

Acme Robotics Proposal – Perception

Gaurav Raut
UID: 118199306
University of Maryland

Advait Patole
UID: 118130743
University of Maryland

OVERVIEW

Our design uses monocular camera to detect humans and get their positions in the robot's reference frame. The module is developed in such a way that it detects humans ($N \geq 1$) and then creates bounding box around it. The distance of human is calculated from the pixel values of the bounding boxes. This system is perfect for use in robots like autonomous caddies in malls or airports for transportation which is under-development by Acme. The module detects humans and if the distance is below a threshold, the vehicle slows down. The distance measurements can be fed to the odometry of the robot in order to control its motion.

The project could be easily expanded and trained to detect more objects and take actions based on the type of the object. The AIP design process allows continuous development. Because of this, new features and modules could be added over new sprints and thus making the module highly customizable.

Deliverables:

- A module that is able to detect humans in front of the caddy.
- The module is able to determine the distance between the caddy and the human.

DEFINITION AND ACRONYMS

HOG – Histogram of Gradients
SVM – Support Vector Machines
AIP – Agile Iterative Process
TDD – Test Driven Development
STL – Standard Template Library
OS – Operating System

PROCESS

The process to be followed will be based on an **AIP** mindset which uses **Scrum** as its framework [3]. As with most AIP strategies, our implementation will take place in two sprints of one week each. The **total iteration capacity** for each sprint is $4 \times 7 \times 2 = 56$ hours/week. The team consists of two programmers who will assume all-rounded duties. The process will follow the waterfall model and will start by understanding the **requirements** of the project. Next, a functional preliminary prototype of the final product will be made and this phase is called as **prototyping**. This will assist the programmers to get valued inputs for the product backlog as the users/stakeholders will be able to experience a tangible product as opposed to a verbal explanation [2]. Hereafter, after

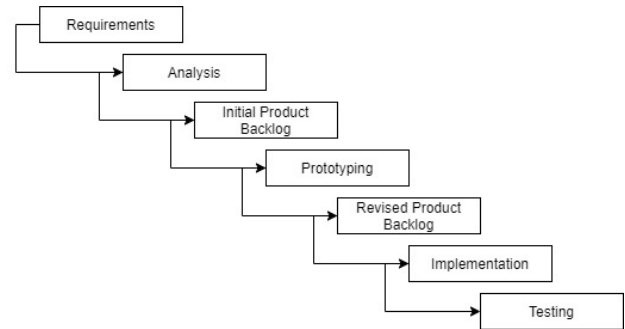


Figure 1 Waterfall Model

procuring the revised product backlog, the first implementation will be executed.

Quality: This implementation will be paired with **TDD** to ensure concurrent quality in the software. The two programmers will study a certain required feature and write the unit tests which will then test the program written by the other programmer. The process of testing and software implementation will hence be executed concurrently. Some simple strategies that are to be followed for proper coordination are making use of **daily loops**, which involve scrum meetings. Such meeting will allow the programmers to be in sync and clarify conflicts. Some complex functionalities and features will follow the practice of **Pair Programming**, for example, algorithm implementation.

ORGANIZATION

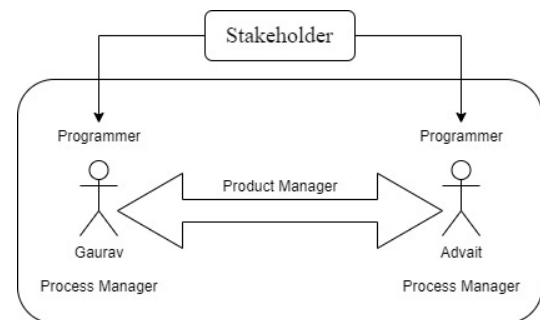


Figure 2 Organization

The organization consists of a modest two programmer team. Every programmer will assume to be a process manager and the product manager. The stakeholders and users will drive the product backlog and requirements. This suggests that the product backlog is a rather a live document rather than a predefined checklist.

TECHNOLOGY

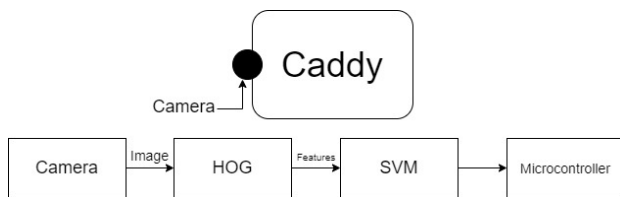


Figure 3 Interfacing

The monocular camera inputs a stream of images, i.e., video, to our microprocessor. Every frame in the image will be converted into a feature vector by our custom-made HOG feature descriptor. The feature vector will be given as an input to our pre-trained custom-made linear SVM which will classify the features as Human and Non-Human. Further, if there is a human in the frame, the algorithm will calculate the distance of the human from the caddy based on the pixel location and pre-defined equations and assumptions.

RISKS AND MITIGATION:

A HOG is not a rotation invariant representation. This means that HOG can only vectorize features of images within a certain orientation range for object detection assignment. This can be corrected if we train images with different orientation. Monocular camera we have made calibration before implementing while comparing the given distance with the calibrated calculations instead we can use depth cameras for more accurate results like Intel® RealSense™ depth camera D435.

HOG is not scale invariant we need to normalize the data to a particular scale to overcome this.

Multiple detection boxes get created during detection we can use non max suppression to solve that problem.

COST

Our product costs less computational power as opposed to its competitors like CNN and will provide a fairly accurate prediction. Our product will also make use of a monocular camera as opposed to the other expensive sensors in the market, for example, Depth Camera. Because of this, our product will work on the most basic of microprocessors and will eventually save costs and energy. This will prove to be vital, especially during the rush hours or busy days.

TECHNICAL SPECIFICATION

Language: C++

Build System: CMake

OS: Ubuntu Linux 18.04

Libraries: OpenCV (3-clause BSD license), STL C/C++

Algorithm: HOG with Linear SVM

Dataset: INRIA dataset <http://pascal.inrialpes.fr/data/human/>

ALGORITHM

The HOG descriptors convert image into a feature vector. The input image ($64 \times 128 \times 3$) is converted to a 3780-length vector. The SVM classifier is trained by fitting appropriate parameters

to train whether image has human or not. Once it is trained, a sliding window is created of size 64×128 which creates different image patches each of length 3780 feature vector. The SVM is used on each of the feature vector to get whether human is found in that image or not. If it is found, then we store the coordinates to create the bounding boxes. In-order to solve the problem of multiple bounding boxes, we can use a non max suppression to remove overlapping boxes [1].

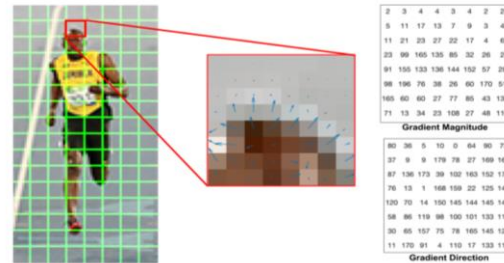


Figure 4 HOG Gradient [6]

REFERENCE MATERIALS

- [1] Dalal, Navneet & Triggs, Bill. (2005). Histograms of Oriented Gradients for Human Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005). 2.
- [2] Software Engineering: The Current Practice by Vaclav Rajlich Copyright Year 2012 ISBN 9781439841228 Published November 17, 2011 by Chapman and Hall/CRC 111 B/W Illustrations
- [3] <https://www.atlassian.com/agile/scrum>
- [4] www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection
- [5] <https://medium.com/@richa.agrawal228/person-detection-in-various-posture-using-hog-feature-and-svm-classifier-2c3a3991022c>
- [6] Histogram of Oriented Gradients explained using OpenCV (learnopencv.com)

Gaurav Raut

UID: 118199306

Advait Patole

UID: 118130743

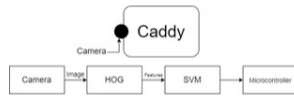
ACME Robotics Proposal - Perception	
Overview Technology Our design uses monocular camera to detect humans and get their positions in the robot's reference frame. The module is developed in such a way that it detects humans ($N \geq 1$) and then creates bounding box around it. The distance of human is calculated from the pixel values of the bounding boxes. Process The process to be followed will be based on an AIP mindset which uses Scrum as its framework. As with most AIP strategies, our implementation will take place in two sprints of one week each. The total iteration capacity for each sprint is $4 \times 7 \times 2 = 56$ hours/week.	Algorithm The HOG descriptors convert image into a feature vector. The input image ($64 \times 128 \times 3$) is converted to a 3780-length vector. The SVM classifier is trained by fitting appropriate parameters to train whether image has human or not. Once it is trained, a sliding window is created of size 64×128 which creates different image patches each of length 3780 feature vector. The SVM is used on each of the feature vector to get whether human is found in that image or not. If it is found, then we store the coordinates to create the bounding boxes. In-order to solve the problem of multiple bounding boxes, we can use a non max suppression to remove overlapping boxes.
Basic block diagram  Cost Parity Our product costs less computational power as opposed to its competitors like CNN and will provide a fairly accurate prediction. Our product will also make use of a monocular camera as opposed to the other expensive sensors in the market, for example, Depth Camera. Because of this, our product will work on the most basic of microprocessors and will eventually save costs and energy. This will prove to be vital, especially during the rush hours or busy days.	Schedule 10/06 - ACME Proposal 10/07 - Implementation using mock product backlog 10/07 - Sprint Meetings and Iterative backlog 10/14 - First Sprint review meeting and new backlog 10/15 - Second sprint based on new product log 10/16 - Rigorous testing setup using first implementation as reference for stub 10/16 - Adding regressive testing suite 10/18 - Code Inspection 10/23 - Final implementation due 10/24 - Testing and maintenance Deliverables <ul style="list-style-type: none">• A module that is able to detect humans in front of the caddy.• The module is able to determine the distance between the caddy and the human

Figure 5 Quad Chart