Fencing Referee Bot

Jia Wen Goh, Elliot Pavlovich, Quinn Collins

March 9, 2018

Introduction

The motivation for this project came about because all of us are fencers and know the struggles of referees at tournaments. Refereeing requires a lot of manpower that isn't always available, especially during practice when everyone wants to fence. If some of the process could be automated, it would be a lot easier.

Another issue that arises with human referees is that it can sometimes be hard for the referee to maintain attention on both the fencers and the score box. Additionally, the score boxes are sometimes faulty and are unable to keep the score of the bout and the referee must keep track on their fingers or a phone app. Another problem lies with poor human handwriting - it is very hard to tell the what the scoresheet says. Most score-related documents at a fencing tournament are done on paper. Referees and athletes must walk back and forth to a specific person whose job is to wait at a table with a computer and input all the paper score sheets that are brought to them. Having this digitized would make this process far more efficient.

How fencing works:

There are 3 different types of fencing (foil, epee, and saber). For this project, we will be focusing on epee as it has the least complex rules due to the lack of "right of way." In epee, the fencer is awarded a point when he/she gets a touch (versus saber and foil, where there are certain 'ground-rules' that determines if the point is valid). Figure 1 shows what a fencing match at a tournament looks like. There is one referee who stands facing the fencers and the lights of the scorebox. The referee will walk alongside the fencersm making sure that there are no illegal moves. They will wait till someone makes a touch, before looking back at the score box to confirm.

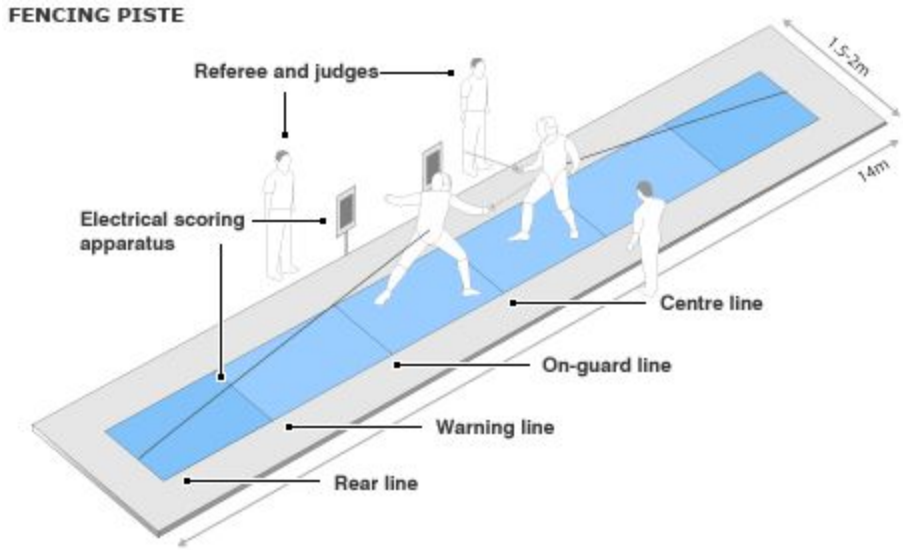

Figure 1: What a typical fencing set up looks like. The signs behind the fencers are the score box that lights up red or green, depending on who scored the touch.

Rules (for epee):

- The bout starts off with the referee "testing" the weapon. They put a 700g weight on the top of the weapon and wait for the spring to push the weight back up. Players then touch each other's bell guards to confirm that the weapons are grounded, and hitting your opponents guard will not award you a touch.
- Points are scored when a player hits their opponent with the tip of their weapon. It is possible for both players to be awarded a touch in one "play".
- Any part of your body is target area (including head and feet)
- There are penalties for running off the strip (either side or end). There are also penalties for turning around.
- Players go back to the on guard line (start lines) after each point

Scoring/Tournament Rules:

- If the bout is going to 5 and the score is 4-4, both players remain at 4-4 until a single person gets a touch.
- In a 5 touch bout (pool round), the player to win is the player who gets to 5 before the 3 minute timer runs out. If the timer runs out before a player has scored 5 touches, then the person with the most points at the end of the 3 minutes wins. If there is a tie at the end of the 3 minute period, a coin is flipped (or an alternative method to randomly pick one player is used) to choose who will have "priority". Then there is one minute of sudden death overtime. If no point is awarded during overtime, then the person who had the "priority" is awarded the point and the victory for the bout.
- In a 15 touch bout (direct elimination round), the player to win is the player who gets to 15 before the two 3 minute timer runs out. There is a one minute break when a fencer gets to 8 touches or when the first 3 minute timer runs out. After the one minute the bout resumes, either with the old timer finishing the time that is left on it or with the start of the new timer. The same rules apply for when both timer run out, player with most touches wins, or in case of a tie there is the one minute overtime.
- If a player runs off the back of the strip, the play stops and the other player is awarded a point. If the player runs off the side of the strip, the play stops, the players are "recentered" and the player who ran off the strip loses a meter.

How a tournament works:

All fencers signed up for the tournament are initially seeded based on their rankings prior to the tournament. A round robin of 5-8 fencers is played. This is the pool round, where each bout only goes to 5 touches. Each person is assigned one round robin group, and all groups have mixed skill levels. Based on how you do in the round robin (how many bouts you won, how many touches you scored, how many touches were scored on you), you are seeded for the direct elimination round. A tableau is made based on your post round robin seeding. Then you are

matched with someone and fence them to 15. Winner moves on. There may or may not be a fence off for 3rd place.

The rules are strict and easy to follow. This ruleset is the reason for wanting to have a robot to automate it. It will eliminate the amount of manpower to run each tournament, and make the behind the scenes score recording more efficient. Primary goals for our robot include: moving along side the fencers, tracking the score box and awarding points, as well as running the overall tournament (setting up the pools and DE (direct elimination) bouts). These goals, our plan on how to accomplish them, and potential problem related to the goals are detailed below. A stretch goal is to have another camera on the robot that detects people in its proximity. If there is someone nearby, the robot will warn the person to step away.

By the end, we hope to have a robot that can follow the movements of both fencers, while maintaining the score box in its field of view + tracking the score of the bout. The robot should also a screen that shows the bout progress.

Related Work

From reading newspapers, it is easy to see that there are very mixed feeling about whether or not to have a robotic referee[2]. We have all the capabilities to have robot referee professional sports. For example, computers are able to render a tennis ball that was hit, and be able to tell/show if it hit the line or was just out of bounds. Hsu talks about how though the refereeing would be much better, it may be less fun for viewers. People like it when there is a "villian" it makes it more interesting. The referee for many sports is seen as the "villian" as they may make bad calls (or miss them completely) due to just being human, and not being able to see everything. Take that away and there is not one to really be mad at. The anger at the referee is what makes watching the sport so frustrating, yet so entertaining and rewarding. Do we really want to take that away?

More technical related works:

Mehran Pakdaman and M. Mehdi Sanaatiyan outlined the hardware and algorithm for a line following robot [1]. While the requirements for their robot are more complex, and we benefit from having a more controlled environment: our line is a single color, and will have no turns. The basics of the algorithm used are certainly still useful, and will provide a good baseline of comparison for our line following algorithm.

Related to vision, and being able to detect/track objects, Das [3] talked about service robots that need to locate and identify objects. He discussed the use of color and how its distinctive feature helps to identify an object. Luckily the score box has very distinct red and green lights that indicate when someone has gotten a point. The article also talks using the range. This is also a

very good in that we want the robot to be able to read and understand the score box from different distances.

Additional ROS Tutorials:
http://wiki.ros.org/leg_detector
http://wiki.ros.org/cob_people_detection
http://wiki.ros.org/ppl_detection

Technical Approach

To ensure our robot is fully functional with no bugs, we will build and code successive layers of functions. At each stage, we will ensure complete functionality and fix any potential bugs/problems before we proceed to the next. This layered approach will allow us to detect and solve any problems related to the function before we move on to deeper layers, where debugging the code may become increasingly difficult. For example, when building the movement capabilities of the robot, we may face difficulties in ensuring that the robot adheres to the line on the floor. If we do not solve this problem before we move on to the next (tracking the fencers), debugging the issue will be hard - we will find difficulties in understanding if the problem lies with the movement or tracking modules.

The robot's movements will be parallel to the strip. In other words, the front/back directions are the extreme ends of the strip. A camera will be mounted on the robot's left and right sides; the side that is facing the strip will be used as the primary tracking camera that follows the fencers, and keeps track of the lights on the score box. The other camera will be used for our stretch goal of detecting any nearby people and warn them to stay clear of the robot. Figure 2 illustrates these details.

There are 2 distinct systems that will work together: the **movement system** and the **tracking system**.

- The movement system ensures that the robot stays within a predefined path that is parallel to the fencing strip.
- The tracking system is responsible for keeping the 2 main elements, the fencer and the score box, within its view. It should be able to track points scored based on the lights on the score box.

The tracking system will be built 'on-top' of the movement system, i.e. where the camera is trained on depends on the position of the robot (and fencers).

The stages are described as such:

1.	**Function:**	**Track the lights of the score box** - the robot should be able to detect the red/green lights on the score box when it lights up (i.e. when either/both fencers score a touch).
	Details:	Able to **detect the light** and award a point upon the **first change of the state of the light** (from off to on). - It needs to be able to detect a **rough 'square' of pixels** Understand the basic rules of fencing: - The winner is the first to 5 points, or the fencer with the higher points after 3 minutes. - Double lights awards both fencers with 1 point. - There is a time limit of 3 minutes per bout. The timer is halted when a touch is scored, and resumes when the fencers resume. - If both fencers are at equal points by 3 minutes, the bout goes into priority, i.e. 1 minute sudden-death. - If both fencers are at 4-4, the score remains at 4-4 till one person scores a single light.
	Potential Problems:	We will need to prevent the camera from falsely detecting people with **clothing of the same color as the lights** (red/green). - We intend to either prevent it by limiting the pixel-cap (i.e. only if it detects *x* amount of pixels of this color), or by placing a distinct orange tape around the lights of the score box to signal to the robot that it should only look at colors in that area.
2.	**Function:**	**Robot movement along a line** - the robot will have to be able to follow a line of tape that is placed alongside the fencing strip. This tape dictates the path of the robot.
	Details:	Follow line of tape forward and backwards. - The robot will consistently turn to the right until line is detected, then turn to the left to avoid passing over the line. This results in a behavior that looks like the robot is 'bouncing' off the line. Detect different color tape at ends to avoid going too far.
	Potential Problems:	Distinguishing the colored tape from the floor . - We will thus have to use a distinct color of tape, and account for different colored tapes due to different floor areas in different fencing locations.

3.	**Function:**	**Camera tracking fencers (while robot is stationary)** - identify the two fencers and track their movements while they are within the robot's field of view
	Details:	Needs to identify fencers as distinct objects, and follow their movements - Use existing ROS package for detecting people with the kinect: - http://wiki.ros.org/ppl_detection Needs to be able to train its camera on the fencers as they move.
	Potential Problems:	Due to the stance of fencers, it may be difficult to label the fencer as a person. Thus, this requires some tweaking and testing to ensure its effectiveness.
4.	**Function:**	**Robot movement while keeping score box (only) in view** - move alongside the fencing strip (with no fencers on it), while keeping the score box within the robot's field of view.
	Details:	The robot must be able to identify the score box as an object, and detect lights of the score box while the robot is moving, and at different points along the fencing strip - This will build off stage (1) The robot must be able to follow a line of tape placed alongside the fencing strip - This will build off stage (2)
	Potential Problems:	Identifying the score box when no lights are on might be difficult, because the robot will be viewing it from different angles, and therefore, it might appear to be slightly different shapes. To solve this, we might put a box of tape around the score box.
5.	**Function:**	**Robot movement while keeping fencers in view** - The robot must be able to move forwards and backwards, following fencers as they move along the fencing strip
	Details:	The robot needs to be able to identify the 2 fencers while both the fencers and robot are moving - This will build off stage (3). We will possibly smooth the movements/locations of fencers so they are easier to follow The robot must follow the fencers' movements on the strip to keep them in view

		- The robot must determine whether it should move forward or backwards based on the positions of the fencers in its field of view. It should then move in that direction, following the line of tape placed alongside the fencing strip
	Potential Problems:	The robot must be able to keep up with the fencers. If the robot is too slow, it may be moved further away to expand its field of view. This might make detection of the fencers or score box more difficult.
6.	**Function:**	**Robot movement while keeping score box and fencers in view** - the robot should move along the fencing strip, following the movements of the fencers. It must also keep the score box in view, and move out of the way if the fencers are blocking its view of the score box
	Details:	The robot must be able to follow fencers while keeping the score box in view - Build off of (4) and (5), making sure both function simultaneously - The robot must also correct its position if one of the fencers is blocking its view of the scorebox
	Potential Problems:	Correcting so that fencers are not blocking the score box happens simultaneously with fencers moving, perhaps leading to them blocking it again. Detection of continuous blocking might be necessary to form a better plan to move out of the way
7.	**Function:**	**Have algorithm to organize tournament format** - the robot must keep track of scores and who should be fencing
	Details:	The robot must be able to organize the tournament itself. This includes creating pools (round-robin style fencing where each fencer fences each other fencer to 5 points), creating direct elimination matches (based on scores from pool, pair off fencers, who will fence to 15 points. The loser is eliminated), and store the scores of bouts - Keeps score - Make pools - Make DE bouts
	Potential Problems:	This should be straightforward because it only relies on detecting points, as in step (1). With extra time, a GUI to display the status of the tournament could be useful.

Stretch Goals		
8.	**Function:**	**Be able to call certain fencers to strip** - inform fencers when it is their turn to fence
	Details:	Add a way for the robot to inform users who should be fencing, and tell them the score as well - Add text-to-speech - call fencers at start of bouts - call out score at end of bout
	Potential Problems:	Fencing tournaments are often loud, both from people talking and blade contact. The robot might not be loud enough to be heard
9.	**Function:**	**Use another camera to detects people near the robot, and warn them to move out of the way**
	Details:	The robot should see people blocking its path and warn them to move - Detect people with second camera - Use text-to-speech to warn them to move out of the way, or stop if they don't move
	Potential Problems:	The robot's warnings are no guarantee that people will move out of the way, and it might still not be able to move into a position to see the scoreboard or fencers.

Evaluation Plan

To evaluate the success of the referee bot, we can track how successful it is at each task, and thus get a quantitative measure of the performance of the robot's movement system, tracking system, and overall performance.

The robot's movement system is the most critical, because it has the most risk to the robot. The robot should not lose the tape line it is set to follow. Thus, the robot will have a set max speed, assigned such that it does not cause the robot to lose the tape line. We can conduct multiple trials with the robot moving at variable speeds, and then at max speed: if the robot loses the tape line, it will be considered to have failed.

The robot's tracking system can be measured quantitatively as well. We will track the amount of time the robot identifies each object, and compare it to the total runtime. For example, we can track the robot's success at tracking all objects as follows:

$(T_{scorebox} + T_{fencer1} + T_{fencer2})\ (3 * T_{running})$

$T_{scorebox}$: amount of time the robot can identify the score box
$T_{fencer1}$: amount of time the robot can identify the fencer to its left
$T_{fencer2}$: amount of time the robot can identify the fencer to its right
$T_{running}$: amount of time the robot is running

We can further break this down to measure the robot's success at tracking the score box alone:

$T_{scorebox} / T_{running}$

And the robot's success at tracking the fencers alone:

$(T_{fencer1} + T_{fencer2})\ (2 * T_{running})$

We can also measure the robot's success at avoiding having its view of the score box blocked by comparing it to a human tracking the score. We can compare the amount of points the robot registered for both fencers to the total amount that a human registered. Thus, we can track separately the robot's skill at avoiding having its view blocked.

References

1. Pakdaman, Mehran and Sanaatiyan, M. Mehdi. "Design and Implementation of a Line Follower Robot" ICCEE, 2009.
http://ieeexplore.ieee.org/document/5380235/

2. Hsu, Hua. "What If We Had Perfect Robot Referees?" *The NewYorker*, The New Yorker, 30 March 2017, 9 March 2018
https://www.newyorker.com/culture/cultural-comment/what-if-we-had-perfect-robot-referees

3. Das, Dipankar. "Object Detection for Service Robot Using Range and Color Features of an Image." *International Journal of Computer Science, Engineering and Applications*, vol. 4, no. 3, 2014, pp. 17–29., doi:10.5121/ijcsea.2014.4302.