|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding by Artificial Intelligencewww.mpai.community |

|  |  |
| --- | --- |
|  | **Public document** |
| **N944** | 2022/11/23 |
| **Source** | Requirements (ARA) |
| **Title** | MPAI-ARA Progress report and plans |
| **Target** | MPAI-26 |

Requirements (ARA) have come to the following conclusions.

The current scope of MPAI-ARA

Avatar Representation and Animation (MPAI-ARA) specifies the technologies enabling the implementation of the Avatar-Based Videoconference Use Case specified in Chapter 5 - Avatar-Based Videoconference, specifically:

1. The Representation of a Digital Environment populated with objects.
2. The Representation of a Digital Human.
3. The Representation of the features of a human.
4. The Representation of a Digital Human Model of the human.
5. The Representation of the motion of a human.
6. The Animation of a Digital Human

The current state of the art is:

1. Environment Model
	1. glTF: Almost ready-to-use technology. Some plugins for importing into Unity are not currently available but will be available soon. There is the OSS Cesium plugin to import scanned data to the Unreal Engine. Ditto for, e.g., an avatar model.
2. Avatar Model
	1. ISO/IEC 19774 (H-Anim)
	2. There are other skeletons.
	3. We select Kinect and Unity which appear not to be so different from H-Anim.
		1. A Kinect skeleton can be mapped H-Anim.
		2. H-Anim can be mapped to Unity.
	4. H-Anim is slightly different from H-Anim in Unity and Azure Kinect but they overlap.
	5. Our goal is to
		1. H-Anim should be the common intermediate format
		2. Use Unity where the animation takes place
		3. Capture body descriptors using Kinect
		4. Map the Kinect body model to H-Anim and map H-Anim to the avatar system
		5. Use glTF to represent the body model in the Environment Model.
		6. We do not currently address the issue of creating a body replicating a specific human body. SMPL <https://smpl.is.tue.mpg.de/> is a potential solution but is it compatible with H-Anim?
		7. The issue of clothing is left for future investigations.
3. Common Coordinate System
	1. The coordinate system is located at a specified point of the sensor (e.g., Kinect). The human body is placed at xb,yb,zb. The origin (x=0, y=0, z=0) is located at the center of the IR sensor on Kinect. The coordinate system is defined as follows:
		1. X grows to the sensor’s left [from the sensor’s POV] *horizontal*
		2. Y grows up (note that this direction is based on the sensor’s tilt) *vertical*
		3. Z grows out toward the direction the sensor is facing (the human bnody) *depth*
		4. 1 unit = 1 meter
		5. https://miro.medium.com/max/640/1\*85uPFWLrdVejJkWeie7cGw.png
	2. A new coordinate system must be defined in the virtual world which is “independent” of the coordinate system of the real world in order to reflect the human motion in the avatar motion in the virtual world. However, a mapping between the two may have to be provided.
4. Body Motion Descriptors
	1. Body motion. H-Anim defines the changes in the angles between the joints: the joint's coordinates and the joint's angle, subject to externally imposed constraints.
	2. We should find how Kinect represents the motion of the joints. Kinect also interprets the basic motion (motion of joints and angles) to derive hand gesture recognition. We do not need this.
5. Face motions.
	1. A head and face model.
		1. We can use the Unity avatar model, but we need a model that defines a set of FACS-compliant blend shapes. We need to find a 3D h&f model that allows an external application to modify both the facial actions (AU) and the h&f model modified to map a specific human face and head. Possible options:
			1. alexismorin/OpenFace-FACS-Unity-Facial-Animator
			2. Microsoft (should have the MIT licence)
			3. Flame model: has two separate parameter sets for controlling a specific modification of a h&f model
		2. We can use ARKit which defines FACS-compliant blend shapes.
		3. Does the Unity head and face model include FACS (i.e., ARKit)-compliant blend shapes.
	2. We need to be able to morph the model to the actual face features
6. Emotion-extraction from FACS AUs.
	1. Extract AUs using OSS e.g., OpefFace, Py-feat
	2. Use Eckman’s mappings from AUs to prototypical facial expressions or
	3. Use AI to predict the facial expression from AU.
7. Descriptors to enable Face identification.
	1. The problem is to have a face and head model that is parametric to identity and facial.