DANCE

D4.1 – DANCE System Platform, First version

Version	Edited by	Changes
0.1	UNIGE	first version

TABLE OF CONTENTS

1. Introduction

The *DANCE software platform,* based on the EyesWeb XMI research platform, allows synchronized recording, playback, and analysis of a multimodal stream of data.

The platform supports the following research activities of DANCE:

- the creation of a multimodal repository of recordings of movement qualities;
- the fine-grain synchronization of multimodal data
- the segmentation of the recordings in fragments, according to the chosen qualities
- the playback and testing of the repository
- the extraction of the movement features and qualities
- the real-time interaction sonification
- the design and development process of scientific experiments of DANCE
- the design and development of the prototypes of applications
- the design and development of artistic projects exploiting the results of the DANCE project (e.g., artistic performances)

In this deliverable you can find instructions necessary to download, install and run the DANCE software platform, and details about the file data formats exploited by the platform. The platform can be freely downloaded from the website:

http://dance.dibris.unige.it/index.php/dance-platform

The platform is based on EyesWeb XMI, allowing users to perform synchronized recording, playback, and analysis of a multimodal stream of data. EyesWeb XMI is a modular system that allows both expert (e.g., researchers in computer engineering) and non-expert users (e.g., artists) to create multimodal installations in a visual way. EyesWeb provides software modules, called blocks, that can be assembled intuitively (i.e., by operating only with mouse) to create tools and programs, called patches, that exploit system's resources such as multimodal files, webcams, sound cards, multiple displays and so on.

The last version of EyesWeb is the 5.6.0.0. (released in February 2016). You can download it from the following link: ftp://ftp.infomus.org/Evaluate/EyesWeb/XMI/Nightly/trunk/EyesWeb_XMI_setup_5.6.0.0.exe

To support the DANCE project, EyesWeb has been integrated with a number of modules, tools and patches which are presented in this document.

2. Synchronized multimodal recording and playback

2.1 Architecture

The overall architecture for multimodal recordings is shown in the following figure:

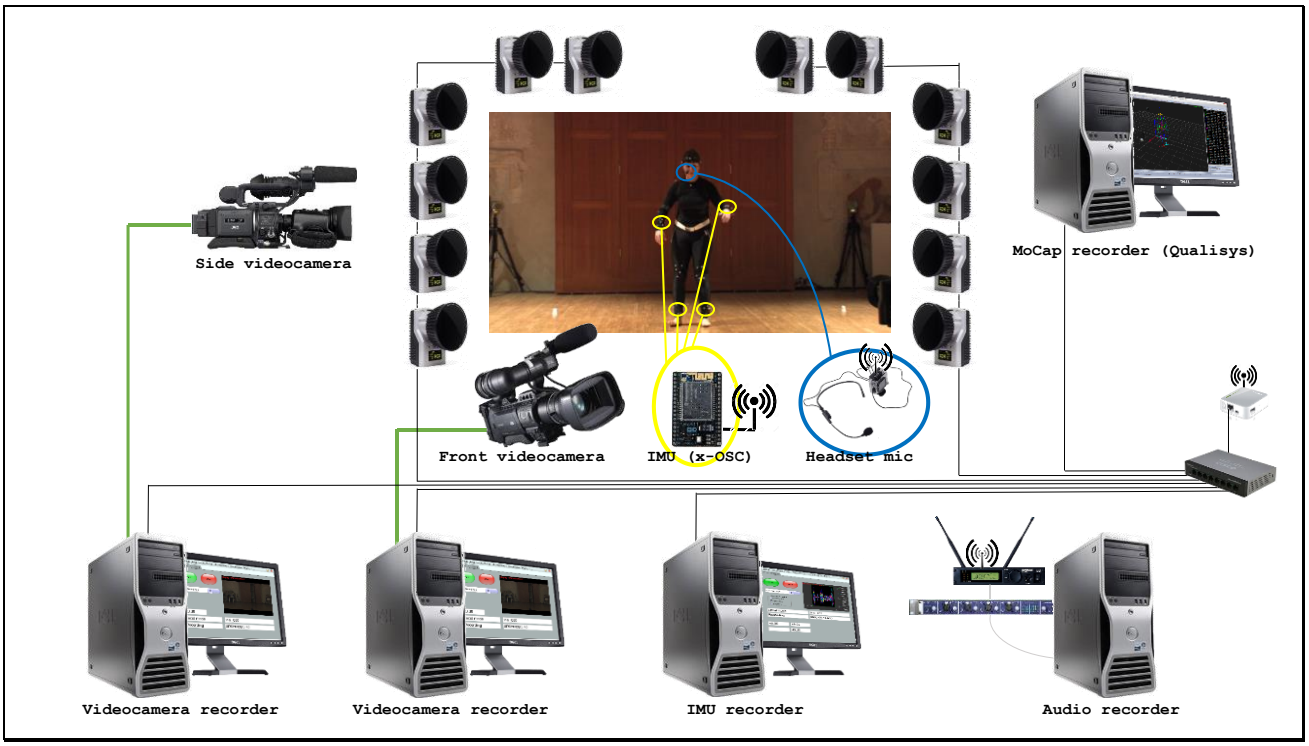


Figure 1. The architecture of the DANCE platform.

The performer movements are captured by a Motion Capture system (*MoCap recorder* in the picture). The performer also wears a headset microphone, which is used to record breathing noise for possible further analysis. Moreover, the performer also wears four Inertial Measurement Units (IMU), two on the wrists and two on the ankles. Finally, two broadcast quality video cameras are observing the scene, one from the front and one from a side.

Synchronization is guaranteed by the EyesWeb platform. On the MoCap recorder computer, EyesWeb is used to generate the reference clock used by all other recorders. The generated reference clock is sent to the other device in a format compatible with each specific device. As an example, the Qualisys Motion Capture system receives such clock encoded in an audio stream, in SMPTE format. Also the two broadcast video-cameras and the *Audio recorder* use SMPTE encoded as an audio signal. The *IMU recorder* receives the reference clock via network, through the OSC protocol.

To guarantee synchronization EyesWeb keeps track, for every recorded frame or sample, of the timestamp when the data was received. As a matter of facts, not all streams can be hardware-synchronized (e.g., with a genlock signal), thus, a software synchronization is performed by EyesWeb by keeping track of the time at which the data was received in a separate file, and using such information when playing back the data. IMU sensors or Kinect are examples of devices which are synchronized in this way.

The DANCE platform tools and patches are programs, written to be execute by EyesWeb, that allow the user to record, playback and analyze multimodal data (video, audio, motion capture, sensors). To run tools you will need to download the corresponding installers, launch them and execute the tools as normal Windows applications. To run patches you will need to download and load them into the EyesWeb application. The current version of the DANCE example tools and patches includes applications allowing you to perform different tasks:

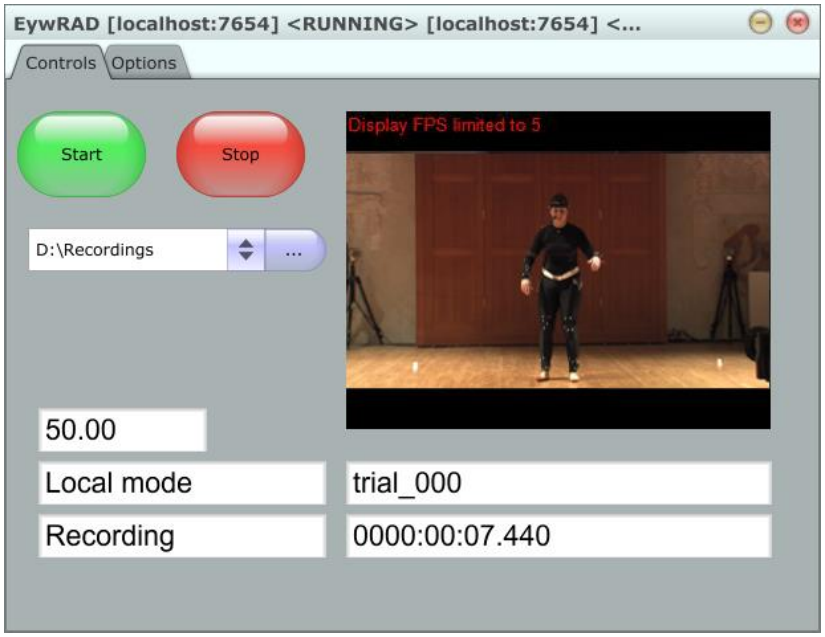
1. to record and playback multimodal data
2. to analyze multimodal data

In the next sections we illustrate some example tools and patches belonging to both of the above categories.

2.2 Kinect recorder tool

Download:
http://dance.dibris.unige.it/user_files/DANCE_Platform/release_march_2016/Eyesweb_XMI_Recorder_For_Kinect_V2_setup_5.5.0.0.exe.zip

The Kinect recorder tool is depicted in the following picture.



The tool shows the current framerate (50.00 frames per second in the example), the name used for this trial (`trial_000`, progressive numbers are automatically assigned to each trial), and the value of the reference clock (`HHHH:MM:SS.mmm; 0000:00:07.440` in the above picture).

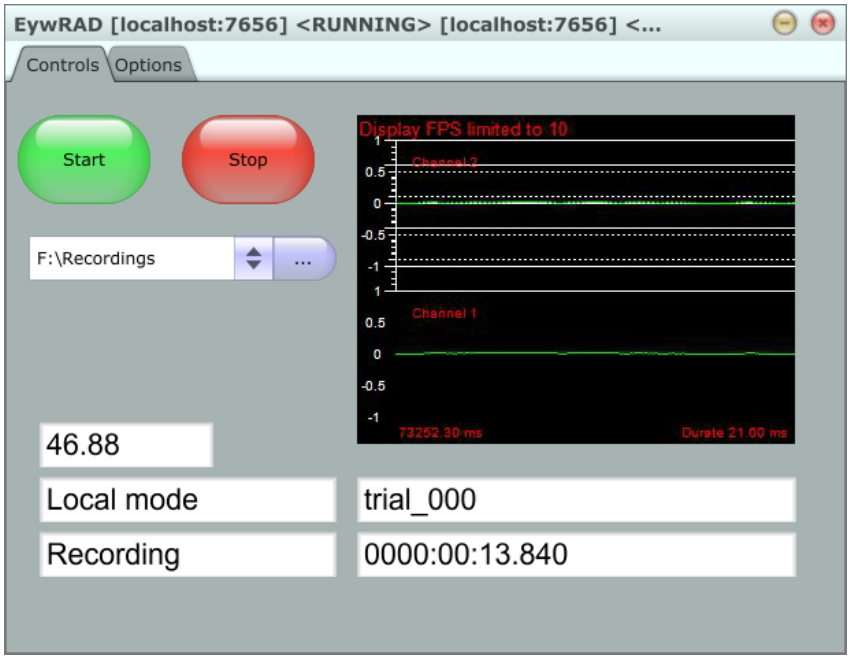
The recording tool records avi files. The video is encoded in MPEG4 format, the resolution is 1280x720 and the framerate is 50 fps. Audio is encoded in AAC format at 48000Hz. Two channels are recorded: the left channel contais audio from the system's audio input device (e.g., a microphone), whereas the right channel is the reference clock encoded in SMPTE audio format.

Multiple instances of the video recorder tool can be started and can work standalone, or synchronized with the other recorders. The options panel (see Section 2.5) allows you to configure the working mode of the recorder.

2.3 Audio recorder

Download:
http://dance.dibris.unige.it/user_files/DANCE_Platform/release_march_2016/Eyesweb_XMI_Recorder_for_ASIO_Audio_setup_5.5.0.0.exe.zip

The audio recorder tool is depicted in the following picture.



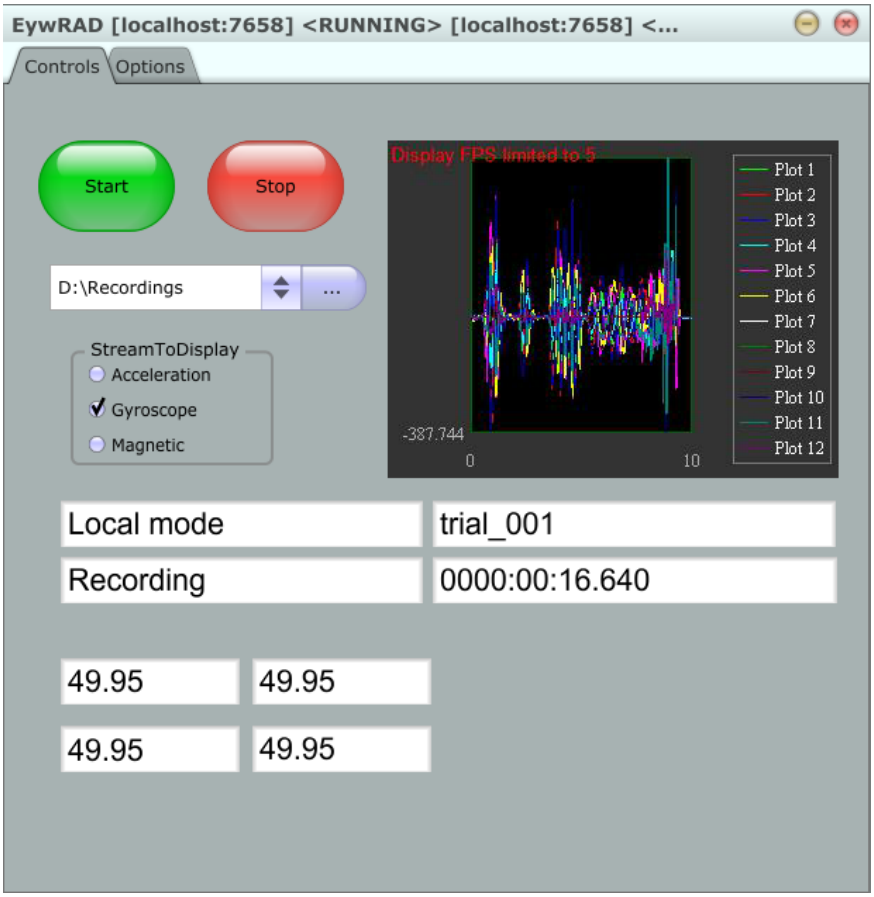
Currently, only the headset microphone is being recorded. The headset microphone provides a monophonic signal, but it is saved as a stereo file in WAVE format, as the SMPTE is added as the right channel. Audio is sampled at 48000Hz. The user interface is very similar to the video recorder tool. The main difference being of course the viualization part. In this tool the audio waveform is shown instead of the video stream. The options panel (see Section 2.5) allows you to configure the working mode of the recorder.

2.4 IMU recorder

Download:

http://dance.dibris.unige.it/user_files/DANCE_Platform/release_march_2016/Eyesweb_XMI_Recorder_for_XOSC_Sensors_setup_5.5.0.0.exe.zip

The *IMU recorder* tool is depicted in the following picture.

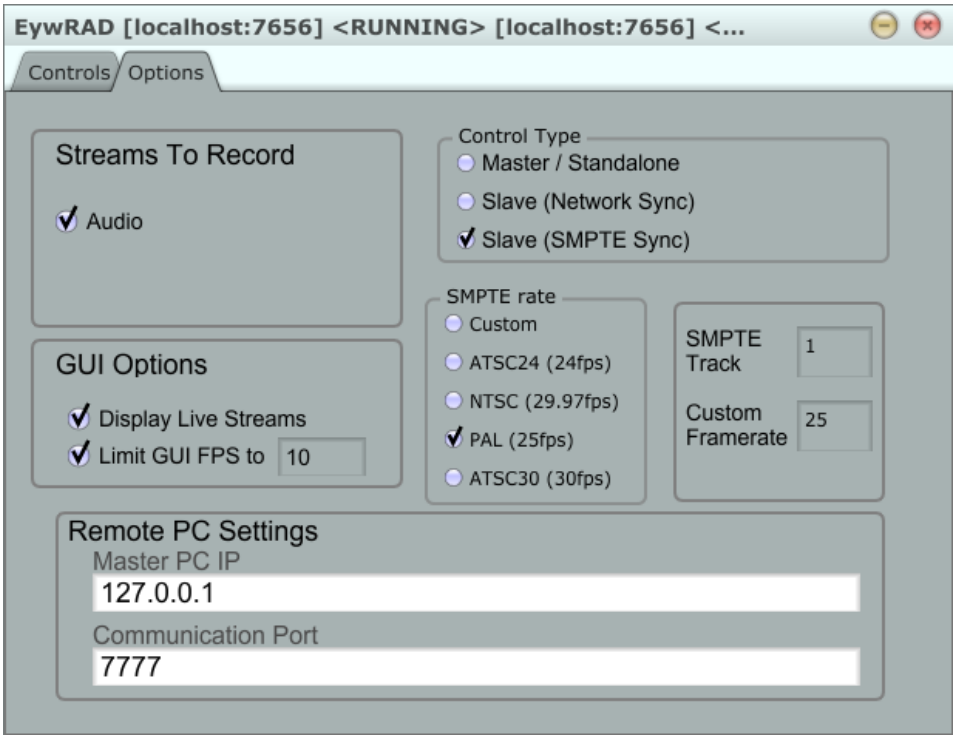


The graph shows the values selected by the user (acceleration, gyroscope, or compass) for each of the four IMUs. In the lower left you may read the current framerate of each of the four sensors (49.95 samples per second). Below the graph you may see both the trial name and the reference clock.

The data is saved by the recording tool in txt files, in a format which is easy to be read by external software (e.g., Matlab), and can be of course read by EyesWeb itself for playback or analysis purposes. The options panel (see Section 2.5) allows you to configure the working mode of the recorder.

2.5 Options panel

All the recording tools share the same options panel depicted below:

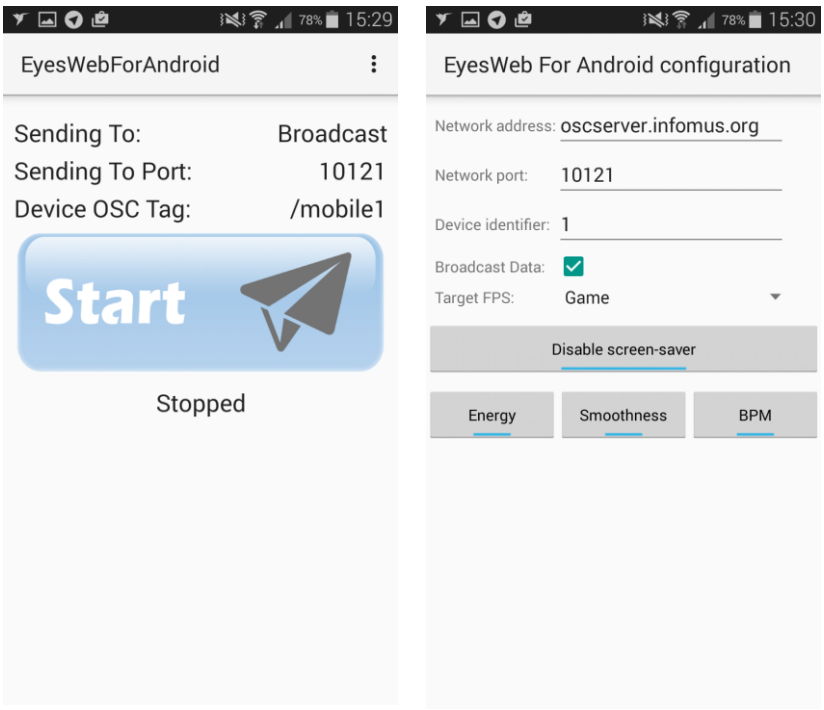


The control type section controls the synchronization mode.

If set in *master/standalone* mode the tool works with its own clock, without synchronizing to other devices. In slave mode the tool receives the clock from an external device (the master). The clock can be received both via Network (OSC protocol) and via audio (SMPTE Sync).

2.6 The EyesWebForAndroid app

It is an Android client application allowing the computation of some features directly on the mobile device. This enables the creation of more portable setups.



The features computed are Energy, fluidity, and bpm (Beats Per Minute).

2.7 The EywSensorToOsc app

Another app is also available on the PlayStore, called EywSensorToOsc, which sends the raw sensor data (accelerometer, gyro, etc) to OSC.

2.8 Playback patch

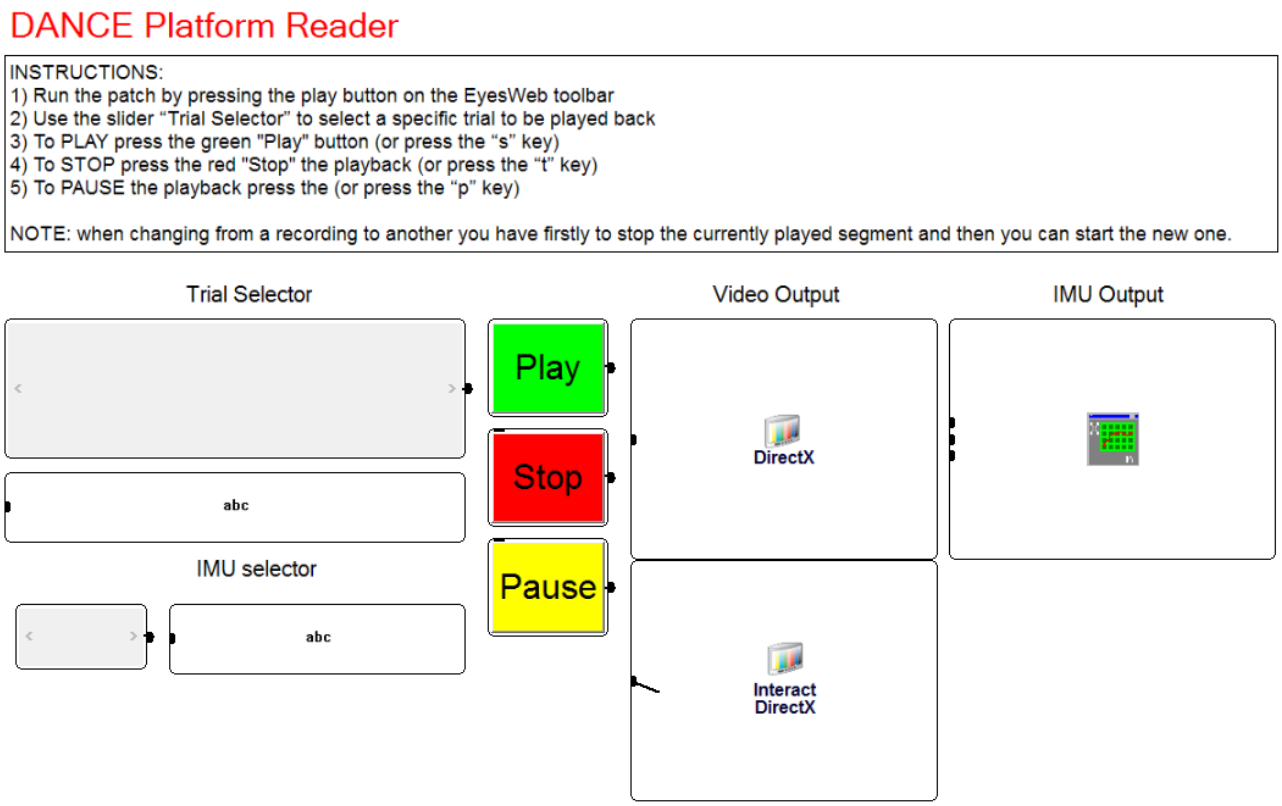
Download:
http://dance.dibris.unige.it/user_files/DANCE_Platform/release_march_2016/DANCE_platform_reader.zip

Once you recorded some audio, video and IMU data, you can play it back using the playback patch.

Download the patch and copy the downloaded file to the parent folder of the recorded data. If you did not record any data you can download some sample data from this website.

For example, if the recorded/downloaded data is located in:
`C:\Users\my_username\recordings`
then the patch path will be:
`C:\Users\my_username\recordings\DANCE_platform_reader.zip`

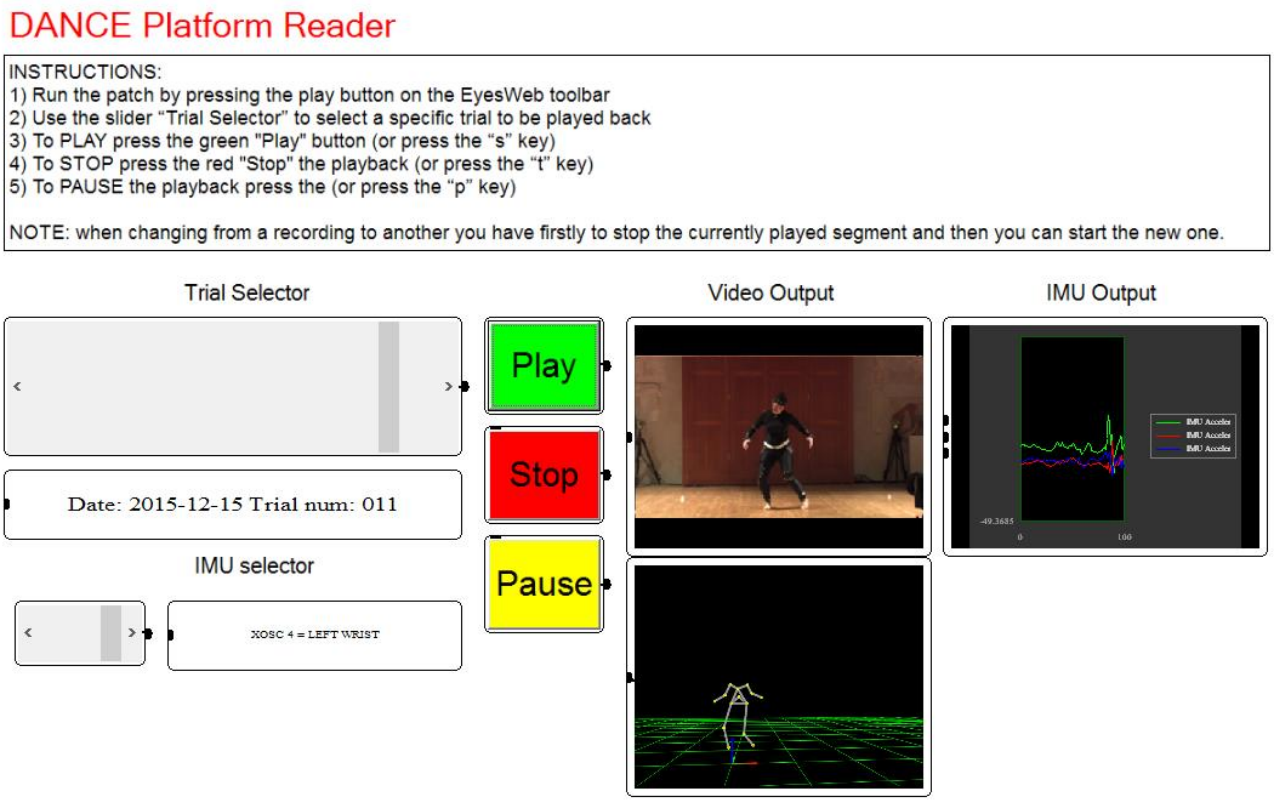
Start EyesWeb and open the patch. You will see the following screen:



1) Run the patch by pressing the play button on the EyesWeb toolbar

2) Use the slider "Trial Selector" to select a specific trial to be played back
3) To PLAY press the green "Play" button (or press the "s" key)
4) To STOP press the red "Stop" the playback (or press the "t" key)
5) To PAUSE the playback press the (or press the "p" key)

When changing from a recording to another you have firstly to stop the currently played segment and then you can start the new one. During the playback you will see video, 3D and IMU data as in the following example:



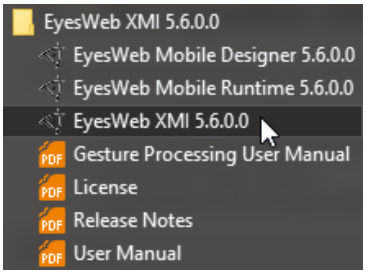
3. Sample multimodal data & patches for analyzing it

The DANCE platform provides a palette of patches to perform multimodal data analysis. Sample data is also available to test the platform.
Details about patches and sample data are provided in Deliverable 2.1 as well as on the webpage:
http://dance.dibris.unige.it/index.php/dance-platform

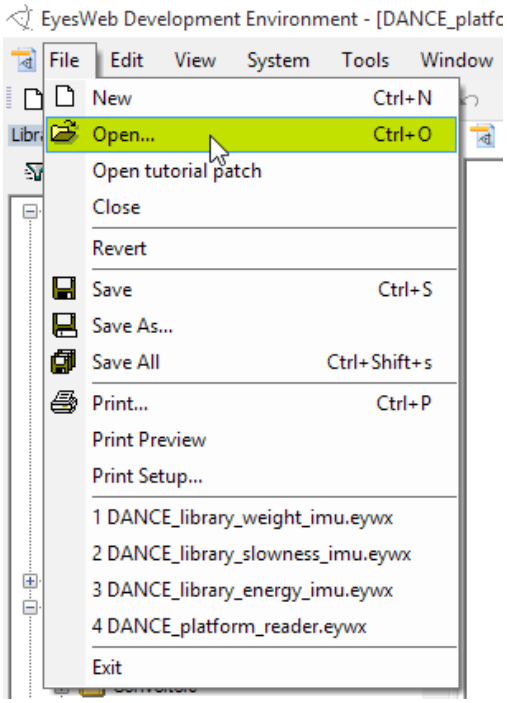
4. Run EyesWeb XMI, load one or more patches and execute them

When you downloaded EyesWeb, you installed it and you downloaded some example patches plus the needed sample data you are ready to run the patches:

1. run EyesWeb, by clicking on the corresponding shortcut in the start menu:



2. load an example patch: select the file->open menu item; browse for the patch file; select "open":



3. click on the "play" button:

4. depending on the example patch your executing, different display windows will appear

Appendix A: Synchronized data formats

A.1 Filenaming convention

The following filenaming convention has been adopted when segmenting the recordings:

```
YYYY-MM-DD (main folder)
   |
   |----Recording notes.xls (Notes taken manually during the recording, e.g.,
   |                           instructions given to performer)
   |
   |----mocap
   |       |
   |       +--- YYYY-MM-DD_tHHH_mocap.qtm
   |       +--- YYYY-MM-DD_tHHH_mocap.qam
   |       +--- YYYY-MM-DD_tHHH_mocap.tsv
   |
   |----audio
   |       |
   |       +--- YYYY-MM-DD_tHHH_audioII.avi
   |
   |--- kinect
   |       |
   |       +--- YYYY-MM-DD_tHHH_kinectK_lsb.avi
   |       +--- YYYY-MM-DD_tHHH_kinectK_msb.avi
   |       +--- YYYY-MM-DD_tHHH_kinectK_3d.tsv
   |       +--- YYYY-MM-DD_tHHH_kinectK_video.avi
   |       +--- YYYY-MM-DD_tHHH_kinectK_video_timings.txt
   |       +--- YYYY-MM-DD_tHHH_kinectK_blob.avi
   |       +--- YYYY-MM-DD_tHHH_kinectK_blob_timings.txt
   |
   |----video
   |       |
   |       +--- YYYY-MM-DD_tHHH_videoJJ.avi
   |       +--- YYYY-MM-DD_tHHH_videoJJ_timings.txt
   |
   |----imu
   |       |
   |       +--- YYYY-MM-DD_tHHH_imuNN_acceleration.txt
   |       +--- YYYY-MM-DD_tHHH_imuNN_gyro.txt
   |       +--- YYYY-MM-DD_tHHH_imuNN_magnetic.txt
   |----pics
   |
   |----segments (two versions)
          +--- YYYY-MM-DD_tHHH_UU_segments.txt (version 1)
          +--- YYYY-MM-DD_UU_segments.txt  (version 2)
```

where:

1) `YYYY-MM-DD` codifies the recording date
2) `tHHH` is the trial number (e.g., `t001` is the first trial)
3) in `sTT-KKK`
   a) `TT` is the segmentation type (e.g.,: 1 = fluidity, 2 = rigidity, ...)
   b) `KKK` is the segment number extracted from the `tHHH` (e.g., `t001_s001` is the first segment extracted from the first trial)

Details on Segments Files:

- `YYYY-MM-DD_tHHH_sTT-KKK_skeleton.avi`
  - 3d visualization of mocap data, rendered by EyesWeb

- `YYYY-MM-DD_tHHH_sTT-KKK_kinectK_lsb.avi`
- `YYYY-MM-DD_tHHH_sTT-KKK_kinectK_msb.avi`
  - Since Kinect depthmap is captured with more than 8 bits per pixel, it is stored in two files: `lsd` contains the less significant bits, `msb` contains the most significant bits
- `YYYY-MM-DD_tHHH_sTT-KKK_kinectK_3d.tsv`
  - Coordinate of the skeleton, as computed by Kinect
- `YYYY-MM-DD_tHHH_sTT-KKK_kinectK_video.avi`
  - Video from the Kinect webcam
- `YYYY-MM-DD_tHHH_sTT-KKK_multi.avi`
  - Composition of two or more synchronized file, e.g., the front camera view together with the side camera view

- `YYYY-MM-DD_tHHH_sTT-KKK_mocap.tsv`
  - Mocap data (i.e., coordinates of the optical markers) related to this segment
  - `YYYY-MM-DD_tHHH_sTT-KKK_audioII.avi`
- `YYYY-MM-DD_tHHH_sTT-KKK_videoJJ.avi`
- `YYYY-MM-DD_tHHH_sTT-KKK_video_lowresJJ.avi`
- `YYYY-MM-DD_tHHH_sTT-KKK_videoJJ_timings.txt`
  - Synchronization info related to the video track, used by EyesWeb to guarantee synchronization during playback
- `YYYY-MM-DD_tHHH_sTT-KKK_imuNN_acceleration.txt`
- `YYYY-MM-DD_tHHH_sTT-KKK_imuNN_gyro.txt`
- `YYYY-MM-DD_tHHH_sTT-KKK_imuNN_magnetic.txt`
  - The data from the Inertial Measurement Unit (IMU) is splitted into these 3 files. Each file contains 3 columns related to the x, y, z components
- `YYYY-MM-DD_tHHH_UU_segments.txt`
- `YYYY-MM-DD_UU_segments.txt`
  - The segmentation data. The first version of the segmentation file refers to a specific trial number and contains, in consecutive rows, the starting and ending time and/or frame number of each segment. The second version of the segmentation file contains, in consecutive rows, the trial number followed by the starting and ending time and/or frame of each segment

A.2 OSC messages

The DANCE platform modules communicate by exchanging data through network using OSC messages.

A.2.1 Sensors messages

The following is the format of the OSC addresses used to communicate data extracted by sensors (IMUs) embedded in wearable devices (x-OSC) or smartphones (Android devices):

- x-OSC sensors messages:
  - `/imu`: 10 floating-point values corresponding to the data captured by the sensors integrated on the IMU (`gyro-x,gyro-y,gyroz,acc-x,acc-y,acc-z,mag-x,mag-y,mag-z,gyro-temp`)

- Android apps

  In the following OSC tags, X represents the device ID. If more than one device is used, the user can change the mobile ID in the settings of the Android application. The default ID is 1.

  - EyesWebForAndroid app messages:
    - `/mobileX/bpm`: scalar floating-point number corresponding to the Beats Per Minute. The useful range is from 40.0 to about 250.0
    - `/mobileX/smoothness`: scalar floating-point number representing the smoothness of the movement. The range is from 0.0 to 1.0
    - `/mobileX/energy`: scalar floating-point number representing the energy of the movement. The range is from 0.0 to 1.0
  - EywSensorToOSC app messages:
    - `/mobileX/linear`: 3 floating-point values representing the acceleration, without the contribution of the gravity, along the device X, Y, and Z axis
    - `/mobileX/gyro`: 3 floating-point values representing the rate, in radians per second, of rotation around the device X, Y, and Z axis. Rotation is positive in the counter-clockwise direction.
    - `/mobileX/acceleration`: 3 floating-point values representing the acceleration along the device X, Y, and Z axis. Units are m/s^2.
    - `/mobileX/gravity`: 3 floating-point values indicating of the magnitude of the gravityalong the device X, Y, and Z axis. Units are m/s^2.

A.2.2 Platform messages

The following is the format of the OSC addresses used to communicate data read from recorded files or computed by the DANCE platform:

- `/imu/imuNN/SENSOR_NAME`: it corresponds to 3 floating point values representing the 3D value of the data captured by the `SENSOR_NAME` sensor endowed in the `imuNN` device (default port is 6666)
- `/imu/imuNN/features/FEATURE_NAME`: it corresponds to 1 floating point value representing the feature `FEATURE_NAME` extracted on the data captured by the `imuNN` device (default port is 6666)

Appendix B: EyesWeb blocks

Many EyesWeb modules have been enriched with new functionalities, and other modules have been created from scratch, in order to satisfy the Dance requirements:

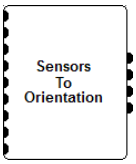
TSV reader

The module reads a TSV-fromatted file (Tab Separated Values) as exported from the Qualisys QTM software. The module has been extended in order to support 6-dof data (i.e., rigid bodies)

CSV reader

The module reads a CSV-formatted file (Comma Separated Values) as exported from the OptiTrack Motion capture software. It provides the same functionalities as the TSV reader but deals with a different file format.

SensorsToOrientation



Converts gyro, accelerometer and magnetometer data to 3D orientation respect to the Earth's magnetic pole. The block is derived from Madgwick IMU/AHRS algorithm: http://www.x-io.co.uk/open-source-imu-and-ahrs-algorithms/

MediaFileWriter

This block writes to file the input audio/video streams

The block has been extended by adding the support of many ready-to-use formats, e.g., mp4, avi, mov, mpeg, vob, etc.