremely limited test space of typically 2-5 images per ingredient. The choice of training with fifty images per item was arbitrary and clearly insufficient to properly train a library with ten classes, but was chosen to maintain sanity in manual data collection.

Future Work

Comprehensive Training Data

The primary improvement for Ubi-cook would involve creating a more extensively trained database of items. With many nuances in position, angle, color, ripeness, size and type for each ingredient, and as more ingredients are added to the library, an exponential increase in images per ingredient is required to calculate clearer distinctions between codewords. Calculation of k-nearest neighbor means is also rather time consuming; the Caltech dataset [4], which contains 9146 images belonging to 101 categories, is reported to take hours to train. Along with more data, we can also use better data. Kinect Fusion uses a depth feed allowing us to scan items and create models. The same implementation for training and classification used with flat images can be applied to 3D point clouds, though SIFT extraction and clustering would require more computation. Whether using images or models, we suspect that future training data collection would involve some form of crowdsourcing, either by consumers, retailers or producers. For example, many people already photograph food to share and tag on Instagram, Tumblr and Facebook, though a filter would be required to ensure images of raw ingredients. More realistically, cameras are already utilized to automate food quality control and this data could be shared in a vast database.

Smarter Device

Many additional features could be added to Ubi-cook's back-end to transform it into a smarter device. Firstly, querying recipes would not solely involve what ingredients you have, but how much of a particular ingredient you have compared to what is required in recipes. Time can be saved because scanning does not have to be a single item task; algorithms exist that can detect an item within a group of items and thus scan multiple items in view.

Secondly, ingredients can include more properties, such as quality and freshness. By taking into account the age of an item to track its expected lifespan, the system could prioritize usage to optimize taste and food freshness while simultaneously reducing food waste. Gamification of such a system could even reward users for using their groceries before expiration, in turn creating habits that produce less waste.

Thirdly, queries can be customized by exploiting the keyword feature in Recipe Puppy to personalize results. Ubi-cook's interface could incorporate types of food rather than raw ingredients (e.g. gourds versus squash) or take input from the user for certain health and dietary restrictions. The system could be taught to recognize alternative ingredients for a recipe, such as automatically substituting tofu for cheese with vegan users, or make the best of what is available by substituting (e.g. using shallots in place of onions).

This would not only make cooking easier, but could introduce new cuisines and cultures to a user's cooking library.

Smarter Environment and Interaction
Similar to the Insider mentioned earlier, Ubi-cook can include features that start its camera feed depending on changes in the environment. A timer can be set to scan at specific times of the day or whenever new ingredients are added to your kitchen, eliminating the need to interact with the scanner to input food.

Expanding upon the usage of features and capabilities within the Kinect SDK that were not leveraged, there is an expansive set of interaction based future work we see possible for Ubi-cook. Gesture-based control could allow for interaction flexibility by scrolling through recipe results with a flick of the wrist, or dismissing results with a wave of the arm. Expanding this further, refinement of skeletal tracking could enable activity awareness, such as differentiating between the act of chopping vegetables or stirring a pot. This system awareness would aid in the actual food preparation and cooking process by dynamically advancing through content depending on current procedure status.

Voice-based control could enable users to interact with and customize Ubi-cook by using keywords for recipe refinement and commands to retrieve recipes and to get/reset searches. Ultimately, we see Ubi-cook moving beyond functioning solely as an automated recipe finding application and becoming an all-purpose "Smart Kitchen" tool.

Moving Beyond Tools
Delving deeper into the future of Ubi-cook, we imagine a robust and intelligent system that incorporates itself into your life as an integral part of your health. By understanding what food you buy and what meals you prepare, it would keep track of information pertaining to an individual's diet and provide insight. With an increasing trend towards lifelogging and the "Quantified Self", functions such as a dietary tracker could collect nutritional metrics that might provide suggestions to improve eating habits, leading to healthier dietary habits. Incorporating an individual's health record into Ubi-cook could enable system awareness of food allergies, dietary needs, or dietary restrictions. Using this information, Ubi-cook, for example, could filter recipe recommendations for diabetic individuals to maintain optimal blood sugar levels. To expand on this idea, the system could communicate with a diabetic user's blood glucose meter and take into account what meal options would raise, lower, or maintain their current levels for optimal health.

Conclusion
Cooking no longer needs to be a drudgery. Ubi-cook, a Kinect based application for the common household kitchen, enables people everywhere to take the supplies they have on hand and create intelligent solutions for meals. Our application automatically analyzes what the user has on hand in order to suggest recipes, reducing the headache of finding and cooking healthy meals. This, alongside the decreasing cost of scanners, such as the Kinect, will aid in the creation of smarter kitchens. Besides making cooking more enjoyable, Ubi-cook provides time-saving and cost-effective methods to assist everyday people with their kitchen inconveniences.

References

[1] Abdel-Hakim, A., and Farag, A. CSIFT: a SIFT descriptor with color invariant characteristics. vol. 2, IEEE (2006), 1978–1983.

[2] Brower, K. An ingredient based recipe search engine - recipe puppy.

[3] Chang, C.-C., and Lin, C.-J. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2*, 3 (Apr. 2011), 1–27.

[4] Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding 106*, 1 (Apr. 2007), 59–70.

[5] Google. Google goggles - overview.

[6] Google. Search by image inside search google.

[7] Lent, D. Insider | quirky.

[8] Li, F.-F., and Perona, P. A bayesian hierarchical model for learning natural scene categories. vol. 2, IEEE (2005), 524–531.

[9] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (Nov. 2004), 91–110.

[10] News, U. Americans eat out about 5 times a week. *United Press International, Inc.*.

[11] royshil. Foodcamclassifier.

[12] Studio Diip. Vegetable recognizer - studio diip - YouTube.

[13] yankeyan. Object recognition using kinect on the PC - YouTube.