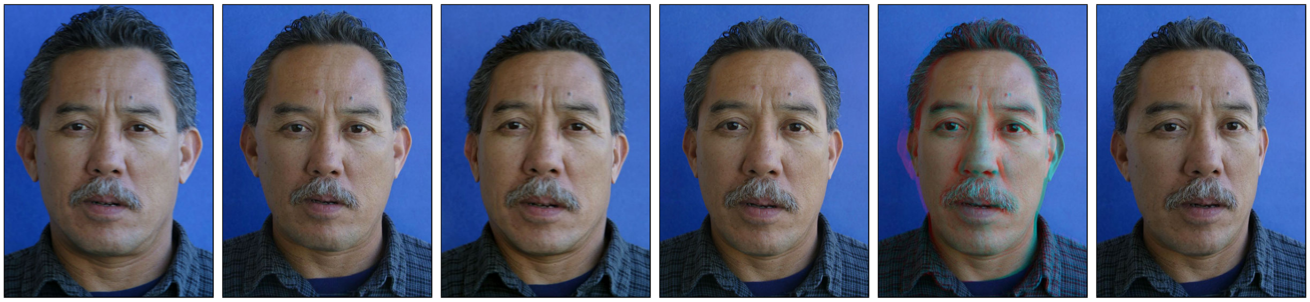
Perspective-aware Manipulation of Portrait Photos

Ohad Fried
Princeton University

Eli Shechtman
Adobe Research

Dan B Goldman*
Adobe Research

Adam Finkelstein
Princeton University



(a) far: real photo (b) far: warped from (d) (c) near: warped from (a) (d) near: real photo (e) anaglyph from (d) (f) rotated from (d)

Figure 1: *Comparing real photos taken with a far (a) or near (d) camera, one can observe the subtle effect of perspective on portrait photos. We simulate this effect by warping (a) → (c) to match the apparent distance of (d); and also (d) → (b) to match the distance of (a). These warps are guided by an underlying 3D head model. This framework can also generate stereo anaglyphs (e) and apparent head rotation (f).*

Abstract

This paper introduces a method to modify the apparent relative pose and distance between camera and subject given a single portrait photo. Our approach fits a full perspective camera and a parametric 3D head model to the portrait, and then builds a 2D warp in the image plane to approximate the effect of a desired change in 3D. We show that this model is capable of correcting objectionable artifacts such as the large noses sometimes seen in "selfies," or to deliberately bring a distant camera closer to the subject. This framework can also be used to re-pose the subject, as well as to create stereo pairs from an input portrait. We show convincing results on both an existing dataset as well as a new dataset we captured to validate our method.

Keywords: faces, portraits, perspective, image enhancement

Concepts: •**Computing methodologies** → **Image manipulation; Computational photography;**

1 Introduction

> *Photographers deal in things which are continually vanishing and when they have vanished there is no contrivance on earth which can make them come back again.* –Henri Cartier-Bresson

In more than a century since the invention of the daguerreotype, photographers have developed a set of conventions for effective composition of a photo. For example, the combination of subject pose, camera angle, and lighting can help define a jawline. Even the *camera distance* to the subject impacts perception; the literature shows that portraits taken up close are associated with terms such as "peaceful" and "approachable", whereas headshots taken from further away are perceived as "attractive", "smart" and "strong" [Bryan et al. 2012; Perona 2007; Perona 2013].

This paper introduces a method that can subtly alter apparent camera distance and head pose *after* a portrait has been taken (Figure 1). This system fits a virtual camera and a parametric 3D head model to the photo, then models changes to the scene in the virtual camera, and finally approximates those changes using a 2D warp in the image plane. Similar frameworks have been used for a variety of applications including changing pose and gender [Blanz and Vetter 1999], face transfer [Vlasic et al. 2005], and expression transfer [Yang et al. 2011]. Our work specifically builds on the FaceWarehouse approach of Chen et al. [2014b]. These prior methods all use a weak perspective camera model, which is a reasonable approximation only when scene points are all at a similar distance to the camera. In contrast, our approach uses a full perspective camera model, which allows us to modify camera distance and handle scenes that come very close to the camera. In a full perspective camera model, the distance and field of view parameters are *nearly* interchangeable, which makes optimization challenging. Nevertheless, this model is necessary for several of the effects that we show, especially treatment of "selfies."

Today most photos are taken using mobile devices with fixed focal length. This trend accounts for the sudden explosion of the "selfie" – 2013 word of the year in the Oxford Dictionary – meaning a portrait taken of oneself, often with a smartphone. Selfies are typically shot at arm's length, leading to visible distortions similar to the fisheye effect but with their own characteristics, most notably an enlarged nose. In some cases this selfie effect may be desired, but professional portrait photographers often prefer to position the camera several meters from the subject, using a telephoto lens to fill the frame with the subject [Valind 2014]. Figure 2 shows two photos of the same subject, revealing the effects of this tradeoff [Orlov 2016]. Our framework allows one to simulate a distant camera when the original shot was a selfie, and vice versa, in order to achieve various artistic goals – reducing distortion, making a subject more approachable, or adapting a portrait such that it may be composited into a group shot taken at a different distance.

We show that our framework can also create convincing stereo pairs from input portraits or videos, rendered as anaglyphs. The approach relies on the full perspective camera available in our 3D model. Finally, our method is also capable of other applications shown in

[*] The author is currently at Google.

SIGGRAPH '16 Technical Paper,, July 24 - 28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925933

previous work using a weak perspective model, such as simulating small rotations of the subject's head. Our main contributions are:

- The ability to edit perceived camera distance in portraits.
- A robust head fitting method that estimates camera distance.
- A new image warping approach that approximates changes in head or camera pose.
- A method to create stereo pairs from an input portrait.
- Evaluation of our approach using an existing dataset and a new dataset captured for this purpose.

2 Related Work

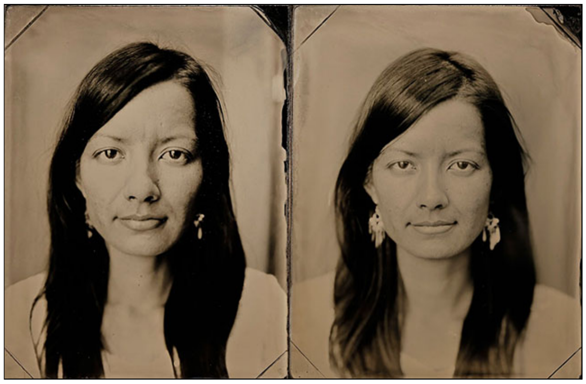
Despite a large body of work on face modeling, 3D face shape estimation from a single image is still considered challenging, especially when the subject is captured under unconstrained conditions (varying expressions, lighting, viewpoint, makeup, facial hair). High quality face reconstruction methods often require the subject to be scanned under controlled laboratory conditions with special equipment such as lighting rigs and laser scanners [DeCarlo et al. 1998; Weise et al. 2007; Alexander et al. 2009; Bradley et al. 2010]. Kemelmacher and Seitz [2011] showed it is possible to reconstruct a face shape from a large Internet collection of a person's photos using ideas from shape from shading. These methods are not applicable in a single photo scenario.

In their seminal work, Blanz and Vetter [1999] fit a 3D face morphable model to a single input image, texture-map a face image onto a 3D mesh, and parametrically change its pose and identity. Vlasic et al. [2005] extended their work using a multilinear model to handle expressions and visemes. FaceWarehouse [Cao et al. 2014b] extended the model from the face region to an entire head shape. Other single-image reconstruction methods include an approach based on patch-based depth synthesis from a 3D dataset [Hassner and Basri 2006], photometric stereo with a 3D template prior [Kemelmacher-Shlizerman and Basri 2011] and a 3D template corrected with a flow optimization [Hassner 2013]. Unlike morphable models, the latter do not allow changing the identity and expression of the subject.

In order to edit 3D face properties in a photograph using any of the above methods, the face has to be accurately segmented from the background, texture-mapped onto the face mesh, and then projected back to the image after the mesh is edited. The background, the rest of the head, and the eyes and teeth must be adjusted – often manually – to fit the pose change. This complex pipeline can result in an unrealistic appearance due to artifacts of segmentation, color interpolation, and inpainting.

An alternative approach uses the 3D model to generate a 2D warp field induced from a change in 3D, and apply this warp directly on the photograph [Yang et al. 2011; Yang et al. 2012a]. This approach doesn't support extreme edits, but it can be fully automated and often leads to more realistic results. We adopt this approach, driving our warp field with a multilinear morphable model with parametrized pose, identity, and expression.

Existing morphable model methods typically have two main drawbacks: First, the camera distance is given as input (or assumed to be infinite) and remains fixed; and second, there are no annotations near the top of the head, which we show poses a major problem for fitting and altering the apparent camera distance. We extend a multilinear model to incorporate camera distance, and present an optimization algorithm for the more challenging fitting problem that results. We also add a few annotations in some key locations and show in Section 4 that these are critical for our application.



©Anton Orlov, used with permission.

Figure 2: *Compare focal lengths. Left: close-up using 90mm wide angle lens with a large format camera (29mm equivalent on 35mm film). Right: distant shot with 265mm telephoto lens (84mm equiv.)*

The methods of Cao et al. [2013; 2014a] and Hassner et al. [2015] estimate a perspective camera model similar to our approach. Cao et al. drive a real-time animation with an input head video, but their system uses multiple frames for accurate estimation of model parameters, whereas our goal is to use a single input image. We tested some of their underlying assumptions and found them inapplicable to the case of single-image input (Section 3.3). Also, Cao et al. reduce the intrinsic matrix to a single parameter estimation (focal length), fixing the principal point offset to zero. In order to support, for example, cropped images, our model estimates this offset as well (2 extra parameters). Hassner et al. frontalize a face given an input image. They estimate the intrinsic camera matrix given a fixed 3D template model, since an accurate fit is not required for their task. In contrast, our method addresses the harder problem of jointly estimating camera and model parameters. Nonetheless, some features of the method proposed by Hassner et al. are complementary to ours, for example "borrowing" features from one side of the face to complete the other could be used to augment our system, in the case where there are occlusions in the input.

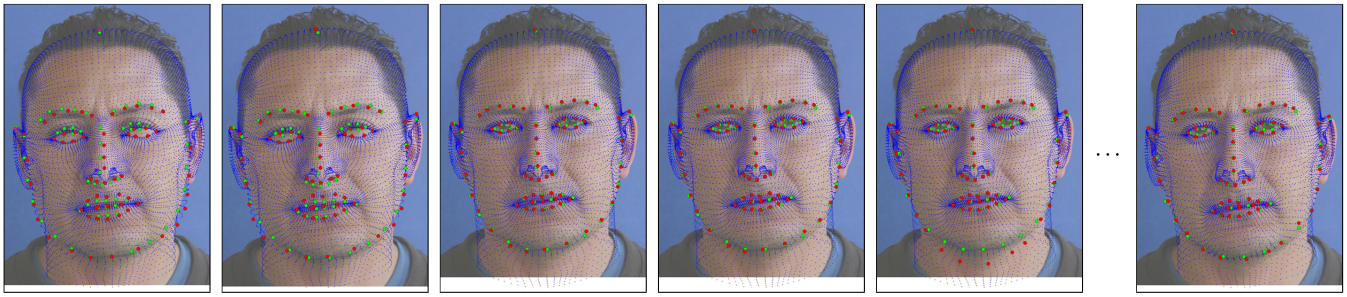
The perceptual literature draws a direct connection between camera distance, lens focal length, and the way we perceive people in photos: Portraits taken from up close are associated with terms such as "peaceful" and "approachable", while those taken from further away are "attractive", "smart" and "strong" [Perona 2007; Bryan et al. 2012; Perona 2013]. Cooper et al. [2012] further showed that portraits taken using a 50-mm lens are most likely to be viewed from a distance from which the percept will be undistorted.

The Caltech Multi-Distance Portraits Dataset [Burgos-Artizzu et al. 2014] contains portraits of different subjects taken from various distances. In their paper, the authors created a way to estimate the camera distance from an input portrait photo. We use their dataset to evaluate our method.

No previous method suggests changing the apparent camera distance in a photo. As far as we know, we present the first work to address the task of fixing portrait distortions due to camera distance.

3 Our Method

To perform perspective-aware manipulation, first we formulate a parameterized 3D model for a head and camera (Section 3.1), automatically detect fiducials in a photo (Section 3.2), and fit the model to the observed fiducials (Section 3.3). Next, we can alter the parameters of the model (e.g., move the camera or head pose, Section 3.4) and then approximate the resulting 3D changes as a 2D warp to the input image (Section 3.5).



Initialization Rotation & Translation Identity & Expression Full Camera 3D Landmarks . . . Final Fit

Figure 3: *Fitting procedure. Green dots are 2D fiducial points. Red dots are corresponding points on 3D mesh (shown in blue). Purple dots in leftmost image are three manually annotated fiducials for top of head and ears.* Images, from left to right: *Initialization gives a rough fit, but with some misalignments (e.g. eyes). Solving rotation and translation improves the silhouette fit (e.g. chin). Solving identity and expression fixes the eye and mouth misalignment. Solving the full camera model improves, in this case, the top of the head. After 3D landmark update the alignment is worse, but landmark locations on the 3D mesh are more accurate. Repeating the process produces a good final fit.*

3.1 Tensor Model

Our head model builds on the dataset collected by Chen et al. [2014b]. This dataset contains scans of 150 individual heads, each in 20 poses. Each head has 11,510 vertices in 3D (34,530 DOF). Expressions are represented by a blendshape model using 47 shapes.

Let us denote the average of all heads in the dataset as $A \in \mathbb{R}_{34530\times1}$. We calculate the difference of each head from the average and arrange the data in a tensor $Z \in \mathbb{R}_{34530\times150\times47}$, with dimensions corresponding to vertices, identities and expressions, respectively. We use high order SVD (HOSVD) [Tucker 1966] to calculate a core tensor $C \in \mathbb{R}_{40\times50\times25}$. Here our approach differs from that of Chen et al. [2014b], who do not perform SVD on the vertex dimension. We find that our compact representation still produces good results. Given the core tensor we use an identity vector $\beta \in \mathbb{R}_{1\times50}$ and an expression vector $\gamma \in \mathbb{R}_{1\times25}$, together with the original vector expansion calculated by HOSVD $v \in \mathbb{R}_{34530\times40}$ to generate a head with a specific expression and identity F' via:

$$F' = (C \otimes_1 v \otimes_2 \beta \otimes_3 \gamma) + A \tag{1}$$

Here $\otimes_i$ is the standard tensor-vector multiplication in the i-th dimension. Let us denote $F'' \in \mathbb{R}_{4\times11510}$ as the natural reshape of F' such that each row contains x, y, and z coordinates respectively, with an added row of ones to create homogeneous coordinates. In order to generate a head in a specific location and orientation, as seen by a camera, we need to multiply the head vertices (which are in the model coordinate system) with translation $T = [\mathbb{1}_3| - t] \in \mathbb{R}_{3\times4}$, rotation $R \in \mathbb{R}_{3\times3}$ and the upper-triangular intrinsic matrix $K \in \mathbb{R}_{3\times3}$. Thus, our full model (omitting the perspective divide for simplicity) is:

$$F = K \cdot R \cdot T \cdot F'' \tag{2}$$

We found that a general intrinsic matrix K with five parameters leads to bad shape estimation. Instead we constrain the skew to be zero and the horizontal and vertical focal length parameters to be the same – reasonable assumptions for unaltered photos from modern cameras. This intrinsic matrix constrained to three DOFs significantly improves the fit.

We contrast this full perspective model with previous work that uses weak perspective (e.g. [Vlasic et al. 2005; Yang et al. 2011; Yang et al. 2012b; Cao et al. 2014b]) – essentially using orthographic projection, followed by non-uniform scaling. With weak perspective camera distance is represented by scaling, so there is no way to adjust distortions due to nearby cameras, e.g., as seen in selfies.

3.2 Fiducial Detection

The method of Saragih et al. [2009] automatically detects 66 fiducial points on faces: chin (17), eyebrows (10), nose stem (4), below nose (5), eyes (12), and lips (18). Unfortunately, these locations (which are also common for other detectors) are not sufficient for our purposes because they lack points above the eyebrows and on the ears. Since our system manipulates perspective, such points are crucial to model the effects on apparent head shape.

Rather than invent a new fiducial detector, which we leave for future work, we use an existing detector [Saragih et al. 2009], and manually annotate three extra points on top of the head and ears. We chose a small number of points to facilitate quick annotation (less than five seconds).

3.3 Fitting

Given an input image and the 69 fiducial point locations (Section 3.2) we would like to fit a head model to the image. Since all models in our dataset share the same vertex ordering, we know the location of the corresponding fiducial points on the 3D models. Armed with Equations (1) and (2) the task is now to find the best parameters β, γ, K, R, t (50 + 25 + 3 + 3 + 3 = 84 in total) such that the Euclidean distance between the fiducial points and the projection of the 3D landmarks is minimized.

Many fitting strategies are possible. We experimented with several and discuss them before describing our proposed approach. A naïve approach is to treat the problem as one large non-linear least square optimization. However, we found this approach gets stuck in local minima. Using coordinate descent, as described in Algorithm 1, obtained lower global error. Other works [Yang et al. 2011; Yang et al. 2012a; Cao et al. 2014b] also used coordinate descent. However our optimization problem is much harder due to the inherent non-linearity of the camera projection model (Algorithm 1 Line 6), which introduces ambiguity between the camera distance, focal length and the expression and identity parameters. We also tried adapting this naïve approach by using even more fiducial points, and achieved sub-par results. Our experience suggests that merely adding more points does not completely solve the problem.

We also experimented with the approach of Cao et al. [2013; 2014a] for focal length estimation. It assumes that the fitting error taken as a function of focal length is convex. We tested this convexity assumption in the context of our global optimization, by repeating their experiments using un-cropped images from the Caltech Multi-

Distance Portraits (CMDP) Dataset [Burgos-Artizzu et al. 2014], and found that convexity does not hold when calculated using a single image. Moreover, the global optimum of the focal length was off by an average of 35% and up to 89% from the EXIF value. In contrast, Cao et al. were able to obtain errors below 2% using multiple frames.

The aforementioned experiments led to a more deliberate design of the initialization and of the gradient descent order (e.g. adding Line 2 as a precursor to the optimization loop). All the results shown in this paper use a total of 3 iterations. The following sections explain the different subparts of the optimization. Figure 3 contains an overview of the fitting procedure.

Algorithm 1 Fit model to image

```
1: Initialize camera, identity and expression parameters (§3.3.1)
2: Solve rotation and translation (§3.3.2)
3: for $i$ in 1..num_iterations do
4:     Solve identity (§3.3.2)
5:     Solve expression (§3.3.2)
6:     Solve camera (§3.3.2)
7:     Update 3D landmark location (§3.3.3)
8: end for
```

3.3.1 Initialization

We extract the focal length f_E from the EXIF data of the image as an initial guess. We allow this value to change during optimization, to account for EXIF inaccuracies and the uncertainty of the exact location of the focal plane. We also use the distance t_c between camera and subject if it is known (e.g. in the dataset of Burgos-Artizzu et al. [2014]). We initialize our camera parameters to be:

$$K_0 = \begin{bmatrix} f_E & 0 & 0 \\ 0 & f_E & 0 \\ 0 & 0 & 1 \end{bmatrix}, t_0 = \begin{bmatrix} 0 \\ 0 \\ t_c \end{bmatrix}, r_x = r_y = r_z = 0 \qquad (3)$$

Here, r_x, r_y and r_z are the x, y and z rotation, respectively. If distance t_c is unknown we use a default value of 1m. Initializing β_0 and γ_0 to the average of all identity and expression vectors in our dataset, respectively, we solve for initial parameters β, γ, K, R, t using an interior-reflective Newton method [Coleman and Li 1996], minimizing the Euclidean distance between 2D fiducial points and the 2D projections of corresponding 3D landmarks. Specifically, let $L = \{l_i\}$ be the 2D fiducial locations (Section 3.2) and $H = \{h_i\}$ be the corresponding 3D head vertices projected to the image plane by Equation (2). Our objective is then:

$$\min_{\beta,\gamma,K,R,t} \sum_{i=1}^{N} \|l_i - h_i\|_2^2 \qquad (4)$$

where N is the number of fiducial points (69 throughout this paper).

3.3.2 Parameter Update

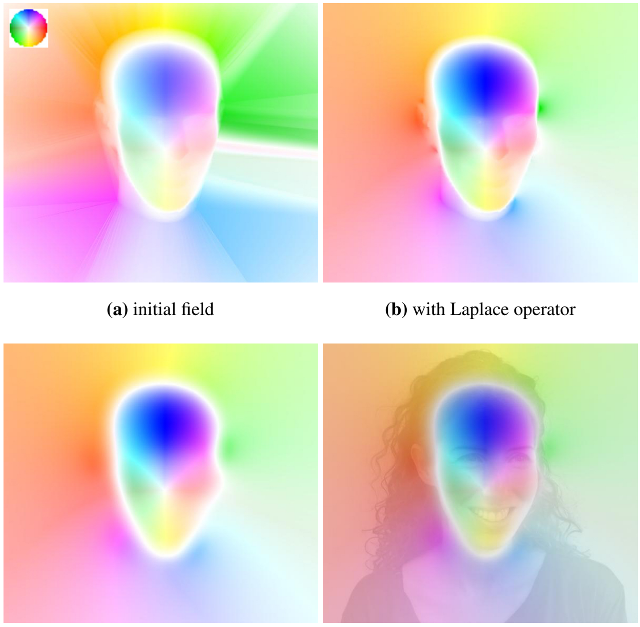
As introduced in Algorithm 1, we solve for rotation R and translation t once. Next holding these parameters fixed, we repeatedly solve for identity β, expression γ, and camera parameters K, R, t. As with initialization (Section 3.3.1), these optimizations use the interior-reflective Newton method to minimize Equation (4). We find it critical to solve first for rotation and translation only: Solving first for expression or identity results in a distorted face that overcompensates for bad pose. Solving first for the full camera matrix occasionally results in erroneous focal length.

3.3.3 3D Landmark Update

Some landmark locations are expected to remain fixed on the 3D model, regardless of view angle. For example, the corner of the eye should be the same vertex for any pose. However, other landmarks are pose-dependent. Specifically, the chin and the top of the head are entangled with pose. Of course, the chin doesn't actually change location; rather our fiducial detector detects contour points along the chin, and these contours are view-dependent. Thus, after initial calculation of a face shape and location, we need to recalculate the location of these "soft" landmarks. This step needs to be reasonably efficient because it is iterated many times in Algorithm 1. We follow an approach similar to that of Yang et al. [2011], with two modifications: First, we add the top of the head as a movable landmark. Second, their work used a face model, rather than a full head model. Because the projected shape is nearly convex, they described an approach that iteratively projects towards the convex hull in 2D to find the contour. Since we have a full head (including protruding ears) our projected shape is far from convex. We address this problem by a one time preprocessing step in which we find a restricted set of "valid" chin and head points (omitting ears and neck, for example) and then restrict the landmark update to consider only these valid points.

3.4 Changing Distance and Pose

Given a good fit between the input image and the head model, we can now manipulate the model. We move the virtual camera towards or away from the subject by changing the translation t. To rotate the head we adjust both translation t and rotation R, since translation is applied before rotation. Rotation is achieved by translation in a diagonal direction (relative to the line between camera and subject), followed by a rotation to place the head back in the visible frustum. These modifications result in a new projected head shape, which will guide the warp described next.


(a) initial field

(b) with Laplace operator

(c) smoothed

(d) with image

Figure 4: *Generating the dense warp field. (a) Initial dense field, with discontinuities in background and around face. (b) Improved background via discrete Laplace operator. (c) Smoothed using an averaging filter. (d) Overlay of the final warp field and input image.*

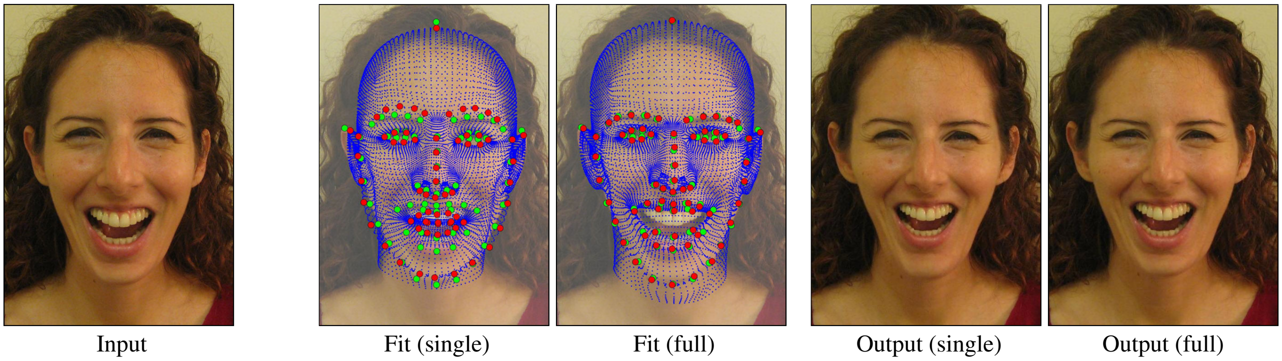


Figure 5: *Using a single head model vs. our full model that allows expression and identity variation. Dot colors as in Figure 3. The single model yields a bad fit, especially near the smile, thus resulting in an unnaturally narrow chin.*

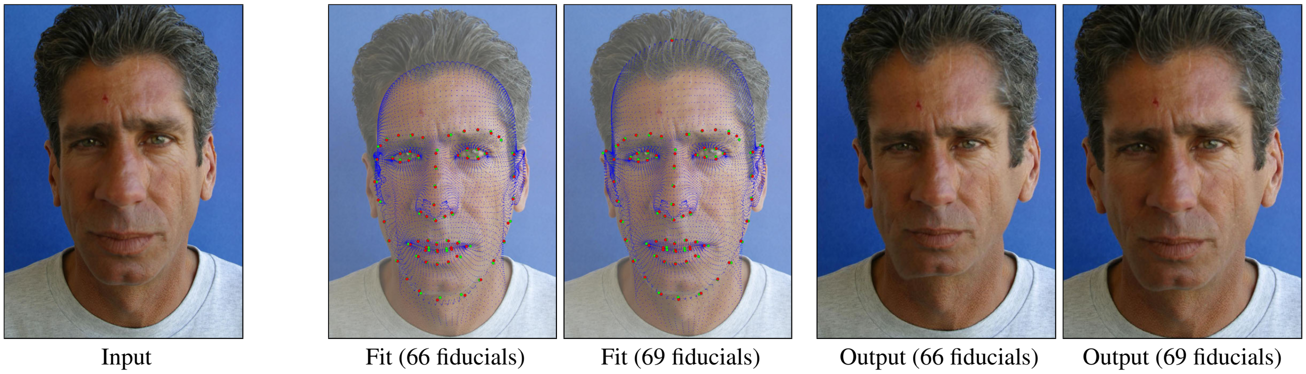


Figure 6: *Using the standard 66 fiducial points vs. adding 3 points for top-of-ear and top-of-head. Dot colors as in Figure 3. Fitting to 66 points produces inaccurate alignment near the ears and the top of the head, thus resulting in unnatural proportions and skew.*

3.5 Warping

After manipulating distance or pose we now have two sets of points: First, the original 3D face vertices that match the input image, and second, the manipulated vertices representing a change of distance, pose, expression or any other 3D manipulation. Given these two sets, we find a 2D image warp to produce the output image. However, some points are "occluded" for the purpose of the warp. For example, our head model includes back-facing areas, but such areas move in the direction opposite from the front-facing areas when changing camera distance. Therefore we remove occluded vertices before calculating the warp.

Given a sparse set of before and after points, we need to extrapolate the vector field to the entire image. We use triangulation-based cubic interpolation to get an initial estimate of the dense vector field. Although correct in 3D, strong discontinuities in the vector field may cause artifacts in the output. Consider, for example, an extreme rotation of the head. Cheek points that had the same x and y values (but different z values) need to be stretched to different x locations, causing a shear. Note that the vector field in this case is correct in 3D, but cannot be approximated well by a 2D warp. Therefore, we smooth out large gradients in the warp field, as follows: We first replace all values outside the face region with a smooth interpolation of the valid values, by computing the discrete Laplacian and solving a Neumann boundary condition. We next blur the vector field by convolving with a disk with radius $\frac{1}{20}$ of the photo diagonal. Finally with the smooth warp field we use reverse mapping to calculate the origin of each pixel in the output image. We use linear interpolation since it is simple, fast, and produces satisfactory results. Figure 4 shows a breakdown of these steps.

4 Evaluation

In this section we evaluate our method. Section 4.1 demonstrates the importance of different parts of the pipeline by showing results with various stages disabled. Section 4.2 and 4.3 compare our results against ground truth photos of synthetic and real heads, respectively, taken at known distances. Section 4.4 discusses the impact of our warp on the background of the portrait.

4.1 Pipeline Evaluation

For a face with neutral expression and common proportions, a single average head model might suffice (Section 4.3). However, when the input image is expressive, it is important to use the full face model. Figure 5 shows results using an average face model, instead of optimizing a specific identity and expression to our image. Clearly a single face cannot be a catch-all solution, resulting in artifacts due to bad alignment.

Fiducial point based matching from a 3D head to a 2D image is sensitive to the choice of landmarks. Many existing works use 66 standard points spread across the chin, eyebrows, nose, eyes and lips [Yang et al. 2011; Yang et al. 2012a; Cao et al. 2014b]. This choice is motivated mostly by ease of recognition. When fitting a face model and manipulating only the face internals, such landmarks might suffice. However, we found that full head manipulation, especially one where the camera location is changed, requires more fiducial points. Adding three points (top-of-head and ears) leads to significantly better results for our scenario. Figure 6 shows failure cases when these additional landmarks are not used.

Our warping procedure (Section 3.5) uses well established methods (such as triangulation and sampling). However, we use a specific

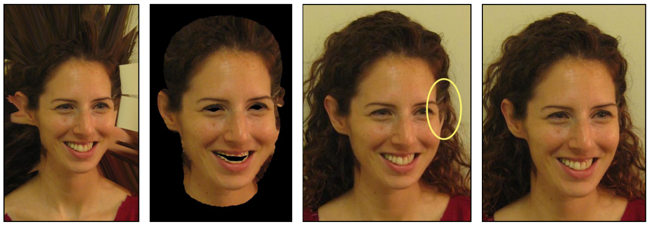
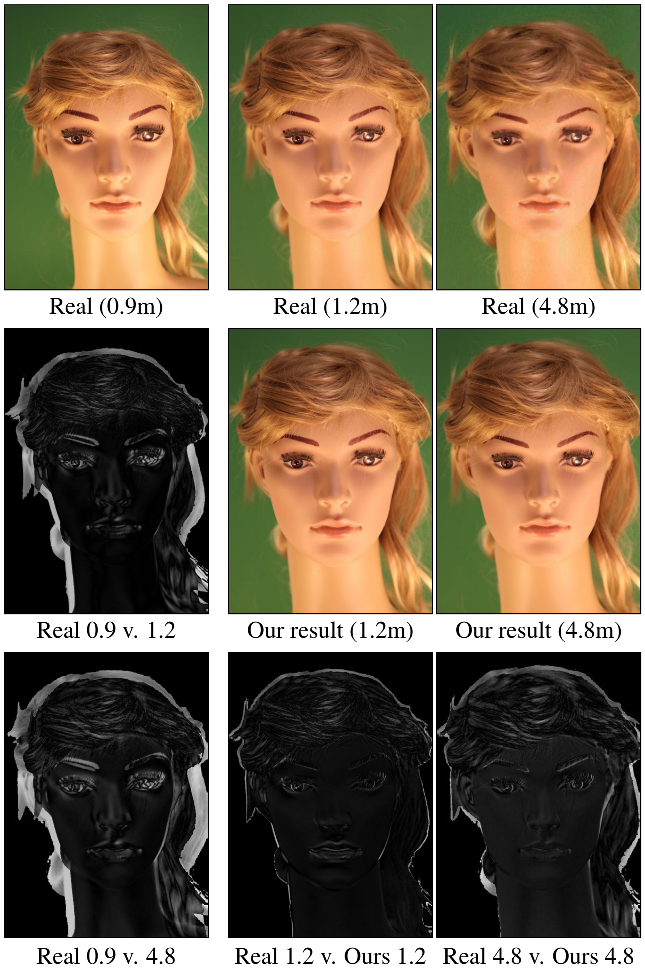


Figure 7: *Warp comparison. L-to-R: PiecewiseLinearTransformation2D (Matlab), LocalWeightedMeanTransformation2D (Matlab), our result without smoothing, our result. Input image in Figure 13.*



Real (0.9m) | Real (1.2m) | Real (4.8m)

Real 0.9 v. 1.2 | Our result (1.2m) | Our result (4.8m)

Real 0.9 v. 4.8 | Real 1.2 v. Ours 1.2 | Real 4.8 v. Ours 4.8

Figure 8: *Ground truth evaluation. We use a mannequin to make sure no pose or expression changes occur in the ground truth images. Our results closely match the ground truth, both in overall head shape and the location of internal face features.*

procedure with added steps to reduce potential artifacts. Figure 7 compares our warping results to results obtained by standard image warping techniques.

4.2 Synthetic Heads

Numerically evaluating our method is hard. Photos of real people taken from different distances at different times have slight variations in illumination, expression and pose, thus the "ground truth" image does not match exactly a warped version of the input. To tackle this issue we perform evaluation on two types of data: mannequin heads and real people. The mannequin heads provide a controlled environment, for which we can get accurate ground truth.

Figure 8 shows several results with mannequin heads. Our input image is taken from a distance of 90cm. We warp it to simulate a range of distances between 120cm and 480cm. We compare each warped result to the ground truth by calculating the absolute difference of the gray-scale pixel values (black indicates equality; white indicates the largest difference.) Note that the method manages to simulate both the head shape and the location of internal features such as eyes and ears.

4.3 Real Heads

To obtain a similar evaluation of real-world cases, we use the CMDP dataset [Burgos-Artizzu et al. 2014], which contains portraits of people captured from a few controlled distances. We evaluate the process of changing the camera distance from an image shot at 60cm to 480cm and then compare to a real image captured at that distance. This is the harder direction of manipulation, as features in the close-up image (e.g. ears) are not always visible.

However, a naïve pixel difference will not suffice here, due to slight pose, expression and illumination changes in the ground truth images. Therefore to compare two images we:

1. Register the images using a rigid transform, to avoid penalizing simple rotations or translations.
2. Use Large Displacement Optical Flow [Brox and Malik 2011] to calculate optical flow between the images.
3. Mask out the background regions, since we are only interested in the head warp.
4. Calculate the median optical flow magnitude in the head region. To normalize, we multiply by 100 / image diagonal.

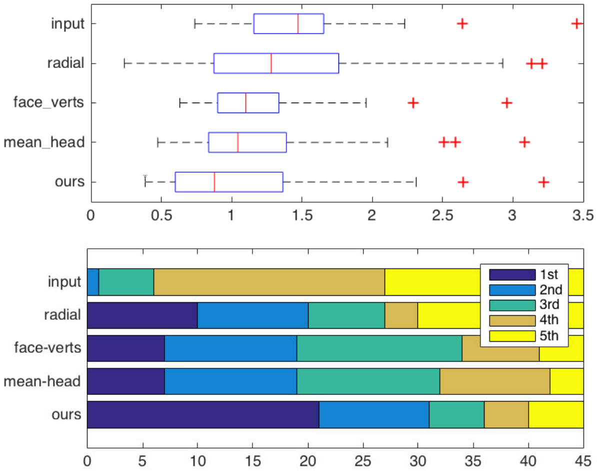


Figure 9: *Score comparison with the 45 images from CMDP dataset that have EXIF data. We warp images taken from 60cm to appear like 480cm away, comparing to ground-truth photos from that distance. Energies are shown for: (1) input images, (2) radial distortion, (3) warping with a face only (4) warping using an average head, and (5) our full model. Top: median values of our energy function, where lower is better (Section 4.3). Boxes are 25th to 75th percentile and red line is median of medians. Bottom: we rank each method vs. all others, counting how often a method had each rank. Our method outperforms all others.*

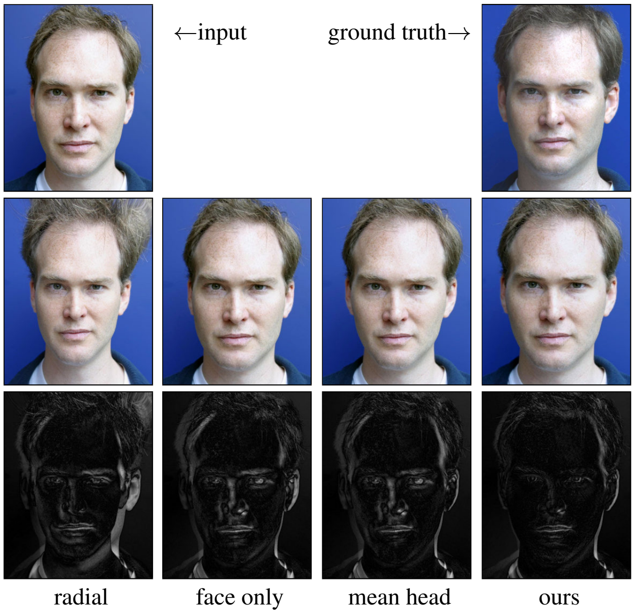


Figure 10: *Comparing methods. Top: input and ground truth at target distance. Middle compares alternate approaches to ours: optimal radial distortion, fiducials on face only, mean head model, and ours. Bottom: visualizing error from ground truth.*

Figure 9 numerically compares our method to the following four alternatives: 1) Compute an optimal radial distortion correction given known ground truth (giving maximal advantage to this method to test its potential); 2) Use only the fiducials and the vertices of the face to drive the warp, simulating the fitting done by methods like [Yang et al. 2011; Cao et al. 2014b] and many others; 3) Fit an average head instead of the multi-linear deformable model and warp using our method, representing methods like [Kemelmacher-Shlizerman and Basri 2011; Kemelmacher-Shlizerman et al. 2011; Hassner et al. 2015] that use a single model; We use a mean full head model, averaged from the dataset in [Cao et al. 2014b] as apposed to just a face model as was done in previous methods, to explore a full potential of this approach for our task. 4) Fit our full model. Figure 10 shows representative results.

Figures 5 and 13 show our results for input images with a non-neutral pose and expression. We compare against a static model, showing that a deformable model is important.

4.4 Background Preservation

Most of the examples shown so far had a rather uniform background that might hide warp artifacts if they exist. While some works in the area are limited to these types of inputs, we would like to have a system that works in the wild. Moreover, we cannot expect the user to mask the area around the head, since we aim for a fully automatic method.

Thus, we require minimal distortion in the background, which we achieve by using a 2D warping approach (Section 3.5) rather than a 3D texture mapping approach requiring perfect head segmentation. In Figure 12 we show several examples of our warp result on noisy backgrounds.

4.5 Runtime

Our method is implemented in Matlab, and can be further optimized. Typical runtime is around 5 seconds to fit the model to the input image, and less than 1 second for warp field generation and warp calculation. To support real-time interactivity, we also created a WebGL viewer that can adjust warps on the fly (Section 5.3). We pre-calculate warp fields for a few predefined distances or other parameters such as pitch and yaw. The pre-calculation takes 3 seconds for 4 samples of the distance parameter. After pre-computing these warp fields, the interpolated warp is rendered in the web browser in real time (more than 60 FPS).

5 Applications

Our primary application is to adjust camera distances (Section 5.1). We also discuss other applications including stereoscopic portraits (Section 5.2) and pose adjustment (Section 5.3).

5.1 Distance Correction

Our main motivating application is to adjust camera distance in portraits. Figure 11 shows distance manipulation results for seven subjects from the CMDP dataset. In each case the 60cm portrait was warped to match the 480cm one, and vice versa, so they can be compared to ground truth. Note that the changes are subtle but noticeable. Moreover, these changes are more prominent when the subject is known (yourself, family or a friend). We refer the reader to the accompanying video as well as the interactive viewer in the supplemental materials for more examples.

All the above results are from a controlled dataset, for comparison to ground truth. However, our system also works well on images "in the wild." Figure 12 shows distance manipulation on real images tagged as #selfie on Twitter and Flickr. Our system works across a variety of expressions and poses despite cluttered backgrounds and complex lighting. Figure 13 and Figure 5 further illustrate the robustness of our method to exaggerated expressions and poses. More examples are in the supplementary materials.

5.2 Headshot Stereoscopy

We can create stereoscopic images using our framework. Given the distance from the subject and the average human interpupillary distance, we can modify the viewpoint to obtain two new images — one for each eye. Those images can then be displayed on devices such as VR headsets. Figure 14 shows 3D anaglyphs automatically created from 2D photos using this approach. These can be viewed using a standard pair of red/cyan glasses (red eye left).

5.3 Other Applications

Our 3D fitting pipeline is based on the multi-linear morphable model framework. As such, we can replicate some of the face manipulation tasks shown in previous work using similar models [Vlasic et al. 2005; Yang et al. 2011; Cao et al. 2014b]. These include pose and expression manipulation, and animating a moving face from a still image (see Figure 1f and the accompanying video).

Our WebGL based user interface supports interactive photo manipulation. The user is presented with the image and sliders to manipulate camera distance and head pose (Figure 15). We calculate warp fields for predefined parameter values (4 distances, 5 pitch values, 5 yaw values). When the user selects a specific parameter combination, we use trilinear interpolation to generate a warp field. Then, we use the warp field to create the output via reverse mapping. Output images are rendered at real-time rates, allowing users to adjust parameters to their liking.

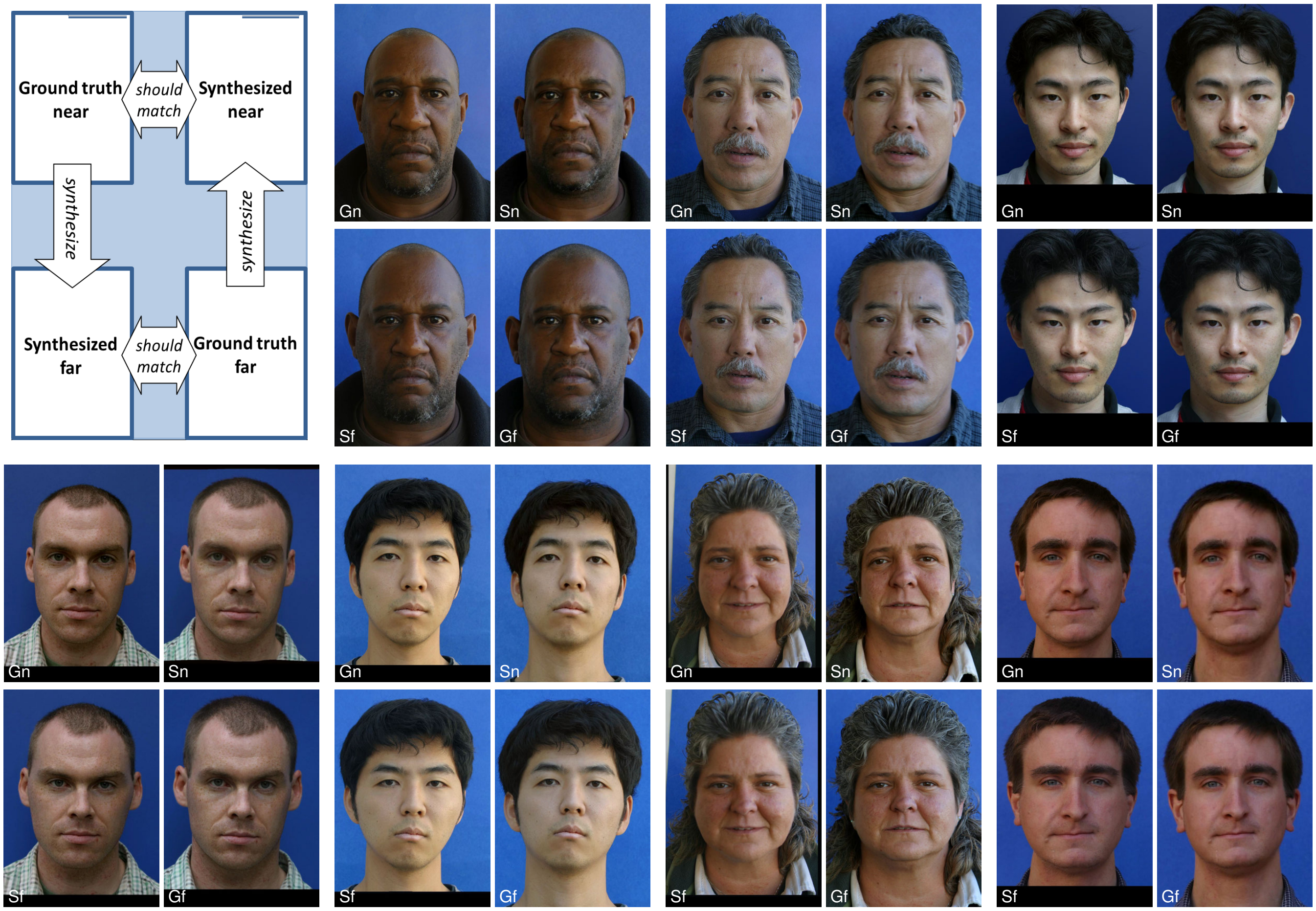


Figure 11: *Fixing/generating selfies. Legend in upper-left corner shows arrangement of each quadruplet. Input ground truth "near" photos were taken at 60cm, whereas "far" photos were taken from 480cm (CMDP Dataset [Burgos-Artizzu et al. 2014]). Synthetic images were warped from near to far and vice versa, and are arranged and color matched for ease of comparison. When evaluating, compare the head shape and the location of internal face features. These results are selected from a larger set available in supplemental materials.*

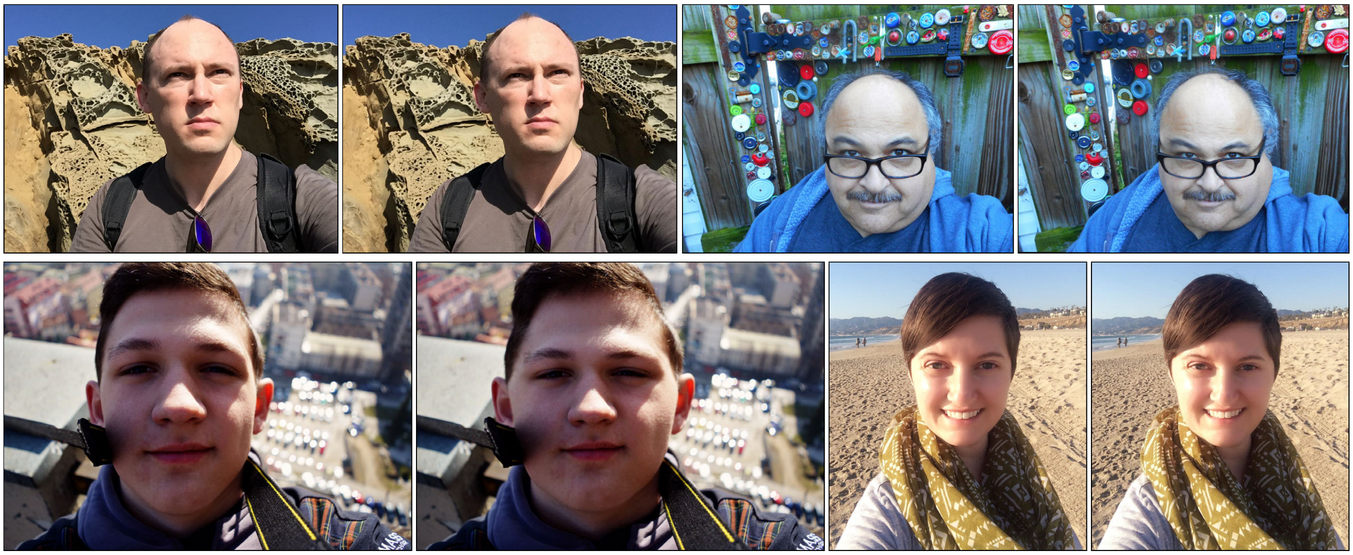


Figure 12: *In-the-wild selfie correction. We use Twitter and Flickr images tagged as #selfie. Left: original, right: our result. Results shown for various head shapes. Background remains largely undistorted.* ©*Flickr users Justin Dolske, Tony Alter, Omer1r, and Christine Warner Hawks.*

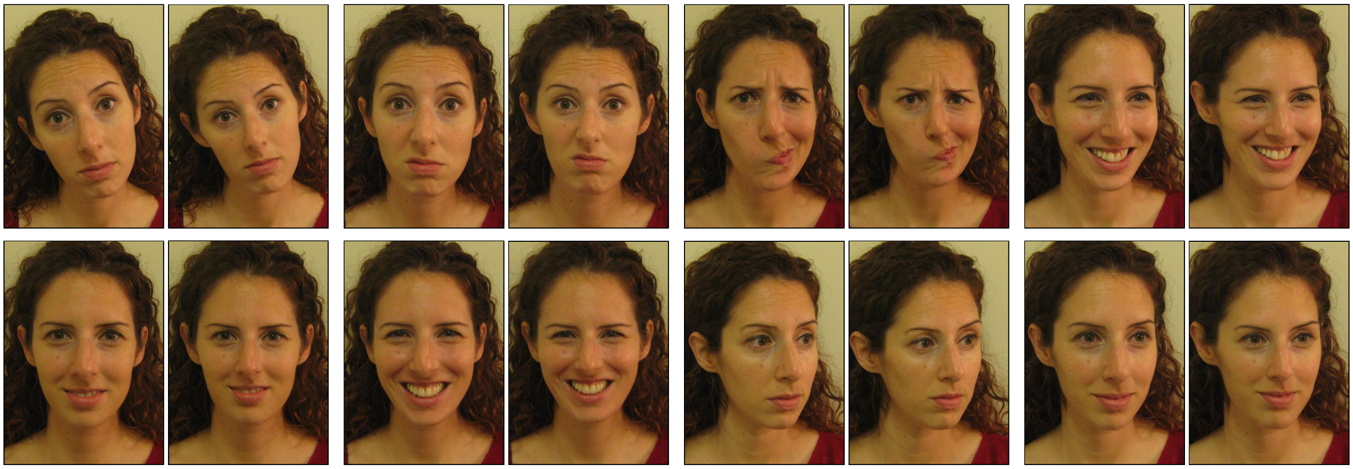


Figure 13: *Manipulating distances for expressive faces. Each pair contains: original (60cm, left image), our output (480cm, right image).*

6 Limitations and Future Work

We present a unified framework for altering the camera and subject pose in a portrait photo. This method can be used to improve selfies, make a subject look more approachable or adapt the camera distance to match a different shot for compositing. We display results for various scenarios and compare with ground truth data. Our editing operations remain in the realm of "plausible" – they do not create new people, rather they show the same people under different viewing conditions. In that sense, they are the post-processing equivalent of a portrait photographer making a different decision about the composition. Our framework also supports creating stereoscopic views from portraits and video, as well as making video with apparent camera and subject motion from a still portrait. More results, video and demos may be seen on our project page `http://faces.cs.princeton.edu/`.

Our approach has several weaknesses that suggest opportunities for future work. First, the pipeline relies on a good fit between input and model, and if the fit fails, the results will be distorted. While our optimization has proved robust in many cases, occasional failures remain. Future approaches might build on larger, more varied head shape datasets, or rely on 2.5D sensor data emerging in new camera rigs. Second, we only warp the data that exists in the original image. This produces convincing results in many cases, but will not handle significant disocclusions such as can arise from significant head rotations. One way to address this might be by filling missing regions via, e.g., texture synthesis with a strong face prior [Hassner et al. 2015]. Third, the way we currently treat hair is by a smooth extrapolation of the warp field outside of the head region. This is often insufficient, and could be improved with a specialized hair model. Fourth, our method does not handle eye gaze correction and extreme expression change which may be desired in some scenarios. One could experiment with existing techniques for editing gaze [Giger et al. 2014] and expression [Yang et al. 2011]. Finally, while the accompanying video shows a couple speculative applications for video (stereoscopic video and a "moving portrait") a proper investigation of such applications remains for future work.

7 Acknowledgments

We thank the anonymous reviewers for helpful comments, the researchers and photographers (especially Anton Orlov) that generously shared their photos and datasets with us, and our photo subjects for their extraordinary patience. This research was funded in part by Adobe and by a Google Graduate Fellowship.

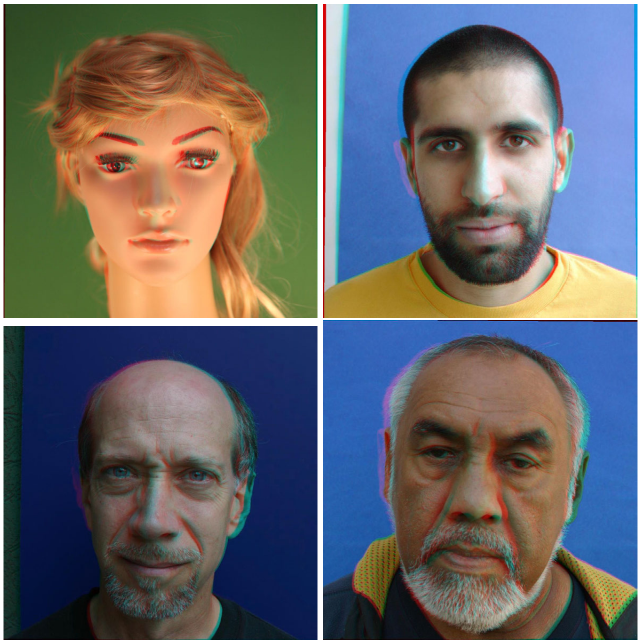


Figure 14: *3D anaglyphs created from a single image. To view, wear red-cyan 3D glasses and zoom the image to fill your screen.*

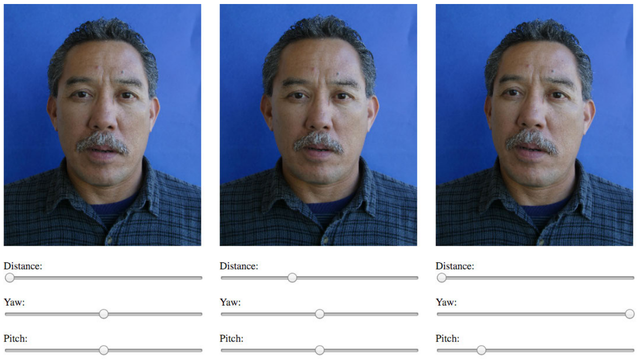


Figure 15: *Interactive editing, in which sliders control the resulting warp field. (See video and demo on the project page.)*

References

ALEXANDER, O., ROGERS, M., LAMBETH, W., CHIANG, M., AND DEBEVEC, P. 2009. The digital Emily project: Photoreal facial modeling and animation. In *ACM SIGGRAPH 2009 Courses*.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, 187–194.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. *ACM Trans. Graph. 29*, 4 (July), 41:1–41:10.

BROX, T., AND MALIK, J. 2011. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 33*, 3 (Mar.), 500–513.

BRYAN, R., PERONA, P., AND ADOLPHS, R. 2012. Perspective distortion from interpersonal distance is an implicit visual cue for social judgments of faces. *PLoS ONE 7*, 9 (09).

BURGOS-ARTIZZU, X. P., RONCHI, M. R., AND PERONA, P. 2014. Distance estimation of an unknown person from a portrait. In *European Conference on Computer Vision (ECCV)*. Springer, 313–327.

CAO, C., WENG, Y., LIN, S., AND ZHOU, K. 2013. 3d shape regression for real-time facial animation. *ACM Trans. Graph. 32*, 4 (July), 41:1–41:10.

CAO, C., HOU, Q., AND ZHOU, K. 2014. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph. 33*, 4 (July), 43:1–43:10.

CAO, C., WENG, Y., ZHOU, S., TONG, Y., AND ZHOU, K. 2014. FaceWarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics 20*, 3, 413–425.

COLEMAN, T. F., AND LI, Y. 1996. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization 6*, 2, 418–445.

COOPER, E. A., PIAZZA, E. A., AND BANKS, M. S. 2012. The perceptual basis of common photographic practice. *Journal of Vision 12*, 5, 8.

DECARLO, D., METAXAS, D., AND STONE, M. 1998. An anthropometric face model using variational techniques. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, 67–74.

GIGER, D., BAZIN, J.-C., KUSTER, C., POPA, T., AND GROSS, M. 2014. Gaze correction with a single webcam. *IEEE International Conference on Multimedia & Expo*.

HASSNER, T., AND BASRI, R. 2006. Example based 3d reconstruction from single 2d images. In *Beyond Patches Workshop at IEEE CVPR'06*.

HASSNER, T., HAREL, S., PAZ, E., AND ENBAR, R. 2015. Effective face frontalization in unconstrained images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

HASSNER, T. 2013. Viewing real-world faces in 3D. In *International Conference on Computer Vision (ICCV)*.

KEMELMACHER-SHLIZERMAN, I., AND BASRI, R. 2011. 3d face reconstruction from a single image using a single reference face shape. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 33*, 2 (Feb), 394–405.

KEMELMACHER-SHLIZERMAN, I., AND SEITZ, S. M. 2011. Face reconstruction in the wild. In *International Conference on Computer Vision (ICCV)*.

KEMELMACHER-SHLIZERMAN, I., SHECHTMAN, E., GARG, R., AND SEITZ, S. M. 2011. Exploring photobios. *ACM Trans. Graph. 30*, 4 (July), 61:1–61:10.

ORLOV, A., 2016. Selecting a portrait lens with correct focal length. Accessed 2016-01-15: http://petapixel.com/2016/01/04/selecting-a-portrait-lens-with-correct-focal-length/.

PERONA, P. 2007. A new perspective on portraiture. *Journal of Vision 7*, 992–992.

PERONA, P. 2013. Far and yet close: Multiple viewpoints for the perfect portrait. *Art & Perception 1*, 1-2, 105–120.

SARAGIH, J. M., LUCEY, S., AND COHN, J. 2009. Face alignment through subspace constrained mean-shifts. In *International Conference on Computer Vision (ICCV)*.

TUCKER, L. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika 31*, 3, 279–311.

VALIND, E. 2014. *Portrait Photography: From Snapshots to Great Shots*. Pearson Education.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIĆ, J. 2005. Face transfer with multilinear models. *ACM Trans. Graph. 24*, 3 (July), 426–433.

WEISE, T., LEIBE, B., AND VAN GOOL, L. 2007. Fast 3d scanning with automatic motion compensation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 861–868.

YANG, F., WANG, J., SHECHTMAN, E., BOURDEV, L., AND METAXAS, D. 2011. Expression flow for 3d-aware face component transfer. *ACM Trans. Graph. 30*, 4 (July), 60:1–60:10.

YANG, F., BOURDEV, L., SHECHTMAN, E., WANG, J., AND METAXAS, D. 2012. Facial expression editing in video using a temporally-smooth factorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 861–868.

YANG, F., SHECHTMAN, E., WANG, J., BOURDEV, L., AND METAXAS, D. 2012. Face morphing using 3d-aware appearance optimization. In *Proceedings of Graphics Interface (GI'12)*, 93–99.