

# Comp Photography (SPRING 2016)

## Final Project

Jason Malia  
[jmalia3@gatech.edu](mailto:jmalia3@gatech.edu)

<https://cs6475.wordpress.com/2016-spring/>

# In a galaxy far, far away...MOSAICS!

In this project I used a library of scenes from Star Wars movies and explored four different methods for creating photographic mosaics computationally.

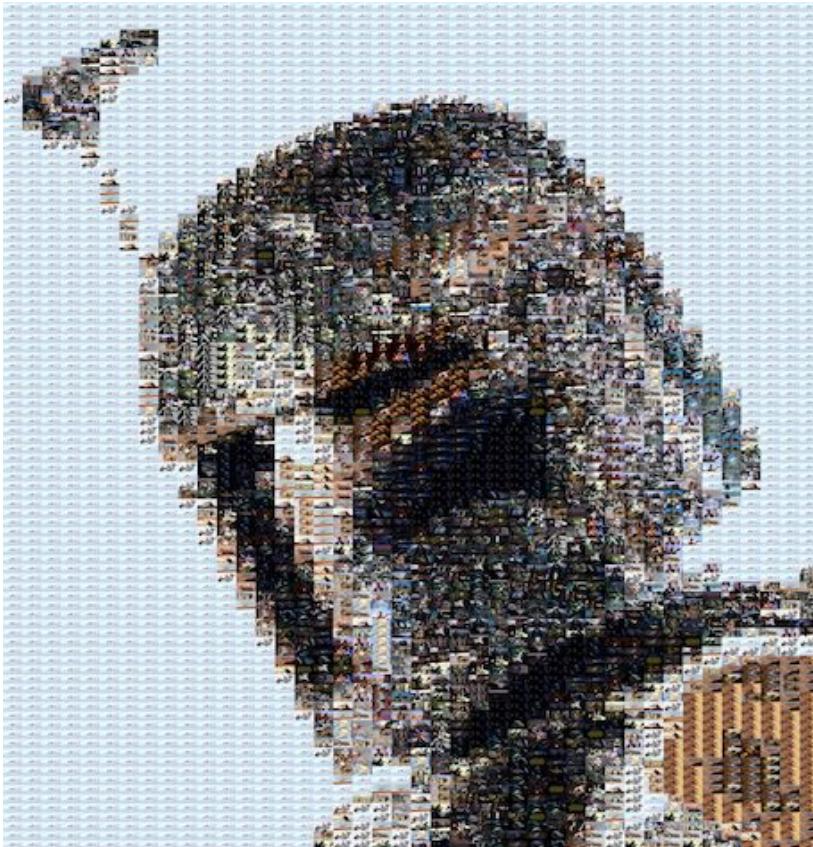
# The Goal:

I wanted to understand, implement, and evaluate different methods for creating photographic mosaics. After reviewing some of the different project ideas posted to Piazza and reading up on mosaics, I thought the approaches were interesting and directly used some of the ideas we learned over the course of the semester. And I've always been a big fan of Star Wars so thought it would be an especially fun subject matter to test out my implementation.

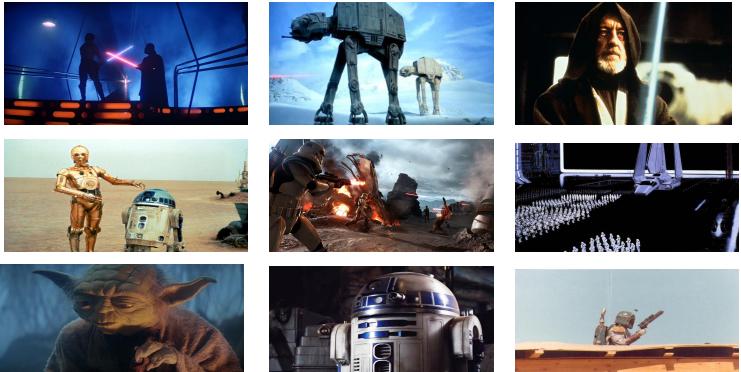
Input: Example Main Image



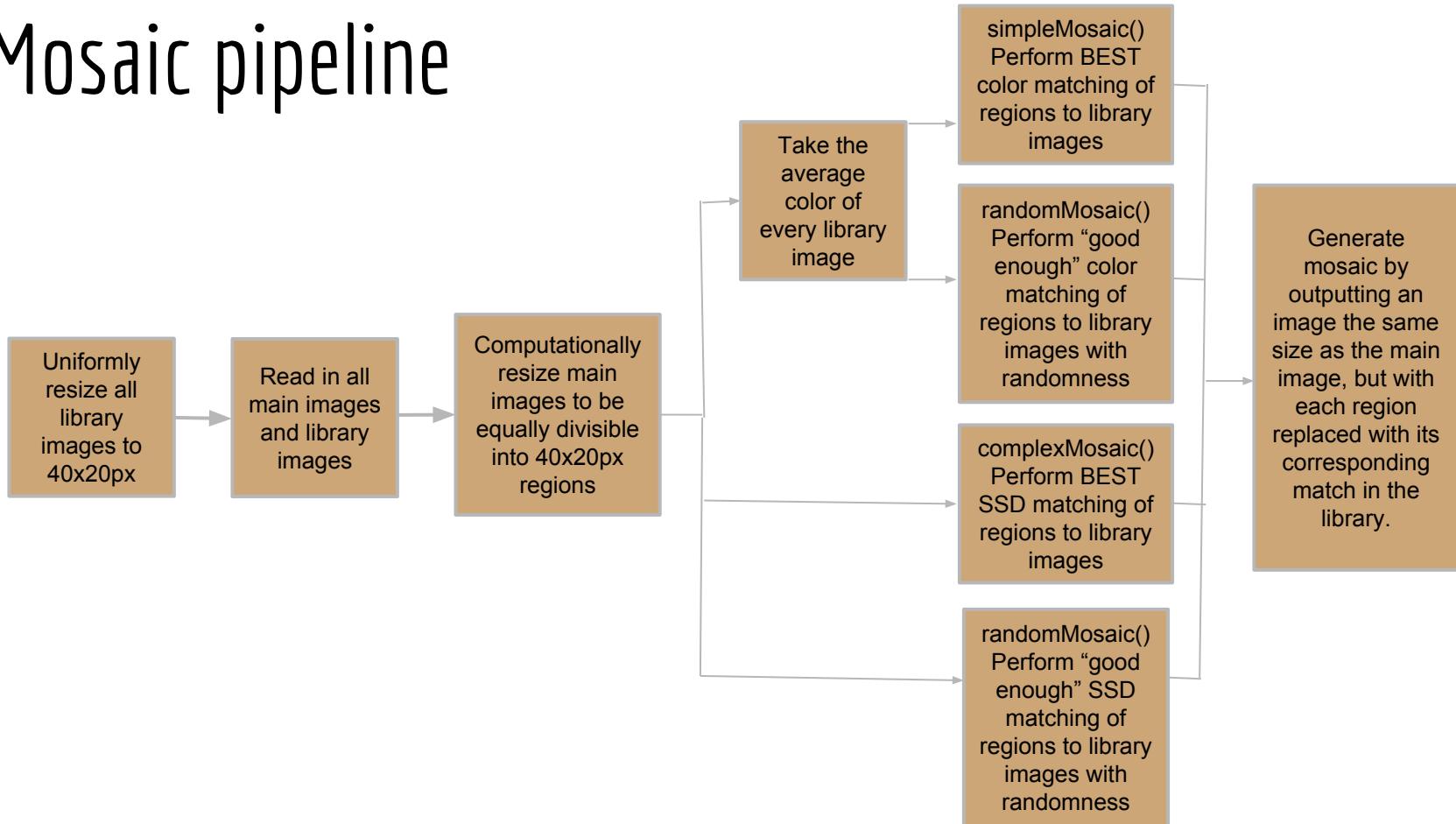
Example Output



Input: Example Library Images



# Mosaic pipeline



# What is the best way to see your project?

- Code:

<https://drive.google.com/open?id=0B8ybbCCMVKktVU1EbS1VM3NvSTg>

- Output images:

<https://drive.google.com/open?id=0B8ybbCCMVKktMC1EWUdpHM0YXc>

# 4 different methods for creating mosaics

- SINGLE COLOR METHODS - for both of these methods each of 200 library images was mapped to a single color, its average
  - **simpleMosaic()** - for this method I iterated through the 40x20 regions of the main image, computed the average color, then iterated over all the library colors, computed the best match (the smallest square difference in color channels), and added the corresponding library image at that region's location in the mosaic
  - **randomMosaic()** - performed similar to above, but started iterating over the library images at a random location (wrapping around if necessary) until I found a squared difference that was below a provided threshold (the greater the threshold, the less strict the match). If no color was below the threshold, the best match was used similar to simpleMosaic()

# 4 different methods for creating mosaics

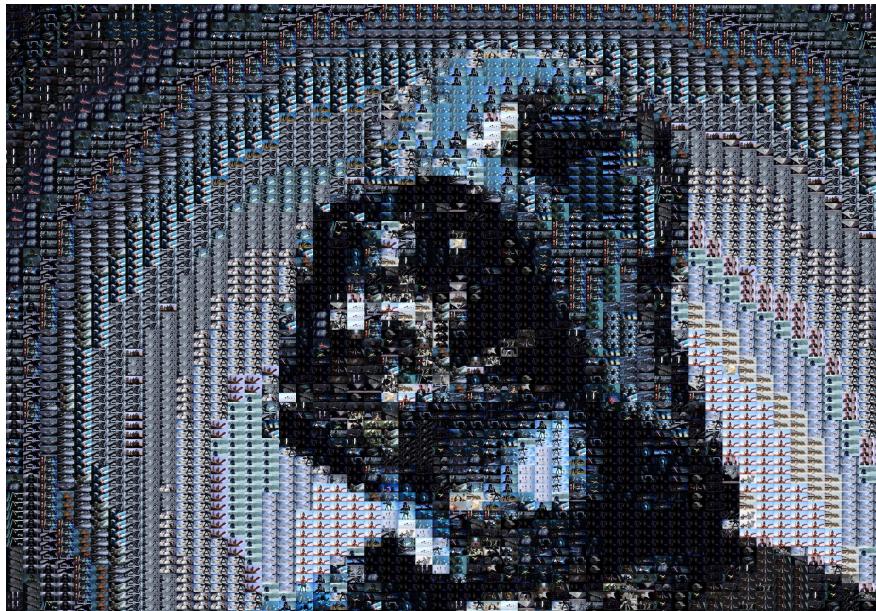
- PIXEL-BY-PIXEL SSD METHODS - for both of these methods the library images were used directly - a pixel-by-pixel sum of squared differences was computed to find the matches
  - **complexMosaic()** - for each region the library image that provided the lowest SSD was used
  - **rcMosaic()** - performed similar to randomMosaic() - a threshold was provided, and the first image after the random starting location that was lower than the threshold was used. If no such image existed, the best match was used.

# What worked?

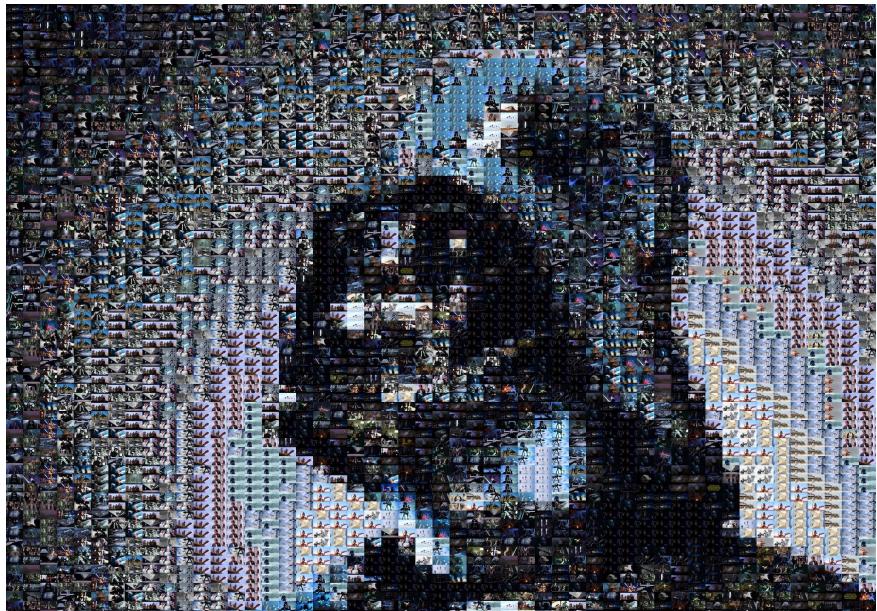
Surprisingly the single color methods worked the best. They understandably were much more efficient and so much faster, but they also created what I believe to be the desired effect of a mosaic. The regions were small enough that they were able to give the impression of the color of that region, while having enough variety to look like a commercial mosaic.

Of the two color methods, I like random method best. For the simple method, large similar regions would result in the same image being chosen repeatedly. By adding randomness and a threshold, I believe the variety achieved made for the best looking mosaics.

# What worked? - You be the judge...



Simple mosaic



Random mosaic

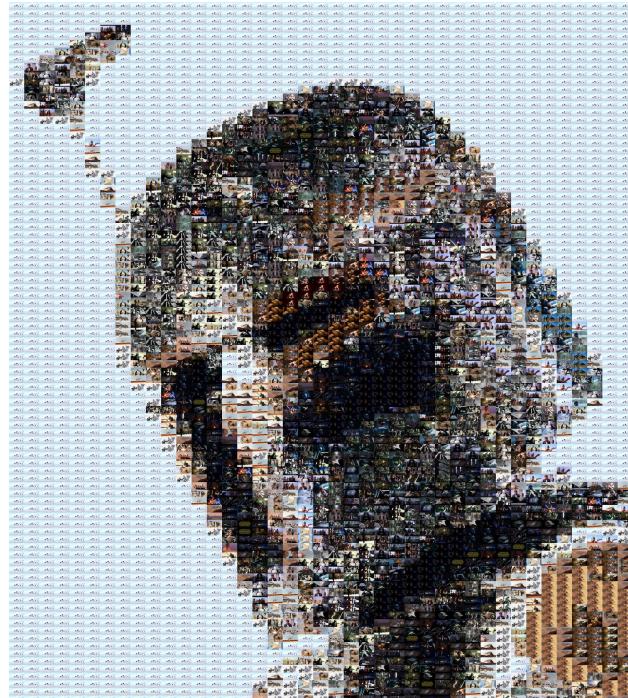
# What did not work? Why?

I do not believe the SSD methods worked well for creating the desired effect of a mosaic. For similar reasons to why I preferred the random mosaic to simple mosaic, I preferred the single color mosaics to SSD ones because of the additional variety seen.

# What did not work? - You be the judge...



SSD mosaic



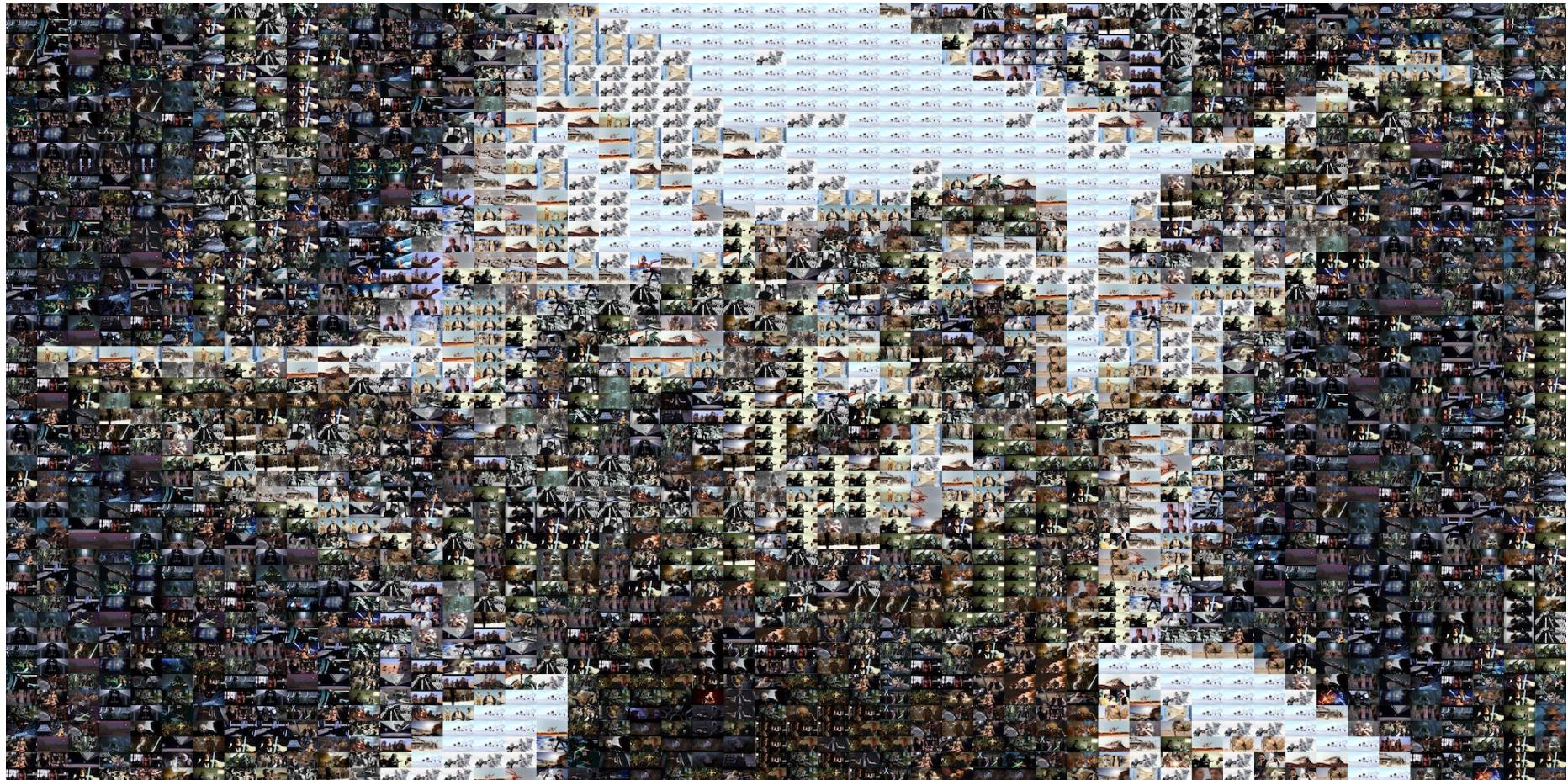
Single color mosaic

# Any additional details?

- I do believe I was limited by only using 200 library images. Having many more images would have allowed for better matching and more variety in the final mosaics.

A couple of my favorites...





# References / Pointers

- I based my approaches on this article from wikipedia...  
[http://en.wikipedia.org/wiki/Photographic\\_mosaic](http://en.wikipedia.org/wiki/Photographic_mosaic)
- I also used and based my I/O on that provided by the TAs in the test files for previous assignment.