Appliance Reader

Antonius Denny Harijanto
Honor Thesis
The Department of Computer Science and Engineering
University of Washington

June 13, 2013

Abstract
Nowadays, almost all appliances are embedded with digital displays. Unfortunately, due to its non-tactile nature, digital display is impractical for blind/visually-impaired user. The goal of Appliance Reader project is to build a mobile application that aids blind/visually-impaired people reading digital display. Appliance Reader allows users to interpret the reading of a digital display by taking an image of the display using their smartphone. Using computer vision and machine learning algorithms over the images captured by the phone, the semantic of the display is interpreted and delivered to the user using text-to-speech engine.
# Contents

1 Introduction .................................................................................................................. 3  
2 Approaches .................................................................................................................. 4  
   2.1 Database Creation ................................................................................................. 4  
   2.3 User Application ................................................................................................... 5  
3 Technical Overviews ................................................................................................... 6  
   3.1 Reference Image and Annotations ........................................................................ 6  
   3.2 Image-taking phase .............................................................................................. 7  
   3.3 Feature detection, matching, and perspective transformation ......................... 7  
4 In-depth: Feature Detection, Matching, and Homography ......................................... 9  
5 Annotations and Information Extractions .................................................................. 13  
6 Interpretations ............................................................................................................ 15  
7 Future Works ............................................................................................................. 16  
8 Acknowledgments ....................................................................................................... 18  
9 References .................................................................................................................. 18
1 Introduction

The goal of Appliance Reader research project is to provide for blind/visually-impaired user access to digital display through mobile application. Appliance Reader project uses smartphone, which is ubiquitous and relatively inexpensive these days, to make digital display accessible for blind/visually-impaired people.

Jane is blind and in need of a new wall oven for her apartment. As she shops for the new oven she discovers that almost all ovens have digital displays for showing the state of the oven: its temperature and the time left on its timer. They also have buttons flush with the display panel for providing input. She knows from past experience that she can put Braille labels on the buttons and will eventually memorize which button does what, so input will not be a problem. However, how will she read the digital display?

These days, newer appliances are usually integrated with digital display. Due to the non-tactile nature of digital display, those appliances are not very accessible to blind/visually-impaired user. Though specially made blind-accessible appliances exist, they are often very expensive because of their limited market size.

Appliance Reader project aims to allow blind/visually-impaired user to interpret the reading of a digital display using smartphone. The user first takes an image of the digital display he wants to read using the camera on their smartphone. Appliance Reader smartphone application then uses a set of computer vision and machine learning algorithms to interpret the reading of the display from the user-taken image. After the interpretation is finished, it reads out what on the display using a text-to-speech engine.

Appliance Reader project uses OpenCV open-source computer vision library. OpenCV is used not only because it has the implementations for most of the necessary computer vision algorithms that Appliance Reader requires for its tasks, but also because it runs across different platforms. Some of the prototyping for Appliance Reader project were done as desktop applications, while the main smartphone application is implemented to run on Android devices.

By the time of the writing of this paper, Appliance Reader project is still an on-going research. The main focus of this paper is to present the key concepts of how the Appliance Reader is intended to work. Some of the ideas, difficulties, and challenges encountered for what have been implemented so far are also presented within this paper.
2 Approaches

This section illustrates the high-level diagram of the two components of Appliance Reader: Database Creation and User Application.

2.1 Database Creation

![Diagram]

1. **Take the image to be used as a reference image**
2. **Annotate and label the regions using LabelMe**
3. **Create a database about how the annotations are used to come up with the final result.**
2.2 User-Application

1. Take image of the display to be interpreted.
2. Feature detection is run over the user-taken image.
3. Compute feature matches between reference image and user-taken image.
4. Warp the user-taken image to overlap the reference image.
5. Extracts the regions that are labeled using LabelMe.
6. Interprets the regions using machine learning.
7. Speaks out the result using Text-To-Speech engine.
3 Technical Overviews

3.1 Reference Image and Annotations

Appliance Reader has a specialized database for each of the appliances that it supports. This database contains the information necessary to interpret the image of the digital display that the user takes. This information includes a reference image and annotations. A reference image is an image Appliance Reader refers to when it interprets the user-taken image. Annotations consist of labeled regions over the volatile components of the digital display. The annotations are done using LabelMe, which is an online open annotation tool. Figure 3.1 illustrates a reference image with some annotations drawn over it. Each of the boxed digits and symbols on the display are annotations. LabelMe produces XML file that encapsulates information regarding the annotations.

![Figure 3.1. Annotated Reference Image](image)

Figure 3.1 shows an image of a thermometer/hygrometer annotated on its temperature and the percentage of humidity digits.
3.2 Image-Taking Phase

The first step is for the user to take an image of the digital display to be interpreted using the camera on the smartphone. A single image or multiple image frames can be taken for this phase. Due differences in the amount of computational of different smartphones, the image resolution is kept at a relatively small VGA resolution. Not only that smaller image resolution reduces the time it takes to run the computer vision algorithm, but it also uses smaller memory.

![Image taken by the user to be interpreted](image1.png)

Figure 3.2. Image taken by the user to be interpreted

3.3 Features Detection, Matching, and Perspective Transformation

Given an image of the digital display that the user has taken, Appliance Reader transforms the image to overlap the reference image. This transformation is called Homography or also known as projective transformation. To do this transformation, Homography matrix needs to be computed. When applied to the user-taken image, this Homography matrix warps the image to overlap the reference image. In order to compute Homography matrix, at least four valid correspondences are required. Correspondence is a mapping of a point on the original image into the same point on the warped image. The idea of correspondence is illustrated on Figure 3.2.

![Correspondences drawn over warped square image](image2.png)

Figure 3.2. Correspondences drawn over warped square image
To compute correspondences, feature detection and feature matching algorithms are used. Features can be thought as interesting or distinguished parts detected over an image. After the features are computed on the two images, correspondences are found by matching the features from one image to the other. Figure 3.2 illustrates the idea of features and matches. Each of the corners on the rectangle is a feature. The colored lines indicate matches among the features. There are different kinds of features with different kind of characteristics. The one that is used for Appliance Reader is ORB feature detector that is based on BRIEF feature descriptor.

After the user-taken image is warped, the next step is to look up the coordinates of the labeled regions from the appliance database, extract the annotated regions from the warped image, interpret them, and finally put together the semantics of the display reading. Individually annotated digits or symbols on the digital display are separately interpreted using machine-learning algorithm. The final semantics of the digital display is deduced by combining the individually interpreted annotated regions. This idea is illustrated by figure 3.3. Annotated regions from the image on figure 3.1 are extracted, interpreted using machine learning, and put together to form the word “Monday”.

![Figure 3.3. Semantics interpretation](image-url)
4 In-depth: Feature Detection, Matching, and Homography

From an image, different kinds of features with different properties and invariances can be computed. For the illustration with two rectangles on figure 3.2, corner happens to be the features that the original and warped images share in common. Since the goal of feature detection in Appliance Reader project is to do a perspective warping using Homography matrix, the features to be computed are ideally better if they are invariant towards scale, rotation, perspective changes, different illumination, or many other related aspects. The reason is the user-taken image might be taken from a different distance, different angle, or even different ambient situation.

SIFT feature algorithm is one of the best detection algorithms in a number of applications such as image stitching, object recognition, or visual mapping. However, at the time of research SIFT was no longer available on the latest OpenCV library for Android. The reason is that SIFT was just patented and even though it is possible to re-compile the library to enable it back, it was rather unclear if it will still be supported on the future version of OpenCV library or not. Because of this ORB detection algorithm is used instead in place of SIFT. ORB uses variant of BRIEF descriptor, which performs as good as SIFT in many respects such as robustness to lighting, blur, and perspective distortion [1]. More than that, ORB algorithm is also about two magnitudes faster than SIFT, which makes it beneficial for smartphone with a relatively limited amount of computational power and no GPU acceleration.

The first step to do the transformation is to compute features on both the reference and the user-taken image. Figure 4.1 shows computed ORB features on the image, indicated by colored circles. After features are computed on both images, they are matched to each other. Matching is done by comparing how similar a feature is to any of the features on the other image. Figure 4.2 shows features matches computed between the two images on Figure 4.1. Among the computed feature matches, there are inliers and outliers matches. Inliers are valid matches and outliers are the invalids ones. The homography matrix is estimated using the correspondences. If there are more valid correspondences, the chance of successfully estimating the matrix increases. To increase the success probability of computing a homography matrix, the number of outliers needs to be suppressed. For this, non-deterministic RANSAC algorithm is used. RANSAC is an iterative method to estimate mathematical model from a set of observed data that contains outliers [5]. Given as an input a set of correspondences with inliers and outliers, RANSAC algorithm outputs subset of matches with certain probability of having more inliers.
Figure 4.1. Features computed on the reference image (left) and on the user-taken image (left) using ORB algorithm.

Figure 4.2. Correspondences computed using brute-force matching algorithm
After correspondences are found, homography matrix can be computed. Homography matrix is computed by solving the following matrix equation:

\[
\begin{bmatrix}
  x'_i \\
  y'_i \\
  1
\end{bmatrix}
= 
\begin{bmatrix}
  h_{00} & h_{01} & h_{02} \\
  h_{10} & h_{11} & h_{12} \\
  h_{20} & h_{21} & h_{22}
\end{bmatrix}
\begin{bmatrix}
  x_i \\
  y_i \\
  1
\end{bmatrix}
\]

\(X_i\) and \(Y_i\) are the coordinate of a point on the original image and \(X'_i\) and \(Y'_i\) are the coordinate of the same point after warp. \(h_{00}\) through \(h_{22}\) are components that build up homography matrix. Image transformation is done by applying the matrix to each of the points on the image. The equation above can be rearranged as follow and then approximated using statistical approaches such as least square. At the very minimum, four valid correspondences are required to solve the equation. But the more the correspondences are, the higher the chance of computing the correct matrix. Figure 4.3 shows the result of applying homography matrix to the original image.

\[\begin{bmatrix}
  x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\
  0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\
  \vdots \\
  x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\
  0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \\
\end{bmatrix}
= 
\begin{bmatrix}
  h_{00} \\
  h_{01} \\
  h_{02} \\
  h_{10} \\
  h_{11} \\
  h_{12} \\
  h_{20} \\
  h_{21} \\
  h_{22}
\end{bmatrix}
\]

Figure 4.3. User-taken image transformed using Homography matrix.
Computation of homography matrix using correspondences is not always perfect. Sometimes it produces incorrect matrix that warps the input image into an unrecognizable image. Other times, it produces warped image that does not perfectly overlap with the reference image because of some translation error. One of the reasons for this is because there might be too many outliers among the correspondences. Each of these outliers contributes to certain degree of error in the matrix computation. Having more inliers amongst the correspondences can increase success probability of computing the matrix.
5 Annotations and Information Extractions

After the user-taken image of the display is warped to overlap the reference image, the next step is to extract information from it. Each of the digital displays is annotated on certain regions of its volatile display. Each of these regions is interpreted individually using machine-learning algorithm. These individual interpretations are put together to form the final semantics of the digital display. Finally, the result is read out to the blind/visually-impaired user using text-to-speech engine.

Annotations are done using LabelMe open annotation tools, which is a web application that lets its user puts labeled polygons over an image. LabelMe produces XML file that encapsulates information about the polygons’ coordinates and labels. For each of the appliances, there is a specialized LabelMe XML file for it.

The annotations are specifically chosen so that the number of positive samples that needs to be provided to the machine learning algorithm can be small, which means less training is needed for the machine learning algorithm. For the wristwatch example on figure 3.1, the top left of the display indicates day of today. It looks as follow:

To interpret this, one may annotate on each of the letters, interpret each of them individually using machine learning algorithm and finally put together the interpretation results to form a meaningful semantic.

Another approach to interpret the day is by annotating it over the entire letters and interprets the final result immediately using machine learning.

The first approach of annotating requires the machine-learning algorithm to be able to recognize all possible letters in every possible day. On the other way, the learning algorithm needs to only be able to recognize seven possible days: Monday through Sunday.

After the annotated regions are extracted and interpreted, the next step is to put together the final semantic of the digital display. How the annotated regions are put together to form the final semantic is specified on the appliance’s specific database. Finally, the final semantic is read out to the blind/visually-impaired through text-to-speech engine. Figure 5.1 illustrates the idea of individual regions forming the final semantic. “69 Degree of Fahrenheit, 43% Humidity” is the final semantic that is read out using text-to-speech engine.
Figure 5.1. Annotations extracted from the appliance

“69 Degree of Fahrenheit, 43% Humidity”
6 Interpretations

6.1 Machine Learning

After the annotated regions are extracted from the warped user-taken image, the next step is to interpret the semantic of the region. Even though homography matrix warps the user-taken image to overlap the reference image, aspects such as brightness, reflection, and contrast cause the reference image and the warped user-taken image to always be slightly different. This motivates the use of machine learning to solve the variability of the problems.

6.2 ExemplarSVM

Appliance Reader uses machine learning to do object detections over the annotated regions. The machine-learning algorithm requires training in order to be able to detect the semantic of the image. Since computer sees raw image merely as pixels of color at different two-dimensional positions, more meaningful features out of the image need to be computed for the learning algorithm. Histograms of Oriented Gradients (HOG) are feature descriptors used for the purpose of object detections. It divides the image into smaller regions and computes the histogram of oriented gradients over them. The idea of HOG feature descriptor is to represent an image using its edge directions. Moreover, HOG descriptor is also invariant to geometric and photometric transformation. This HOG representation of the image is used to train the learning algorithm.

There are many different kind of machine learning algorithms that can be used. Most learning algorithms require large number of positive and negative training samples (possibly in the order of thousands) for training purposes. However, taking that many sample images for each of the supported appliances is not very feasible. For this, the use of machine learning algorithm ExemplarSVM is proposed. ExemplarSVM algorithm requires only a single positive training sample. The algorithm is capable of generating millions of negative training samples. By the time of writing of this paper, this phase is still on going.
7 Future Work

7.1 Running the algorithms Over Stream of Images

7.1.1 Problem
The current approach of having the user to take only a single image and then running the entire interpretation pipeline over the image is prone to failure. If the user-taken image is not very good, for example because it is blurry or has too much reflection, it may fail one of the intermediate steps in the interpretation pipeline. When one of the steps fails, the entire pipeline needs to be repeated possibly with different user-taken image. When this happens, the user needs to retake the image and wait for the algorithms to finish.

7.1.2 Possible Solution
Instead of having the user to take only a single image of the display, Appliance Reader may have the user to take video streams of the appliance display. The interpretation pipeline is then run on each of the images in the stream until successful interpretation result is obtained.

7.2 Glare/Reflection reduction

7.2.1 Problem
Appliances often have reflective surface of its display. Due to the surrounding lighting conditions, sometimes there is too much reflection/glare that causes the display interpretation to fail.

7.2.2 Possible Solution
Some kind of glare/reflection reduction algorithm might be used to improve the user-taken image. One of the works from the Mobile Accessibility research group allows reduction of glare/reflection by using two images of the same object.

7.3 Voice-guided picture taking

7.3.1 Problem
Blind/visually-impaired user might have trouble to aim the phone’s camera to the appropriate region of the display.
7.3.2 Possible Solution
Voice-guidance can be added to Appliance Reader to help the user taking picture of the digital display.

7.4 Utilizes the characteristics of the digital display
Since Appliance Reader has appliance-specific database, properties of the digital display to be interpreted is already known by Appliance Reader ahead of time. This ahead of time knowledge about the appliance can be used to improve on the result of feature detection and feature matching. Features from the volatile area of the display can be excluded from the computation of feature matches. Features coming from the background of the appliance might also be excluded by using image filtration, because the properties of the appliance is already known (e.g. Its color)

7.5 Open database
Crowd sourcing can be used to have Appliance Reader support more appliances. An online service such as Mechanical Turk can be used to collect more appliance data.
Acknowledgment

I thank Prof. Richard Ladner and Bryan Russell for all of the guidance and ideas. I also want to thank Michael Hotan, my collaborator on this research. This research project would not be possible without their continuous dedication throughout the past year. This research uses open source project OpenCV and open mobile platform Android. I would like to also acknowledge Intel Science and Technology Center.

References

4. UW CSE 455: http://www.cs.washington.edu/education/courses/cse455/12au/