EyeOpener: Editing Eyes in the Wild

ZHIXIN SHU
Stony Brook University
ELI SHECHTMAN
Adobe Research
DIMITRIS SAMARAS
Stony Brook University
and
SUNIL HADAP
Adobe Research

Closed eyes and look-aways can ruin precious moments captured in photographs. In this article, we present a new framework for automatically editing eyes in photographs. We leverage a user's personal photo collection to find a "good" set of reference eyes and transfer them onto a target image. Our example-based editing approach is robust and effective for realistic image editing. A fully automatic pipeline for realistic eye editing is challenging due to the unconstrained conditions under which the face appears in a typical photo collection. We use crowd-sourced human evaluations to understand the aspects of the target-reference image pair that will produce the most realistic results. We subsequently train a model that automatically selects the top-ranked reference candidate(s) by narrowing the gap in terms of pose, local contrast, lighting conditions, and even expressions. Finally, we develop a comprehensive pipeline of three-dimensional face estimation, image warping, relighting, image harmonization, automatic segmentation, and image compositing in order to achieve highly believable results. We evaluate the performance of our method via quantitative and crowd-sourced experiments.

Categories and Subject Descriptors: I.3.8 [**Computer Graphics**]: Image Manipulation, Applications

Additional Key Words and Phrases: Face editing, eye editing, gaze editing, image compositing, computational aesthetics

The research was partially supported by a gift from Adobe, by NSF Grant No. IIS-1161876, and by the SUBSAMPLE project of the DIGITEO Institute France.

Authors' addresses: Z. Shu and D. Samaras, Computer Science Building, Stony Brook, NY 11794-4433; emails: {zhshu, samaras}@cs.stonybrook.edu; E. Shechtman, 801 N 34th St, Seattle, WA 98103; email: elishe@adobe.com; S. Hadap, 345 Park Ave, San Jose, CA 95110; email: hadap@adobe.com.

DOI: http://dx.doi.org/10.1145/2926713

ACM Reference Format:
Zhixin Shu, Eli Shechtman, Dimitris Samaras, and Sunil Hadap. 2016. EyeOpener: Editing eyes in the wild. ACM Trans. Graph. 36, 1, Article 1 (September 2016), 13 pages.
DOI: http://dx.doi.org/10.1145/2926713

1. INTRODUCTION

Faces are of great interest in photography and photo-sharing. According to a recent study [Bakhshi et al. 2014], Instagram photos with faces are 38% more likely to get "likes" than photos without faces. However, shooting a good portrait is challenging for several reasons: The subject may become nervous when facing the camera, causing unnatural facial expressions; the flash light could cause the subject's eyes to close; the camera might capture the image before the subject depicts the perfect expression, and there is obviously also the physiological aspect. The corneal reflex, or the blink reflex, is an *involuntary* motion of the eyelids elicited by stimulation of the cornea, such as by touching or by bright light. Keeping the eyes open is therefore hard for subjects to control, sometimes resulting in less-attractive expressions [Zhu et al. 2014]. In a recent study, Zhu et al. [2014] tried to predict the attractiveness of facial expression using crowd-sourced knowledge. They designed an app to help users train for their most attractive expressions. A more convenient way to improve facial appearances in photos is through image editing. Image editing is especially useful for group photos, because it is difficult to ensure that everyone in the photo has the "perfect" look when the shutter is released [Agarwala et al. 2004]. However, general image editing tools, such as Adobe Photoshop, require significant expertise and manual work. In this article, we introduce a way to post-process the appearance of the eyes to enhance facial expressions in photos. We describe a fully automatic method for eye editing that does not require any interaction with the subject. If desired, however, then the method also provides the user with the option to pick a few reference images him- or herself (Figure 1).

Example-based approaches have already been successfully used in face editing [Bazin et al. 2009; Bitouk et al. 2008; Joshi et al. 2010; Yang et al. 2011; Dale et al. 2011; Garrido et al. 2014]. Our method applies the eye region from selected suitable image(s), preferably images of the same subject, in order to replace the eye region in the original image. Given that a personal photo album typically has a large collection of photographs, it is easy to obtain multiple "example" eye sets. The variety of examples to choose from enables our method to provide very realistic eye appearances. The

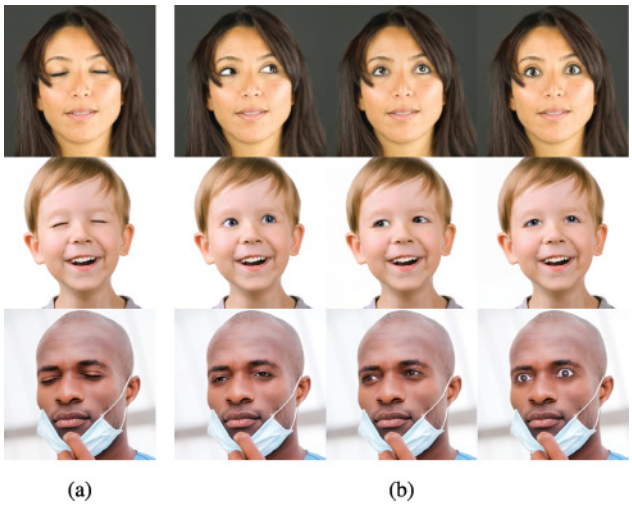


Fig. 1. Given a shut-eyes input face (a), our example-based method generates fully automatically a number of open-eyes images (b), based on appropriate reference images of the same face (not shown). Image credits: Adobe Stock/bruno135_406 (first row), Adobe Stock/Tanya Rusanova (second row), Adobe Stock/gstockstudio (third row).

method is even able to account for eye expression, gaze direction, make up, and more.

One limitation of an example-based approach is that the quality of the result largely depends on how compatible the source and the target images are [Hays and Efros 2007; Bitouk et al. 2008; Yang et al. 2011; Laffont et al. 2014; Shih et al. 2014]. There are many variables in the context of face editing, such as identity, pose, illumination condition, and facial expression. The number of variables increases the likelihood of incompatibility between the example image(s) and the target image. We can account for some of these incompatibilities through correction, but other factors are more difficult to correct. In this article, we show how to correct the factors related to geometric and photometric incompatibilities, as well as how to avoid using highly incompatible examples with factors that may be hard to fix. We approach the problem by adapting the example to match the target image. We do this by warping the three-dimensional (3D) pose of the example and by adjusting the color and intensity of the example using a novel approach. We then seamlessly composite the example eyes onto the target eyes using a new method that we call Background-Foreground Guided Poisson (BFGP) compositing. It is robust to the mismatched boundaries and is based on a combination of Poisson blending and alpha blending via a Guided Filter [He et al. 2013]. Although Expression Flow [Yang et al. 2011] uses a similar approach for face warping, it requires detailed user interaction to define the compositing boundaries in order to obtain satisfactory results. In contrast, our system automatically alleviates the boundary mismatch artifacts through local color transfer and robust compositing.

In order to provide a fully automatic function, we needed to guarantee the quality of the resulting image for each and every image that was edited. This proved to be quite challenging. We approached the task by learning the compatibility between the input (target) image and the example images using crowd-sourced human annotations. For a few subjects, we collected two "in-the-wild" datasets of face images, one with the eyes shut and one with eyes open. We then randomly selected 8,000 shut-eyes/open-eyes pairs (same subject) and used our system to open the eyes. We subsequently collected viewer ratings for these results on Amazon Mechanical Turk. We learned a model from these human labels to predict the quality of the results for novel input/example image pairs, even before generating the outputs. This model enabled us to efficiently find the references that are most likely to produce realistic results. We used 30 to 50 reference images per person, a reasonable number for a typical photo album. The average precision of our model w.r.t. human annotations is 0.70 on our test, while chance is 0.46.

Our contributions are as follows:

(1) A fully automatic pipeline for example-based eye editing.
(2) A new compositing method (BFGP) combined with local color adjustment that ensures excellent results, outperforming traditional techniques.
(3) A learning approach based on crowd-sourced human annotations for predicting the compatibility of an input/example image pair for eyes replacement.

In addition to opening eyes, we show that our example-based framework can also be used to change gaze and to improve expression attractiveness [Zhu et al. 2014]. Furthermore, our initial results of eye appearance transfer across individuals demonstrate the promise of applying our method to creative face editing.

2. RELATED WORK

This work is related to previous work involving face editing, local color transfer, image compositing, and crowd-sourcing human feedback.

Face Image Editing. For decades, faces have been prime targets for image enhancement and editing. Because of similarities across faces, data-driven methods are widely applied: Blanz and Vetter [Blanz and Vetter 1999] proposed a 3D morphable model to synthesize, from a single image, faces with novel pose, expression, or illumination; Nguyen et al. [2008] applied a subspace method for beard and eyeglasses removal, Leyvand et al. [2008] developed a system to warp a face to improve its attractiveness, and Kemelmacher et al. [2014] leveraged a dataset of people across several years to synthesize faces at different ages. A closer line of research is based on an example-based approach, where certain properties of the examples are transferred to the target input. Bitouk et al. [2008] developed a system that automatically transfers an entire face from one image to another. This approach was later extended to full face performance replacement in video [Dale et al. 2011; Garrido et al. 2014]. Guo et al. [2009] introduced a method to transfer facial make up; Joshi et al. [2010] sought to improve face image quality using personal examples; Shih et al. [2014] proposed to transfer style from professional photographs to casual ones; and Yang et al. [2011] developed an interactive system for example-based facial expression editing. Differing from all previous work, our system focuses on the local appearance of eyes, taking into account both color and texture. Furthermore, it is fully automatic. We combine the benefit of using real examples, with a learned model that predicts result quality based on human feedback to find good examples.

Local Color Transfer. As shown in previous work, when faces need to be composed from different sources [Bitouk et al. 2008], the inconsistency of color and intensity distribution, which, in general, is caused by different illumination conditions, is a major problem jeopardizing the look of face composites. We adopt local color transfer to bridge the gap of color and intensity distributions between the input target and a given example. To deal with inconsistent illuminations, Bitouk et al. [2008] relighted the example face with the illumination of the target image. Common face relighting approaches include the work done by Wen et al. [2003], Zhang et al. [2006],

and Wang et al. [2009] that use spherical harmonics to estimate illumination and apply ratio images [Liu et al. 2001; Wang et al. 2007, 2009] to transfer illumination; Chen et al. [2011] estimated the illumination layer of a face using edge-preserving filters. In recent work, Shih et al. [2014] achieved a certain amount of illumination transfer effects via local color transfer, generating robust results in the process. Inspired by this, we also addressed the color and intensity inconsistency problem with a local color transfer approach.

Image Compositing. In copy-paste editing, the compositing method is crucial for obtaining high-quality results. Commonly used techniques include alpha blending, multi-scale compositing, and gradient domain blending. Seamless cloning of content is usually powered by compositing in the gradient domain [Pérez et al. 2003; Agarwala 2007; Farbman et al. 2009; Bhat et al. 2010]. These methods tend to suffer from artifacts when the boundaries of the foreground and the background do not match well. Tao et al. [2013] proposed to "hide" the errors in textured areas to avoid noticeable bleeding artifacts. Sunkavalli et al. [2010] developed a multi-scale histogram matching approach, which allows textures to be naturally blended. In the context of eye editing, color artifacts are highly noticeable. We therefore developed a simple technique that combines the advantages of alpha blending and seamless cloning for more plausible results.

Inverse Rendering. Recent advances in inverse rendering techniques allow dynamic reconstruction of face geometry from videos [Garrido et al. 2013; Shi et al. 2014; Suwajanakorn et al. 2014] and enable interesting face manipulation applications [Garrido et al. 2013, 2015; Suwajanakorn et al. 2015]. Since we are only focusing on editing eyes, our method is more lightweight. It is based on image input and does not require very accurate geometry of the input faces.

Learning via Crowd-sourcing. In Shih et al. [2014] and Bitouk et al. [2008], examples are selected by empirical hand-crafted strategies. However, these strategies are not optimized according to human preferences. Inspired by recent advances in computer vision [Deng et al. 2009; Welinder et al. 2010; Parikh and Grauman 2011a] and computer graphics [Zhu et al. 2014; O'Donovan et al. 2014; Laffont et al. 2014] in which human knowledge is heavily taken into account, we seek to improve example selection using crowd-sourced feedback. Human labeled data have been used in many different tasks. For example, Parikh et al. [2011b] proposed to model relative attributes with crowd-sourced data and learning to rank; Kiapour et al. [2014] studied what clothing reveals about personal style via collecting human judgments of styles; Laffont et al. [2014] proposed a high-level image editing method to enable users to adjust the attributes of a scene by harvesting crowd-sourced annotations; and Zhu et al. [2014] developed a system that provides feedback for portrait attractiveness by learning from crowd-sourced evaluations.

3. AUTOMATIC EYE OPENING

In this section, we describe the overall eye-editing pipeline, illustrated in Figure 2. Given an image with undesired eyes (*target*), we attempt to replace them with the eyes from another input image/example (*reference*).

We first fit 3D face models to the *reference* and *target* faces (Section 3.1). We estimate the poses and expressions for the 3D fitting using the fiducial points given by a face detector [Saragih 2011]. We use a copy-blend approach for image synthesis. Local contrast and lighting is paramount for producing highly believable results, accurate alignment, and image harmonization in terms of perceived skin tones. Subsequently, we warp the *reference* face in 3D to match the *target* face (Section 3.2) and perform local color correction (Section 3.3). After automatic selection of the eye regions, we seamlessly blend the new eyes using a technique that is robust to unmatched local boundaries (Section 3.4).

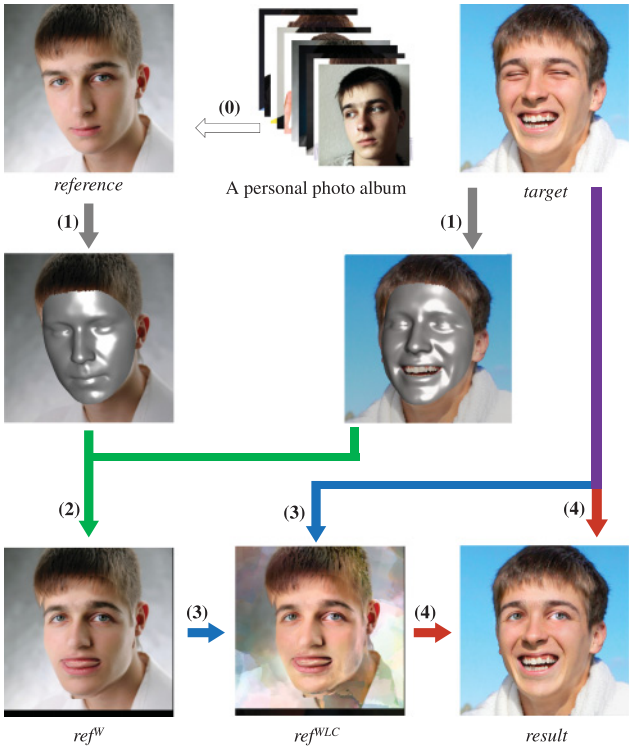


Fig. 2. Eye-editing overview. Given a *target* image in which eyes need to be edited, and a *reference* with the desired eye appearance, our system automatically transfers eye appearance by the following steps: (1) 3D face fitting (3.1); (2) pose correction (3.2); (3) local color adjustment (3.3), and (4): robust blending (3.4). In the latter part of our article, we also introduce a tool to help users with step (0): selecting appropriate *reference*s. Image credits: Adobe Stock/Sabphoto.

3.1 3D Face Fitting

Let the concatenation of n 3D points represent the 3D face: $S = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n)$. We represent the space of all 3D faces via a 3D morphable model [Blanz and Vetter 1999]. We conduct Principle Component Analysis on a 3D face shape dataset to obtain the eigenshapes (eigenvectors) of the face space denoted by $\mathbf{V}_{n\times m} = [V_1, \ldots, V_m]$ (we choose only the first m significant eigenvectors). A 3D face can be approximated by the linear combination of the face eigenshapes as follows:

$$S = \bar{S} + \sum_{i=1}^{m} \beta_i V_i = \mathbf{V} \cdot B, \tag{1}$$

where $\bar{S}$ denotes the average shape in the 3D face dataset. In an image, a 2D face shape is assumed to be generated by projecting a 3D face shape to an image plane: $S^{2D} = \mathbf{R} \cdot S^{3D}$. Assuming a weak perspective projection, we jointly recover the projection matrix $\mathbf{R}$ and the shape eigenvalues B, by minimizing the error between the projected 3D landmarks and 2D landmarks detected in the image:

$$E = \frac{1}{2}\|\mathbf{R} \cdot L^{3D} - L^{2D}\|^2, \tag{2}$$

(a) *target* (b) *reference* (c) ref^W (d) ref^{WLC}

Fig. 3. Local color adjustment using multidimensional histogram matching. We match the color distribution of the face from ref^W (warped and aligned *reference*) to the color distribution of the *target* face to get ref^{WLC}. Note that we only use the eye region from ref^{WLC} for compositing into the *target*. Image credits: Adobe Stock/Sabphoto.

where L^{3D} denotes the 3D positions of the landmarks in the face model and L^{2D} denotes the 2D positions of the landmarks detected in the image. With every iteration of the optimization, as the pose of the 3D face (w.r.t the camera) varies, the landmarks along the occluding contour of the 2D face correspond to different vertices of the projected 3D face. To handle this situation, we use the image fitting algorithm proposed by Yang et al. [2011] to optimize for $\mathbf{R}$ and B. Since we focus on the eyes, we impose higher weights (4 times the normal weights) on landmarks that are located on the eyebrows, eyes, and nose. This reduces the fitting error due to expression variations and yet robustly captures identity and pose variations.

3.2 Pose Correction

For image warping, we adapt the 3D point displacement pipeline of Yang et al. [2011]. As described in the previous section, fitting the 3D morphable face model to the *reference* and the *target* establishes explicit 3D vertex-to-vertex correspondences between the two images. By projecting the images onto their corresponding 3D face shapes that share the same topology, we establish a pixel-to-pixel 2D displacement field that robustly captures the non-linear effect of pose rotation. We use the 2D displacement field to warp the *reference* image and denote it ref^W. The ref^W has eyes roughly aligned to the undesired eyes in the *target*. Any small discrepancies in the shape matching at the individual pixel level are robustly handled by *dense optical-flow* estimation and correction, as described in Section 3.4.

3.3 Local Color Adjustment

After rough alignment of the eyes into the ref^W that match the *target*, the second step is to harmonize the ref^W image to match aspects of the *target* image in terms of overall skin-tones, lighting, local contrast, and so on. In this section, we introduce a novel approach of multi-dimensional histogram matching to achieve robust local color transfer:

Given two sets of N-dimensional data $X_{N\times M_1}$ and $Y_{N\times M_2}$, as well as their distribution in N-dimensional space $h(X)$ and $h(Y)$, respectively, we seek a mapping function $f_{N\to N}(\cdot)$:

$$Z_{N\times M_1} = f(X), \tag{3}$$

such that the distribution of Z matches the distribution of Y: $h(Z) = h(Y)$. This is the histogram matching problem in N-dimensional space. Pitié et al. [2005] proposed an iterative method to estimate a continuous transformation that maps one N-dimensional distribution to another. When $N = 1$, the histogram matching problem can be easily solved by a discrete lookup table. The algorithm in Pitié et al. [2005] is briefly outlined as follows: Letting $X^{(1)} = X$, in the ith iteration, the algorithm first applies a random rotation $R^{(i)}_{N\times N}$ to the data $X^{(i)}$ and Y to get $X^{(i)}_r = R^{(i)}X^{(i)}$ and $Y^{(i)}_r = R^{(i)}Y$. Then the marginals of $X^{(i)}_r$ are matched to the marginals of $Y^{(i)}_r$, respectively, using 1D histogram matching. We denote the matching result with $Z^{(i)}_r$. After that, the data are rotated back to the original space: $Z^{(i)} = (R^{(i)})^{-1}Z^{(i)}_r$ and $X^{(i+1)} = Z^{(i)}$. In theory, the algorithm converges when $i \to +\infty$. However, in practice, the Kullback–Leibler divergence between Z and Y drops quickly [Pitié et al. 2005].

Pitié et al. [2005] applied multi-dimensional histogram matching to color transfer when $N = 3$ (RGB color space). For local color transfer application, we adopt the idea of multi-dimensional distribution matching but with extra dimensions to enforce spatial locality during matching.

Image statistics like RGB histograms are global in nature and hence do not capture local effects such as illumination and shading variation on the face. In order to model local effects while color matching, we propose to rewrite the image representation from $I^{(3)}(x, y) = [r(x, y), g(x, y), b(x, y)]$ to

$$I^{(5)}(i) = [r_i, g_i, b_i, x_i, y_i], \tag{4}$$

where i is the pixel index. This representation explicitly encodes the locality of image pixels. Thus, we can build a five-dimensional histogram for an image that encodes both locality and semantic pixel information. We then carry out the multi-dimensional histogram matching from *target* to ref^W.

The result of an exact matching from ref^W to *target* will be *target* itself since the representation is unique. We remove all changes in image coordinates x and y (the third andfourth dimensions of $I^{(5)}$) to maintain the texture of ref^W. Since we are only interested in the region around the eyes, the matching functions are only computed from the face area in every iteration. In practice, we smooth the matching functions to avoid quantized color artifacts.

3.4 BFGP Compositing

After applying local color transfer to ref^W, we obtain a pre-processed example image ref^{WLC}. The remaining eye-editing step is to paste the eye region of ref^{WLC} to the *target*. Before the final compositing, we re-align the eye region ref^{WLC} to the *target* using dense optical flow [Brox and Malik 2011] around the eye region. This removes most of the discrepancies caused by the warping method introduced in Section 3.2. The discrepancies are caused by expression change or inaccurate pose estimation. However, we do not warp the image using the optical-flow. Instead, we compute the mean motion of the optical-flow field $[\bar{\Delta}x, \bar{\Delta}y]$ and apply it to ref^{WLC}. Subsequently, the boundary of the eye region is automatically defined by using the landmarks in the eye region and applying graph-cuts [Shi and Malik 2000; Kolmogorov and Zabin 2004] on the *target* and ref^{WLC}.

The idea of optimizing the region for gradient domain blending is similar to the work done by Agarwala et al. [2004] and Yang et al. [2011]. Following the work of Yang et al. [2011], we define the region on the image gradient domain. Specifically, high gradient regions around a region of interest (ROI) are encouraged to be part of the cut. In our application, the ROI is the eye region defined by eye landmarks. We refer the reader to Section 3.6 [Yang et al. 2011] for implementation details of the graph cut boundary optimization.

In previous copy-paste-based methods [Yang et al. 2011; Bitouk et al. 2008], the compositing is obtained in the image gradient domain using Poisson blending [Pérez et al. 2003]. Gradient domain compositing methods produce a seamless image composite only if the boundary conditions in the image pair are roughly matched. Real-life face photos are taken under unconstrained illumination conditions. Our local color transfer algorithm matches the lighting and local shading of *target* and *reference* at a relatively large scale.

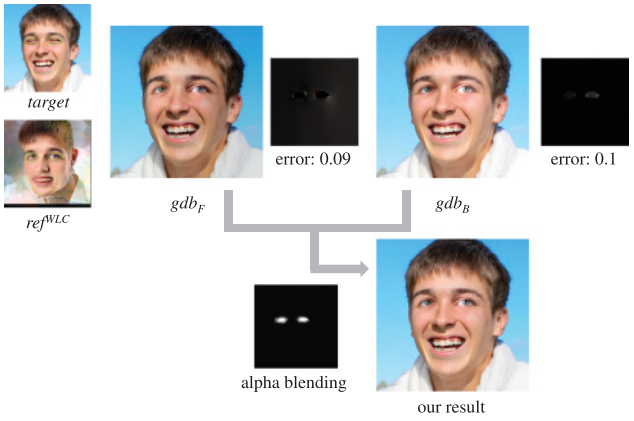


Fig. 4. BFGP compositing: combining gradient domain blending and alpha blending. We perform gradient domain blending twice, once on the *target* and once on the ref^{WLC}. gdb_F is the result of gradient domain blending so the foreground (i.e., eyes) is preserved, while in gdb_B the background (i.e., face) is preserved. For gdb_F, the error is the mean pixel value error of the background compared to *target*. For gdb_B, the error is the mean pixel value error of the foreground compared to ref^{WLC}. The blending boundary is optimized by graph-cuts. The Alpha matte is obtained by feathering the eyes mask to ref^{WLC} using the Guided Filter. Image credits: Adobe Stock/Sabphoto.

Thus, a boundary conditions mismatch may still exist due to small-scale local shadows and shading around the eyes. Under unmatched boundary conditions, visual artifacts arise in the form of color bleeding. Tao et al. [2013] proposed error-tolerant image compositing, in which the color bleeding error caused by mismatched boundary conditions is "hidden" in a highly textured area such that it will be less noticeable. However, since we observed our users to be very critical of errors in the edited eye results, we took extra care when blending eyes in the presence of unmatched boundary conditions.

In our system, we combine seamless gradient domain blending with alpha blending to avoid visual artifacts. Since the background of eye (skin region) is matched using local color transfer (Section 3.3), alpha blending will only work if a proper alpha matte can be defined for the eye foreground (eye region). Otherwise, the non-smooth transitions in the boundaries or texture details from the *target* (i.e., shut eyes in Figures 5 and 9) will cause artifacts. Our approach, named BFGP compositing, takes advantage of the seamless blending property of gradient domain blending in a selective manner by combining the best parts of multiple blended results, as detailed below.

We define M_F as the eye region (foreground) and M_B as the remaining face region (background). We first compute two image composites, gdb_F and gdb_B, by gradient domain blending. For gdb_F, we fix the foreground such that the mismatched boundary error will be propagated to the background; for gdb_B, we fix the background such that the mismatched boundary error will be propagated to the foreground. Our final composite will be an alpha blending of gdb_F and gdb_B that takes the background from gdb_F and the foreground from gdb_B (Figure 4). The alpha matte hereby is simply a smooth masking of the foreground region M_F. In practice, we obtain the alpha matte by the Guided Filtering [He et al. 2013] of M_F to the *reference*.

4. CROWD-SOURCING HUMAN EVALUATIONS

We evaluate the performance of our automatic eye editing system via Amazon Mechanical Turk (AMT) crowd-sourcing. We first

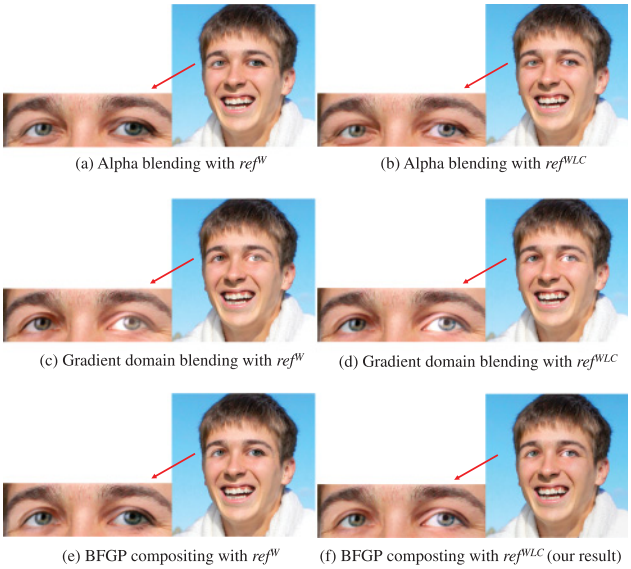


Fig. 5. Results of different compositing methods. (a) Alpha blending with ref^W, with artifacts of both incompatible color and unwanted shut-eyes detail. (b) Alpha blending with ref^{WLC}, with artifacts from shut-eyes detail. (c) Gradient domain blending with ref^W, resulting in poor contrast and eye color caused by mismatched boundaries. (d) Gradient domain blending with ref^{WLC}, with less severe but still noticeable color and contrast artifacts. (e) BFGP compositing with ref^W, with somewhat incompatible eye colors. (f) BFGP compositing with ref^{WLC} (our result) generates the most natural-looking result.

generate a set of edited results (denoted by $\{\mathcal{O}\}$) by running our algorithm on a collection of "shut-eyes" images (which we call the *target set*, denoted by $\{\mathcal{T}\}$), together with a collection of "open-eyes" example images (which we call the *reference set*, denoted by $\{\mathcal{R}\}$). In this section, we describe the collection of $\{\mathcal{T}\}$, $\{\mathcal{R}\}$ (Section 4.1) and the evaluations of $\{\mathcal{O}\}$ collected with AMT (Section 4.2 and Section 4.3).

4.1 Data Collection for Eye Opening Results

Although the system does not limit the identity of *reference* and *target* to be the same, in our data collection, we do not conduct cross-identity synthesis (both images are of the same person). We sought to collect sufficient data (both shut eyes and open eyes) having varying poses, illuminations, and expressions. However, people seldom post accidental shut-eyes images in their online personal photo albums (most are discarded). To simulate the real-life scenario of correcting accidental shut-eyes photos, we (1) collected shut-eyes *target* images from video stills and (2) specifically used photographs as *references*.

We collected a *target set* $\{\mathcal{T}\}$ from HD videos. Given a video, we detected face landmarks in every frame using a face alignment algorithm [Saragih 2011]. To detect frames with shut eyes (e.g., a blink), we computed the eye-corner angle, based on which a threshold applied. Together with each shut-eyes frame, we also collected a *ground-truth* counterpart of the same person with open eyes. The open-eyes frame was collected manually from a nearby frame to ensure a similar appearance apart from the eyes. We denote the collected ground-truth set by $\{\mathcal{G}\}$.

For a subject k, we first collected her *target set* $\{\mathcal{T}^{(k)}\}$ as previously described. Then we collected her *reference set* $\{\mathcal{R}^{(k)}\}$

consisting of photos of the same person but not from the video in which $\{\mathcal{T}^{(k)}\}$ was collected. For the purpose of testing our algorithm under variable data collection conditions, the *target set* and *reference set* were collected with different face pose, expression, illumination conditions, and so on. If, for a particular subject, we had m *target* images and n *reference* images, then $\{\mathcal{O}\}$ contained $m \times n$ results for that subject. We used celebrities as the subjects in our dataset. For each subject, we collected shut-eye frames from 10 different interview videos in which the faces are clear and in relatively high resolution. For every subject, the differences between the videos included head pose, expression, illumination, makeup, age, and shooting location, and so on. The *reference set* were photos of the same celebrities from the Internet, also under uncontrolled conditions. Therefore, both the *targets* and *references* were collected "in-the-wild."

4.2 Crowd-Sourcing Image Ratings

We use AMT to collect evaluations by asking the workers (participants) the following question: "Do the eyes look realistic in this image?" Four answers are made available for each question:

1. Definitely unrealistic;
2. Probably unrealistic;
3. Probably realistic, and
4. Definitely realistic.

To control the quality of the evaluation, in each assignment, we present a worker with 24 images to rate. Three of the 24 images are unedited open-eyes frames from $\{\mathcal{G}\}$, called *positive controls*. One image is a manually chosen obviously failed result called the *negative control*. The other 20 images are random synthetic results in $\{\mathcal{O}\}$. The identities of the results are mixed in the assignments such that each assignment contains images from different subjects. We collected evaluations for 8,000 randomly sampled results from eight subjects (approximately 20,000 results in total). Each image is evaluated by three different workers. We exclude random clicks by removing assignments for which either the negative control is rated high or the positive controls are rated low. Specifically, we require that for each assignment:

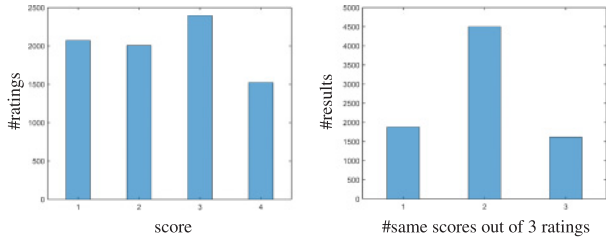
(1) At least two of three *positive controls* should be rated 3 or 4;
(2) None of the *positive controls* should be rated 1;
(3) *Negative control* should be rated either 1 or 2.

4.3 Human Evaluations

We collected 24,000 clicks from AMT on the results in $\{\mathcal{O}\}$:
28.7% were rated 1, Definitely unrealistic;
23.3% were rated 2, Probably unrealistic;
23.6% were rated 3, Probably realistic;
24.6% were rated 4, Definitely realistic.
Please find in our online supplementary document[1] examples of the results with corresponding evaluation scores from AMT workers.

Among all unedited open-eyes images from $\{\mathcal{G}\}$, 3.3% were rated 2, 18.9% were rated 3, and 77.9% were rated 4. The average score of an unedited image in our experiment is 3.7. In addition, among all negative controls, 94.8% are rated 1 and 5.2% are rated 2.

For every result, for which we have collected ratings from Amazon Mechanical Turk, we assigned a unique score to it by taking the most agreed score from different workers. For example, if result $\mathcal{O}_i$ received three AMT scores [3, 3, 4], then its score is $S(\mathcal{O}_i) = 3$, and the agreement level for this result is $A(\mathcal{O}_i) = 2$. The distribution of unique scores and agreement levels is shown in Figure 6. We discard data with an agreement level less than 2.



(a) Statistics of ratings (b) Number of agreed scores

Fig. 6. Statistics of ratings harvested from Amazon Mechanical Turk. (a) Most-agreed score statistics. The most-agreed rating score is assigned as a label for each image in the results. For those images with no agreed rating, we take the lowest score. (b) Agreement level statistics: Most results have two equal scores from three different raters.

5. LEARNING TO PREDICT BETTER EXAMPLES

Under a fully automatic pipeline, with a few computer vision components involved, it is hard to guarantee the quality of the result for every *target* and *reference*. For example, opening the eyes in a face image with frontal pose using a profile face example would be difficult or may not even be feasible in some extreme cases. Given an image with shut eyes, there will be a limited number of images in a personal album that can be used as an example to generate a realistic result. It is interesting and important to identify which of those examples would ensure good results by applying our method. In related previous work [Shih et al. 2014; Bitouk et al. 2008], the compatibility of example and input were considered important, as image pairs were ranked in a pre-defined manner. In contrast, we learn the ranking that automatically predicts good examples for eye editing on the basis of crowd-sourced annotated data.

5.1 Input Pair Compatibility

Our method works in an example-based fashion. It can be described as a function F from an input pair $\langle \mathcal{T}, \mathcal{R} \rangle$ to an output result $\mathcal{O}$:

$$\mathcal{O} = F(\langle \mathcal{T}, \mathcal{R} \rangle). \tag{5}$$

We represent the quality of the result (how realistic the result looks) using a scoring representation, which can be written as a function of the result $S(\mathcal{O})$ where S denotes a human perceptual scoring function. We can see that, given an input pair $\langle \mathcal{T}, \mathcal{R} \rangle$, the quality of result is

$$S(\mathcal{O}) = S(F(\langle \mathcal{T}, \mathcal{R} \rangle)) = f(\langle \mathcal{T}, \mathcal{R} \rangle). \tag{6}$$

In other words, the quality score of the output is a function f of the input pair $\langle \mathcal{T}, \mathcal{R} \rangle$. Note that both F and S are highly nonlinear and very complex to model.

In related previous work [Shih et al. 2014; Bitouk et al. 2008] the similarity of the input image pair was considered an indicator of the output quality. The function $S(\cdot)$ was defined simply as

$$S(\mathcal{O}) = f(\psi(\mathcal{T}, \mathcal{R})) = f(\|\phi(\mathcal{T}) - \phi(\mathcal{R})\|), \tag{7}$$

where $\psi(\cdot, \cdot)$ is a function of the input pair, $\phi(\cdot)$ is a feature extracted from the input image, and $f(\cdot)$ is either a real-value function (for ranking or score regression) or a thresholding function (for classification). However, the limitations of this approach are

[1] http://www3.cs.stonybrook.edu/ cvl/content/eyeopener/eyeopener.html.

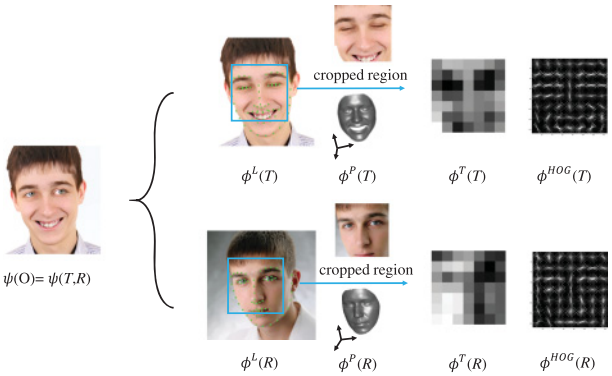


Fig. 7. Image pair feature extraction. The result of our algorithm is computed based on the features extracted from the *target* and *reference* inputs: (1) ϕ^L: landmarks; (2) ϕ^P: estimated face pose; (3) ϕ^T: "tiny image" intensity descriptor, and (4) ϕ^{HOG}: HOG descriptor extracted from a cropped face region. Image credits: Adobe Stock/Sabphoto.

(1) The distance between the input pair features captures only the difference between them but not their individual properties (only a binary term, no unary), and (2). the similarity measure and $f(\cdot)$ are defined heuristically and are not necessarily optimal in any sense. In our work, we propose to learn a more general function using human evaluations as a better strategy to predict the output score.

5.2 Image Pair Features

The score depends on multiple factors in both the *target* and *reference*, including their poses, expressions, and illumination conditions. However, since the algorithm is not affected by any region of the image other than the face, we design features that focus on the face area.

We use $\mathcal{F}$ to denote the feature type and, given an image I, we extract the following features $\phi^{\mathcal{F}}(I)$:

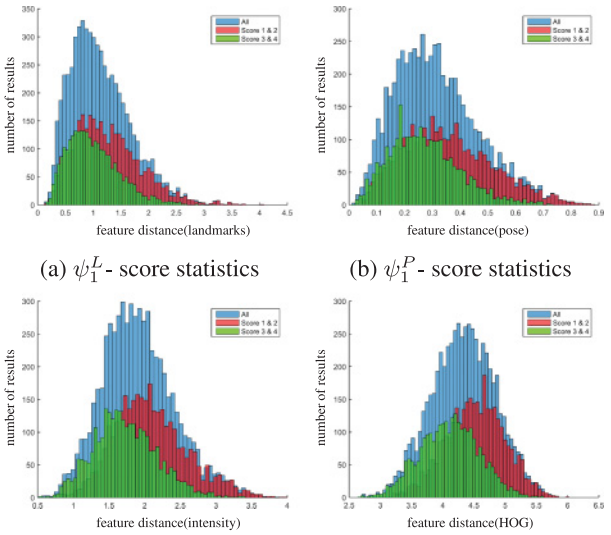
—$\phi^L(I)$ is simply the normalized landmark positions in the image. Landmarks play an important role in our algorithm, especially in pose warping and expression flow.
—$\phi^P(I)$ is a three-dimensional vector representing the pose of the face in the image.
—$\phi^T(I)$ is a "tiny image" [Torralba et al. 2008] representation of the intensity pattern on the face. Since the face is aligned by landmarks, we crop image I to I_{face} such that I_{face} is a square region on the face space(see Figure 7). We subsample the intensity of I_{face} to a 7×7 patch to capture the large scale intensity pattern of the face, and use the 49-dimensional vector as the feature.
—$\phi^{HOG}(I)$ is the Histogram of Oriented Gradients feature [Dalal and Triggs 2005] that captures both pose and expressions. We extract HOG on the cropped image shown in Figure 7. We divide the crop into 7×7 cells (the crop is re-scaled to 224×224 pixels, where each cell has size 32×32).

As previously described, the simplest feature for an input pair is the feature distance between two images:

$$\psi_1^{\mathcal{F}}(\mathcal{T}, \mathcal{R}) = ||\phi^{\mathcal{F}}(\mathcal{T}) - \phi^{\mathcal{F}}(\mathcal{R})||, \tag{8}$$

in which $\mathcal{F} \in \{L, P, T, HOG\}$.

Figure 8 illustrates how the crowd-sourced evaluation varies with different feature distances $\psi_1(\mathcal{T}, \mathcal{R})$. From Figure 8, we can see that outputs with larger feature distances are more likely to produce a lower score (less realistic). Thus, all chosen features are at least weakly correlated with the human evaluated quality of the output.



(a) ψ_1^L- score statistics (b) ψ_1^P- score statistics

(c) ψ_1^T- score statistics (d) ψ_1^{HOG}- score statistics

Fig. 8. Feature distance-score statistics. The distribution of different feature distances: (a) ψ_1^L (landmarks), (b) ψ_1^P (pose), (c) ψ_1^T (intensity), and (d) ψ_1^{HOG} (HOG). We use different colors to denote the distribution of the results for different scores. Blue: All scores. Red: Scores 1 and 2 (unrealistic); Green: Scores 3 and 4 (realistic). The distributions show that pairs with larger feature distances are more likely to generate results that are scored as unrealistic (i.e., score 1 or 2).

Note that ψ_1 is a similarity-based method like those in Shih et al. [2014] and Bitouk et al. [2008] in which compatibility is defined as feature similarity between the *target* and *reference* images. However, unlike Shih et al. [2014] and Bitouk et al. [2008], when there are no human evaluations, for example, selection, our method builds on well-defined criteria.

One alternative feature for describing an input pair is the feature concatenation of *target* and *reference*:

$$\psi_2^{\mathcal{F}}(\mathcal{T}, \mathcal{R}) = \left[\phi^{\mathcal{F}}(\mathcal{T}), \phi^{\mathcal{F}}(\mathcal{R})\right] \tag{9}$$

in which $\mathcal{F} \in \{L, P, T, HOG\}$.

Combining different types of ψ_1, we obtain a feature representing the differences of the image pair on all feature types described above:

$$\psi_1^C = \left[\psi_1^L, \psi_1^P, \psi_1^T, \psi_1^{HOG}\right]. \tag{10}$$

Concatenating different types of ψ_2, we have a feature representing the information in both the *target* and *reference*:

$$\psi_2^C = \left[\psi_2^L, \psi_2^P, \psi_2^T, \psi_2^{HOG}\right]. \tag{11}$$

5.3 Realistic result prediction

In order to predict what *references* are potentially compatible with a given *target*, we train a predictive model based on crowdsourced data.

We are interested in predicting a score to indicate how realistic the result looks using the AMT feedback. We model the problem as a regression task, in which we learn a regression function

(a) *target* (b) *reference* (c) alpha blending (d) gradient domain blending (e) image harmonization (f) our result

Fig. 9. A comparison between different image compositing methods. (a) Target image; (b) reference image; (c) result by alpha blending; (d) result by Error Tolerant Gradient-domain Blending [Tao et al. 2013] ; (e) result by Image Harmonization [Sunkavalli et al. 2010] ; (f) our result. Our method is more robust to differences in illumination conditions. Image credits: Adobe Stock/mimagephotos (first row), Adobe Stock/auremar (second row), Adobe Stock/Sabphoto (third, fourth, and seventh row), Adobe Stock/BillionPhotos.com (fifth row), Adobe Stock/gstockstudio (sixth row).

between the human evaluation score $S(\mathcal{O})$ and the image pair feature $\psi(\mathcal{T}, \mathcal{R})$. Our regression model is learned by support vector regression (SVR) [Smola and Schölkopf 2004] with a non-linear radial basis function kernel. We report the correlation and root-mean-square error (RMSE) between the prediction and ground-truth human labeling in Table I. We obtain a 0.48 correlation of predicted scores with human-labeled scores by using the combination feature ψ_{c2} where the RMSE is 1.07.

Table I. Accuracy of the Regression Model (SVR) Learned Using Different Features, Reported as Correlation and RMSE w.r.t. Human Labeling

Feature Type($\mathcal{F}$)	Landmarks		Pose		Intensity		HOG		Combination	
Feature	ψ_1^L	ψ_2^L	ψ_1^P	ψ_2^P	ψ_1^T	ψ_2^T	ψ_1^{HOG}	ψ_2^{HOG}	ψ_1^C	ψ_2^C
Correlation	0.29	0.38	0.35	0.4	0.38	0.41	0.45	0.44	0.47	**0.48**
RMSE	1.18	1.18	1.16	1.18	1.12	1.15	1.09	1.07	1.07	**1.07**

Fig. 10. Predicted scores indicate the level of realism of the result directly from the input pair features described in Section 5.2. Given a target, we use a trained regression model to predict the score of the results. The higher the score, the more realistic the result. The violet numbers above the result images indicate the prediction and the black numbers are human evaluations. The image below each result is the corresponding reference. Results are sorted from the highest predicted score to the lowest. Training data do not include any image of the test subject. Image credits: Adobe Stock/Sabphoto.

6. EXPERIMENTAL RESULTS

We compare the results of our color adjustment and blending (Section 3.3 and Section 3.4) with other common techniques described in Figure 9. All results are produced after applying the pose correction [Yang et al. 2011] described in Section 3.2. Our approach generates more natural-looking eye-opening results compared to alpha-blending, gradient domain blending [Tao et al. 2013], or multi-scale image harmonization [Sunkavalli et al. 2010]. Note that Yang et al. [2011] used a gradient domain compositing technique, similar to the one shown in Figure 9(d), which is sensitive to varying illumination conditions. Therefore, carefully selected examples and certain human interactions were required for expression flow [Yang et al. 2011] in order to achieve realistic results. Instead, our approach is fully automatic and our results are more robust to boundary mismatches caused by different illumination conditions. Moreover, our experiments verify the argument of Expression Flow [Yang et al. 2011] that 3D warping achieves better face-edit results compared to a purely 2D warping (We compared results of SIFT flow [Liu et al. 2011] and large displacement optical flow (LDOF) [Brox and Malik 2011]). In the 3D warping method, the correspondences being regularized by the 3D face structure and pose prior are more functionally robust.

In previous sections, we have shown that AMT users found approximately half of our results to be realistic. Using this feedback, we train models to predict compatible references from a personal album, given a target. We show two examples in Figure 10, where our model can find reasonably good examples for a given target. We trained a SVR [Smola and Schölkopf 2004] model with features extracted from an input image pair ψ_{c2} (see Section 5.2) to predict a score $\hat{S}(\mathcal{O})$ to describe how compatible the pair is. Our trained model computes only simple features from the input pair to predict the score. Therefore, we can efficiently mine a very large album for potentially good references that can generate realistic results, even before running the editing algorithm.

Figure 10 shows the predicted score values $\hat{S}(\mathcal{O})$ (in violet) on the top-left of the results for a few subjects. The black number under the predicted score is the actual human rating. Below the results are the corresponding references used by the editing algorithm. The results are sorted according to the predicted score. Please find more results in our online supplementary document.[1]

We directly apply our regression model to *reference* retrieval. We assign the following labels $L(\mathcal{O})$ to the results according to AMT scores $S(\mathcal{O})$:

$$L(\mathcal{O}) = \begin{cases} +1 & \text{if } S(\mathcal{O}) \in \{3,4\} \\ -1 & \text{if } S(\mathcal{O}) \in \{1,2\} \end{cases}. \tag{12}$$

From the album, we retrieve the *references* that will generate results with $L(\mathcal{O}) = 1$. In testing, we decide on the labels using a single

threshold t on the scores predicted by the SVR model:

$$\hat{L}(\mathcal{O}, t) = \begin{cases} +1 & \text{if } \hat{S}(\mathcal{O}) \geq t \\ -1 & \text{if } \hat{S}(\mathcal{O}) < t \end{cases}. \quad (13)$$

We perform leave-one-out subject cross-validation. On the AMT feedback dataset, we train on data from seven subjects' data and test on the eighth subject. We repeat this for all eight subjects. Figure 13 shows the average of the Precision-Recall curves of the models trained by the different features defined in Section 5.2. Precision and recall are defined as follows with a threshold t:

$$\text{Precision}(t) = \frac{\#(\hat{L}(\mathcal{O}, t) = +1 \wedge L(\mathcal{O}) = +1))}{\#(\hat{L}(\mathcal{O}, t) = +1)}, \quad (14)$$

$$\text{Recall}(t) = \frac{\#(\hat{L}(\mathcal{O}, t) = +1 \wedge L(\mathcal{O}) = +1))}{\#(L(\mathcal{O}) = +1)}. \quad (15)$$

Using the SVR model trained by feature ψ_2^C, we achieved a 0.7 average precision (AP) in retrieving "suitable" *reference* images for realistic eye opening results, while the random guess AP was 0.46. For all *target* images in our dataset, the average hit rate of top 1, top 2, and top 5 ranked results are 0.7, 0.69, and 0.64, respectively. We also trained Random Forests Regression [Breiman 2001] on our collected dataset and observed a 0.67 AP and RMSE of 1.10 with the ψ_2^C feature (see Figure 17 in our online supplementary document[1] for the precision-recall curve).

To provide intuition on how future work could be directed to improve the performance of the automatic editing pipeline, we conducted a visual inspection of 2,000 randomly sampled results in our dataset. For each result, we assigned one or more of the following five tags: (a) realistic results, (b) with pose warping artifacts, (c) with unmatched illumination artifacts, (d) with unrealistic expression, and (e) look unrealistic due to other reasons. Of the results, 48.6% were tagged (a), 25.1% were tagged (b), 25.8% were tagged (c), 15.9% were tagged (d), and 4.3% were tagged (e). We conclude that improvements in illumination matching and pose-warping algorithms would be the most beneficial for our system. Please find examples with these labels in the online supplementary document.[1]

According to Zhu et al. [2014], the eye expression is crucial for overall facial expression attractiveness. Figure 11 shows comparisons of facial expression attractiveness between shut-eyed images (Figure 11(a)) and results generated by our algorithm (Figures 11(b) and (c)). We randomly sampled 100 images from our datasets, which included shut-eyed images from different subjects and corresponding open-eyed images generated by our algorithm with predicted score higher than 3. The average expression attractiveness score for shut-eyes images was 0.50, while for open-eyed images it was 0.74. Based on the scores predicted by the off-the-shelf model trained by Zhu et al. [2014], appearing on the top-left of the images (Fiure 11), we demonstrate that our algorithm can improve perceived facial expression attractiveness for appropriately chosen examples.

Moreover, besides the eye opening, our system is applicable to example-based gaze editing as shown in Figure 12. Gaze correction and editing are of interest in applications such as video conferencing [Yang and Zhang 2002; Wolf et al. 2010; Kuster et al. 2012; Kononenko and Lempitsky 2015], where a small range of corrections to frontal gaze in real time is important.

So far in this article, we have discussed the application of our technique to image pairs of the same subject. However, the algorithm scope extends to *cross-identity* eye appearance transfer. This could be useful for artistic purposes or avatar creation. We show a few cross-identity results in Figure 14. We use Anne Baxter's portrait as the *target* and Bette Davis's portraits as *references* (actresses from the movie "All About Eve" (1950)). The algorithm works reasonably well for both eyes compositing and predicting the realism level of the results. See additional results in our online supplementary document.[1]

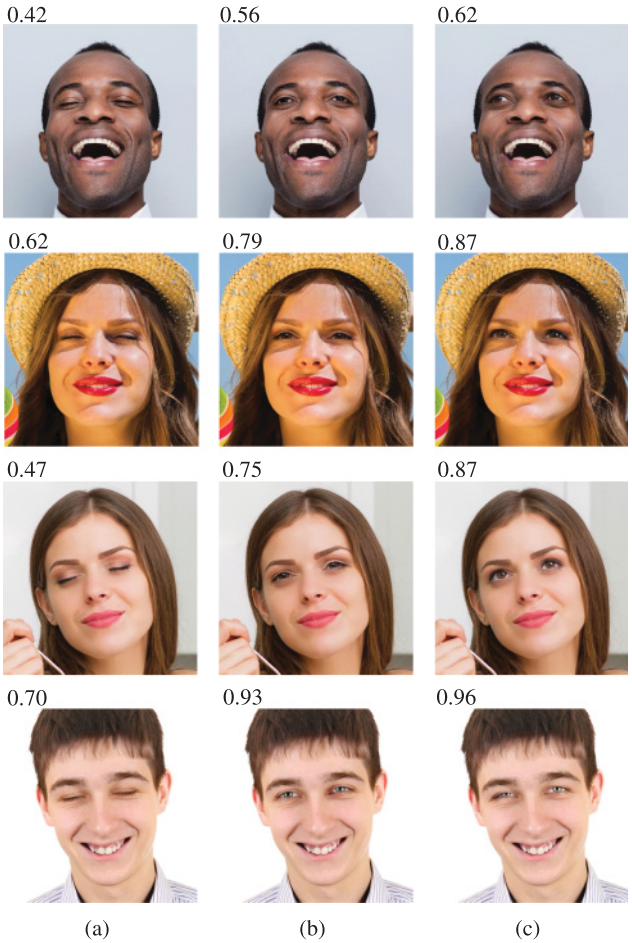


Fig. 11. Improving expressions. We compare the perceived attractiveness of the facial expressions in the shut-eyes images and the open-eyes results generated by our algorithm using the off-the-shelf model trained by Zhu et al. [2014]. (a) Shut-eyes images; ((b) and (c)) open-eyes results. The attractiveness score is indicated above each image. Higher values indicate a more attractive expression. Our algorithm can be applied to make a facial expression more attractive. Image credits: Adobe Stock/gstockstudio (first row), Adobe Stock/Tinatin (second and third row), Adobe Stock/Sabphoto (fourth row).

Table II. Average Running Time of Each Step in Our Implementation

Image size	Warp & Align	Local color	Boundary	Compositing	Total
550^2	3.7s	3.4s	1.8s	2.1s	11s
1000^2	11.8s	10s	6s	6.2s	34s

The typical image size in our experiments is 550 × 550 pixels. In our MATLAB implementation, the average running time of the entire synthesis pipeline is 11s. For 1000 × 1000 pixel inputs, the average running time is 34s. We show average times per step in Table II.

Fig. 12. Gaze editing. Our method can be used to change gaze given an appropriate example. Image credits: Adobe Stock/StockRocket (first row), Adobe Stock/Tanya Rusanova (second row), Adobe Stock/auremar (third row).

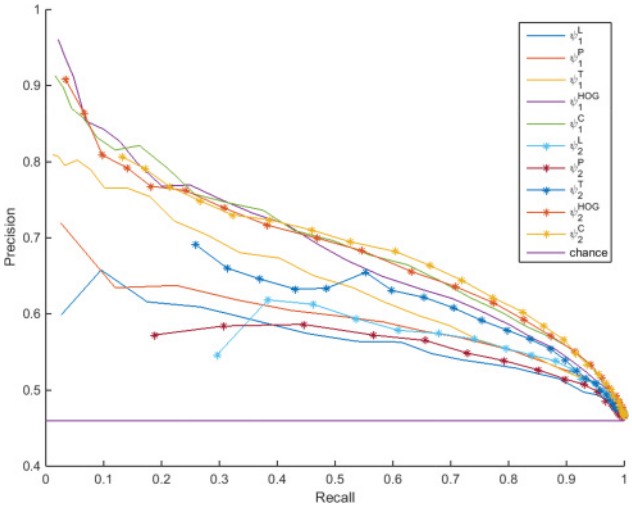


Fig. 13. Precision-Recall curves for compatible reference retrieval using an SVR model with different features. Combining features provides better performance than using individual features. Using feature ψ_2^C, our model achieves average precision of 0.70, while chance level is at 0.46.

7. LIMITATIONS AND FUTURE WORK

Our method has certain limitations: At this time, the automatic eye-opening system cannot handle extreme differences in pose and illumination (Figure 15) between the *target* and the *reference*. We can see from Figure 8 that the more distinct the input pair is, the less likely our algorithm is to generate plausible results. It is apparent that our system would benefit from further progress in face relighting and

Fig. 14. She's Got Bette Davis Eyes: experiment on cross-identity eye appearance transfer. *Target*: Anne Baxter as Eve Harrington in the motion picture *All About Eve* (1950). *References*: portraits of Bette Davis. On the top right of the result images, the yellow number indicates the predicted level of realism. Image credits: Getty Images/Bettmann Archive (first reference from left to right).

Fig. 15. Large differences in poses or illumination conditions between *target* and *reference* images are challenging. Image credits: Adobe Stock/gstockstudio (first row), Adobe Stock/Sabphoto(second row).

pose correction. However, despite this limitation, for most image pairs, our system manages to automatically recover and "edit away" pose and illumination mismatches. This explains the relatively lower impact these features have on the prediction scores (Figure 13). On the other hand, the compatibility of a given *reference* and *target* in global expression (as measured by HOG) is a stronger predictor of a good result. That is because our system does not automatically minimize expression differences in the image pair but relies on a good match from the photo collection instead. In future work, such automatic expression morphing may allow the use of a wider range of *reference* images.

There are several potential ways to improve our system:

—Robustness. Our method is based on a few assembled components. The robustness of every stage is crucial to the entire system. For example, pose warping cannot handle large differences in pose; local color transfer might produce artifacts under drastically different illumination conditions, and blending techniques do not address warping and re-coloring failures well. We believe the robustness of each stage itself can be improved in the future.
—Face relighting. Manipulating the illumination of face images is a challenging task. Our system, as well as other image editing applications, can benefit from better techniques for the control of face illumination.

—Predicting realism of face images. There are multiple factors influencing human perception of face realism and our current work explores the compatibility of the input images using handcrafted image features. In future, it would be useful to learn (possibly using deep learning) which features are important for our task.
—Run-time. Our MATLAB implementation takes around 10s, which is relatively slow. Improving the speed of the system will be important for interactive and video applications.

8. CONCLUSION

We presented an automatic eye-editing method that works effectively on face photos "in the wild." Its success is based on a new algorithm for local color adjustment and robust blending. We also present a learning approach for predicting the compatibility of an image pair based on crowd-sourced human annotations. This crowdsourcing approach can be extended to other example-based editing tasks such as entire face replacement [Bitouk et al. 2008; Dale et al. 2011; Garrido et al. 2014].

ACKNOWLEDGMENTS

We acknowledge Jianchao Yang and Kalyan Sunkavalli for helpful discussions.

REFERENCES

A. Agarwala. 2007. Efficient gradient-domain compositing using quadtrees. *ACM Trans. Graph.* 26, 3, 94.

A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. 2004. Interactive digital photomontage. *ACM Trans. Graph.* 23, 3, 294–302.

S. Bakhshi, D. A. Shamma, and E. Gilbert. 2014. Faces engage us: Photos with faces attract more likes and comments on instagram. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 965–974.

J. C. Bazin, D. Q. Pham, I. Kweon, and K. J. Yoon. 2009. Automatic closed eye correction. In *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2433–2436.

P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless. 2010. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Trans. Graph.* 29, 2, 10.

D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar. 2008. Face swapping: Automatically replacing faces in photographs. *ACM Trans. Graph.* 27, 3, 39.

V. Blanz and T. Vetter. 1999. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniquesk*. ACM Press/Addison-Wesley Publishing Co., 187–194.

L. Breiman. 2001. Random forests. *Mach. Learn.* 45, 1, 5–32.

T. Brox and J. Malik. 2011. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 3, 500–513.

X. Chen, M. Chen, X. Jin, and Q. Zhao. 2011. Face illumination transfer through edge-preserving filters. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 281–287.

N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 886–893.

K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. 2011. Video face replacement. *ACM Trans. Graph.* 30, 6, 130.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and Fei-L. Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 248–255.

Z. Farbman, G. Hoffer, Y. Lipman, Cohen-D. Or, and D. Lischinski. 2009. Coordinates for instant image cloning. In *ACM Transaction on Graphics (TOG)*. 28, 67.

P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormaehlen, P. Perez, and C. Theobalt. 2014. Automatic face reenactment. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 4217–4224.

P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Perez, and C. Theobalt. 2015. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In *Eurographics 2015*.

P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt. 2013. Reconstructing detailed dynamic face geometry from monocular video. In *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2013)*. Vol. 32. 158:1–158:10.

D. Guo and T. Sim. 2009. Digital face makeup by example. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 73–79.

J. Hays and A. A. Efros, 2007. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)* 26, 3.

K. He, J. Sun, and X. Tang. 2013. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 6, 1397–1409.

N. Joshi, W. Matusik, E. H. Adelson, and D. J. Kriegman. 2010. Personal photo enhancement using example images. *ACM Trans. Graph.* 29, 2, 12.

I. Kemelmacher-Shlizerman, S. Suwajanakorn, and S. M. Seitz. 2014. Illumination-aware age progression. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 3334–3341.

M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. 2014. Hipster wars: Discovering elements of fashion styles. In *Computer Vision–ECCV 2014*. Springer, 472–488.

V. Kolmogorov and R. Zabin. 2004. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 2, 147–159.

D. Kononenko and V. Lempitsky. 2015. Learning to look up: Realtime monocular gaze correction using machine learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4667–4675.

C. Kuster, T. Popa, J.-C. Bazin, C. Gotsman, and M. Gross. 2012. Gaze correction for home video conferencing. *ACM Trans. Graph.* 31, 6, 174.

P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. 2014. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 33, 4.

T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski. 2008. Data-driven enhancement of facial attractiveness. *ACM Trans. Graph.* 27, 3, 38.

C. Liu, J. Yuen, and A. Torralba. 2011. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 5, 978–994.

Z. Liu, Y. Shan, and Z. Zhang. 2001. Expressive expression mapping with ratio images. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 271–276.

M. H. Nguyen, J.-F. Lalonde, A. Efros, and F. De la Torre. 2008. Image-based shaving. *Comput. Graph. Forum* 27, 2, 627–635.

P. O'Donovan, J. Lībeks, A. Agarwala, and A. Hertzmann. 2014. Exploratory font selection using crowdsourced attributes. *ACM Trans. Graph.* 33, 4, 92.

D. Parikh and K. Grauman. 2011a. Interactively building a discriminative vocabulary of nameable attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 1681–1688.

D. Parikh and K. Grauman. 2011b. Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 503–510.

P. Pérez, M. Gangnet, and A. Blake. 2003. Poisson image editing. In *ACM Transactions on Graphics (TOG)*. Vol. 22. ACM, 313–318.

F. Pitié, A. Kokaram, and R. Dahyot. 2005. N-dimensional probability density function transfer and its application to color transfer. In *ICCV 2005*. Vol. 2. 1434–1439 Vol. 2.

J. Saragih. 2011. Principal regression analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2881–2888.

F. Shi, H.-T. Wu, X. Tong, and J. Chai. 2014. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Trans. Graph.* 33, 6, 222.

J. Shi and J. Malik. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 8, 888–905.

Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand. 2014. Style transfer for headshot portraits. *ACM Trans. Graph.* 33, 4, 148.

A. J. Smola and B. Schölkopf. 2004. A tutorial on support vector regression. *Stat. Comput.* 14, 3, 199–222.

K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister. 2010. Multi-scale image harmonization. In *ACM Trans. Graph*. 29, 125.

S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz. 2014. Total moving face reconstruction. In *Computer Vision–ECCV 2014*. Springer, 796–812.

S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. 2015. What makes tom hanks look like tom hanks. In *Proceedings of the IEEE International Conference on Computer Vision*. 3952–3960.

M. W. Tao, M. K. Johnson, and S. Paris. 2013. Error-tolerant image compositing. *Int. J. Comput. Vis.* 103, 2, 178–189.

A. Torralba, R. Fergus, and W. T. Freeman. 2008. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 11, 1958–1970.

Y. Wang, Z. Liu, G. Hua, Z. Wen, Z. Zhang, and D. Samaras. 2007. Face relighting from a single image under harsh lighting conditions. *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, 1–8.

Y. Wang, L. Zhang, Z. Liu, G. Hua, Z. Wen, Z. Zhang, and D. Samaras. 2009. Face relighting from a single image under arbitrary unknown lighting conditions. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 11 (Nov.), 1968–1984.

P. Welinder, S. Branson, P. Perona, and S. J. Belongie. 2010. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*. 2424–2432.

Z. Wen, Z. Liu, and T. S. Huang. 2003. Face relighting with radiance environment maps. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. IEEE, II–158.

L. Wolf, Z. Freund, and S. Avidan. 2010. An eye for an eye: A single camera gaze-replacement method. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 817–824.

F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. 2011. Expression flow for 3d-aware face component transfer. *ACM Trans. Graph.* 30, 4, 60.

R. Yang and Z. Zhang. 2002. Eye gaze correction with stereovision for video-teleconferencing. In *Computer VisionECCV 2002*. Springer, 479–494.

L. Zhang and D. Samaras. 2006. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 3, 351–363.

J.-Y. Zhu, A. Agarwala, A. A. Efros, E. Shechtman, and J. Wang. 2014. Mirror mirror: Crowdsourcing better portraits. *ACM Trans. Graph.* 33, 6, 234.

Received October 2015; revised April 2016; accepted April 2016