Image Based Face Replacement in Video

Yu Liang (梁彧), *Bing-Yu Chen* (陳炳宇), *Yung-Yu Chuang* (莊永裕), *Ming Ouhyoung* (歐陽明)
National Taiwan University
{wildmb, robin, cyy, ming}@cmlab.csie.ntu.edu.tw

ABSTRACT

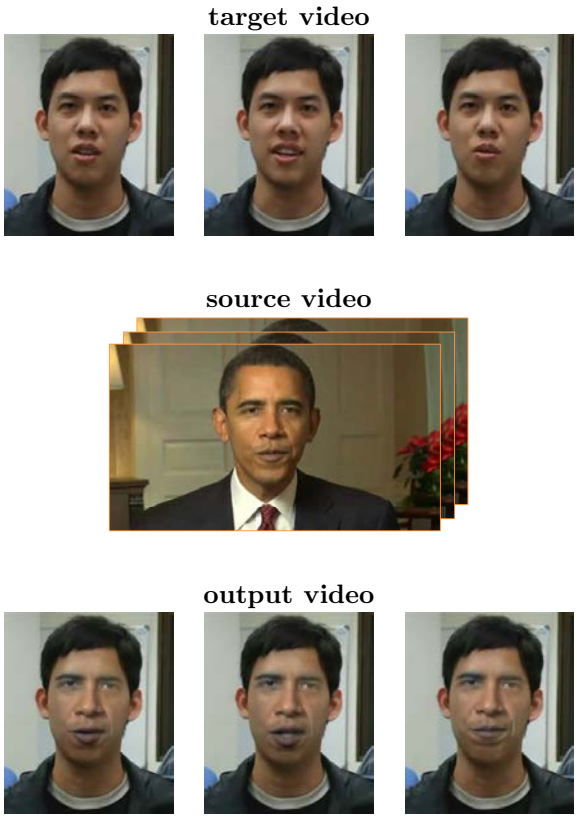
In this paper, we present a practical and complete system for plausible face replacement of videos. In our system, we take two videos as inputs, and aim to replace the target human subject's face with another human subject's in the other video. Our replacement algorithm has three main stages. First, given an input video, we detect all faces that are present, and align such detected faces. Second, we analyze facial expressions of each detected faces and select candidate face images from source video that are most similar to the target face in pose and expression. Third, we blend candidate replacements to target video. Our approach requires no 3D model, is fully automatic and generates plausible results.

Keywords: Face Replacement, Image-Based Rendering, Face Alignment

1. INTRODUCTION

Advances in digital photography have made it possible to capture large collections of high-resolution images and videos, and share them on the Internet. It leads to plenty of amazing applications and important problems. Face replacement in video automatically is not only a compelling application but a challenging problem. In fact, it is widely applied to movie visual effects. However, it will waste lots of time to make a high quality video manually. In particular, it will soon become intractable while videos grow rapidly.

In this paper, we present a practical and complete system for replacing faces in video based on a collection of images. These collections contain lots of photos of the same person such that we can either seek a proper pose or best matched expression to replace. To make life easier, the collection of source images can be attained from a recorded video. Once we build the library for replacement, we can apply to whatever target video for the personalized replacement application. The goal of our system is to automatically replace target character's face with source character's face given a target video and a source video.



Given a target video and a source video, our goal is to replace faces in the target video with the most similar candidate face selected from the source video.

We call the video to be replaced the target video, and define the video for replacement as the source

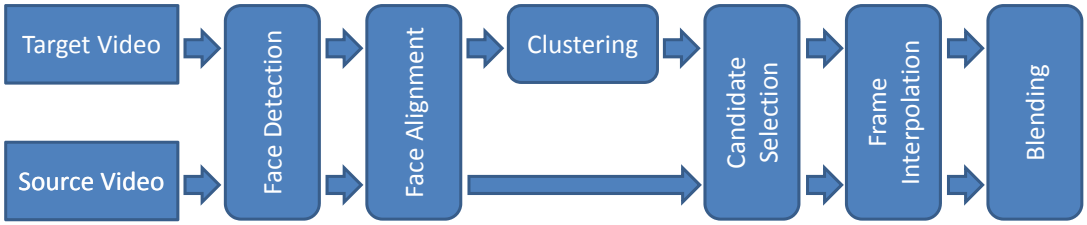


Figure 1: The flowchart of our system. When our system is supplied with a target video and a source video, it performs face detection to extract the face, and estimates the positions of facial features. Next the system tries to select the best candidate face from the library for replacement. Finally, we blend the best candidate face into the target video seamlessly. The final output is then produced.

video. The basic steps of our replacement approaches are shown in Figure 1. To complete the task, we divide our system into several parts.

1. Face detection: A face detection algorithm should locate the positions of the character human precisely.

2. Face alignment: However, it is not enough for knowing where the face appear in a video. To replace the face, we have to detect not only the face, but also the outlines of the facial profile and structures. Thus, a face alignment algorithm plays a critical role in our system.

3. Clustering: We would split the target video into several partitions rather than select the best candidate for each frame independently in the target video. The temporal coherency should not be ignored, thus we cannot just seek the best candidate face for each frame in the target video.

4. Candidate selection: After clustering, we must tell which face the best matched is for each cluster. The best candidate face must have the most similar pose and expression.

5. Frame interpolation: Once we selected the best candidate face for each cluster, what we all to do is synthesize in-between frames. A smooth transition between candidate faces is crucial and meanwhile we have to mimic the target character's expressions to produce the desired sequence.

6. Blending: The eventual result is ended by pasting up the synthesized sequence onto the target character's face. A visually convincible output video is done by blending the synthesized sequence and the target video together seamlessly.

2. RELATED WORK

2.1. Face Replacement

The replacement of faces in images has received rare attention. The work of Bitouk *et al.* [4] replaces the target face with the best candidate selected from the face library. They built the system for de-identification automatically. However, they just replace faces in single image. Agarwala *et al.* [1] makes use of graph-cut optimization to choose good seams within images and fusions the combined image in gradient domain. By combining multiple images, it may create a final composite with attractive looking.

The 3D based approach to face replacement also received no attention. Blanz *et al.* addressed several work about manipulating 3D model to replace faces. In [7], given a single image of a face, they can estimate its 3D shape, its orientation in the space, and illumination conditions in the scene. Thus the reconstructed face extracted from 2D image can be manipulated in 3D. In [6], they estimated 3D shape and texture along with all relevant scene parameters. After estimating, they can transfer the reconstructed face to the target image. Furthermore, they addressed a framework [5] to reanimating faces in image or video based on a dataset of 3D scan of different facial expressions and mouth shapes. With 3D shape reconstruction, they applied these expressions to any novel face in an image.

As we know, face replacement in video is an interesting topic but rare work is proposed especially based on images. Thus, our contribution is in that

we proposing an image based approach to replacing faces in video sequence without any information about 3D shapes.

2.2. Face Alignment

There is a wealth of research in computer vision on face alignment. The Active Shape Model (ASM) [9] is one of the early approaches that tries to fit human faces with a deformable face model from a training set. The Active Appearance Model (AAM) [8] is a popular extension to ASM. In AAM, the appearance is modeled by PCA on the mean shape coordinates. The shape parameters are searched by the residual of textures. After ASM and AAM, many variants have been proposed, for example, Bayesian Tangent Shape Model (BTSM) [27] is proposed to make the parameter estimation process more accurate and robust by using an EM based algorithm. Boosted Appearance Model (BAM) [17] trained a conventional point distribution model and a boosting-based classifier. Compared to AAM-based approach, BAM improves the robustness, accuracy and efficiency of face alignment by a large margin, especially for unseen data. Component-based Search [16] proposed a component-based discriminative approach without requiring initialization. The discriminative search is extremely effective and able to find very good alignment results only in a few search iterations.

2.3. Blending

Poisson image editing [21] has been widely used for seamlessly image composition. It blends two images by solving a Poisson equation with a guidance field. Image stitching [15] in the gradient domain has also been proposed to alleviate the boundary artifacts. Interactive digital photomantage [1] fusions a set of roughly aligned images into a single image.

The effectiveness of Poisson image editing depends heavily on how the user draws the boundary to blend. To find the appropriate boundary for blending two images in the gradient domain, a lot of work have been proposed. Graph cuts [14] is used for finding the best seams that minimizes the image discrepancy across boundaries. Jia *et al.* [13] designed a shortest close-path algorithm to search for the location of the boundary.

3. FACE ALIGNMENT

3.1. BTSM

Zhou *et al.* [27] proposed a Bayesian Tangent Shape Model (BTSM) formulated in Bayesian framework. The probabilistic formulation of shape registration problem contains two models: one denotes the prior shape distribution in tangent shape space and the other is a likelihood model in image shape space. Based on these two models, the posterior distribution of model parameters can be derived.

In a 2D image, a shape is described by N landmark points. We can represent it by a $2N$-dimensional vector $\mathbf{s} = (x_1, y_1, \ldots, x_N, y_N)^T$. The geometry information of $\mathbf{s}$ can be decoupled into two parts: a canonical shape $\mathbf{x}$ in the tangent shape space [27] and a similarity transformation T. Hence, the face shape $\mathbf{s}$ can be expressed as:

$$\mathbf{s} = T(\mathbf{x}; \boldsymbol{\gamma}), \tag{1}$$

where $\boldsymbol{\gamma}$ is a global similarity transformation parameter. Our goal is to find the tangent shape $\mathbf{x}$ and pose parameter $\boldsymbol{\gamma}$.

3.2. Prior Model and Likelihood Model in BTSM

After aligning all training samples to a common coordinate, we apply a probabilistic PCA [25] to model tangent shape variations. The model can be written as

$$\mathbf{x} = \bar{\mathbf{x}} + \boldsymbol{\Phi}_r \mathbf{b} + \boldsymbol{\Phi}\boldsymbol{\epsilon}, \tag{2}$$

where $\bar{\mathbf{x}}$ is the mean shape, $\boldsymbol{\Phi}$ is a $2N \times (2N-4)$ shape basis matrix, $\boldsymbol{\Phi}_r$ is a $2N \times r$ shape basis matrix that consists of the first r columns of $\boldsymbol{\Phi}$, $\mathbf{b}$ is a shape deformation parameters, and $\boldsymbol{\epsilon}$ denotes an isotropic noise vector. By adding noise vector $\boldsymbol{\epsilon}$, we can incorporate PCA into the Bayesian framework. After local texture matching step in BTSM, a $2N$-dimensional vector $\mathbf{y}$ called observed shape is estimated. The discrepancy between the observed shape $\mathbf{y}$ and tangent shape $\mathbf{x}$ is modeled as a Gaussian distribution.

$$\mathbf{y} = T(\mathbf{x}; \boldsymbol{\gamma}) + \boldsymbol{\eta} = s\mathbf{U}_\theta \mathbf{x} + \mathbf{t} + \boldsymbol{\eta}, \tag{3}$$

where s is the scale, $\mathbf{U}_\theta = \mathbf{I}_N \otimes \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ is the rotation matrix, $\mathbf{t} = \mathbf{1}_N \otimes \begin{bmatrix} t_x \\ t_y \end{bmatrix}$ is the translation matrix ($\otimes$ denotes Kronecker product [1]), and

[1] $\mathbf{A}_{m\times n} \otimes \mathbf{B}_{p\times q} = (a_{ij}\mathbf{B})_{ij} : mp \times nq$

$\boldsymbol{\eta}$ is an isotropic observation noise vector, which is distributed as a Gaussian model $N(\mathbf{0}, \rho^2\mathbf{I}_{2N})$.

3.3. Expectation Step and Maximization Step

The conditional probability of the tangent shape $\mathbf{x}$ given the observed shape $\mathbf{y}$ and model parameters is

$$p(\mathbf{x}|\mathbf{y}, \mathbf{b}, \boldsymbol{\gamma}) \propto \exp\big[-\frac{1}{2}(\sigma^{-2}\|\mathbf{x} - \bar{\mathbf{x}} - \Phi_r\mathbf{b}\|^2 + s^2\rho^{-2}\|\mathbf{x} - T^{-1}(\mathbf{y}, \boldsymbol{\gamma})\|^2)\big].$$

After some simple substitutions, the conditional expectation of $\mathbf{x}$ and $\|\mathbf{x}\|^2$ with respect to $p(\mathbf{x}|\mathbf{y}, \mathbf{b}, \boldsymbol{\gamma})$ is

$$\langle\mathbf{x}\rangle = \bar{\mathbf{x}} + (1-p)\boldsymbol{\Phi}_r\mathbf{b} + p\boldsymbol{\Phi}\boldsymbol{\Phi}^T T^{-1}(\mathbf{y}; \boldsymbol{\gamma}), \quad (4)$$

$$\langle\|\mathbf{x}\|^2\rangle = \|\langle\mathbf{x}\rangle\|^2 + (2N-4)\delta^2, \quad (5)$$

where $p = \frac{\sigma^2}{\sigma^2 + s^{-2}\rho^2}$ and $\delta^2 = (\sigma^{-2} + s^2\rho^{-2})^{-1}$.
The M step maximizes the posterior to obtain model parameters.

$$p(\mathbf{b}, \boldsymbol{\gamma}|\mathbf{y}, \mathbf{x}) \propto \exp\big[-\frac{1}{2}(\mathbf{b}^T\boldsymbol{\Lambda}^{-1}\mathbf{b} + \sigma^{-2}\|\mathbf{x} - \bar{\mathbf{x}} - \boldsymbol{\Phi}_r\mathbf{b}\|^2 + s^2\rho^{-2}\|\mathbf{x} - T^{-1}(\mathbf{y}; \boldsymbol{\gamma})\|^2)\big].$$

By computing the derivative, we have

$$\mathbf{b} = \boldsymbol{\Lambda}(\boldsymbol{\Lambda} + \sigma^2\mathbf{I})^{-1}\boldsymbol{\Phi}_r^T\langle\mathbf{x}\rangle,$$

$$\boldsymbol{\gamma} = \big(\frac{\mathbf{y}^T\langle\mathbf{x}\rangle}{\langle\|\mathbf{x}\|^2\rangle}, \frac{\mathbf{y}^T\langle\mathbf{x}\rangle^\perp}{\langle\|\mathbf{x}\|^2\rangle}, \frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_{x,i}, \frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_{y,i},\big)$$

$s = \sqrt{\gamma_1^2 + \gamma_2^2}$, $\theta = \tan^{-1}(\gamma_1/\gamma_2)$, $\mathbf{t} = [\gamma_3, \gamma_4]^T$, where $\langle\mathbf{x}\rangle^\perp$ is obtained by rotating planar shape $\langle\mathbf{x}\rangle$ by 90° [2].

3.4. Adding Virtual Shapes

For the above face alignment process, we used AR [19] and FERET [24] [23] sets for training, which contain frontal image only. However, a character subject would present a variety of poses and expressions in videos. It is intractable for face alignment algorithm trained by frontal faces only to locate landmark points precisely on the outlines of non-frontal-faces. Hence, we add some virtually slight rotation faces to build the training model [20].

[2] $\langle\mathbf{x}\rangle = (x_1, x_2, \ldots, x_{2N})^T \Rightarrow \langle\mathbf{x}\rangle^\perp = (-x_2, x_1, \ldots, -x_{2N}, x_{2N-1})^T$

For each training face, we stretch and contract each side of the face by multiplying the x position (relative to the face center) of each landmark point by $1 + \epsilon$, where ϵ is set to ± 0.1. This is roughly equivalent to rotating the face slightly.

3.5. Face Alignment in Video

Our goal is to localize facial outlines in videos. However, it is still an open problem for applying face alignment algorithm to videos. Not only the landmark points should be localized precisely, but also the temporal continuity must be stable.
It is well-known that bilateral filter [26] can smooth out the noises but preserve edges from blurring. After aligning the landmarks frame-by-frame, we want the trajectories of landmark points as smooth as possible, however the landmark points should respond quickly to fast movement. Thus, bilateral filter is a plausible noise removal tool. For each landmark point i, let $T_i = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_t)$ be a vector of positions in a video sequence with t frames. We can treat the vector as a signal and perform bilateral filtering directly on the signal. In bilateral filtering, the spatial variance σ_s controls the spatial neighborhood, and the temporal variance σ_t influences the smoothness. The spatial distance plays an edge-stopping role.

4. CLUSTERING

After aligning landmark points to the outlines of target face, the next step is to replace the target character's face with the most appropriate face from the source video based on the alignment result. However, it will introduce strong artifacts if we replace the face frame by frame independently without considering the temporal behavior of target character's face. Thus, we have to decompose the target video into groups in terms of pose, expression, ..., etc.

4.1. Hierarchical Clustering

To segment the target video, we make use of hierarchical clustering to build a hierarchy of clusters. In our system, we build the hierarchical clustering in a bottom up manner.
A cluster $\mathbf{c} = \{\mathcal{I}, \mathcal{X}, \bar{\mathbf{x}}\}$ is consist of n continuous target video frames $\mathcal{I} = \{I_1, I_2, \ldots I_n\}$, the corresponding shapes $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\}$ and a

mean shape of $\bar{\mathbf{x}}$, where $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$. Initially, each frame is independent, hence we have T clusters $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_T\}$ totally. For each cluster $\mathbf{c}_i$, we compute the distance d_i between cluster $\mathbf{c}_i$ and cluster $\mathbf{c}_{i-1}$. The distance d_i is computed as the *Mahalanobis distance* between the mean shape of cluster $\mathbf{c}_i$ and cluster $\mathbf{c}_{i-1}$. Then, we pick the shortest distance d_j from $\mathcal{D} = \{d_2, d_3, \ldots, d_{|\mathcal{C}|}\}$ to merge the corresponding clusters $\mathbf{c}_j$ and $\mathbf{c}_{j-1}$. After merging cluster $\mathbf{c}_j$ and cluster $\mathbf{c}_{j-1}$, we have $T-1$ clusters in total currently. The merged cluster $\mathbf{c}^*$ is updated by $\mathbf{c}^* = \{\mathcal{I}^*, \mathcal{X}^*, \bar{\mathbf{x}}^*\}$, where $\mathcal{I}^*$ and $\mathcal{X}^*$ are the union set of the images and tangent shapes in cluster $\mathbf{c}_j$ and cluster $\mathbf{c}_{j-1}$ respectively. $\bar{\mathbf{x}}^*$ is the mean shape of $\mathcal{X}^*$. Since we only merge the adjacent clusters, the image set and shape set of merged cluster are guaranteed to be continuous. The process recursively proceeds until the cardinality of $\mathcal{C}$ is lower than the predefined threshold of cluster number.

4.2. Shape Based Selection

Suppose that we have split the target video into $|\mathcal{C}|$ partitions. Each partition is denoted by a cluster node $\mathbf{c}$ which is the output of the hierarchical clustering. Our goal is to pick the best candidate face from the source video for each cluster $\mathbf{c}$ by comparing mean shape vector $\bar{\mathbf{x}}$.

Suppose the neutral face vector of the target video and source video are $\mathbf{N}_T$ and $\mathbf{N}_S$ respectively. For tangent shape $\mathbf{x}_i$ of each frame, we can obtain the difference vector between $\mathbf{x}_i$ and neutral face vector. The difference vector represents the variation due to either poses or expressions. This is what we really regard for. After we have computed the difference vectors, we can pick the most similar difference vector in the source video for current frame.

4.3. Interpolation Between Clusters

After segmenting the target video, the discontinuity between clusters is still an issue. To concatenate the clusters, we have to interpolate the frames between clusters. Our method is based on the intuitive notion proposed by Beier and Neely [3]. A smooth image interpolation can be done by warping two images to the same positions and then dissolve the image textures together. After producing the smooth image sequence, the final step is to warp the facial features to appropriate positions to match the expressions exactly [18]. We warp the smoothly interpolated image sequence according to the difference vectors. The warped images are about to be blended into the target video.

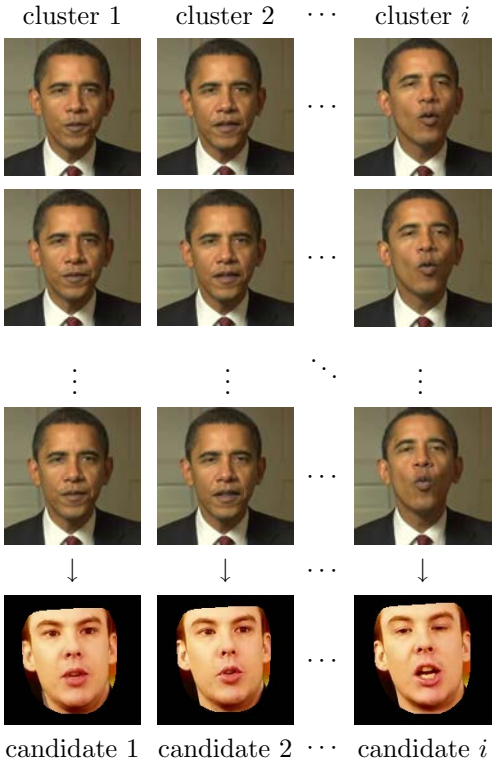


Figure 2: The selection process. Each column represents one cluster, and the last row is the best candidate selected from the source video.

5. BLENDING

5.1. Face Concealment

To replace the target face seamlessly, a proper region for replacement is critical. Occasionally, the size of the target character's face and the source character's do not coincide with each other due to scale. The blending process relies heavily on cloning boundary. The eyebrows or wrinkles would disturb the boundary and the seamless cloning will not make a satisfied result. Thus, we conceal the facial features of the target face before cloning the source face. See figure 3 for illustrations.

Face concealment can be easily done by clearing the gradient field to zero in the facial region then reconstructed by integrating the modified gradi-

Figure 3: The top left figure shows that some usdesirable features cannot be covered by the source patch. The top right figure is the result after concealing face. The bottom left figure is shown for comparing to the top left figure. The bottom right figure is done by blending source patch after cocealing face.

ent. By assigning null gradient to the pixels within the facial region, we can obtain a smooth and de-identification face.

5.2. Boundary Selection

While we paste up the source face on the target face, the pasting region may exceed the boundary of the target face. The irrelevant background would be blended together such that the smudge effect will occur. In particular, hair around eyebrows is typically blended with the eyebrows of source face in our system implementation. When the target character rotates in an out-of-plane manner, the eyebrow would approach the hair very closely. Hence, we have to deal with such case very carefully.

Our goal is to locate the boundary outline which guards the region to prevent the source image from blending over it. [2] inspires us to work out an approach to searching the proper seam to guard the region. They want to remove unnoticeable pixels that blend with the surroundings. On the contrary, we want to seek a seam that runs parallel to the edge and will not cross it. We make use of the modified energy function $e^*_{HoG}(\mathbf{I})$ they proposed to complete the task. The $e^*_{HoG}(\mathbf{I})$ is defined as follows:

$$e^*_{HoG}(\mathbf{I}) = \frac{\max\left(HoG(\mathbf{I}(x,y))\right)}{|\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|}, \qquad (6)$$

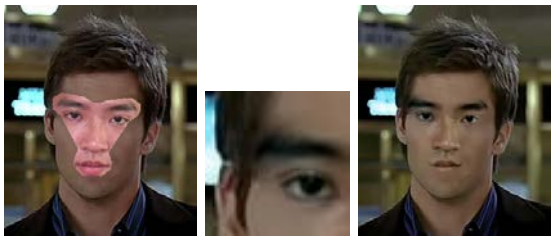


Figure 4: We have to avoid blending with irrelevant background. Otherwise, it would introduce undesired smudge effect. The left image shows that the source patch is pasted over the face region. Note the region around the left eyebrow. The right image shows the artifacts due to incorrect boundary.

Figure 5: In the first row, the left figure is the result of just pasting the source patch on the face, and the right figure shows the clipping boundary in red color. In the second row, the left figure demonstrates the source patch after clipping, and the right figure is the result after blending.

where $HoG(\mathbf{I}(x,y))$ is taken to be the histogram of oriented gradients at every pixel [10].

We let the seam find the optimal path from the landmark endpoints of face profile as origins, and let the destination locate on the line formed by two points above the eyebrows that derived from adding a half of eye width to the eyebrow. The profile from aligned landmark points and found optimal seams form a closed cycle. As we expected, the path go along the edge and we get a closed boundary.

5.3. Mean Value Coordinate

To seamlessly clone the source patch into a target image, the operation is typically carried out by solving a large linear system. Instead of solving

Figure 6: We replace the actor Chris Pine's face with President Obama's. The first row is the source video. The middle row is the target video. The third row is the output video. The target video is extracted from the movie "Just My Luck." [22]

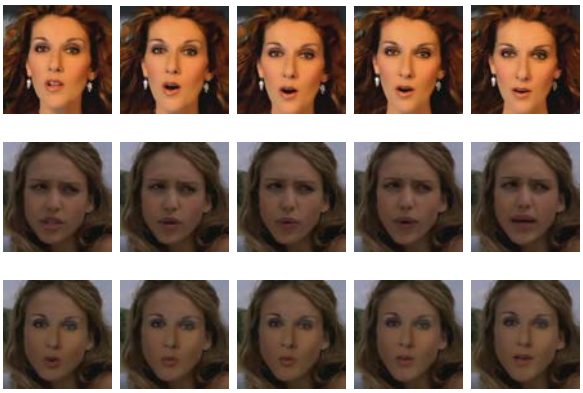


Figure 7: We replace the actress Jessica Alba's face with singer Celine Dion's. The first row is the source video. The middle row is the target video. The third row is the output video. The target video is extracted from the movie "Good Luck Chuck." [12]

the large linear system of Poisson equation, we use the method proposed by Farbman *et al.* [11]. The MVC cloning algorithm is a coordinate-based approach. It is a generalized Barycentric coordinate. The value of the interpolant at each interior pixel is given by a weighted combination of values along the boundary. The use of coordinates is advantageous in terms of speed, ease of implementation, small memory footprint.

6. EXPERIMENT AND CONCLUSION

This paper presents an overall system for face replacement in video. Using the face alignment algorithm, we can find out the precise positions for replacement. Moreover, a clustering mechanism lets us be able to select a best candidate for each cluster and generate matched expressions. Finally, the blending technique is required to produce the plausible results. We have demonstrated the advantages that images can offer for generating a video sequence. Figure 6 and Figure 7 show the results of our system.

REFERENCES

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 294–302, New York, NY, USA, 2004. ACM.

[2] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. volume 26, 2007.

[3] T. Beier and S. Neely. Feature-based image metamorphosis. *SIGGRAPH Comput. Graph.*, 26(2):35–42, 1992.

[4] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar. Face swapping: automatically replacing faces in photographs. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–8, New York, NY, USA, 2008. ACM.

[5] V. Blanz, C. Basso, T. Vetter, and T. Poggio. Reanimating faces in images and video. In P. Brunet and D. W. Fellner, editors, *EUROGRAPHICS 2003*, volume 22 of *Computer Graphics Forum*, pages 641–650, Granada, Spain, 2003. The Eurographics Association, Blackwell.

[6] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel. Exchanging faces in images. *Computer Graphics Forum*, 23(3):669–676, 2004.

[7] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99:*

Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[8] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[9] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, 1995.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[11] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, and D. Lischinski. Coordinates for instant image cloning. *SIGGRAPH '09: ACM SIGGRAPH 2009 Papers*, 2009.

[12] M. Helfrich. Good luck chuck, 2007.

[13] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum. Drag-and-drop pasting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 631–637, New York, NY, USA, 2006. ACM.

[14] V. Kwatra, A. Sch绎dl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, 22(3):277–286, July 2003.

[15] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pages 377–389. Springer-Verlag, 2003.

[16] L. Liang, R. Xiao, F. Wen, and J. Sun. Face alignment via component-based discriminative search. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 72–85, Berlin, Heidelberg, 2008. Springer-Verlag.

[17] X. Liu. Generic face alignment using boosted appearance model. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 1079–1088, 2007.

[18] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 271–276, New York, NY, USA, 2001. ACM.

[19] A. Martinez and R. Benavente. The ar face database, June 1998.

[20] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. *ECCV*, 2008. `http://www.milbo.users.sonic.net/stasm`.

[21] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH'03)*, 22(3):313–318, 2003.

[22] D. Petrie. Just my luck, 2006.

[23] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The feret evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1090–1104, 2000.

[24] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing J*, 16(5):295–306, 1998.

[25] M. E. Tipping, C. C. Nh, and C. M. Bishop. Mixtures of probabilistic principal component analysers, 1998.

[26] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 839, Washington, DC, USA, 1998. IEEE Computer Society.

[27] Y. Zhou, L. Gu, and H.-J. Zhang. Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.