Thesis Report 11 : 11 May - 18 May

Goals

- Start implementing Motion Synthesis ✅
- Implement coefficient computation ✅
- Implement first draft of joint alteration rules ✅

Last Week Leftovers:

None

Done

- Analyzed the manner in which the paper Emotional Control of Unstructured Dance Movements synthesized the changes to motion
    - They designed **4 heuristic rules**
    - Each **heuristic rule** is responsible for changing one type of **core joint** and the **framerate** - Head, Pelvis, L/R Hand and Framerate
    - Each heuristic rule computes new positions for their pertaining joint (or framerate) using a **set of coefficients**
    - Each **coefficient** is associated with a **set of LMA features**
    - Each coefficient is the value that minimizes the Sum of Squared features between the original LMA features of the current motion, and the current LMA features generated to have the target emotion, on each keyframe (meaning we have a different coefficient for each keyframe (**I think, I could not confirm this, but from extensively reading the paper I believe this to be true**)
    - $$\min_{c_1} \sum_{t} ||\hat{f}_t^* - \hat{f}_t(c_{1_t})||^2$$
- **Implemented the Coefficient Computation**
    - First we get our keyframes, which we considered to be each frame at the 0.5s interval (which for our framerate corresponds to each 15th frame)
    - We then get the LMA features at this keyframe and store them
    - Then we compute each of our 4 coefficient sets using the Scipy.optimize.minimize

function using the **Powell Algorithm**

```
def compute_coefficient(self, coefficient_number):
    feature_index = []
    if(coefficient_number == 1):
        print("== COMPUTING COEFFICIENT C1 - PELVIS ==")
        feature_index = C1_INDICES
    elif(coefficient_number == 2):
        print("== COMPUTING COEFFICIENT C2 - HEAD ==")
        feature_index = C2_INDICES
    elif(coefficient_number == 3):
        print("== COMPUTING COEFFICIENT C3 - HANDS ==")
        feature_index = C3_INDICES
    elif(coefficient_number == 4):
        print("== COMPUTING COEFFICIENT C4 - FRAMERATE ==")
        feature_index = C4_INDICES
    else:
        print("UNKNOWN COEFFICIENT!")

    self.reference_features = []
    for feature in feature_index:
        self.reference_features.append(self._reference[feature])

    self.reference_features = np.asarray(self.reference_features)

    self.current_features = []
    for features in [frame["lma_features"] for frame in self._lma]:
        temp = []
        for feature in feature_index:
            temp.append(features[feature])
        self.current_features.append(temp)

    self.current_features = np.asarray(self.current_features)

    coefficient = [1.0] * len(self.current_features)

    res = minimize(self.compute_sse,
                   coefficient,
                   method='Powell',
                   #callback = self.minimize_callback,
                   options={'maxiter': 50000, 'disp': False})

    coefficient = res.x
    print(str(coefficient) + "\n")
```

```python
def compute_norms(self, reference, current_features, coefficient):
    norms = []

    for i in range(len(current_features)):
        norms.append(np.linalg.norm(
            reference - (current_features[i] * coefficient[i])) ** 2)

    return norms

def compute_sse(self, coefficient):
    norms = self.compute_norms(
        self.reference_features, self.current_features, coefficient)
    return np.sum(norms)
```

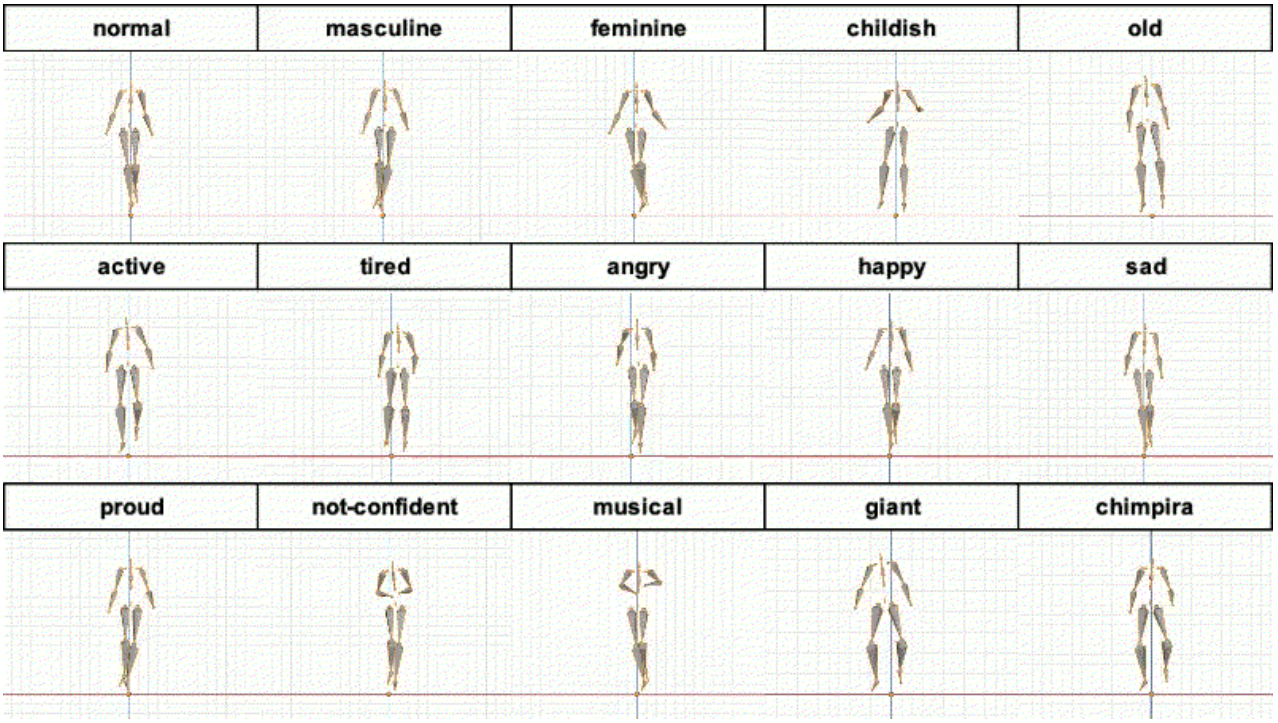
```
== COMPUTING COEFFICIENT C1 - PELVIS ==
[0.41153899 0.47771997 0.7158245  0.8443865  0.56208161 0.37859836
 0.3798366  0.43822972 0.61806454 0.86001617 0.67116499 0.40880306]

== COMPUTING COEFFICIENT C2 - HEAD ==
[0.8678899  0.89507473 0.9505033  0.96159584 0.91569305 0.85101851
 0.85970701 0.88873381 0.94118669 0.96967911 0.94075568 0.86736153]

== COMPUTING COEFFICIENT C3 - HANDS ==
[0.81507873 0.86266047 0.92068662 0.89489541 0.83109952 0.77877141
 0.79004283 0.84228133 0.90972417 0.91063661 0.85358101 0.78805848]

== COMPUTING COEFFICIENT C4 - FRAMERATE ==
[0.80399028 0.67854116 3.34252604 3.39291157 3.02242954 3.08527886
 3.23193815 2.99813069 2.5829971  2.37595156 2.4256148  2.32413738]
```

- **Implemented a first draft of the Heuristic Rules**
  - **Using our coefficients we modify each of the keyframes** following a rudimentary implementation of the heuristic rules from the Emotion Control of Unstructured Dance Movements

```
== RULE 1 - PELVIS ==
Original:(0.11696700396015841, 0.920651488299532, 0.013586263410536422)
Synthesized:(0.11696700396015841, 1.46241899560726, 0.013586263410536422)

Original:(0.2152666315852633, 0.9351480142805713, 0.018041975039001557)
Synthesized:(0.2152666315852633, 1.4235571486380587, 0.018041975039001557)

Original:(0.30624606996279824, 0.9438900914436577, 0.01717372998919957)
Synthesized:(0.30624606996279824, 1.212120532035703, 0.01717372998919957)

Original:(0.3886702245289807, 0.9421259271570863, 0.013845506060242428)
Synthesized:(0.3886702245289807, 1.0887334375624054, 0.013845506060242428)

Original:(0.47689506420256744, 0.9340780751230049, 0.009330661106444328)
Synthesized:(0.47689506420256744, 1.3431280392786822, 0.009330661106444328)

Original:(0.5809476311052442, 0.9249006905076202, -0.004321885875434952)
Synthesized:(0.5809476311052442, 1.4996354962255112, -0.004321885875434952)

Original:(0.7010496011040447, 0.9182550144005761, -0.010503591503660167)
Synthesized:(0.7010496011040447, 1.4877231630534422, -0.010503591503660167)

Original:(0.8023909074763002, 0.9300688798751953, -0.01734966530661228)
Synthesized:(0.8023909074763002, 1.4525539382055843, -0.01734966530661228)

Original:(0.8963065977439114, 0.94229572374895, -0.0172888757950318)
Synthesized:(0.8963065977439114, 1.3021918715566876, -0.0172888757950318)

Original:(0.9810183604944217, 0.9435902419296771, -0.015196377535101302)
Synthesized:(0.9810183604944217, 1.0756776172053777, -0.015196377535101302)

Original:(1.0646526595463832, 0.9370550675627023, -0.011911591623664806)
Synthesized:(1.0646526595463832, 1.2451915840511214, -0.011911591623664806)

Original:(1.1647571350054007, 0.9275971190447616, -0.0010143000120004003)
Synthesized:(1.1647571350054007, 1.4759896947257962, -0.0010143000120004003)


== RULE 2 - HEAD ==
(0.10979854254029503, 1.4856319589432885, 0.013286692228550522)
(0.11088420647318686, 1.5680166943384544, 0.013155660615636023)

(0.21205081486127567, 1.500180036403906, 0.01793266519763485)
(0.2124235397244652, 1.557363871205838, 0.01780945380332375)

(0.3066600615436356, 1.5089348765177801, 0.01718245987911147)
(0.3066366154319034, 1.5228710016441513, 0.017121579625030584)

(0.3888277606430525, 1.5071708960416281, 0.013847937263359094)
(0.38882002111501046, 1.5130017329791055, 0.013801497134348957)

(0.476151094592684, 1.4991223812305223, 0.009321289121436703)
(0.47621625338512796, 1.53672709860605, 0.009215317314467127)

(0.5818638849622343, 1.4899446380953807, -0.0043129629953347235)
(0.5816971367089994, 1.59045256717605, -0.004517663472929485)

(0.6978347117912368, 1.4832870344066076, -0.010363585595512903)
(0.698353423889768, 1.5761193770965312, -0.010575790999647141)

(0.7973791547470882, 1.4950823470662202, -0.01715514828964805)
(0.7980020689389764, 1.5604231076552688, -0.017324775890221255)
```

- **Took notice of a very recently released dataset of emotional Mocap data by Bandai Namco and tried incorporating it into our project** - https://github.com

/BandaiNamcoResearchInc/Bandai-Namco-Research-Motiondataset



  - Tried downloading and using this data, but unfortunately the BVH format used in this mocap is very much incompatible with DeepMimic's format. Even using our own BVHToDeepmimic conversion tool (which we used for the rest of the data in our project), there were too many artifacts that rendered this data unusable

- Gathered more data and added more LMA features to try to improve our LMA-PAD and PAD-LMA models. Currently training our models with this new set to see if there are any notable improvements

Left Undone

Problems

Notes

Thoughts

So I began this week exactly where I left the last one - Beginning the implementation of Motion Synthesis. Overall the process was relatively painless. I did have some struggles, for example at first I was trying to implement my own Minimization algorithm, akin to what the authors of the paper did, before realizing that I could just use the minimize method from the SciPy library.

There was also another thing to consider which was the fact that the LMA feature set we use is different from the one the authors of the paper used originally. We have more or less the same

amount of features, but include different ones, with ours having been a mix up of theirs, plus those of other papers (especially the one the coordinators sent me a while ago, and which will be included in the thesis references).

Also managed to implement, at least, a first draft of the Heuristic rules, so we're already able to synthesize new keyframes which is neat. The next step will be integrating the synthesis into PyBullet, so that we actually manifest these changes and force the joint position alterations. This is what I'll be focusing on during this following week. Moreover I also want to iterate on these rules, change them a bit, play with them, create new ones and so on.

On Friday I saw that Bandai Namco had recently released a very cool set of free animations for research mocap data. Besides the overall quality of the showcased animations, I really wanted to try to incorporate them into our project due to the fact that Bandai is a very well known powerhouse in the video game ecosystem, so being able to say I used one of their datasets in my own research may have been quite the boon when looking for a job later this year. Unfortunately, despite spending my entire weekend trying to incorporate their datasets, the BVH format that they used was pretty incompatible with both the format that Deepmimic expects, and with our own Bvh to Deepmimic conversion tool. So that was 48 hours that didn't amount to much. Nevertheless, it was worth the effort.

Now that I'm working on the more algorithmic part of the project, I also decided it was a good idea to keep passively experimenting with our ML models on the side. I added more mocap data from our kinematic dataset, changed our LMA feature set slightly (included more volume features), and left them reextracting. Over this week I also want to try and retrain our models using this new dataset to see if there's any notable changes.

Finally, when looking at the Dissertation page on Fenix, I noticed that they mention that the deadline for submission would be in October with the presentation dates still yet to be defined. Regardless, whether or not I'll be able to present/deliver earlier, my own imposed deadline for the completion of the project is mid of June (at most). I might end up only delivering in October nevertheless, however, so that I can write the necessary deliverable papers during July, August and September.

2º Semestre 2021/2022

Entrega da dissertação (inscritos na cadeira de dissertação no 2º semestre; provas no 1º semestre)

- **31 de Outubro de 2022**: Data limite para entrega da dissertação e resumo estendido
- **A DEFINIR**: Data limite para a realização das defesas
- **A DEFINIR**: Data limite para entrega da versão final da dissertação e resumo estendido: upload de PDFs no Fénix

I was also wondering whether you think there would be a feasible possibility of this work ending up being published, or if that's not an avenue I should pursue.

Work Hours

- Worked everyday from about 1pm to 6pm-9pm