

6-9-2016

# Smart Photos

Daniel Hernandez  
*Santa Clara University*

Angela Laar  
*Santa Clara University*

Jaelin McCreary  
*Santa Clara University*

Follow this and additional works at: [https://scholarcommons.scu.edu/cseng\\_senior](https://scholarcommons.scu.edu/cseng_senior)



Part of the [Computer Engineering Commons](#)

---

## Recommended Citation

Hernandez, Daniel; Laar, Angela; and McCreary, Jaelin, "Smart Photos" (2016). *Computer Engineering Senior Theses*. 74.  
[https://scholarcommons.scu.edu/cseng\\_senior/74](https://scholarcommons.scu.edu/cseng_senior/74)

This Thesis is brought to you for free and open access by the Engineering Senior Theses at Scholar Commons. It has been accepted for inclusion in Computer Engineering Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact [rscroggin@scu.edu](mailto:rscroggin@scu.edu).

**SANTA CLARA UNIVERSITY**  
**DEPARTMENT OF COMPUTER ENGINEERING**

Date: June 9, 2016

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

**Daniel Hernandez**  
**Angela Laar**  
**Jaelin McCreary**

ENTITLED

**Smart Photos**

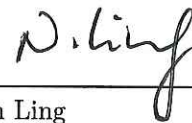
BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING



---

Ahmed Amer



---

Nam Ling

# Smart Photos

by

Daniel Hernandez  
Angela Laar  
Jaelin McCreary

Submitted in partial fulfillment of the requirements  
for the degree of  
Bachelor of Science in Computer Science and Engineering  
School of Engineering  
Santa Clara University

Santa Clara, California  
June 9, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Motivation . . . . .	4
1.2	Contemporary Solutions . . . . .	4
1.3	Our Solution . . . . .	5
<b>2</b>	<b>Requirements</b>	<b>6</b>
2.1	Functional Requirements . . . . .	6
2.2	Non-Functional Requirements . . . . .	6
<b>3</b>	<b>Use Cases</b>	<b>7</b>
3.1	Use Case #1 . . . . .	7
3.2	Use Case #2 . . . . .	8
3.3	Use Case #3 . . . . .	9
3.4	Use Case #4 . . . . .	9
<b>4</b>	<b>Activity Diagram</b>	<b>10</b>
<b>5</b>	<b>Conceptual Model</b>	<b>11</b>
<b>6</b>	<b>Technologies Used</b>	<b>12</b>
6.1	Development Tools . . . . .	12
6.2	Computer Vision Tools . . . . .	12
6.3	Machine Learning Tools . . . . .	12
<b>7</b>	<b>Architectural Design</b>	<b>13</b>
7.1	Overview . . . . .	13

7.2	Architecture Diagram . . . . .	13
<b>8</b>	<b>Design Rationale</b>	<b>14</b>
<b>9</b>	<b>Testing</b>	<b>15</b>
<b>10</b>	<b>Risk Analysis</b>	<b>17</b>
10.1	Overview . . . . .	17
10.2	Risk Table . . . . .	17
10.3	Risk Descriptions . . . . .	18
<b>11</b>	<b>Development Timeline</b>	<b>19</b>
11.1	General Overview . . . . .	19
11.2	Development Details . . . . .	19
11.2.1	Documentation . . . . .	19
11.2.2	Design . . . . .	20
11.2.3	Implementation . . . . .	20
11.2.4	Testing . . . . .	20
<b>12</b>	<b>Conclusion</b>	<b>21</b>
12.1	Augmented Reality Today . . . . .	21
12.2	Future of Smart Photos . . . . .	21
12.3	Last Words . . . . .	21
<b>13</b>	<b>References</b>	<b>22</b>

## List of Figures

1	Use Case Diagram . . . . .	7
2	Activity Diagram for the system . . . . .	10
3	Home Screen Concept . . . . .	11
4	Smart Photo with Label Concept . . . . .	11
5	Architectural Overview of Smart Photos . . . . .	13
6	3D capture of a standing individual . . . . .	15
7	3D capture of Santa Clara University book statue . . . . .	16
8	Development Timeline . . . . .	19

# 1 Introduction

## 1.1 Motivation

Recent technological leaps have been a great catalyst for changing how people interact with the world around us. Specifically, the field of Augmented Reality has led to many software and hardware advances that have formed a digital intermediary between humans and their environment. As of now, Augmented Reality is available to the select few with the means of obtaining Google Glass, Oculus Rifts, and other relatively expensive platforms. Be that as it may, the tech industry's current goal has been integration of this technology into the public's smartphones and everyday devices. One inhibitor of this goal is the difficulty of finding an Augmented Reality application whose usage could satisfy an everyday need or attraction. Augmented reality presents our world in a unique perspective that can be found nowhere else in the natural world. However, visual impact is weak without substance or meaning. The best technology is invisible, and what makes a good product is its ability to fill a void in a person's life. The most important researchers in this field are those who have been augmenting the tasks that most would consider mundane, such as overlaying nutritional information directly onto a meal [4].

In the same vein, we hope to incorporate Augmented Reality into everyday life by unlocking the full potential of a technology often believed to have already reached its peak. The humble photograph, a classic invention and unwavering enhancement to the human experience, captures moments in space and time and compresses them into a single permanent state. These two-dimensional assortments of pixels give us a physical representation of the memories we form in specific periods of our lives. We believe this representation can be further enhanced in what we like to call a Smart Photo. The idea behind a Smart Photo is to unlock the full potential in the way that people can interact with photographs. This same notion is explored in the field of Virtual Reality with inventions such as 3D movies, which provide a special appeal that ordinary 2D films cannot. The 3D technology places the viewer inside the film's environment. We intend to marry this seemingly mutually exclusive dichotomy by processing 2D photos alongside their 3D counterparts.

## 1.2 Contemporary Solutions

Beyond digitizing the physical format, there have not been many solutions for improving photographs. One notable example is GfyCat, which augments animated GIFs by incorporating the features of videos into the format. Similar to Gyfcats, Apple recently released their Live Photos feature on the iPhone 6S, which capture live moments as images that can behave as both static pictures and videos. These two augmentations of photos are great strides in that they demonstrate that a simple photograph can be enhanced and bring us

closer to the reality we attempt to capture. While these two solutions work well in their established context, they do not touch upon another aspect that we believe could strengthen the way we process information by simply viewing a photo. We would like to integrate the 3D world into photographs that we can view on our every day devices, storing our fondest memories in a format that brings us closer to the reality that we experience.

### **1.3 Our Solution**

Using Google's Project Tango as our platform, we will build a system that provides an application as an interface for users to capture a Smart Photo, identify its content, and store it in a gallery. The Tango is an Android tablet made by Google that can map 3D space in real-time, which is a feature that we use heavily in our application. Our application's interface will provide a method for a user to capture a Smart Photo, which will be made of three components. The first component of a Smart Photo is a 2D representation of the object, which will be stored as a typical photograph in the format. The second component will be depth information about the object and its shape, which will be stored as a point cloud. A point cloud is a representation of a 3D shape, which one can simply imagine as a 3D photograph that uses points in space rather than pixels. After the Smart Photo is captured, the 2D image component will be processed by image-recognition tools within our system that will identify the subject or context of the photograph and tag the Smart Photo with a label. This will create an association between the photograph, the the shape, and the label, all of which will be packaged and visible in our application's gallery. None of these concepts, from object recognition to 3D scans, are new to the world of technology, but now they will be available as one simple, accessible format. Everyone will have full access to their memories, now with more detail than ever.



## 2 Requirements

Based on our understanding of the problem, the following requirements are to be met within the scope of the Smart Photos project.

### 2.1 Functional Requirements

- The system will take photographs
- The system will capture 3D point clouds
- The system will present 2D and 3D visuals to the user
- The system will smartly label the subjects and contexts of photographs
- The system will accept and process user feedback on smart labels

### 2.2 Non-Functional Requirements

- The system will be intuitive
- The system will be responsive
- The system will be integrated with Google's Tango platform

### 3 Use Cases

Use Cases describe how our user will interact with the Smart Photos system.

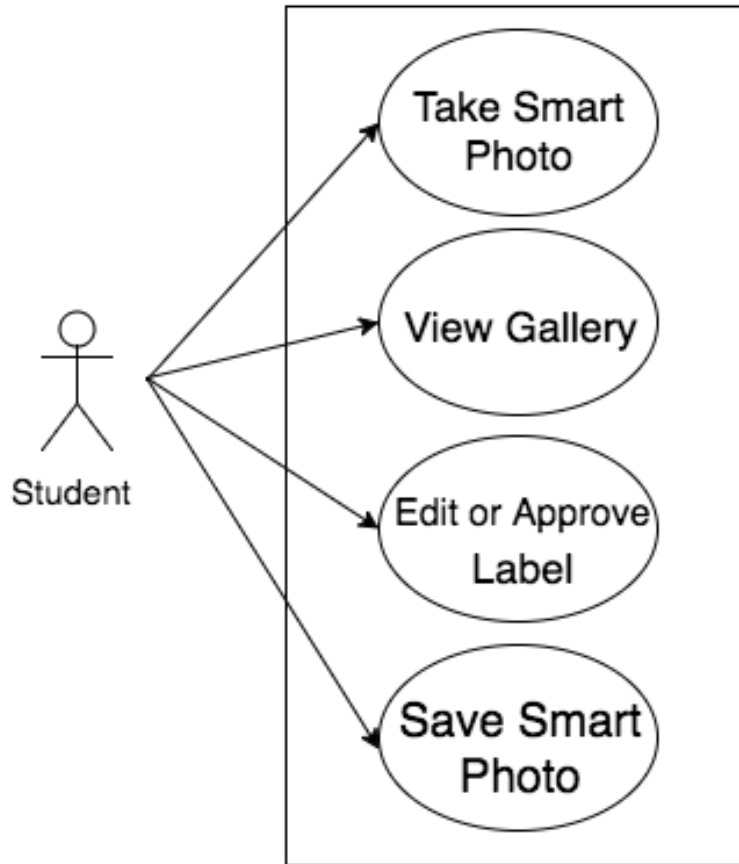


Figure 1: Use Case Diagram

#### 3.1 Use Case #1

**Name** : Take Smart Photo

**Goal** : The system will capture a Smart Photo, which has 2D, 3D, and Smart Label components.

**Actors** : User

**Pre-conditions** :

- Application ready to photograph

**Post-conditions :**

- 2D image and 3D point cloud have been captured and identified and are displayed to the user

**Steps :**

1. User triggers function with button press
2. User captures 2D data
3. User captures 3D data
4. System smartly labels subject and context of Smart Photo

**Exceptions:** User cancels or close capture activity

### 3.2 Use Case #2

**Name :** View Gallery

**Goal :** The user will be able their Smart Photo in a gallery view.

**Actors :** User

**Pre-conditions :**

- At least one Smart Photo has been taken correctly

**Post-conditions :**

- 2D image, 3D scan, and Smart Label are presented together to user

**Steps :**

1. User clicks gallery button on landing page
2. User clicks on a thumbnail to see a full-size image with its Smart Label
3. User presses a 3D Toggle button to switch to 3D view of the Smart Photo

**Exceptions:** None

### 3.3 Use Case #3

**Name:** Edit Smart Label

**Goal:** User provides feedback on the smart label that has been generated by the system for the Smart Photo.

**Actors:** User

**Pre-conditions :**

- Smart Photo has been taken and is displayed to user alongside its label

**Post-conditions :**

- System has received feedback and adjusted its labeling of the Smart Photo

**Steps :**

1. User selects button to edit label
2. User inputs new label
3. New label replaces old label

**Exceptions:** If the generated label is satisfactory, the user will keep it

### 3.4 Use Case #4

**Name:** Save Smart Photo

**Goal:** Save Smart Photo to gallery

**Actors:** User

**Pre-conditions :**

- Smart Photo has been taken and user has completed any feedback.

**Post-conditions :**

- Smart Photo will be saved to the gallery within the app.

**Steps :**

1. Press button to save.
2. New Smart Photo is automatically added to the gallery

**Exceptions:** None

## 4 Activity Diagram

Figure 2 shows an activity diagram describing how users will interact with the Smart Photos application. Our system has been designed to be very simple so the user can take Smart Photos without struggle. Upon opening the application, the user will have the option of either viewing their Smart Photos gallery or capturing a new Smart Photo. If the user decides to capture a new Smart Photo, the camera on the device will activate and allow them to take a 2D photo of a subject. After the 2D photo has been successfully captured, the user will continue by taking a 3D scan of the subject. Once the 3D scan has been successfully rendered, the system will process the Smart Photo to identify its contents. If the generated Smart Label is incorrect, the user may enter in a more appropriate label. After the user has finished the process, they may continue taking more Smart Photos or navigate back to the home screen and then gallery to view the Smart Photos that they have curated. In the Gallery view, the user will be presented with a gallery of Smart Photos, which can be viewed in both 2D and 3D modes.

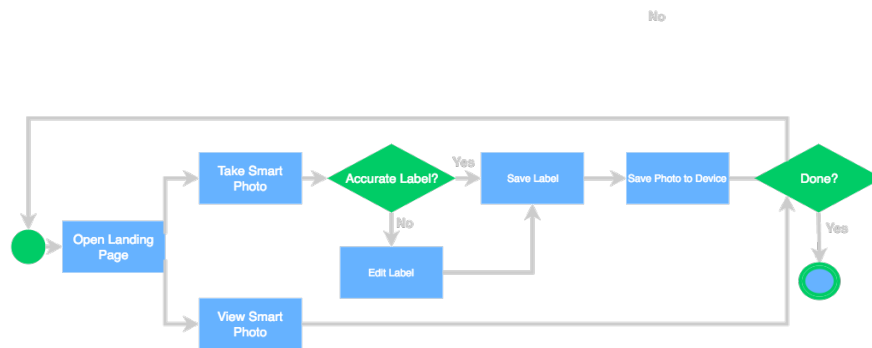


Figure 2: Activity Diagram for the system

## 5 Conceptual Model

Our application's primary concerns are the capturing of Smart Photos and the displaying of Smart Photos. Each presents a different technical and visual challenge. Figure 3 shows the concept for the landing page of the application, where the user can choose to take a photo or view a previously taken photo. Figure 4 shows the 3D component of our Smart Photo with a Smart Label provided by the system, which can be edited if the user likes. The user will have full flexibility to rotate and view their 3D photo from any angle.



Figure 3: Home Screen Concept



Figure 4: Smart Photo with Label Concept

## 6 Technologies Used

We spent a significant amount of time considering the appropriate tools for the creation of our Smart Photos system. While the roster has changed significantly over the course of the project's timeline, the items presented here are the ones used in the final version of our application.

### 6.1 Development Tools

- **Sublime Text** - Text Editor
- **Github** - Version control and code repository
- **Google Drive - Docs - Slides** - Documentation Collaboration Tools
- **Overleaf** - LaTeX Collaboration Tool
- **Adobe Creative Cloud** - Design, Animation, and Graphics Tools
- **Android Studio** - Official Android IDE

### 6.2 Computer Vision Tools

- **Google Project Tango** - Google's Augmented Reality Platform
- **OpenCV and PCL** - Real-Time Computer Vision Libraries
- **Google Vision** - Google's Computer Vision API

### 6.3 Machine Learning Tools

- **Google Tensor Flow** - Google's Machine Learning Framework

## 7 Architectural Design

### 7.1 Overview

Figure 5 is our system’s architectural design, an overview of component layouts and the information flow through the system. Any low-level complexities will remain hidden from the user, who will only interact with the system via touch-screen buttons and gestures.

The client-side makes heavy use of OpenCV and PCL for the scanning and rendering of point clouds. These need to process data quickly, so we used the Android Native Development Kit, which uses C and C++ rather than the Java code that runs on the Android Virtual Machine. The Tango API provides functions in both C and Java, so marrying the Android SDK and NDK went smoothly.

Our server-side heavily relied on communication with Google’s Tensor Flow and Vision services. These services provided our system with the perfect set of algorithms to interpret our image data. Our system sends identification queries to the appropriate REST endpoints and runs image and object recognition algorithms on the data. Afterwards, it passes the image information back down the data pipeline and to the client, which presents the data as a Smart Label to the user.

### 7.2 Architecture Diagram

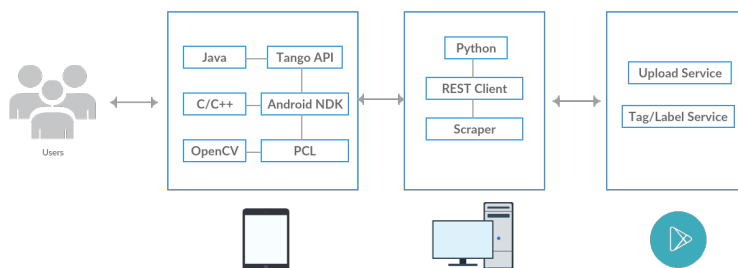


Figure 5: Architectural Overview of Smart Photos



## 8 Design Rationale

**Why Project Tango?** A key component of our system focuses on capturing 3D scans of objects. This presents a several issues, such as the heavy load the task would place on a device’s CPU and GPU, as well as the use of many different computer vision algorithms that would have to be run in order to extrapolate 3D data from 2D images. Project Tango suits our needs because it is capable of mapping the world around itself in real time and returning the map as a point cloud. This is an extremely useful tool because it means that we do not have to use any extra computer vision algorithms such as shading extraction or texture-based recognition in order to extract the shape of an object from its environment. This allows us to skip a step. Whereas most systems would use a camera to capture a photo and then process the image to guess the subject’s 3D shape, the Tango tablet can immediately and accurately fetch the subject’s true shape due to the sensors built into the device.

**Why Android Studio?** Our decision to use the Tango also meant that we would be working on the Android OS platform, so we chose the appropriate development tools that would match that need. We used Android Studio because Google specifically designed it for creating Android applications. For any additional coding and editing, we used Sublime Text.

**Why GitHub and Google Drive?** For our version control and code backup system, we chose GitHub because of its proven reliability in both academic and professional settings as a reputable tool that even large companies use for sharing code. For writing documentation, we used Google Drive’s document-editing applications Google Docs and Google Slides for their live collaborative features, which allowed us to quickly create powerful visual presentations for our demos and reviews.

**Why OpenCV and PCL?** In addition to the point cloud provided by the Tango’s API and capturing tools, we needed to render them conveniently and quickly, so we used these computer vision and point cloud processing libraries.

**Why Adobe Creative Cloud?** Both for our concept mockups and actual application, we used the Adobe Creative Cloud Suite tools in order to facilitate our rapid prototyping. We could quickly create mockups and animations before implementing a feature. Because of this, we were able to focus most of our time and attention on coding, as all animations and designs were quickly handled with these tools.

## 9 Testing

We divided testing for our application into the two most complex features: 3D capture and smart labeling.

Figure 6 shows a 3D scan taken with our application. We conducted test scans of a variety of objects, making sure to vary their size, color, and environment, so that we could note any deficiencies in our system’s capturing functionality.

While taking 3D scans we quickly realized the Tango’s sensors have difficulty detecting the color black. This made taking 3D scans of any dark screens such as laptops, TVs, and cellphones nearly impossible. Darker objects and subjects also proved difficult to render with higher detail, as seen in Figure 6. Additionally, because the Tango relies on infrared light, high levels of ambient light such as from sunlight or incandescent bulbs would interfere with accurate capturing of the scale or proportions of objects [1]. Moving forward with such knowledge, we limited our testing of the 3D capture to night time or rooms with soft, tungsten lighting. Figure 7 shows a picture of a statue outside the Santa Clara University library. This photo was taken at night under Tungsten lighting. The shadowed parts of the image were unable to be captured due to the aforementioned limitation of the Tango sensor.



Figure 6: 3D capture of a standing individual



Figure 7: 3D capture of Santa Clara University book statue

To test the labeling module, we ran dozens of images through the image recognition services on the server-side. Table 1 displays an excerpt of our data. Google’s Tensor Flow and Vision have proven rather vague when identifying objects, as the API’s accessibility is currently limited to recognizing only general contexts. When photographing objects to be identified, we found a higher chance of accurate results when isolating the subject of the photo or positioning the item to take up at least 80 percent of the frame.

Input	Output
Cheetos Bag	"Food"
Apple	"Fruit"
Angela	"Person"
Daniel	"Athlete"
Jaelin	"Black"
Flowerpot	"Flower"
Car	"Vehicle"
iPhone	"Mobile Device"
Water Bottle	"Water Bottle"
Eyeglasses	"Vision Care"

Table 1: Object Recognition Test Results

## 10 Risk Analysis

### 10.1 Overview

We analyzed the risks our project faced and brainstormed mitigation strategies. The risks were ranked by their probability and severity in the table below.

### 10.2 Risk Table

Risk	Probability	Severity	Impact	Mitigation
Bugs in Code	1	6	6	Code properly.
Poor Code Design	.7	8	5.6	Use proper design practices and modularity.
Lack of Reusable Existing Tools	.7	8	5.6	Shape design around existing tools.
Platform Failures and Bugs	.6	9	5.4	Monitor Tango updates, patch notes, and announcements.
Conflicting Agendas	.8	6	4.8	Create group schedule.
Component Incompatibility	.6	6	3.6	Design components to be modular; implement good coding practices.
Poorly Defined Requirements	.4	6	2.4	Research properly.
Communications Breakdown	.2	9	1.8	Be upfront. Be empathetic. Be a team.
Data Storage Loss	.2	9	1.8	Make backups.
Exhausting Funds	.1	8	0.8	Make a budget.

## 10.3 Risk Descriptions

**Bugs in Code** - These seemed very likely given that we worked with an unfamiliar platform and code. This turned out being very accurate, as several times we faced heavy delays doing what seemed to be simple tasks such as saving a photo.

**Poor Code Design** - We told ourselves "If we don't take the right approach, we will make our lives much harder". We ensured our code and architecture is easy to understand.

**Lack of Reusable Existing Tools** - Due to our lack of experience in the fields of Computer Vision and Machine Learning, we hedged on the fact that we could find libraries and tools to help us. We cycled through various tools before settling on our current selection, which meant changing the direction and scope of the project several times.

**Platform Failures and Bugs** - Project Tango is a fairly new platform and very prone to bugs or failures that need to be patched. We were held up many times by these limitations and had to find creative ways around them.

**Conflicting Agendas** - The team had to juggle everything from classes to job applications to our social lives, so we learned to coordinate our schedules and meetings, both online and in-person, as well as learning to work alone.

**Component Incompatibility** - Since we were building two complex modules, one for 3D scanning and one for object recognition, we knew there was a risk of the data formats being incompatible. To resolve this, we made each module of the system pass only basic pieces of data such as strings or images, and all complex data such as 3D maps were handled solely by their owner modules.

**Poorly Defined Requirements** - We reshaped our application's direction, goals, scope, and requirements multiple times. Because we didn't have a client other than ourselves, we had the luxury of doing so. At the beginning, each of us had our own particular interpretation of what the application would do, and we had to communicate properly, as well as refine, redefine, and research what our project could and would do.

**Communications Breakdown** - Several times, we faced dead weeks of silence or stillness in the advancement of the project, but we maintained strong communication between the team members about our lives and our work. Despite our arguments, frustrations, and exhaustion, we came together as a team.

**Data Storage Loss** - We considered this risk very unlikely to occur, and it never did, but we made sure to have backups on GitHub and Google Drive of all our data, documentation, and code.

**Running Out of Funds** - Our adviser and our school have been very supportive, so we continued receiving the funds we needed in order to purchase Tango tablets for everyone on the team to work off. Additionally, we made a budget that we have been monitoring and have run into no financial issues.

# 11 Development Timeline

## 11.1 General Overview

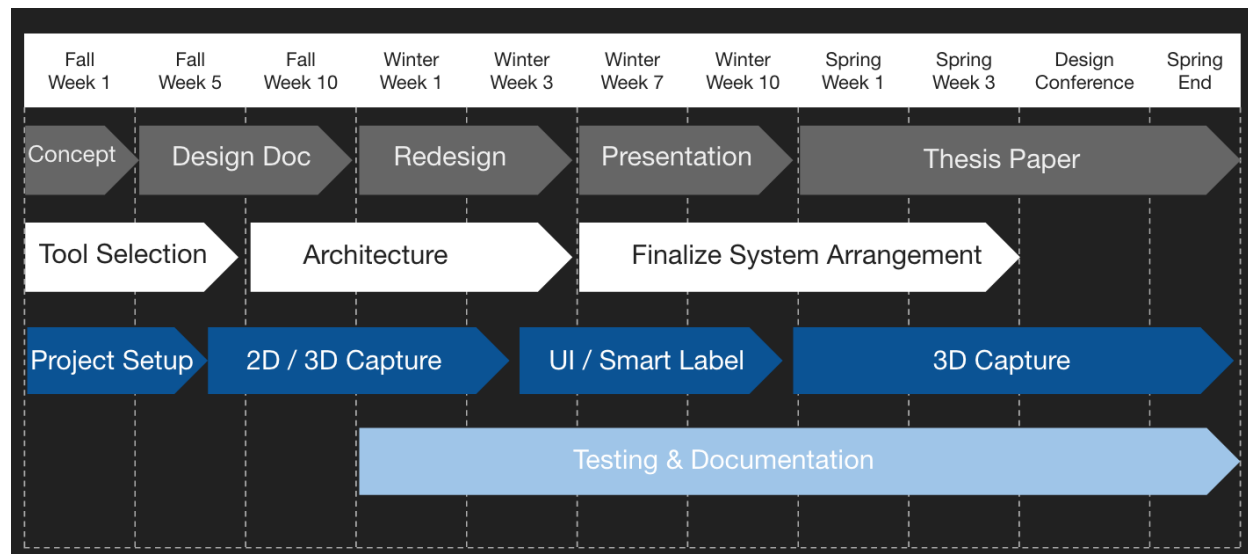


Figure 8: Development Timeline

## 11.2 Development Details

### 11.2.1 Documentation

**Concept** - At the very beginning, we had several documents filled with ideas for directions the project could take. We had many links to articles, videos, and papers that discussed a variety of topics. We knew we wanted to do Augmented Reality, and we decided to focus on the Tango's 3D capture capabilities at this point.

**Design Document** - Our early Design Document had a heavy focus on designing a machine learning cluster that could dissect the shapes in a 3D point cloud. We ended up cutting many of these later and replacing our own prototypes with open source tools we found after we learned more about the internals of recognition systems.

**Redesign** - By the start of Winter Quarter, we had each developed our own vision of the project, and we needed to link our ideas. At this point, we settled on the concept of Smart Photos, made of three components.

**Presentation** - During Winter Quarter, we prepared what would become the final version of our presentation before several demos we gave to various professors.

**Thesis Paper** - Most of Spring Quarter was spent piecing together the final thesis paper,

especially once the Design Conference Presentation was finished.

### 11.2.2 Design

**Tool Selection** - During the drafting and early prototyping stage of the Smart Photos project, we searched for tools that could help accelerate our development process.

**Architecture** - At this point during the project, we placed most of the Computer Vision components onto the Tango client because the GPU could handle the load well. The Machine Learning clusters would have to be on the server-side.

**Finalizing System Arrangement** - Our original architecture design was very thorough, but we had to make adjustments after the project redesign. Afterwards, we continued to make adjustments to the components as we finished writing the system.

### 11.2.3 Implementation

**Project Setup** - The first few weeks were spent setting up our development environments and tools. Everything from IDEs, text editors, repositories, libraries, compilers, SDKs, platforms, and tools were configured.

**2D / 3D Capture** - We first created rough demos for how the 2D and 3D capture modules would look and feel.

**UI / Smart Label** - After running into difficulties with 3D capture, we focused on the Machine Learning portion of the project and implemented the Smart Label functionality. Additionally, we created the UI design elements and rules that our application would follow. We began tying the modules of the system together into the core application.

**3D Capture** - We made good use of the Tango's internal tools to add 3D capture functionality to our application. This was the most rigorous part of the project and was the final feature we completed before our Design Conference demo.

### 11.2.4 Testing

**Testing and Documentation** - Starting shortly after we finished our Design Document, we made sure to add proper documentation in our code and to frequently test all our features. Overall, we've probably taken hundreds of Smart Photos over the course of the project.

## 12 Conclusion

### 12.1 Augmented Reality Today

In their original promotional video for Project Tango, Google showcased the tablet's Augmented Reality functionality in many contexts, from entertainment and games to a virtual guide dog for the visually impaired. It has been over a year and a half since that moment, and many companies and developer teams have made their own products for the mass market, rather than niche applications. This has proven interesting enough to catch the media's and public's attention with major releases such as Microsoft's HoloLens and Tesla's self-driving cars. One group has developed algorithms to generate detailed and accurate information, in real time, for blind users to explore photos on a touch screen device [2]. Some teams have even beginning to augment meditation and online shopping [3, 5]. Google has officially partnered with Lenovo to bring Project Tango's technologies to smartphones; soon, our mobile devices will include powerful processors and cameras that have the ability to grant us various perspectives of our every day world [6]. What started as a niche field has been booming in popularity in 2016, and although most work is still in the hands of a small batch of developers, new applications and uses are released every day.

### 12.2 Future of Smart Photos

Our ultimate goal for Smart Photos is to seamlessly package a 2D photo, 3D mesh, and a smart label in a single format. We hope for the 2D and 3D components of the photo to be stored concurrently or as one file. We also hope for our project to be integrated into people's current devices such as phones or tablets, rather than an exclusive developer's device like the Tango, so as to be unobtrusive and succinct. With minimal additional effort, we can add features such as voice-retrieval, which uses voice commands to search for captured photos by context, as there is no current way to search through a phone's Camera Roll for a particular image other than manually browsing through the entire feed. This is one example of the fluidity that Augmented Reality can give photographs. We didn't create Smart Photos to be an application, we created them to be a format.

### 12.3 Last Words

With Smart Photos, we have fulfilled a need. Our product can integrate itself into people's everyday lives. At any moment, photos are being shared and broadcasted across various social media platforms. Entire websites, products, and companies have been built around the use of taking and sharing photos. People use photographs to depict emotions



and capture stories, or for storing information and remembering moments. By augmenting photographs, we have enhanced the way people interact with the world. We can capture more of the reality we see; all the colors, the shapes, and contexts are preserved. Now anyone can do this. It seemed like a smart idea at the time. That's why we named them Smart Photos.

## 13 References

1. "Project Tango UX Best Practices." Google Developers, n.p., n.d. Web. 31 Mar. 2016.
2. Banf, Michael, and Volker Blanz. "Man Made Structure Detection and Verification of Object Recognition in Images for the Visually Impaired." Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications - MIRAGE '13 (2013): n. pag. ACM Digital Library. Web. 30 Nov. 2015.
3. Roo, Joan Sol, Renaud Gervais, and Martin Hachet. "Inner Garden." Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '16 (2016): n. pag. Web.
4. Henze, Niels, Thomas Olsson, Stefan Schneegass, Alireza Sahami Shirazi, and Kaisa Väänänen-Vainio-Mattila. "Augmenting Food with Information." Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia - MUM '15 (2015): n. pag. Web.
5. Tseng-Lung Huang<sup>1</sup>, blessing@saturn.yzu.edu.tw, and 136466@mail.tku.edu.tw Chung-Hui Tseng<sup>2</sup>. "Using Augmented Reality To Reinforce Vivid Memories And Produce A Digital Interactive Experience." Journal Of Electronic Commerce Research 16.4 (2015): 307-328. Applied Science and Technology Source. Web. 23 January 2016.
6. Welch, Chris. "Lenovo Is Making the First Consumer Phone with Google's Project Tango." The Verge. N.p., 07 Jan. 2016. Web. 02 May 2016.