

# CS231A Project Midterm Report: Pool Safety System

Ben Gilboa  
Stanford University  
gilboab@gmail.com

## Abstract

*This midterm report reviews the problem that the project is intended to investigate and solve. It explains the methods that were tried already to address some of the phases of the problem and the ones still planned for further investigation. It details the progress and efforts done so far and the steps still planned.*

## 1. Introduction

The project goal is to explore and design key portions of a safety and maintenance system for private pools. Owning a pool in the back yard is a source of joy and fun during the summer but it is also a source of concern throughout the year. The worst concern is having a kid drowning in the pool when it is not monitored. The most common solution to this problem today is building a fence around the pool with a gate that a child can't open. Building these type of fences cost between \$1000 and \$2000 and it has negative appearance on the look of the back yard. Other solutions are to cover the pool or to have an electronic detection system based on computer vision or infrared rays. All of those solutions as of today are priced way above \$1000 price point. Other concerns that pool owners often face are related to pool maintenance. Small animals often fall into pool and end up dead in the pool skimmers. Ducks sometimes decide to make home in certain pools and cause a lot of mess. Excessive amount of leaves and branches often falls from trees during autumn and winter and overload the pool cleaning system cause equipment wear out. There are many other bad things that can happen and result by pool owner paying hundreds or thousands of dollars.

The objective of this project is to explore and design portions of a low cost computer vision based system that can potentially address most of the concerns raised. The main idea is to explore a system that is based on one or two cameras that are positioned in the back yard outside of the pool. The envisioned system can provide online stream video of the pool. It is continuously detecting certain events based on computer vision techniques. Once an event is detected the system can either notify the owner by sending

a notification to owner phone or turn in an alarm in case of detecting a life hazardous event. Building the entire envisioned system is outside the scope of this project. This project focus is the detection portion of objects in the pool.

## 2. Problem statement

Given an image of a pool the main problem is detecting objects in the image and deciding whether a certain object is inside or outside the pool. Solving this problem in a robust and reliable way is the main focus of my project. There are several key steps to achieve this goal.

Given an image that contains a swimming pool, the first step is to detect the boundary of the pool as a polygon in the image. The second step is to detect objects in the image. The third step is to detect for each object whether it is inside the pool or not.

The dataset that I am using for this project is constructed from images found in the internet that contain outdoor swimming pools with or without humans in them. For the third step where I intend to explore different methods to detect whether a specific object is inside or outside the pool I might try stereo methods that we have learnt in the class and for that I can take images of a specific pool myself.

## 3. Technical approach

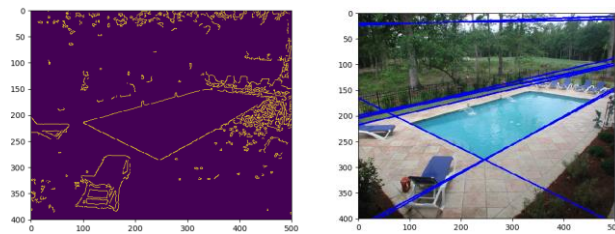
### 3.1. First step – detect pool as polygon

For this step I have used both techniques from the class and other techniques that I explored. The objective is to have a solid algorithm that can detect the boundaries of a pool as a polygon. In phase one the algorithm needs to work well for empty pool and in phase two it needs to perform for pools with objects. In the lack of enough annotated images for pools, my dataset is taken from the internet and tested manually by comparing the boundaries of the polygon found by the algorithm to the boundaries of the pool. As a simplification for this project I started by exploring rectangle shaped pools.



**Figure 1: example of rectangle pool**

The first method explored to detect the boundaries of the pool was using Canny edge detector. I applied different gaussian filters and different configuration of the canny edge detector from OpenCV. Using the output of the Canny edge detector I tried to detect the straight lines that construct the boundaries of the pool using Hough Transform. The edge detector performed as expected but it was not easy to find the right setting for the Hough Transformation and get definite lines.



**Figure 2: Canny edge detector and Hough transformation for lines**

### 3.2.

This technique does not achieve the desired result. Even for a single image it was very hard to adjust the parameters to get a decent detection of the pool area. As can be seen in figure 2, Hough Transform misses pool boundary lines and detects wrong lines.

The second method used rely on the fact that pools are usually in shades of blue. The technique mask the image with certain shades of blue as upper and lower thresholds. Then change using OpenCV threshold function to convert to binary image. Using the binary image, I searched the image for finding contours. Out of all detected contours I choose the largest one assuming the pool is the largest object in the image. The last step was to approximate the largest contour to using OpenCV ApproxPolyDP function to get a polygon with the desired number of sides. In this case since the pool was rectangular I optimized the polygon to a shape with 4 sides. The above technique works well on rectangular pool images without occlusions as can be seen in figure 3.

For pools that are not rectangular 4 sides approximation

of a polygon does not work well. In case of occlusions the result is also not satisfying as in figure 4.



**Figure 3: Pool boundary detection in the second method**



**Figure 4: 4 sides approximation to non rectangle pool**

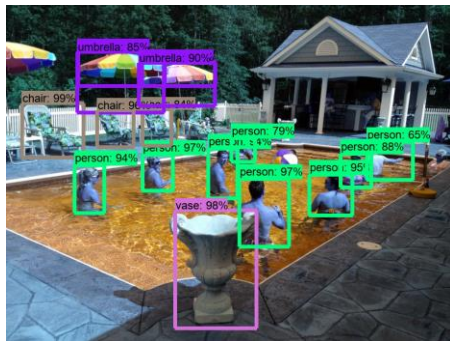
I examined other ways to use polygon approximations with more sides, use convex hull method to eliminate sharp edges and other methods. As of now, I plan to spend more time to investigate ways to achieve better detection of non rectangular pools.

### 3.3. Second step – object detection

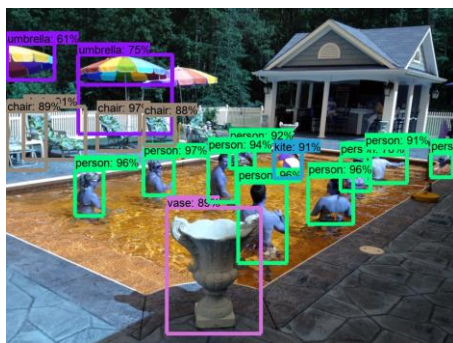
For object detection I use pre-trained convolutional neural network. I installed TensorFlow by google and started working with the object detection API. I examined how the different pre-trained models work and detect object on images of pools. The default model that object detection API tutorial is using “ssd\_mobilenet\_v1\_coco” runs very fast but its performance is not good enough. it often misses many obvious objects. It also classifies object wrong in many cases. The second model that I tried was “faster\_rcnn\_resnet50\_coco”. There is a significant improvement in the performance of this model in the expense of longer run-time. The 3rd model that I tried “faster\_rcnn\_inception\_resnet\_v2\_atrous\_coco” is much heavier than the other two but it provides very good results.



**Figure 5: object detection using  
ssd\_mobilenet\_v1\_coco model**



**Figure 6: object detection using  
faster\_rcnn\_resnet50\_coco model**



**Figure 7: object detection using  
faster\_rcnn\_inception\_resnet\_v2\_atrous\_coco model**

detected in the first step as polygon and the objects detection in the second step and build the decision algorithm whether the object is inside the pool or outside of the pool. I haven't reached to this step yet. I intend to examine simple algorithm in 2D like comparing the points in the object boundary to the boundaries of the polygon that represent the pool. If the results will not meet my expectations I intend to explore other techniques that represent the scene in 3D as been taught in the class.

### 3.4. Third step – location of object

The third step of the project is to use the pool area