1.00

CS180 Project 3 -- Ryan Nader

github: https://github.com/berkeleybear22ryan/CS180_Project3

website: https://berkeleybear22ryan.github.io/CS180_Project3/

youtube links ...

1by2 with music: https://youtu.be/7lLSLp_FY-E

2by3 no music: https://youtu.be/YMR4u9ATedI

single sample: https://youtube.com/shorts/_67T6Cf6VCE

ALSO: may need to convert file extension name in order to run i.e. .jpeg to .jpg (as I worked with .jpg) for instance due to my compression on photos.

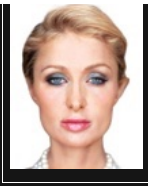
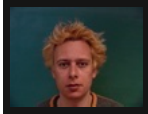
face morph, high resolution

Overview -- PLEASE READ IMPORTANT FOR CONTEXT -- done

For "my" photo in the first section, I will be using Paris Hilton's photo and will transitioning it to Tavi Gevinson's photo then continue from there.

For "my" photo in the second section, I will be using image 40 from the sample data.

I attached them both below.

Also note that all the images are high resolution but because of web limitations I had to compress a lot but even though the video's were greater then 4k you can at least see it in 4k in you go to youtube links.

This is the citation for the data that I used for Martin Schoeller.

WEBSITE: https://martinschoeller.com/

This is the citation for the data that I used for IMM Face.

Also for more info on the dataset please refer to the pdf in folder imm_face_db.

Citation

M. B. Stegmann, B. K. Ersboll, and R. Larsen. FAME - a flexible appearance modelling environment. *IEEE Transactions on Medical Imaging*, 22(10):1319-1331, 2003

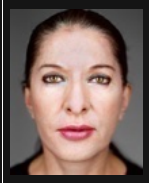
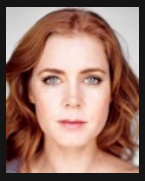
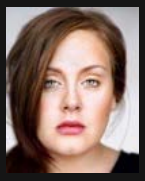
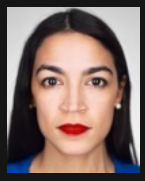
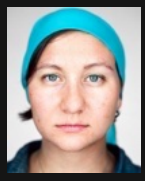
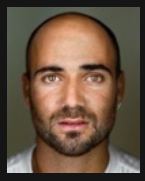
```
@ARTICLE{Stegmann2003tmi,
    author = "M. B. Stegmann and B. K. Ersboll and R. Larsen",
    title = "{FAME} -- A Flexible Appearance Modelling Environment",
    year = "2003",
    pages = "1319-1331",
    journal = "IEEE Trans. on Medical Imaging",
    volume = "22",
    number = "10",
    publisher = "IEEE"
}
```

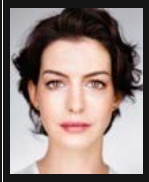
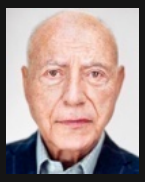
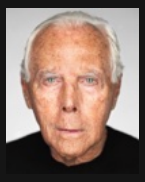
Part 1 -- done

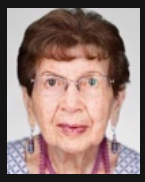
The code for this is in `a_start1.py` and `a_start2.py`.

Also these are my preprocessing steps that I used ...

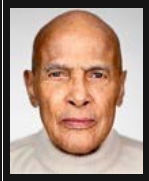
I first found the images as follows ... and then I cleaned them by removing the backround and aligning the faces to be close to the same size and area.


Marina Abramovic

Amy Adams

Adele

Alexandria Ocasio-Cortez

Alyssa Ratkewitch Haughwout

Andre Agassi


Anne Hathaway

Alan Arkin

Armani

Christian Bale

Batsheva Dagan

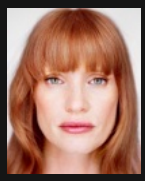
Bekwynhy Kayapo

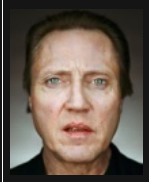

Harry Belafonte

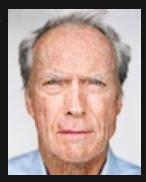
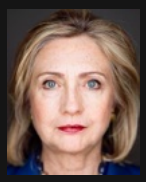
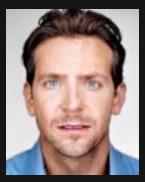
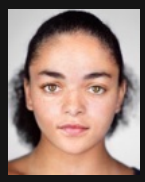
Bono

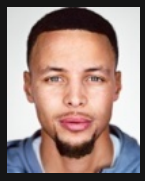
Kobe Bryant

Warren Buffet

David Chang

Jessica Chastain


Christopher Walken

George Clooney

Clint Eastwood

Hillary Clinton

Bradley Cooper

Imani Cornelius


Russell Crowe

Stephen Curry

Larry David

Robert De Niro

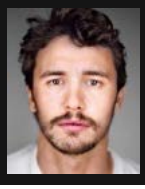
Gabby Douglas

Drake


Kirk Bloodsworth

Damon Thibodeaux

Juan Melendez

Roger Federer

James Franco

Bill Gates

In this I implemented both a way to manually select the points by hand in a_start1.py as well as automating the process.

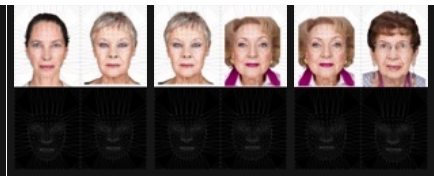
As the instructions mentioned that it would be better to have more points and I wanted to create a longer morph I decided to use the automated way as I got good results when testing.

Also another important note is that the automated way also enabled the point mapping to have a consistency to it that I think helped the results a lot.

I computed all the points and saved them in `./code/points` and there are 558 points per images and 94 total images.

After I used delaunay triangulation as recommended.

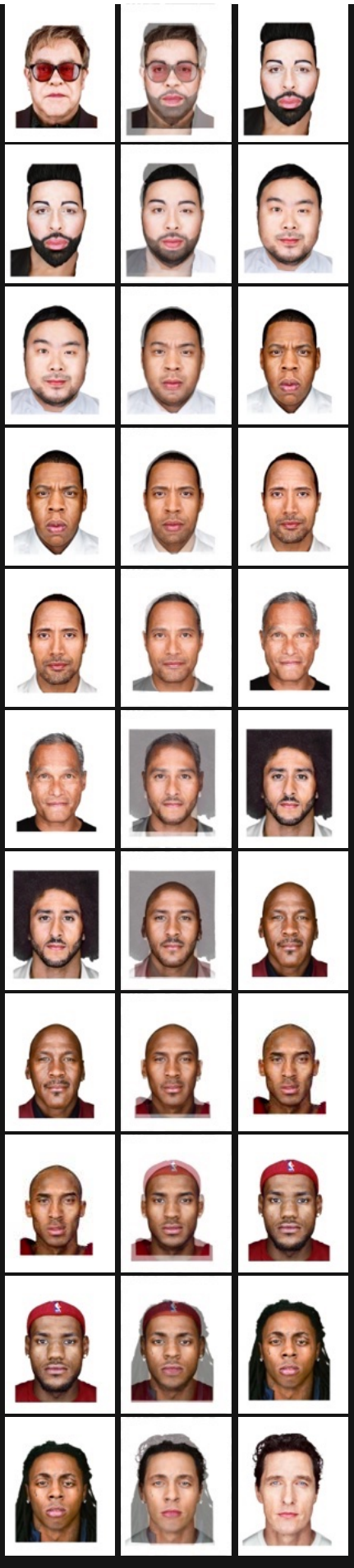
Here is an example of what it looked like:



Part 2 -- done

The code for this is in `a_start3.py`.

Here is an `A` and `B` image and a `midway faces` that I got ...



Part 3 -- done

The code for this is in `a_start3.py`.

The video results can be found here ...

youtube links ...

1by2 with music: https://youtu.be/7lLSLp_FY-E

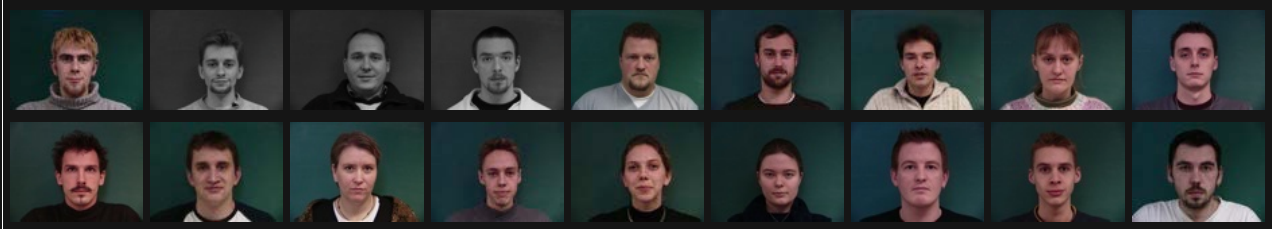
2by3 no music: https://youtu.be/YMR4u9ATedI

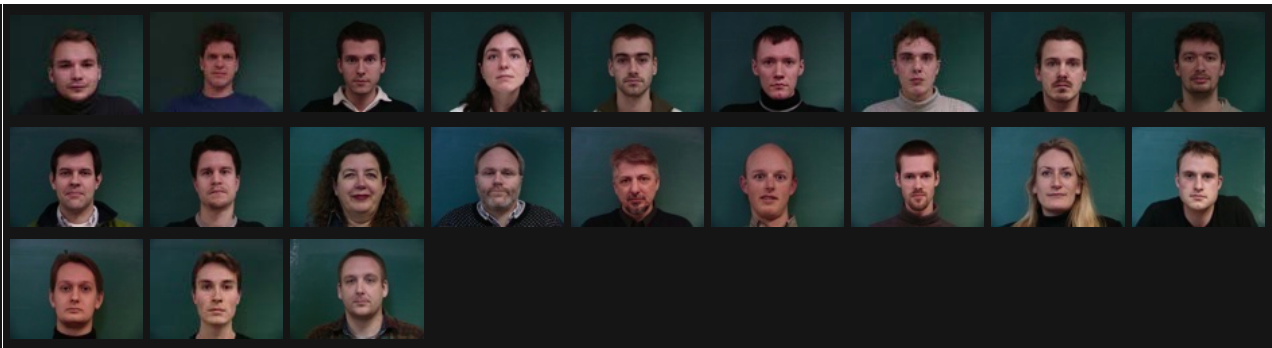
single sample: https://youtube.com/shorts/_67T6Cf6VCE

Part 4 -- done

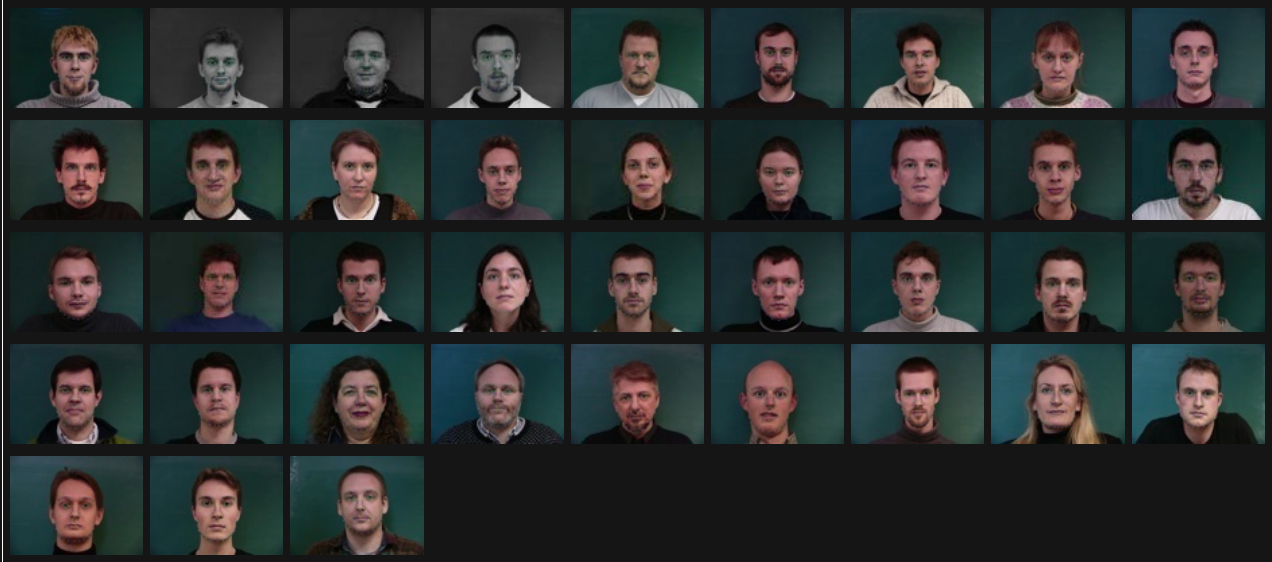
The code for this part is in `a_start4.py`.

for this section I picked the immface dataset with the following population ...

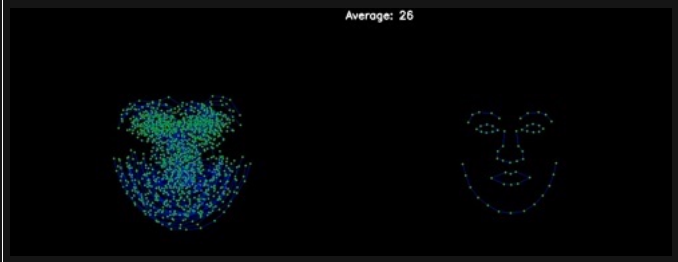




now from this we were given the face labels ...

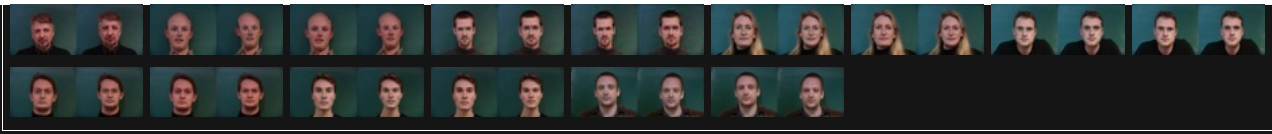


then from here we computed the average face shape of the whole population

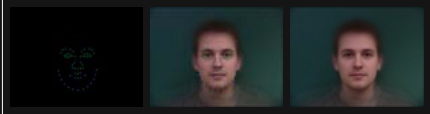


the final result is ...

now we then morphed each of the faces in the dataset into the average shape ...

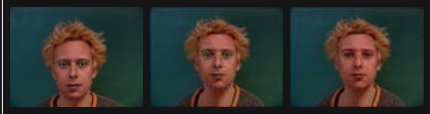


then we compute the average face of the population ... which is ...

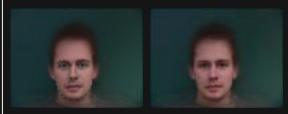


Show the mean image that you got (I described what image I used as me in the overview), as well as ...

(1) your face warped into the average geometry



(2) the average face warped into your geometry



Part 5 -- done

The code for this part is in `a_start5.py`.

Facial Feature Path Mapping and Example ASF File

Facial Feature Path Mapping

Based on the point paths and observations, we can infer the following mapping between the path numbers and facial features:

- path# = 0: Jawline (chin and face contour).
- path# = 1: Right eyebrow.
- path# = 2: Left eyebrow.
- path# = 3: Nose bridge and tip.
- path# = 4: Right eye.
- path# = 5: Mouth (lips).
- path# = 6: Left eye.

This mapping aligns with the 7 point paths mentioned in the description (3 closed paths and 4 open paths). The closed paths are likely the mouth and eyes, while the open paths are the eyebrows, nose, and jawline.

Now the theory for this part is: $S_{\text{caricature}} = S_{\text{mean}} + \alpha \cdot (S_{\text{muse}} - S_{\text{mean}})$

When $\alpha > 1$: Exaggerates your features (caricature).

When $0 < \alpha < 1$: Makes your features closer to the mean.

When $\alpha < 0$: Inverts your features relative to the mean.

When $\alpha = 0$, then $S_{\text{caricature}} = S_{\text{mean}} + 0 \cdot (S_{\text{user}} - S_{\text{mean}}) = S_{\text{mean}}$ i.e. it because the mean which we can see mathematically.

...

$\alpha < 0$: Inverts your features relative to the mean.

$\alpha = 0$: Your face is morphed to the mean face shape.

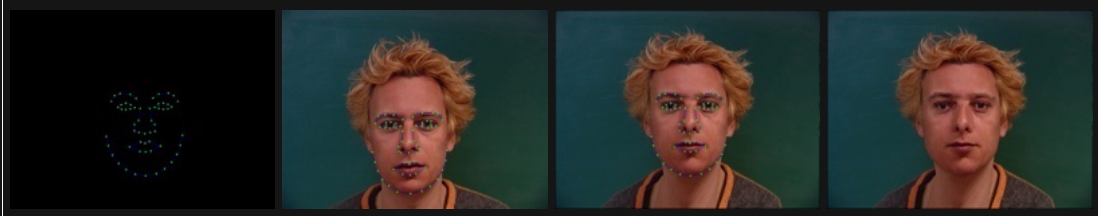
$0 < \alpha < 1$: Your face moves closer to the mean face shape, reducing your unique features.

$\alpha = 1$: The caricature shape is your original face shape ($S_{\text{caricature}} = S_{\text{user}}$).

$\alpha > 1$: Exaggerates the differences between your face and the mean, creating a caricature.

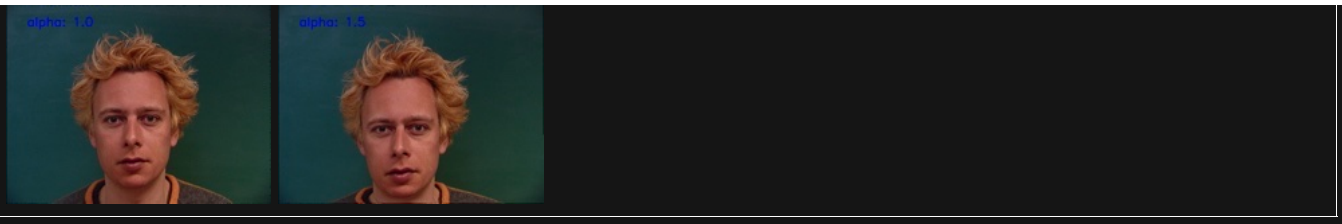
here are the results using this technique ...

so we know that the mean and user results that we got earlier were ...



now we just apply the formula to create the caricature and we get the following ...

from this you can see that $\alpha > 1$ creates ...



Mandatory BW + Extra -- done

I had to do one mandtory BW ... here are some things that I did.

Create a system that could layout points witha accuracy and consistency on faces

Did this process on very high resolution photos.

Created a theme video centered around Martin Schoeller.

Made a video with music and showed the triangulation movement in a visual.

Came up with the idea to use what we learned about last time with sharpening to help keep the eyes sharp during the morph, something that really is important in Martin Schoeller images.

Manually created the order of the photo to have a nice flow and style

Lined up all the photos close enough by hand and cropped one that had the entire body.

Remove the backround of all the photos as well using the masking technique.

Also a cool extension of this would be to find the average face and basically do part 2 with this as my tool can place points so would have cool result, but I ran out of time.

NOTE: links for video are at the top of the page and also below.

The video results can be found here ...

youtube links ...

1by2 with music: https://youtu.be/7lLSLp_FY-E

2by3 no music: https://youtu.be/YMR4u9ATedI

single sample: https://youtube.com/shorts/_67T6Cf6VCE