Enhancing Fencing Lunge Performance through AI and VR Analysis

Nathan Guan

Abstract

Background/Objective: Fencing, a sport requiring precise movements and rapid decision-making, utilizes various techniques, with the lunge being fundamental across all weapons. The study aims to address the gap in optimizing lunge techniques to reduce injury risks and enhance performance. This research explores the use of Artificial Intelligence (AI) and Virtual Reality (VR) technologies to analyze and improve fencing lunges.

Methods: The methodology involved a multi-stage analysis, beginning with 2D analysis using Google Video Intelligence API for initial posture assessment. Subsequently, a more comprehensive 3D analysis was conducted using DeepMotion's AI motion capture, allowing detailed examination of body mechanics. The data was then visualized in an immersive environment using an Oculus VR application developed with Unreal Engine, enabling real-time feedback and adjustment.

Results: Findings indicate significant improvements in lunge technique, with a reduction in movements that could lead to knee injuries. The 3D analysis provided detailed insights into optimal body positions for power and safety. The VR application demonstrated its effectiveness in offering an interactive platform for technique refinement.

Conclusions: The integration of AI and VR in sports training, specifically in fencing, shows promising potential for enhancing athletic performance and reducing injury risks. This study contributes to the broader field by demonstrating the applicability of advanced technologies in sports training and opens avenues for further research in other sports techniques.

Keywords: Fencing Performance, Artificial Intelligence, Virtual Reality, Lunge Technique, Biomechanical Analysis, Injury Prevention, Sports Training, 3D Motion Capture, Video Intelligence API, Unreal Engine.

1. Introduction

Fencing is a sport that demands pinpoint accurate movements with quick on-the-fly decision-making. Many different skills and techniques are present in fencing but the lunge is a fundamental footwork technique employed in all three fencing weapons - foil, epee, and sabre - and is considered essential to contemporary fencing styles. It involves a forward movement with the front foot while the body is propelled forward with the back leg, forming the foundation for offensive actions and combos. Hence, improving the lunge is crucial for success in fencing.

Figure 1: Raluca Cristina Sbârcia (left) Lunges at Joséphine Jacques-André-Coquin, Women's Epee Event of the 2013 World Fencing Championships.

Unfortunately, many young fencers, including myself, experience sport-related injuries, such as knee pain. One factor that can contribute to knee pain in fencing is an incorrect lunge technique. Using 2D and 3D data analysis, we are able to examine the specific positions and angles which can help fencers to adjust their lunges to prevent future injuries.

This study aimed to explore the utilization of advanced artificial intelligence (AI) (J. Kim, "Artificial Intelligence in Sports Training," AI Review, vol. 7, pp. 23-30, (2018).) and virtual reality (VR) (S. Su, Y. Zhang, and L. Wang, "A Virtual Reality System for Fencing Training," Journal of Virtual Reality, vol. 18, pp. 5-12, (2013).) technologies in improving fencing performance, with a particular focus on the technique of a lunge. Our journey began with a 2D machine learning model using Google Video Intelligence APIs (Google, "Video Intelligence API documentation," Google Cloud. https://cloud.google.com/video-intelligence/docs (2022).), followed by the implementation of a more advanced 3D AI model using DeepMotion 3D Animation APIs (DeepMotion, "Animate 3D," DeepMotion. https://www.deepmotion.com/animate-3d (2022).). Lastly, the 3D animation model was imported into Unreal Engine and used to create an Oculus VR application for visualizations.

2. Analyze 2D Posture with GCP Video Intelligence API

2.1. Advice from my fencing coach

To commence my research, I began by capturing some video footage of my lunges. During practices and tournaments, I would video my actions to begin analyzing my lunges. Afterwards, I sent the clip to my fencing coach, an international fencer and coach Mario Jelev, who provided me with valuable feedback that read as follows:

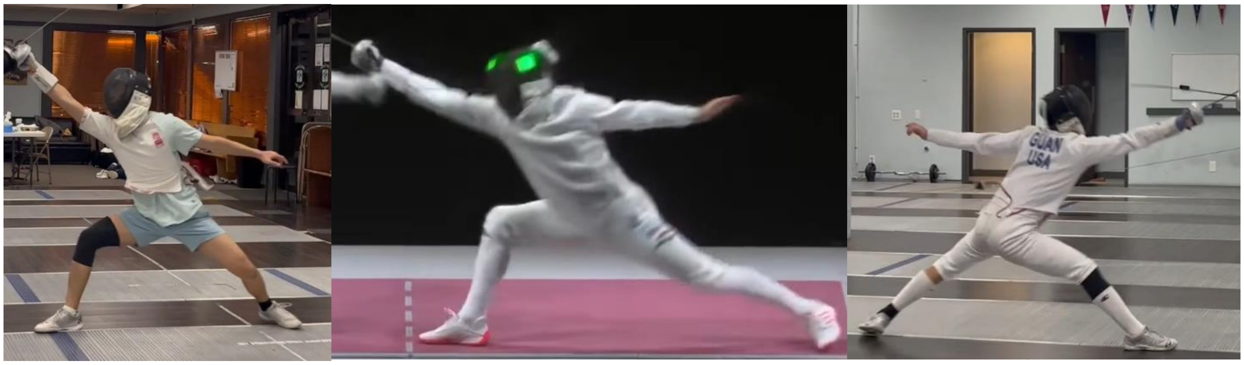

Figure 2: Nathan’s Training Lunge(Left), Olympic Champion Siklósi’s Lunge(Middle), and Nathan’s Tournament Lunge(Right)

1. Nathan’s upper body needs to be lower, which means that the front leg should form an angle closer to 90-degrees.
2. Nathan’s back leg needs to be fully extended to form a stronger push forward. The majority of the power in a lunge comes from the back leg and leaving it not fully extended is wasting speed and power.
3. Nathan’s lunge moves too much on the vertical plane while lunges are more horizontal. For stability purpose, the upper body needs to move in the horizontal direction and not as much movement vertically. This is because the bouncing in the vertical direction decreases tip accuracy. Additionally, the upwards movement is wasting time and can cause a greater force to be applied to the knee increasing the chances of injury.
4. When looking at Nathan’s training and tournament lunges, they are noticeably different. This can be due to nervousness but there is a second jittery action that follows. This creates unnecessary stress and strain in the knee. Fixing this habit and making the lunge more fluid and a one-tempo action will improve speed, power, accuracy, and knee health.

This feedback brought up few questions that I would like to analyze using machine learning technologies:

1. Are the positions of my front and back knee optimal for power, speed, and knee health?
2. Could the positions of my body be improved? (Hip, Shoulder, ect.)
3. Did my front arm fully extend?
4. How was far does my lunge travel? (Positions of hand and leg)
5. How does a tournament lunge compare with one in practice and one from an Olympic champion?

2.2. Pose Detection using Person Detection Feature

The GCP Video Intelligence API offers a wide range of variety of features for Person Detection. From among the available tools, I chose to utilize it to determine the angle of my knees and arms. According to the Person Detection API documentation (Google, "Video Intelligence API documentation," Google Cloud. https://cloud.google.com/video-intelligence/docs (2022).), the following are specific attributes that the Person Detection feature can identify:

Characteristic or landmark	Possible values
Body parts	nose, left_eye, right_eye, left_ear, right_ear, left_shoulder, right_shoulder, left_elbow, right_elbow, left_wrist, right_wrist, left_hip, right_hip, left_knee, right_knee, left_ankle, right_ankle
Upper clothing color	UpperCloth:Black, UpperCloth:Blue, UpperCloth:Brown, UpperCloth:Gray, UpperCloth:Green, UpperCloth:MultiColor, UpperCloth:Orange, UpperCloth:Purple, UpperCloth:Red, UpperCloth:White, UpperCloth:Yellow
Upper clothing type	UpperCloth:Coat, UpperCloth:Dress, UpperCloth:Jacket, UpperCloth:Shirt, UpperCloth:Suit, UpperCloth:Sweater, UpperCloth:T-Shirt, UpperCloth:TankTop
Upper clothing pattern	UpperCloth:Floral, UpperCloth:Graphics, UpperCloth:Plaid, UpperCloth:Plain, UpperCloth:Spotted, UpperCloth:Striped
Sleeve length	UpperCloth:LongSleeve, UpperCloth:ShortSleeve, UpperCloth:NoSleeve
Lower clothing type	LowerCloth:LongPants, LowerCloth:ShortPants, LowerCloth:LongSkirt, LowerCloth:ShortSkirt

Table 1: GCP Video Intelligence API PERSON_DETECTION Flags

I started to trim and crop the video of my fencing lunge to just the sections where the lunge action lasts and where only I could be seen in frame. I created a new Nathan Lunge project in Google Cloud Platform (GCP) console, and enabled the Cloud Video Intelligence API:

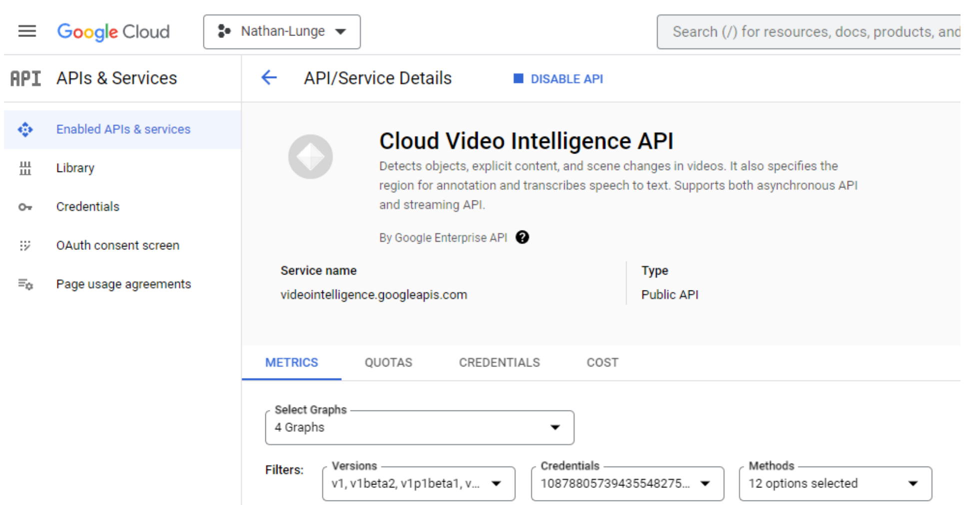

Figure 3: GCP Console for Video Intelligence API

Then I uploaded the video to Google Cloud Storage and ran it through the video Intelligence API using Google Collaborative notebook. In code it looks like this:

Fencing_Lunge_AI_Analysis.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```python
# This function comes from the docs
# https://cloud.google.com/video-intelligence/docs/people-detection
def detect_person(input_uri):
    """Detects people in a video."""

    client = videointelligence.VideoIntelligenceServiceClient(
        credentials=service_account.Credentials.from_service_account_file(
            './key.json'))

    # Configure the request
    config = videointelligence.types.PersonDetectionConfig(
        include_bounding_boxes=True,
        include_attributes=True,
        include_pose_landmarks=True,
    )
    context = videointelligence.types.VideoContext(person_detection_config=config)

    # Start the asynchronous request
    operation = client.annotate_video(
        request = videointelligence.AnnotateVideoRequest(
            features=['PERSON_DETECTION'],
            input_uri=input_uri,
            output_uri=output_uri,
            video_context=context,
        )
    )

    print("\nProcessing video for person detection annotations.")
    result = operation.result(timeout=300)
    print("\nAnnotation complete!")

    return result
```

Figure 4: detect_person Function in Python

When the Video Intelligence API finished analyzing my video, the result 2D animation model can be visualized with points for a 2D skeleton like this:

Figure 5: Visualization of the Video Intelligence API Result of Nathan's Lunge

2.3. Further Data Analysis of Position and Angle Movement

For demonstration purposes, let me present a graph showing the x-coordinate of my right knee and right ankle, versus time:

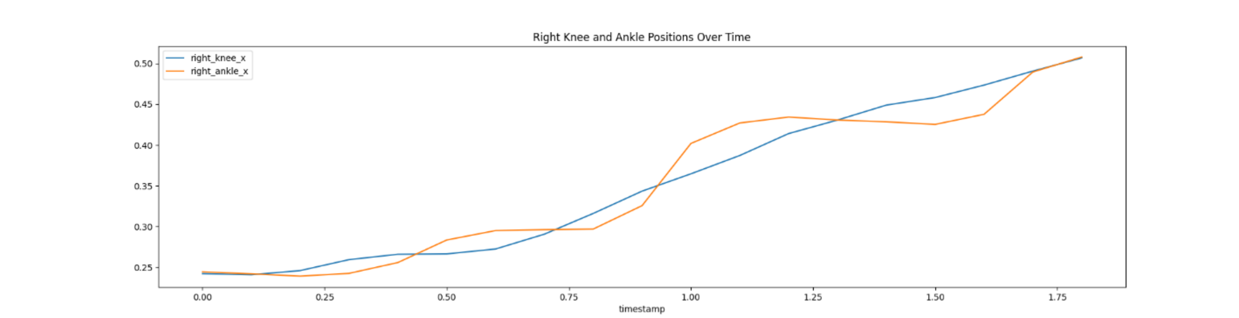

Figure 6: Nathan's Right Knee and Ankle Position Over Time

This graph helps to illustrate the x-position of my right knee and ankle, better analyzing and understanding what occurred during that action. Analyzing this data allows me to see when I extended my leg to begin the lunge, if the lunge extension was carried out correctly, and the starting and ending positions, where the x-position of the knee and ankle should be equal.

To gain a deeper understanding of my lunge, I would also like to examine different areas of my body including the angle of my right arm as I extend the weapon towards my opponent. The Law of Cosines is a useful tool for converting spatial coordinates into angles:

$$\gamma = \tan^{-1}\frac{c_y - b_y}{c_x - b_x} - \tan^{-1}\frac{b_y - a_y}{b_x - a_x}$$

In code, this might look something like this:

```python
class Point:
  def __init__(self, x, y):
    self.x = x
    self.y = y

def getAngle(a, b, c):
    ang = math.degrees(math.atan2(c.y-b.y, c.x-b.x) - math.atan2(a.y-b.y, a.x-b.x))
    return ang

def computeElbowAngle(row, which='right'):
  wrist = Point(row[f'{which}_wrist_x'], row[f'{which}_wrist_y'])
  elbow = Point(row[f'{which}_elbow_x'], row[f'{which}_elbow_y'])
  shoulder = Point(row[f'{which}_shoulder_x'], row[f'{which}_shoulder_y'])
  return getAngle(wrist, elbow, shoulder)

def computeShoulderAngle(row, which='right'):
  elbow = Point(row[f'{which}_elbow_x'], row[f'{which}_elbow_y'])
  shoulder = Point(row[f'{which}_shoulder_x'], row[f'{which}_shoulder_y'])
  hip = Point(row[f'{which}_hip_x'], row[f'{which}_hip_y'])
  return getAngle(hip, shoulder, elbow)

def computeKneeAngle(row, which='right'):
  hip = Point(row[f'{which}_hip_x'], row[f'{which}_hip_y'])
  knee = Point(row[f'{which}_knee_x'], row[f'{which}_knee_y'])
  ankle = Point(row[f'{which}_ankle_x'], row[f'{which}_ankle_y'])
  return getAngle(ankle, knee, hip)
```

Figure 7: computeAngle Function in Python

Using these formulae, I plotted the angle of my right elbow, right shoulder, and right knee over time:

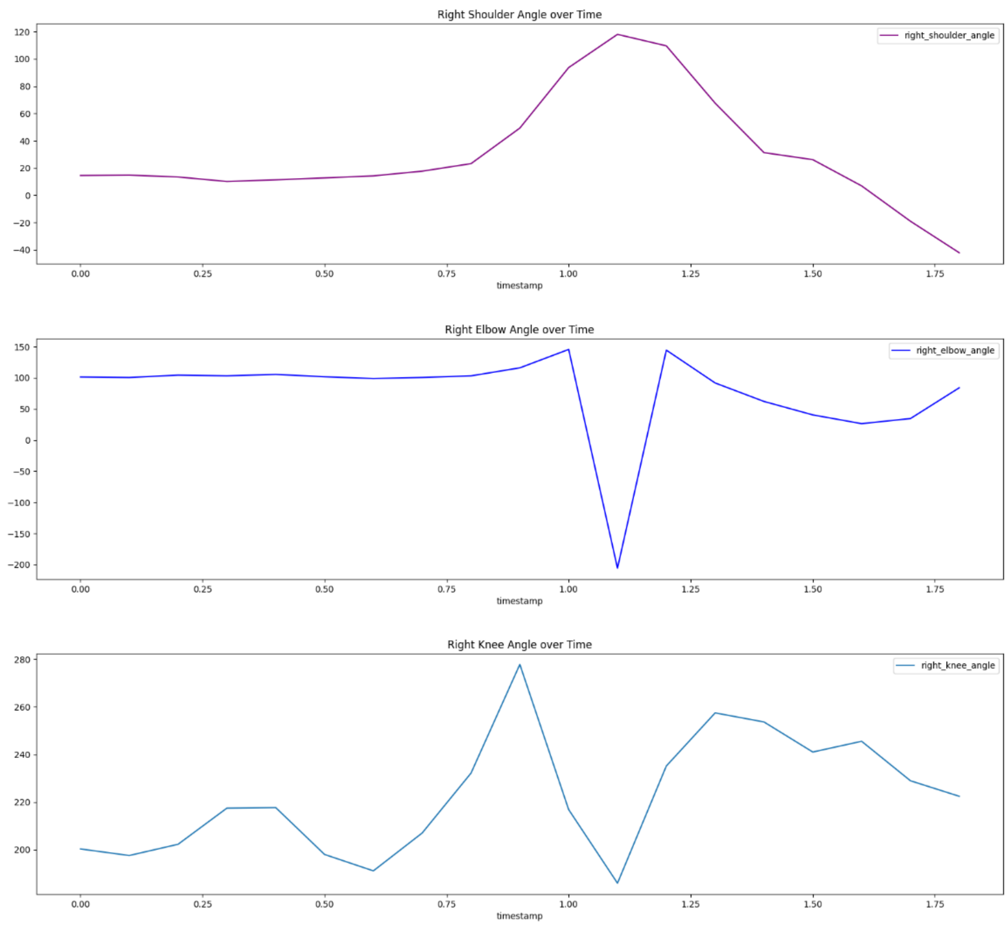

Figure 8: Nathan's Right Shoulder, Elbow, and Knee Angle Over Time

By aligning the height of my wrist and the angle of my knees, I was able to find that the angle of my front knee was greater than 90 before the lunge (at 0.6 seconds) but decreased to 90 degrees during the lunge (which can be seen at 0.6 seconds to 1.1 seconds, by calculating the explementary angle which at 270 around 0.85 seconds). I applied the same formula to measure the angles of my shoulders and knees.

Figure 9: My Right Wrist Position and Right Knee Angle Over Time

I also compared my lunge to that of Olympic champion Gergely Siklósi, and the comparison plot are shown below:

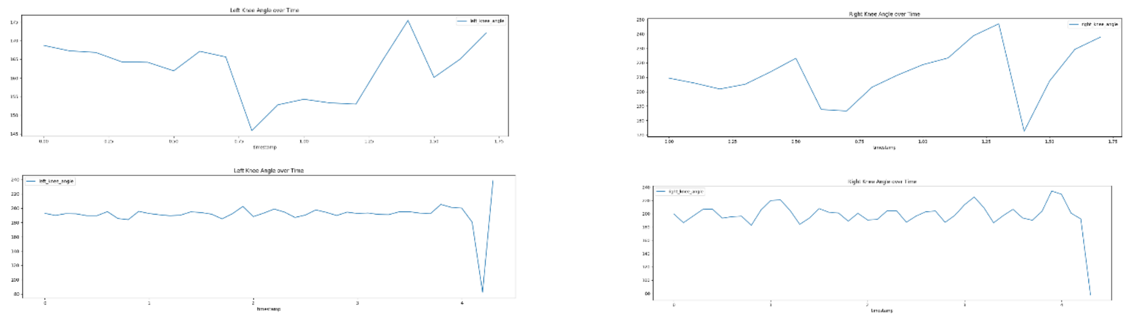

Figure 10: Top Row: Nathan's Left and Right Knee Angle;
Bottom Row: Olympic champion Siklósi's Left and Right Knee Angle

The top row is my left and right knee angle, the bottom is Olympic champion Siklósi's. It is very clear that his movement is smoother and doesn't have as much variance.

3. Analyze the 3D Posture Using Deep Motion AI Motion Capture API

DeepMotion introduces a cutting-edge solution for recording and reconstructing complete body movements, including facial expressions and hand movements. Animate 3D enables you to transform videos into 3D animations suitable for games, augmented/virtual reality, sports analysis, and numerous other purposes. You can either upload a video, choose the desired output format and job settings, and operate through the graphical user interface or utilize the A3D RESTful APIs (DeepMotion, "Animate 3D," DeepMotion. https://www.deepmotion.com/animate-3d (2022).).

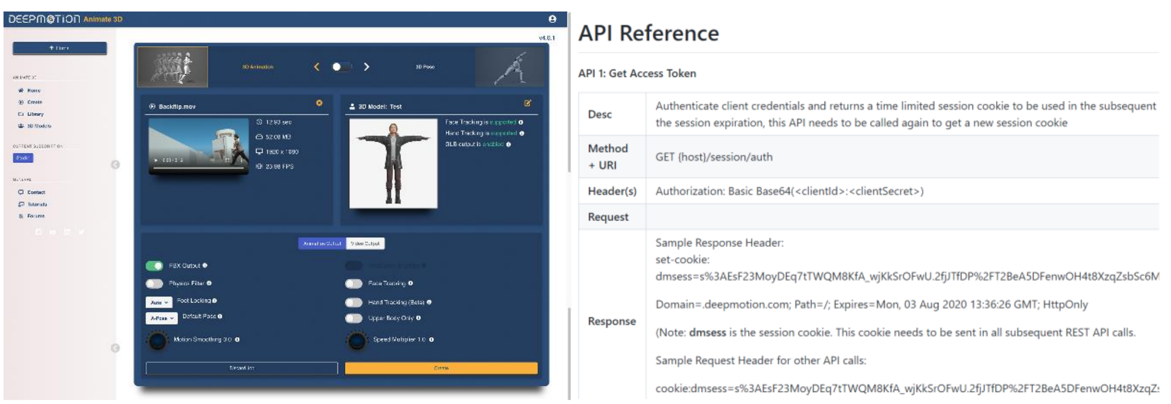

Figure 11: DeepMotion Animate 3D User Interface and API Reference

Animate 3D enables you to generate animations from video footage within minutes, significantly cutting down development time and expenses. Equipped with innovative features like Physics Simulation, Foot Locking, Hand Ground Contact, Slow Motion handling, and Face & Hand Tracking, you are empowered with greater control and versatility to produce high-quality 3D animations.

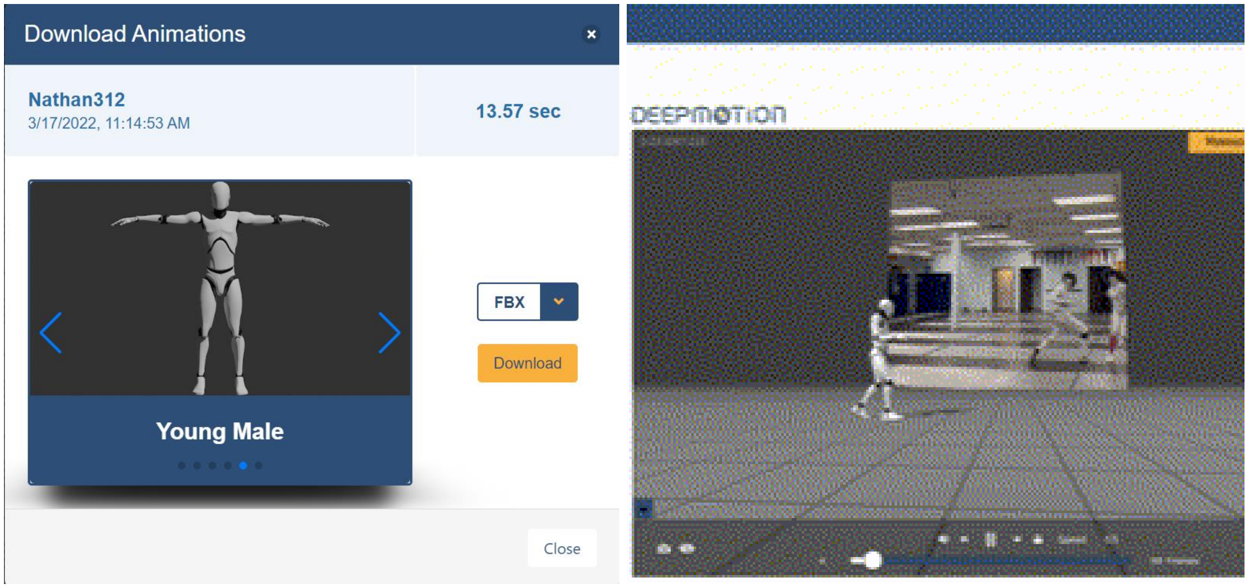

Figure 12: DeepMotion 3D Model for Nathan(Left), and the Generated 3D Animation(Right)

Upon completion of the data processing, the 3D motion data of my lunge action is recorded in the FBX (Filmbox) format. FBX is a widely used file format in the 3D modeling, animation, and visual effects fields, known for its ability to store a diverse range of data including geometry, materials, textures, animations, lights, cameras, and other scene information. The format is versatile, accommodating a variety of 3D assets from simple objects to intricate animations and scenes. This versatility makes FBX a favored choice for transferring 3D data between different stages of a project or among different teams and artists.

4. Create Oculus Quest 2 VR App using the 3D model with Unreal Engine

Unreal Engine is a popular and powerful real-time 3D game engine developed and maintained by Epic Games. It is widely used in the game development industry to create high-quality games, visual effects, and interactive experiences across multiple platforms.

Unreal Engine has many features that make it a popular choice for game developers, including a robust scripting language (Blueprints), advanced rendering capabilities, a physics engine, and extensive support for VR and AR experiences. It also has a large and active community of users, making it easier to find resources and support.

In comparison to other popular 3D engines, Unreal Engine has tons of advantages and little to no disadvantages. For example, Unreal Engine is often considered to be more powerful and feature-rich compared to other engines like Unity, but it can also have a steeper learning curve for new users. On the other hand, Unity has a simpler interface and is often favored for its ease of use and cross-platform support.

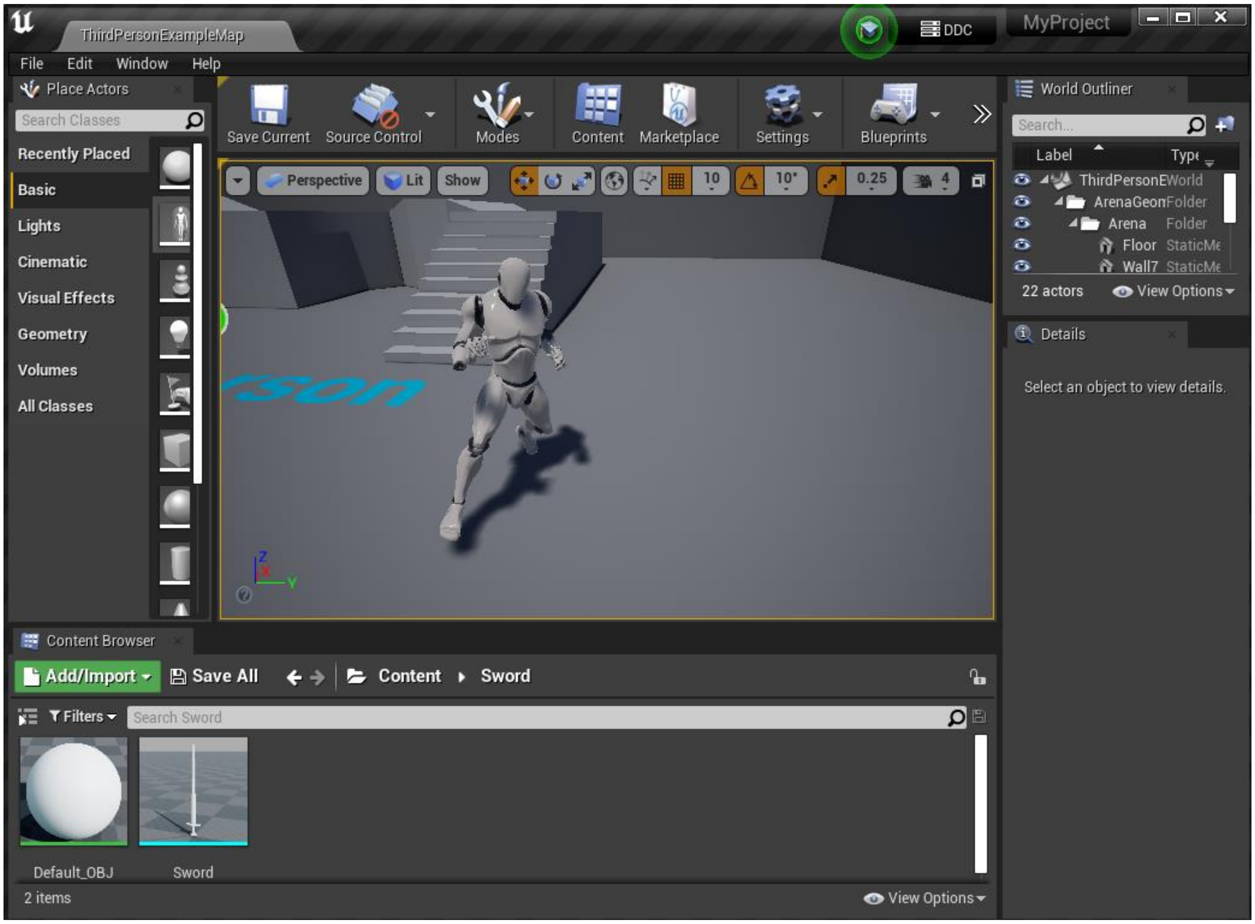

Figure 13: Generate the 3D Animation in Unreal Engine for VR Applications

I must admit that the learning curve is very steep, and I watched several YouTube video tutorials to learn how to build VR apps using UE4. Then setup the build environment using Android Studio and finetune the configuration. To build a VR application for Oculus, the following steps are taken:

1. Create a new project in Unreal Engine and select the Games template category and then the Virtual Reality template.
2. Import the DeepMotion FBX model into Unreal Engine. Load the target character and import the animation.
3. Once the model is imported, I adjusted its properties and materials as needed in the Unreal Editor, such as modifying the mannequin from the A-pose to T-pose and retarget the animation ("Tutorial on Building Quest 2 VR apps with Unreal Engine (Win)," YouTube, https://www.youtube.com/watch?v=veKyDnzb7-k (2022).).
4. Next, I set up the VR environment, by installing Android Studio, Visual Studio, Android SDK and NDK, and Java JDK. And make many configuration change according to the YouTube tutorial video ("Tutorial on using Deepmotion AI motion capture animation retargeting on Unreal Engine," YouTube, https://www.youtube.com/watch?v=j5wfOF1JrJA&t=0s (2023).).

5. With the VR environment set up, I installed Side Quest app so that I can side load the compiled application to my Oculus 2 as the target device.
6. I built the VR application from Unreal Engine and once the project compiled, I deployed the VR application to my Oculus 2 device for testing and use. Below is the screenshot from the Oculus virtual world:

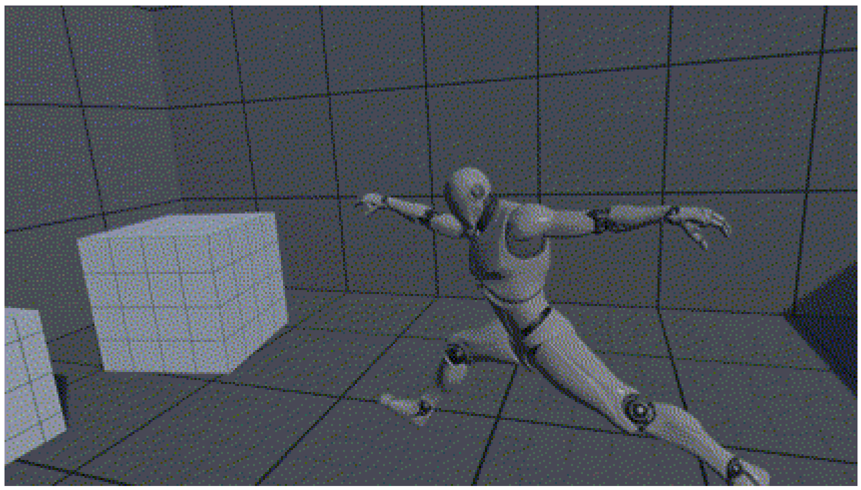

Figure 14: Meta Oculus VR Experience of Nathan's Lunge

5. Conclusion

In this study, data was gathered on the movements of fencers during the lunge and analyzed them by utilizing 2D and 3D AI algorithms to pinpoint areas for improvement. A VR system was employed to simulate various fencing scenarios. The findings of the analysis could be utilized to create a training program that is beneficial for both beginner and intermediate fencers.

Upon sharing the results with fellow fencers and my fencing coach, there was consensus that utilizing AI and VR technologies to develop the training program could significantly enhance the performance of the lunge. This would result in more accurate and faster lunges, and also reduce the risk of sports-related injuries caused by improper lunge positions.

6. Future Work

1. Future work can expand upon this study by utilizing AI and VR to analyze other techniques other sports.
2. This project can also be used for other fencing movements that are more complex in the future as the lunge is a basic building block for future combos.
3. Another avenue for exploration would be to use the API output, which provides information on the position of the joints over time, as input features to a machine learning model that predicts the success of the lunge.

Acknowledgements

I wish to express my sincere gratitude to Coach Mario for his valuable comment to this manuscript. And deeply appreciate the time and effort from all these reviewers dedicated to reviewing my work:

- Mr. Kevin He, CEO of DeepMotion. kevin.he@deepmotion.com
- Dr. Fengnian Xia, Prof of EE at Yale University. fengnian.xia@yale.edu
- Mr. Mario Jelev, Coach of South Florida Fencing Club. jelevepee@gmail.com

References

[1] J. Kim, "Artificial Intelligence in Sports Training," AI Review, vol. 7, pp. 23-30, 2018.

[2] S. Su, Y. Zhang, and L. Wang, "A Virtual Reality System for Fencing Training," Journal of Virtual Reality, vol. 18, pp. 5-12, 2013.

[3] Google, "Video Intelligence API documentation," Google Cloud. [Online]. Available: https://cloud.google.com/video-intelligence/docs.

[4] Deepmotion, "Animate 3D," Deepmotion. [Online]. Available: https://www.deepmotion.com/animate-3d.

[5] "Tutorial on Building Quest 2 VR apps with Unreal Engine (Win)," YouTube, 2022. [Online Video]. Available: https://www.youtube.com/watch?v=veKyDnzb7-k.

[6] "Tutorial on using Deepmotion AI motion capture animation retargeting on Unreal Engine," YouTube, 2023. [Online Video]. Available: https://www.youtube.com/watch?v=j5wfOF1JrJA&t=0s.