Automatic Face Replacement on a Morphing Model

Han Li
Department of Computer Sciences
University of Wisconsin-Madison
hli337@wisc.edu

Wentao Wu
Department of Computer Sciences
University of Wisconsin-Madison
wwu69@wisc.edu

Abstract

With the unprecedented and exponentially increasing of web images and videos, there is a growing concern of online iconographic privacy. Thus, editing and rewriting these image-based media resources is becoming an interesting and urgent task. In this project, we propose a new automatic face replacement in a resource (image or video) based on face morphing and blending from an existing facial data resource. The procedure consists of four steps: (1) face recognition and landmark representation based on a third party computer vision library, Face ++, (2) face morphing using Thin-Plate-Spline method, (3) optimal image seam searching based on a graph-cut algorithm, and (4) face blending with color and illumination adjusting based on Poisson blending. We conduct face replacement tasks on both images and videos. The empirical experiments show that the results not only look natural to human judgements but also can handle the facial expressions well, and proved the effectiveness of the method.

1. Introduction

With the unprecedented and exponentially increasing of high-definition web images and videos, online iconographic privacy is becoming a severe problem and concerned by more and more people. For example, online systems such as Google Street View allow users to browse photos of public images possibly containing many people who might not consent to be photographed. However, currently Google solves this problem by using blur masks, which is a typical, low-cost way but makes the image looks pretty ugly. Fig. 1 shows an example from Google Street View that contains a child face with blur mask. Thus, it becomes more and more urgent and popular to develop face replacement techniques to alleviate this problem. In this project, we propose a new face replacement method based on a shape-morphing model for image and video facial information hiding. Given a source image and a target resource (image or video), the pipeline contains four steps:

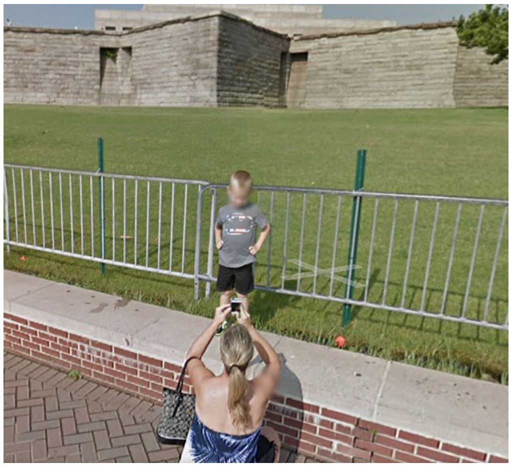


Figure 1: An image from Google Street View that contains a human face with blur mask.

1. face detection and landmark localization based on Face ++, a third party library focusing on computer vision;
2. face morphing according to the landmarks by Thin-Plate-Spline method to fit the source face into the target image;
3. optimal seam searching to include the most important part of the face;
4. face blending with possible color and illumination adjusting to make the synthesized face look natural.

We test our method on a set of images and two videos. The empirical experiments show that the results look natural to human judgements. Also, since many of the previous methods haven't cared much about the facial expressions, our method could handle them well, and thus the contribution of this project would be integrating the facial expression into the face replacement procedure.

We organize the paper as follow. We introduce the background and some related work in Section 2, describe each of the pipeline steps in detail from Section 3 to Section 6. We show our experimental results in Section 7. And finally, we make the conclusion and prospect some future work in Section 8.

2. Related Work

Our project is most related to Min's work [6]. They proposed an automatic face replacement approach in video based on 2D Morphable model. This approach includes three main modules: face alignment, face morph, and face fusion. They use the Active Shape Models (ASM) for face alignment. They also consider color and lightning adjustment to keep consistent. Their approach is fully automatic without user interference, and generates natural and realistic results. Furthermore, as they use 2D model, their algorithm is also efficient. However, they don't consider facial expression. The tolerance to pose and expression variance is limited by ASM.

Liang [5] proposed a video face replacement system which allows replacing target human face from target video with source face in source video. For each target face in target video, they select the best candidate face with similar face expression. Finally, they blend candidate replacement to target video. In their approach, however, they require the target video should have similar face expression and pose with the source video.

Similar to Liang's work, Dale [3] also proposed an algorithm to provides face replacement in target video from source video. They tracks both faces in target and source video using multilinear model. Using this tracked 3D Geometry, source face is warped to target face in every frame of video. But, their tracking algorithm is based on optical flow, so the light should change slowly in the video.

Afifi [1] presented a system for video face replacement that requires only two videos of a source actor and a target actor using only a single digital camera. They could generate realistic results without using special equipment and 3D model. Also, they use a new face blending technique based on poisson blending. But their algorithm only works for fixed pose, i.e. front face.

Instead of using source video, Cheng [2] uses only two face pictures: one frontal view and one profile view. First, they use these two images to construct 3D face model. Then, they track the faces in the video and project the source face model to align and replacement the faces in target video. However, the tolerance to the pose is limited to robustness of their alignment algorithm.

A. Niswar [7] proposed a system where one image is required to replace the faces in target video. Also, the image is not limited a specific pose. There are four steps in their approach: 3D face reconstruction, 3D face animation, feature points tracking, 2D projection with blending. Their approach, however, has high complexity and hence high time consuming.

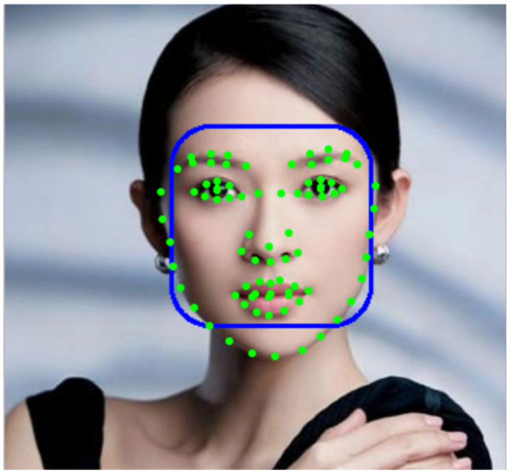


Figure 2: Landmarks in a human face. The landmark point set contains face contour, nose, mouth, eyes, and eyebrows.

3. Landmark Localization

To replace a face in an image, the first thing we need to do is to recognize the face(s) in the given image. In this project, we use *landmarks* as the representation of a face. The landmarks are a set of key points on a face that sketch the outline of a face. Fig. 2 shows an example of landmarks in a face. There are two reasons we use landmarks. First, we can use the convex hull of the landmarks as the face region recognition. A more important reason is that landmarks provide us the anchors for face morphing in the next step, which would be explained in detail in the next section.

Since the landmark localization is the fundamental step of our pipeline, and the quality of the landmarks would influence the performance of our method greatly, in this project we choose to use a third party library, Face ++[1], for this task. The Face ++ landmark localizer is based on a deep convolutional neural network proposed by Zhou *et al*. [9], and for each image, it provides 83 landmarks. Fig. 3 shows the system workflow. The basic idea is to use 4 hidden layers to refine the results. The first level networks predict the bounding boxes for the inner points and contour points separately. For the inner points, the second level predicted a initial estimation of the positions which are refined by the third level for each component. The fourth level is used to further improve the predictions of mouth and eyes by taking the rotated image patch as input. Two levels are used for contour points. Currently, the Face ++ achieves the state-of-the-art performance on landmark localization.

Because Face ++ is not open source, we use the mat-

[1] http://www.faceplusplus.com/

lab API to landmark localizer which uploads the image to the server and returns the coordinates of the landmarks in the image. The code can be downloaded directly from the website [2].

4. Face Morphing

Based on the landmarks generated from the first step, the next thing to do is face morphing. The reason we need to perform morphing is that people have different faces, e.g. different shape of contour, size of eyes, size of nose. Even if we have the convex hull of the face contour from the landmarks, we cannot directly replace old face with the new one without any morphing. We need to change the structure of the face to make sure that the components in the new face have the similar relative distance from each other as in the old face.

To achieve this goal, we choose to use Thin-Plate-Spline (TPS) method. TPS is an interpolation method for non-rigid surface morphing. Here the name ”thin plate” refers to a physical analogy involving the bending of a thin sheet of metal to get the non-rigid surface morphing. Given a set of 2D data points, the basic idea of TPS is to produce a weighted combination of thin plate splines centered at each designated point which gives the interpolation function that passes through the points exactly and also minimizes the *blending energy* of the surface. Here the blending energy is defined as follow:

$$I[f(x,y)] = \int\int_{R^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)dxdy$$

There are three reasons that we select TPS for face morphing: (1) TPS produces non-rigid, smooth surface morphing which is required by our task; (2) there are no free parameters in TPS that need manual tuning; (3) TPS has closed-form solutions for parameter estimation and morphing, which guarantees the performance and also the training speed.

In the face replacement scenario, to morph the source face to the target one, there are two steps: first, align all of the source landmarks to the corresponding target ones to get the interpolation functions; second, perform TPS morphing based on the interpolation functions. Formally, given the source landmark vector $P_{src} = (s_1, ..., s_K)$, target landmark vector $P_{tgt} = (t_1, ..., t_K)$ with K landmarks each, and function $U(r) = r^2 \log r^2$ with $r = \sqrt{x^2 + y^2}$, we need to solve the following equation:

$$\begin{bmatrix} K & P_{src} \\ P_{src}^\top & 0 \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} P_{tgt} \\ 0 \end{bmatrix} \qquad (1)$$

where w and a are the model parameters that we need to estimate, and

$$K_{ij} = U(||(x_{src,i}, y_{src,i}) - (x_{src,j}, y_{src,j})||)$$

Note that we could expand Eq. 1 to get the following form:

$$f(x,y) = a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U(||(x_i, y_i) - (x_j, y_j)||) \qquad (2)$$

thus once we get the parameter estimation from Eq. 1, we could fit the parameters into Eq. 2 and perform the point morphing in the whole image.

Fig. 4 shows an example of face morphing after applying TPS model. Now we can see that the new morphed face is pretty similar to the target one with respect to the face components, and could fit into the new image perfectly.

5. Optimal Seam Searching

After we warped the two images, we need to find an optimal seam. A seam is a series of continuous pixels along which the area extracted from the source image can be determined. There are three main concerns for the optimal search. First, we want to avoid the the important features in face, such as eyes, mouth, nose and so on. Second, the pixels on the boundary of the extracted area should be similar between the two images so that it will look more natural when we composite them together. Third, we want to reduce the computation complexity.

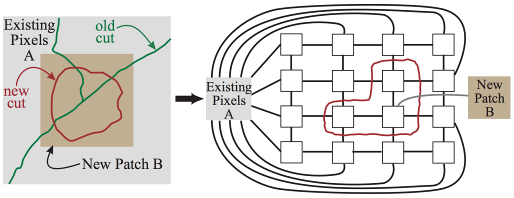


Figure 5: Graph representation of image, from [4]

Our approach is based on Kwatra et al’s work [4]. The algorithm works by formulating the image as a connected graph with edge weights based on the difference between neighboring pixels. This graph is then treated as a max-flow/min-cut problem where the sources are any pixels to be taken only from the first image and the sinks are any pixels to be taken only from the second image.

As the figure 5 shows, in our project, the pixels in patch A must from our target image while the pixels in patch B must from source image. The goal is to find the optimal cut between patch A and patch B. We represent the image graph as an adjacency matrix while each entry in the adjacency matrix can be the flow. We use the sum of absolute difference between the source image and the target image

[2] http://www.faceplusplus.com/dev-tools-sdks/

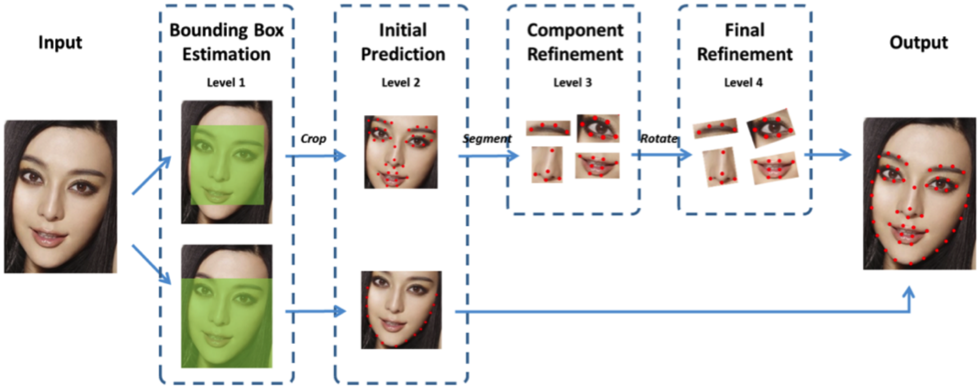


Figure 3: System workflow of Face ++.

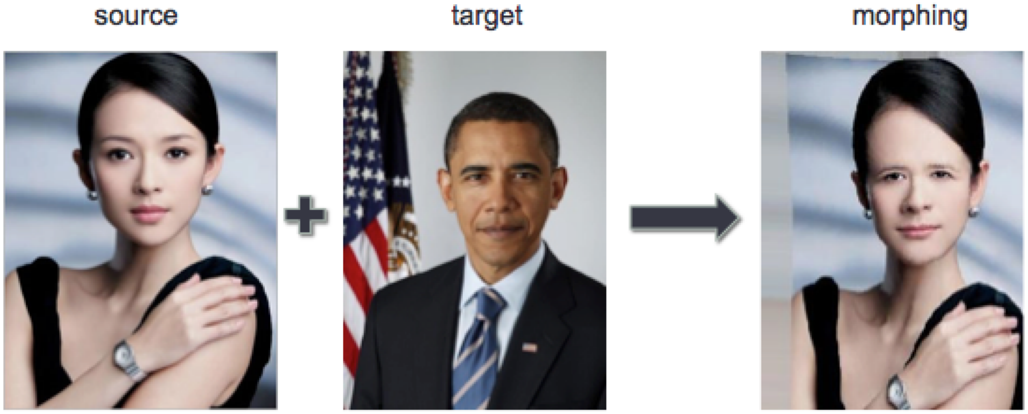


Figure 4: An example of face morphing.

plus the sum of absolute difference of the source image and the target image at the neighboring pixel.

$$M(c, n, S, T) = \|S(c) - T(c)\|_1 + \|S(n) - T(n)\|_1 \quad (3)$$

where c is the current pixel, n is the neighbor pixel, S is the source image and T is the target image. The $L1$ norm is the sum of absolute difference of these two pixels in three channels. Note that the weight between the pixels in patch A and its neighbors must be infinity. This is same for patch B.

In our implementation, we first define the permissible path region by generating two masks based on the landmarks generated from the first step, as shown in 6. The permissible path region is the area we want to find the optimal cut. All the pixels which are outside the right larger mask must from target image, i.e. patch B, and all the pixels inside the left smaller mask must from source image, i.e. patch A. Therefore, the permissible path region is the area which is inside the larger mask but outside the smaller mask. A good permissible region is crucial for the following steps. The permissible path should not be too strict, then it's highly possible that we cannot find the optimal seam within the strict area. Also, the larger the permissible path region is, the higher computation complexity will be. When constructing the two masks, a convex hull is constructed by the landmarks. As the forehead region of a face is usually more flat than the other regions, we split the face into two regions: the forehead region and the bottom region. For the forehead region, as the convex hull is right above the eyes, the mask is eroded so that the permissible path region will cover larger area. For the bottom region, the convex hull is almost at the boundary of the face, so it is not reasonable to extend the permissible path outside the face, the mask is dilated. After generating two masks for each region via erosion and dilation, we stitch them together to get the final permissible path region.

For the construction of the adjacency matrix, we extend the patch A and patch B by including all the pixels which are immediate neighbors. In other words, if the neighbor of a pixel is in patch A or B, then the weight between them is also infinite. If the weight between two pixels is not infinite, then these two pixels must be 1) not in patch A and patch B and 2) their neighbors are also not in patch A and patch B.

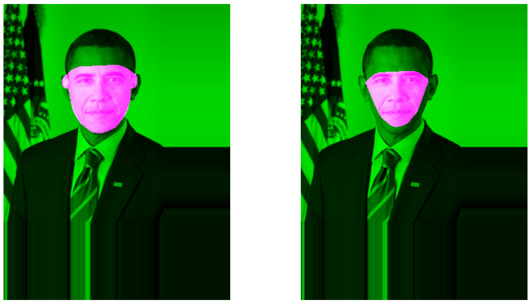


Figure 6: Mask for optimal seam search

The result is shown in Fig.7. The pink area is the permissible path region and the red circle in the left figure is the result.

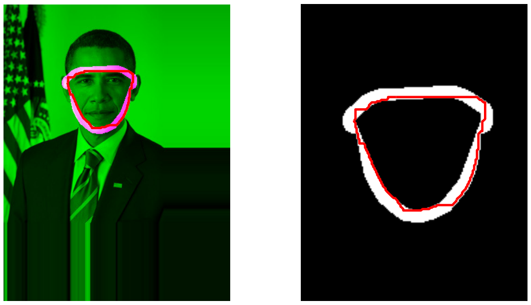


Figure 7: Result for optimal seam search

6. Face Blending

The optimal seam tells us what is the best area which should be extract from the source image. If we simply cut that area from the source image and paste it to the target image, the result will not be satisfiable, as the left image shown in Fig9. The problem is the color of these two images are very different from each other. Therefore, we can see the color change abruptly around the boundary. In order to avoid the abrupt color changes, we use Poisson blending approach.

Poisson blending is one of gradient domain image processing methods. As Fig.8 shown, v is gradient of a region in an image, g is selected region of source, f^* is a known functions that exist in domain S, f is an unknown functions that exist in domain Ω, Ω is a region g that is now placed on domain S (target background), $\partial\Omega$ is boundaries between the source and target regions. The goal is given v, find the value of f in the unknown region that optimize

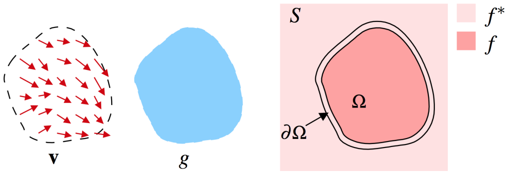


Figure 8: Poisson blending notation, [8]

$$min_f \quad \iint_\Omega |\nabla f - v|^2 \tag{4}$$

$$\text{subject to} \quad f\,|_{\partial\Omega} = f^*\,|_{\partial\Omega} \tag{5}$$

where $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ is the gradient vector. Its solution is the unique solution of the following Poisson equation with Dirichlet boundary conditions:

$$\Delta f = \text{div } v \quad \text{over } \Omega \tag{6}$$

$$\text{subject to} \quad f\,|_{\partial\Omega} = f^*\,|_{\partial\Omega} \tag{7}$$

where div$v = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial x}$ is the divergence of $v = (u, v)$, and Δ is the Laplacian operator as Matrix 1 shown.

0	-1	0
-1	4	-1
0	-1	0

Table 1: Laplacian Operator

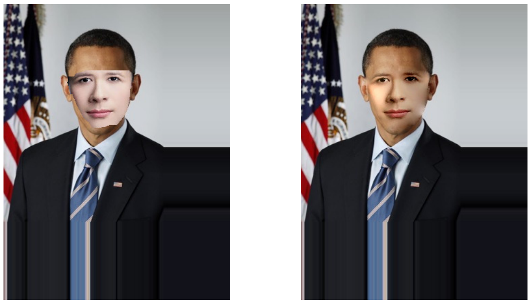
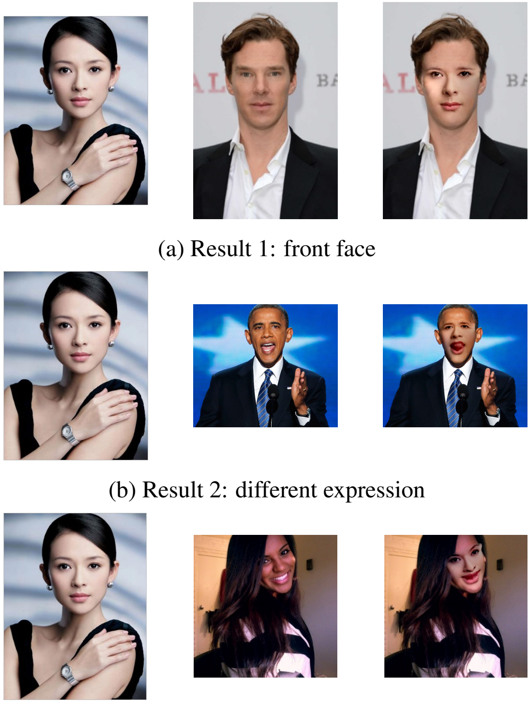


Figure 9: Poisson blending

This is the fundamental machinery of Poisson editing of color images: three Poisson equations of the above equation are solved independently in the three color channels of the chosen color space. The result is shown in the left image in Fig.9.



(a) Result 1: front face

(b) Result 2: different expression

(c) Result 3: part face

Figure 10: Results

7. Experiments

We tested our algorithm with various images. The results are shown in Fig.10 and Fig.11. Each example shows, from left to right, the source image, the target image and the composite image. Note that even the skin color in the source image and target are different, the skin color in composite image looks quite natural by using Poisson blending as shown in Fig.10a. One issue in previous work is that it's usually difficult to handle the different expressions in the source and target image. In our approach, using Thin-Plate-Spline method after extracting the facial landmarks, the source image can be morphed to the target image perfectly even the two images have quite different facial expression, as shown in Fig.10b. Note that the mouth in the target image is open while it is closed in the source image.

If the face is covered by some objects, such as hair or hands, our approach may not generate satisfiable results. As Fig.10c shows, part of the face in the target image is covered by her hair. In the composite image, that part of hair is replaced by the face from the source image. This is because we construct the masks based on the facial landmarks and we assume the whole face, i.e. the convex hull of the landmarks, appears in the image. If this assumption is violated, the composite image will look unnatural.

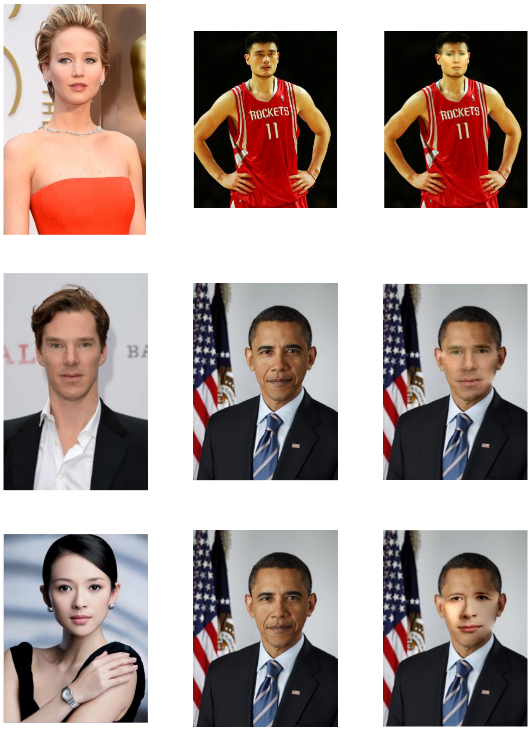


Figure 11: More Results

8. Conclusion and Future Work

In this project, we propose a new automatic face replacement method based on Thin-Plate-Spline morphing algorithm. The method is fully automated without any human intervention, and the four-step pipeline guarantees the performance of the procedure. The empirical experiments show that this new method could fit a source face into the target image effectively and also capture the facial expressions, thus does a perfect job on front face replacement both in images and videos.

To make the method even more powerful, several improvements can be done in future. First, currently we just use one image as the source image, and it would inevitably cause the bad performance on side face morphing. To alleviate this, we could instead use a set of images containing different poses. Second, because the Face ++ cannot recognize some side faces, and thus leads to discontinuous frames in the video face replacement, in future we could use some interpolation methods to plug in the missing frames using the neighborhoods. Finally, a better color adjustment might be invented for better image blending.

Acknowledgments

We thank Prof. Singh and TA Jia for their devotion and meticulous guidance during this semester.

References

[1] M. Afifi, K. F. Hussain, H. M. Ibrahim, and N. M. Omar. Video face replacement system using a modified poisson blending technique. In *Intelligent Signal Processing and Communication Systems (ISPACS), 2014 International Symposium on*, pages 205–210. IEEE, 2014. 2

[2] Y.-T. Cheng, V. Tzeng, Y. Liang, C.-C. Wang, B.-Y. Chen, Y.-Y. Chuang, and M. Ouhyoung. 3d-model-based face replacement in video. In *SIGGRAPH'09: Posters*, page 29. ACM, 2009. 2

[3] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. Video face replacement. *ACM Transactions on Graphics (TOG)*, 30(6):130, 2011. 2

[4] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003. 3

[5] Y. Liang, B.-Y. Chen, Y.-Y. Chuang, and M. Ouhyoung. Image based face replacement in video. 2

[6] F. Min, N. Sang, and Z. Wang. Automatic face replacement in video based on 2d morphable model. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2250–2253. IEEE, 2010. 2

[7] A. Niswar, E. P. Ong, and Z. Huang. Face replacement in video from a single image. In *SIGGRAPH Asia 2012 Posters*, page 12, 2012. 2

[8] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003. 5

[9] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 386–391. IEEE, 2013. 2