16-385 Computer Vision Term Project Proposal

Scott Martin
semartin@andrew.cmu.edu

1. Introduction

In this paper, a term project for 16-385 is discussed. This project attempts to refine the control of a homemade NERF gun turret by designing an algorithm that tracks and predicts the motion of human targets using a rotating camera.

2. Hardware & Control

The turret consists of a NERF Stampede gun mounted to a plastic turntable, which is capable of 360° of rotation. By using a continuous-rotation servo motor [1] and an onboard power supply, the turret has completely unrestricted, continuous horizontal rotation.

The PWM signal for the servo motor and the trigger relay that fires the gun are controlled using an Arduino Uno microcontroller board, which receives command from a mounted laptop through a USB serial port. The laptop controls odometry and tracking, pulling a video stream from the webcam sighted down the gun barrel and using this information to control the servo and the gun. See Fig. 1 for an overview of how control works.

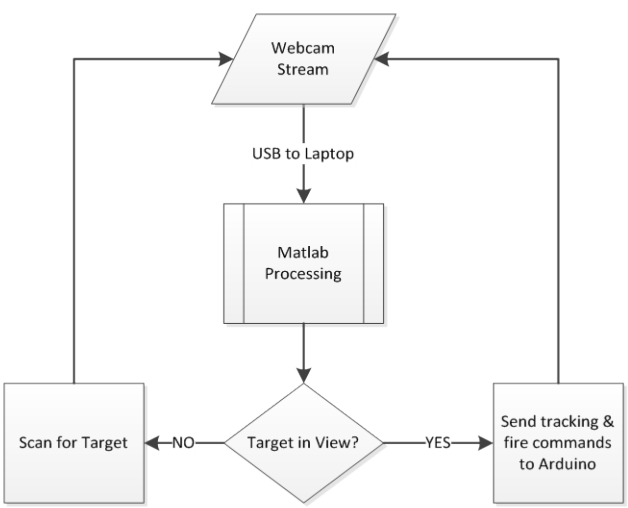

Fig. 1: Control loop for NERF turret

3. Proposal & Objectives

Previous attempts at automatic target tracking and elimination used a fixed camera, which greatly simplifies tracking but is deficient for two reasons. Firstly, this limits the field of view of the turret to the viewing angle of the camera, which may allow enemies to approach from an untracked direction and disable the turret. Secondly, the method of tracking with a fixed camera also introduces a cable attached to the ground at some point, which constrains the rotation of the turret and may prevent it from tracking to an enemy. The solution explored in this research paper removes the fixed positioning of the camera. Instead, the camera is mounted to the base of the plastic turntable and sighted along the barrel of the gun. Without any external fixed points, the turret is allowed to rotate indefinitely in either direction in its search for enemy targets.

The computer vision problem becomes more complicated. Instead of a fixed background with simple motion tracking, the algorithm must filter out the moving background in the image caused by rotation of the turntable in order to track to an enemy. However, the movement of the background also allows the potential to estimate odometry of the servo motor. By providing this real-time angular speed estimate, we can predict the future angle of the gun and use this information to further refine our tracking algorithm.

4. Procedure & Implementation

The algorithm shall be developed sequentially in three steps.

The first step is implementation of background homography and rotation speed estimation. First the servo must rotate the camera a full 360° to capture the background scenery, which will be removed in order to detect targets in part 2. To accomplish this, homography must be established between the image frames, possibly through the use of feature tracking or simple squared-error estimation of image patch locations. Since movement is constrained in one dimension, the matching process should be very fast, however care must be taken to prevent inaccuracies in camera position. For instance, rotation around the axis of view (Z-axis) would produce a non-horizontal direction of movement, and would have to be rectified before image stitching.

Once the background is acquired, the algorithm can then estimate the rotation speed of the servo motor by tracking the speed at which the background moves. Spatial locality of subsequent images allows us to search in a small range around the last known image position to determine where the camera has moved. Prior knowledge of the direction and power applied to the servo also narrows the area of search.

The second step in implementation factors in the added complication of a moving target. Background tracking and odometry will be revisited with an added moving foreground object to test robustness and accuracy. Of course, maintaining accuracy in the odometry step necessitates identifying and filtering out the foreground object, so in this step we must also identify foreground targets and separate them from the background.

The third, final, and most satisfying step is testing the capabilities of the NERF gun in eliminating targets. Foreground objects must be identified as either passive - such as a slightly moved background object or camera noise - or as an enemy that requires pacification. Once identified, a tracking algorithm shall be developed that anticipates the target's motion and fires at a high enough rate and accuracy as to discourage further ingress.

Once accurate tracking of a single target is achieved, the identification algorithm shall then be refined to assign priorities to multiple detected enemies based on closeness, target size, movement speed, and other factors that influence importance of the target and the ability of the turret to hit the target. Such refinement will prevent waste of ammunition and potential fooling of the algorithm.

5. Preliminary Research

For background calibration, rotation speed of the servo can be set such that consecutive frames deviate by a very small number of pixels. Combined with a fixed movement axis and a known direction, matching can be exceedingly simple. Images will likely be able to be paired by calculating squared error of a small number of distributed image patches – this will run very fast. In order to average out camera rotation, a few frames could be gathered from fast servo rotation. Either matching via homography or Lucas-Kanade flow estimation could be used to detect motion deviation from horizontal.

Another consideration is the width of the panoramic image buffer used to store the background image. This will vary from camera to camera, but a standard webcam with roughly 60° FOV will require a buffer roughly six times as wide as the image.

Background subtraction requires odometry to determine our camera's viewpoint in the background image buffer, but odometry also requires background subtraction to remove foreground objects which should not contribute to background motion. Assuming the foreground object occupies less than half of the image, a voting scheme is one possible solution to this – rotation speed can be measured in various locations of the image and the majority of the responses is calculated. Our knowledge of the direction and speed of servo rotation can also be an influencing factor.

For detection of enemies, the subtracted background can be thresholded to a binary (black and white) image, all holes filled with *imfill(IM, 'holes')*, and independent components detected with the *bwconncomp* function [3]. The centroid of the object, which presents a good starting point for targeting, can be calculated from the mean of the object's pixels.

Once targets have been established, particle filters are a good first choice in predicting the velocity of each target [2]. Target priority could be calculated by some weighted measurement of the number of pixels of the target (i.e. target size), the speed estimate from the particle filter, closeness of the target to the gun's crosshairs, and human-like shape. Rudimentary face detection may possibly be used to exclude non-human targets.

6. Project Schedule

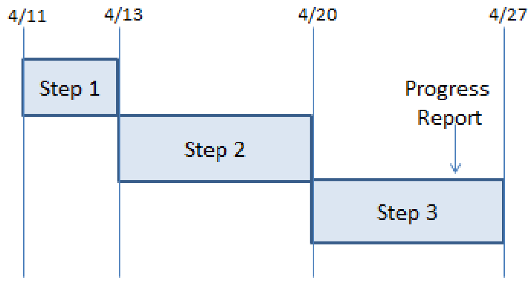

Fig. 2: Gantt chart of project schedule

April 11th: Project proposal (this paper) is due.

April 11th-13th: Step 1 implementation. Begin testing robustness of odometry against foreground objects.

April 13th-20th: Step 2 implementation. Begin target identification.

April 20th-27th: Step 3 implementation. Revisit project requirements and revise schedule.

April 25th: Single page progress report & additional slide is due.

May 8th: Term project final report is due

May 11th: Term project class presentation

7. References

1. "Continuous Rotation Servo (#900-00008)" . Parallax Inc., 29 4 2004. Web. 11 Apr 2013. <http://www.parallax.com/dl/docs/prod/motors/crservo.pdf>.

2. Kaijen, Hsiao. "Particle Filters and their Applications." . MIT, 11 Apr 2005. Web. 11 Apr 2013. <http://web.mit.edu/16.412j/www/html/Advanced lectures/Slides/Hsaio_plinval_miller_ParticleFiltersPrint.pdf>.

3. "bwconncomp - find connected components of a binary image." MathWorks. N.p.. Web. 11 Apr 2013. <http://www.mathworks.com/help/images/ref/bwconncomp.html>.