Mobile Tools for Point-of-Care Diagnostics in the Developing World

Nicola Dell and Gaetano Borriello

Computer Science and Engineering, University of Washington
{nixdell,gaetano}@cs.washington.edu

ABSTRACT

Remote health monitoring and disease detection in the developing world are hampered by a lack of accurate, convenient and affordable diagnostic tests. Many of the tests routinely administered in well-equipped clinical laboratories are inappropriate for the settings encountered at the point of care, where low-income patients may be best served. To address this problem, medical researchers have developed innovative rapid diagnostic tests (RDTs) that are capable of detecting diseases at the point of care within a single patient visit to a clinic. However, for these new diagnostic technologies to be effective, tools must be developed to support the health workers who will be responsible for administering the tests and interpreting their results. This paper describes the design and initial implementation of ODK Diagnostics, a smartphone application that supports health workers in three ways: (1) by facilitating the creation of digital job aids that provide in-context assistance to users administering RDTs, (2) by automatically interpreting the test results and delivering the diagnosis, and (3) by automating the data collected regarding the type and outcome of the test. Our technical evaluation suggests that the system is capable of accurately reading RDT results and is ready to be field tested with health workers to ensure that it is usable and appropriate for point-of-care settings in developing countries.

Categories and Subject Descriptors

I.2.10 [**Vision and Scene Understanding**]: Intensity, color, photometry, and thresholding

General Terms

Design, Measurement, Human Factors

Keywords

Computing for development, DEV, ICTD, point-of-care diagnostics, POC, rapid diagnostic tests, RDT, job aids, protocols, computer vision, mobile phone, smartphone.

Figure 1: A low-cost, disposable rapid diagnostic test for HIV showing a positive test result.

1. INTRODUCTION

One of the great challenges in science and engineering today is to develop technologies to improve the health of people in the poorest regions of the world. Providing communities living in these regions with access to accurate, convenient and affordable diagnostic tests could facilitate the control of many infectious diseases. However, many of the diagnostic tests that are routinely administered in clinical laboratories in developed countries are inappropriate for point-of-care settings in developing countries, where low-income patients may be most accessible. As a result, these patients do not have access to adequate medical testing and suffer a high mortality rate from several diseases that could otherwise be quickly diagnosed and cured.

Recent medical research has led to the emergence of innovative rapid diagnostic tests (RDTs) that specifically target the needs of patients in developing countries. These low-cost, disposable tests are capable of diagnosing a variety of common diseases, such as malaria, syphilis and HIV. An example of an RDT for HIV is shown in Figure 1. RDTs contain all of the elements required to process a biological sample at the point of care, so that the sample does not need to be refrigerated, protected, or transported. Additionally, many of these tests run rapidly, allowing medical personnel to view results and treat patients immediately. This speed is advantageous because many rural patients travel long distances to reach medical facilities and may be unable to return easily to collect test results, delaying treatment.

However, although the potential benefits of these new diagnostic technologies are immense, initial research has shown that health workers often make mistakes when administering the test or interpreting the results by eye [14]. The number and variety of RDTs that are available is also expanding rapidly, and different tests often require slightly different procedures or interpretation. In addition, medical researchers are in the process of developing more sophisticated tests whose results require quantification or time-sensitive analysis. For example, researchers at the Burnet Institute are developing a rapid test to quantify a patient's CD4 count [27], while Stevens et al. demonstrated
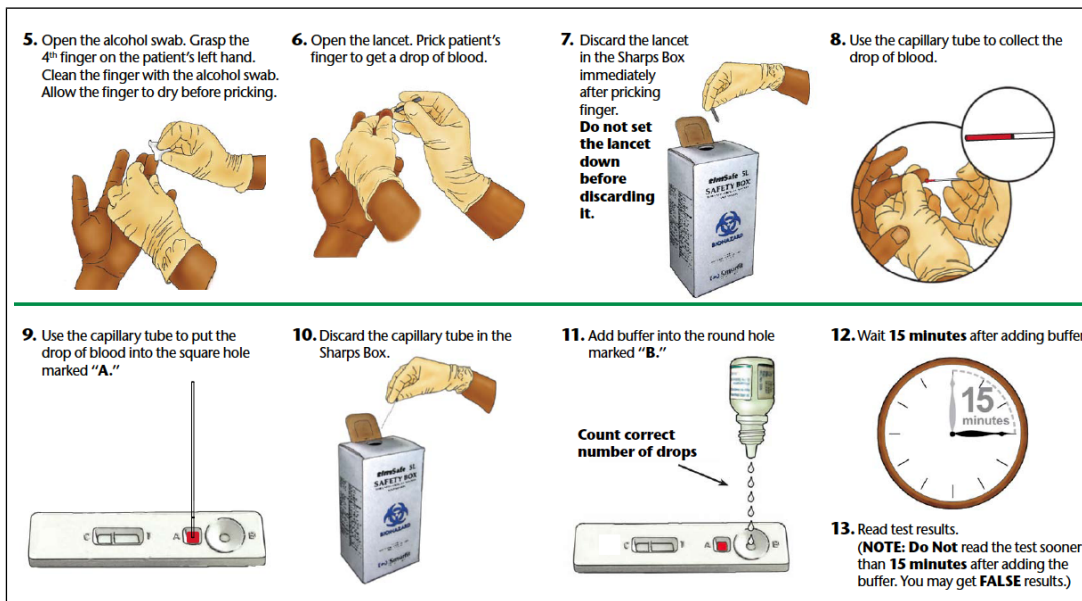
Figure 2: A portion of the paper-based malaria RDT job aid that assists users with running the test.

that time-sensitive analysis of a microfluidic assay can substantially increase the dynamic range of the test [26]. These tests will require health workers to simultaneously manage the timing sequences for the test, monitor the rates at which test signals are changing, and quantify the overall result. Furthermore, manually recording data about the RDTs administered, and updating patients' records with the test results can be a complex and time-consuming process.

These challenges suggest that there is a need for tools to assist health workers with administering and interpreting RDTs. One set of tools that has been developed to this effect is paper-based job aids. A job aid is typically a one or two-page instruction sheet that walks the user through the process of running an RDT. Figure 2 shows part of a paper-based job aid. Job aids have been shown to increase the proportion of tests that are administered correctly [14]. However, paper-based job aids are difficult to distribute, translate and update. Moreover, as the tasks become more complex, the job aids become more difficult to navigate.

As an alternative to paper-based aids, prior research has demonstrated the potential for mobile technologies to strengthen healthcare systems [7] [19]. In particular, the growing adoption of smartphones in developing countries suggests that they may be an effective platform for a point-of-care diagnostic system. Smartphones are portable and battery powered, and come pre-installed with cameras that are capable of capturing high-quality images and videos to facilitate automated image analysis of the test results.

Broadly, this work seeks to answer the following research question: *How can we support health workers as they are required to make increasingly complex diagnostic decisions for an increasing number of diseases and medical conditions?* To address this challenge, we designed ODK Diagnostics, a smartphone application that supports health workers in three ways: (1) by facilitating the creation of digital job aids that provide in-context assistance to users administering RDTs, (2) by automatically interpreting the RDT results using computer vision algorithms running on the phone, and (3) by automating the process of collecting data regarding the test administered and its outcome.

The main contribution of this paper is an end-to-end solution for administering and analyzing a variety of commercially available RDTs. Our solution provides several key benefits over existing solutions, including:

- Digital job aids can provide in-context, on-demand assistance to health workers in the field.
- Commercially available smartphones can be used to analyze a wide variety of different tests, negating the need for specialized reader devices.
- The system can handle both simple, binary tests and more complex tests that may require quantification or time-sensitive analysis.
- The system can easily keep a record of all the tests administered and their outcomes for quality control monitoring and evaluation.
- Analyzing the test results can become a standardized, auditable and adjustable process without needing to retrain users.
- New diagnostic tests can easily be added to the system as they become available.

2. RELATED WORK

There is a wide range of research that demonstrates the potential for mobile technologies to strengthen healthcare systems in developing countries. Categories of interventions include informing people about health issues [2], providing remote medical consultation [24], and enabling data collection and retrieval [13]. We focus on two categories of related research: systems that support health workers, and systems for automatically processing point-of-care diagnostic tests.

2.1 Health Worker Support Systems

The lack of highly-trained medical professionals in many developing countries means that community health workers are often relied on to provide critical health services in rural regions. This has resulted in a large amount of research that focuses on improving health worker effectiveness. For example, Rowe et al. [23] describe strategies for better health worker training, while DeRenzi et al. [6] target health

worker reminder systems. Mobile devices have been used to aid decision making in a number of different scenarios. Mhila et al. [19] describe CommCare, a phone-based application that supports health workers as they provide home-based care and social support to patients. DeRenzi et al. [7] developed eIMCI, an application that guides health workers through a digital version of the Integrated Management of Childhood Illness (IMCI) algorithm. These systems differ from ODK Diagnostics in that they provide health workers with a series of simple questions, and the treatment or diagnosis is recommended based on the answers to these questions, while ODK Diagnostics is more of an instructive tool that guides users through the completion of tasks, like drawing blood from a patient. Research has also shown that job aids can help health workers perform preventative health tasks and reduce the resources needed for training [14]. In addition, job aids can also help health workers build patient counseling skills, and have been found to promote patients' adherence to medical regimens [9]. ODK Diagnostics aims to combine the benefits of job aids with the advantages of mobile technologies to create a point-of-care diagnostic system that aids disease diagnosis in remote areas.

2.2 Automated Processing of Diagnostic Tests

There are a several other projects that couple image processing with diagnostics tests. Mudanyali et. al [20] have developed a smartphone-based RDT reader platform that can work with several RDTs. Their platform differs from ODK Diagnostics in a number of ways. First, although the system works with a variety of RDTs, each test type requires a custom built holder that fits tightly around the test cartridge and clips it to the rest of the platform. In contrast, the ODK Diagnostics stand is capable of fitting a wide variety of RDTs that have different shapes and sizes without the need for any additional parts. Their platform also requires a plano-convex lens, three LED arrays, and two AAA batteries to precisely control the environment in which the test image is captured. ODK Diagnostics does not require any of this additional hardware, which is advantageous since requiring additional parts increases the likelihood that something will get lost or broken and lower the reliability of the system or render it non-functional. Finally, Mudanyali et al. focus entirely on optically reading the RDT results, and do not consider the human challenges that administering RDTs may represent for lightly trained health workers, whereas ODK Diagnostics aims to support health workers with the process of administering RDTs correctly in addition to automatically reading the results.

Dell et. al [5] built an application capable of automatically quantifying diagnostic test data on a smartphone. The application targets one sophisticated diagnostic test and shows that the phone is capable of correctly analyzing time-sensitive data and quantifying the test result. Although we hope to target sophisticated tests like this in the future, they are not yet ready to be field-tested or marketed commercially. Therefore, we have built ODK Diagnostics to work with today's commercially available RDTs, and have designed the system to be capable of handling more complex tests as they become available. Our hope is to have a stable, usable and tested platform already in place to which new, more sophisticated tests can be easily added.

Other projects that utilize smartphones to read diagnostic tests include Skannex [25], who market a proprietary test reader system in which each test is marked with an identifying barcode, and a paper-based dengue test that can be imaged and processed by a smartphone [17].

3. DESIGN

We designed ODK Diagnostics to assist health workers with administering and interpreting a variety of RDTs. We wanted to create a platform that can be easily extended to incorporate many RDTs of different types and formats and so we have tried to minimize the technical knowledge required to add a diagnostic test to the system. However, we anticipate that the process of adding an RDT to the system will not be done by the health worker who will use the system, but rather by someone higher up in the information hierarchy and then distributed to health workers for use in the field. ODK Diagnostics aids health workers in three ways: (1) by providing a variety of digital job aids to give in-context assistance to users administering RDTs, (2) by automatically analyzing the RDT and delivering the diagnosis, and (3) by automating the data collection regarding the type and outcome of the test. We have designed these components so that they can be installed and operated independently of each other, since some programs may want to make use of digital job aids but not automatic interpretation of RDTs, while some may elect to skip the job aids and use only the component for automatically interpreting RDTs.

3.1 Digital Job Aids for RDTs

A job aid is typically a one- or two-page instruction sheet or poster that walks the user through a series of steps to perform a specific task. Job aids range from covering simple step-wise processes to more complex decision-making including performing calculations along the way. In the public health context in developing countries, job aids are often designed to assist health workers who usually have limited training, such as a single brief training session on a particular procedure. In this regard, job aids are particularly important in guiding health workers through procedures they rarely perform or that are complex to conduct. Research has shown that the use of job aids designed to guide health workers through the process of administering RDTs can significantly increase the proportion of tests that are administered and interpreted correctly [14].

Traditionally, job aids for RDTs are paper-based documents that provide a step-by-step explanation of how to administer an RDT along with clearly worded instructions and visual depictions that help guide the health worker through the procedure. However, these paper-based job aids have a number of limitations. They can get lost or damaged. They need to be printed, transported and distributed to health workers, which makes it difficult to easily update the materials or change the type of RDT being used, which could be problematic since health workers might inadvertently be referring to out-of-date or incorrect information in a paper-based job aid. Furthermore, as the variety and complexity of the RDTs used at the point of care increases, health workers will be required to carry around and refer to a large number of different paper-based job aids. Most importantly, the procedure might be complex and require calculations or decision-making that are difficult to follow on paper, particularly when a health worker is juggling other tasks and interacting with one or more patients.

To overcome some of these limitations, we have designed a system for creating digital, smartphone-based job aids. The rapidly decreasing price of smartphones in developing countries has made them attractive as a platform for a wide variety of health applications, including data collection [13], reminder systems [6] and decision support systems [7]. Using the phone to also provide digital job aids that assist health workers with point-of-care diagnostic tests will eliminate the need for them to transport and protect paper-based job aids. The digital job aid can be programmed to automatically guide the health worker through the correct flow of steps that apply to the particular RDT she is administering, including providing accurate timing for events such as when to read the RDT results, which will increase the likelihood that the test will be administered and interpreted correctly. Digital job aids will also be easier to distribute, update and translate than paper-based job aids, provided each health worker with a device is able to periodically visit an area with Internet connectivity and download the requisite materials, a now common situation in most developing countries. In addition, smartphones are portable and have the capacity to store a large number of job aids, so the device also serves as an instantly accessible RDT instruction manual.

In this paper, we focus on digitally replicating several widely used and WHO approved paper-based job aids, such as the malaria RDT job aid (partially shown in Figure 2). We rely on medical experts to create the appropriate job aid content and are instead focused on building a system that simplifies the process of creating and distributing the job aids. However, the capabilities of the technology mean that more sophisticated job aids could be created that use a variety of additional media, like voice recordings or short videos that demonstrate how to correctly complete tasks.

ODK Diagnostics also leverages the fact that many RDTs share a number of common steps, such as asking the health worker to put on latex gloves, or waiting a specific amount of time after starting the test before reading the results. We allow users to reuse and customize a set of basic job aid template files, rather than requiring these common steps to be reprogrammed every time a new job aid is created. Our design also allows the digital job aids to be used either as a standalone platform or in conjunction with our algorithm for automatically interpreting the RDTs. The advantage of this separation is that the system is not limited to the creation of job aids for RDTs, but could also be used to create and distribute supportive material for other medical procedures.

3.2 Optical Processing of RDTs

Although interpreting RDT results may appear to be simple, research has shown that incorrect interpretation of test results accounts for a large proportion of the errors made by health workers when administering RDTs [14]. In addition, as more RDTs are developed for a wider variety of diseases, the number of different RDTs that health workers are required to use is expanding, and many of these RDTs vary slightly in their interpretation, such as having a different number of test lines that need to be read. In addition, several companies are starting to manufacture multi-RDTs that test for more than one disease on a single cartridge [18], which increases the complexity of correctly interpreting all of the tests' results. Furthermore, more sophisticated point-of-care diagnostic tests are now being developed that will require the user to quantify the test

Figure 3: Low-cost 3D-printed platform used to capture images of RDTs.

result [3] or record the rate of change of the test signal [26], which is likely to be an error-prone process for lightly trained health workers reading the test results by eye.

To address these challenges, we designed an algorithm that automatically interprets the RDT results using computer vision algorithms running on the phone. Although our current implementation focuses on reading simple RDTs that are already commercially available, we have designed the system to be easily extended to more complex tests by using a lightweight test description language to specify the regions of interest on the RDT and how they should be interpreted. Our goal is to have a platform for interpreting more sophisticated RDTs already in place by the time these tests become commercially available.

To make it easy for health workers to process RDTs correctly, we designed and 3D-printed a small stand, shown in Figure 3, that holds the phone in position above the RDT. This stand ensures that users can interact with the system without needing to pick up the phone, and we anticipate that this will be advantageous for health workers who are busy interacting with patients and handling RDTs and biological samples. The stand design leaves the upper portion of the phone exposed so that the system will work correctly with different phone models and varying camera positions. In addition, the position at which the RDT should be placed below the camera is indicated by a rectangular ridge that allows a variety of RDTs of different shapes and sizes to be correctly positioned without needing to adjust the stand or camera placement. In our mechanical design, we sought to develop an approach that requires minimal handling (and thus lowers risk of contamination of both the RDTs as well as the stand and phone). In fact, we are also developing techniques to operate the phone without physically touching it (using audio, accelerometry, and vision). Finally, our design does not require any additional lighting, sensors or batteries to be attached to the phone. Instead, we normalize the lighting of the RDT using a portion of the test strip that does not contain any reactive agent and can therefore be reliably compared to the reactive regions of the test.

3.3 Data Collection

Collecting timely and accurate data regarding the number, type and results of RDTs administered by health workers is extremely important for a variety of reasons. First, de-

cision makers need to monitor consumption of RDTs so that they can ensure health workers are supplied with the correct number of tests to cover the patient population. Oversupply of RDTs will result in expired tests that should be discarded since they might produce inaccurate results if inadvertently used to diagnose patients, while undersupply will result in health workers being unable to diagnose patients since they do not possess the relevant test. RDT consumption data is also important for generating reports for donor and aid agencies regarding the use of funds that they provided. Additionally, providing quick access to accurate statistics regarding the location and results of RDTs administered could help decision makers to analyze the spread of infectious diseases and facilitate the early detection of outbreaks.

RDT data is currently predominantly collected on paper forms. A plethora of prior research discusses the limitations of paper-based data collection [13] [22], and the benefits provided by digital data collection solutions [19] [10]. Thus, our design replaces paper-based collection of RDT data with a digital data collection toolkit running on the device. The type, location and outcome of the test can be collected automatically by the software and stored locally on the phone or uploaded to a remote database. This will substantially decrease the time that it takes to make the data available to decision makers and will also eliminate the potential for transcription and aggregation errors.

4. IMPLEMENTATION

We built ODK Diagnostics as an Android application. The decreasing cost of devices and open source nature of Android, coupled with the growing adoption of the platform in developing countries, made Android an attractive choice. Although we designed ODK Diagnostics as a generalizable tool to assist users with a variety of diagnostic tests, our initial implementation focuses on a set of commercially available and widely used RDTs for malaria and HIV, which will allow us to deploy and evaluate the system with health workers who currently use these RDTs in the field.

4.1 Digital Job Aids for RDTs

We chose to implement the digital job aids using HTML and JavaScript rather than native Android code, since we anticipate that it will be easier for users with limited technical experience to work with HTML rather than modify Android code. There are also a vast number of tools available to assist users with the creation of HTML content. We anticipate that the person who is responsible for creating the digital job aids will not be the health workers who use them in the field, but rather someone higher up in the information hierarchy who has some technical experience.

Each job aid consists of a set of simple HTML pages with a small amount of common JavaScript. To construct a new job aid, users create or modify simple template HTML and JavaScript files. We implemented the navigation and user interface components using jQuery Mobile, and the job aid is packaged as an Android application using PhoneGap [1], which allows it to be installed locally and additionally provides access to native functionality. We are also currently in the process of developing a new version of our data collection tool, ODK Collect, that will also use Javascript and HTML, and we expect to soon be able to implement the job aids directly with Collect and better integrate job aids with data collection tasks.
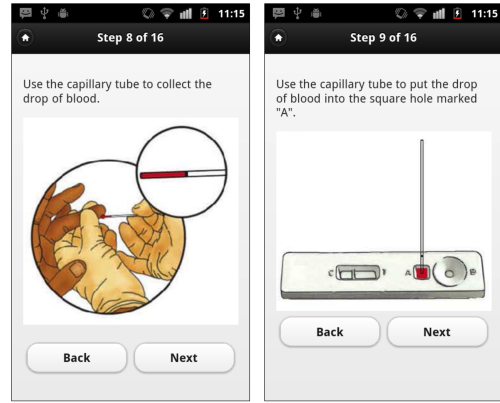

Figure 4: Representative screenshots from the digital job aid for the malaria RDT.

Thus far, we have created digital versions of several paper-based job aids that are used by health workers to administer malaria and HIV RDTs. Figure 2 shows part of the paper based job aid for the malaria RDT. For the digital job aid, each step in the process is shown individually on the phone's screen and numbered to provide some context of where the user is in the process (e.g. Step 5 of 16). Figure 4 shows some representative screenshots of the digital job aid for the malaria RDT. Users navigate the system by pressing buttons or swiping to go to the next or previous step.

When the user reaches the point in the RDT at which she has to wait before reading the results, the application automatically starts a timer, and alerts the user when the correct amount of time has passed and the results are ready for interpretation. To ensure that health workers can use the digital job aids as a standalone tool, we have also implemented the decision tree that guides users through the process of manually interpreting the RDT results. Users can then choose to read the results visually or automatically using the algorithm described in the next subsection.

4.2 Optical Processing of RDTs

After the health worker has been guided through the process of administering the RDT, the next task is to obtain the test result. Our algorithm for automatically interpreting the RDT results is implemented as a native Android application, with the image processing components making use of a Java implementation of OpenCV [21], an open source computer vision library. All of the image processing is performed on the phone without requiring an Internet connection. This functionality will eventually be integrated as a custom *widget* within ODK Collect (similar to how we integrate reading barcodes or obtaining GPS coordinates). There are five main processing steps required to automatically read and interpret an RDT. We describe each of these steps in detail in the sections below.

4.2.1 Create RDT Description and Reference Image

Our system design attempts to minimize the complexity of adding new RDTs to the system. For each type of RDT to be processed, the user must provide two files: (1) an image of an unused RDT that will be used as a template for aligning captured images and (2) a test description file that will be used to locate and identify the regions of interest. We utilize a generalizable JSON [4] format to specify the
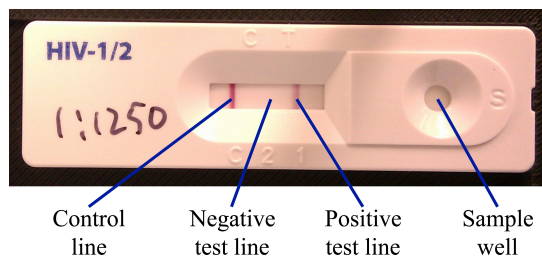
Figure 5: Different regions of interest on an RDT: control line, test lines and sample well.

location, size and type of each RDT region of interest, such as the control lines, test lines and sample wells. Figure 5 shows an example RDT and the relevant regions of interest. In addition, users must specify a region of the RDT strip that is *empty*, in which no test lines or control lines are located. Creating the test description file is a preliminary step that only needs to be performed once for each type of RDT. As with the creation of the job aids, we anticipate that the health workers who use the application in the field will not create the test description file. Instead, the description file will be created by a supervisor or technician higher up in the information hierarchy and distributed to health workers in the field. Our plans for building a tool to allow users to easily create new description files are outlined in Section 6.

4.2.2   Image Capture

Once the test description file and reference image have been created and loaded on to the phone, the application is ready to capture and process RDTs. The first step in the processing pipeline is to use the phone's camera to capture an image of the RDT. To ensure accurate processing, images should be well focused and taken while the camera is steady. The captured image should contain all of the RDT content while maintaining a minimal distance from the test. To make it easy to take photos under optimal conditions, we designed and 3D-printed a small stand that holds the phone in position above the test. As shown in Figure 3, the phone is placed on the stand and the RDT is placed beneath the camera, thereby ensuring that both the camera and the RDT are correctly positioned. While using the stand may not be the most convenient method of capturing an image, it increases the chances that the captured image will be of sufficient quality to be accurately processed. Additionally, making use of the stand may actually be advantageous, since the user is not required to pick up the phone and can instead focus on handling the RDT correctly and minimize contamination of the phone. To capture an image, the user places the RDT in position below the phone and presses a button to launch the Android camera application. The user can then take and retake photographs of the test. When satisfied with the captured image, the user presses a button on the phone to accept the image, which is saved and passed to ODK Diagnostics for further processing.

4.2.3   Image Registration

After an image of the RDT has been captured, the next step is to locate the portion of the image that contains the test content. This involves locating and spatially transforming the picture of the RDT to align with the reference image that is stored on the phone. Alignment is necessary to ensure that the entire RDT has been captured and to determine the locations of each specific RDT element. To perform the image registration, we first convert the captured color image to grayscale, and then use a k-means clustering algorithm [16] to separate the image into the portion that contains the RDT and the portion that contains the background. We then use contour-finding to isolate the set of points that represents the RDT, and compute the minimum-area bounding rotated rectangle that contains this set of points. We rotate the rectangle that contains the RDT to be horizontal, and crop the image to contain only this rectangle. Finally, we compute a transformation that maps the cropped RDT image to the reference image, thereby establishing a point-by-point correspondence between the two images. We then save an image of the aligned RDT and display it on the screen so that users can check the alignment prior to processing the test results.

4.2.4   Locate RDT Elements and Compute Results

Once the RDT has been successfully aligned, the next step is to locate and process those parts of the RDT that depict the test results. To do this, we normalize the color intensities of the captured image and use a Gaussian blur to remove noise. Then, the JSON description file for the RDT is loaded and parsed, and the locations of each RDT element extracted. The algorithm then looks for a field that has been labeled as *empty*. This *empty* region corresponds to a portion of the test in which no test lines or control lines are located. We compute the average pixel intensity of this *empty* region and use it to determine a threshold intensity that will be used to determine whether a line is present in the fields that are labeled as *control* and *test* regions.

After the threshold value has been calculated, each RDT field that is labeled as being either a *control* line or a *test* line is processed. To do this, we perform an image subtraction to determine the difference in pixel intensities between the *empty* image region and the *test* or *control* image region. If the region of the image that we are testing does not contain a line, the results of the image subtraction will be close to zero, since the pixel intensities in the *test* region will be equal to the pixel intensities in the *empty* region. However, if the region that we are testing does contain a line, the results of the image subtraction will differ from zero.

To determine if a line is present, we analyze the results of the image subtraction for every column of pixels in the region being tested. For each pixel in the column, we see if the absolute value of the image subtraction for that pixel is greater than the threshold value we calculated from the *empty* region. To increase the chances that the algorithm will successfully process faint test lines that represent low-positive results, we set the threshold value to be small, but require that the majority of pixels in the column exceed the threshold before we determine the presence of a line.

Once we have determined whether the region being tested contains a line, we use the label of the region to compute the final outcome. If the region is labeled as a *control* region, a line indicates the test is valid, and no line indicates the test is invalid. If the test is invalid, no further results are displayed. The application instead alerts the user that the test is invalid and recommends that the user repeat the test using a new RDT. If the test is valid, then for each region labeled as a *test* region, the presence of a line indicates a positive result, and no line indicates a negative result.

4.2.5 Save Output and Display Results

After all of the *control* and *test* regions of the RDT have been processed, the outcome of the test needs to be saved and displayed in a usable format. To do this, the application constructs a JSON output file that contains the name and type of each RDT region and the corresponding result of processing for that region. In addition to the output file, the application also saves the original captured image and a marked-up image depicting the results of alignment and processing. The application draws a colored box around each region of interest that indicates the result: a green box indicates that a line was detected, while a red box indicates that no line was detected. These saved images can be used to resolve any discrepancies that may arise after the RDT is processed, and will ensure that a record of the test is preserved and archived. The test results are also displayed textually and visually on the phone's screen.

4.3 Data Collection

After the RDT has been interpreted and the diagnosis displayed to the health worker, the next step is to collect and store data to keep a record of the test and its outcome. As discussed in Section 3.3, it is important to collect timely and accurate data regarding the numbers, types and results of RDTs administered and make this data available to decision makers quickly and in a usable format. Since there are already a number of open-source data collection tools available that have been designed and tested in developing countries, we decided not to build our own data collection software. Instead, we chose to integrate ODK Diagnostics with the ODK toolkit, and allow users to export the results of RDT processing to ODK Collect. Integrating ODK Diagnostics with ODK Collect has a number of advantages. First, users of ODK Diagnostics will be able to easily combine the RDT data with other data that might be relevant, including GPS coordinates of the test location or additional patient specific data. Second, through ODK Collect we allow users of ODK Diagnostics to gain access to a number of other useful ODK tools, such as ODK Aggregate and ODK Tables [15], which will facilitate the analysis and visualization of the data collected. In addition to the ODK tools, an increasing number of other platforms, such as Formhub [11] and DHIS [8], have also chosen to integrate their tools with ODK to provide additional services for hosting and data analysis.

5. EVALUATION

The long term goal of this project is to create tools that support health workers as they are required to make increasingly complex diagnostic decisions about an increasing number of diseases and medical conditions. The work described in this paper represents our first steps towards this goal, and as such we have not yet conducted the extensive field work necessary to rigorously evaluate the application with health workers at the point of care. Instead, we focus on evaluating the technical performance of our algorithm for automatically interpreting RDTs in the laboratory. Our plans for field testing the system with health workers in Peru are described in Section 6.

5.1 Apparatus

The experiments described in this section were conducted by researchers at the Global Solutions for Infectious Diseases
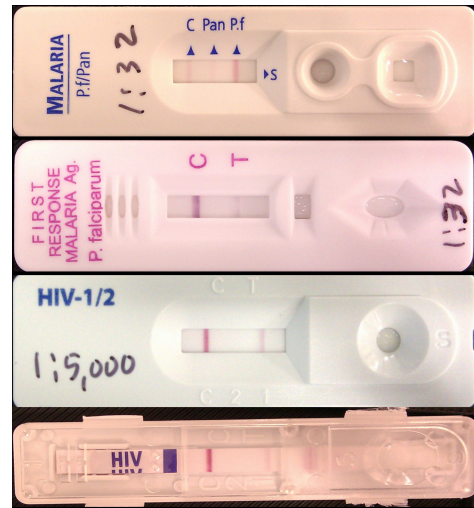

Figure 6: We tested four different RDTs. From top to bottom: SD Bioline malaria with a 1:32 dilution, First Response malaria with a 1:32 dilution, SD HIV 1/2 3.0 RDT with a 1:5000 dilution and SD 1/2 Multi-device RDT with a 1:5000 dilution.

(GSID) [12] laboratory in San Francisco, USA. GSID is a non-profit global health organization engaged in the development of products to help prevent the spread of infectious diseases, especially in the developing world. The experiments were designed to test the robustness and accuracy of the interpretation algorithm using four sets of commercially available RDTs that were administered with varying sample concentrations. We chose to test two malaria RDTs, First Response Malaria and SD Bioline Malaria, and two HIV RDTs, an SD 1/2 3.0 Multi-device HIV and an SD 1/2 3.0 HIV. Figure 6 shows the four RDTs tested. These tests were selected for a variety of reasons. They are all used extensively at the point of care and target diseases that place a significant burden on many developing countries. Additionally, we had access to enough tests to be able to run a dilution series for each test using different sample concentrations, and also possessed the necessary malaria whole blood and HIV positive serum for administering the tests in the laboratory. The experiments were performed using an HTC Nexus One Android device that was placed in the 3D-printed platform depicted in Figure 3. Since we wanted to minimize the complexity of future user training, we set all of the camera parameters to automatic.

5.2 Procedure

Before we could evaluate the performance of our algorithm, we needed to create the appropriate diluted samples and run each set of RDTs. For the malaria RDTs, the dilutions we used were: *undiluted, 1:8 (strong positive), 1:32 (weak positive)* and *negative*. These dilutions were selected based on the researchers' prior experience running RDT dilution series', and knowledge of which concentrations would produce strong positive and weak positive results. Each First Response malaria RDT has one control line and one test line for Plasmodium falciparum, while each SD Bioline malaria RDT has one control line and two test lines, one for Plasmodium falciparum (Pf) and another for Falciparum/Pan (Pan). The malaria whole blood that we used to run the tests contained both Pf and Pan, and so each

Table 1: Experimental results for each of the two malaria and two HIV RDTs tested

		Indoor lighting						Outdoor lighting					
		Control Lines			Test Lines			Control Lines			Test Lines		
	Dilution	Correct	F-N	F-P	Correct	F-N	F-P	Correct	F-N	F-P	Correct	F-N	F-P
SD Bioline Malaria	Undiluted	4	1	0	9	1	0	5	0	0	10	0	0
	1:8	5	0	0	10	0	0	5	0	0	10	0	0
	1:32	5	0	0	10	0	0	5	0	0	10	0	0
	Negative	5	0	0	10	0	0	4	1	0	8	0	2
First Response Malaria	Undiluted	5	0	0	5	0	0	5	0	0	5	0	0
	1:8	5	0	0	5	0	0	5	0	0	5	0	0
	1:32	5	0	0	5	0	0	5	0	0	5	0	0
	Negative	5	0	0	4	0	1	5	0	0	5	0	0
SD 1/2 3.0 HIV	1:50	5	0	0	10	0	0	5	0	0	10	0	0
	1:250	5	0	0	10	0	0	5	0	0	10	0	0
	1:1250	5	0	0	10	0	0	5	0	0	10	0	0
	1:5000	5	0	0	10	0	0	5	0	0	10	0	0
	Negative	5	0	0	10	0	0	5	0	0	10	0	0
SD 1/2 Multi-device HIV	1:50	5	0	0	10	0	0	5	0	0	10	0	0
	1:250	5	0	0	10	0	0	5	0	0	10	0	0
	1:1250	5	0	0	10	0	0	5	0	0	10	0	0
	1:5000	5	0	0	10	0	0	5	0	0	5	5	0
	Negative	5	0	0	10	0	0	5	0	0	10	0	0

positive SD malaria test contains three lines (one control and two test lines), while each positive First Response test has two lines (one control and one test). Each negative First Response or SD malaria test has one control line and no test lines. We included a *negative* test series to ensure that the algorithm could correctly determine the absence of a line.

For the HIV RDTs, the dilutions that we used were: *1:50 (very strong positive), 1:250 (strong positive), 1:1250 (weak positive), 1:5000 (very weak positive)* and *negative.* Each HIV RDT contains one control line and two test lines. However, as shown in Figure 6, the HIV positive serum that we used produces a positive result for only one of the test lines, and so each positive test that we ran contained two lines (one control line and one test line), while each negative test contained one control line and no test line.

Lighting conditions in rural health centers are likely to vary, so in addition to testing the algorithm on RDTs with varying sample concentrations, we also ran tests in both indoor and outdoor lighting conditions. For each RDT, five test images were captured indoors, under the lighting in the laboratory, and another five were captured outdoors in direct sunlight. Our final set of test images for First Response Malaria therefore consisted of 4 dilutions x 5 images of each dilution x 2 lighting conditions x 2 lines per test: 80 lines, while the SD Bioline Malaria tests consisted of 4 dilutions x 5 images of each dilution x 2 lighting conditions x 3 lines per test: 120 lines. For each of the SD 1/2 multi-device HIV and the SD 1/2 3.0 HIV, the experiment consisted of 5 dilutions x 5 images of each dilution x 2 lighting conditions x 3 lines per test: 150 lines each. This makes a total of 500 lines for all of the experiments: 350 positive and 150 negative.

5.3 Analysis

To analyze the performance of the application, we ran our algorithm on all of the test images. For each control line or test line, we categorize a correct result as being a true positive (T-P) if a line is correctly detected, or true negative (T-N) if the absence of a line is correctly detected, and we categorize an incorrect result as being a false positive (F-P) if a nonexistent line is detected, or false negative (F-N) if a line that should be detected is missed. It is interesting to consider the relative importance of each error type. For example, the consequences of a false negative test, which may result in an infected patient not receiving treatment, are more severe than a false positive test, which may result in a healthy patient receiving unnecessary treatment.

5.4 Results

The experiment results are given in Table 1. Out of a total of 140 valid control lines, 138 were correctly interpreted. Two were incorrectly interpreted, and both of these errors were false negatives. One error resulted from a highly overexposed image, depicted in Figure 7. This suggests that there is a need to create a way to automatically check the quality of the image captured to ensure that images are not overexposed, underexposed or blurry. However, it is preferable that any errors that do occur are false negatives rather than false positives, since a false negative would simply result in the RDT being repeated for the patient, rather than a potentially dangerous misdiagnosis.

Out of a total of 150 negative results, 147 were correctly interpreted, and the three errors were all false positives: one with a First Response malaria negative test and two with an SD Bioline malaria negative test. Out of a total of 210 positive test lines, 204 were correctly interpreted. One of these errors occurred due to the same overexposed image described previously and depicted in Figure 7. The other five errors were false negatives that occurred with the SD 1/2 multi-device HIV RDT at a dilution of 1:5000 under the outdoor lighting condition. As shown in Figure 8, we observed that placing the SD 1/2 multi-device RDT in direct sunlight for a few minutes resulted in condensation on the inside of the test cartridge, which made it difficult to correctly process the low-positive lines. The condensation also made it impossible to read the test result by eye, and indicates that this particular RDT should not be used if the test is to be performed outside or in a hot environment.

We also measured the time that it takes to process each RDT. On average, it takes 2.42 seconds to align the captured image, and 5.72 seconds to process the results, giving a total of 8.14 seconds for the entire RDT. This time could be reduced by using a native implementation (rather
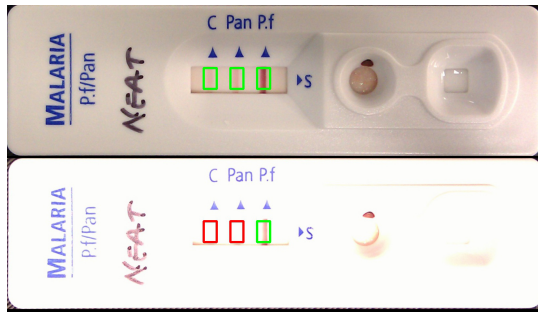
Figure 7: Two marked-up SD Bioline undiluted malaria RDTs: a properly interpreted RDT (top), and an overexposed RDT (bottom) with a false negative control line and one false negative test line.

than a Java implementation) of the image processing code. However, since most RDTs require that the user wait 15 to 30 minutes before reading the test results, we feel that 8.14 seconds is sufficiently fast for our initial field tests.

6.   DISCUSSION AND FUTURE WORK

This paper describes our first steps towards the goal of creating an electronic point-of-care diagnostic system. There are still numerous challenges that need to be addressed to create a usable and appropriate tool for administering and interpreting diagnostic tests effectively at the point of care. However, our initial technical evaluation of the system has raised a number of interesting issues.

First, although running dilution series' may be a good method for initially evaluating the system, the results obtained from diluted samples might not be quite the same as real weak positives. In addition, we do not yet know which sample concentrations represent the limit of what can be visually interpreted by humans, or how many errors health workers make when manually interpreting the test results. To evaluate the system more effectively, we plan to run a trial in which we observe health workers administering RDTs and visually reading the results. We will then use our system to capture images of the same RDTs and automatically process the results. We will then be able to rigorously compare automated analysis with visual analysis.

Second, the sensitivity of RDTs that were manufactured in different batches is known to vary slightly. In addition, for some diseases, like malaria or syphilis, the consequences of a false negative test result are severe, since an infected patient may not receive the necessary treatment. In other cases, a false positive test result might cause a patient to be treated with unnecessary drugs that may have adverse side-effects. These issues suggest that it will be necessary to provide some mechanism for calibrating the sensitivity of the system for each RDT to account for these variations.

Our current design makes use of a low-cost plastic stand to position the phone correctly above the RDT. We anticipate that it will be advantageous for users to not have to pick up and handle the phone, since this will leave their hands free to focus on handling the RDTs and biological samples correctly. However, as with many medical procedures, health workers wear latex gloves when administering RDTs, and although the capitative touchscreen of the device still works with latex gloves, the gloves may hamper the usability of the interface
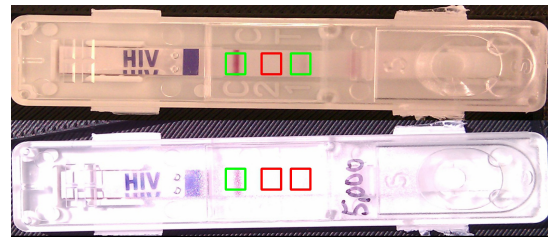

Figure 8: Two marked-up SD 1/2 multi-device HIV RDTs with 1:5000 dilution: a properly interpreted RDT (top), and an RDT with a false negative test line due to condensation on the inside of the RDT cartridge (bottom).

if they are not tight fitting. Additionally, if a health worker is handling infectious material, such as HIV positive blood, it is likely to be undesirable for her to touch the device at all, since she may accidentally transfer some of the infectious material on to the device, and it will then be dangerous for anyone who is not latex gloves to touch the device. This is an issue that is likely to be relevant for many different mHealth applications, since health workers typically wear gloves when they interact with patients, but may not always wear gloves when they handle the device. One solution is to create a hands-free method of interacting with the system, possibly using voice or gesture recognition, and we believe that this topic holds rich potential for future research.

There are a number of additional features that would enhance the utility of ODK Diagnostics. We anticipate that creating the RDT description file is likely to be the most technically challenging task for organizations using the system, and so we will create an application that guides users through the process of creating new test descriptions. In addition, we will create a method for checking the quality of the image captured, so that the system can warn users if it detects that an image is overexposed or out of focus. We also believe that the technology will be particularly useful for automatically interpreting RDTs whose results require quantification or time-sensitive analysis (which are difficult to do by eye), rather than simple binary decisions. Creating algorithms for handling these more complex RDTs will be one of our priorities in the coming months.

We have chosen to integrate ODK Diagnostics with the rest of the ODK toolkit so that it is easy for users to collect and analyze data regarding the number, type and results of RDTs administered. However, it will also be important to integrate the data with an individual patient's medical record so that patient specific data can be recorded and tracked over time. There are a number of medical record systems for Android that are currently in use in developing countries, and it would be advantageous to integrate ODK Diagnostics with one or more of these systems.

ODK Diagnostics provides digital job aids to assist health workers administering RDTs. However, in some cases it may also be appropriate to supply health workers with recommendations for treatment if the test result is positive. Alternatively, the system could suggest additional tests or procedures to try if the test result is negative. For example, if a child has a fever, but the malaria RDT is negative, the system could provide suggestions for other tests that might help to diagnose the patient. Implementing the job aids with the Javascript/HTML functionality of ODK Collect 2.0 will

greatly improve the flexibility of the tools as ODK Collect 2.0 already supports most of these additional features.

Finally, we have not yet rigorously evaluated the usability of the system with front-line health workers in developing countries. We have plans to test ODK Diagnostics in Peru in the coming months and we will use this opportunity to run a study that evaluates the entire system, including comparing the efficacy of the new digital job aids to the paper-based job aids. We look forward to working with health workers to ensure that the system is usable and appropriate given the constraints experienced by these low-resource communities.

7. CONCLUSION

Disease detection and epidemiology are limited by the scarcity of accurate, convenient and affordable diagnostic tests in the developing world. The use of low-cost rapid diagnostic tests has the potential to facilitate the diagnosis and treatment of many infectious diseases and medical conditions at the point of care within a single patient visit to a clinic. However, in order for these new diagnostic technologies to be effective, tools must be developed to support the health workers who will be responsible for administering the tests and interpreting their results. To address this challenge we built ODK Diagnostics, an end-to-end solution for administering and analyzing a variety of existing RDTs. This paper describes the technical implementation and initial evaluation of ODK Diagnostics, which represent only the first steps towards building a effective point-of-care diagnostic system. Our results suggest that the system is ready to be field tested with health workers, and our next steps will involve more focused field studies that rigorously evaluate both the digital job aids and the algorithm for automatically interpreting the test results to ensure that they are usable and appropriate for point-of-care settings in developing countries.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] Adobe PhoneGap. http://phonegap.com/.
[2] V. W. Consulting. mHealth for Development: The Opportunity of Mobile Technology for Healthcare in the Developing World. pages 1–70, 2009.
[3] B. Croal, D. Finlay, E. Davidson, W. Mutch, D. S, I. Rothnie, A. Dickie, and R. Newall. Evaluation of the Bayer Multistix PRO 10LS Point-of-Care Urine Test. *Point of Care: The Journal of Near-Patient Testing and Technology*, 2:144–148, 2003.
[4] D. Crockford. Introducing JSON. json.org, 2009.
[5] N. Dell, S. Venkatachalam, D. Stevens, P. Yager, , and G. Borriello. Towards a Point-of-Care Diagnostic System: Automated Analysis of Immunoassay Test Data on a Cell Phone. In *NSDR*, 2011.
[6] B. DeRenzi, L. Findlater, J. Payne, B. Birnbaum, J. Mangilima, T. Parikh, G. Borriello, and N. Lesh. Improving Community Health Worker Performance Through Automated SMS. In *ICTD*, 2012.
[7] B. DeRenzi, N. Lesh, T. Parikh, C. Sims, M. Mitchell, W. Maokola, M. Chemba, Y. Hamisi, D. Schellenberg, and G. Borriello. e-IMCI: Improving Pediatric Health Care in Low-Income Countries. In *CHI*, 2008.
[8] District Health Information System. http://dhis2.org.
[9] W. Edson, M. Boucar, P. Koniz-Booher, S. Djibrina, and I. Mahamane. Developing job aids to increase adherence to an antibiotic regimen in children with pneumonia in Niger. *Operations Research Results. Published for USAID by the Quality Assurance Project*, 2004.
[10] Episurveyor. http://www.episurveyor.org/.
[11] Formhub. http://formhub.org/.
[12] Global Solutions for Infectious Diseases. http://www.gsid.org/.
[13] C. Hartung, Y. Anokwa, W. Brunette, A. Lerer, C. Tseng, and G. Borriello. Open Data Kit: Building Information Services for Developing Regions. In *ICTD*, 2010.
[14] S. Harvey, L. Jennings, M. Chinyama, F. Masaninga, K. Mulholland, and D. Bell. Improving community health worker use of malaria rapid diagnostic tests in Zambia: package instructions, job aid and job aid-plus-training. *Malaria Journal*, 7:160, 2008.
[15] Y. Hong, H. Worden, and G. Borriello. ODK Tables: Data Organization and Information Services on a Smartphone. In *NSDR*, 2011.
[16] J. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. *5th Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297, 1967.
[17] J. Matthews, R. Kulkarni, G. Whitesides, M. Sarrafzadeh, M. Gerla, and T. Massey. A Light-Weight Solution for Real-Time Dengue Detection using Mobile Phones. In *MobiCase*, 2009.
[18] Medmira. Introducing Multiplo - A Single Test That Gives Multiple Diagnoses in Minutes. *Executive Healthcare Magazine*, 2012.
[19] G. Mhila, B. DeRenzi, C. Mushi, T. Wakabi, M. Steele, P. Dhaldialla, D. Roos, C. Sims, J. Jackson, and N. Lesh. Using Mobile Applications for Community-based Social Support for Chronic Patients. *HELINA*, 2009.
[20] O. Mudanyali, S. Dimitrov, U. Sikora, S. Padmanabhan, I. Navruz, and A. Ozcan. Integrated Rapid-Diagnostic-Test Reader Platform on a Cellphone. *Lab on a Chip*, 2012.
[21] OpenCV. http://opencv.willowgarage.com/wiki/.
[22] T. Parikh, P. Javid, S. K, K. Dhosh, and K. Toyama. Mobile Phones and Paper Documents: Evaluating a New Approach for Capturing Microfinance Data in Rural India. In *CHI 2006*.
[23] A. Rowe, D. de Savigny, C. Lanata, and C. Victora. How can we achieve and maintain high-quality performance of health workers in low-resource settings? *Lancet*, 366:1026–35, 2005.
[24] Sana. http://sana.mit.edu/.
[25] Skannex. Lateral Flow and Elisa Reader Systems. http://www.skannex.com/.
[26] D. Stevens. *Development and Optical Analysis of a Microfluidic Point-of-Care Diagnostic Device*. PhD dissertation, University of Washington, Department of Bioengineering, 2010.
[27] The Burnet Institute. http://www.burnet.edu.au/.