

Student: Yachi Chang

Term: Fall 2015

Lab: Camera Culture, MIT Media Lab

Supervisor: Dr. Pratik Shah

Virtual Reality in Healthcare

6.UAP Final Report

I. Introduction

In healthcare, to receive treatment, a patient often has to go through a series of diagnostic tests prescribed by the doctor. Often, this prescription and the eventual diagnosis depend heavily on the doctor's experience, expertise, and intuition. This transition can be inefficient at best, or, in the case of complex diseases, dangerous at worst.

In an attempt to improve the objectivity of doctors' judgements, we notice that doctors, upon receiving the test results, often have to manually parse through these data, and the sheer amount and the disorganized nature often lead them to not considering all the information at hand. To alleviate this problem, specifically with visual data like x-rays, we envision it would be useful to create a tool that can process these information, automatically locate these "problem spots", and even point the doctors to those spots on the patient directly.

The Oral Imaging project at the MIT Media Lab Camera Culture group aims to create such a tool for oral diseases, such as gingivitis and oral cancer. Given test results about where patients potentially have indicators, such as dental plaques and caries, we want to label these spots directly onto all the images that the dentists have obtained of their patients' mouths and also, via augmented reality, render these spots directly onto the live feed video when the dentists observe patients through a video feed.

Prior Work

Currently available tools, such as Soprocure, allow doctors to identify dental plaques and caries by shining various wavelengths of light onto the patients' teeth. Since plaques and caries react to different wavelengths than healthy teeth, doctors can visually identify these problem spots.

Doctors can also use Soprocure to capture oral images. However, the only way to connect all these images into a comprehensive assessment of the patient's mouth is through the doctor, as there are no existing tools to simulate or visualize what the doctor has constructed in his or her mind. In other words, given a set of images, we currently cannot compute how they are in relation with each other without an human eye.

Several components of the Oral Imaging initiative has also been developed prior to the proposal of this project, and some are being used in combination with Soprocure. Namely, these modules include:

1. *Intraoral camera*: A preliminary image capturing device that allows dentists to take detailed pictures of their patients' mouths.

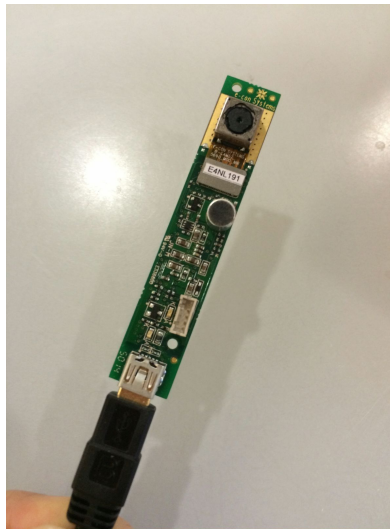


Figure 1: The camera used in this project.

2. *Image processing algorithm*: An algorithm capable of processing oral images and labelling features such as dental plaques, caries, and gingivitis. Below in Figure 2, we see that the plaques on the teeth are labelled green by the algorithm.

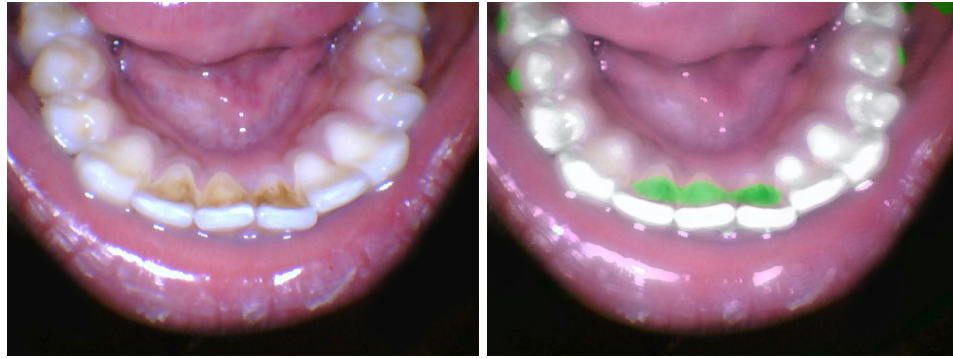


Figure 2: The raw (left) image and the processed, labelled image (right) of a patient's mouth.

II. Goals

Our goal, hence, is to create a platform that can automatically persist these disease features across all images and videos taken by the dentists. Given an image, we want to deduce exactly where in the mouth the dental plaque or caries is located, and with this information, given any other image or video of the mouth, we will be able to render that feature if it is within the frame. This technology will allow us to construct and visualize the condition of a patient's entire mouth without human intervention, making the diagnosis more objective and consistent.

Our solution is to create an Android application that interfaces both the data collection (image capture) and the data analysis (image processing, feature persisting, and rendering) portions of the Oral Imaging initiative. The application will create an intuitive workflow for dentists to capture images of their patients' mouths, browse through these data, which will be annotated with features like dental caries and plaques, and see these features rendered in a live feed.

To support this, the application will have the following functionalities:

1. *User can capture images via a guided interface.* The user can use the intraoral camera, which should plug into the Android device, and take picture of a patient's mouth. The application will instruct users on how to position the camera so the users can capture a set of pictures that comprehensively cover all parts of the mouth.
2. *Application will process the images through the feature labelling algorithm.* After the user captures the images, the images will be fed into the algorithm for processing. Features like dental plaques and caries will be identified, labelled, and stored.
3. *User can view through these data effectively.* The user will be able to browse through images in an organized fashion, and toggle between viewing the labelled or the raw versions of the images.
4. *Application can render these identified features onto a live feed video.* This is the augmented reality portion of the project. The application, while the user looks at the live camera feed through his or her phone, tablet, or VR headset, will label these identified features when they come into the user's view.

III. Design

In this section, we describe the designs behind the application, from the higher levels to the lower, namely: the user interface, the system design, and the application infrastructure.

User Interface (UI)

The application has three different modes the user can interact with:

1. *Capture mode*, where the user uses the intraoral camera to capture comprehensive images of the mouth

2. *Browse* mode, where the user browse through the raw and processed data in an organized way
3. *Live Feed* mode, where the user views the world via the live camera and the application labels the previously processed features as they come into view.

We also plan to use two viewing mechanisms in the application. The *capture* and *browse* modes are accessible through a flat Android device, like a phone or a tablet. The *live feed* mode, on the other hand, will be accessible via a virtual reality (VR) headset. For this project, we choose to use Google Cardboard as our headset. Table 1 shows the UI we envisioned for *capture* and *browse* modes. The *live feed* mode will simply show the feed from the camera.

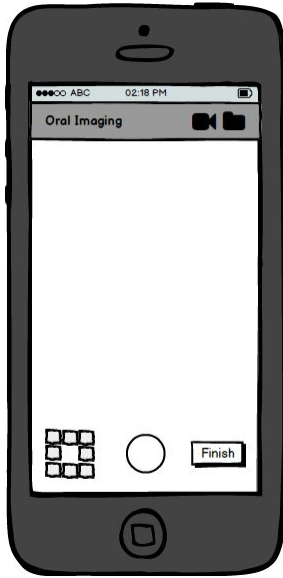
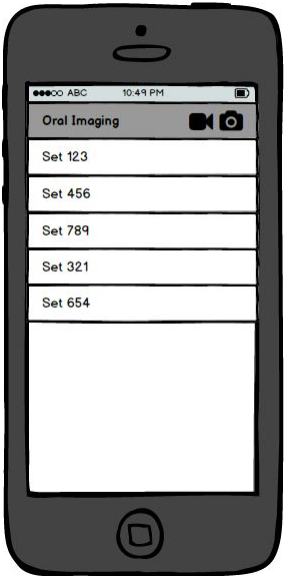
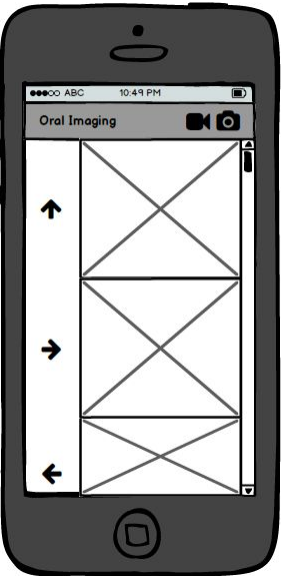
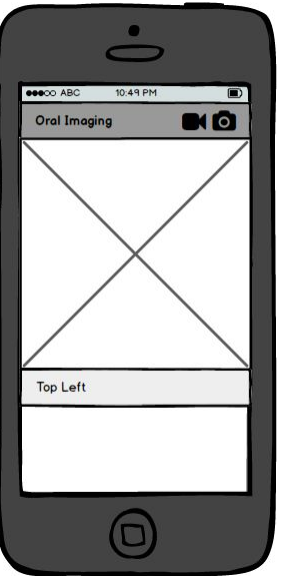
Capture Mode	Browse Mode Browse Sets	Browse Mode View Set	Browse Mode View Image
			

Table 1: UI mock-ups for the different modes of this app

Capture Mode

In the *capture* mode, the user will need to plug the intraoral camera via an OTG cable into phone. The application will display the view of the camera, and by clicking on the middle circle, the user can capture the image.

To comprehensively capture images of the entire mouth, the user must capture a “set” of 8 images, one of each orientation: top left, top center, top right, center left, center right, bottom left, bottom center, and bottom right. One of the eight squares in the bottom left corner will be highlighted to represent which orientation the currently captured images will be labelled as.

When the user has finished capturing all images, he or she can click on the finish button to indicate that the current set of 8 images should be stored into the phone. The user can click on the video icon on the action bar to switch to the *live feed* mode, or the folder icon to switch to the *browse* mode.

Browse Mode

After the user finishes capturing all images, he or she can return to the *browse* mode of the application to view the set just captured or any previously stored set. The mode will start off listing all the sets available. The user can click on a set to view all of the 8 images, and he or she can click on any of these 8 images to view the specific image. The user can also left or right swipe on the image to switch to another image in the set, or double tap on the image to switch between viewing the raw or the labelled version of the image.

We note that the *browse* mode will be the default mode when the application is first started. The user can click on the video icon on the action bar to switch to the *live feed* mode, or the camera icon to switch to the *capture* mode.

Live Feed Mode

The headset will simply display the views of the source cameras (see *Implementation, Live Feed Mode* section for details), and when the previously identified features come into view, they will be labelled. The user can click on the trigger, or the magnet in the case of Google Cardboard, to switch back to *browse* mode.

System Diagram

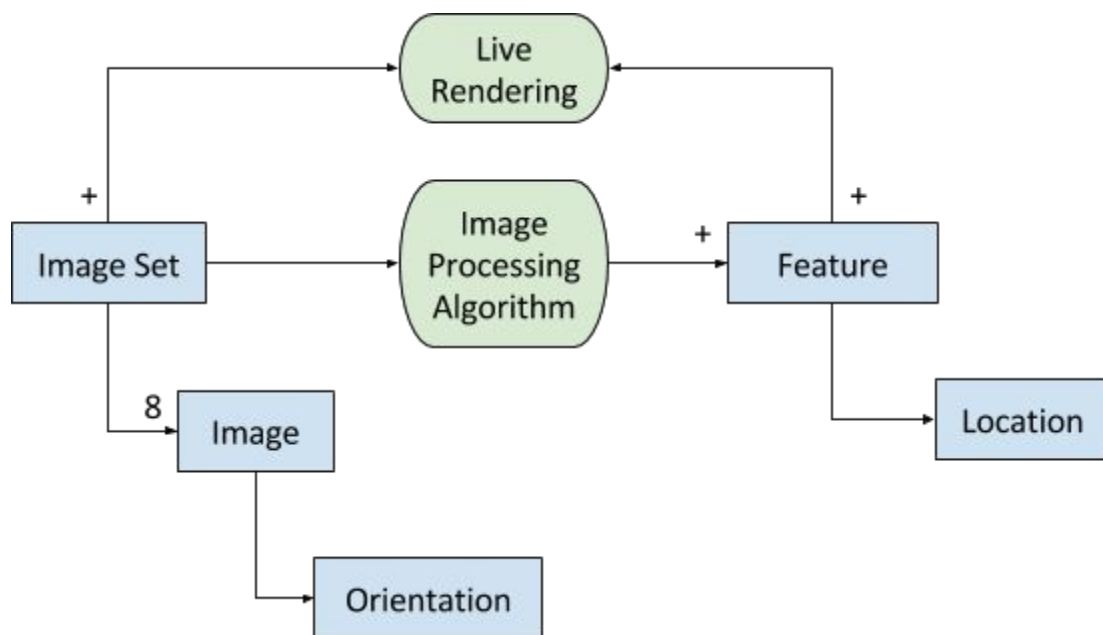


Figure 3: System design for the application.

The system design for our application is outlined above. The light blue rectangles represent object concepts, while the light green represent abstract modules of the application. We explain each of the concepts below:

1. Image: An image is simply a picture captured during the *capture* mode of the application. Each image is additionally labelled with an *orientation*.

2. Orientation: An orientation corresponds to how the camera is positioned when an image is captured and which areas of the mouth is covered by the image. Each image will have one orientation, and the possible eight orientations are: *top left, top center, top right, center left, center right, bottom left, bottom center, bottom right*.
3. Image Set: An image set is a group of 8 images, taken in short succession of each other, and each belonging to a different orientation. Hence, an image set consists of images that collectively captures images of all parts of a patient's mouth at a given time.
4. Image Processing Algorithm: After an image set is taken, it's inputted into the image processing algorithm. The algorithm will process the images in combination with their orientations, and outputs a list of features, namely spots in a patient's mouth that a doctor should pay attention to.
5. Feature: A feature is a clinical spot marked on the image that a doctor should pay attention to, namely what the algorithm identifies as dental plaques or caries
6. Location: Each feature has a location. These are simply stored as bounding boxes in relation to the image they are found in.
7. Live Rendering: This algorithm takes a list of features and outputs the 3D positions of these features with respect to the camera. And using stereo vision and triangulation techniques (see *Implementation, Live Feed Rendering*), we can determine how each of the 3D positions projects onto the 2D feeds of the headset's source cameras.

Application Infrastructure

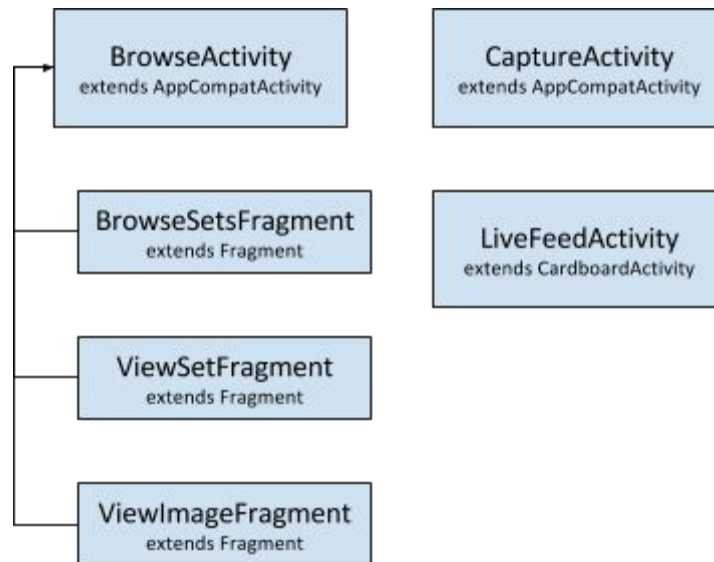


Figure 4: Infrastructure of the application

The application is implemented using the Android SDK and the Google Cardboard SDK. Each mode will be abstracted out as an Android activity. Within the *browse* mode, each “level” of view, whether browsing through sets, viewing a particular set or image, will be abstracted out into an Android fragment used in the *BrowseActivity*. We note that since *live feed* mode needs to be available on and only on Google Cardboard, its Activity must extend *CardboardActivity* from the Cardboard SDK instead.

The advantages of this design is that every mode is modular in implementation, and it would be easy to add more modes later. However, the transition between Android activities can be slow, potentially disrupting the user experience.

IV. Implementation

The modular infrastructure design of the application allows us to develop each mode in parallel. Here we will describe the implementation challenges of each module.

Browse Mode

The browse mode is first set up so that we can view the images taken, and the layout follows closely with our UI designs above. The file structure system we use to store images is as follows. The application will create an “Orallmaging/” folder in the root directory for the app. For each set of images captured from the *capture* mode, a folder named “Set_xxxxxx_yyyyyy” is created, where “xxxxxx” is replaced with the date the set is taken and “yyyyyy” is replaced with the time the set is finished. Inside the folder, eight different folders, named “TL”, “TR”, “TC”, “CR”, “CL”, “BL”, “BC”, and “BR”, one of each of the eight orientations are created.

Inside each of these eight folder, the raw image is stored as “original.bmp”, and the processed image, with features labelled, will be stored as “rendered.bmp”. For example, a *top center* raw image may have the file path “Orallmaging/Set_121415_211632/TC/original.bmp”. This structure makes it easy to retrieve images the application needs to display.

Capture Mode

The capture mode is set up next so we can start capturing the images. The intraoral camera is plugged into the phone, and its input view is displayed on the application. To create the guided interface we envisioned, we decide to incorporate an IMU (inertial measuring unit) with the camera.

We can calculate how much the camera is tilted from the IMU input. This angle helps us determine which orientation the camera is currently viewing in. For example, treating the positive x-axis as 0 degrees, we know the center left orientation must cover -22.5 to +22.5 degrees and the top left orientation will cover +22.5 to +67.5 degrees.

If the camera’s field of vision doesn’t cover that of an entire orientation, the application will show how much the user need to turn the camera to cover the nearest orientation. The

application also shows and keeps track of the images the user has taken, and the user is not allowed to “finish” a set until all 8 images have been captured.

Image Processing Algorithm

An image processing algorithm has been developed by other members of the application. Currently, it will process the images captured and highlight, in green, the parts that have been identified as dental plaques or caries. We altered this output to fit our application. Instead of just outputting a labelled image, we also extract and store the location of each feature as a bounding box within the image it is located in.

Live Feed Mode

The live feed mode is available only when the user is wearing a VR headset, specifically Google Cardboard for this project. Two small cameras are strapped onto the headset, and they are angled such that one represent the left eye while the other represent the right eye camera. The feed from each camera will be displayed in the the corresponding eye. As the doctors observe the patient through the headset, the features will be highlighted as they come into view.

We also ask patients to wear a brace while the doctors use this mode to observe the patients. The brace should have distinct, identifiable markers that can be easily spotted by the rendering algorithm. The rendering techniques and the motivation behind this setup are described in more details in the next section.

Live Feed Rendering

We employ computer stereo vision techniques to enable live feed rendering, specifically stereo reconstruction. The concept (Figure 5) behind this approach is that given two frames, one from the left eye and the other from the right eye, how the two cameras are positioned, and the 2D pixel location of a feature on each of these frames, we can use triangulation to determine the 3D position of the feature in world coordinates.

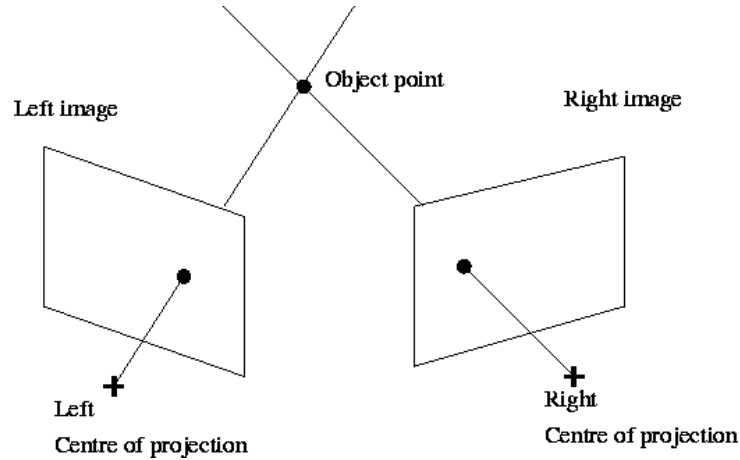


Figure 5: Illustrating the intuition behind stereo triangulation

Hence, given a 3D coordinate, we can project it onto the two frames using a projection matrix. This is the motivating factor behind the patient wearing a brace. By having a distinct marker on the brace, we can fixate a world point as the origin, find its 2D location on the two frames via feature detection, and obtain this projection matrix using matrix transformation.

Now, our goal is to render features, or bounding boxes, that we have identified from previously captured images onto these two frames. In other words, the goal is also to translate these bounding boxes into world 3D coordinates. Once we get these 3D coordinates, we can transform them via the projection matrix above and obtain their proper 2D locations on the two live feed frames. This will allow us to render these features onto the live feed video.

The question that still remains is how we can transform these bounding boxes into 3D coordinates. Due to time constraints, this step of the process has yet to be detailed. However, we envision that we can use the IMU data we stored when the image is captured to compute the camera's field of view and obtain the subset of world coordinates that the original image contains.

An idea that can help us compute the world 3D coordinates of the feature more accurately is as follows. We can have the patient also wear the same brace when the images are captured. If

both a marker and a feature are both within the image frame, we may be able to estimate the real world distance between the marker and the feature and use that to compute the world coordinate of the feature. However, for a more accurate estimation, we may even consider incorporating another camera onto the intraoral camera for stereo vision. This allows us to easily calculate the world coordinate of the feature relative to the intraoral camera, and it would be a relatively simpler task to find the transformation between the intraoral camera coordinate system and the VR headset coordinate system.

Once we obtain the transformation matrices we describe above, we can easily translate these bounding boxes into world 3D coordinates for the VR headset to render onto the application live feed.

V. Conclusion

By creating the Android application outlined above, we enable dentists to collect and visualize comprehensive, annotated images of their patients' mouths. The automatic feature rendering allows dentists to see these problems spots like dental plaques and caries clearly and make diagnoses more consistently and objectively.

We also note that this workflow of doctors collecting data via capturing images, the application identifying features via image processing, and the doctor viewing these features rendered onto other image and video sources, including a live feed, can be easily applied to other diagnostic processes. For example, an x-ray scan for a broken bone can be processed and the exact spot of the injury labelled. Now when a doctor views the patient via a VR headset, the location of the broken bone can be rendered onto the live feed, and knowing where the breakage has occurred exactly can help doctors diagnose and treat the patients better.

The framework and platform we have created in this project are beneficial for the oral imaging diagnostics as we aimed for, but can also be extended to any other type of data collection and

visualization workflow. The application is certainly set up to be further developed into a larger medical data collection and analysis platform that can potentially benefit healthcare workflows in hospitals all over the world.

VI. Acknowledgements

I would like to thank Dr. Pratik Shah, Mrinal Mohit, Shantanu Sinha, and Hyunsung Park, all part of the Oral Imaging initiative, for supporting me in this project. The Camera Culture group were instrumental in providing me with the funding, workspace, mentorship, and materials required to implement my designs.

VII. Reference

Rechmann, P., Shasan W. Liou, Beate M. Rechmann, and John D. Featherstone. "SOPROCARE - 450 Nm Wavelength Detection Tool for Microbial Plaque and Gingival Inflammation: A Clinical Study." *Lasers in Dentistry XX* (2014): n. pag. Web.

UW CSE Vision Faculty. "Lecture 16 - Stereo and 3D Vision." Web. 9 Dec. 2015.
<<https://courses.cs.washington.edu/courses/cse455/09wi/Lects/lect16.pdf>>.

Zisserman, A., and S. Lazebnik. "9 - Stereo Reconstruction." Web. 9 Dec. 2015.
<http://cs.nyu.edu/~fergus/teaching/vision/9_10_Stereo.pdf>.