Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2019

Project Title:	**Augmented Reality-assisted Human Robotic Interaction**
Student:	**Aufar P. Laksana**
CID:	**01093575**
Course:	**EIE4**
Project Supervisor:	**Dr Yiannis Demiris**
Second Marker:	**Dr Tae-Kyun Kim**

Abstract Powered wheelchairs are becoming increasingly commonplace in the modern world. However, a major issue faced by powered wheelchair users (PWUs) is navigating the device in crowded areas. Controlling the powered wheelchair in crowded areas requires increased concentration from the PWU, as people in crowds often move unpredictably, or are hidden from view due to standing behind another person or object.

This project utilizes computer vision techniques to predict the direction of travel of individuals in crowds, and implements an augmented reality system using the Microsoft Hololens that aids the PWU by displaying visual aids that indicate the motion of people. The system further aids the user by warning the user of potential collisions, allowing the PWU to make better navigation decisions. The project also explores the use of the system as a method of assitive control of the wheelchair, preventing collisions by stopping the powered wheelchair should the PWU not notice an individual crossing their path.

Contents

Chapter 1

Introduction and Requirements

1.1 Introduction

This report was written as part of the Final Year Project for the MEng Electronic & Information Engineering course. The project was supervised by Dr. Yiannis Demiris at the Imperial College London.

1.2 Motivation

Chapter 2

Background

This project is focused on computer vision for detecting and tracking humans in the surroundings, estimating their trajectories and distance from the PWU, the reactive control systems that prevent collisions with the detected objects as well as the augmented reality display to provide visual cues to the PWU.

2.1 Human Detection

Human detection is a subset of the classic computer vision problem of object detection. In order to develop an augmented reality system that will help PWUs to navigate in public spaces, it is essential for the system to be able to discern humans from the surroundings.

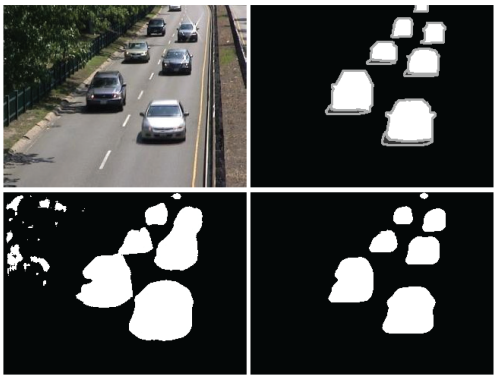
2.1.1 Definition of Requirements

The problem arises in crowded areas, whereby individuals are occluded by other people or objects in front of them, leaving only certain body parts visible. As such, we began our research with the problem of being able to detect people in images where identifiying parts of the body are not always visible.

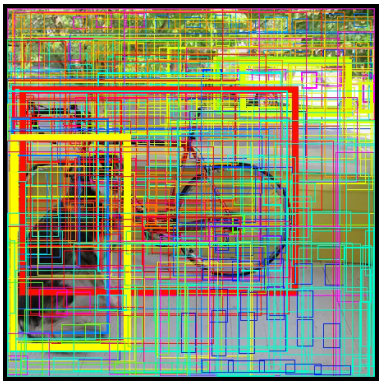
2.1.2 Review of Existing Methodologies

A related field of research is that of people counting and human detection in visual surveillance in public areas. Where the problem differs is that surveillance benefits from being able to rely on cameras with a good view of the crowd from above, whereas for a PWU, the camera will not have as high of a vantage point, making detecting every single individual in a crowd impossible.

Despite the disadvantage, similar techniques can be used to detect humans in video. Most methods can be classified into two categories [1]. The first technique, foreground detection, attempts to model the background of an image and then detect the changes that occur between frames. The second category involves exhaustively searching the image with a scanning window, and deciding if each window can be classified into a human shape.



(a) Foreground Detection [2]



(b) Scanning Window [3]

Figure 2.1: Comparison of Foreground Detection and Scanning Windows

Foreground Detection

Background subtraction is a widely used approach for detecting moving objects [4]. A temporal average filter can be used to find the median of all the pixels in an image to form a reference image. Frames with moving objects can then be compared pixelwise to the reference, and a threshold set to determines if the pixel is part of the background or foreground. People counting and human detection can then be achieved by segmenting the foreground image into individuals.

However, this technique often relies on a static camera in a well placed location. This brings up several reasons as to why this method would not be suitable for this project. Firstly, the camera available is part of a head-mounted augmented reality device. The wearer has the ability to move the camera in 6 degrees of freedom. Secondly, the wearer will also be navigating a powered

wheelchair. As a result, the background is constantly changing, and the reference image would require constant recomputation before human detection can even begin.

Scanning Windows

Due to the ever-changing surroundings of a mobile robot, a better approach for object detection is to exhaustively search an image using scanning windows and determining if an object was detected in each window. However, it must be noted that this method is computationally expensive. In order to achieve real-time detection on a mobile robot, the use of a graphics processing unit (GPU) should be considered [5].

Classical Object Detection

Haar Cascades Haar cascades classifies images based on the value of simple features [6], which are variants of the difference between the sum of pixel values in rectangular regions. An intermediate representation of the original image is used to rapidly compute a small set of representative rectangular features.

A cascade of classifiers is then used to determine if the region is detected as a human. The detection process is that of a degenerate decision tree, where a positive result in the first cascade will trigerr an evaluation in the second, more successful classifier. As such, the initial classifier can eliminate a large number of negative examples with very little processing. After several stages, the number of sub-windows has been reduced radically

Histograms of Oriented Gradients The method proposed is implemented by dividing the image window into small spatial regions and calculating a local 1-D histogram of gradient directions for all the pixels in the region. The combined local histograms form the overall feature representation of the image.

The detection window is tiled with the Histogram of Oriented Gradient (HOG) descriptors. In the original paper [7], these feature vectors were then used in a conventional SVM based window classifier to give human detections.

Deep Learning Object Detection

Modern approaches for human detection largely depend on Deep Convolutional Neural Networks (CNN). The approach provides the best in class performance, as well as scaling effectively with more data. An added advantage of

using CNN based object detection systems for this project is that they are also capable of detecting multiple classes of objects.

An issue with CNN approaches is that the methods are trying to draw bounding boxes around objects of interest in images. However, we do not know the number of objects in the image beforehand. As such, to be completely sure every object has been detected, a naive solution is to take a huge number of regions and attempt to classify all the objects in the region, a computationaly expensive process.

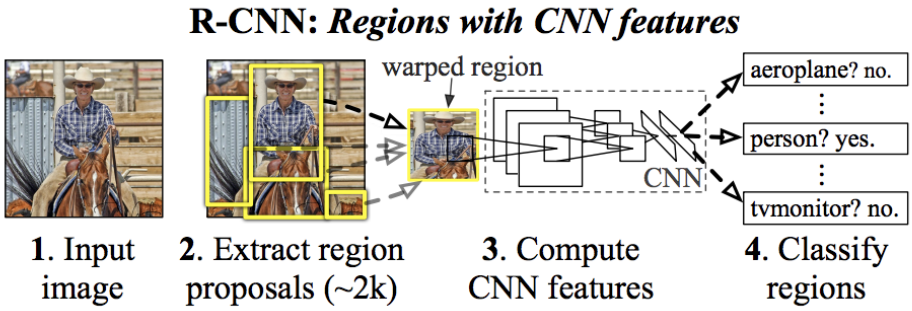


Figure 2.2: R-CNN Approach

R-CNN The R-CNN method uses a selective search to extract 2000 regions from an image [8]. The regions are selected by generating a large number of candidate regions and using a greedy algorithm to recursively combine similar regions into larger ones. The regions are then fed into a CNN that acts as a feature extractor and the output dense layer consists of the features extracted from the image, which are then fed into an SVM to classify the presence of objects in the region.

The major disadvantage to this approach is the amount of time required to train the network. Each training image has to be classied once for each of the 2000 region proposals. Furthermore, the selective search algorithm is a fixed algorithm (no learning is done), and as such, could lead to generation of bad candidate region proposals.

YOLO Whereas R-CNN uses regions to localize the object within an image, You Only Look Once (YOLO) looks at the image as a whole and uses a single

CNN to predict the bounding box and the class probabilities [3]. By looking at the image as a whole, the network can use features from the entire image to predict each bounding box.

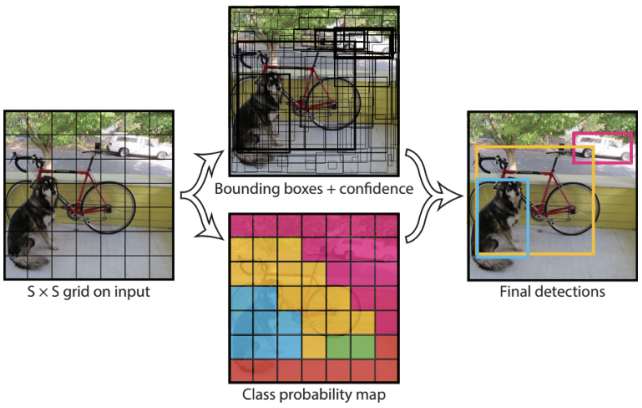


Figure 2.3: YOLO Approach

The model divides the image into an $S \times S$ grid, and for each cell, predicts a number of bounding boxes, the confidence for those boxes and the class probabilities.

2.1.3 Comments

As seen from the research, we can clearly see that there are many ways to solve the human detection problem. The classical approaches, although computationally efficient, are siginficiantly outperformed by the deep learning approaches. For a mobile robot in a public area, we want to be able to detect almost all humans in the surroundings to better inform the PWU.

However, the major disadvantage of the deep learning approach is the time taken to train the network, as well as the requirement of a GPU to achieve real-time performance. These issues will be addressed in a later section of the report.

2.2 Object Tracking

Object tracking can be defined as the ability to detect objects in consecutive frames and determining if the same objects are present. The techniques are often used in security and surveillance to track individuals across multiple cameras. A more relevant use of object tracking is in augmented reality with ARMarkers

to allow for more accurate placements of holograms as the user moves through the AR world.

2.2.1 Definition of Requirements

A common scenario for PWU in public spaces is having multiple people walking in the surroundings. Ideally, the augmented reality system should be able to track the same people across frames to be able to determine their direction of motion. As such, we focus our research on the multiple object tracking (MOT) problem in real-time. For an augmented reality system for a PWU, the object tracking must be done in real-time in order to feedback to the PWU. This narrows our field of research to online object tracking techniques.

2.2.2 Review of Exisiting Methodologies

Pedestrian detection is often achieved by using a high quality object detector and associating the detections across frames [9]. The associations are based on the appearance and location similarity. Furthermore, it is possible to discern simple motion patterns from the tracked pedestrians, allowing for more accurate tracking.

SORT

Methodology The Simple Online and Realtime Tracking (SORT) method relies on the accurate detections of a CNN to calculate bounding boxes of the tracked objects across frames [10]. The technique estimates the inter-frame displacements of each detected objects with a linear constant velocity model. The state of each tracked object is modelled using the bounding box centroids u and v, the scale and aspect ratio, s and r.

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]$$

When a new detection is associated with a tracked object, the bounding box of the new detection is used to update the tracked object state, and using a Kalman filter to update the velocity components [11]. To determine associations between new detections and tracked targets, the SORT algorithm relies on the intersection-over-union (IOU) distance between each detection and the predicted bounding boxes of all the existing targets.

For every detection to be tracked, a uniqe tracker identity must be created and destroyed when the object enters and leaves the image. The original implmentation of the algorithm relied on a IOU_{min} value to signify the existence of an untracked object. The tracks are then terminated if they are not detected for an allotted number of frames, to prevent the unbounded growth of trackers.

Limitations Due to the simplicity of the association metric, the significant overhead and complexity of object re-identification is removed, allowing for the system to work in real-time applications. However, this also reduces the accuracy of the tracking, since occlusions will spawn new trackers for the same objects. Furthermore, the accuracy of the tracking is largely dependent on the object detector providing accurate bounding boxes.

Deep SORT

The original SORT suffered from a high number of identity switches, since the association metric was only accurate if the state estimation uncertainty was low. Wokje proposed a solution to the issue by learning a deep association metric on a re-identification dataset [12].

Methodology The tracking and Kalman filtering in Deep SORT is mostly identical to the original SORT implementation. However, Deep SORT uses a Mahalonibis distance as an association metric between the Kalman predicted states and new detections. It further uses a second metric, whereby an appearance descriptor is calculated for each bounding box. A gallery of the previous $L_k = 100$ descriptors are kept for each track. The algorithm then iterates and measures the smallest cosine distance between the existing tracks and the detection.

The appearance descriptor is implemented using a CNN that has been trained offline on a person re-identification dataset. The Github implementation of the Deep SORT algorithm uses a simple nearest neighbour query without any additional metric learning.

Limitations Although the accuracy of the the tracking is improved and the issue of occlusions is reduced, the increased complexity of the algorithm requires more computational power. As stated in the paper, a modern GPU would be required to run this in real-time, due to the need for an appearance descriptor to be calculated for each detection.

2.2.3 Comments

For this project, we have limited ourselves to researching simple object tracking methods that work in real-time. We can clearly see a trade-off between accuracy of tracking and computational power. Further investigation into the hardware available and the importance of object tracker accuracy will be needed to decide what method would be best for the augmented reality system.

2.3 Head and Body Pose Estimation

Pose estimation is a general computer vision problem where we attempt to detect the position and orientation of an object. This process can be achieved by detecting keypoint locations that describe the pose of the object. For instance, in body pose estimation, we identify the joints in the body.

2.3.1 Definition of Requirements

An interesting concept to explore is that of head and body pose estimation as a way of infering the direction a person is walking in. For instance, people tend to look in the direction they are currently walking, but should they want to change direction, they also tend to look in that direction before changing. Similarly, if we can determine the body pose of a person, the system will be able to tell if a person is walking to or away from the PWU without relying on depth sensors.

2.3.2 Review of Existing Methodologies

Head Pose Estimation

Head pose estimation is intrinsically linked with visual gaze estimation [13]. If we can characterize the direction and focus of a person's eyes, it may be possible to determine the direction they will walk in next. As such, our research focuses on detecting facial keypoints in order to determine a persons gaze.

Facial Landmark Detection Before head pose estimation can be done, keypoints on the face must be detected [14]. These points will then be used to solve a Perspective-n-Point (PnP) problem to determine the head pose. There are many facial landmark detection techniques, depending on the number of landmarks to be detected. As the number of landmarks increase, the more accurate the pose estimation can be. However, it also increases the complexity of the detection, and as such, it becomes a trade-off between the two factors.

Gaze Estimation

Body Pose Estimation

2.4 SLAM

2.5 Augmented Reality Headsets

Chapter 3

Requirements Capture

Bibliography

[1] Ya Li Hou and Grantham K.H. Pang. Human detection in crowded scenes. *Proceedings - International Conference on Image Processing, ICIP*, (September 2010):721–724, 2010.

[2] Dongdong Zeng, Ming Zhu, Tongxue Zhou, Fang Xu, and Hang Yang. Robust Background Subtraction via the Local Similarity Statistical Descriptor. *Applied Sciences*, 7(10):989, 2017.

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. Technical report.

[4] Massimo Piccardi. Background subtraction techniques: a review*. 2004.

[5] Manato Hirabayashi, Shinpei Kato, Masato Edahiro, Kazuya Takeda, Taiki Kawano, and Seiichi Mita. GPU Implementations of Object Detection using HOG Features and Deformable Models. Technical report.

[6] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. Technical report, 2001.

[7] Navneet Dalal, Bill Triggs, Navneet Dalal, and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[9] Caglayan Dicle, Octavia I Camps, and Mario Sznaier. The way they move: Tracking multiple targets with similar appearance. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2311, 2013.

[10] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *Proceedings - International Conference on Image Processing, ICIP*, volume 2016-Augus, pages 3464–3468, 2016.

[11] R. E. Kalman and R. S. Bucy. New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*, 83(1):95, 1961.

[12] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Proceedings - International Conference on Image Processing, ICIP*, volume 2017-Septe, pages 3645–3649, 2018.

[13] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626, 2009.

[14] Vahid Kazemi and Josephine Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.