Generalized Server for using Motion Capture Data in Virtual Reality Applications

Amey Edwankar | Dr. Joe Geigel

Introduction

• The rise of affordable VR and AR devices have allowed to the creation of new immersive experiences which were not possible earlier.
• With the use of motion capture devices like the Microsoft Kinect we can develop new experiences which allows one to be teleported into new environments.
• The objective of this project is to design and implement a generalized server to send and receive full body motion capture data.
• The motion capture will be performed at a capture site where performers will perform in front of a Microsoft Kinect.
• The motion capture data is streamed to a server which has a unity application.
• The server will render all the motion capture data in the form of a 3D character which will mirror the movement of the performers.

Current Status and Future Work

• Currently the system can only track one performer and render their movement onto a character.
• Successfully able to capture motion capture data for every frame.
• Successfully leveraged Unity’s networking suite to stream data between capture site and server.
• Successfully rendered the captured motion data from the capture site at the server site.

Technologies Used

Microsoft Kinect:
• Device used for capturing the motion of the performers.
• Generates various feeds from its sensors like RGB video, infrared video, and raw depth feed which can be used to capture various.
• Creates a skeletal frame which joins all the important tracked positions on a body.

Unity:
• Game engine which allows the development of video games and simulations.
• Contains plugin for Kinect which allows motion capture and rendering motion on characters.
• Possesses robust networking suite allowing to stream data between two sites.

Performance Capture

Streaming and Rendering Motion Capture

• Once performance is captured the data containing the 3D coordinates of each body point for a frame is written to a dictionary and serialized.
• The serialized dictionary is streamed to the server site using Unity’s LLAPI(Lower Level API) which uses UDP on the transport layer.
• Server deserializes received serialized dictionary containing the 3D coordinates of the body points.
• Each body points 3D coordinates are extracted and applied to a 3D character.
• The above process is repeated for every frame allowing the 3D character to mirror the performer’s movement.

Results

• The above images are the user interface at the clients side which accepts the IP address of the server and the rendering of the captured performance data at the servers site.

Selected References

2. N3K EN, Unity Multiplayer (UNET) - https://www.youtube.com/watch?v=qGkkaNkq8co

Current Status and Future Work

• Currently the system can only track one performer and render their movement onto a character.
• Currently the system only takes input from one Kinect sensor and one capture site.
• In the future we would like to extend the system to enable it to receive data from multiple capture sites and render the multiple performers in the same scene while maintaining a relative position.