Baby Face Generator

Sarah Divel
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
Email: sdivel@stanford.edu

Picha Shunhavanich
Department of Bioengineering
Stanford University
Stanford, CA 94305
Email: picha@stanford.edu

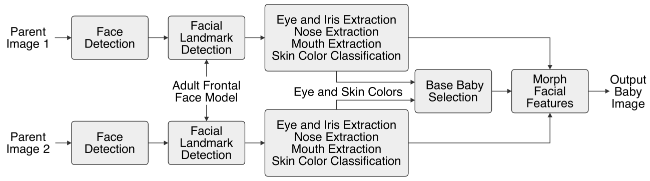


Fig. 1. Baby Face Generator algorithm

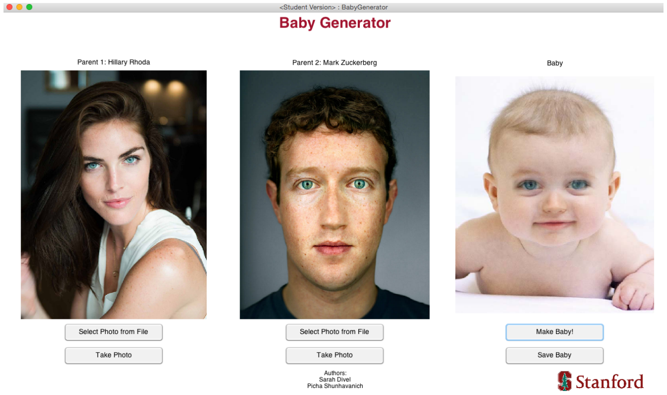


Fig. 2. Baby Face Generator GUI

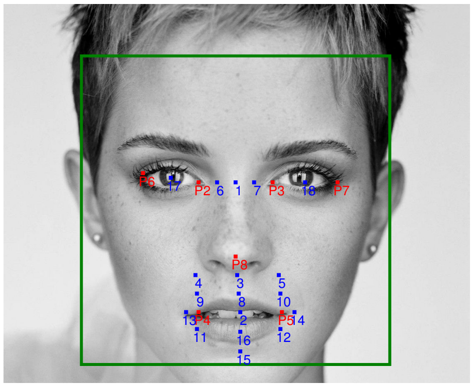


Fig. 3. Face bounding box (green), detected landmarks (red), and calculated keypoints (blue)

***Abstract*—This paper presents a method to generate a composite baby face from two parent images. Face detection, facial landmark identification, and feature extraction are first implemented to collect the relevant information from each parent image. Using this information, a facial morphing algorithm including the partitioning of features into triangles and quadrilaterals based on feature keypoints, warping triangles and quadrilaterals using affine transformations and homography respectively, and cross-dissolving the warped images is implemented. The composite adult features are then combined with the base baby features and placed onto the base baby face at the corresponding locations.**

I. Introduction

The identification, analysis, and modification of images containing human faces remains an active research area involving many image processing techniques. One subset of this research area, the automatic detection of faces and extraction of facial features has a wide range of applications from facial expression analysis to privacy protection in Google Street View [1]. Another subset of this research area, facial morphing, also has a wide range of applications including facial recognition and comically combining the faces of two individuals. The goal of this project was to leverage these two research areas for the entertaining application of intelligently morphing two parent images to generate a baby face.

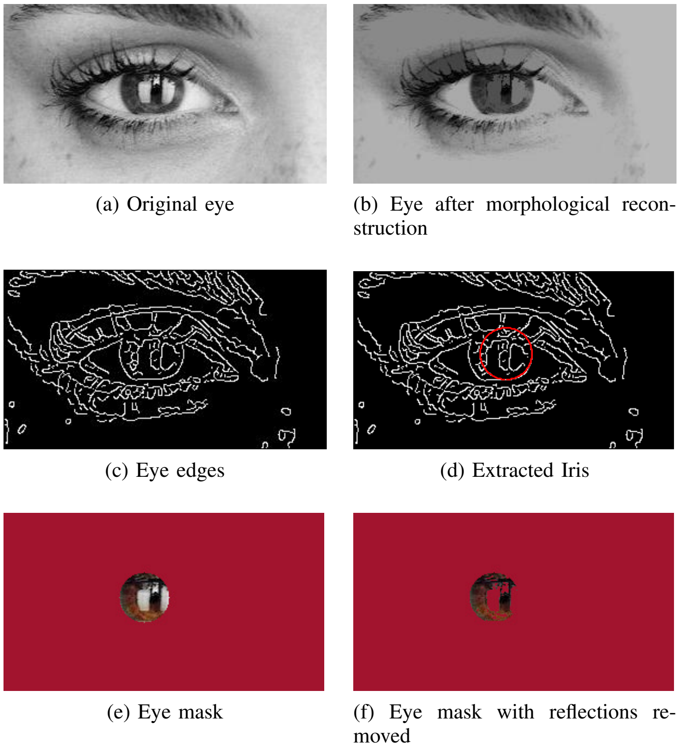
The algorithm used for generating the baby faces, seen in Fig. 1, involves three major components: facial feature extraction, base baby selection, and facial morphing. Sections II through IV discuss these methods and implementations in detail. The final system implementation included a MATLAB Graphical User Interface (GUI) seen in Fig. 2, allowing users to select the parent images either from a file or webcam snapshot and run the morphing code with ease.

II. Facial Feature Extraction

The first step to extracting the facial features for each parent image involved facial detection using the Viola-Jones face detector implemented in OpenCV [2]. For each parent image, the facial landmarks (eye corners, mouth corners, and nose tip) were detected using flandmark, an open source facial landmark detector [3]. The bounding box and facial landmarks were then used to calculate additional facial keypoints seen in Fig 3. Details of the keypoint calculations are given in Appendix A. These keypoints were then used in combination with edge detectors to extract each facial feature described in Sections II-A through II-D.

A. Nose

The extraction of the nose relied solely on the trapezoidal region formed by connecting keypoints 4, 5, 7, and 6 in a counterclockwise fashion. This choice followed from two observations: (1) the landmarks and keypoints surrounding the nose were detected with high accuracy and (2) the nose has


(a) Original eye

(b) Eye after morphological reconstruction

(c) Eye edges

(d) Extracted Iris

(e) Eye mask

(f) Eye mask with reflections removed

Fig. 5. Extraction of iris and eye color

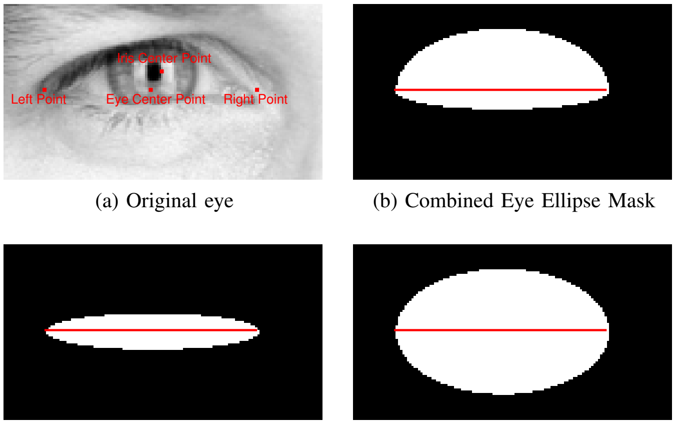
no sharp edges other than the nostrils so edge detection adds limited new information. Results of this nose extraction can be seen in Fig. 4b and Fig. 4g.

B. Eyes

Feature extraction of the eyes included extracting the shape of the eye as well as the eye color. The first step in extracting each eye involved finding the iris center location and radius. The algorithm for detecting the iris, based on [4], involved a combination of morphological reconstruction [5], edge detection using a Canny edge detector [6], and circle detection using a circle Hough transform. The first step in the algorithm involves extracting a large bounding box containing the eye of interest and converting the region to grayscale. Morphological reconstruction is then used to fill the reflections in the eyes and edges are found using the Canny edge detector. A circle Hough transform is computed on the edge image to detect the three most prominent circles [7]. The center location of each circle is then compared to the center point between the two corner locations of the eye. The iris circle is chosen to be the circle whose center point minimizes the distance between the two center locations.

To determine the eye color, a mask of the circle computed for the iris was applied to the eyes to extract the entire iris. Additionally, a reflection mask was constructed using a thresholded difference between the original eye and the eye after morphological reconstruction and overlaid on the iris mask. Using this mask, the HSV colors of the eye were extracted. The approach is illustrated in Fig. 5.

Using the corner landmarks, iris center, and iris radius, the mask of the eye shape was calculated by combining two ellipse-shaped masks. Because the eye often has a different distance from the corner points to the upper and lower lids of the eyes, using two ellipses accommodates for this difference by allowing for different minor-axis radii. The major-axis radius for both ellipses was calculated using half the distance between $(0.98X_{\text{left corner}}, Y_{\text{left corner}})$ and $(1.02X_{\text{right corner}}, Y_{\text{right corner}})$. The minor-axis radius for the upper ellipse was calculated as $1.2(r_{\text{iris}}+(y_{\text{eye center}}-y_{\text{iris center}})$ and the minor-axis radius for the lower ellipse was calculated as $1.2(r_{\text{iris}}-(y_{\text{eye center}}-y_{\text{iris center}})$ where r_{iris} is the calculated radius of the iris. The boundary line between the ellipses was calculated as the line connecting the two eye corners. An example of creating this mask for the left eye is shown in Fig. 6. Results of the eye extraction can be seen in Fig. 4c-d and Fig. 4h-i.



(a) Original eye

(b) Combined Eye Ellipse Mask

(c) Lower Ellipse with Dividing Line

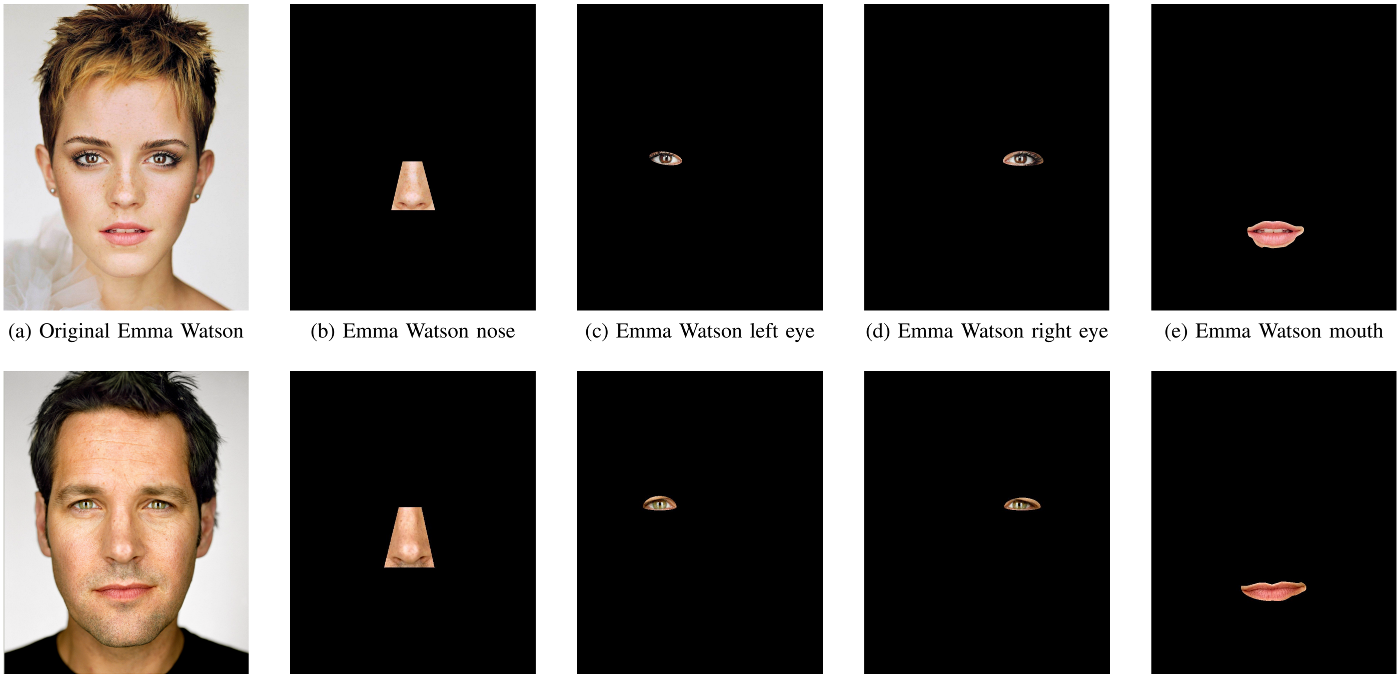
(d) Upper Ellipse with Dividing Line

Fig. 6. Extraction of eye

C. Mouth

A two-fold process was implemented to extract the mouth of each parent image. First, the method described in Section II-B of using two ellipse masks was applied to extract a general mask for the mouth. The second portion of the algorithm to extract the mouth involved edge detection on a lip-enhanced image. Transforming from RGB color space into the lip-enhanced color space described by $\frac{1}{4}(2G-R-0.5B)$ greatly increased the success of the lip edge detector because it clearly differentiates between skin and lip color as seen in Fig. 7a [8]. Edge detection using a Laplacian of Gaussian operator was then performed on the lip-enhanced image as seen in Fig. 7b. The edges closest to the mouth landmarks, $P4$ and $P5$, were found and if the edge closest to $P4$ was connected to the edge closest to $P5$, this edge was identified as the lip edge. If this condition was not met, the edge mask was discarded and the two ellipse mask was the only mask used.

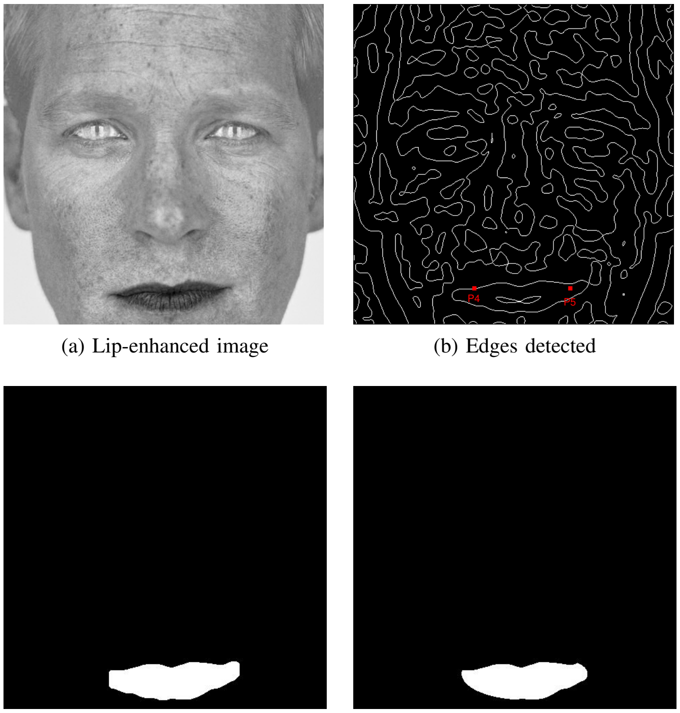
To create the edge-based mask for the mouth, the edges of the lips were filled and then slightly dilated to ensure the lips were not cut off. Finally, the two ellipse mask was overlaid on the mouth edge mask to form the final mouth mask. Results of the mouth extraction can be seen in Fig. 4e and Fig. 4j.



(a) Original Emma Watson (b) Emma Watson nose (c) Emma Watson left eye (d) Emma Watson right eye (e) Emma Watson mouth

(f) Original Ben Affleck (g) Ben Affleck nose (h) Ben Affleck left eye (i) Ben Affleck right eye (j) Ben Affleck mouth

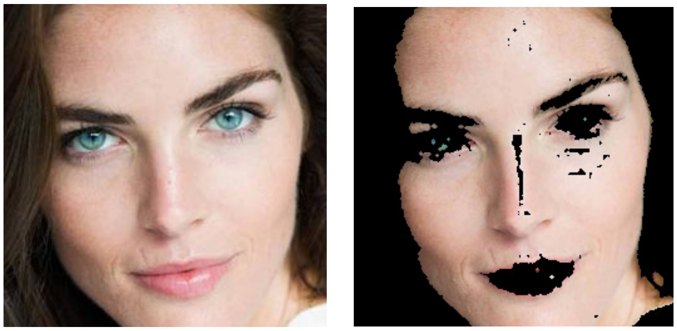
Fig. 4. Extracted facial features



(a) Lip-enhanced image (b) Edges detected

(c) Mouth mask based on lip edges (d) Combined mouth masks

Fig. 7. Extraction of mouth



(a) Original facial bounding box of Hillary Rhoda (b) Skin regions detected

Fig. 8. Extraction of skin regions

D. *Skin*

The skin color was the final feature extracted from each parent image and was based on segmenting skin colors by six planes in HSV color space [9]. Using this HSV color space segmentation, skin regions were successfully identified and used to calculate the mean HSV values for each parent skin color. A resulting skin mask can be seen in Fig. 8.

III. Base Baby Selection

A baby database including nine babies of varying skin and eye colors was curated and compared to the adult facial features to select the baby of closest resemblance to the parent images. Because the flandmark library was based on an adult frontal face model and baby facial geometry differs significantly from adult facial geometry, the facial landmarks were manually identified. The facial keypoints were then calculated

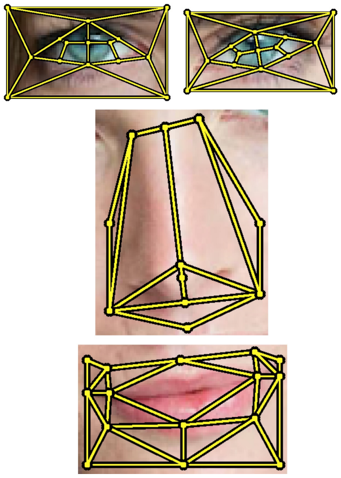


Fig. 9. Partitioned facial features of Hillary Rhoda

using slightly modified facial keypoint equations adjusted for baby facial geometry. Finally, the features were extracted using the methods described in Sections II-A to II-D.

To select the base baby to be morphed with the parent images, the primary consideration was skin color. This comparison was calculated by linearly combining the HSV values of each parent's skin color to form a composite skin color and finding the baby skin color closest to this composite skin color in HSV color space. Additionally, if both parents had green or blue eyes, a baby with blue or green eyes was selected.

IV. Feature Morphing

The combined facial features (i.e. eyes, nose, and mouth) of two parents and a base baby are generated using a morphing algorithm based on mesh warping [10], [11]. Four phases are involved in this algorithm: image partitioning, warping, cross-dissolving, and generating a composite baby image.

A. Image Partitioning

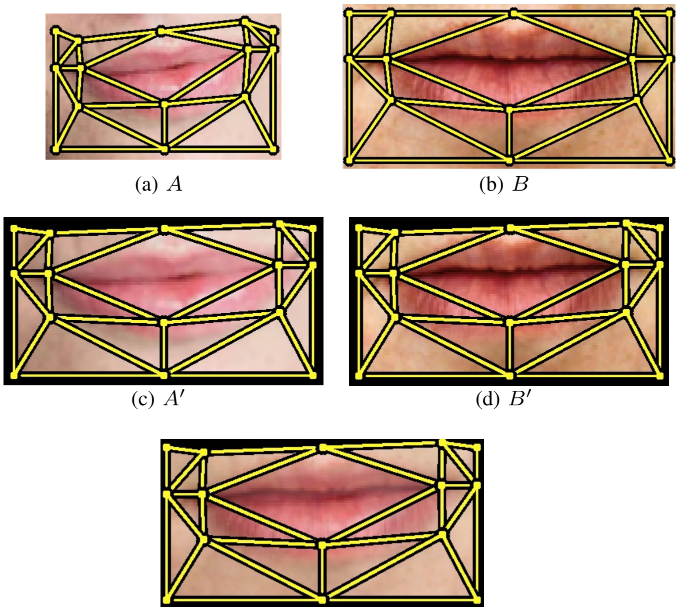
Each facial feature is first partitioned into triangles and quadrilaterals based on the extracted facial keypoints as shown in Fig. 9.

B. Warping

When the facial features of the two parents are referred to as images A and B, the image A is to be warped to A' and the image B to B' so that A' and B' have the same shape as shown in Fig. 10a-d. The shape of A' and B' is calculated using linear interpolation of the corresponding grid point locations of A and B. In the warping process, an affine transformation is applied to map the coordinates of the triangles in the original image to the triangles in the warped image, while homography is used for the quadrilaterals. The equation for coordinate transformation is

$$\underline{\mathbf{x}}\text{'} = T\underline{\mathbf{x}}$$

where $\underline{\mathbf{x}}$ and $\underline{\mathbf{x}}$' are homogeneous coordinates in the form $[x \quad y \quad 1]^T$ of the original and warped images, respectively.



(a) A

(b) B

(c) A'

(d) B'

(e) Cross-dissolving of A' and B', I

Fig. 10. Morphing of two parent features, A and B

For affine transformations,

$$T = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ 0 & 0 & 1 \end{bmatrix}$$

and for homography,

$$T = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}.$$

The implementation maps the coordinates of pixels in the target image to the coordinates in the original image and assigns the interpolated color of the original image to the target pixel using

$$\underline{\mathbf{x}} = T^{-1}\underline{\mathbf{x}}\text{'}.$$

This guarantees that every pixel in the warped image is assigned a value.

C. Cross-Dissolving

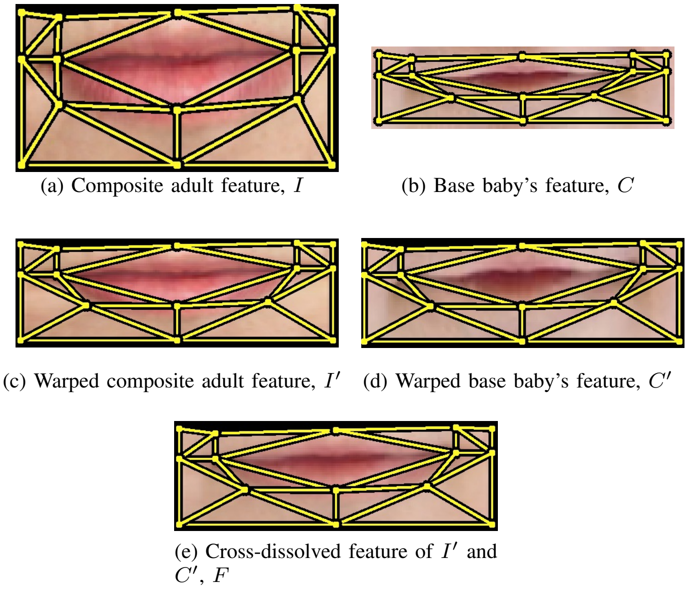
After the two feature images are warped to the same intermediate grid locations, they are cross-dissolved by calculating the weighted average of the two images using

$$I[x, y] = \alpha A'[x, y] + (1 - \alpha)B'[x, y]$$

where $0 < \alpha < 1$. The cross-dissolve of the two parent features in Fig. 10c and Fig. 10d is shown in Fig. 10e.

D. Generating a Composite Baby Image

The composite adult features are then morphed with the respective features of the selected base baby using the morphing steps (image partitioning, warping, and cross-dissolving) previously described. This is required to make the final combined features look like those of a baby since certain aspects of baby features differ from adult facial features. For example, babies


(a) Composite adult feature, I

(b) Base baby's feature, C

(c) Warped composite adult feature, I'

(d) Warped base baby's feature, C'

(e) Cross-dissolved feature of I' and C', F

Fig. 11. Morphing between composite adult feature and base baby's feature

have larger irises and flatter nose bridges. Results of this final warping step can be seen in Fig 11.

Next, the final composite features, F, are pasted onto the base baby at the same center location as the extracted baby features. Smoothing is employed at the boundary of the features in order to create a smooth transition from the composite features to the outer areas in the base baby's face. A binary mask around F, BW, is created, open filtered and eroded to attain a smoother and smaller binary mask. A Gaussian filter is applied to the eroded binary mask to create weight mask σ. Depending on the facial feature, either an average or a Gaussian filter is applied to the inverted binary mask of BW to create weight mask β. Fig. 12 illustrates the weight masks used in the smoothing process.

The feature to be placed onto the base baby face is calculated from the following equation.

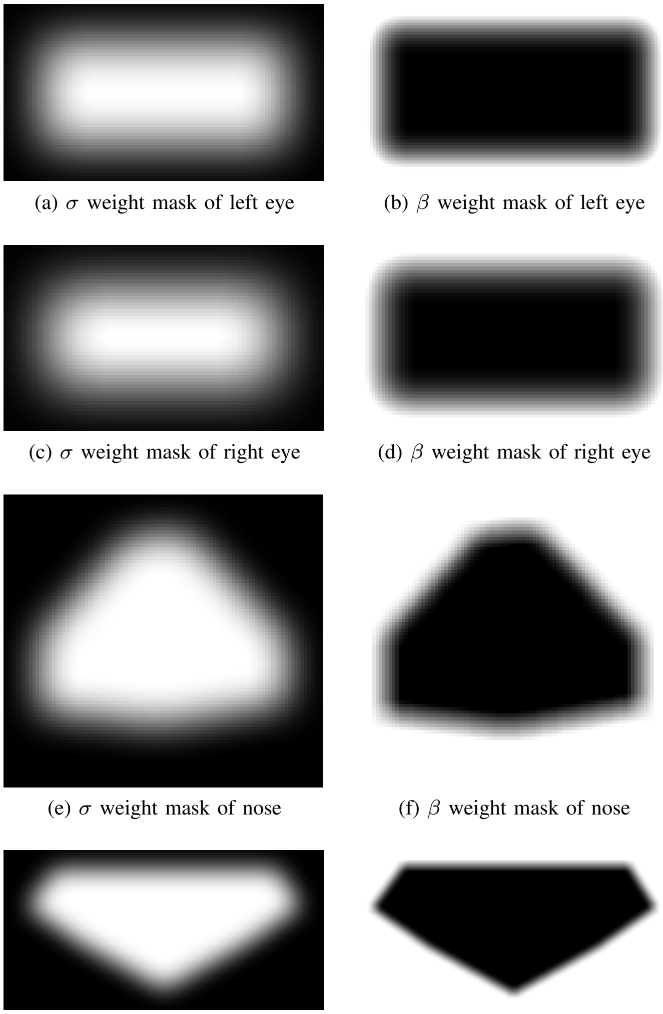
$$\text{Feature} = \beta I_{\text{Baby}} + (1-\beta)(\sigma F + (1-\sigma)C')$$

where I_{Baby} is the base baby image.

Smoothing with the warped baby feature, C', is included because the original base baby features can be too different from the composite features to apply a gradual transition along the contour of the features. For instance, the base baby may have much wider nose than the composite nose, making the side of the base baby's nose appear in the final image if gradual transition is applied. To accommodate for this, more surrounding regions are included in the extracted features.

V. Results

Our proposed Baby Face Generator program was tested on 30 different adult images. The sample results are presented in Fig. 13. From qualitative examination, the generated baby images have features that pleasantly resemble the two parents, especially the shape of nose and mouth, eye color, and skin color.


(a) σ weight mask of left eye

(b) β weight mask of left eye

(c) σ weight mask of right eye

(d) β weight mask of right eye

(e) σ weight mask of nose

(f) β weight mask of nose

(g) σ weight mask of mouth

(h) β weight mask of mouth

Fig. 12. Weight masks used in smoothing boundaries of each feature

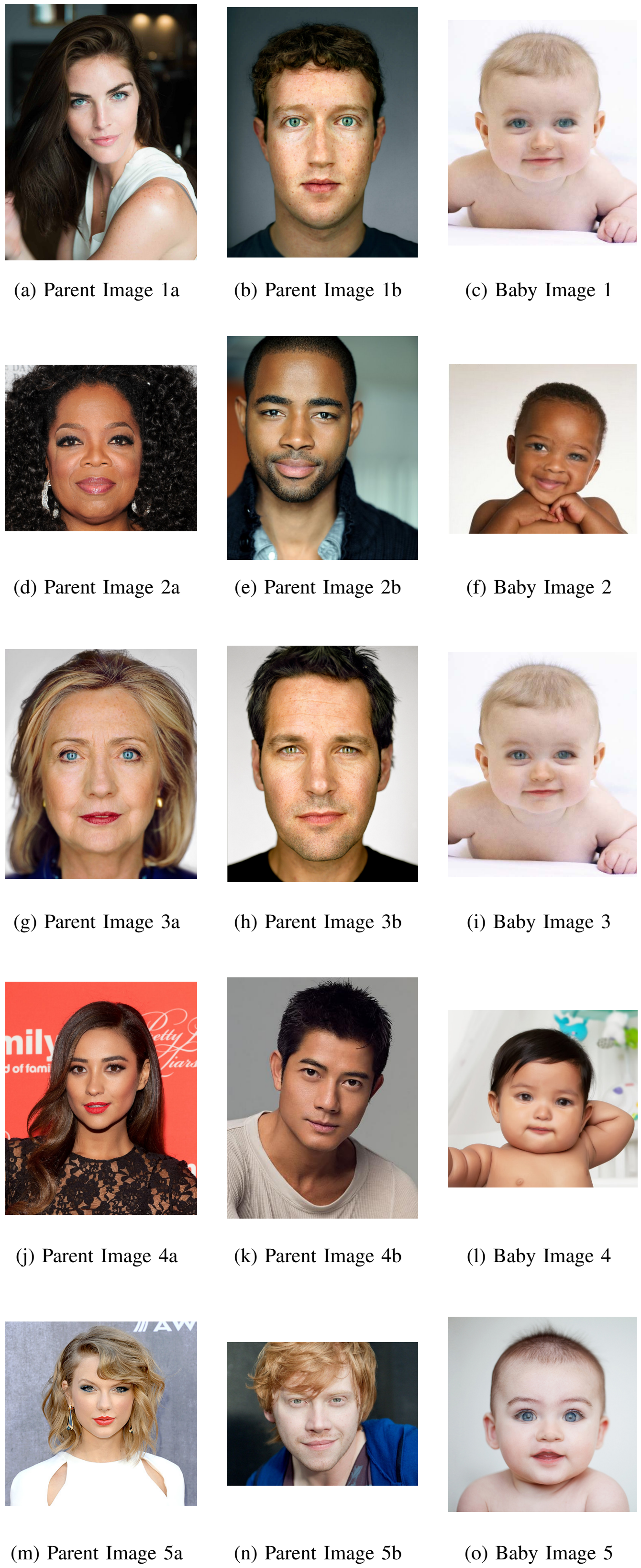
Significant artifacts, however, can occur if a parent's face is turned by a considerable angle. Nonetheless, the program can be constrained to only accept forward facing input faces since it is intended to interact with the parent users directly.

VI. Conclusion

In conclusion, an automatic Baby Generator Program has been developed. Facial features extraction and facial morphing algorithms have been employed to produce a satisfying composite baby face of two parent images. Further improvements could be achieved by using a larger database of base baby images and adding the detection of parents' ethnicity to enhance the selection of base baby images. Additionally, an age progression algorithm could be implemented to first generate a baby face of each parent before morphing the two together. This could potentially reduce artifacts introduced by the base baby and the age-variant features.

Acknowledgment

The authors would like to thank Professor Bernd Girod, Professor Gordon Wetzstein, Huizhong Chen, and Jean-



(a) Parent Image 1a (b) Parent Image 1b (c) Baby Image 1

(d) Parent Image 2a (e) Parent Image 2b (f) Baby Image 2

(g) Parent Image 3a (h) Parent Image 3b (i) Baby Image 3

(j) Parent Image 4a (k) Parent Image 4b (l) Baby Image 4

(m) Parent Image 5a (n) Parent Image 5b (o) Baby Image 5

Fig. 13. Example results of generated baby images. The right column shows the generated baby image of the parent images in the left and middle columns

Baptiste Boin for teaching Digital Image Processing and for teaching us the skills to succeed in this project. The authors would especially like to thank Huizhong Chen, their project mentor, for his guidance and assistance.

References

[1] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, "Large-scale privacy protection in google street view," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2373–2380.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–511.

[3] M. Uřičář, V. Franc, and V. Hlaváč, "Detector of facial landmarks learned by the structured output SVM," in *VISAPP '12: Proceedings of the 7th International Conference on Computer Vision Theory and Applications*, G. Csurka and J. Braz, Eds., vol. 1. Portugal: SciTePress — Science and Technology Publications, February 2012, pp. 547–556.

[4] A. Dantcheva, N. Erdogmus, and J. Dugelay, "On the reliability of eye color as a soft biometric trait," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. IEEE, 2011, pp. 227–231.

[5] L. Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms," *Image Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 176–201, 1993.

[6] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[7] D. Young, "Hough transform for circles," MATLAB Central File Exchange, 2010.

[8] U. Canzler and T. Dziurzyk, "Extraction of non manual features for video based sign language recognition," in *IAPR workshop on machine vision applications*, 2002, pp. 318–321.

[9] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *Trans. Multi.*, vol. 1, no. 3, pp. 264–277, Sep. 1999. [Online]. Available: http://dx.doi.org/10.1109/6046.784465

[10] D. B. Smythe, "A two-pass mesh warping algorithm for object transformation and image interpolation," *Rapport technique*, vol. 1030, p. 31, 1990.

[11] G. Wolberg, "Image morphing: a survey," *The visual computer*, vol. 14, no. 8, pp. 360–372, 1998.

Appendix A
Facial Keypoint Equations

When the top left corner of the facial bounding box has coordinates (X_{B1}, Y_{B1}), the bottom left corner has coordinates (X_{B1}, Y_{B2}), top right corner has coordinates (X_{B2}, Y_{B1}), and the detected facial landmarks have coordinates $(X_{P2}, Y_{P2}), (X_{P3}, Y_{P3}), \ldots, (X_{P8}, Y_{P8})$, the keypoints surrounding the nose seen in Fig. 3, $(X_1, Y_1), (X_2, Y_2), \ldots, (X_7, Y_7)$ are calculated as follows:

$$\begin{aligned}
X_1 &= \frac{1}{2}\big(X_{P2} + X_{P3}\big) & Y_1 &= \frac{1}{2}\big(Y_{P2} + Y_{P3}\big)\\
X_2 &= \frac{1}{2}\big(X_{P4} + X_{P5}\big) & Y_2 &= \frac{1}{2}\big(Y_{P4} + Y_{P5}\big)\\
X_3 &= \frac{1}{3}\big(2X_{P8} + X_2\big) & Y_3 &= \frac{1}{3}\big(2Y_{P8} + Y_2\big)\\
X_4 &= X_3 - X_2 + X_{P4} & Y_4 &= Y_3 - Y_2 + Y_{P4}\\
X_5 &= X_3 - X_2 + X_{P5} & Y_5 &= Y_5 - Y_5 + Y_{P5}\\
X_6 &= \frac{1}{2}\big(X_{P2} + X_1\big) & Y_6 &= \frac{1}{2}\big(Y_{P2} + Y_1\big)\\
X_7 &= \frac{1}{2}\big(X_{P3} + X_1\big) & Y_7 &= \frac{1}{2}\big(Y_{P3} + Y_1\big)
\end{aligned}$$

The keypoints surrounding the mouth seen in Fig. 3, $(X_8, Y_8), (X_9, Y_9), \ldots, (X_{16}, Y_{16})$ are calculated as follows:

$$\begin{aligned}
X_8 &= \frac{1}{3}\big(X_{P8} + 2X_2\big) & Y_8 &= \frac{1}{3}\big(Y_{P8} + 2Y_2\big)\\
X_9 &= X_8 - X_2 + X_{P4} & Y_9 &= Y_8 - Y_2 + Y_{P4}\\
X_{10} &= X_8 - X_2 + X_{P5} & Y_{10} &= Y_8 - Y_2 + Y_{P5}\\
X_{11} &= X_9 & Y_{11} &= \frac{1}{2}\big(Y_9 + Y_{B2}\big)\\
X_{12} &= X_{10} & Y_{12} &= \frac{1}{2}\big(Y_{10} + Y_{B2}\big)\\
X_{13} &= X_{P4} - 0.3\big(X_2 - X_{P4}\big) & Y_{13} &= Y_{P4}\\
X_{14} &= X_{P5} + 0.3\big(X_{P5} - X_2\big) & Y_{13} &= Y_{P5}\\
X_{15} &= X_2 & Y_{15} &= \frac{1}{4}\big(Y_2 + 3Y_{B2}\big)\\
X_{16} &= X_2 & Y_{16} &= \frac{1}{2}\big(Y_2 + Y_{B2}\big)
\end{aligned}$$

The keypoints in the eye seen in Fig. 3, $(X17, Y_{17}), (X_{18}, Y_{18})$ are calculated as follows:

$$\begin{aligned}
X_{17} &= \frac{1}{2}\big(X_{P2} + X_{P6}\big) & Y_{17} &= \frac{1}{2}\big(Y_{P2} + Y_{P6}\big)\\
X_{18} &= \frac{1}{2}\big(X_{P3} + X_{P7}\big) & Y_{18} &= \frac{1}{2}\big(Y_{P3} + Y_{P7}\big)
\end{aligned}$$

Appendix B
Work Breakdown

1) Sarah: Integrated external libraries for face detection and facial keypoint detection. Wrote code for facial feature extraction, skin and eye color classification, and database baby selection. Developed MATLAB GUI and incorporated webcam interface.
2) Picha: Wrote code for iris detection, facial morphing, and creation of the output baby image.