Development of a Tridimensional Measuring Application For iPads

Authors:

Michael Casebolt

Nicolas Kouatli

Jack Mullen

Industrial Technology Department

California Polytechnic State University at San Luis Obispo

26 May 2015

ABSTRACT

In today's fast-paced distribution centers workers and management alike are constantly searching for the quickest and most efficient way to package items for distribution. Even with the advancement of app-oriented solutions to a variety of problems across many industries there is a distinct unmet need in distribution environments for an application capable of increasing the efficiency and accuracy of packaging items. This senior project focused on the development and testing of an application utilizing the Structure Three Dimensional Sensor and a 4^{th} generation iPad to scan an object or group of objects to be packaged and determine the overall dimensions (length, width, & height) of the minimum sized package necessary to contain the objects. In cooperation with a Computer Science student, two Industrial Technology students developed multiple iterations and alternate methods of testing to determine and optimize the accuracy and functionality of the application. This project has been developed in coordination with the technical advisors Industrial Technology professor Javier de la Fuente (PhD) and Computer Science professor John Clements (PhD).

ACKNOWLEDGEMENTS

This research was supported by The Industrial and Packaging Technology Department within The Orfalea College of Business at California Polytechnic State University, who provided the necessary equipment to conduct research and develop the application. We thank our technical advisors Javier de la Fuente (PhD) and John Clements (PhD) who provided insight and expertise that greatly assisted with the development of this application. We thank Jay Singh (PhD) for being one of the initial professors who took interest in this project and provided valuable insight at key times throughout the process. We would also like to show gratitude to Koushik Saha (PhD) for his insight into the development process and progress reports and allowing us to perform this project within his field of study.

TABLE OF CONTENTS

List of Figures:

List of Tables:

SECTION I

INTRODUCTION

Problem Statement

Workers in fast-paced distribution environments need to optimize the process of dimensioning objects for placement into packages. Over the course of this project the team aims to develop a tridimensional measuring application for iPad utilizing Occipital's Structure Sensor, which enables users to scan an object arranged as it would be in the final package and determine the minimum dimensions necessary to contain the objects in a package. The application is being marketed towards workers at large-scale, fast-paced distribution centers that work with a variable product mix and as such are packaging a wide assortment of items that do not normally fit standard box sizes. There is currently no application able to dimension items for packages through the use of a tridimensional scanner. For this project, our team aims to fulfill that unmet need by developing an intuitive and simple application to allow workers with minimal or no training to speed up the process of dimensioning objects for shipping while maximizing the amount of accurate and error free measurements. The team aims to allow application users to email themselves or a central database the dimensions for certain items that are commonly shipped or of particular importance. Currently it is a simple and relatively common mistake for workers dimensioning objects to misread their measuring device or simply forget the values, costing valuable time and energy for the company (Besseris 2013). In theory with this application workers will be able to accurately determine dimensions quickly and accurately, and will be able to save those dimensions to share with others at a later time or streamline the process the next time an object with the same dimensions enters the distribution environment. One issue with the development of this application is the trade-off between speed and accuracy in scanning. The application needs to be able to scan and determine the minimum bounding box dimensions in a shorter period of time than workers are able to do with traditional methods using a ruler or tape measure to figure out the necessary package size. However the app also needs to be accurate enough to justify its use in place of traditional methods. If the app is unable to determine accurate dimensions then it is effectively useless for users. A major part of the development of the application revolves around ensuring it is more efficient than standard methods and allows companies and workers who utilize it to save time and money.

Justification

This tridimensional measuring application for iPads will reduce time spent measuring objects by hand, increasing productivity and resulting in more accurate package dimensions. By reducing time spent determining package sizes this app will allow workers to package more parcels in a shift and allow companies to push more product out in the same amount of time, increasing profits. The app will also allow users to save frequently scanned items or items of importance, meaning easier access to the dimensional data about important or common items. Use of this app will in theory reduce material waste because workers will be able to find or construct the most efficient package to contain the object(s) to be contained, as opposed to current distribution environments where workers choose packages from a set of standard sizes and use void-fill materials to fill the unused space in the package (Besseris 2013). This app should be able to eliminate much of the void-fill material used as well as the material used to construct the packages by determining the minimum dimensions necessary to contain the object(s) meant for distribution. By reducing material needs for packages the application allows companies to save money on material as well as engage in more sustainable practices by eliminating or reducing the over-packaging of objects. For this application the user requirements are as follows in Table I-Tabulated Needs:

Table I- Tabulated Needs

#	**User Requirements/Features**	**Importance Rating**
1	Intuitive interface	3
2	Accurate dimensions	4
3	Option to store dimensions	2
4	Ability to scan wide variety of objects	3
5	Operation of app is fast and smooth	4

Importance Scale: 4= Highest Importance, 1= Lowest Importance

After analyzing the five primary user requirements that were identified, it was determined the highest priority should be placed on ensuring the application is able to have accurate dimension outputs to ensure accurate package dimensions, as well as ensuring the application operation is streamlined, more efficient, and faster than traditional dimensioning methods, otherwise the app loses its inherent value. The whole process of using the app must be hassle-free and painless to the user to facilitate its use and ensure that use of the application streamlines the dimensioning process. The second-highest rated features are the app's ability to scan a variety of objects with ease and having an intuitive interface. In order for this app to be successful it must be able to scan practically any object regardless of its level of complexity, orientation, reflectivity, or color. If the app is not functional in situations where standard methods would be easy for workers to utilize it is unlikely that the app will be successful in streamlining the packaging process. It is also important to users for the app to have an intuitive interface that will not take an inordinate amount of time to figure out or maneuver, even without any prior training. Lastly, users would like the option to store dimensions for later review and to streamline dimensioning procedures. In order to implement this the application will have the option to email the dimensions of an object after a scan is complete, this email will include a 3D model of the object in an object file (.obj) and a picture of the object, and the subject of the email will include the dimensions and a title or identifier for the scanned object to allow for ease of locating. This allows users to save dimensions which are commonly used or that need to be distributed among other workers. By allowing the dimensions to be sent to any email address, all dimensions can be sent to a central account from multiple devices being used at a distribution center. This allows workers to refer back to not only the models that they themselves scanned but also those captured by their coworkers or anyone else using that same email address to reference previously scanned objects. Another advantage of emailing the dimensions is that workers can reference previously recorded values without risking misreporting or misremembering these values critical to ensuring objects are packaged most efficiently with the least amount of material and energy expended during the process.

Background or Related Work

The structure sensor is the world's first 3D sensor for mobile devices. It is attached to any compatible iPad with an easy-to-use bracket to make a completely mobile 3D scanner. The scanner has a built-in battery to avoid draining the iPad's and can actively scan for up to 4 hours on a single charge. The iPad's color camera is utilized to improve the object tracking algorithm and to add high quality textures and color to the captured models (Structure IO).

In order to simplify the process of designing an app for determining the necessary dimensions of packages, we are modifying the current scanning app "Scanner" that has been released for use with the structure sensor. The current app is able to scan an object and create a 3D color model. In developing this application the app is being altered to utilize a minimum bounding box algorithm to have the minimum dimensions for a package containing the scanned object to be displayed on top of the 3D model.

Objectives

This project aims to develop an iOS application by the name of PackIt for iPad using Occipital's Structure Sensor that is capable of taking a tridimensional scan of an object or group of objects and determining the minimum dimensions necessary to contain the objects in a package. The application needs to be intuitive and responsive in order to streamline usage and ensure it offers a viable alternative to current industry practices used in the distribution environment. In order to achieve this the team is working together to design the interface, settings, and instructions for the application, ensuring it is not unnecessarily complicated and that a user can easily figure out how to operate the application without training. Through a process of statistical analysis and testing the operation of the application is verified and confirmed to be accurate and functional. This is accomplished by iteratively altering the programming and configuration of the app until all the application's requirements are met. Experimentation is done with differing work envelope designs to narrow the options down to the best suited envelope for this application.

Project objectives are:

- Determine requirements for a measuring application.
- Design application (interface, settings, background.)
- Develop protocol for measuring (work instructions, standard operating procedures.)
- Application testing and verification.
- Create final report.

Contribution

Once complete the objective is for this application to replace traditional methods of dimensioning objects for packages such as measuring by hand or using packages with the same dimensions regardless of the object being packaged and just using void-fill to ensure empty space is minimized. A worker in a distribution environment traditionally uses a tape measure or calipers and measures each object to be shipped in three directions and records that data, then manually finds boxes that roughly meet the specifications. A worker in a distribution environment using the app that this team is developing will be able to place multiple objects to be shipped on the scanning table (ensuring they are in the same relative position and orientation as they will be shipped in) and move the Structure Sensor-equipped iPad around the objects to determine the minimum dimensions necessary for a package that will contain the objects. Ideally this new method will take less time than traditional methods and will also help workers to avoid Repetitive Stress Injuries (RSIs) because there will be fewer repetitive movements and fewer movements overall in the dimensioning process. The ability for users to store dimensions from scans will assist in the distribution environment by streamlining the process of distributing necessary dimensions throughout workers needing the information. The stored data ensures there are no variations in the dimensions between users and the values remain constant throughout the distribution environment.

Scope of Project

This project aims to develop an iOS application by the name of PackIt utilizing Occipital's Structure Sensor and compatible iPad model that will be able to take a tridimensional scan of an object or collection of objects and determine the minimum dimensions capable of containing the object or collection of objects in a package. The development, testing, and optimization of the application will take place over the course of three quarters, spanning from September 2014 to June 2015. The budget for this project is minimal, however in order to streamline the testing process a new bracket was purchased for $49.99 to allow the sensor to work with a fourth generation iPad.

Deliverables include:

- Tridimensional measuring application for iPad using Occipital's Structure Sensor.
- Standard operating procedures for the use of the application.
- Quality test reports from product testing of the application.
- Final report, formatted to specifications.

SECTION II

REVIEW OF LITERATURE

Introduction/Overview

This project focuses on the development of a novel iOS application (designed for supported 4th generation iPads) that utilizes the Structure 3D Sensor to generate completely custom and functional packaging for both individual products and groups of products. The application is scan-based, meaning that all the information needed to develop a functional product/package system is retrieved simply by 3D scanning the object or objects in question. The application effectively recognizes the object's maximum outer dimensions and outputs those dimensions for recording or emailing to share the values. After a final model is completely rendered, all prior art and information can be accessed and exported for production or other applications.

Three-dimensional scanners are used in a variety of industries to analyze real-world objects and environments to collect data on their shape and appearance. This collected data is then used to construct digital 3D models that are utilized in applications ranging from product development to quality inspection. 3D scanners share traits with digital cameras in that they have a cone-shaped field of view and cannot collect data from obscured surfaces. While cameras capture the colors of an object or surface within its field of view 3D scanners capture distance information using a variety of different methods. The data captured by the scanners describes the distance to a surface at each point in the "picture" taken. Using this information the scanners then create a digital 3D model of the object in its field of view (Hock 2014).

Hardware Background

This virtual package generating application builds off of the functionality of the Structure Sensor, the first three dimensional sensor/scanner ever created for use with a mobile device (Structure IO). The sensor unit itself, measuring roughly 10 centimeters in length, is composed of dual infrared LEDs and a structured light projector housed within a lightweight anodized aluminum chassis that clips onto the backside of any supported fourth generation iPad (Structure IO). Occipital, the company behind Structure, is well known for their advanced computer vision technologies and commitment to bringing seemingly confusing technology to the layman. The Structure Sensor utilizes new 3D imaging technology, known as structured light, projecting an infrared light pattern on an object or scene and recording the distortions in the reflection. The distortions in the projected pattern, created by interference of real-world geometry allows for the Structure to effectively map out and understand the world around it in three dimensions. The Structure Sensor has the ability to recognize and record color information it receives through the iPad's camera function, which allows for capabilities to create real-time, full-color digital 3D models of the world around us. On top of retrieving and processing real-world data, the Structure Sensor allows users to supplement their physical reality using programmed virtual physics. Ultimately, virtual models can be merged with the physical environment and function dynamically as they would in real life. Augmented reality is a fairly novel concept that can be applied across all types of industries, and has already been embraced by the gaming, training, and engineering industries respectively (Hoffman 2014). Virtual reality and the efficiency it allows for opens up entirely new possibilities in the realm of packaging dynamics testing.

One of the most compelling features the Structure Sensor has to offer is its ability to scan objects or scenes of virtually any size. While the Structure is limited in terms of range (maxing out at around 3.5 meters from the unit itself), the mobile nature of the entire systems allows users to scan virtually anything; whether that be sculptures, automobiles or entire rooms (Structure IO). This capability, on top of its fair price, relative novelty and unprecedented resolution makes the Structure Sensor a fierce competitor in the hand-held 3D scanner market.

In the past the majority of 3D scanners utilized laser dot or laser line technology. As structured light technology emerged and advanced it has begun to take over the 3D scanning market. Now roughly half of all scanners introduced to the market utilize structured light technology, typically white or blue LED technology. LED light sources have taken over the market because of their small size, low heat emission, and long lifespan; previously metal-halide lamps were the standard. Advances in imaging technology have led to the move from charge coupled device (CCD) technology to complementary metal oxide semiconductor (CMOS) technology. This change has been the result of significant advances in CMOS-based imaging devices. Their main advantage stems from their ability to provide high frame rates at relatively high resolution, require much less complicated electronic drive circuits, and consume less power, making them the obvious choice for hand-held and mobile 3D scanners (Hock 2014).

Another key factor in the improvement of 3D scanning technology is the increase in computational power. Scanners are getting faster and capturing larger amounts of data, meaning that there is more information to process and the bottleneck is often the computer. In the past three years hand-held scanners have increased in speed at an extremely high rate increasing from a few thousand points per second to hundreds of thousands of points per second; the ability to inspect small parts at a high speed is in a trend of continual improvement. While a lot of these advancements can be credited to increases in imaging technology, the majority of improvements come from the scanners using more complex structured laser or white light patterns that cover the entire field of view of the scanner instead of using a single laser trace. New light coding techniques were developed to enable spatial decoding of the pattern, which is the basis of single-frame full-field 3D imaging. Advances in real-time surface reconstruction and visualization as well as high-speed imaging reconstruction are also essential to the advances experienced in the last few years for portable and accurate hand-held 3D scanners (Hock 2014).

The two primary uses for 3D scanners at this time are in the R&D/product development industry and manufacturing as quality control/inspection systems. In product development 3D scanners allow R&D/product development teams to save considerable time when modeling existing objects in CAD software, reducing time and cost when designing, enhancing, and testing products. In the manufacturing industry it is essential to detect errors in production as early as possible to minimize costs and defective units. Using a 3D sensor that can ‘see’ what is happening and quickly implement quality control measures saves costs in the short-term and long-term (Hock 2014).

Previously in the 3D scanning market the primary issues were the speed and accuracy of scans. As technology advances manufacturers have more choices of where they want to position their scanners such as smart sensors, scanner mobility, and embedded modules. As the technology continues to move forward users will see more real-time captures of scan data in the form of video and interactive 3D models. The near future holds more advancement in portability, ease of use, accuracy, resolution, and speed, as well as an increase in integration between hardware and software (Hock 2014).

Similar Products (Hardware)

As a whole, the consumer 3D scanner marketplace can largely be divided into two main functional groups: desktop scanning devices and hand-held scanning devices. Desktop scanners have traditionally been the only feasible option for the average consumer, simply because most industrial 3D scanners are unaffordable and relatively large. Desktop scanning is typically much cheaper and more space efficient than using industrial scanners – but usually tends to produce less precise scans and can be limited by its small size. Hand-held scanning devices are a relatively new innovation in the consumer 3D imaging realm and they are quickly becoming the new standard. Although many desktop scanners can produce higher resolution scans than current hand-held scanners on the market, the hand-held market is still exploding. Hand-held scanners give the user the extra freedom to travel with ease and scan objects of virtually any size or shape. While desktop scanners are limited by their allotted scan area's maximum dimensions, hand-held devices often utilize different technology that constructs models using a point-and-shoot method – making the hand-held device the optimal choice for the average price-conscious, fairly active 3D hobbyist (Kasriel-Alexander 2014).

For functional purposes, two of the most popular hand-held three-dimensional scanning devices currently on the market will be examined and discussed in detail. The Structure Sensor builds 3D models from a scan using structured light technology, which is just one of the many principles of imaging utilized by current digitizers. Common short-range scanners employ laser triangulation, pattern fringe triangulation, or a combination of the two. Laser triangulation works by scanning across an object with a laser beam, which reflects light that is then retrieved by a sensor, and trigonometric triangulation algorithms are used to determine real-time spatial dimensions. Creaform, an industry leader in 3D measurement solutions has recently released the HandySCAN 3D, a portable digital dimensioning and modeling device designed less for the common consumer and more for budding industrial needs – mainly rapid prototyping and reverse engineering. The HandySCAN 3D is the fastest hand-held digitizer currently on the market and boasts an accuracy of up to 0.030 mm (0.0012 in.) and a resolution of up to 0.050 mm (0.002 in.), which is extremely high-definition compared to other hand-held scanners currently on the market. The HandySCAN 3D utilizes VXelements, an integrated 3D software platform that allows for a wide range of 3D scanning and measurement functions (Heffner 2007). Fuel3D is a new crowd sourced point-and-shoot hand-held 3D digitizer that was designed with the average hobbyist in mind. As far as hand-held 3D scanners are concerned, the Fuel3D is about as opposite of HandySCAN 3D as hand-held digitizers can be from one another. Fuel3D works by combining geometric and photometric stereo 3D reconstruction techniques to retrieve real world dimensional and color data from the object in question (Dunker & Luther 2014). In essence, Fuel 3D is an economic, less precise scanner designed for the average consumer hobbyist while HandySCAN 3D is geared towards industry professionals interested in streamlining their reverse engineering and rapid prototyping processes.

Software Background

Banking off of the hardware's enticing existing capabilities and their inherent untapped potential, Occipital has released an open sourced software development kit specifically designed to encourage development of innovative new applications for the Structure Sensor (Structure IO). Using the framework provided, the mobile package generating application will be programmed in Xcode using Objective-C++ to incorporate the Structure's three dimensional real-time modeling and dimensioning capabilities along with functional mathematical algorithms to calculate and deliver a 3D rendering and set of dimensions necessary for the final package. In theory, the user scans an object/objects and approves a 3D preview of the scan while the application calculates the length, width, and height of a package to contain the object(s).

Earlier on in the application development process, the potential hidden within the Structure Sensor's various functional features became very apparent. Curiously enough, the sensor's overarching potential seems to pose challenges in itself – if only because it opens up so many opportunities for creativity and innovation in design. During the first phases of ideation, the group proposed several concepts designed to make use of the Structure's key features in unique, cross-functional ways. For instance, the group originally had plans to include a function that would use the Structure's 3D scanning capabilities to locate problem geometry in a product and relay said information to design the package accordingly. Much like the automated defect detection technology currently used in the packaging industry today, our problem geometry locater would work by utilizing 3D solid structural analysis to recognize portions of an object's geometry that would compromise the overall structural strength of the product package system. This concept and a handful of other ideas were eventually found to be unfeasible with respect to this project's scope and were discarded fairly early on.

Developing a mobile application on a tight timeline and trying to finalize key functional components usually means that certain functional features have to be left behind in order to simplify the coding process and stay on track. With this in mind, several ideas that added significant value and functionality to the application itself were eventually abandoned as well. For quite some time, the group was seriously considering incorporating a feature that would retrieve important background information on the object itself. It was decided that after scanning the object, a pop-up window would ask to input key variables, such as an estimated weight, type of material, general level of fragility and other factors that could aid in automated package design. On top of streamlining the design process, the information collected in this step was intended for use as key inputs for ASTM and ISO testing protocols. The group also considered expanding on this idea by creating a large database of important ASTM and ISO test standards that the app could access to run standardized dynamics testing on the virtual box in real-time. Other databases would also be created to hold information on different flutings and their respective structural integrity. The sensor would use programmed virtual physics to determine the most dynamically efficient package and other algorithms to use the least amount of material possible. Coding for mobile applications that combine physical dimensional input and virtual physical testing in real time is undoubtedly a really interesting idea, but ultimately not feasible for the scope of this project.

The final design will focus on creating a simple, intuitive and overall aesthetically pleasing interface. The application will effectively generate custom dimensions for packages from data retrieved by scanning one or multiple objects using the Structure Sensor. The application's main functionality, generating simple dimensions for 3D scanned objects, uses the Structure Software Development Kit (Structure SDK) to program the application to retrieve vital information from the sensor that can then be processed to calculate the three required dimension values. Minimum bounding box algorithms are used to interpret data from the scan and find the minimum-top-area dimensions.

The application's ability to retrieve and process information from multiple objects and generate a singular consolidated package was questioned from the start. This feature posed several major design challenges that mainly centered around figuring out how to efficiently piece together completely independent geometries and retrieve information on one volume. Although it once looked as though it might be too difficult to program, this function proved valuable and solidified its place in the final design. Using algorithms to perform this function would have been too computationally expensive; the packing problem is notoriously difficult even when restricted to ideal spheres. In order to avoid the implementation difficulty and computational cost, the application relies on the user's knowledge of efficient packaging to combine multiple objects into a single package. The user arranges the multiple objects on the scanning table in the way they want them to be scanned, and the application simply treats them like a single object.

Similar Products (Software)

Although there are no other three-dimensional package-generating applications currently on the market for the Structure Sensor or any other hand-held 3D sensors, there are a handful of applications that operate using similar technology. In the packaging industry, Laetus technologies offers several different 3D scanners that are used in conjunction with packaging machines to read barcodes and determine package strength, among other things. Industrial 3D scanning software is used in the packaging industry for defect detection and inspection (Yoon). There are even automated object dimensioning systems used in industry that incorporate 3D scanning and function a lot like the application in development (Zhu). In recent years, further development of three-dimensional imaging technology has allowed for 3D scanning to even further solidify its utility in the packaging industry.

Since the Structure Sensor is so new to the market, there are only a handful of applications that have been developed for use with it thus far. However, there are a few mobile device applications that promise users precise 3D scanning functionality without the need for extra hardware. Autodesk's 123d catch is the most popular three-dimensional scanning application currently available and it relies solely on a mobile device's digital camera. Unlike the Structure's structured light imaging approach to created 3D scans, 123d catch works by retrieving a wide, panoramic array of two dimensional pictures and rebuilds the captured scene in three dimensions using projective geometry. This technology, photogrammetry, was originally used by industrial 3D scanners designed for large-scale scans because of its ability to scale (Sanders). While the 123d catch works surprisingly well for a mobile application that utilizes no supplemental hardware, it cannot even compare to the models generated using applications for the Structure in terms of precision and accuracy.

Occipital, along with other developers, have independently released several mobile applications designed to work with the Structure Sensor. Occipital, the company behind the Structure Sensor, has developed a handful of sample iOS applications to illustrate the diverse range of the Structure Sensor's functionally and the possibilities of the hardware itself. Occipital's "Room Capture" and "Scanner" allow users to capture three-dimensional information from the physical world and model it digitally. Other apps like "Fetch" and "Ball Physics" highlight the product's ability to digitally augment reality and the possibilities that this functionality presents. The three dimensional imaging industry leaders at 3D System's have also released a 3D scanning application that works with the Structure. The application itself is a redesign on an existing 3D Systems application that has been retrofitted to be marketed together with the Structure Sensor as a single, consolidated product under the name iSense. 3D System's scan application, unlike Occipital's "Object Scanner", supplements the standard structured light technology by incorporating the iPad's camera feature to effectively colorize 3D models using real-time color data. Another major industry name, ItSeez, has developed ItSeez3D, the most recent 3D scanner app to be released for the Structure thus far. ItSeez3D works by combining several different 3D scanning techniques, much like 3D System's iSense, which allows for a colorized and more dimensionally-accurate final deliverable. However, unlike the other scanning applications for Structure ItSeez3D doesn't actually render the model on the iPad. Rather, it sends data to the cloud where the data is processed and the model is completed in minutes. Using cloud computing to generate a final 3D model proves to work quite well, as the resolution achieved using ItSeez3D seems to trump all of the other applications (Hoffman).

Below, Table 1.3 illustrates different criterion from "An affordance-based methodology for package design" compared to the Needs discussed in earlier in the Introduction. Each criterion scored from one (1) to five (5), with five (5) indicating the criterion strongly met the Need.

Table II - How Well Needs Are Met

Customer Needs	**Importance (1-4)**	**Autodesk 123d Catch (1-4)**	**Handiscan 3D (1-4)**	**ItSeez3D (1-4)**
Intuitive interface	**3**	**2**	**3**	**2**
Accurate dimensions	**4**	**3**	**4**	**4**
Option to store dimensions	**2**	**2**	**2**	**2**
Ability to scan wide variety of objects	**3**	**2**	**4**	**3**
Operation of app is fast and smooth	**4**	**3**	**3**	**3**

Literature Review Conclusion

The global 3D scanner market is a growing industry, predicted to expand from $2.06 Billion in 2013 to $4.08 billion by 2013. Studies show the technology is improving at a fast rate with demand and possible applications expanding as the technology improves more. So far the costs for the scanners have not been decreasing substantially but manufacturers are offering a wider range of solutions at a wider range of price points to meet needs in diverse markets and to reach customers previously uninterested or unable to utilize 3D scanning technology. As technologies in cameras, controllers, and processing continue to miniaturize and improve their capabilities, more hand-held and mobile options for these scanners are being developed and marketed to consumers that previously were unable to utilize this technology because of size and mobility constraints (Hock 2014).

Realizing that 3D imaging and scanning technologies are now commonplace in a number of influential industries, it's difficult to understand that the three-dimensional imaging industry itself is still in its infancy - but it truly is. While a lot of the technologies utilized by consumer three-dimensional scanning hardware and software have been used in industry applications for some time now, most of this technology has only become accessible to the average consumer within the past couple of years. As this technology permeates further into the consumer marketplace, the sensor's functionalities will continue to improve in interesting cross-functional ways. With that in mind, the Structure Sensor seems like the most logical platform choice when considering development of a functional iOS application geared towards the average consumer or maker. Not only does the software developer kit offer a lot of flexibility in terms of programming pre-existing features but the hardware itself is extremely accessible, in terms of affordability and usability. Applications that code for virtual physics allow room for rapid dynamics testing among many other functions that can be used for a variety of purposes in industry. If market trends continue and innovative applications are constantly being developed, the future of packaging dynamics testing and rapid package prototyping might lie in the hands of 3D dimensioning and imaging technologies.

SECTION III

PROCEDURE/ METHODOLOGY/ SOLUTION

Alternatives/Solution (Preliminary)

The objective of this project is to develop an iOS application that utilizes Occipital's Structure Sensor and compatible iPad model to scan an object or group of objects arranged for distribution. The application will display an output of the necessary dimensions to construct a suitable package able to completely contain the dimensioned items. This application will streamline the dimensioning process in high-volume distribution centers by reducing the amount of time taken by workers to dimension objects and allow users to refer back to saved dimensions, another time and cost-saving aspect of the application.

Design Alternatives

In the design of this application the team had numerous ideas on how to improve the functionality and usability of it to benefit customers. Some of the solutions or alternative application designs that have been discussed include: the ability for the application to output dimensions in an ArtiosCAD file that would have the exact dimensions necessary to contain the desired objects. While in theory this would be an obvious capability for a package dimensioning application to have, in practice it is impractical for the team to develop file format conversion code that would be necessary for such a complicated addition to the app. While this addition to the app has been discarded for now is has not been entirely discounted and if the team discovers a manageable way to implement the technology it will be undertaken.

Design Procedure

The design process undertaken throughout this project utilizes a number of different iterations utilizing a variety of different potential solutions. Through trial and error the design is optimized with alterations in algorithms and processes implemented. Using the structure developer's forum, and taking advantage of Occipital's willingness to allow users to design their own applications for use with the sensor, it is a relatively simple process to begin development of the PackIt app.

The first iteration utilized the O'Rourke Minimum Volume Bounding Box (MVBB) algorithm from Geometric Tools Engine, using an OBJ library to load the points from a wavefront OBJ file and calculate the "minimum dimensions" of the object. While promising because of its ease of implementation there was no way to verify dimensions were accurate because it was tested on an OBJ file downloaded from the internet of unknown dimensions. This iteration was developed prior to obtaining the sensor and iPad needed to fully test and validate the design. While not leading to a breakthrough of any kind this iteration shows the relative ease at which algorithms can be implemented to work with the sensor and applications already on the market without causing the app to crash or behave unpredictably. A key takeaway is that even for a small model with less than 1000 data points plotted, the algorithm took a significant amount of time to calculate dimensions and hung the user interface (UI).

After receiving the iPad and bracket that made the scanner compatible with the version of iPad being utilized, the application was updated. This was the first iteration where the scanner was used in conjunction with the iPad and the PackIt app. The MVBB algorithm was integrated with the addition of hooks thrown into the code that ran the points from the scanned 3D model through the MVBB algorithm, displaying the resulting dimensions on top of the three-dimensional model of the scanned object(s). The calculations took an excessive amount of time to generate, and even once the app stopped crashing the UI still hung indefinitely when attempting to generate dimensions. In order to speed up the process, a fraction of the points used to generate the dimensions were eliminated. To ensure the dimensions were generated in a reasonable timeframe (around 15 seconds), 98% of points in the scan were discarded randomly. With models having an average of ~20,000 points, the app was now using ~400 points to calculate the dimensions. This decision was a trade-off between accuracy and runtime. An issue encountered here was that the app would hang (freeze) during the calculation period. To mitigate this the calculation was moved to the background queue, slowing the calculation time down a slight amount but allowing the user to move the object around for inspection while the dimensions were being generated.

This iteration frequently incorrectly or inaccurately scanned the desired object. Usually picking up data points not on the object, such as the floor or any nearby surface. This resulted in wildly inaccurate dimensions because the minimum bounding box algorithm determined these points to be a part of the object and included those outlying points as part of the actual object. To address this issue only the edges of the model were used to determine the connected components of the model. Any connected component that did not have at least 5% of the total points was discarded from the resulting point set. This algorithm added a significant amount of time to the calculations. It was also problematic because very large meshes (meshes with more than 65536 points) are split up into multiple meshes with no edges connecting separate meshes. This causes the algorithm to throw away medium-sized but important pieces of the model if the mesh is split up as stated.

After extensive testing the dimensions reported by this iteration were determined to be so inaccurate to be for all intents and purposes useless for users. The dimensions seemed to be on the correct scale relative to each other but did not have any common conversion factor to translate them into a useful and accurate set of dimensions. Even after consulting the Structure SDK forum, a solution could not be found. After further evaluation it was discovered that the dimensions found were not actually correctly scaled to each other. The MVBB algorithm was either incorrectly implemented or programmed from the beginning, and debugging it requires an extensive knowledge of the algorithm and its implementation in the Geometric Tools Engine, which is unavailable. In the interest of keeping the program as simple as possible until accurate dimensions can be found, the MVBB algorithm has been discarded.

The third iteration began with the reasonable assumption that the object was already oriented upright the way a user would like to package it. So the calculation of the height dimension became a trivial linear scan. To determine the orientation of the box and therefore the other two dimensions only one more defined angle is needed. This was solved by having the user manually rotate the scanned model, lining up one side perpendicularly to the defined vertical angle. The user is presented with a strictly top-down view of the scanned 3D model and allowed to rotate it using a swipe gesture. The idea is that the user rotates the model so it is aligned with the screen in the orientation desired for the final package. Once they have the model oriented they can press the "calculate dimensions" button to see the calculated dimensions along the three axes, which involves a simple linear scan of all the points. This iteration was rewritten using version 0.4 of the Structure SDK, which improved some of the problems experienced with scanning excessively dark or shiny objects.

Issues experienced with this iteration mainly deal with the accuracy being greatly affected by the ability of the user to manually determine the correct orientation of the scanned object. Another issue is that the app did not clarify and inform users exactly how the orientation process was to be performed, resulting in extremely inaccurate dimensions in a testing phase that had to be discarded because the testing procedure was incorrect and resulted in completely inaccurate and unusable data. Having the three-dimensional model display in a manner where users were unable to move it around to verify the scan correctly captured the object also added to the infeasibility of this iteration.

The fourth and final iteration of the application resulted from a conversation with computer science professor and advisor John Clements, where the app was demonstrated and the difficulties explained. Recommendations for improvement centered around attempting to implement a brute-force check on many possible angles and making the dimension calculation automatic again (i.e. requiring no user input) by just picking the angle which gives the box a minimal top face (and bottom face) area. This version operated the fastest of all the iterations even though it checks hundreds of angles (to within a quarter of a degree); it is significantly faster than the MVBB method. This version proves to be more accurate than prior iterations without needing to remove any points from the scanned model like the second iteration. In this context that would have actually slowed down the dimension generation and caused problems with large and highly detailed objects. To deal with the issue of the scanner picking up points off of the object directions are included for the user that scans should be restarted if points are picked up places other than the object. Unless they disappear quickly they will most likely stay there and greatly affect the accuracy of the dimensions generated.

Testing Procedure

In order to determine the accuracy of the output dimensions, series of scans are taken of objects repeatedly. By recording the dimensional output from these scans and comparing them to the true dimensions as measured with digital calipers, it is a simple matter to determine the standard deviation of dimension data from scan to scan as well as the percent error between the expected value and the actual value. Originally in the testing procedure for the second and third iteration of the app, the scanned objects were chosen to represent objects that the application would actually be utilized for, such as a metal stapler and oddly shaped plastic toys. Realizing that the accuracy needed to be confirmed before experimenting with a wide variety of objects, it was decided to concentrate on scanning wooden blocks of known, simple dimensions and compare the app output to these true dimensions to determine the accuracy and consistency of the program.

The initial testing procedure was undertaken when the dimension output was not scaled correctly and resulted in a wide variety of values being measured but could not be linked back to common measurements. Each object is first measured using digital calipers in the X, Y, and Z directions to determine the true value needed for a package to contain it, then each object is scanned with the application a total of three times. That data is recorded and the average values as well as the standard deviation between scans is calculated. This round of testing was ineffective as at this point the dimensional output of the app was still in an unknown metric so it was impossible to compare the dimensional output to the expected values without determining a conversion factor which was unsuccessfully attempted.

The final testing procedure centers more on establishing and determining the actual accuracy of the application now that the output is able to be switched between metric and imperial units. For this stage of testing there are three separate items or groupings of items that were scanned and dimensioned: a wooden block painted black, an unpainted wooden block, and finally the two blocks placed next to each other. Each configuration is scanned a total of 10 times, then the standard deviation and average of all the scans is taken. To determine the difference between the expected and actual values the percent error is calculated between the values measured with digital calipers and the average outputs from the application.

Instructions for Operation of Application

An essential part of developing this application is creating detailed instructions and troubleshooting advice for users to mitigate any issues that may arise during standard operation of the app. To facilitate this, instructions appear as a dialogue box when first opening the app. This dialogue box introduces the basic functions and operations as well as some common issues with scanning objects. To avoid unnecessary reminders of the operating instructions the dialogue box has an option to stop it from showing up each time the app is opened. Pictures and screenshots from actual app operation are used to clarify and show users the specific components of the app that will be dealt with during standard operating procedures. Screenshots of the actual dialogue boxes implemented in the app are included in the appendix of this report.

The first set of instructions a user sees when opening the app introduces the basics of the program, reminding users PackIt is an application designed to scan and object or group of objects and determine the minimum interior dimensions of a package capable of containing them. When scanning multiple objects for a single package they must be oriented in the same way the user wishes them to be packaged and shipped. PackIt is able to find dimensions but is unable to orient scanned items to the most favorable position. The recommended minimum distance is 0.75 meters from object to scanner, though this may be reduced if the item(s) to be scanned are particularly small, generally any item(s) smaller than 0.04m need to be scanned from a distance of 0.50m. The smaller the object the less reliable the scans tend to be as well. This will be covered more in the data analysis and accuracy portions of the report.

Using screenshots of the program in operation users are shown how to orient the scanning box that displays on the screen: two fingers are used to enlarge or minimize the size of the box in order to completely contain the object(s) to be scanned. Once the item(s) to be scanned are contained in the scanning box the blue button on the middle right is engaged and the scan begins. The app shows what parts of the item(s) have been scanned by plotting white points on it until the entire object is completely covered, meaning the objects are completely scanned. To ensure the objects are completely scanned users must hold the scanner level and at the same distance from the objects while circling around the scanned objects, ensuring all faces are scanned and points are plotted along the entire model. Once a scan is started users will notice that the distance is displayed in meters below the scanning cube. This is to facilitate keeping the scanner at a constant distance from the objects.

During the scanning process, some errors occurred frequently enough to warrant comments that assist users with ensuring they do not affect the functionality of the program. At times the scanning application will detect some of the surface the objects are situated on as a part of the objects to be scanned. Through trial and error it was found that by holding the scanner level for an extended period of time (usually around 10 seconds) the program will realize its mistake and correct it. If this does not work then users are advised to press the cancel scan button located directly under the scan button and restart the scan from the beginning. To ensure accurate dimensions it is essential no data points lie off of the scanned objects. When scanning more complicated objects or objects without much contrast to their background, the program will sometimes lose track of the model and show the warning “Please put the model back in view.” at the bottom of the screen. When this occurs the scanner must be moved back to the position it was at when tracking was lost. Once at the correct position users will see that the notification has changed to “Recovering, please move gently.” This means the app is regaining the model scan from the correct orientation. This error generally occurs when a user is moving quickly around the object and not allowing for a large enough set of data points to be plotted on the three-dimensional model.

Once the scan has been completed users see the three-dimensional model shown in the next screen covered in the white data points. In the middle of the screen are the three dimensions necessary to contain the object(s). The default option shows them in meters but this can be toggled to imperial inches with a switch at the top middle of the screen. There is also an option to email the dimensions and an OBJ file of the model using the button on the top right of the screen.

Conclusion

While the development of this application would be considered an original solution to the problem of streamlining the dimensioning process for distribution there are many aspects of the design that would be better categorized as incremental improvements to existing designs. The idea of an application that uses a tridimensional scanner to determine package dimensions for objects has not been developed yet, however all the technologies utilized in the application's operation have already been created. Each component must be implemented into this application and optimized for functioning within the app as well as with the other API's being utilized. By moving through multiple design iterations and constantly improving the functionality of the program the end result should be a useful and valued addition to any fast-paced distribution environment. To ensure smooth operation for users detailed instructions and troubleshooting advice has been created and embedded in the application itself.

SECTION IV

TESTING AND RESULTS

Experimental Plan

After drafting a working prototype version of the application, several objects are chosen for their respective color, intricacy, overall shape and material. Each object is manually dimensioned and its actual dimensions are recorded as control data. In a well-lit room, the objects are then placed on a beige table (with no work envelope) and scanned three times using a fourth generation iPad running our app and Occipital's Structure sensor, manually noting the dimensions for each scan. Each scan's recorded data is statistically judged against the control data to develop a clearer understanding of general spread, standard deviation and a consistent standardized multiplier.

The final iteration of the program is tested to maximize the analysis of consistency and accuracy as opposed to determining the program's capability to scan complicated objects. This is based on the fact that the previous testing has not resulted in useful data that can be utilized to enhance the functionality and accuracy of the app, two of the most important aspects of the design and testing processes. To facilitate this testing the objects chosen are simple blocks of wood painted different colors.

Sample Preparation and Test Procedure

The sample objects used to calculate the general accuracy and help the overall function of the application are chosen for their unique characteristics, which helps to better illustrate the application's shortcomings and perhaps even help expose possible methods to overcome these setbacks. The samples are chosen and dimensioned using digital calipers, and all measurements are double-checked to ensure their accuracy. From past experimentation with available applications, the Structure Sensor seems to have difficulty recognizing the true dimensions of simple objects, shiny objects and extremely dark objects. For this reason, a range of five different samples were chosen to better understand how the sensor will be able to recognize a wide array of shapes and materials in conjunction with the application. Each sample object is placed on a beige desk, in a well-lit room and scanned using the Structure Scanner in conjunction with a fourth generation iPad and the tridimensional packaging application.

The final testing procedure is centered more on establishing and determining the actual accuracy of the application now that the output is able to be switched between metric and imperial units. For this stage of testing there are three separate items or groupings of items that were scanned and dimensioned: a wooden block painted black, an unpainted wooden block, and finally the two blocks placed next to each other. Each configuration is scanned a total of 10 times, then the standard deviation and average of all the scans is calculated. To determine the difference between the expected and actual values the percent error is calculated between the values measured with digital calipers and the average outputs from the application.

Preliminary Test Results

Figure 1- KS Industries Coffee Mug

Table III- KS Industries Data

Test Object	Test Number	Height (m)	Length (m)	Width (m)
KS Industries Mug	Control	0.1905	0.13716	0.09525
KS Industries Mug	1	0.09345	0.05842	0.04167
KS Industries Mug	2	0.09344	0.06203	0.0353
KS Industries Mug	3	0.08935	0.05278	0.04336
Average:		0.09208	0.05774	0.04011
Standard Deviation:		0.00193	0.0038	0.00347

Figure 2- Green Pig Doll

Table IV- Green Pig Doll Data

Test Object	Test Number	Height (m)	Length (m)	Width (m)
Green Pig Doll	Control	0.142	0.086	0.083
Green Pig Doll	1	0.06539	0.04126	0.03734
Green Pig Doll	2	0.0627	0.0404	0.03889
Green Pig Doll	3	0.06351	0.04063	0.03599
Average:		0.06387	0.04076	0.03741
Standard Deviation:		0.00113	0.00036	0.00119

Figure 3- Grey Stapler

Table V- Grey Stapler Data

Test Object	Test Number	Height (m)	Length (m)	Width (m)
Grey Stapler	Control	0.21336	0.05398	0.0668
Grey Stapler	1	0.09839	0.02176	0.02813
Grey Stapler	2	0.08887	0.0259	0.01484
Grey Stapler	3	0.08993	0.02669	0.01705
Average:		0.0924	0.02478	0.02001
Standard Deviation:		0.00426	0.00216	0.00581

Figure 4- Rubik's Cube

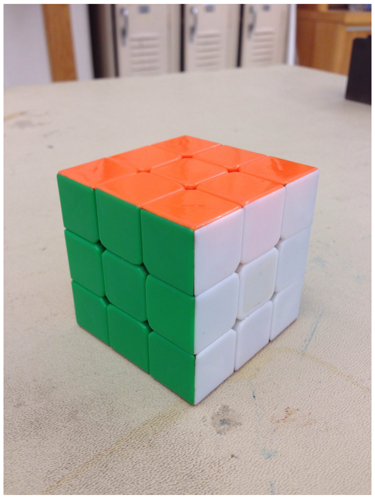


Table VI- Rubik's Cube Data

Test Object	Test Number	Height (m)	Length (m)	Width (m)
Rubik's Cube	Control	0.05715	0.05715	0.05715
Rubik's Cube	1	0.02921	0.02879	0.02373
Rubik's Cube	2	0.02531	0.0287	0.02919
Rubik's Cube	3	0.02985	0.02512	0.02943
Average:		0.02812	0.02754	0.02745
Standard Deviation:		0.00201	0.00171	0.00263

Figure 5- Steel Vise

Table VII- Steel Vise Data

Test Object	Test Number	Height (m)	Length (m)	Width (m)
Steel Vise	Control	0.18415	0.06731	0.0762
Steel Vise	1	0.09085	0.03155	0.02739
Steel Vise	2	0.09014	0.02822	0.0332
Steel Vise	3	0.09091	0.0282	0.03444
Average:		0.09063	0.02933	0.03168
Standard Deviation:		0.00035	0.00158	0.00307

Figure 6- Bison Mold

Table VIII- Bison Mold Data

Test Object	Test Number	Height (m)	Length (m)	Width (m)
Bison Mold	Control	0.14318	0.13589	0.03124
Bison Mold	1	0.05855	0.05548	0.01039
Bison Mold	2	0.05885	0.05607	0.00986
Bison Mold	3	0.05971	0.04847	0.03444
Average:		0.05904	0.05334	0.01823
Standard Deviation:		0.00049	0.00345	0.01146

Final Test Results

Table IX- Black Cube data (Meters & Inches)

Black Cube	Actual Dimensions		
	Length (m)	Width (m)	Height (m)
	0.086	**0.086**	**0.088**
	Observed Dimensions		
	Length (m)	Width (m)	Height (m)
1	0.111	0.111	0.084
2	0.11	0.11	0.081
3	0.107	0.109	0.08
4	0.113	0.111	0.08
5	0.106	0.106	0.081
6	0.113	0.114	0.08
7	0.113	0.111	0.081
8	0.109	0.11	0.081
9	0.107	0.112	0.081
10	0.115	0.117	0.081
Average	**0.1104**	**0.1111**	**0.081**
Standard Deviation	**0.002939388**	**0.002773085**	**0.001095445**
% Error	**28.37%**	**29.19%**	**7.95%**

Black Cube	Actual Dimensions		
	Length (in)	Width (in)	Height (in)
	3.3858286	**3.3858286**	**3.4645688**
	Observed Dimensions		
	Length (in)	Width (in)	Height (in)
1	4.3700811	4.3700811	3.3070884
2	4.330711	4.330711	3.1889781
3	4.2126007	4.2913409	3.149608
4	4.4488213	4.3700811	3.149608
5	4.1732306	4.1732306	3.1889781
6	4.4488213	4.4881914	3.149608
7	4.4488213	4.3700811	3.1889781
8	4.2913409	4.330711	3.1889781
9	4.2126007	4.4094512	3.1889781
10	4.5275615	4.6063017	3.1889781
Average	**4.34645904**	**4.37401811**	**3.1889781**
Standard Deviation	**0.115723987**	**0.109176631**	**0.043127784**
% Error	**28.37%**	**29.19%**	**7.95%**

Table X- Wood Block Data (Meters & Inches)

Wood Block	Actual Dimensions		
	Length (m)	Width (m)	Height (m)
	0.037	**0.084**	**0.082**
	Observed Dimensions		
	Length (m)	Width (m)	Height (m)
1	0.063	0.088	0.073
2	0.059	0.087	0.076
3	0.045	0.086	0.072
4	0.039	0.085	0.076
5	0.065	0.088	0.073
6	0.064	0.087	0.074
7	0.066	0.089	0.075
8	0.064	0.089	0.077
9	0.067	0.09	0.074
10	0.058	0.089	0.07
Average	**0.059**	**0.0878**	**0.074**
Standard Deviation	**0.009011104**	**0.001469694**	**0.002**
% Error	**59.46%**	**4.52%**	**9.76%**

Wood Block	Actual Dimensions		
	Length (in)	Width (in)	Height (in)
	1.4566937	**3.3070884**	**3.2283482**
	Observed Dimensions		
	Length (in)	Width (in)	Height (in)
1	2.4803163	3.4645688	2.8740173
2	2.3228359	3.4251987	2.9921276
3	1.7716545	3.3858286	2.8346472
4	1.5354339	3.3464585	2.9921276
5	2.5590565	3.4645688	2.8740173
6	2.5196864	3.4251987	2.9133874
7	2.5984266	3.5039389	2.9527575
8	2.5196864	3.5039389	3.0314977
9	2.6377967	3.543309	2.9133874
10	2.2834658	3.5039389	2.755907
Average	**2.3228359**	**3.45669478**	**2.9133874**
Standard Deviation	**0.354768076**	**0.057861994**	**0.0787402**
% Error	**59.46%**	**4.52%**	**9.76%**

Table XI- Combination Data (Meters & Inches)

Combo	Actual Dimensions		
	Length (m)	Width (m)	Height (m)
	0.123	**0.086**	**0.088**
	Observed Dimensions		
	Length (m)	Width (m)	Height (m)
1	0.139	0.118	0.078
2	0.138	0.12	0.078
3	0.138	0.119	0.079
4	0.139	0.118	0.078
5	0.136	0.113	0.077
6	0.138	0.114	0.077
7	0.139	0.118	0.075
8	0.142	0.12	0.079
9	0.137	0.116	0.078
10	0.137	0.117	0.078
Average	**0.1383**	**0.1173**	**0.0777**
Standard Deviation	**0.001552417**	**0.002238303**	**0.0011**
% Error	**12.44%**	**36.40%**	**11.70%**

Combo	Actual Dimensions		
	Length (in)	Width (in)	Height (in)
	4.8425223	**3.3858286**	**3.4645688**
	Observed Dimensions		
	Length (in)	Width (in)	Height (in)
	5.4724439	4.6456718	3.0708678
	5.4330738	4.724412	3.0708678
	5.4330738	4.6850419	3.1102379
	5.4724439	4.6456718	3.0708678
	5.3543336	4.4488213	3.0314977
	5.4330738	4.4881914	3.0314977
	5.4724439	4.6456718	2.9527575
	5.5905542	4.724412	3.1102379
	5.3937037	4.5669316	3.0708678
	5.3937037	4.6063017	3.0708678
Average	**5.44488483**	**4.61811273**	**3.05905677**
Standard Deviation	**0.061118831**	**0.08812221**	**0.04330711**
% Error	**12.44%**	**36.40%**	**11.70%**

Problems With Test Results

As expected, the first working prototype application undoubtedly produced some problematic results, which appear to fluctuate depending on several different factors. One main problem with the data retrieved through the application is its inability to scale correctly, proving to be off by different factors that are somewhat dependent on the sample's shape, material and size. In many cases, the standard deviation of the multiplier effect could be used to accurately dimension the sample item - this just needs to be developed within the framework of the application itself. Awkward, protruding segments of different items also serve somewhat of a problem for the sensor, as it has trouble correctly mapping out their relationship in respect to the rest of the object's body. This was very apparent when scanning the KS Industries Mug, in which the curved, black handle proved difficult for the scanner to properly recognize. Geometric simplicity and glare from the object's finish also seemed to throw off the final results, as illustrated by the larger standard deviations in the Grey Stapler scan and the Rubik's Cube scan.

The final test results obtained with the final iteration of the program did not turn out as expected. There was considerable variation from the expected to actual values found. The percent error ranged from 5% to near 60% in some cases. While an error of 5% could be acceptable in some circumstances that is the maximum limit for error if this application is going to provide any value to users. An error value approaching 60% is too high to even consider the dimensional output to be of any use in any context. The standard deviation between the output dimensions was not significant but regardless of how precise the dimensional output is, it is the accuracy that creates value for the end user and will make this application a functional and useful addition to the distribution environment.

Results and Discussion

Overall the initial dimensional output by the prototype application did not align with any known or identified logical dimensions. Though the structure sensor is supposed to output its data in meters when the scanned items were measured with calipers and rulers the values did not match up with the data recorded. The team did notice that the measurements seemed to have the correct scale, in that if the measured height was twice the measured length the dimensional output by the scanner also had one value that was twice one of the other values. It was theorized that the team would need to implement some multiplier or equation to reach a workable dimensional output. This proved to be unsuccessful in practice.

A major issue with the scanner that will need to be solved is the trouble it has locking onto and scanning items that are reflective. It is theorized that the reflective surface distorts the structured light pattern that the sensor uses to gather data points. If the reflection of the structured light is altered by the reflective nature of the object it seems logical that it would affect the accuracy of the scan.

As for the final dimensional output and the large variation between the expected and actual values collected, it is theorized that this is a result of the angle determination programming used to collect the dimensional data from the scanned model. For the scanned models of the black wood block and the combination objects, the height value had the least variation, and for the unpainted wood block the width had the lowest variation between the expected and actual values. It seems that because the vertical angle is determined first perpendicular to the scanning surface that it would be the most accurate and less likely to experience variation. The relatively large discrepancy in the vertical dimension may be caused by the placement of the scanning cube. The scanning cube hovers slightly above the calculated position of the scanning surface, so as to avoid picking up points on the flat surface. Points on the object below the bottom of the scanning cube are discarded. Because some of these points may be actual model points, the accuracy of the vertical dimension suffers.

SECTION V

CONCLUSION

Summary

In order to develop a tridimensional measuring application for use with the iPad, utilizing a Structure Sensor, it was necessary to develop a clear understanding of what must be accomplished and the process of performing the necessary tasks. A preliminary literature review was conducted, focusing on the technology being utilized in this project as well as related technologies. With a wide variety of tridimensional scanners on the market for various uses and utilizing a wide variety of technology this review had a broad base of knowledge to draw from. Research included information about software as well as hardware involved in this and similar processes. Doing this provided a solid base of knowledge to form a comprehensive outline of what should be accomplished over the duration of the process. Through an iterative process of design and redesign, an application capable of scanning an object or group of objects in three dimensions and outputting three values representing the length, width, and height of a package necessary to contain the model has been created. Multiple different algorithms and implementation of a variety of methods have been utilized in an attempt to maximize the accuracy and functionality of the application. Using the Structure SDK, Sensor, and a 4th generation iPad the program was tested throughout the process to determine its accuracy and functionality in a fast-paced distribution environment.

Initially the O'Rourke Minimum Value Bounding Box Algorithm (MVBB) pulled from Geometric Tools Engine was used to determine dimensions of scanned three-dimensional models, though in theory the perfect algorithm for the needs of the project, being that its whole purpose is to find the minimum size box able to contain an object, the algorithm proved to be unfeasible due to an inordinately excessive calculation time between scanning a model and outputting dimensions. In an attempt to mitigate this issue, fewer data points from the scan were used to calculate the dimensions, while this stopped the application from crashing it did not speed up calculation to an acceptable duration. A key issue with the MVBB and one that led to discarding the algorithm entirely was its inability to output dimensions in a logical metric. This was determined through scanning a variety of different items repeatedly to analyze the results and determine the accuracy and reliability of the application. While the values seemed correctly scaled relative to each other, a conversion factor to any common dimension could not be found, rendering it useless for the needs of this application. This testing phase was not completely useless as it did result in minor interface changes to increase the functionality of the app as well as the realization that comprehensive directions and troubleshooting guides must be developed for users to be able to utilize the app to its full potential once complete.

The final iteration of the application relies on a brute-force check of all possible angle configurations of the model to be scanned, assuming that the vertical axis is perpendicular to the surface the model is situated on. By establishing the constant angle the program is then able to determine the angle that gives the model the minimum possible area for the top and bottom face, and therefore the minimum size of a package to contain the model. This alteration resulted in exponentially faster computation time for the dimensions, and also it output dimensions in metric as well as imperial units. It was during this stage that comprehensive instructions and troubleshooting guides were implemented into the application to ensure ease of use by end users. This was accomplished by creating dialogue boxes that appear when opening the application explaining how to operate the program and providing troubleshooting advice for common issues experienced. Once the brute-force analysis was implemented, testing began. Instead of scanning relatively complex shapes, as in the first round of testing, this testing phase purposely scanned simple cubic shapes with simply defined sizes in order to determine the accuracy of the application. Input from professors in the Computer Science and Industrial Technology departments was taken into consideration and implemented when applicable and a wide variety of alternate programming solutions were utilized to attempt to mitigate the apparent weaknesses in the application.

Issues

Though it is capable of outputting dimensions in imperial and metric units, the variation between the actual values of the models (as measured with digital calipers) and the dimensional output from the application itself is too great for the application to provide any value to users at this point. The difference between actual dimensional values and those output from the scanning application range from an error of just under 5% to almost 60%. While 5% error in dimensions would be acceptable, it is nearing the limit of the acceptable percent error and that is the lowest percentage in any scan. 60% error is completely unacceptable for this application and makes it completely unfeasible for use in a fast-paced distribution environment. While the true reason behind this high degree of error is unknown it is theorized to stem from the implementation of the angle determination program and possibly how it interacts with the built-in programming for the three-dimensional scanner and iPad. The variation between scanned values was negligible, showing that the application was outputting precise results as their standard deviation was all under 0.35 and that was an outlier, with the vast majority of scans resulting in a standard deviation of <0.12, an acceptable value for variation between scans.

While this primary issue has for now rendered the application useless for the defined purpose, if it is solved there are still other issues that will need to be dealt with before the application can be brought to market. Since the beginning there have been major issues with scanning objects with extreme curvature or reflective properties. One way to mitigate these errors is to choose a scanning surface that contrasts well with the model to be scanned. This seems to enable the scanner to track the model with increased accuracy compared to a background of similar color to the model. This issue seems to be reduced in severity with every update from Structure for the Sensor which points to it being an issue with the built in software and not an issue that will be solved by any alterations on the part of the development process for PackIt. To mitigate this issue is a process of trial-and-error to determine which background surface fares best with each specific object, in general the higher level of contrast between the background and the object(s) to be scanned optimizes the process.

Options for Continuation

Even though at this point the application is not functional for the purpose it was designed for, there is still the option to continue work on it through next year. This is dependant on the cooperation of all team members and technical advisors agreeing to allow the project to continue. Though the senior project portion of the project must end, all members of the team at this point are willing and able to continue work on the project during Fall quarter of 2015 because all members will still be attending Cal Poly. Continuation would include analysis of the reason behind the output data lying so far from the true values of the dimensions and where the programming error lies. It could also include the implementation of another algorithm or alternate coding to correct the issue.

Lessons Learned

Over the course of this project in the iterative process of developing and testing different versions of the application a great deal has been learned about the importance of research and exploring alternate approaches to the same issue. By establishing a strong base of knowledge about the technology used in the sensor and similar technologies and their applications it enabled a much smoother transition and defined process to follow throughout the project. The importance of statistical analysis was also emphasized, showing precise measurements of the variation experienced between expected and actual measurements taken by the scanning application. Before this project the programming process was completely unfamiliar to the Industrial Technology students involved. Over the course of the various iterations of the application they gained a more complete understanding of just how much work goes into any application no matter how simple it may seem on the surface. Problems are rarely, if ever, solved with the first attempted solution, often requiring multiple iterations and possible solutions to be attempted and tested to verify or discount their success. More so than other group projects participated in, this project, lasting almost a full school year, shows just how important it is to have full engagement from all team members to ensure success. Another key element is the advice of technical advisors on what is expected for the project and their strategies and recommendations are for how to accomplish those goals. Without the input of those with much more experience than the team members it would have been next to impossible to narrow down the scope of the project and accomplish as much in the same amount of time.

Table XII- Skills Utilized from Curriculum

Senior Project CapStone Experience	**Use:**			**Skills Utilized**
Industrial Technology Courses:	**Slight**	**Moderate**	**Heavy**	
IT 326- Product Evaluation			■	Product development process
IT 330- Fundamentals of Packaging	■			General knowledge of packaging
IT 403- Quality Systems Management			■	Percent error, standard deviation
IT 407- Applied Business Operations			■	Time/team management, product development
IT 303- Lean Six Sigma Green Belt		■		Continuous improvement
IT 408- Paper and Paperboard Packaging	■			Expanded knowledge of packaging
Support Classes:				
STAT 251- Statistical Inference/Management I		■		General knowledge of statistical tools
STAT 252- Statistical Inference/Management II		■		General knowledge of statistical tools

SECTION VI

REFERENCES

Besseris, George. "Improving Quality Cost Performance with Qualimetrical Methods: A Case from a High-speed Packaging Process."The International Journal of Advanced Manufacturing Technology, 69.5 (2013): 1775-1789.

Client: Panoramic. (2012). In *Miniature Case Studies*. Atlanta: 3D Scan Company. http://www.3dscanco.com/products/software/reverse-engineering/

Dunker, Thomas, and Sebastian Luther. "Calibration of an Infrared 3d Scanner." *Tm – Technisches Messen*, 81.1 (2014): 8-15.

Heffner, S. (Ed.). (2007). Virtual Reality in U.S. Healthcare (Markets for Remote Surgery, 3D Modeling, Pain Distraction and Other Applications). *Kalorama Information Market Intelligence Report*. Retrieved from http://academic.marketresearch.com.ezproxy.lib.calpoly.edu/product/display.asp?productid=1514009&curl=&surl=/search/results.asp?prid=897840883&query=3d+model&cmdgo=Go&prid=897840883

Hock, Lindsay. "Scanning Products into 3-D." *Research & Development*. Research & Development Magazine, 6 Aug. 2014. Web. 17 Jan. 2015. <http://www.rdmag.com/articles/2014/08/scanning-products-3-d>.

Hoffman, T. (2014, August 14). Occipital Structure Sensor. *PC Mag*.

Kasriel-Alexander, D. (2014). Digital Life and Consumers. *Consumer Trends - Analysis*. Retrieved November 9, 2014, from http://www.portal.euromonitor.com/portal/analysis/tab

Sanders, L. (2011). *Developing New Products and Services : Learning, Differentiation, and Innovation*. Retrieved November 10, 2014, from http://site.ebrary.com/lib/calpolyslo/docDetail.action?docID=10504208&p00=3d scanner mobile application

Unver, E., Paul, A., & Dave, T. (2013). Applying 3D Scanning and Modeling in Transport Design Education. *Computer-Aided Design and Applications, 4*(1-4), 41-48.

Yoon, J., Lee, J., & Kim, B. (2011). An Automated Detection Method of Solder Joint Defects Using 3D Computed Tomography for IC Package Inspection. *Software and Network Engineering (SSNE), 2011 First ACIS International Symposium on*. Retrieved November 3, 2014, from http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6150069&searchWithin=3D CT Scan&queryText=Electronics Packaging Manufacturing

Zhu, Xiaoxun (2006). U.S. Patent No. 7527205 B2 Washington, DC: U.S. Patent and Trademark Office.

3D Sensor for Mobile Device. (n.d.). Retrieved November 11, 2014, from http://structure.io

APPENDIX

A. Statistical Analysis Data

Black Cube	Actual Dimensions		
	Length (m)	Width (m)	Height (m)
	0.086	**0.086**	**0.088**
	Observed Dimensions		
	Length (m)	Width (m)	Height (m)
1	0.111	0.111	0.084
2	0.11	0.11	0.081
3	0.107	0.109	0.08
4	0.113	0.111	0.08
5	0.106	0.106	0.081
6	0.113	0.114	0.08
7	0.113	0.111	0.081
8	0.109	0.11	0.081
9	0.107	0.112	0.081
10	0.115	0.117	0.081
Average	**0.1104**	**0.1111**	**0.081**
Standard Deviation	**0.002939388**	**0.002773085**	**0.001095445**
% Error	**28.37%**	**29.19%**	**7.95%**

Black Cube	Actual Dimensions		
	Length (in)	Width (in)	Height (in)
	3.3858286	**3.3858286**	**3.4645688**
	Observed Dimensions		
	Length (in)	Width (in)	Height (in)
1	4.3700811	4.3700811	3.3070884
2	4.330711	4.330711	3.1889781
3	4.2126007	4.2913409	3.149608
4	4.4488213	4.3700811	3.149608
5	4.1732306	4.1732306	3.1889781
6	4.4488213	4.4881914	3.149608
7	4.4488213	4.3700811	3.1889781
8	4.2913409	4.330711	3.1889781
9	4.2126007	4.4094512	3.1889781
10	4.5275615	4.6063017	3.1889781
Average	**4.34645904**	**4.37401811**	**3.1889781**
Standard Deviation	**0.115723987**	**0.109176631**	**0.043127784**
% Error	**28.37%**	**29.19%**	**7.95%**

Wood Block	Actual Dimensions		
	Length (m)	Width (m)	Height (m)
	0.037	**0.084**	**0.082**
	Observed Dimensions		
	Length (m)	Width (m)	Height (m)
1	0.063	0.088	0.073
2	0.059	0.087	0.076
3	0.045	0.086	0.072
4	0.039	0.085	0.076
5	0.065	0.088	0.073
6	0.064	0.087	0.074
7	0.066	0.089	0.075
8	0.064	0.089	0.077
9	0.067	0.09	0.074
10	0.058	0.089	0.07
Average	**0.059**	**0.0878**	**0.074**
Standard Deviation	**0.009011104**	**0.001469694**	**0.002**
% Error	**59.46%**	**4.52%**	**9.76%**

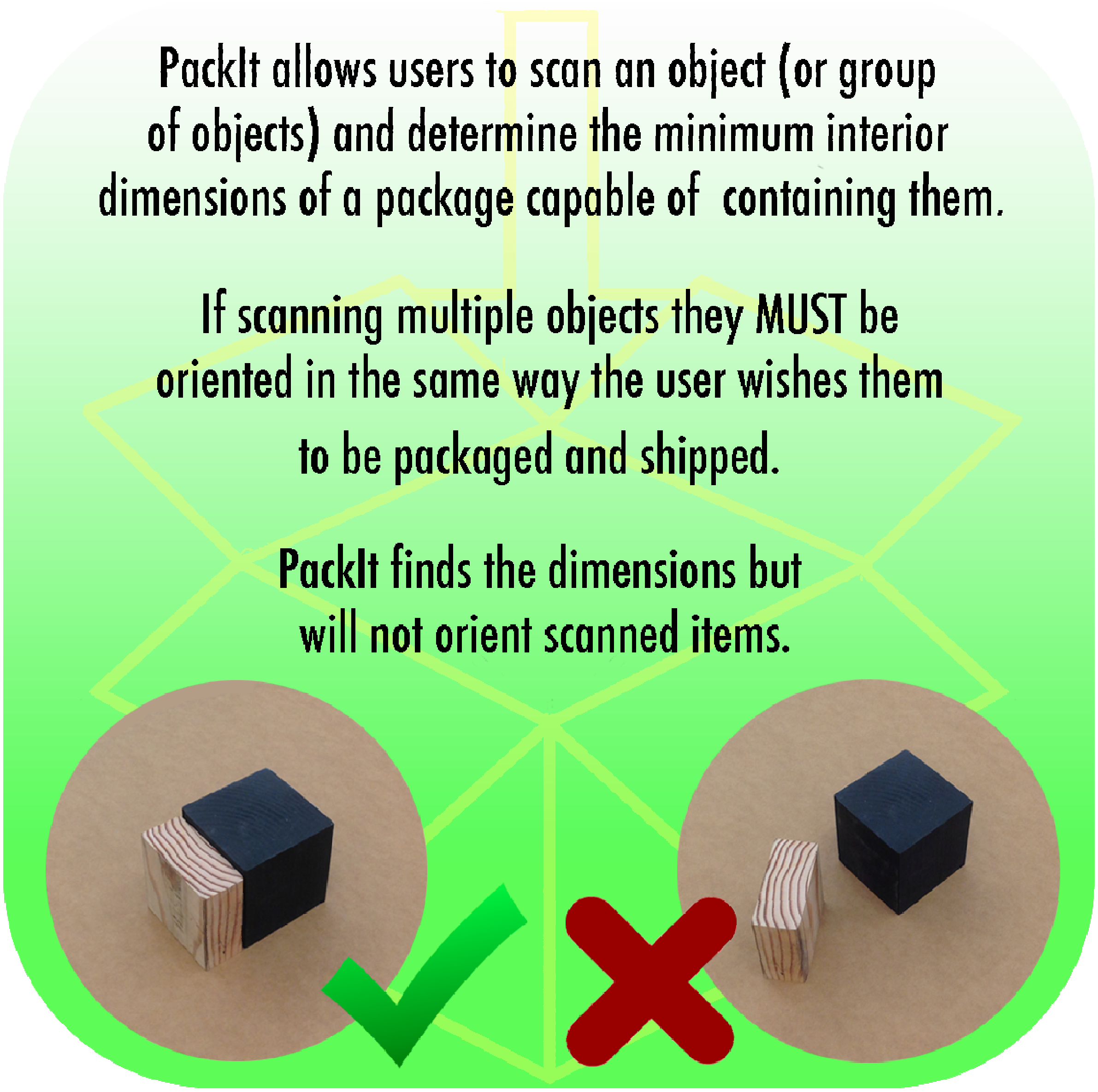
Wood Block	Actual Dimensions		
	Length (in)	Width (in)	Height (in)
	1.4566937	**3.3070884**	**3.2283482**
	Observed Dimensions		
	Length (in)	Width (in)	Height (in)
1	2.4803163	3.4645688	2.8740173
2	2.3228359	3.4251987	2.9921276
3	1.7716545	3.3858286	2.8346472
4	1.5354339	3.3464585	2.9921276
5	2.5590565	3.4645688	2.8740173
6	2.5196864	3.4251987	2.9133874
7	2.5984266	3.5039389	2.9527575
8	2.5196864	3.5039389	3.0314977
9	2.6377967	3.543309	2.9133874
10	2.2834658	3.5039389	2.755907
Average	**2.3228359**	**3.45669478**	**2.9133874**
Standard Deviation	**0.354768076**	**0.057861994**	**0.0787402**
% Error	**59.46%**	**4.52%**	**9.76%**

Combo	Actual Dimensions		
	Length (m)	Width (m)	Height (m)
	0.123	**0.086**	**0.088**
	Observed Dimensions		
	Length (m)	Width (m)	Height (m)
1	0.139	0.118	0.078
2	0.138	0.12	0.078
3	0.138	0.119	0.079
4	0.139	0.118	0.078
5	0.136	0.113	0.077
6	0.138	0.114	0.077
7	0.139	0.118	0.075
8	0.142	0.12	0.079
9	0.137	0.116	0.078
10	0.137	0.117	0.078
Average	**0.1383**	**0.1173**	**0.0777**
Standard Deviation	**0.001552417**	**0.002238303**	**0.0011**
% Error	**12.44%**	**36.40%**	**11.70%**

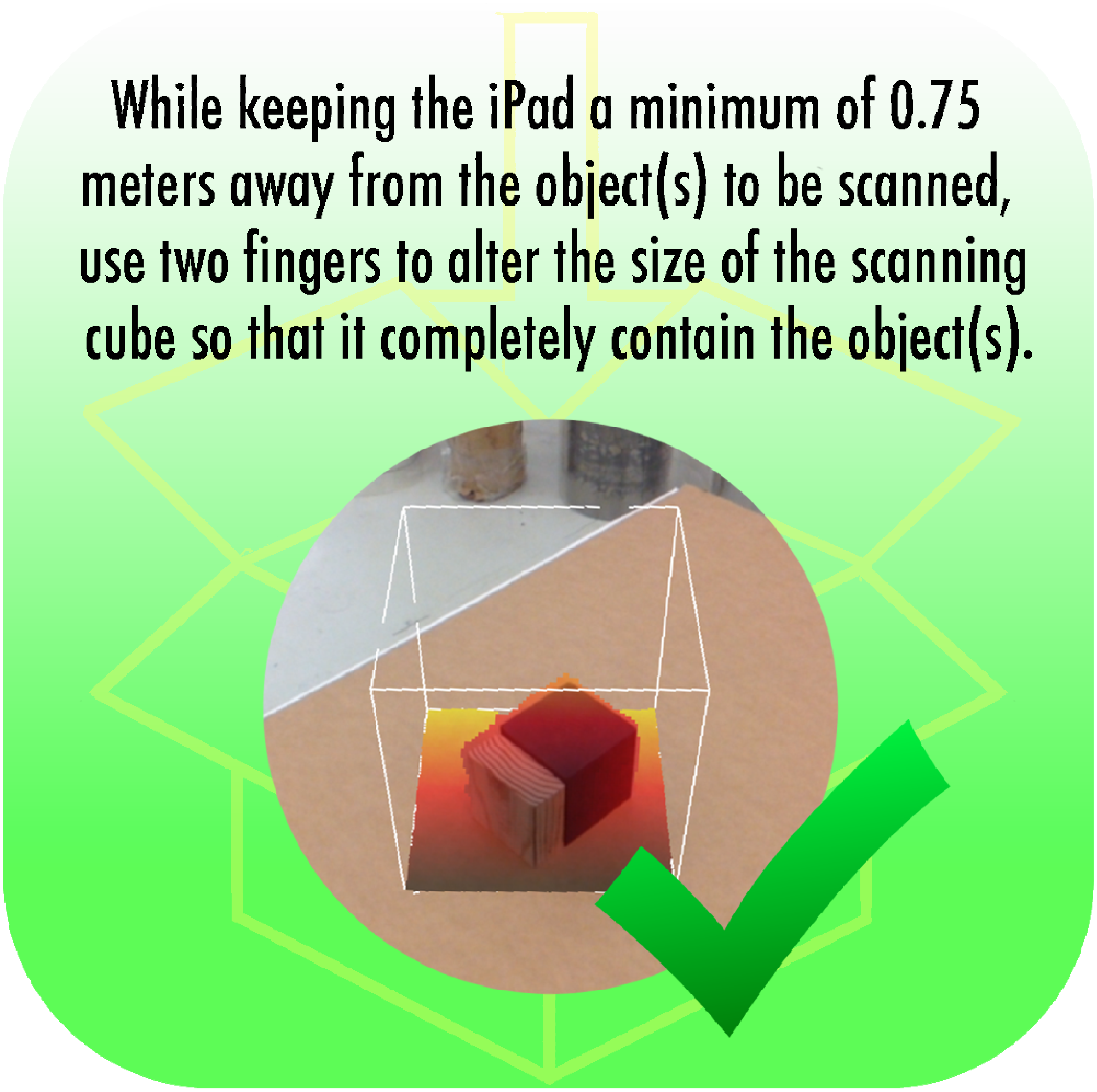
Combo	Actual Dimensions		
	Length (in)	Width (in)	Height (in)
	4.8425223	**3.3858286**	**3.4645688**
	Observed Dimensions		
	Length (in)	Width (in)	Height (in)
	5.4724439	4.6456718	3.0708678
	5.4330738	4.724412	3.0708678
	5.4330738	4.6850419	3.1102379
	5.4724439	4.6456718	3.0708678
	5.3543336	4.4488213	3.0314977
	5.4330738	4.4881914	3.0314977
	5.4724439	4.6456718	2.9527575
	5.5905542	4.724412	3.1102379
	5.3937037	4.5669316	3.0708678
	5.3937037	4.6063017	3.0708678
Average	**5.44488483**	**4.61811273**	**3.05905677**
Standard Deviation	**0.061118831**	**0.08812221**	**0.04330711**
% Error	**12.44%**	**36.40%**	**11.70%**

B. Instructional Dialogue Boxes:

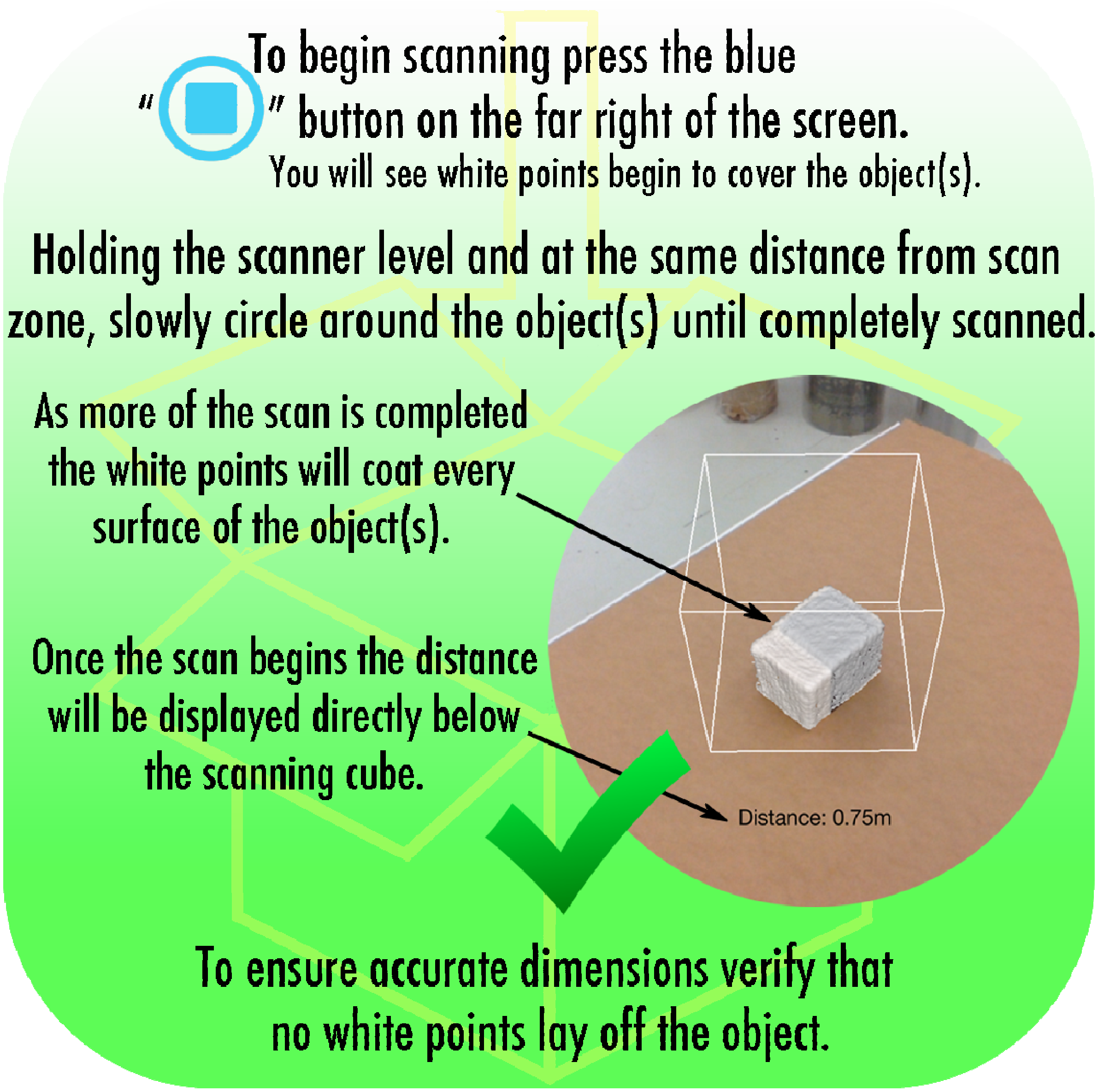
a. Dialogue Box 1:



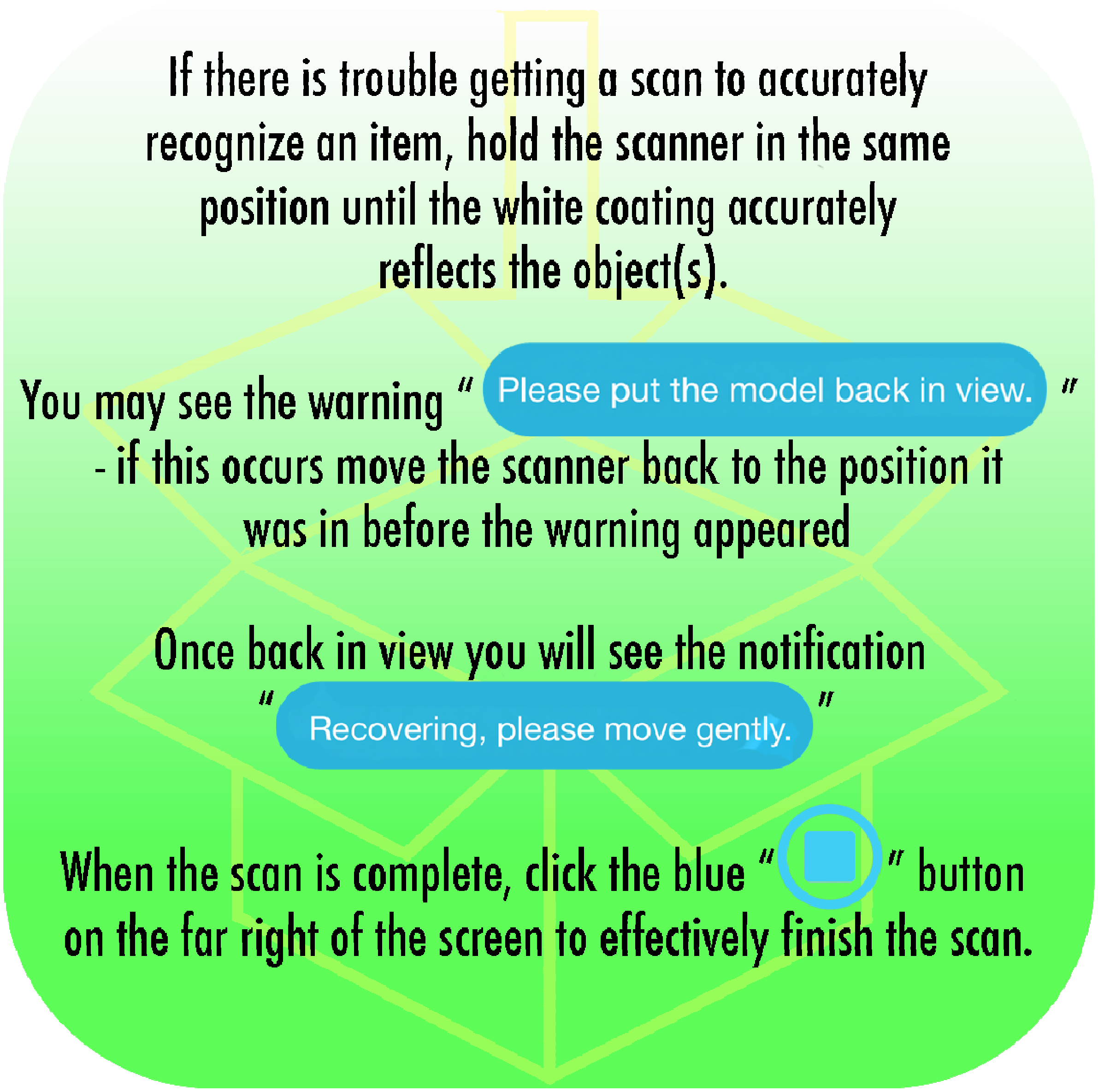
b. Dialogue Box 2:



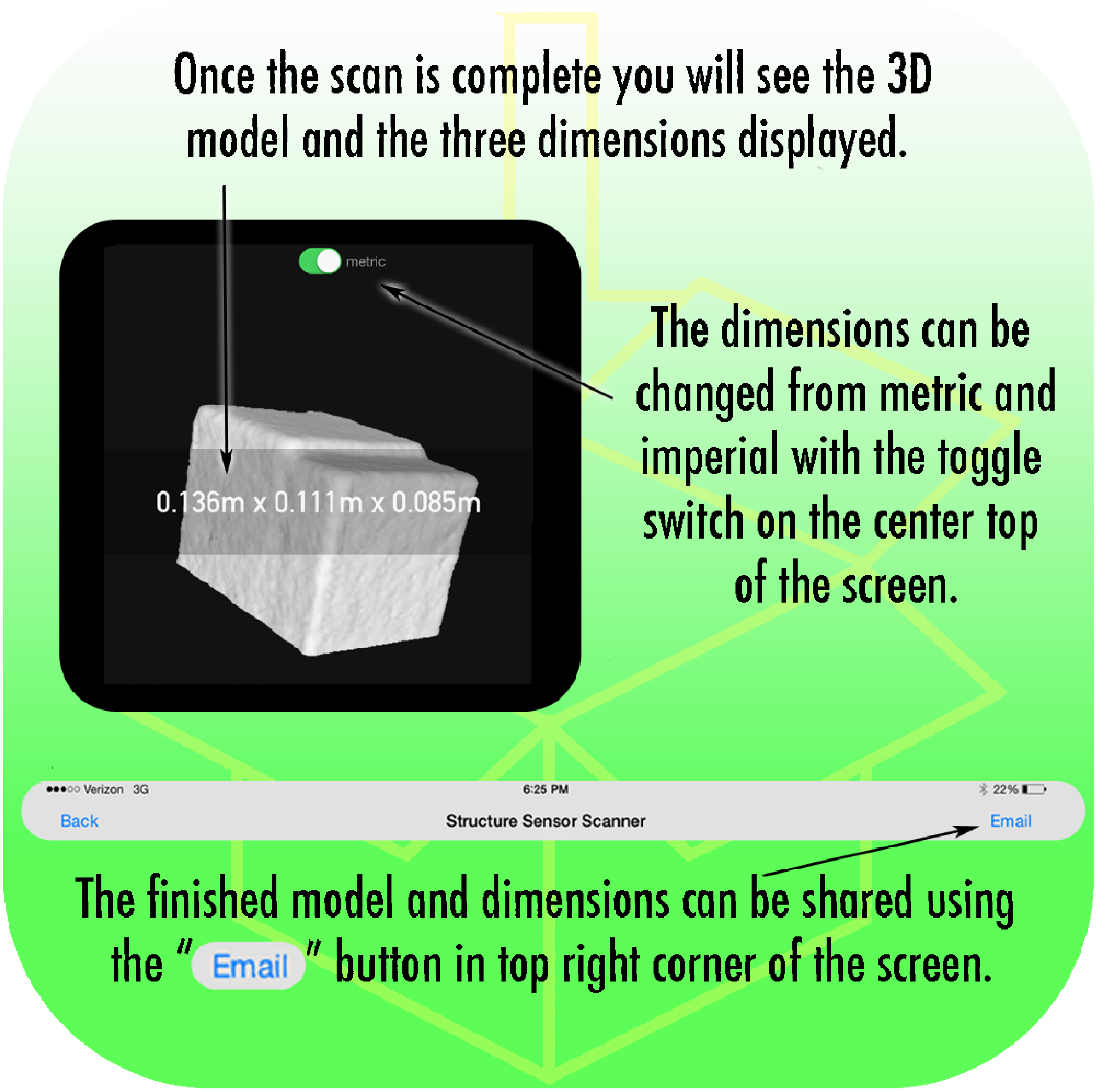
c. Dialogue Box 3:



d. Dialogue Box 4:



e. Dialogue Box 5:



f. Application Icon: