

Hair Interpolation for Portrait Morphing

Yanlin Weng¹ Lvdi Wang² Xiao Li³ Menglei Chai¹ Kun Zhou¹

¹Zhejiang University ²Microsoft Research Asia ³University of Science and Technology of China

Abstract

In this paper we study the problem of hair interpolation: given two 3D hair models, we want to generate a sequence of intermediate hair models that transform from one input to another both smoothly and aesthetically pleasing. We propose an automatic method that efficiently calculates a many-to-many strand correspondence between two or more given hair models, taking into account the multi-scale clustering structure of hair. Experiments demonstrate that hair interpolation can be used for producing more vivid portrait morphing effects and enabling a novel example-based hair styling methodology, where a user can interactively create new hairstyles by continuously exploring a “style space” spanning multiple input hair models.

1. Introduction

Image and shape morphing research dates back decades [Wol98, LV98, ACOL00], among which *portrait morphing* [Bic96, ZVR07, YSW*12] is arguably the most extensively studied and popularized, because of people’s innate interest in faces. However, as far as we know, no existing methods explicitly handle *hair*. Instead they use cross-dissolve solutions, which are likely to cause ghosting and other visual artifacts.

In this paper we tackle the problem of *hair interpolation* by utilizing recent advances in single-image 3D hair modeling [CWY*12, CWW*13]. Being able to generate a 3D strand-based hair model from a given portrait allows us to concentrate on a core question: given two such hair models, how can we generate a sequence of intermediate 3D hair models that naturally transform from one input to another?

Although judging the quality of a *general* morphing sequence is subjective, one often wants the sequence to be *temporally smooth* and at any time instance t , *similar to both the source and the target* with the similarities intuitively proportional to $1 - t$ and t respectively. However, as evidenced by simple cross-dissolving effects, just satisfying these two conditions does not promise a natural looking portrait, or even aesthetically pleasing results. One usually needs to consider domain-specific constraints as well. For human hair, we would expect that at any instance the morphing result still looks like a reasonable hairstyle with vivid strands. Furthermore, traditional morphing problems typically consider manifold surfaces where correspondences can be relatively easy to establish. Human hair can contain hundreds of thou-

sands of thin 3D curves densely packed in a volume. The number of strands, the distribution of hair roots, the shape, and the appearance of each strand may vary dramatically from person to person. Even though we have two plausible 3D hair models at hand, there are hardly any proper strand-to-strand correspondences between them. Simply building correspondences according to the root positions on the scalp surface, as an example, will generate unsatisfactory results.

We formulate the strand correspondence as a many-to-many mass transportation problem, and propose an efficient solution that automatically handles the differences of the given input models. By further taking into account the hierarchical clustering of hair strands, our algorithm is able to morph strands coherently and ensure any intermediate hair model is still visually plausible. Our experiments show that the proposed hair interpolation algorithm can be easily integrated into existing 3D-aware face morphing frameworks [YSW*12], producing much more appealing results (Figure 1). To take it one step further, we experiment with *N-way hair interpolation*, where more than two hair models serve as input, and effectively span a “hairstyle space.” We present an interactive system that allows the user to explore this space (i.e., generate a weighted intermediate hair model) in *real time*. We have surprisingly discovered that such a system actually serves as a novel and powerful example-based hair/face synthesis tool (Figure 2).

2. Related Work

Morphing. Traditional image and shape morphing are often achieved by first building a warp function from corresponding features, and then performing some value interpolation



Figure 1: Comparison of different hair interpolation approaches. From left to right: (a) source images, (b) feature-based morphing [BN92], (c) MLS-based morphing [SMW06], (d) image melding [DSB*12], (e) opacity-based strand interpolation, (f) length-based strand interpolation, and (g) our proposed method.



Figure 2: A random sampling of the portraits generated from three input portrait images (top row).

between warped sources [BN92, Wo198, LV98, ACOLO0]. By recovering the 3D face geometry and combining the expression flow [YWS*11], Yang et al. [YSW*12] propose an automatic algorithm for morphing the face region in portrait images. But these methods do not handle hair explicitly due to the extreme difficulty of establishing a proper strand correspondence in the 2D image space.

Regenerative Morphing [SRAIS10] and Image Melding [DSB*12] are two patch-based techniques that are able to generate a smooth interpolation automatically even between two inconsistent images. Unfortunately their results cannot retain visual fidelity and physical plausibility of hair in intermediate frames.

Bonneel et al. [BPPH11] have introduced a method to interpolate two distributions by decomposing each continuous function into discrete components and solving a mass transport problem. Inspired by their work, we formulate the strand correspondence problem in a similar way, but also take into account the multi-scale clustering of hair.

Hair Modeling. The modeling/styling, rendering, and simulation of hair are considered very challenging [WBK*07].

There exist several image-based hair capture techniques that can reconstruct a real hairstyle by taking multiple photographs from different angles [PCK*08, HZW12, LLR13] or with different focal distances [JMM09]. In order to extract hair strands from a general portrait captured in an uncontrolled environment, we adopt the approach proposed by Chai et al. [CWW*13].

Wang et al. [WYZG09] have presented a method to analyze the hierarchical clustering structure of a “raw” hair model. Analogous to texture synthesis [WLKT09], their method also allows the user to synthesize new hairstyles from existing exemplars by transferring geometric details among hair clusters. On the other hand, our method allows the user to continuously interpolate multiple hair models in real time, whereas their method does not.

3. Algorithm

We define the problem of *hair interpolation* as: given a source hair model \mathcal{S}_0 and a target model \mathcal{S}_1 , to generate a sequence of intermediate hair models $\{\mathcal{S}_t \mid t \in (0, 1)\}$ that transition from \mathcal{S}_0 to \mathcal{S}_1 . We follow the same notation used by Chai et al. [CWW*13] where a strand ξ_i in \mathcal{S} is a 3D curve described by a list of vertices ordered from the hair root to the tip, and the j -th vertex of ξ_i is denoted as $\xi_{i(j)}$.

We start by introducing an automatic algorithm for computing a many-to-many strand correspondence between two hair models (§ 3.1), which can be used to generate intermediate strands (§ 3.2). We then explore two extensions to this basic algorithm: a *multi-scale strand correspondence* algorithm that takes into account the hierarchical cluster structure of hair (§ 3.3), and *N-way interpolation*, which allows more than two different hairstyles to be smoothly interpolated in real time (§ 3.4).

3.1. Many-to-Many Strand Correspondence

Inspired by Bonneel et al.’s recent work on displacement interpolation [BPPH11], we formulate the establishment of strand correspondence between hair models \mathcal{S}_0 and \mathcal{S}_1 as a classical *Monge-Kantorovich transportation problem*. An intuitive interpretation is to imagine a bipartite graph where

each strand $\xi_i \in \mathcal{S}_0$ corresponds to a source node, each strand $\xi_j \in \mathcal{S}_1$ corresponds to a target node, and each edge has a cost $d(\xi_i, \xi_j)$. Each strand ξ is also associated with a positive weight (mass), $w_s(\xi)$, which is proportional to its length $|\xi|$. Without any loss of generality, we assume the total weights of the two models are equal. The challenge is then to find a flow f that transports all the weights from source nodes to target nodes while minimizing the total transportation cost. This can be expressed as

$$\arg \min_f \sum_{i,j} d(\xi_i, \xi_j) \cdot f(i, j), \quad (1)$$

subject to $f(i, j) \geq 0$, $\sum_i f(i, j) = w_s(\xi_j)$, and $\sum_j f(i, j) = w_s(\xi_i)$.

We define the cost $d(\xi_i, \xi_j)$ in such a way that it measures the geometric dissimilarity between two strands. After aligning the source and the target hair models with a rigid transformation estimated from the 3D head models, we calculate a high dimensional *feature vector* for each strand as described by Wang et al. [WYZG09]. The cost between a pair of strands is then defined as the \mathcal{L}^2 distance between their corresponding feature vectors: $d(\xi_i, \xi_j) = \mathcal{L}^2(\mathbf{u}_i, \mathbf{u}_j)$.

To solve the minimum cost flow problem in Equation (1), we use the modified network simplex algorithm proposed by Bonneel et al. [BPPH11]. The result is a many-to-many pairing between source and target hair strands.

3.2. Morphing Sequence Generation

Given a pair of corresponding strands ξ_i and ξ_j , we would like to generate a sequence of intermediate strands $\{\xi_t\}$ that transform from ξ_i to ξ_j smoothly.

We perform simple linear interpolation between the curve vertices so that the user can explore the morphing space in real time. We resample both ξ_i and ξ_j with the same number of vertices (48 for all the results in this paper). At any time instance t , the l -th vertex of the intermediate strand, denoted as $\xi_{t(l)}$, is a linear combination of the respective source and target vertices: $\xi_{t(l)} = (1-t)\xi_{i(l)} + t\xi_{j(l)}$, where $0 \leq t \leq 1$. The vertex color is interpolated in the Lab color space.

Note that each non-zero $f(i, j)$ corresponds to a strand in \mathcal{S}_t , which means during morphing, the number of intermediate strands might be higher than both the source and target strands (but no higher than $|\mathcal{S}_0| + |\mathcal{S}_1| - 1$ [Flo53]). This is actually *desirable* as it naturally models the characteristic splitting and grouping of strands observed in real hair.

Although more advanced interpolation algorithms (e.g. [SG95, ACOL00]) can also be used for our purpose, we have found that as long as we have defined good strand correspondence, even a simple linear interpolation scheme can yield satisfactory morphing results in most cases. It also allows us to conveniently store only the source and the target hair models and evaluate any intermediate model in real time.

3.3. Multi-scale Strand Correspondence

In § 3.1 we consider each strand as an independent curve. In real hair, due to friction, electrostatic force, or the use of cosmetic products, neighboring strands tend to be grouped into *wisps* and behave *coherently*. In many hairstyles such wisps further form larger-scale *clusters* (e.g., those strands combed in the same direction in a middle parting hairstyle). Finding correspondences globally at the strand level without respecting these important structures could result in visually unpleasant artifacts, even though the transportation cost defined in Equation (1) is minimized.

The above observation motivates us to propose a multi-scale extension that takes into account the cluster structure of the hair models. Instead of finding global correspondence, it decomposes the problem into multiple cluster-wise sub-mass transportation problems and solve them independently.

To extract the cluster structure from a given hair model, we adopt the geometric analysis method from Wang et al. [WYZG09] to build a multi-level hair hierarchy for each \mathcal{S}_0 and \mathcal{S}_1 from the bottom up. The basic idea is to group strands in one hierarchical level that are both *geometrically similar* and *spatially adjacent* into the same cluster, and calculate a representative strand for each cluster as one strand in the next (coarser) hierarchical level.

Specifically, we denote the h -th level of \mathcal{S} as $\mathcal{S}_{(h)}$, the p -th cluster of $\mathcal{S}_{(h)}$ as $\mathcal{S}_{(h)}^p$, its representative strand as $\bar{\xi}_p$, and any strand belonging to $\mathcal{S}_{(h)}^p$ as ξ_i^p . The weight of the representative strand, $w_s(\bar{\xi}_p)$, is defined as the sum of all the strands in cluster p .

We then solve the strand correspondence in a top-down fashion. At the h -th hierarchical level, we calculate the minimum cost flow from a source cluster $\mathcal{S}_{0(h)}^p$ to a target cluster $\mathcal{S}_{1(h)}^q$ if and only if there exists a non-zero flow from $\bar{\xi}_p$ to $\bar{\xi}_q$, i.e., the two respective representative strands at the $(h+1)$ -th level. The correspondence at the coarsest level is simply calculated as in § 3.1, since each hair model at this level has only one cluster.

This multi-scale, “cluster-wise” calculation of strand correspondence has multiple benefits. In terms of *quality*, 1) flows are usually calculated only between *compatible* clusters whose representative strands at the coarser level already have a low cost flow in-between. 2) Strands in the same cluster deform more *coherently*. Therefore, intermediate hair models look more natural and contain less visual artifacts (see Figure 3 for a comparison). In terms of *performance*, 3) by partitioning n strands into k clusters, the average-case complexity of the strand correspondence problem is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^2/k)$, and these k sub-problems can be solved parallel to each other. Overall, the multi-scale approach generally yields better interpolation results while running one to three orders of magnitude faster in practice when compared to the single-scale version.

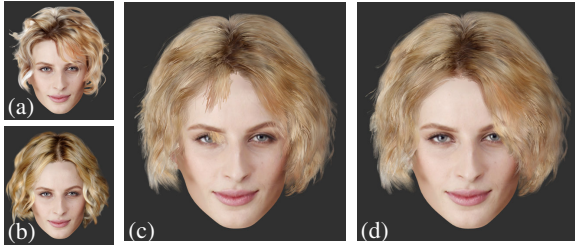


Figure 3: Morphing from (a) to (b) using (c) the single-scale approach and (d) the multi-scale version. Precomputing the correspondence takes 40 min. for (c), and 20 sec. for (d).

3.4. N -Way Hair Interpolation

For N input hair models, where $N \geq 3$, directly solving a multi-index optimal mass transport problem becomes computational prohibitive when N is large [Sha77]. We instead develop a pairwise approximation that extends the bi-partite graph described in § 3.1 to an N -partite one, and calculates the $N(N - 1)/2$ pairwise multi-scale correspondences between every two models.

In the case of two-way interpolation, every edge in the graph with a positive flow indicates a pair of morphable hair strands that can be interpolated to generate an intermediate strand during morphing. But when $N > 2$, we must carefully select the morphable sets of input strands, so that any N strands from each of the input models are said to be morphable (i.e., correspond to an intermediate strand) if and only if they are *connected by a subtree* of the N -partite graph with positive edge flow. We use a greedy algorithm to find such subtrees with ascending flow cost until either no valid subtree exists or each strand belongs to at least one subtree.

Once the sets of morphable strands are determined, we can calculate any intermediate strand as a weighted average of the relevant input strands using barycentric coordinates.

4. Results and Discussion

We evaluated our hair interpolation algorithm on a variety of portrait images. Our method generally handles different hair shapes and numbers of strands well without any user intervention. The hair modeling process takes around one minute to produce a model of a few tens of thousands of strands, with all the parameters being set according to the original paper [CWW*13]. Calculating the strand correspondence between two hair models takes tens of minutes using the single-scale algorithm, or less than twenty seconds using the multi-scale extension on typical 3-level hair hierarchies, where the ratio of strand numbers between succeeding levels is set to 100 : 1 (Figure 3). Although the user is free to choose the number of levels depending on hair’s geometrical complexity, we have found that three levels work well for all the input in this paper.

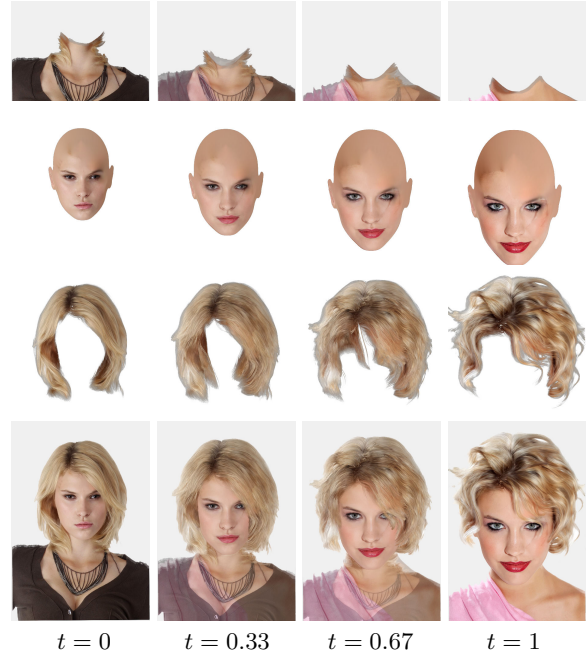


Figure 4: Different components in one of our portrait morphing sequences. From top down: background and body layers, head, hair, and final composite.

4.1. Portrait Morphing

Given that Chai et al.’s hair modeling approach [CWW*13] also reconstructs a textured 3D face mesh based on the morphable face model [BV99] and an estimated transformation, we can easily implement 3D-aware face morphing: after aligning the two source meshes to an intermediate transformation, we linearly interpolate the corresponding mesh vertices and texture colors.

For the remaining parts of the portrait (if there are any), we simply adopt 2D feature-based image morphing [BN92]. These 2D-morphed body/background layers, as well as the 3D-interpolated hair strands and face model are then composed together to yield the final image (Figure 4).

Since to our knowledge there is no other dedicated hair morphing research, we compare our method with 3D-aware face morphing [YSW*12] and several general image morphing techniques, including feature-based morphing [BN92], Moving Least Squares-based morphing [SMW06], and Image Melding [DSB*12], as shown in Figures 1 and 5. As an additional comparison, we have also implemented two naive hair interpolation methods based on the 3D hair models. One is to simply interpolate the global *opacity* of the two models, and the other is to interpolate the *curve length* of strands. The results are shown in Figure 1 (e) and (f) respectively. Please also see the video for dynamic comparisons.

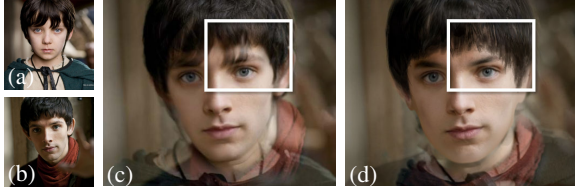


Figure 5: Two input images (a), (b), and the intermediate frames generated using (c) Yang et al.'s method [YSW*12] and (d) our method respectively.

4.2. Interactive Hair Styling

When the user interpolates between more than two hair models using N -way hair interpolation (§ 3.4), these N input models in some sense span a “hairstyle space.” We have developed an intuitive GUI that allows the user to continuously explore this space and design interesting hybrid hairstyles from a few exemplar portraits in real time. To enable intuitive control over the influence from each input model, we adopt barycentric coordinates inside a convex polygon with N vertices as the weights for interpolation. Moreover, we allow the user to control the weights for hair shape, color, and face independently.

Figure 2 shows a number of portraits generated using N -way interpolation from three different hairstyles with random weights for hair shape, color, and face respectively. Figure 6 demonstrates a few samples from a hairstyle space spanned by six portrait images. Please see the video for interactive results.

4.3. Limitations

For now, we are only considering linear interpolation between corresponding strands, which allows intermediate hair models to be efficiently evaluated using the GPU in real time. Accommodating more complex interpolation algorithms, e.g., to minimize inter-strand collision, would be an interesting problem for future research.

Since we do not explicitly handle hair-head collision during interpolation, a small number of intermediate strands may partially intersect with the head, causing artifacts. One way around this is to check the potential vertex morphing paths when calculating the strand correspondence and simply trim such intermediate strands from the morphing sequence.

Interpolating strand colors linearly may result in the loss of visual richness in intermediate strands, partly due to the fact that specular highlights are blended incorrectly. This issue becomes more obvious when the input models vary dramatically in appearance. One possible workaround is to separate each strand’s shading and albedo, and interpolate the former using, e.g., mass transport. But this can make real-time interpolation more difficult.

Note that our approach for strand correspondence is motivated by the intuition that during a “good” morphing sequence, any intermediate hair model should still look like a reasonably realistic hairstyle. The algorithm thus tries to minimize the total vertex displacements of all strands, but does not ensure minimal distances traveled by *hair roots*. Someone may argue that allowing hair roots to drift on the scalp during morphing is counter-intuitive and “biologically impossible.” However, we find in practice it frequently produces more aesthetically pleasing results.

5. Conclusion

We have studied the problem of hair interpolation based on the 3D hair models reconstructed from single images. To this end, we have presented an automatic algorithm based on mass transportation for establishing correspondences and interpolating strands between two hair models, a multi-scale version of the algorithm to better handle strand coherence within hair clusters, and an extension for more than two hair models. As shown by our experiments, explicitly handling hair can result in a more faithful portrait morphing effect. More intriguingly, by interactively exploring the “hairstyle space,” normal users can now potentially utilize the vast amount of image resources on the Internet for *hairstyle designing and modeling*, a field which has traditionally been relegated to a small number of experts.

Acknowledgment

We thank the image owners for letting us use their pictures, and the anonymous reviewers for their helpful comments. This work is partially supported by NSFC (No. 61003145 and No. 61272305).

References

- [ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *Proc. SIGGRAPH '00* (2000), pp. 157–164. 1, 2, 3
- [Bic96] BICHEL M.: Automatic interpolation and recognition of face images by morphing. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition* (1996), pp. 128–135. 1
- [BN92] BEIER T., NEELY S.: Feature-based image metamorphosis. In *Proc. ACM SIGGRAPH* (1992), pp. 35–42. 2, 4
- [BPPH11] BONNEEL N., PANNE M. V. D., PARIS S., HEIDRICH W.: Displacement interpolation using Lagrangian mass transport. *ACM Trans. Graph.* (2011). 2, 3
- [BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH '99* (1999), pp. 187–194. 4
- [CWW*13] CHAI M., WANG L., WENG Y., JIN X., ZHOU K.: Dynamic hair manipulation in images and videos. *ACM Trans. Graph.* 32, 4 (2013). 1, 2, 4
- [CWY*12] CHAI M., WANG L., YU Y., WENG Y., GUO B., ZHOU K.: Single-view hair modeling for portrait manipulation. *ACM Trans. Graph.* 31, 4 (2012), 116:1–8. 1
- [DSB*12] DARABI S., SHECHTMAN E., BARNES C., GOLDMAN D. B., SEN P.: Image melding: Combining inconsistent



Figure 6: Six input hair models ($S_0 \sim S_5$) and 22 new output models generated automatically using our N -way interpolation algorithm. Each output's position in the above layout reflects its relative weights for the corresponding input models.

images using patch-based synthesis. *ACM Trans. Graph.* (2012). 2, 4

[Flo53] FLOOD M. M.: On the Hitchcock distribution problem. *Pacific Journal of Mathematics* 3, 2 (1953), 369–386. 3

[HZW12] HERRERA T. L., ZINKE A., WEBER A.: Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Trans. Graph.* 31, 6 (2012), 146:1–9. 2

[JMM09] JAKOB W., MOON J. T., MARSCHNER S.: Capturing hair assemblies fiber by fiber. *ACM Trans. Graph.* 28, 5 (2009), 164:1–9. 2

[LLR13] LUO L., LI H., RUSINKIEWICZ S.: Structure-aware hair capture. *ACM Trans. Graph.* 32, 4 (2013). 2

[LV98] LAZARUS F., VERROUST A.: Three-dimensional metamorphosis: a survey. *The Visual Computer* 14 (1998), 373–389. 1, 2

[PCK*08] PARIS S., CHANG W., KOZHUSHNYAN O. I., JAROSZ W., MATUSIK W., ZWICKER M., DURAND F.: Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph.* 27, 3 (2008), 30:1–9. 2

[SG95] SEDERBERG T. W., GREENWOOD E.: Shape blending of 2-D piecewise curves. In *Mathematical Methods for Curves and Surfaces* (1995), pp. 497–506. 3

[Sha77] SHARMA J. K.: Extensions and special cases of trantransport problem: a survey. *Digital library of India* 9, (1977), 928–940. 4

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Trans. Graph.* 25, 3 (2006), 533–540. 2, 4

[SRAIS10] SHECHTMAN E., RAV-ACHA A., IRANI M., SEITZ S.: Regenerative morphing. In *Proc. CVPR* (2010). 2

[WBK*07] WARD K., BERTAILS F., KIM T.-Y., MARSCHNER S. R., CANI M.-P., LIN M. C.: A survey on hair modeling: styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (2007), 213–234. 2

[WLKT09] WEI L.-Y., LEFEBVRE S., KWATRA V., TURK G.: State of the art in example-based texture synthesis. In *Eurographics, State of the Art Report, EG-STAR* (2009), Eurographics Association. 2

[Wol98] WOLBERG G.: Image morphing: a survey. *The Visual Computer* 14 (1998), 360–372. 1, 2

[WYZG09] WANG L., YU Y., ZHOU K., GUO B.: Example-based hair geometry synthesis. *ACM Trans. Graph.* 28, 3 (2009), 56:1–9. 2, 3

[YSW*12] YANG F., SHECHTMAN E., WANG J., BOURDEV L., METAXAS D.: Face morphing using 3D-aware appearance optimization. In *Proc. Graphics Interface* (2012). 1, 2, 4, 5

[YWS*11] YANG F., WANG J., SHECHTMAN E., BOURDEV L., METAXAS D.: Expression flow for 3D-aware face component transfer. *ACM Trans. Graph.* 30, 4 (2011), 60:1–10. 2

[ZVR07] ZANELLA V., VARGAS H., ROSAS L. V.: Active shape models and evolution strategies to automatic face morphing. In *Adaptive and Natural Computing Algorithms* (2007), vol. 4432, pp. 564–571. 1