Project Proposal

# Mixed Reality for Biopsy Site Localization

## EN.601.456 Computer Integrated Surgery II

**Ruby Liu**

Biomedical Engineering Undergraduate Student

rliu34@jhu.edu

### Other team members:

Liam Wang, Biomedical Engineering Undergraduate Student (external to CIS II)

Dr. Peter Kazanzides: Research Professor, Department of Computer Science

Dr. Ashley Antony: Resident Doctor, Dermatology

Dr. Jeffrey Scott: Assistant Professor, Dermatology

Dr. Kristin Bibee: Assistant Professor, Dermatology

Dr. Elise Ng: Assistant Professor, Dermatology

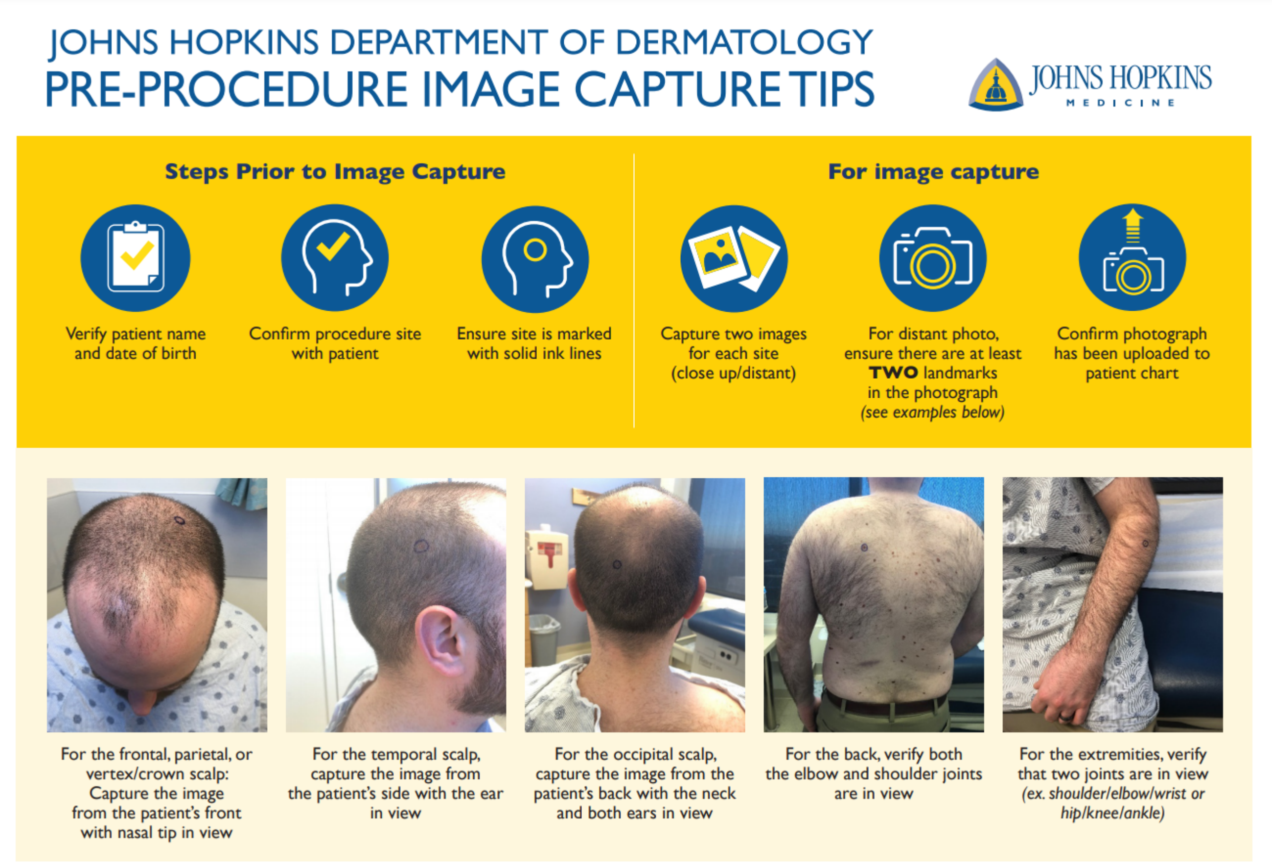
# 1. Overview

Skin biopsies are used by dermatologists to diagnose cutaneous ailments, including tumors and rashes. However, if a surgery becomes necessary after a biopsy, determining the original site of the biopsy can be difficult due to various factors including the skin healing, biopsy depth, and background skin disease. This difficulty can lead to wrong site surgery, which is a never event — an error that is preventable and should never occur.

This project's goal was to create a mobile augmented reality application, which we call **BiopSight**, to be deployed on a phone or tablet. The application can register biopsy images to surgery images and subsequently overlay the biopsy site on live camera images taken by the mobile device. This would provide dermatologists with guidance sufficient to locate the biopsy site on the patient at the time of surgery. We have been successful in building a functional application, and are currently working towards licensing the application to deploy it for widespread testing.

# 2. Background

The usual procedure for biopsy site identification involves photographs of the biopsy site at the day of biopsy. While a lack of standardization is one of the factors that can make biopsy site identification more challenging [10], the Johns Hopkins Department of Dermatology has a typical procedure [12]  that our application is intended to work with.



JOHNS HOPKINS DEPARTMENT OF DERMATOLOGY
PRE-PROCEDURE IMAGE CAPTURE TIPS

JOHNS HOPKINS MEDICINE

**Steps Prior to Image Capture**

- Verify patient name and date of birth
- Confirm procedure site with patient
- Ensure site is marked with solid ink lines

**For image capture**

- Capture two images for each site (close up/distant)
- For distant photo, ensure there are at least **TWO** landmarks in the photograph *(see examples below)*
- Confirm photograph has been uploaded to patient chart

For the frontal, parietal, or vertex/crown scalp: Capture the image from the patient's front with nasal tip in view

For the temporal scalp, capture the image from the patient's side with the ear in view

For the occipital scalp, capture the image from the patient's back with the neck and both ears in view

For the back, verify both the elbow and shoulder joints are in view

For the extremities, verify that two joints are in view *(ex. shoulder/elbow/wrist or hip/knee/ankle)*

Fig. 2.1: Johns Hopkins Department of Dermatology pre-procedure image capture tips. Each capture of a biopsy site requires that the site is marked with a solid ink line, and that two images are taken: one close and one distant.

However, even with two photos, misidentifications can occur due to various factors.

## Clinical Motivation

Wrong-site surgery, which can be a result of biopsy site misidentification, is a never event, a medical error that should never occur. However, in a study, physicians misidentified 5.9% of biopsy sites, and the surgeon and patient both misidentified 4.4% of the sites [11]. Despite the use of photography, biopsy site identification remains challenging.

If we are successful, the mobile application could be used by dermatologists to improve the accuracy of biopsy site localization, reducing the likelihood of incorrect site identification and thus reduce the number of or eliminate wrong-site surgeries resulting from biopsy site misidentification.

## Prior Work

Others have attempted to address this need using various methods and tools, including a UV-fluorescent tattoo [2, 3], a transparent grid [4], confocal microscopy [5], “selfies” [6, 7, 8, 9], and facial recognition with augmented reality [10].

However, none of these have been incorporated into general practice yet, possibly due to cost, insufficient reliability, excessive disruption to the typical workflow, or concern of reaction from the patient (as is the case with a UV-fluorescent tattoo). Additionally, the existing augmented reality method using facial recognition does not provide a live image overlay and is only effective for biopsies on the face.

## Goals

Our specific aim is to create a mobile augmented reality application, deployed on a phone or tablet, that can register biopsy images to surgery images and subsequently overlay the biopsy site on live camera images taken by the mobile device. We hope to provide handheld and convenient augmented reality image guidance that will allow the dermatologist to locate the biopsy site to about 5mm of accuracy, at which point they may be able to identify the biopsy site on their own.

# 3. User Workflow

Our intention is to create an application with the following UI workflow:

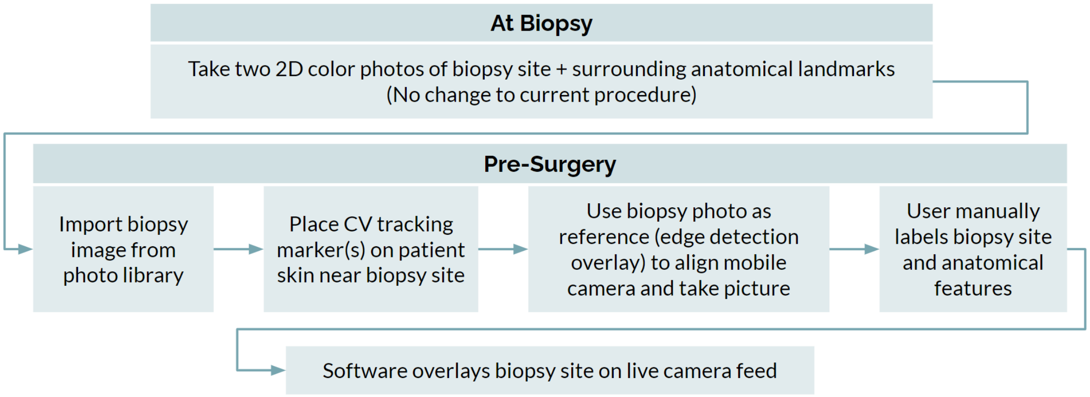

At Biopsy
Take two 2D color photos of biopsy site + surrounding anatomical landmarks
(No change to current procedure)
Pre-Surgery
Import biopsy image from photo library
Place CV tracking marker(s) on patient skin near biopsy site
Use biopsy photo as reference (edge detection overlay) to align mobile camera and take picture
User manually labels biopsy site and anatomical features
Software overlays biopsy site on live camera feed

Fig 3.1: UI workflow. An overview of the user's experience of the application.

At the time of biopsy, the procedure does not change: the dermatologist will take two 2D color photos of the biopsy site, one close up and one at some distance so as to capture anatomical landmarks.

When the patient comes in for surgery, the dermatologist will import the biopsy image from their photo library on their mobile device. They will also place computer vision tracking markers on the patient near the biopsy site.

Then, the application will provide an edge overlay using the biopsy photo in order to assist in taking the surgery photo, so that the two images can be as similar as possible. The user will then manually label the biopsy site on the biopsy photo(s) and anatomical features on both the biopsy and pre-surgery photos.

After that, the software will internally register the biopsy site to the markers and then overlay the biopsy site on the live camera feed.

# 4. Technical Approach

Broadly speaking, our application has three parts: the registration algorithm, the live marker tracking, and the mobile augmented reality application. The registration algorithm and live marker tracking were initially drafted in Python on a laptop before being rewritten into Objective-C, which can be used directly for the app when developing in XCode.

Below is the initial concept for our approach, and below that is a flowchart of the final functional approach for both the Python code and the iOS application.

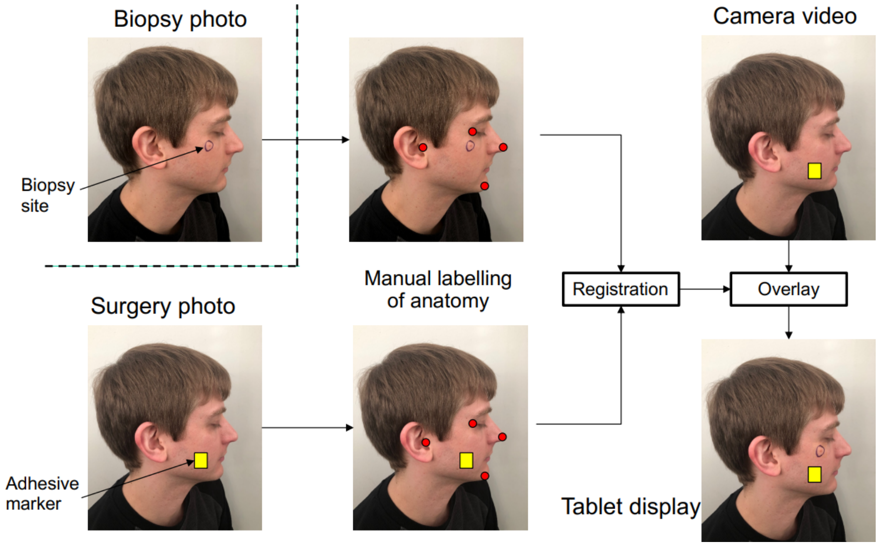


Fig. 4.1: A very basic overview of how the application will work. The user will have biopsy photos and surgery photos available, and will place tracking marker(s) on the patient. The user will manually label anatomical tracking points on the biopsy and surgery photos before the program registers the two photos, after which the program will provide a live overlay of the biopsy site.

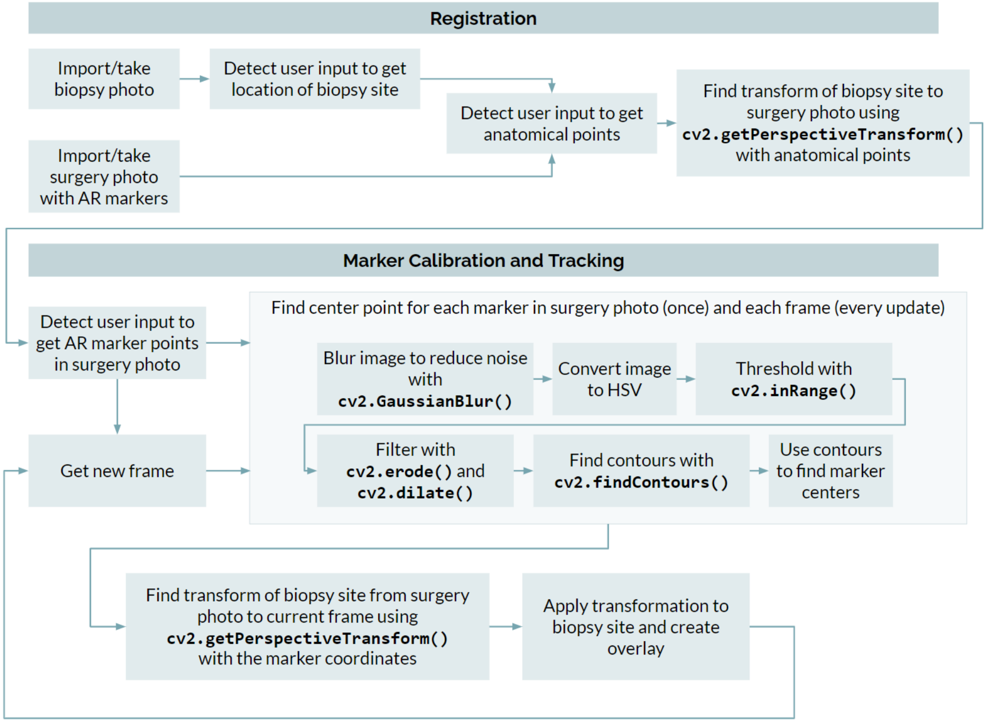

Registration
Import/take biopsy photo
Detect user input to get location of biopsy site
Detect user input to get anatomical points
Find transform of biopsy site to surgery photo using cv2.getPerspectiveTransform() with anatomical points
Import/take surgery photo with AR markers
Marker Calibration and Tracking
Detect user input to get AR marker points in surgery photo
Find center point for each marker in surgery photo (once) and each frame (every update)
Blur image to reduce noise with cv2.GaussianBlur()
Convert image to HSV
Threshold with cv2.inRange()
Get new frame
Filter with cv2.erode() and cv2.dilate()
Find contours with cv2.findContours()
Use contours to find marker centers
Find transform of biopsy site from surgery photo to current frame using cv2.getPerspectiveTransform() with the marker coordinates
Apply transformation to biopsy site and create overlay

Fig. 4.2: A flowchart outlining how our application works. Full resolution image is available on the repository **here**.

## Registration Algorithm

The registration algorithm was first implemented using Python on Windows 10 with OpenCV packages. For detailed explanations of the code, see the PythonDraft README documentation in the Gitlabs repository.

The program inputs user clicks as pixel coordinates in both biopsy and surgery photos for the biopsy site (in the biopsy photo) and anatomical points (in both photos).

Then, the program uses the OpenCV function `cv2.getPerspectiveTransform()` to find the homographic transformation from the biopsy photo anatomical points to the surgery photo anatomical points. This transformation is then used to transform the biopsy site to its location in the surgery photo.

Since the markers we used (Avery Removable Color Coding Labels, 0.25 Inches) are of known dimension, we determined visually that the accuracy of registration was sufficient enough. For future work, once we are able to deploy the application for widespread testing, we can take data on inter-rater and intra-rater reliability for registration accuracy, as it depends strongly on how well the anatomical points are picked.

## Live Marker Tracking

The live marker tracking was also implemented with OpenCV packages, first drafted using Python on desktop; implementation details are in the PythonDraft README.

The markers we used have been mentioned previously: Avery Removable Color Coding Labels, 0.25 Inches. The user must place these near the biopsy site.

After registration is complete, the application will request that the user select points for each colored marker on a camera freeze frame that acts as the surgery photo; this is used to calibrate the hue/saturation/value (HSV) threshold and get the initial marker coordinates.

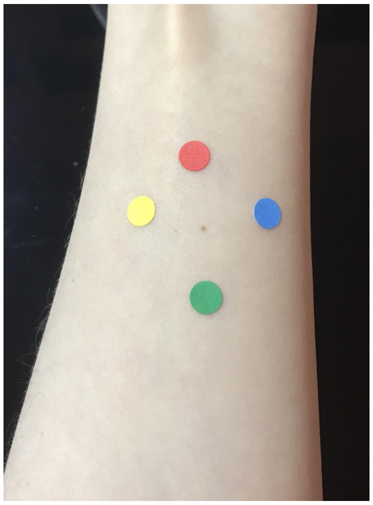


Fig. 4.2: The stickers used for live image tracking.

For each frame, including the freeze frame, the `getMarkerCoords()` function is used to find the centers of the markers. The function first uses the `cv2.inRange()` function for the HSV threshold, and then filters out some noise using the `cv2.erode()` and `cv2.dilate()` functions. Then, the contours are found using `cv2.findContours()`, and if one is found, the largest one is used as the marker contour, and the center of it is used as the marker coordinate.

To find the biopsy site location for each frame from the live camera feed, we find the homographic transform (`cv2.getPerspectiveTransform()`) from the marker coordinates in the surgery photo / freeze frame to the marker coordinates in the new frame. Then, we use this transformation to transform the biopsy site coordinates from the registered surgery photo to the current frame.

## Application Development

We decided to use XCode to develop the application and rewrite the Python code for the registration and live marker tracking into Objective-C, which can be used directly on the application. This was fairly straightforward, as OpenCV was the primary package used, and is available for both languages; the same functions were thus used for both the Python draft and Objective-C code, with some minor differences for user experience improvements.

The photo registration and setup GUI was developed using the iOS storyboard system with the UI backend written in Swift. Meanwhile, the image processing pipeline backend was written in Objective-C using OpenCV for iOS. The CocoaPods dependency manager was used to load a precompiled OpenCV binary with Objective-C headers into the XCode project. We used the CVVideoCamera interface to do real-time image processing on a live iOS camera feed.

One change between the Python and iOS versions is a feature added to allow multiple tracking markers with the same color to be used. To use the previously generated image registration, we needed to match each tracking marker in the camera frame with the corresponding marker in the surgery image. By requiring that a minimum of two colors of tracking markers are used, we could use the order of the tracking markers in the circular pattern to identify corresponding markers. More details can be found in the iOSApp README.

# 5. Results

For the desktop application, we qualitatively tested its functionality by simply running the application and observing how close the marked biopsy site is to the real biopsy site. In the image below, you can see a freeze frame of the camera feed with the four markers indicated by their centers and contours, and the biopsy site marked by a cyan dot. The “real site” is the freckle just slightly above and to the left of the cyan dot; we observed that the two are very close, less than the diameter of the markers, which are a known 1/4 inch (6.35 mm) and thus proceeded to transfer this functionality into the iOS application.

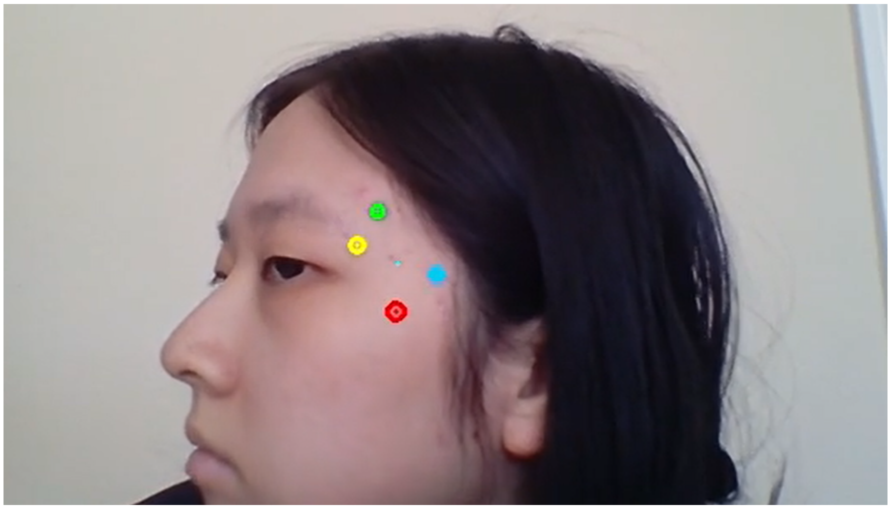
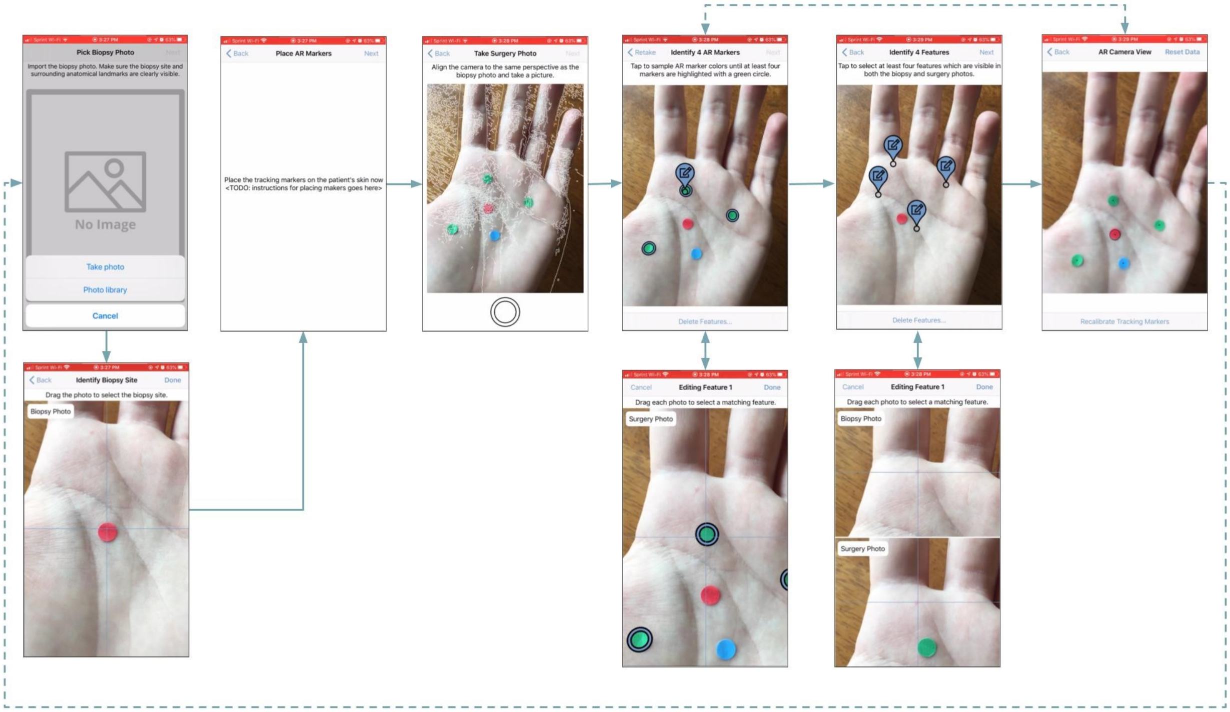


Figure 5.1: The overlay result of the registration and marker tracking on the Python desktop application.

For the iOS user interface, we accomplished our main goals of importing a biopsy photo, taking a surgery photo, requesting the user to select anatomical points and marker locations, and displaying an augmented video feed. An outline of the process can be seen below.



Import the biopsy photo. Make sure the biopsy site and surrounding anatomical landmarks are clearly visible.

Place the tracking markers on the patient's skin now <TODO: instructions for placing makers goes here>

Align the camera to the same perspective as the biopsy photo and take a picture.

Tap to sample AR marker colors until at least four markers are highlighted with a green circle.

Tap to select at least four features which are visible in both the biopsy and surgery photos.

Drag the photo to select the biopsy site.

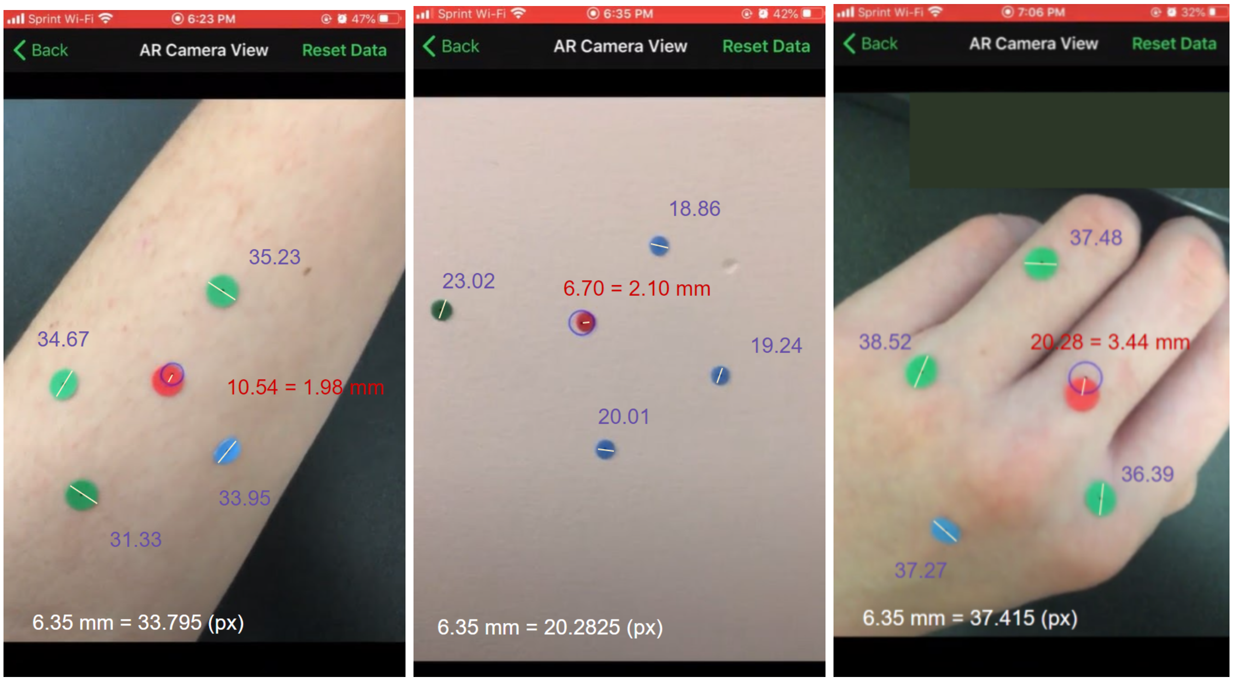
Drag each photo to select a matching feature.

Drag each photo to select a matching feature.

Figure 5.2: Flowchart/outline of the UI process, indicating how the user progresses to each subsequent screen. A higher resolution image can be found on the repository **here**, and a demo video can be seen/downloaded **here**.

For testing the iOS application, we took videos of the final overlay result with three different setups: on an arm, on a wall, and on a hand. The arm can be considered a general case, the wall can be considered a case with few distinguishing features, and the hand could be considered a case for a non-planar surface (which is not ideal for the homographic transformation calculation). In all cases, an untracked red AR marker was used to indicate the "real" biopsy site, i.e. the real site is at the center of that marker.

We took the freeze frames that appeared to have the greatest error for each video, and then used FIJI/ImageJ to measure the diameters of the markers in pixels to find a conversion from pixels to millimeters. Then, we measured the distance of the overlay biopsy site to the "real" biopsy site and converted it to millimeters. The results can be seen below.



**Arm:** max error 1.98mm **Wall:** 2.10mm **Hand:** 3.44mm

Figure 5.3: The overlay result of the iOS application. The blue and green markers are tracked by the application and marked by a black dot in their centers. The overlaid biopsy site is marked by a black dot with a blue ring around it. The "real" biopsy site is the center of the red marker, and is not tracked by the application. The diameters of the markers in pixels as measured by FIJI/ImageJ are annotated as purple text; the final calculation of the error is shown in red text.

We thus observe that we can maintain a sufficiently accurate (less than 5mm, as requested by our clinical mentor) biopsy site overlay given good registration.

# 6. Discussion and Future Work

We have been successful in creating an application that accomplishes all of our goals within the scope of the project fairly consistently, but with a very limited number of testers and experiments. However, we would like to go further in the project in the future through more thorough tests and deploying the app on the Apple App Store for widespread use.

These two things are, in fact, interrelated. We intend to license the application so that we can deploy a beta version on TestFlight that users can download; this can allow many users to provide both qualitative and quantitative feedback. Particularly, we would like to gather quantitative data from users to determine the interpersonal and intrapersonal consistency and accuracy for both the registration and the final overlay. This way, we could determine various things, including how much the registration affects the final result, how much results vary from person to person (which could indicate an external factor determining accuracy, such as personal skill with the application), and how much the results vary for a single individual (which could be used to determine whether user interface (UI) or algorithmic improvements are necessary).

A possibility in the execution of this testing could involve a similar setup to the testing done for the application as seen in the Results section, with a red sticker indicating the real biopsy site, only the red sticker is also tracked independently from the overlay. The application could then measure the distance in pixels between the real and overlaid biopsy site, using the diameters of detected marker contours to estimate the pixel-to-mm conversion and thus estimate the error. This could allow for automated error metrics, which is much more efficient though possibly less accurate at times.

Additionally, the beta testers could very well be the clinicians who may be the ones eventually using the application. They could thus provide feedback on user experience that could help us further improve our application to suit their needs.

We are also still currently working on improving the UI, both in aesthetics and user experience, though the current state is sufficiently functional. A mockup of the goal for the final UI can be seen below.

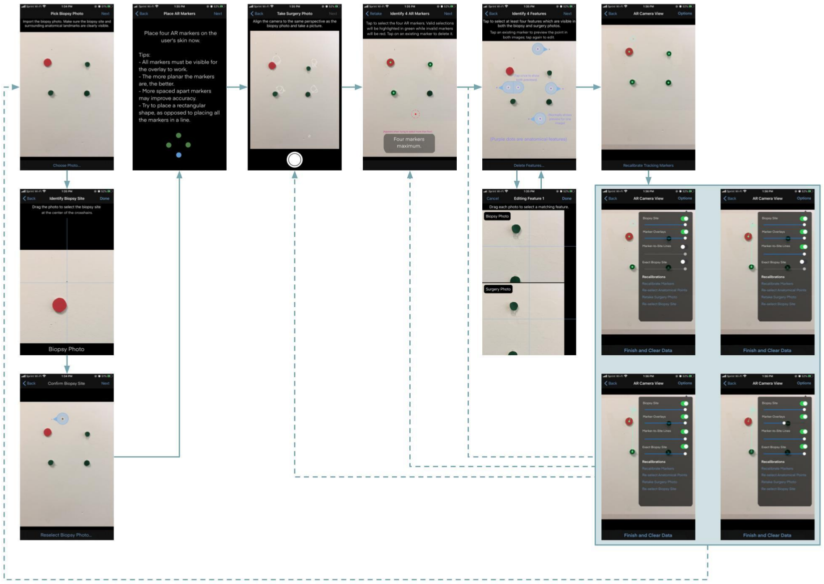


Figure 6.1: Mockup of the goal for the final UI. The full resolution image is available in the repository **here**.

# 7. Progress Evaluation

Overall, the project went very smoothly. There were some delays in documentation due to lack of clarity of what was necessary, but we have altogether accomplished most everything we set out to do.

## Dependencies

All our dependencies were met in some capacity. The only dependency we used a contingency for was the dependency for a cross-platform application development platform; we decided to use XCode, which is available for MacOS only.

| Dependency | Need | Contingency | Status | Planned Deadline | Hard Deadline |
|---|---|---|---|---|---|
| Biopsy photos from Dr. Antony | For testing the registration algorithm | Photos of ourselves | Met | 2/19 | 2/26 |
| Computer/internet access | For software development and communication | If technical difficulties — repair or use alternate device. Internet — mobile data. | Met | Continuous | Continuous |
| Mobile device | For testing mobile application | Use mobile device software simulators | Met | Continuous | Continuous |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Platform to develop application | Platform that isn't specific to iOS or Android and able to develop on Windows and MacOS | If not possible for technical reasons, use XCode (MacOS dev only) | Decided to use XCode | 2/26 | 3/5 |
| Stickers | Markers for computer vision tracking | Print colored dots and tape them on | Met, purchased from Amazon | 3/1 | 3/15 |
| Being able to load our application to an independent device | Independence would be useful for user testing, but iOS development restrictions may prevent easy deployment (may need a license or to stay plugged in to a computer) | Keep device plugged in, look for other methods of deployment, or buy license | Met | 4/1 | 4/15 |

Fig 7.1: Table of dependencies.

## Deliverables

Our deliverables are shown below. All of the deliverables have been met, and we are currently in the process of pursuing further advances beyond the CIS II course.

|  | Deliverable | Expected Completion |
|---|---|---|
| **Min** | Basic placeholder application | 2/26 |
|  | Algorithm to register biopsy site photos to another photo / marked photos with documentation | 3/5 |
| **Expected** | Algorithm to track markers and overlay biopsy site to live video / video with tracking with documentation | 4/2 |
|  | Error metrics to quantify accuracy of the live overlay | 4/9 |
|  | Basic working interface with calibration overlay guidance with application documentation | 4/2 |
| **Max** | Completely functional mobile application with documentation | 5/1 |
|  | Experimental data to quantify the geometric accuracy of our application | 5/1 |

Fig 6.1: table of deliverables.

Most of the deliverables were successfully met on time, though there were some delays in proper documentation for the code due to lack of clarity on what was necessary; still, much of the code was functional and commented properly with docstrings and such on time. We also did the pixel and millimeter error metrics at the same time, as it was a simple matter to find the conversion from available images.

# Timeline and Milestone Adherence

The timeline of our project is split into two sections: the registration and tracking, which Ruby primarily worked on, and the mobile application development, which Liam primarily worked on. The milestones are a slightly more detailed breakdown of the deliverables.

## Registration and Tracking

| Milestone | Expected Completion |
|---|---|
| Create basic I/O application to record user clicks on biopsy images | 2/26 |
| Finish algorithm to register biopsy site photos to another photo + documentation | 3/5 |
| Finish algorithm to track markers | 3/15 |
| Finish algorithm to overlay biopsy site to live video with marker + documentation | 4/2 |
| Quantify accuracy of the live overlay with pixel error metrics | 4/9 |
| Acquire experimental data and quantify errors in real units | 5/1 |

## Mobile Application Development

| Milestone | Expected Completion |
|---|---|
| Create basic placeholder mobile application and determine how programs will interface | 2/26 |
| Create edge detection overlay for photography guidance and document code | 3/5 |
| Have a working UI to select points on images | 3/15 |
| Integrate photo registration and marker tracking into the mobile application | 4/2 |
| Complete and deploy final application with documentation | 5/1 |

As mentioned in the Deliverables section, the documentation was what gave us the most trouble. The pure development of the basic code and application went smoothly and quickly, and we are now polishing the application and preparing to deploy it for beta testing on the Apple Store using TestFlight, which is not technically one of our deliverables; as of the moment, the application has only been deployed on Liam's iPhone for development.

## 8. Conclusion

The overall goal of this project, which we have achieved (though still will pursue further performance testing), was to create an application that can guide dermatologists to biopsy site locations within a 5 millimeter range of accuracy. Our clinical mentor Dr. Antony has stated that this would be sufficient to visually confirm the exact location of the biopsy site. In turn, this may be very helpful to dermatologists when they are struggling to locate a biopsy site and thus lower the likelihood of the highly undesirable outcome of wrong-site surgery.

Of course, in order to truly make an impact, we must license the application and make it accessible to the dermatologists who need it; we would also ideally have thorough data on performance that could be used for user advisements or improvements to the application itself.

Overall, this project was truly an exercise in project management, exposing me to the possible demands of any research project I might pursue in the future. Additionally, I learned much more about how necessary good documentation is for software development, and, while I ended up not making industry-standard documentation files, I did learn a bit about what might be necessary in the medical device industry. Regardless, over this semester, we have accomplished what we set out to do, and we are well on the way to fulfilling the need of better biopsy site localization.

# 9. References

[1] Zhang J, Rosen A, Orenstein L, et al. Factors associated with biopsy site identification, postponement of surgery, and patient confidence in a dermatologic surgery practice. J Am Acad Dermatol. 2016; 74:1185-1193.

[2] Chuang GS, Gilchrest BA. Ultraviolet-fluorescent tattoo location of cutaneous biopsy site. Dermatol Surg. 2012 Mar;38(3):479-83. doi: 10.1111/j.1524-4725.2011.02238.x. Epub 2011 Dec 15. PMID: 22171575.

[3] Russell K, Schleichert R, Baum B, Villacorta M, Hardigan P, Thomas J, Weiss E. Ultraviolet-Fluorescent Tattoo Facilitates Accurate Identification of Biopsy Sites. Dermatol Surg. 2015 Nov;41(11):1249-56. doi: 10.1097/DSS.0000000000000511. PMID: 26445291.

[4] Rajput V. Transparent grid system as a novel tool to prevent wrong-site skin surgery on the back. J Am Acad Dermatol. 2019 Nov 8:S0190-9622(19)33013-0. doi: 10.1016/j.jaad.2019.11.011. Epub ahead of print. PMID: 31712173.

[5] Navarrete-Dechent C, Mori S, Cordova M, Nehal KS. Reflectance confocal microscopy as a novel tool for presurgical identification of basal cell carcinoma biopsy site. J Am Acad Dermatol. 2019 Jan;80(1):e7-e8. doi: 10.1016/j.jaad.2018.08.058. Epub 2018 Sep 20. PMID: 30244067.

[6] Lichtman MK, Countryman NB. Cell phone assisted identification of surgery site. Dermatol Surg. 2013;39(3 Pt 1):491–2.

[7] Nijhawan RI, Lee EH, Nehal KS. Biopsy site selfies—a quality improvement pilot study to assist with correct surgical site identification. Dermatol Surg. 2015;41(4):499–504.

[8] Highsmith JT, Weinstein DA, Highsmith MJ, Etzkorn JR. BIOPSY 1-2-3 in Dermatologic Surgery: Improving Smartphone use to Avoid Wrong-Site Surgery. Technol Innov. 2016;18(2-3):203-206. doi:10.21300/18.2-3.2016.203

[9] DaCunha M, Habashi-Daniel A, Hanson C, Nichols E, Fraga GR. A smartphone application to improve the precision of biopsy site identification: A proof-of-concept study. Health Informatics J. 2020 Mar 16:1460458220910341. doi: 10.1177/1460458220910341. Epub ahead of print. PMID: 32175791.

[10] Timerman D, Antonov NK, Dana A, Gallitano SM, Lewin JM. Facial lesion triangulation using anatomic landmarks and augmented reality. J Am Acad Dermatol. 2020 Nov;83(5):1481-1483. doi: 10.1016/j.jaad.2020.03.040. Epub 2020 Mar 25. PMID: 32222445.

[11] Mcginness, J.L. And Goldstein, G. (2010), The Value of Preoperative Biopsy-Site Photography for Identifying Cutaneous Lesions. Dermatologic Surgery, 36: 194-197. https://doi.org/10.1111/j.1524-4725.2009.01426.x

[12] Johns Hopkins Department of Dermatology Pre-Procedure Image Capture Tips. Acquired from Dr. Ashley Antony on 2021 Feb 18.