**Robotics Dog**



**Undergraduate Study EE 4990**

**Advisor:**

Curtis Y. Wang

**Members:**

Christopher Morales

Anthony Reyes

**Date of Submission:**

May 19, 2022

Table of Contents

[Abstract 3](#_Toc121405488)

[1.0 Introduction 4](#_Toc121405489)

[1.1 Problem and Background 4](#_Toc121405490)

[1.2 Project Objective and Scope 5](#_Toc121405491)

[1.3 Society Impact and Factors 5](#_Toc121405492)

[2.0 Approach 6](#_Toc121405493)

[2.1 Design Process 6](#_Toc121405494)

[2.2 Organization and WBS 7](#_Toc121405495)

[2.3 Deliverables 9](#_Toc121405496)

[3.0 System Overview 9](#_Toc121405497)

[3.1 Requirements 9](#_Toc121405498)

[4.0 Progress 13](#_Toc121405499)

[5.0 Summary and Conclusions 24](#_Toc121405500)

[7.0 References 25](#_Toc121405501)

# **Abstract**

The purpose of this project is to create a tour bot that will guide Cal State LA guests to locations on campus. The tour bot will be given a dataset of the school building’s floor plans and a predetermined path to follow. The bot will be a mobile platform that will incorporate computer vision to identify obstacles and facilitate localization. The collection of sensors used to navigate the environment may also incorporate LiDAR, ultrasonics, cameras, IR proximity, and a GPS module that will focus on gathering data of the surrounding environment. While the internal sensors will consist of IMUs, position sensors, load sensors, and current sensors. The sensors will report to slaved microcontrollers that will feed into the microprocessor. Ubuntu is currently the planned operating system to control high level algorithms. The control software will dictate commands to the tour bot actuators consisting of servo, stepper, and brushless motors. Electronic subsystems will be used to power the motors and to regulate the power supply. The majority of the tour bot chassis will be a 3D printed, built around a load bearing internal frame.

# **Introduction**

## Problem and Background

In recent years, technological advancements have been reshaping various aspects of our lives, and the education sector. One area where innovation has made significant strides is campus tours. Traditional campus tours, led by human guides, are gradually being complemented and even replaced by robotic counterparts. Robotic dog, in particular, have emerged as a novel and engaging means of touring guests and potential students around campus. The robotic dog will be able to explore the profound impact and potential benefits of incorporating robotic dogs into campus tours.

When visitors arrive on campus, they often expect an experience that stands out from the ordinary. By introducing robotic dogs as a tour guide, California State University, Los Angeles, can capture visitors with a unique and memorable encounter. The mere presence of a robotic dog creates an immediate sense of curiosity and excitement, setting the stage for an engaging tour experience that will promote the ECST department of Cal State LA.

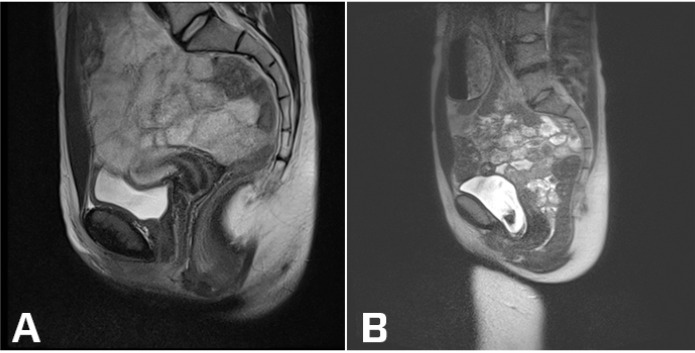
Integrating robotic dogs into campus tours serves as a powerful showcase of an institution’s technological prowess. By featuring these sophisticated machines, universities demonstrate their commitment to staying at the forefront of innovation. Where the robotic dog can be programmed to interact with visitors on a personal level. Some examples are greeting guests, providing campus information, sharing interesting facts, or if possible, answering questions. Visitors can engage in conversations, seek specific information, and receive real time assistance throughout the tour. Implementing predifine routes, ensuring a consistent and accurate tour experience for all visitors.

1. Project Objective and Scope

Develop a program tool to analyze medical images and transport parameters creating a 3D model into FreeCAD to generate patients' anatomy. To create a 3D Pelvic Floor System, the MRI images will go through multiple transformations, elimination, and modifications in order to determine the wanted organ. For team 10, the main goal is to determine a vast number of pixels that can be represented as a blob or will be called a cavity. Where each cavity will have its own correlated ellipse, either the ellipse taking unwanted area or losing area in the blob. Each ellipse will have its own mathematical expression that can be extracted from and sent to FreeCAD software to render and simulate the model.

## Society Impact and Factors

Pelvic organ prolapse describes the falling of pelvic organ from the placement considered normal to placement within the vagina or outside. The pelvic prolapse is most common on women with a weak pelvic floor system. There are multiple dangers of organ prolapse such as frequent urinary tract infections, pain in lower abdomen, heaviness in lower abdomen or lower back, and tissue outside of pelvic organ. Figure 2 contains MRI of a patient’s pelvic floor at rest compared to organ prolapse during strain [3]. It is estimated that 1/3 of the women have or will have a prolapse and half of the women suffering from a prolapse will have surgery treatment. Every patient has a different anatomy structure and different pathologies while the surgical treatments are applied without taking into consideration the patient specificities. A way to improve the quality of the treatment is to run some numerical simulation that would help to differentiate the benefit of each surgical technics to a specific patient. The society impact would then be, once the numerical model of simulation is working, to improve the quality of the treatment based on patient specific anatomy, that we are creating.



**Figure 2:** Sagittal View of pelvic floor at rest vs pelvic organ during strain

# **Approach**

## Design Process

A picture containing text, screenshot, rectangle, design

Description automatically generated

**Figure 3:** Block Diagram of Our Process

# **System Overview**

## Requirements

To meet the electrical and software requirements of the robotic dog, several advanced technologies will be utilized, including LiDAR, computer vision, an inertial measurement unit, local host Bluetooth, multiple microcontrollers, a camera, and GPS. The integration of these components will enable the robotic dog to navigate its environment effectively and perform various tasks.

The LiDAR sensor will play a crucial role in detecting surrounding objects and accurately measuring their distances. By analyzing the reflected laser signals, the robotic dog can create a detailed map of its surroundings, allowing it to avoid obstacles and navigate safely.

Computer vision, in conjunction with a camera, will provide additional perception capabilities to the robotic dog. It will enable the dog to identify vertical walls and recognize objects that may not be detected by the LiDAR sensor due to lighting conditions or limited range. This combination of technologies ensures reliable object recognition and enhances the dog's ability to interact with its environment.

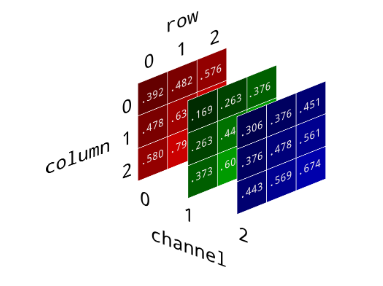
To ensure accurate localization, the robotic dog will utilize computer vision techniques in combination with a GPS module. By comparing visual landmarks from the camera with a preloaded map of the school, the dog can determine its precise location, including room numbers, floor levels, and overall school layout. This integration of computer vision and GPS ensures the robotic dog has an accurate understanding of its position within the school environment.

Additionally, the local host Bluetooth functionality will enable seamless communication between the robotic dog and external devices without the need for physical connections. This capability allows for easy access to raw data collected by the dog, facilitating remote monitoring and control.

By incorporating LiDAR, computer vision, inertial measurement units, local host Bluetooth, multiple microcontrollers, a camera, and GPS, the robotic dog will possess a comprehensive set of capabilities to navigate the school environment efficiently and interact with its surroundings. These advanced technologies ensure that the robotic dog operates with precision, adaptability, and autonomy, making it a valuable tool for various applications, including campus tours and educational demonstrations.

## Computer Vision

To get a better understanding of Computer Vision, it is better to understand how an image works despite its looks. Every image consists of three matrices in RGB format that makes it look colorful. Where, every element or cell of a matrix will have a value range of 0 to 255, which will be presented as the color of the pixel. Where combining the three matrices will result into a colorful image

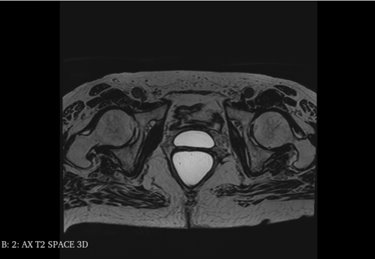


**Figure 1:** RGB Image in matrices format

Furthermore, a video is a list of images in time which is known as a video. In computer vision, this is important due to an image itself will have a vast amount of data and for the robotic dog to determine an object and especially determine shadows in an image.

The video will go through multiple transformations, elimination, and modifications in order to determine the wanted object and color detection. For the Robotic Team, the main goal is to determine a vast number of pixels that can be represented as a blob or will be called an object. Where the object itself will go through the process

One of the necessary implementations for detecting an object from an image is called grayscale. Grayscale is a transformation from RGB, three matrices, that will be converted to one matrix. For the project, reducing as many channels as possible will help detect the wanted image, as the following figure 5 shows.



**Figure 4:** RGB MRI Image in Sagittal view of the women pelvic cavity

From equation, where R presents red pixels, G is green pixels, and B is blue pixels according to OpenCV [2]. The equation shows the process of RGB channels being reduced into one channel using an OpenCV library called cv2.cvtColor().



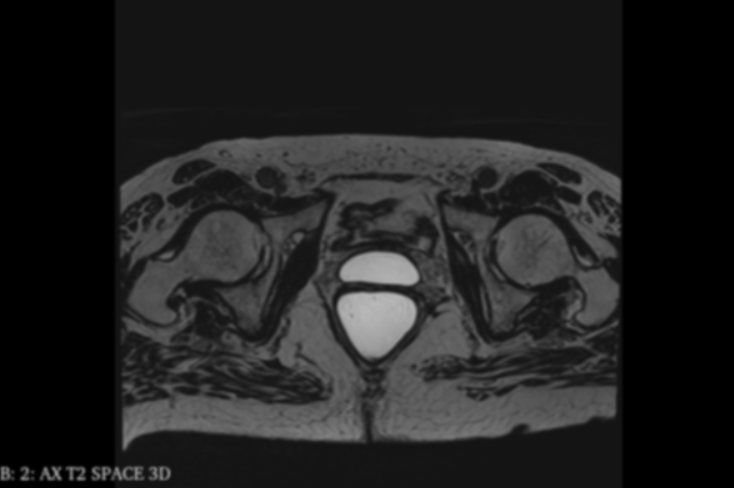
**Figure 5:** Gray Scale MRI Image in Sagittal view of the women pelvic cavity

Where most libraries from OpenCV work best with one channel of an image compared to a 3-channel image. Looking at Figure 5, the image can be assumed gray scaled however the Computer Vision or the computer is able to determine that it is an RGB image that looks like a gray image.

When applying grayscale to an image Fig 5, all three matrices will be multiplied to some fixed number that has been calculated by the library then get the sum of the pixels. Note that the constant values cannot be changed from OpenCV and if all three matrices exceed 255 (the max color pixel) then the value will be set to 255.

**Gaussian Blur:**

One image processing that is widely used is called Gaussian Blur. Where Gaussian Blur is meant to blur an image and is typically used for image noise reduction or reduced detail. Which can be considered a low pass filter, high pass filter, or bandpass filter depending on the pre-configurations for the OpenCV library. Where every blur is dependent on one's matrix, equal size of columns and rows that have one's inside the matrix or known as the kernel. Depending on the size of the kernel, it will determine the smoothness or blurriness of the image. The smaller the kernel, the less blur the image is while if the kernel is bigger than the blur of the image will increase.



**Figure 6:** Gaussian Blur MRI Image in Sagittal view of the women pelvic cavity

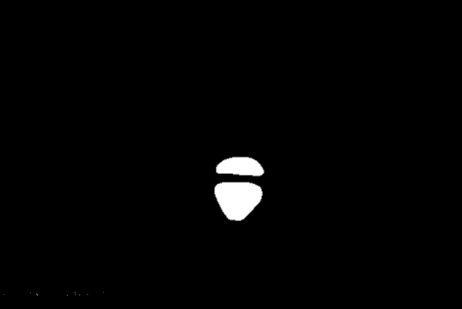
In figure 6, the image lost most of the tiny noise or the small ambient white pixels that was presented in f. The functionality of Gaussian blur is dependent on the size of the kernel that will be going left to right on the image. That is affecting the values in the element of the matrix to get a new image/matrix. Applying Gaussian blur will eliminate false positives to determine and making the bright pixels blurry.

**Image Thresholding:**

The previous few sections led to one of the important implementations of Computer Vision. Determining an image in binary from values (in color, pixels) 0 - 255 to 0 - 1. Where threshold needs two inputs where if the value is smaller than the threshold set the pixel to 0 (black) else set it to a maximum value, 255 (white). It should be stated from OpenCV that there are multiple functionalities for threshold however in our purpose, the threshold will be used for binary response, with no gradient or hue involved in an image. For the reason being, if the image was colorful hue and gradient will be in consideration.

*Ret, threshold = cv2.threshold(blur\_image, 127, 255, cv2.THRESH\_BINARY)*

The first input of the code requires an image, which is a blurred image. While the second input is the condition for the function to determine the output of each pixel in an image. Furthermore, the best numerical number that has worked best for the MRI images is 127. If any number before 127 will make the pixel a zero else the pixel will become 255. While the last output will determine the method of each pixel output, where the threshold can give a binary output or a gradient output. As the figure down below shows:



**Figure 7:** Threshold MRI Image in Sagittal view of the women pelvic cavity

In figure 7, most of the pixels are eliminated by thresholding. If the image was not blurred, then most of the image would have remained in the thresholding process. Due to Gaussian blur divides a certain section evenly across the image which results in lowering the value of the pixel.

**Contours Approximation:**

One of the most important concepts of identifying an object for image processing is called contours. Where contours are defined by joining all the points along the boundary of an object in an image. Where contours need a binary image to detect an object's edges or curvatures. If a multiple-channel image or a wide range of values of pixels are presented in an image, then the contours will not be able to determine any object in an image. As the following code will be presented:

*Contours, hierarchy = cv2.findContours(threshold, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)*

The first input of the parameter requires a threshold image but the second and third parameters have multiple options to determine how to define a contour. The second parameter will determine the extreme outer flags which will affect the output variable called hierarchy. The third parameter will be to determine which method to use to determine an object known as contours. Where contour is a continuous point along the boundary of an object which is the outline of the object. Using *cv2.CHAIN\_APPROX\_SIMPLE*, will determine the number of sides of an object and store the corner coordinate points.

# **7.0 References**

[1] “Basic Operation on Images”, OpenCV. Available: <https://docs.opencv.org/3.4/d3/df2/tutorial_py_basic_ops.html>

[2] ]“Color conversions,” OpenCV. [Online]. Available: <https://docs.opencv.org/3.4/de/d25/> imgproc\_color\_Conversions.html [Accessed: 06-Dec-2022]

[3] *MRI in pelvic organ prolapse*. Aarhus Universitetshospital - English. (2020, December 8). Retrieved December 7, 2022, from <https://www.en.auh.dk/departments/department-of-radiology/reseach-areas---new/pelvic-mri/mri-in-pelvic-organ-prolapse/>

[4] “3D Slicer Image Computing Platform.” *3D Slicer*, www.slicer.org/. Accessed Dec. 2022.

[5] G. Jocher, “Ultralytics/yolov5: Yolov5 🚀 in PyTorch > ONNX > CoreML > TFLite,” *GitHub*. [Online]. Available: https://github.com/ultralytics/yolov5. [Accessed: 04-Dec-2022].

[6] Chablani, Manish. “Yolo - You Only Look Once, Real Time Object Detection Explained.” *Medium*, 31 Aug. 2017, towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006.

[7] “Make sense,” *Make Sense*. [Online]. Available: https://www.makesense.ai/. [Accessed: 04-Dec-2022].

[8] Lebiedzinski, Peter. “A Single Number Metric for Evaluating Object Detection Models.” *Medium*, 24 June 2021, towardsdatascience.com/a-single-number-metric-for-evaluating-object-detection-models-c97f4a98616d.

[9] Bhandari, Aniruddha. “Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023).” *Analytics Vidhya*, 1 May 2023, www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/#:~:text=A%20Confusion%20matrix%20is%20an,by%20the%20machine%20learning%20model.

[10] Solawetz, Jacob. “How to Train Yolov5 on a Custom Dataset.” *Roboflow Blog*, 10 May 2023, blog.roboflow.com/how-to-train-yolov5-on-a-custom-dataset/.

[11] Andrew Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. "Direct least square fitting of ellipses". IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5):476-480, 1999