Combining BCIs with AR/VR: The Openvibe/Unity duet

Hakim Si-Mohammed, University of Lille
(hakim.simohammed@univ-lille.fr)
Communicating and interacting in virtual environments through “thoughts” is a widespread fantasy

Q: How is it possible to interact with virtual elements through brain activity?
Agenda

Virtual and Augmented Reality
Combining BCIs with AR/VR
Unity
Connecting Openvibe to Unity
Useful Links
Virtual reality is a scientific domain that uses computer science and interaction interfaces in order to simulate, in a virtual world, the behaviour of 3D entities that are interacting in real time with themselves and with one or more users. The user’s sensory-motor channels are engaged in a pseudo-natural immersion.

[Arnaldi et al., 2006]
➢ Augmented Reality relates to the integration of virtual objects and information in the real world in real-time [Zhou et al., 2008, Azuma, 1997] :
   1. The combination of real and virtual content.
   2. The Real-Time interaction.
   3. The 3D registration of the virtual content in the real environment.
Why to combine VR and BCIs?
Combining Brain-Computer Interfaces and Virtual/Augmented Reality

- For interacting with the virtual environment itself:

[lotte et al., 2012]
Combining Brain-Computer Interfaces and Virtual/Augmented Reality

- For interaction with physical objects using AR:

  [Si-Mohammed et al.; 2018]  [Si-Mohammed et al.]
Combining Brain-Computer Interfaces and Virtual/Augmented Reality

- For estimating user experience in VR:

  [Argelaguet et al., 2015]

  [Jeunet et al., 2018]
Combining Brain-Computer Interfaces and Virtual/Augmented Reality

- For better control of the experimental environment

[Si-Mohammed et al.; 2020] [Si-Mohammed et al.; 2018]
Combining Brain-Computer Interfaces and Virtual/Augmented Reality

- For better and more engaging feedback modalities

[Rimbert et al.; 2018]
Connecting openvibe to unity
Unity: An easy-to-take-on game engine

- Cross-platform game engine
- Offers a scripting API in C#
- Supports most of the VR hardware ecosystem
- Lots of tutorials and support is available online:

  https://learn.unity.com
Most scripts are built around 2 methods (functions):

```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Rotate : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        
    }

    // Update is called once per frame
    void Update()
    {
        
    }
}
```
How to connect Openvibe with unity

- First of all: the hardware considerations
How to connect Openvibe with unity

- The general architecture

```
Acquisition and Preprocessing → Feature Extraction → Classification
```

Feedback → unity
How to connect Openvibe with unity

- Three supported communication protocols:
  1. VRPN : Virtual-Reality Peripheral Network
  2. TCP : Transmission Control Protocol
  3. LSL : Lab Streaming Layer
TCP is based on a client/server architecture:
- Opencvibe plays the role of the server and delivers the messages
TCP is based on a client/server architecture:
- Openvibe plays the role of the server and delivers the messages
- Unity connects to the server and listens to stimulation to act
How to connect Openvibe with unity

- TCP is based on a client/server architecture:
  - Openvibe plays the role of the server and delivers the messages
  - Unity connects to the server and listens to stimulation to act

```csharp
public class TCPTestClient : MonoBehaviour {
    #region private members
    private TcpClient socketConnection;
    private Thread clientReceiveThread;
    #endregion

    // Use this for initialization
    void Start () {
        ConnectToTcpServer();
    }

    private void ConnectToTcpServer () {
        try {
            clientReceiveThread = new Thread(new ThreadStart(ListenForData));
            clientReceiveThread.IsBackground = true;
            clientReceiveThread.Start();
        }
        catch (Exception e) {
            Debug.LogError("On client connect exception " + e);
        }
    }
}
```
How to connect Openvibe with unity

- TCP is based on:
  - Openvibe protocol
  - Unity connection

```csharp
private void ListenForData() {
    try {
        socketConnection = new TcpClient("localhost", 8052);
        Byte[] bytes = new Byte[1024];
        while (true) {
            // Get a stream object for reading
            using (NetworkStream stream = socketConnection.GetStream()) {
                int length;
                // Read incoming stream into byte array.
                while ((length = stream.Read(bytes, 0, bytes.Length)) != 0) {
                    var incomingData = new byte[length];
                    Array.Copy(bytes, 0, incomingData, 0, length);
                    // Convert byte array to string message.
                    string serverMessage = Encoding.ASCII.GetString(incomingData);
                    Debug.Log("server message received as: " + serverMessage);
                }
            }
        }
        catch (SocketException socketException) {
            Debug.Log("Socket exception: " + socketException);
        }
    }
}
```
How to connect Openvibe with unity

- TCP is based on a client/server architecture:
  - Openvibe plays the role of the server and delivers the messages
  - Unity connects to the server and listens to stimulation to act

- TCP is to use when there is no hard constraint on the timings and the communication is (mostly?) one way
The lab streaming layer (LSL) is a system for the unified collection of measurement time series in research experiments that handles both the networking, time-synchronization, (near-) real-time access.

It is based on the streaming of several sources of data/stimulation over the network, synchronized to a common server.
LSL: One *protocol* to rule them all

- Download the unity project

[Hello World](https://gitlab.inria.fr/openvibe/unity-games/hello-world)

First Example Of Unity Games For OpenViBE.

![GitLab Interface](https://gitlab.inria.fr/openvibe/unity-games/hello-world)

**Update Readme Dependency link**

MONSEIGNE Thibaut authored 5 months ago

- [README](https://gitlab.inria.fr/openvibe/unity-games/hello-world/README)
- [GNU AGPLv3](https://gitlab.inria.fr/openvibe/unity-games/hello-world/LICENSE)
- [CONTRIBUTING](https://gitlab.inria.fr/openvibe/unity-games/hello-world/CONTRIBUTING)
- [CI/CD configuration](https://gitlab.inria.fr/openvibe/unity-games/hello-world/.gitlab-ci.yml)

**Name** | **Last commit** | **Last update**
---|---|---
Assets | Update Dependency LSL4Unity | 6 months ago
Documentation | Update Dependency LSL4Unity | 6 months ago

**Controlled Keywords:**
- LSL
- Communication
- Protocol

**Recommended Links:**
- [OpenViBE](https://openvibe.neuroimage.fr)
- [CRIStAL](http://www.cristal.univ-lille1.fr)
- [Neuroergonomics Conference 2021](https://neuroergonomics-conference.org/2021/)

---

**Acknowledgments:**

- [Neuroergonomics Conference 2021](https://neuroergonomics-conference.org/2021/)
- [CRIStAL](http://www.cristal.univ-lille1.fr)
LSL: One *protocol* to rule them all

The scripts responsible for receiving data from openvibe
LSL: One *protocol* to rule them all

But in order to make life easier for the community, a single protocol must be created or not. If several inputs must be used.

**Dependency**

- LSL4Unity

**Screenshots**

Link to the LSL Library

Download the project folder
Add the project in the Asset folder of the Unity project
The method in the « controller » script where the labels are listened for
The lab streaming layer (LSL) is a system for the unified collection of measurement time series in research experiments that handles both the networking, time-synchronization, (near-) real-time access.

It is based on the streaming of several sources of data/stimulation over the network, synchronized to a common server.

- LSL is to be used when you have strong constraints on the synchronization
- LSL comes in handy when syncing multiple data sources
Useful links

- [https://gitlab.inria.fr/openvibe/unity-games/hello-world](https://gitlab.inria.fr/openvibe/unity-games/hello-world) (The Unity project for LSL)
- [https://gitlab.inria.fr/openvibe/unity-games/LSL4Unity](https://gitlab.inria.fr/openvibe/unity-games/LSL4Unity) (The LSL4Unity library)

- [https://gist.github.com/danielbierwirth/0636650b005834204cb19ef5ae6cceddb](https://gist.github.com/danielbierwirth/0636650b005834204cb19ef5ae6cceddb) (The TCP client for Unity)
Good luck!

Hakim Si-Mohammed (hakim.simohammed@univ-lille.fr)
Twitter: @HakimSMH