AniME: Automatically Generating Anime Portraits

Michelle Nguyen, Jingyi Li
UC Berkeley EECS
aimichelle@berkeley.edu, noon@berkeley.edu

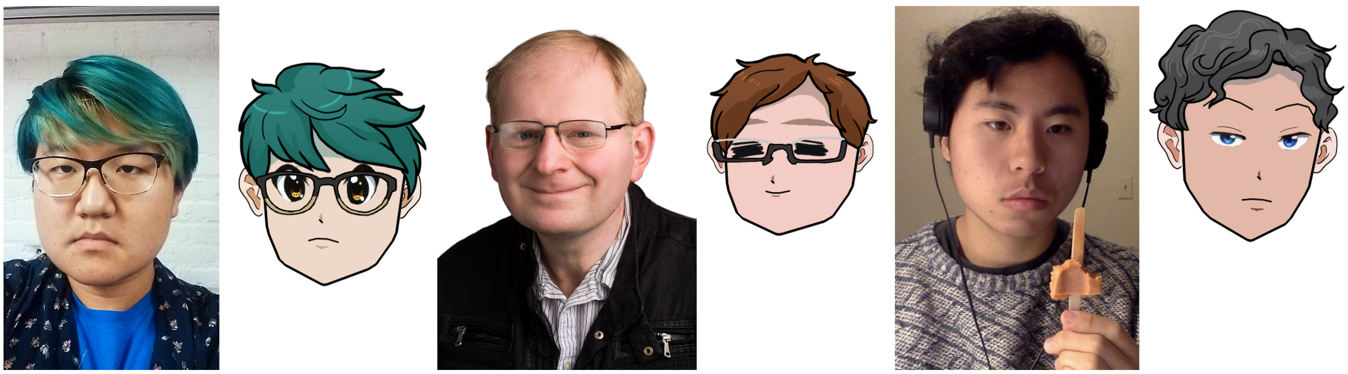

Figure 1. Sample automatically generated anime-style portraits from AniME.

ABSTRACT
The popularity of anime (Japanese animated productions) has transcended cultural boundaries, resulting in high volumes of fan works such as personalized avatars. However, drawing aesthetically pleasing anime is a craft that takes a long time to develop. AniME (pronounced ani-mee) is a pipeline which integrates computer vision, computational photography, and data-driven frameworks to automatically generate an anime-style portrait from a user given photo. Our approach renders the corresponding lineart using landmark facial annotations, while features such as eye shape and hairstyle are obtained via template matching to a hand-drawn library of anime eyes and hairstyles. Finally, colors of the anime portraiture are determined through simple averaging and clustering of selected image patches. We validate our tool using photos of people of varying genders, ethnicity, and hair styles.

Author Keywords
Face stylization; Computational photography

INTRODUCTION
Since the 1960s series *Astro Boy*, anime has captivated audiences across the globe. Hit series such as *Neon Genesis Evangelion* (an artifact of the popular "mecha" genre) have adapted cult followings, Studio Ghibli's *Spirited Away* took home an Academy Award, and to this day countless Internet forums and websites focus on the discussion of both anime itself and the subculture as a whole. Anime portraitures and avatars are common on fandom-based or art sharing social networking sites such as DeviantArt, Tumblr, Pixiv and Nicovideo. However, honing an aesthetically pleasing anime style of art requires practice, and can lead to suboptimal results for novice artists. Furthermore, even for advanced artists, maintaining the key facial features of the subject's face can be difficult.

Figure 2. Hand-drawn hairstyles in the AniMEhairstyle library.

We introduce AniME, a tool for automatically generating anime portraitures from input photographs. AniMEaims to generate anime portraits that appear natural and pleasing to the eye, while staying true to the user's facial features, for a wide variety of uses, including social media icons.

RELATED WORK
Other works aim to cartoonize the input face through differing methods. Some [10, 7] utilize a data-driven approach

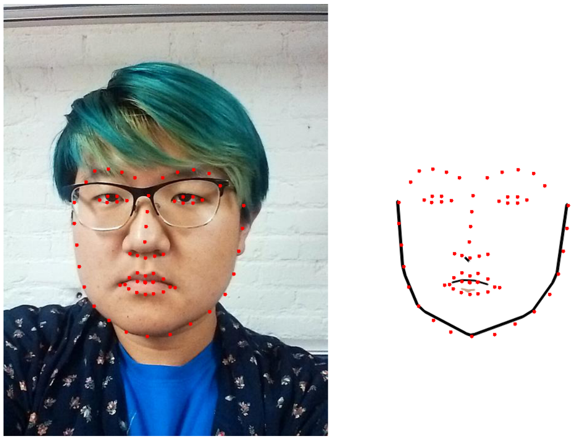

Figure 4. Left: User-uploaded portrait annotated with facial landmarks. Right: Generated anime lineart annotated with facial landmarks.

similar to that of AniME, matching features such as the facial structure, hair, and eyes to a database of cartoon counterparts. Others [1] manipulate the strokes of the image to create the outline characteristic of children's cartoons or storybooks. AniMEcombines these various methods, such as template matching to a database, in order to render anime faces: a style which has yet to been done and requires exaggeration of certain facial features in novel ways.

In order to extract the correct hairstyle from the input photo, we must perform hair segmentation. One method [7] involves performing graph cut on a preliminary hair region, predicted from prior distributions. Some [5] perform a frequency and color analysis to generate markers for a matting process. Because this method has been proven to be effective for detecting hair regions, AniMEutilizes a simplified version of the algorithm which skips the matting process.

OVERVIEW

Given a photo and optional gender, our system generates the structure of the corresponding anime portraiture by detecting key facial annotations and stylizing certain facial features. For features where the relationship to the anime-counterpart is not easily expressible via an equation, such as eye shape or hairstyle, we utilize data-driven methods and compare to a database of hand-drawn anime features. To color the portraiture, we determine the average color of the real face's skin and hair by extracting the corresponding image patch and simply averaging or clustering. Finally, we detect the presence of thicker eyebrows or glasses by performing a gradient or frequency analysis, respectively.

Face

To draw the outline of the anime portraiture's facial structure, the pipeline begins by resizing the image to a width of 700 pixels. 68 landmark facial annotations[4] are then extracted from the photo using a face detector trained on the iBUG 300-W face landmark dataset with Histogram of Oriented Gradients, or HOG, features [2] (see Figure 4).

On top of Python Image Library's draw function, AniMEimplements variable brush stroke widths and anti-aliasing via Xiaolin Wu's line algorithm [8]. The black lineart characteristic of anime is drawn from a subset of the detected landmarks such that the stylistic traits known to anime are exaggerated. Points around the user's chin are excluded from the subset to generate the sharp chin typical of anime characters. Mouths are rendered as either closed or open-grinned. To conform to the style, the mouth is resized to be no wider than the gap between the eyes. Finally, the nose is illustrated as a small line shifted downwards.

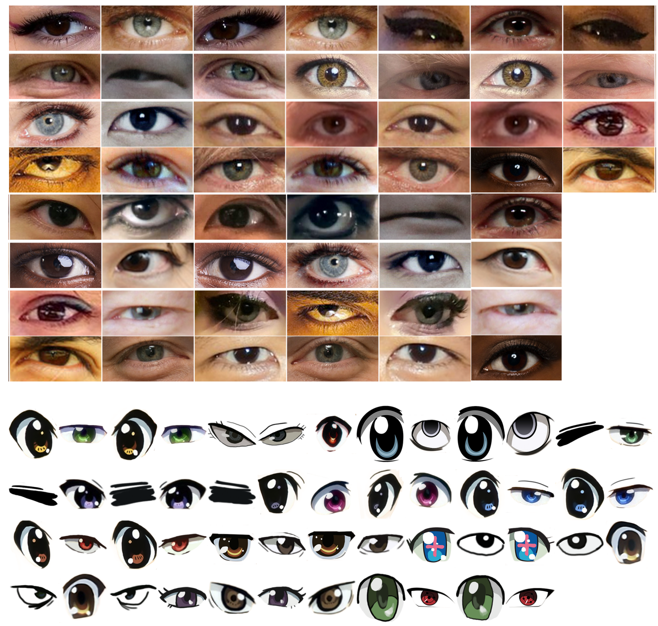

Figure 5. Top: The library of real eyes. Bottom: Corresponding hand selected/drawn anime eyes in the AniMElibrary.

To extract skin color, we select an image patch of the user's nose and cheek region, since it is usually clear of hair and glasses. We perform k-means clustering with k=2 on this image patch to select a light and dark shade [9], for filling the face shape and shading, respectively. Due to the typically pale skintones of anime characters, we further lighten the extracted colors by converting to HLS space, and adding 10% to the lightness value.

Eyes

The user's eyes are cropped from the photo using detected landmarks around the eye region. These image patches are rescaled to fit a 110x220 image patch, converted to grayscale, and normalized. Using normalized cross correlation [6], we perform template-matching against a library of 26 real eyes. Each of these eyes are mapped to a matching eye in a hand-picked library of anime eyes (see Figure 5). The corresponding anime eye of the best match is resized, then placed onto the anime portrait.

Eyebrows

Eyebrows follow a binary: once extracted, we calculate their vertical gradients to detect their bushiness quality. If the sum of the eyebrow energy is above a certain threshold,

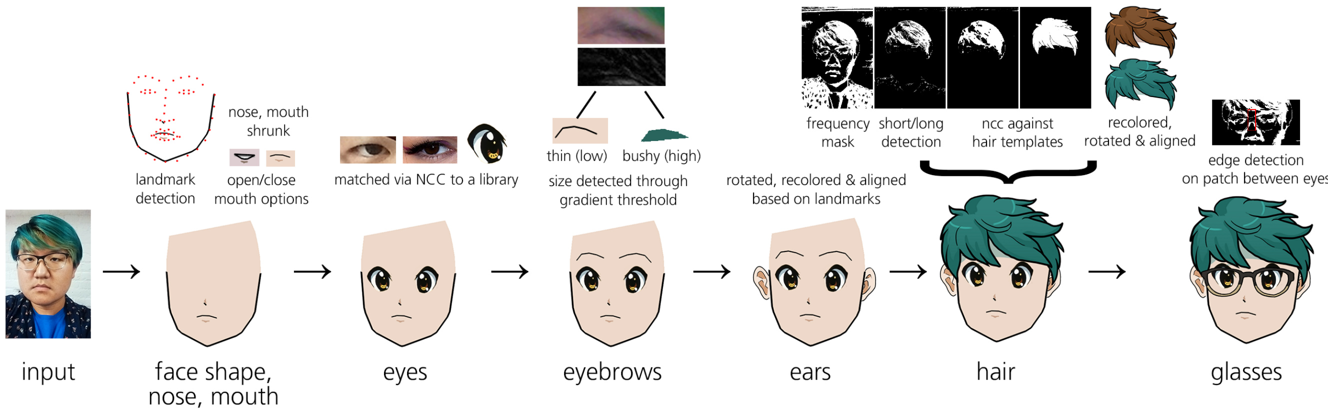

Figure 3. An overview of AniME's pipeline.

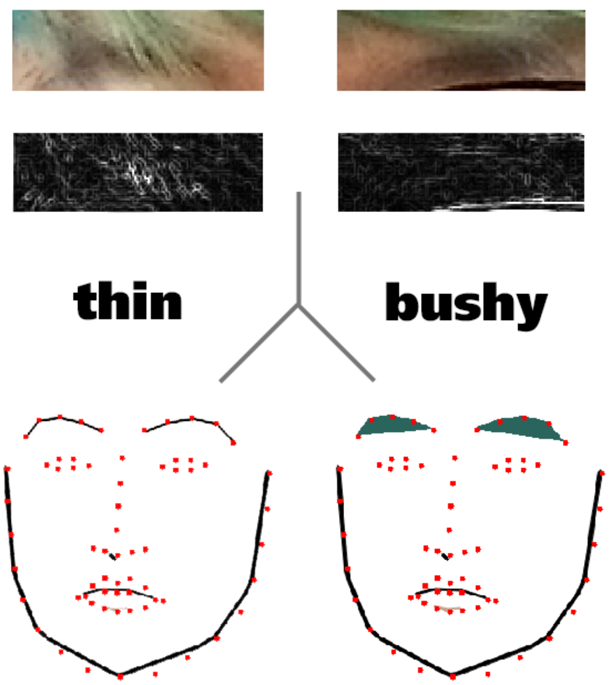

Figure 6. Eyebrow overview.

AniMErenders eyebrows as filled in polygons; otherwise, it draws them as lines (see Figure 6).

Hairstyle

AniMEperforms hair segmentation on the original photograph through a series of steps. Since hair is characterized by a high-frequency texture, we perform a frequency analysis on the image by filtering out the photo's low frequencies [5]. This frequency map is then thresholded to retain only the highest frequencies, creating a frequency mask.

Unfortunately, the frequency analysis also preserves other edges throughout the photograph, such as the user's facial

Figure 7. The hair segmentation used in hair matching is created via a combination of masks. Left: The frequency mask and color mask generated through thresholding. Right: The morphologically opened hair segmentation and binary mask of the best matching anime hair.

features or clothing. To remove these edges, we additionally perform a color analysis by determining the hair color through simple averaging of an image patch cropped above the user's forehead. We also derive the standard deviation of colors within this image patch. All pixels that are not within a standard deviation of the average color are excluded from the hair segmentation.

The hair segmentation is the intersection of the frequency mask and color mask. However, the resultant mask is peppered with holes, which AniMEfills by performing a morphological closing operation. This operation consists of a dilation followed by an erosion.

Next, AniMEdetects the presence of long hair by selecting a thin band around the lower chin and neck region of the resultant mask. The pixels are then summed to determine how much hair is present in the patch. If this sum is above a certain threshold, AniMEdecides the user in the photo has long hair. The mask is then resized and normalized to be aligned with the hand-drawn images of hair in our hand-drawn library. A normalized cross correlation is run across the mask and library images to determine the best anime counterpart to the user's real hair. If a gender was specified, the correspondingly labeled anime hairs are given a larger weight. If the hair is long, the cross correlation is performed with a library of longer hair styles.

The library hairstyle with the largest NCC score is appropriately selected, scaled, translated, and rotated to fit the user's

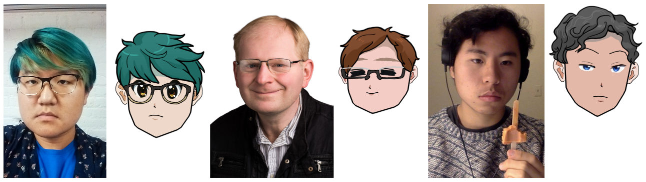

Figure 8. Before and after of anime portraitures, labeled (a), (b), and (c) from left to right

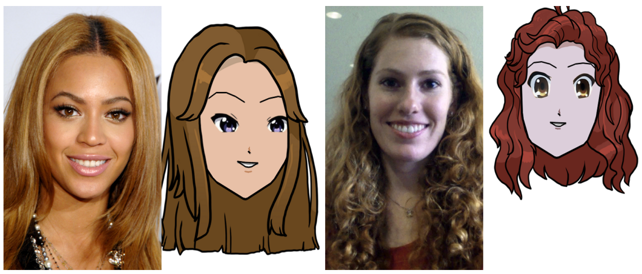

Figure 9. Before and after of anime portraitures, labeled (d), and (e), from left to right

face. If the user has long hair, the hairstyle is then placed in two passes: the first being the hair behind the head, and the second being the fringe. Otherwise, with short hair, only the fringe is necessary.

Finally, the original hair color is extracted by a simple averaging of the colors given in a patch above the hairline, with the mask applied. AniMErecolors the library hairstyle with this color.

Glasses

Glasses are detected on the user's face by extracting the high frequencies in the image patch between the user's eyes. Since shadows or hair may potentially generate high frequencies within this image patch, a threshold is applied such that glasses are only detected when enough high frequencies are present.

If glasses are detected, a pair of pre-drawn glasses corresponding to the user's gender are resized according to the face's width and overlaid on top of the portraiture's eyes.

RESULTS

To validate our tool, we generated anime portraitures using photos with people of varying genders, ethnicity, and hair styles.

In (a), we selected a photo of a user with a unique hair color to exhibit AniME's hair color detection capabilities. Additionally, this example demonstrates the tool's glasses detection and placement algorithm.

As anime characters are typically of Asian descent, (d) and (e) were selected due to the differing ethnicities of the subjects. From these results, we see that AniME's methods of lightening skin tone to match the pale skin tones of anime characters may not generalize well to all ethnicities. (d) and (e) also showcase AniME's ability to detect and render longer hair styles, which consists of a multi-layer placement scheme. These two examples also exhibit the tool's methods of resizing and simplifying open-mouthed smiles.

As previous examples were primarily female, we select a male subject to validate AniME. (b) was also selected due to the age of its subject, showing that the system can generate accurate results regardless of age. Furthermore, (b) is another demonstration of the system's ability to detect and render glasses.

Finally, we selected (c) from a friend's Facebook profile picture, proving that AniMEcan be used on any arbitrary photo and still create a promising anime portraiture.

LIMITATIONS AND FUTURE WORK

AniMEhas several important limitations. Although angular faces are often characteristic of the anime style, the lineart generated by AniMErequires further smoothing. Many lines are rendered by simply connecting the detected landmark annotations. As such, some lines have excessively sharp corners. Generating intermediate points between these landmark demarkations or utilizing anti-aliasing with Python Image Library's arc function may be sufficient.

The current methods of normalized cross correlation to match the user's eyes and hairstyle are highly inefficient. Instead, we can explore the derivation and comparison of feature descriptors of the eyes and hairstyles, utilizing possible comparison methods such as SIFT[11] or RANSAC[3].

AniMEperforms no customization of the selected anime eye counterpart. Using the landmark annotations around the user's eye and the matched eye, we can generate two vectors that span the eye subspace. This will allow us to determine the features characteristic to the user's eye relative to that of the matched eye, in which we can use to warp the anime eye accordingly. Given high resolution photos, eye color may also be determined.

Typically, the colors of anime hair are highly saturated. AniMEcurrently colors the portraiture's hair using the average color extracted from an image patch of the user's real hair. Due to the presence of lighting and shadows, this average color may even be darker than that of the user's real hair color. To fit with the anime style of brightly colored hair, we can perform further saturation of this extracted color.

Finally, our database of eye shapes and hairstyles can be extended. Since there are a wide variety of eye shapes and hair styles in the wild, more samples can be drawn for our library to accommodate for a larger set of possibilities.

CONCLUSION

The integration of computer vision, computational photography, and data-driven methods of hand-drawn libraries makes AniMEan attractive tool to automatically generate pleasing anime portraits, helping users become popular on social media sites and making anime one step closer to becoming a reality.

ACKNOWLEDGMENTS

The authors would like to thank Professor Alyosha Efros and TA's Rachel Albert and Weilun Sun for a great semester! The projects in computational photography were interesting, and we achieved Efros's goal of having fun, as presented in the first day of lecture.

REFERENCES

1. Chen, H., Zheng, N.-N., Liang, L., Li, Y., Xu, Y.-Q., and Shum, H.-Y. Pictoon: A personalized image-based cartoon system. In *Proceedings of the Tenth ACM International Conference on Multimedia*, MULTIMEDIA '02, ACM (New York, NY, USA, 2002), 171–178.
2. Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, IEEE Computer Society (Washington, DC, USA, 2005), 886–893.
3. Fischler, M. A., and Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (June 1981), 381–395.
4. Kazemi, V., and Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, IEEE Computer Society (Washington, DC, USA, 2014), 1867–1874.
5. Rousset, C., and Coulon, Y.-P. Frequential and color analysis for hair mask segmentation. International Conference on Image Processing (ICIP) (2008), 2276.
6. Sarvaiya, J. N., Patnaik, S., and Bombaywala, S. Image registration by template matching using normalized cross-correlation. In *Proceedings of the 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, ACT '09, IEEE Computer Society (Washington, DC, USA, 2009), 819–822.
7. Shen, Y., Peng, Z., and Zhang, Y. Image based hair segmentation algorithm for the application of automatic facial caricature synthesis. *The Scientific World Journal 2014* (Jan. 2014).
8. Wu, X. An efficient antialiasing technique. *SIGGRAPH Comput. Graph. 25*, 4 (July 1991), 143–152.
9. Zhang, Q., Chi, Y., and He, N. Color image segmentation based on a modified k-means algorithm. In *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, ICIMCS '15, ACM (New York, NY, USA, 2015), 46:1–46:4.
10. Zhang, Y., Weiming, D., Deussen, O., Huang, F., Li, K., and Hu, B.-G. Data-driven face cartoon stylization. *SIGGRAPH Asia '14*.
11. Zhong, S., Wang, J., Yan, L., Kang, L., and Cao, Z. A real-time embedded architecture for sift. *J. Syst. Archit. 59*, 1 (Jan. 2013), 16–29.