Convex Hull Escape Perturbation at Embedding Space and Spherical Bins Coloring for 3D Face De-identification

Lanston Hau Man Chu *
Department of Electrical & Computer Engineering
hchu34@wisc.edu

Abstract

This paper proposes a **Convex Hull Escape Perturbation (CHEP)** method at Embedding Space to achieve 3D Face De-identification. For better reconstruction of the 3D faces, this paper also proposes the **Spherical Bins Coloring (SBC)** method to reinstate color lost in the SfS process due to change in vertices number. The top κ-accuracies of faces perturbed by CHEP drop significantly when compared to non-perturbed faces.

1 Introduction and Background

Privacy is a great concern in the modern world. People are concerned about whether personal information are being used and disclosed in an authorized manner. As one of the recent examples, the U.S. Census Bureau plans to apply differential privacy to the 2020 census data so as to ensure the individual data to be confidential. According to the information privacy law, Personally identifiable information (PII) are the set of information in a database which can refer to the identity of a personnel. To protect the privacy of individuals, the de-identification process would transform the PII into non-PII, while preserving some utility of the data [29].

However, when it comes to a problem in the study of human faces. It is not easy to conceal the identities of the subjects as the facial shape *per se* reveals to the identity of a person, even though the main focus of the study may be facial expressions and mental states instead of the facial features. In these cases, the subjects' identities still need to be protected as required by some privacy law such as HIPPA or FERPA [7].

Occlusion-based facial de-identification techniques such as faces blurring may work when the data concerned does not refer to the face, e.g. Google Street View, but the utility or information concerned will be completely removed. Therefore, we need another approach to achieve protection. In this paper, we would focus on perturbation-based facial de-identification, which involves the perturbation of an embedding (encoded by a 3-in-1 bundle of deep learning models) and the corresponding reconstruction back into/onto the 3D/2D space. The purpose of this paper is to reconstruct de-identified 3D human faces, but we would project the reconstructed 3D faces $\ddot{F}'$ onto 2D images $F^{(2D)}$ in the evaluation process, which we will discuss further in the subsequent section 2.4.

1.1 Types of Embeddings

For an Black and White (i.e. B&W; non-colorized) image $x_i \in \mathbb{R}^{H \times W}$ containing a human face, it can be encoded as an embedding $(\alpha, \eta, \rho) = e(x)$. (α, η, ρ) refers to the shape, expression and pose of the face accordingly. Shape means the shape of the human face (including all facial features), which reveal to the identity of the individual. Expression refers to facial expression, which

*Source Code: https://github.com/lanstonchu/Face-DeID

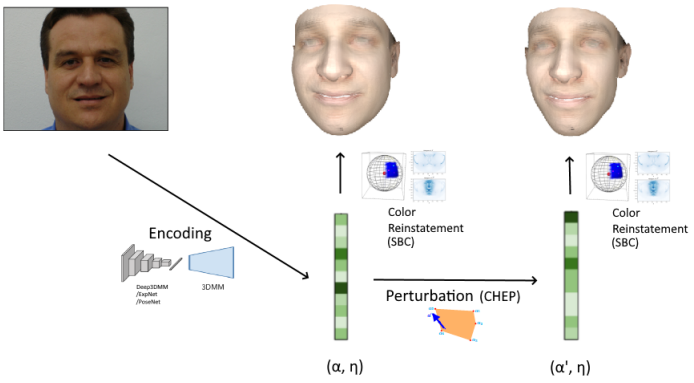


Figure 1: The figure shows the perturbation (CHEP) and the color reinstatement (SBC) processes, which are the main focus and contribution of this paper. The input image will be encoded as an embedding α, which will then be perturbed as $\alpha' = \alpha + \delta$ and reconstructed as a 3D face using Convex Hull Escape Perturbation (CHEP). And then, Spherical Bins Coloring (SBC) would be use to add/reinstate color of the face. The 3D face will then be projected as a 2D image in the evaluation process. The perturbed faces are expected to be less recognizable by machine learning algorithm and human observers than the non-perturbed faces.

is irrelevant to the identity. Pose refers to the rotation (dim-3) and translation (also dim-3) of human face on x_i. For an colorized image $x_i \in \mathbb{R}^{H \times W \times 3}$, it can be further embedded as $(\alpha, \beta, \eta, \rho)$, while β refers to the texture/color embedding. Therefore the embeddings concerned by this paper are:

$$(\alpha, \beta, \eta, \rho) = (shape, color, expression, pose)$$

η and ρ can be ignored in some cases, e.g. tasks of face recognition, where expression and pose are not the main focus/concern of the study. For the encoding process $e(\cdot)$, it can be done by a deep learning model such as Deep3DMM [27], or some traditional approaches based on linear transformations, e.g. eigenfaces determined by PCA (principal component analysis) [30].

For human faces, there are at least two types of embeddings, which are called **coordinates oriented embeddings** and **recognition oriented embeddings** in this paper. Their corresponding encoders are trained based on different goals and their objective functions are of different types.

Coordinates oriented embeddings encoders target to have small coordinate discrepancies between the original input x_i and the output reconstruction at some important points (i.e. landmarks on human faces), and thus the loss function can be formulated as below:

$$\begin{aligned} \mathcal{L}_i^{LM} &= \|v(x_i) - Reconstruct(e(x_i))\|^2 \\ &= \|v(x_i) - Proj2D(Select(S\alpha_i + E\eta_i + \bar{F}))\|^2 \end{aligned} \tag{1}$$

where $v(x_i)$ are the coordinates of the landmarks detected in the image, e.g. by some landmarks detection algorithm such as CLNF [15] with 68 landmarks being detected (in this case $v(x_i) \in \mathbb{R}^{2 \cdot 68}$). The term $S\alpha_i + E\eta_i + \bar{F} \in \mathbb{R}^{3n}$ is the 3D face reconstructed by the 3DMM model, i.e. the coordinates of all n vertices of the face (to discuss further in section 1.2). We would select 68 vertices out of all these n vertices, and then project the 3D coordinates onto the 2D coordinate based on pose ρ (i.e. rotation/translation) of the face and settings of the camera (i.e. extrinsic matrix H and calibration matrix of the camera). If the input x_i is a 3D image (e.g. obtained by say 3D scanning) instead of a 2D image, we can remove $Proj2D(\cdot)$ from (1) and compare the coordinates directly at $\mathbb{R}^{3 \cdot 68}$. In this paper we would focus on the 2D x_i.

In contrast to coordinates oriented embeddings encoders, the recognition oriented embeddings encoders have another target: To do face recognition. In this case, the loss function would be on the basis of Triplet Loss [21]:

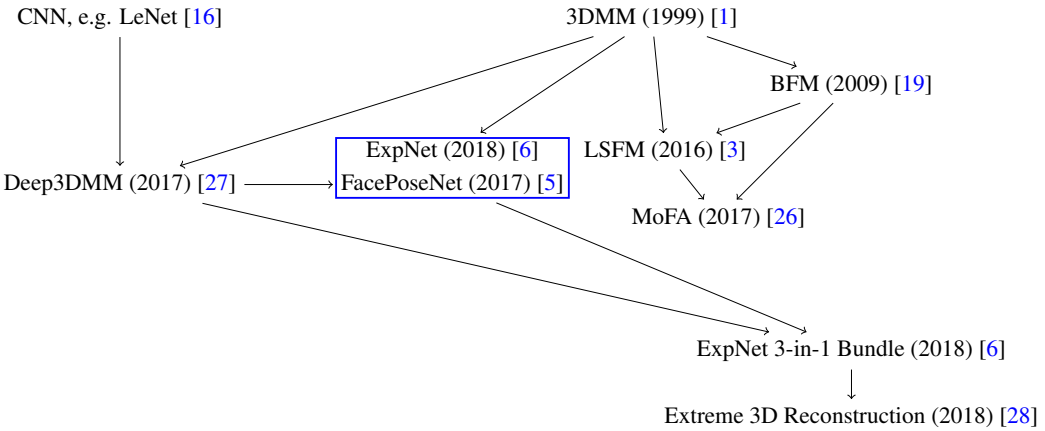


Figure 2: Family Tree of 3DMM related models.

$$\mathcal{L}_i^{TP}(e) = max(\|e(x_i) - e(pos_i)\|^2 - \|e(x_i) - e(neg_i)\|^2 + \zeta, 0) \tag{2}$$

In this case, x_i is called the anchor image, where pos_i would be an image of the same person of x_i (i.e. positive image), neg_i would be an image of a different person (i.e. negative image), and $\zeta > 0$ is a margin to ensure that $\|e(x_i) - e(neg_i)\|^2 \gg \|e(x_i) - e(pos_i)\|^2$.

In this paper, we use both types of these embeddings. Our perturbation would be based on coordinates oriented embeddings (1), and we will only use recognition oriented embeddings (2) for our evaluation purpose. We use a different type of embedding in the evaluation because we want the evaluation to be meaningful. More specifically, we would use Deep3DMM as our $e_{pb}(\cdot)$ for perturbation, and use FaceNet as our $e_{reg}(\cdot)$ in the evaluation.

1.2 The 3DMM Models

This paper aims at the embedding space perturbation and reconstruction of 3D faces, and therefore we would use models that illustrate the relationship between the embedding and the 3D coordinates of vertices, namely the 3DMM models and her descendants. In the 3DMM paper, the linear relationship between the shape/color embeddings (α, β) and the 3D-shape/RGB-texture (F, T) of an individual i is suggested as:

$$\begin{cases} F_i = S\alpha_i + \bar{F} \\ T_i = B\beta_i + \bar{T} \end{cases} \in \mathbb{R}^{3n\times 1} \tag{3}$$

where (S, B) are two matrices and $(\bar{F}, \bar{T})$ are the average shape/color among all individuals. In this paper, $\mathbb{R}^{3n\times 1}$ (or just simply $\mathbb{R}^{3n}$) refers to the set of vectors with length $3n$ which would be used more often, while $\mathbb{R}^{3\times n}$ refers to the $3 \times n$ matrices that will be used in section 2.4.

The BFM model [19] (i.e. the Basel Face Model, or simply the Basel model) further determined the size of the variables (such that the coordinates can be reconstructed in a accurate and efficient manner) and the values of $(S, B, \bar{F}, \bar{T})$. In that model $\alpha, \beta \in \mathbb{R}^{99}$ (theoretically α and β do not need to reside in the same space, but the BFM model determines that they are in the same dimension) and the number of vertices is 46990 (i.e. $F,\ T \in \mathbb{R}^{3\cdot 46990\times 1}$ for 3D and RGB color), and there are 93323 triangles formed among these 46990 vertices. The BFM model is trained by using m 3D human faces $\{F_i\}_{i=1}^m$ obtained by 3D scanner, and $(S, \bar{F})$ are determined by PCA:

$$\{F_i\}_{i=1}^m \xmapsto{PCA} (S, \bar{F})$$

The 99 dimensions can be chosen due to several reasons, e.g. sum of the 99 largest eigenvalues is significant (i.e. $\frac{\lambda_1+\ldots+\lambda_{99}}{\lambda_1+\ldots+\lambda_{3\cdot 46990}} \geq threshold$). The color-related variables $(B, \bar{T})$ are obtained by similar manner.

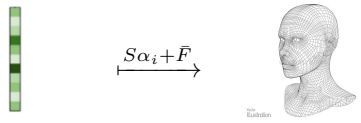


Figure 3: Shape reconstruction of the 3DMM: $\mathbb{R}^a \rightarrow \mathbb{R}^{3n}$(e.g. $\mathbb{R}^{99} \rightarrow \mathbb{R}^{3\cdot 46,990}$ for BFM).

The 3DMM models focus on synthesis, i.e. from embeddings to 3D faces. The encoding direction, i.e. to embed an image x_i into embedding $(\alpha_i, \beta_i, \eta_i, \rho_i)$, is more than that. In the era of deep learning, artificial neural network can be used to obtain these embeddings, and we would obtain (α_i, β_i) using Deep3DMM first, and then obtain (η_i, ρ_i) using ExpNet and FacePoseNet. Deep3DMM [27] with CNN layers is proposed to find the optimized α_i^{target}, which is the average of the initial embeddings of the same person:

$$\alpha_i^{target} = \sum_{x_j \in \mathcal{X}(i)} u_j \alpha_j \tag{4}$$

where $\mathcal{X}(i)$ is the collection of images of the same person that image x_i belongs to, and u_i is the confidence provided by the CLNF landmarks detector [15]. The initial embeddings $\{\alpha_j\}_{x_j \in \mathcal{X}(i)}$ can be obtained using the traditional approach, i.e. linear transformation. Therefore for two images x_i and x_j of the same person, we should have $\mathcal{X}(i) = \mathcal{X}(j) \implies \alpha_i^{target} = \alpha_j^{target}$.

An encoder $e_\omega(\cdot)$ (e.g. with model parameters $\omega \in \Omega$ and architecture of ResNet-101 [11] or VGG-16 [24] removing the last few layers) is then trained over the m images in the data base:

$$\omega^* = \underset{\omega \in \Omega}{\operatorname{argmin}} \sum_{i=1}^{m} \|e_\omega(x_i) - \alpha_i^{target}\|$$

β can be obtained in a similar manner by comparing the color embedding of image with the target color embedding. Therefore we should expect the DeepCNN to output the shape/color embedding (α_i, β_i) of image x_i by using the trained $e_{\omega^*}(\cdot)$. After obtaining the shape embedding α_i, researchers concern how to obtain the expression/pose embedding (η_i, ρ_i). The expression/pose are then incorporated into the loss function: η_i can be obtained by the deep learning models ExpNet [6] and FacePoseNet [5] based on coordinate oriented embeddings at (1), with ρ_i as a side product as below:

$$(\eta_i, \rho_i) = \underset{\eta,\rho}{\operatorname{argmin}}\, \mathcal{L}^{LM}(\alpha_i, \eta, \rho) = \underset{\eta,\rho}{\operatorname{argmin}} \|v(x_i) - \Pi_\rho(S\alpha_i + E\eta + \bar{F}))\|^2 \tag{5}$$

where $\Pi_\rho(\cdot) = Proj2D_\rho(Select_{68}(\cdot))$ and note that α_i is obtained before η_i is obtained. Although it seems that (5) optimized (η_i, ρ_i) simultaneously, in fact η_i and ρ_i are achieved by ExpNet and FacePoseNet separately. We describe the process in one go as (5) just for simple illustration purpose. Or alternatively, camera calibration procedures can be used to solve for ρ which involves eigen-decomposition of matrices. Anyway, in this paper we would use FacePoseNet to solve for ρ.

Apart from shape and color at (3), the expression η_i is also incorporated into the model as below [32]:

$$\dot{F}_i = F_i + E\eta_i = S\alpha_i + E\eta_i + \bar{F} \tag{6}$$

and the values of the transformation matrix E is determined by FaceWarehouse [4] using PCA.

1.3 Embedding Process

The Deep3DMM-ExpNet-FacePoseNet 3-in-1 bundle is firstly used by the ExpNet paper. After using the three models, the embedding $(\alpha_i, \beta_i, \eta_i, \rho_i)$ of an image x_i can be obtained.

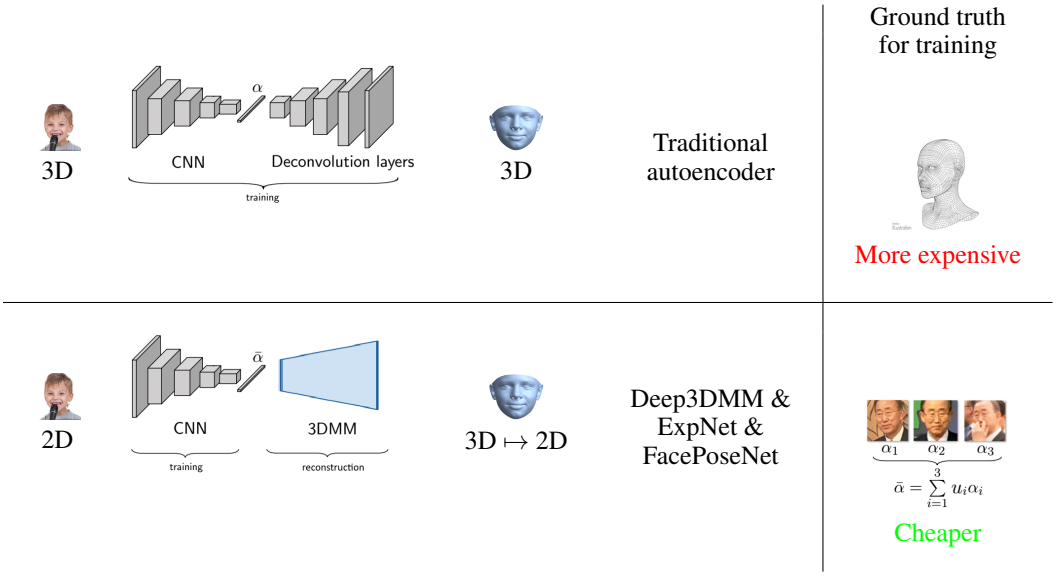


Figure 4: The Deep3DMM is to use a deep learning model (e.g. ResNet-101 [11] or VGG-16 [24]) as encoder to match the output α with the α^{target}. The ExpNet and FacePoseNet would then incorporate with the expression/pose embeddings to do the 3DMM reconstruction, to try to minimize the discrepancies of the landmarks coordinates with the original input (e.g. (1)). In all three models, the parameters of the encoder (i.e. $\omega \in \Omega$) would be trained, and the decoder is just simply the 3DMM model. Instead, if traditional autoencoder with deconvolution layer is used to reconstruct 3D face, we need 3D input as well and the comparison would also on 3D basis, which would be more expensive.

1.4 Bump Map

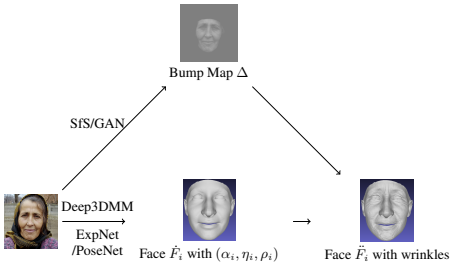
Deep3DMM, ExpNet and FacePoseNet do not consider wrinkles. Therefore the 3-in-1 bundle can reconstruct faces of young people, but does not work so well for senior citizens, who are more likely to have significant amount of wrinkles. Besides, sometimes faces can be partially occluded, e.g. by hijabs, face masks or sunglasses. In the Extreme 3D Reconstruction [28], researchers used SfS (i.e. shape from shading) [17] technique to do depth adjustment, and trained a GAN (i.e. Generative Adversarial Network) [10] to "fill the occluded holes". GAN is composed of a generator and a discriminator, while the generator is trained to fool discriminator by generating synthesis images and the discriminator is trained not to be fool. After training, the generator is expected to synthesize or partially reconstruct the original input whose quality is good enough to fool human and discriminator at certain extent, which is the reason why it would be used to synthesize partial bump map at the holes.

So a bump map Δ_i of image x_i would be produced, and the sparse mesh $\dot{F}_i$ would be converted into a dense mesh $\widehat{\dot{F}}_i$:

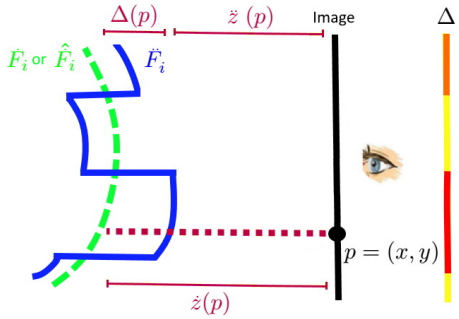
$$\Delta_i = SfS\&GAN_{\Delta}(x_i) \in \mathbb{R}^{H \times W}$$

$$\widehat{\dot{F}}_i = SfS\&GAN_{\mathcal{F}}(\dot{F}_i) \in \mathbb{R}^{3 \cdot \#(\text{pixels of face mask}) \times 1} = \mathbb{R}^{3n' \times 1} \tag{7}$$

where face mask is the mask that determine which pixels in the original image x_i are within the human face area, i.e. $n' \leq H \times W$. The bump map Δ will have the same size to the image x_i, i.e. $\mathbb{R}^{H \times W}$. In fact, values of Δ at the area outside the face mask will just be zeros. $\dot{F}_i$ and $\widehat{\dot{F}}_i$ have shapes that are approximately same, but they have different number of vertices. Number of vertices of $\dot{F}_i$ is fixed and is determined by BFM model (i.e. $n = 46,990$; sparse mesh), while the number of vertices of $\widehat{\dot{F}}_i$ varies and depends on the number of pixels of the face mask (i.e. n'; dense mesh).



(a) A GAN model is trained to output a bump map $\Delta = \Delta(x_i)$ for each image x_i so that wrinkles can be added to the image. Color would be lost at (7) when wrinkles are added, since the number of vertices of $\dot{F}_i$ and $\ddot{F}_i$ are different (i.e. vertices number n from 3DMM vs. n' from pixels). Color will be added to $\ddot{F}_i$ again by using spherical coloring technique in section 2.3.



(b) Depth displacement: $\ddot{z}(p) = \dot{z}(p) + \Delta(p)$ for each point p. $\Delta \in \mathbb{R}^{H \times W}$ will be obtained by SfS/GAN and would be used to change the z-buffer of the original face F_i. Also note that $\dot{F}_i$ and $\widehat{\dot{F}}_i$ have shapes that are approximately same, but they have different number of vertices (i.e. n vs. n') due to (7).

Figure 5: Ideas of bump map Δ of the Extreme 3D Reconstruction [28].

We expect that $n' > n$ for images whose resolution are reasonably high, which is the case for the FEI database we used (see section 2.1). The comparison of sparse mesh and dense mesh can also be shown in histogram based on their spherical coordinate, as illustrated in figure 9.c. Also note that color would be lost at (7) due to the change of vertices number, and thus F would not contain color from now on, before we add the color back by using Sphrical Bin Coloring (SBC) at section 2.3.

We will then use the bump map Δ to do depth displacement:

$$\ddot{F}_i = DepthDisplacement_{\Delta}(\widehat{\dot{F}}_i) \tag{8}$$

Figure 5.b illustrates how to do the depth displacement. Note that for a specific $\widehat{\dot{F}}_i$ and pixel $p = (\mathrm{x}, \mathrm{y})$ on an image (x here refers to x-coordinate instead of the image x itself), we can have a depth value $\dot{z}(p)$ that reflects the depth of $\widehat{\dot{F}}_i$ for that pixel. That is, $\dot{z}(p)$ of all p forms the z-buffer of the image. Δ will then be added to the $\dot{z}$ values to do depth displacement:

$$\ddot{z}(p) = \dot{z}(p) + \Delta(p)$$

Then by using $\ddot{z}(p)$ instead of $\dot{z}(p)$, we would obtain $\ddot{F}_i$ from $\widehat{\dot{F}}_i$. The summary of the procedures using landmarks detection, Deep3DMM/ExpNet/FacePoseNet 3-in-1 bundle and bump map Δ can refer to figure 6.

1.5 Facial De-identification

There are some previous attempts in face de-identification. One of the goals is to achieve a variant version of k-anonymity. The original version of k-anonymity of a database (not necessarily human faces) is achieved if every individuals in the database cannot be distinguished from at least $k-1$ other individuals. To make k-anonymity well defined for human face embeddings, Chi&Hu (2014) [7] suggest a variant:

$$P\Big(h(x_i) = Person_i\Big) \leq \frac{1}{k} \tag{9}$$

where $h(\cdot)$ is an identity classifier and $Person_i$ is the ground truth person of the image x_i. Please note that (9) is a necessary condition (but not sufficient condition) of the original k-anonymity.

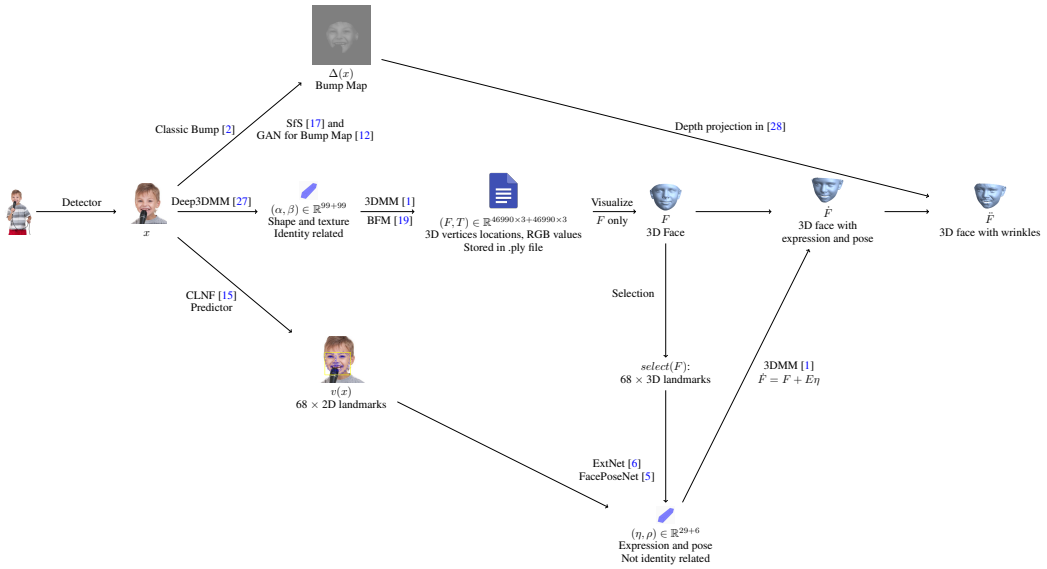


Figure 6: Summary of producing $\ddot{F}$ before applying Convex Hull Escape Perturbation (CHEP) at section 2.2 and Spherical Bins Coloring (SBC) at section 2.3.

Chi&Hu (2014) [7] embed the facial images $\{x_i\}_{i=1}^m$ as $\{\alpha_i\}_{i=1}^m$ by LDA (linear discriminant analysis) which is linear. There are P identities in the database (i.e. $|\mathcal{P}| = P$ where $\mathcal{P}$ is the set of all identifies) and they use only 1 image for each person, and thus $m = P$ and $\{x_i\}_{i=1}^m = \{x_i\}_{i=1}^P$ (i.e. all images = images of all identities). One year later in Chi&Hu (2015) [8], a 3-layer vanilla neural network is used to do the embedding to obtain $\{\alpha_i\}_{i=1}^P$.

For the part of perturbation, they separate the identities into P/k mutually exclusive subgroups, i.e. $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup ... \cup \mathcal{P}_{P/k}$ (if k cannot divide P then take $\lfloor P/k \rfloor$ instead of P/k) with the size of each subgroup $|\mathcal{P}_i| \geq k$. After the embedding $\{\alpha_i\}_{i=1}^P$ is obtained, the embedding vector is perturbed by simply taking the average of the own subgroup, i.e.

$$\alpha_i' = \frac{1}{|\mathcal{P}(i)|} \sum_{j \in \mathcal{P}(i)} \alpha_j$$

where α_k' the perturbation of α_k, and $\mathcal{P}(i)$ is the subgroup that person i belongs to (note that only 1 image for each person). Therefore k-anonymity (9) can be achieved because $|\mathcal{P}(i)| \geq k$. However, people in the same subgroup would have the same perturbed embeddings, i.e. $\mathcal{P}(i) = \mathcal{P}(j) \implies \alpha_i' = \alpha_j'$, which is not very desired. In fact, from the result of [8] we can see that every reconstructed 2D faces look the same when the value of P is large.

In this paper, we will look at another goal instead, namely the reduction of top-κ accuracy (see section 3). Top-κ accuracy regards the classifier $h(\cdot)$ as correct if the ground truth identity is within the top κ classes. Besides, our perturbation will aim at perturbing an embedding α to escape the convex hull $\mathcal{C}(\alpha)$ formed by neighbors of α (will be discussed further in section 2.2), and this idea is inspired by a psychological phenomenon called Perceptual Magnet Effect [9].

1.6 Related Work

There are some other works which aim at protecting the privacy about revealing an identity of a human face. While this paper is focusing on embedding perturbation, another area of adversarial patches has also been studied. Sharif et. al. (2016) [22] applies a physical/digital glasses to lower the chance that a human face being detected/recognized. Pautov et. al. (2019) [18] applies patches on nose/forehead etc. to achieve de-identification.

And for the face embedding model, apart from the Deep3DMM/ExpNet/FacePoseNet 3-in-1 bundle that we use in this paper, some models are also developed for the purpose of face embedding. MoFA

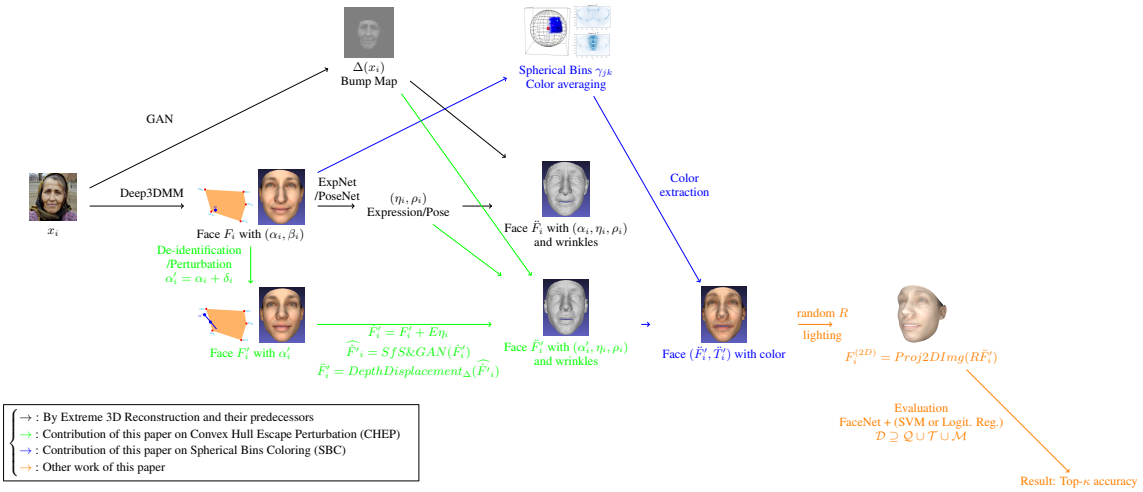


Figure 7: Contribution of this paper (highlighted in green, blue and orange color). The black color arrows are the procedures used in the Extreme 3D Reconstruction [28] and their predecessors.

[26] embeds face by using an autoencoder. Apart from the coordinate oriented loss $\mathcal{L}^{LM}$ we use, MoFA also use pixelwise loss $\mathcal{L}^{PX}$ which measures the discrepancies of pixels' intensities between the original image and the reconstructed 2D image.

Generative models also serves the purpose of reconstructing faces. StyleGAN [13] and her successor StyleGAN2 [14] show stunning performance on generating/synthesizing faces of non-existence people by using GAN with a noise vector as input. It is worth to point out that their noise vectors are random and are not embeddings obtained from existent people, so the purpose of their work is fundamentally different to this paper. Besides, the components of the noise vector doesn't contain semantic meaning unless disentanglement techniques such as [23] are used in the future work for those models. VQ-VAE-2 [20] can also embed faces (or to embed more general non-facial images) in some sense but this generative model is also mainly designed for image synthesis.

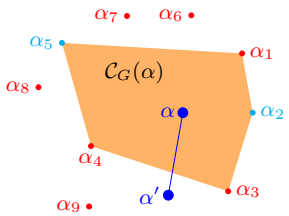
FaceNet [21] and DeepFace [25] are models developed by Google in 2015 and Facebook in 2014, which based on recognition oriented embeddings instead of coordinate oriented embeddings. We will use FaceNet + Machine Learning Model as our evaluation classifier $h(\cdot)$ in section 2.5.

2 Overview

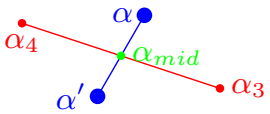
All notations used in this paper are summarized in the notation table at section 6.2 of the appendix. In this paper, we are going to perturb the shape embedding α to α' (i.e. F_i to F'_i), and then we would apply the expression/pose/wrinkles to obtain face $\ddot{F}'_i$ from F'_i. As mentioned before, the color of face is lost at (7), and we will reinstate the color T'_i by using the Spherical Bins Coloring (SBC) method. We will then apply random rotation R to each individual and project the 3D face to a 2D image $F_i^{(2D)}$. The database $\mathcal{D}$ would be split into the target set $\mathcal{T}$, pivot set $\mathcal{Q}$ and classifier training set $\mathcal{M}$ in the evaluation to compute the top-κ accuracy of different perturbation approaches. A summary of these procedures is illustrated at figure 7.

2.1 Database

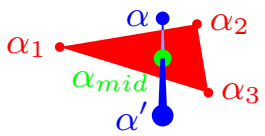
We used the FEI database maintained by C. E. Thomaz. There are 200 individuals in the database, and there are 14 images for each individual, including 10 different angles (same for all individuals), one image with normal expression, one with smile expression, one with darker background and one with very dark background (i.e. $10 + 1 + 1 + 1 + 1 = 14$). All images are colorized and in the size of 640×480, i.e. x_i before cropping $\in \mathbb{R}^{H \times W \times 3} = \mathbb{R}^{480 \times 640 \times 3}$. We would use at most 10 out of 14 images for each individual, i.e. 8 angles to train the evaluation classifier h, and use the two images with normal/smile expression to do the perturbation (i.e. $8 + 1 + 1 = 10$). Therefore the



(a) Perturb α to α' to escape the convex hull $\mathcal{C}_G(\alpha)$. We illustrate α in dim-2 and $G = 5$. Note that
$\begin{cases} 1,2,3,4,5 \in \mathcal{G}_\alpha \text{ (\textbf{neighbors})} \\ 1,3,4,6,7,8,9 \in \mathcal{Q} \text{ (\textbf{pivots})} \\ 3,4 \in q_\alpha \subseteq \mathcal{Q} \cap \mathcal{G}_\alpha \\ G \geq dim(\alpha) + 1 \end{cases}$



(b) The pivot-jump perturbation. "$\alpha' \mapsto \alpha$" is jumping across α_{mid} by using $q = 2$, i.e. $3,4 \in q_\alpha$. α_{mid} would be the midpoint between α_3 and α_4 (i.e. centroid of q_α). We illustrate α in dim-2 (or dim-3). Value of G does not matter.



(c) Another pivot-jump perturbation. "$\alpha' \mapsto \alpha$" is jumping across α_{mid} by using $q = 3$, i.e. $1,2,3 \in q_\alpha$. We illustrate α in dim-3. Again, value of G does not matter.

Figure 8: Perturb α to become α'. We illustrate α at dim-2 and dim-3 for easy explanation. In our 3DMM we have α in dim-99 (i.e. $dim(\alpha) = 99$ or $\alpha \in \mathbb{R}^{99}$). We should expect $q \leq dim(\alpha) \leq G - 1$ and $q \leq Q$.

database $\mathcal{D}$ we use contains $200 \times 10 = 2000$ images. We will talk about the split of database $\mathcal{D}$ into $\mathcal{Q} \cup \mathcal{T} \cup \mathcal{M}$ in section 2.5. We will also crop the images before we proceed on the landmarks detection and embedding processes, but we will not discuss the details as the cropping process can be completed by some standard tool.

2.2 Convex Hull Escape Perturbation (CHEP)

Perturbation of embedding is the key to de-identify our 3D faces. Embeddings have been studied by the Machine Learning researchers for decades, and are considered as knowledge representations of the original input in a condensed way. Therefore the perturbation of embedding from α to α' should be able to change some intrinsic properties of the original input. In the meantime, for face de-identification problem we would like to keep the same expression/pose of the original image x_i and only change the shape such that the identity of the individual can be protected. As such, we will perturb the shape embedding α, and keep the expression/pose embeddings (η, ρ) intact.

In classic classification problems in machine learning, features of data x_i can be geometrically separated into different regions, and any individual data point can be classified as another category if they pass the decision boundary and locate at another region. Now the embedding α_i can be considered as the extracted features of x_i, and therefore we concern about the distribution and regions of α_i. To achieve de-identification, we hope our embedding α_i can go to the "region of another identity" instead of the original one, such that the original identity would not be recognized. The first step of the 3-in-1 bundle embedding is the Deep3DMM model, which use (4) to set the α_i^{target}. Therefore we can expect $\alpha_i \approx \alpha_j$ if the images i and j belong to the same person.

Therefore we suggest **Convex Hull Escape Perturbation (CHEP)**: to use an α_i from each individuals to form a convex hull $\mathcal{C}(\alpha)$ to surround our target α, and we want to push α to α' so as to escape $\mathcal{C}(\alpha)$:

$$\begin{cases} \mathcal{G}_\alpha = \{p_1, ..., p_G\} \\ \mathcal{C}_G(\alpha) = Hull(\mathcal{G}_\alpha) \end{cases} \text{, where } \begin{cases} p_j \in \mathcal{P} \text{ is an individual with } \alpha_{p_j} \text{ nearby } \alpha \\ G = |\mathcal{G}_\alpha| \geq dim(\alpha) + 1 \end{cases}$$

where $\mathcal{G}_\alpha$ is called a **neighbor set**: the set of neighbors of α used to form a convex hull. If α reside in a dim-a space, we would expect $G \geq a + 1$ such that $\mathcal{C}_G(\alpha)$ can be formed. For example, in 3DMM we have $a = 99$, and we expect $G \geq 100$. For example, we can pick say the 100 nearest identities of α to form $\mathcal{C}_G(\alpha)$. In some rare case, it is possible that $\alpha \notin \mathcal{C}_G(\alpha)$ at the very beginning, but it is not an issue as long as we apply the jumping method at (11) to push the α further away.

Note that escaping $\mathcal{C}_G(\alpha)$ not only make sense under the machine learning context, but also make sense on psychological context. Feldman (2007) discovered a phenomenon called perceptual magnet

effect [9] in psychological experiments. Human participants will perceive and classify a data point based on the geometric region that the features locate. So pushing α' out from $\mathcal{C}_G(\alpha)$ will make human participant to perceive the reconstructed face of α' as another identity. Thus α'_i should be able to fool not only face recognition systems but also human participants. Our evaluation classifier $h(\cdot)$ of section 2.5 will focus on the former only.

To practically escape $\mathcal{C}_G(\alpha)$, in view of data availability, we suggest the **pivot-jumping perturbation** at (11). We pick some individuals $\mathcal{Q} \subseteq \mathcal{P}$ (the number $Q = |\mathcal{Q}|$ is a parameter chosen by us), where $\mathcal{P}$ is the number of all identities in a database. We will only perturb α_i where $i \notin \mathcal{Q}$, and we called $\mathcal{Q}$ the **pivot set**. Note that $\mathcal{Q}$ would be the same for all different α_i whose identity $i \in \mathcal{P} \setminus \mathcal{Q}$.

For any α_i that we want to perturb, we will choose a **perturbation set** $\boldsymbol{q}_i \subseteq \mathcal{Q}$ (again, $q = |\boldsymbol{q}|$ is a parameter chosen by us) which are the nearest neighbors of α_i. In other words, elements in $\boldsymbol{q}_i$ are the selected pivots for identity i:

$$\boldsymbol{q}_i = NearestNB_{q,\mathcal{Q}}(i) = \underset{\begin{cases}\boldsymbol{q}_j \subseteq \mathcal{Q} \\ |\boldsymbol{q}_j| = q\end{cases}}{\operatorname{argmin}} \sum_{k \in \boldsymbol{q}_j} \|\alpha_k - \alpha_i\|^2 \tag{10}$$

or we can write $\boldsymbol{q}_\alpha$ (instead of $\boldsymbol{q}_i$) for a specific α when no confusion would be made. In fact we can use some selection functions other than $NearestNB(\cdot)$, say $FurthestNB(\cdot)$ to choose the perturbation set $\boldsymbol{q}_i$. We will talk about that in the future work (i.e. section 5), and we will focus on $NearestNB(\cdot)$ in this paper. When we required $|\boldsymbol{q}_\alpha| \leq dim(\alpha) \leq |\mathcal{G}_\alpha| - 1$, we would expect $\boldsymbol{q}_\alpha \subsetneq \mathcal{G}_\alpha$ (which make it feasible and controllable to escape $\mathcal{C}(\alpha)$). Since $\boldsymbol{q}_\alpha$ is also chosen from $\mathcal{Q}$, we have:

$$\boldsymbol{q}_\alpha \subseteq \mathcal{Q} \cap \mathcal{G}_\alpha, \text{ and thus } \begin{cases} q \leq dim(\alpha) \leq G - 1 \\ q \leq Q \end{cases}$$

In this paper, since we have 200 individuals in our database, we would pick $q = 2$ or 10, and $Q = 20$ or 100, and therefore we have four pairs of (q, Q): $(2, 20)$, $(10, 20)$, $(2, 100)$ and $(10, 100)$. We do not care about the value of G, since by using the jump at (11) we should be able to get rid of the convex hull $\mathcal{C}_G(\alpha)$ regardless the value of G, given that all $\alpha_i \in \mathcal{G}_\alpha$ are at the boundary of $\mathcal{C}_G(\alpha)$.

We will then proceed on the pivot-jump perturbation. We want to reflect α over the centroid of $\boldsymbol{q}_\alpha$, which we called α_{mid}. Therefore the jump δ will be two times $\alpha_{mid} - \alpha$:

$$\begin{cases} \alpha_{mid} = \frac{1}{|\boldsymbol{q}_\alpha|} \sum\limits_{j \in \boldsymbol{q}_\alpha} \alpha_j \\ \alpha' = \alpha + \delta = \alpha + 2(\alpha_{mid} - \alpha) = 2\alpha_{mid} - \alpha \end{cases} \tag{11}$$

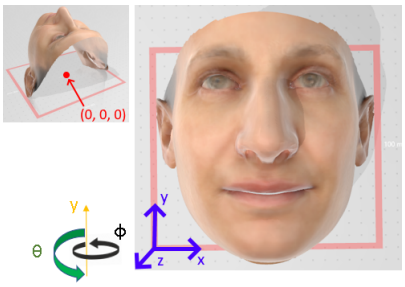
If our goal is just to escape $\mathcal{C}(\alpha)$, in fact we can choose $\delta = b(\alpha_{mid} - \alpha)$ with any $b > 1$, say 1.1 or 10. In (11) we picked $b = 2$ just because it reserved some kind of symmetry. After applying pivot-jumping perturbation at (11), α' should now be outside $\mathcal{C}_\alpha(\alpha)$.

2.3 Spherical Bins Coloring

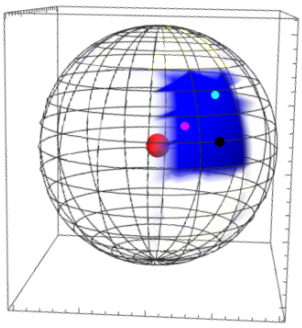
After obtaining $\alpha'_i = \alpha_i + \delta_i$, we can obtain the faces with expression/pose/wrinkles by using a similar manner of (3), (6), (7) and (8):

$$\begin{cases} F'_i = S\alpha'_i + \bar{F} \\ \dot{F}'_i = F'_i + E\eta_i \\ \widehat{\dot{F}'}_i = SfS\&GAN(\dot{F}'_i) \\ \ddot{F}'_i = DepthDisplacement_\Delta(\widehat{\dot{F}'}_i) \end{cases}$$

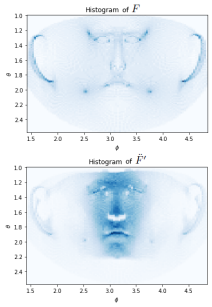
Before taking $\ddot{F}'_i$ to the identity classifier $h(\cdot)$ for our evaluation, note that the color of face is lost at (7) when wrinkles are added, since the number of vertices of $\dot{F}_i$ and $\ddot{F}_i$ are different (i.e. n and



(a) Conversion of rectangular coordinate (x, y, z) into spherical coordinate (r, θ, ϕ). Note that the inclination angle θ is measured from the y-axis. The origin $(0, 0, 0)$ locates behind the human nose.



(b) For a specific bin γ_{jk}, take the average color of vertices inside γ_{jk} to compute the average color Ξ_{jk}. In this case, there are 3 vertices in γ_{jk}, namely purple, green and black-blue. Taking average of 3 points we would obtain blue as the average color for the bin. The sphere represents the entire $\Gamma = [0, \pi] \times [0, 2\pi]$, and the red dot is the center of the sphere.



(c) 2D-histogram of vertices' (θ, ϕ) coordinates of the sparse mesh F and the dense mesh $\ddot{F}'$.

Figure 9: Key ideas on how to obtain the average color of bins γ_{jk}.

n' respectively). We want to use a colorized face for the evaluation classifier $h(\cdot)$, so we need to reinstate the color before we proceed on the identity classification.

As explained, the color $T_i = B\beta_i + \bar{T} \in \mathbb{R}^{3n \times 1}$ is already obtained by Deep3DMM at (3), but it can not be directly applied to $\ddot{F}_i \in \mathbb{R}^{3n' \times 1}$ because $n' \neq n$. So first of all, we will construct spherical bins γ_{jk} and to determine their color. The transformation from rectangular coordinate (x, y, z) to spherical coordinate (r, θ, ϕ) is as below, which is illustrated at figure 9.a:

$$(r, \theta, \phi) = Sph(x, y, z), \text{ i.e. } \begin{cases} r = \sqrt{x^2 + y^2 + z^2} \\ \theta = (arctan\frac{\sqrt{x^2+z^2}}{y}) \ mod \ \pi \\ \phi = (\pi + arctan\frac{x}{z}) \ mod \ 2\pi \end{cases} \tag{12}$$

Note that the above conversion in (12) is different to the spherical coordinate conversion in ISO 80000-2:2019 which is commonly used in Physics. Our inclination angle θ is measured from the y-axis, while θ of ISO 80000-2:2019 is measured from z-axis. Our azimuthal angle ϕ starts from the negative side of z-axis, while ϕ of ISO 80000-2:2019 starts from positive side of x-axis. Our modification allows us to be more convenient to describe our work for the human face problem, especially the histogram visualization of spare/dense meshes F and $\ddot{F}'$ at figure 9.c.

After the spherical coordinate conversion, we would then separate the space $\Gamma = [0, \pi] \times [0, 2\pi]$ into different bins γ_{jk}:

$$\begin{cases} \Gamma = [0, \pi] \times [0, 2\pi] = \bigcup\limits_{j,k} \gamma_{jk} \\ \gamma_{jk} = [\theta_j, \theta_{j+1}] \times [\phi_k, \phi_{k+1}] \end{cases}$$

In this paper, we cut both $[0, \pi]$ and $[0, 2\pi]$ into 100 pieces, and therefore $|\{\gamma_{jk}\}| = 100 \times 100 = 10000$, which is reasonable since there are 46,990 3DMM vertices (with an average of around 4.6 vertices per bin to compute the bin color). For each bin γ_{jk}, we would compute the average color of γ_{jk} for image i based on the color of vertices, i.e. T_i of (3):

$$\Xi_{ijk} = AvgColor(\gamma_{jk}) = \frac{1}{|\{Sph(x_{\ell i}, y_{\ell i}, z_{\ell i}) \in \gamma_{jk}\}|} \sum_{Sph(x_{\ell i}, y_{\ell i}, z_{\ell i}) \in \gamma_{jk}} (r_{\ell i}, g_{\ell i}, b_{\ell i}) \in \mathbb{R}^3$$

where $(x_{\ell i}, y_{\ell i}, z_{\ell i})$ and $(r_{\ell i}, g_{\ell i}, b_{\ell i})$ are the rectangular coordinate and color of vertex v_ℓ (for $\ell = 1$, ..., n) and the values are obtained from $F_i, T_i \in \mathbb{R}^{3n \times 1}$ of (3). For example:

$$\begin{cases} (x_{\ell i}, y_{\ell i}, z_{\ell i}) = F_i[\,3\ell, 3\ell+1, 3\ell+2] \\ (r_{\ell i}, g_{\ell i}, b_{\ell i}) = T_i[\,3\ell, 3\ell+1, 3\ell+2] \end{cases}, \text{ i.e. the } 3\ell\text{-th to } (3\ell+2)\text{-th entries}$$

After we obtained $AvgColor_i(\gamma_{jk})$ of non-perturbed (F_i, T_i) for all bins, we can obtain color $\ddot{T}'_i$ for each vertex $\ddot{v}'_\ell$ of the perturbed $\ddot{F}'_i$ by using color extraction. Instead of taking the color directly, we would smooth the color further by using **color interpolative extraction**, which take the weighted average color of $2 \times 2 = 4$ bins:

$$\ddot{T}'_i[\,3\ell, 3\ell+1, 3\ell+2] = Color(\ddot{v}'_\ell) = \sum_{u,v=0}^{1} w_{(j-u)(k-v)} \Xi_{i(j-u)(k-v)} \tag{13}$$

where w_{jk} is the corresponding weight of the bin γ_{jk}, and the values of j and k are determined by the spherical coordinates of $\ddot{v}'_\ell$. To explain this, let's focus on a vertex $\ddot{v}'_\ell$ of $\ddot{F}'_i$ locating at $(\ddot{x}'_{\ell i}, \ddot{y}'_{\ell i}, \ddot{z}'_{\ell i})$. We can then obtain its $(\ddot{\theta}'_{\ell i}, \ddot{\phi}'_{\ell i})$ by using $Sph(\ddot{x}'_{\ell i}, \ddot{y}'_{\ell i}, \ddot{z}'_{\ell i})$. As illustrated in figure 10, $(\ddot{\theta}'_{\ell i}, \ddot{\phi}'_{\ell i})$ must locate inside a green dotted rectangle which covers $1/4$ of each of the four adjacent bins. We write $(s, t) \in [\,0, 1] \times [\,0, 1]$ to be the proportion of $(\ddot{\theta}'_{\ell i}, \ddot{\phi}'_{\ell i})$ in the green dotted rectangle. Therefore we have:

$$\begin{bmatrix} w_{(j-1)(k-1)} & w_{(j-1)k} \\ w_{j(k-1)} & w_{jk} \end{bmatrix} = \begin{bmatrix} 1-s-t+st & s-st \\ t-st & st \end{bmatrix}$$

Note that the sum of the four weights is one, i.e. $\sum_{u,v=0}^{1} w_{(j-u)(k-v)} = 1$. Therefore, if $(\ddot{\theta}'_{\ell i}, \ddot{\phi}'_{\ell i})$ locates very closed to the bottom right of the dotted rectangle (i.e. $(s, t) \approx (1, 1)$), we will have $w_{jk} \approx 1$ and therefore the color at the bottom right would be of most significant weight.

Therefore we have reinstated the color $\ddot{T}'_i$ by using (13) and we now have a colorized face $(\ddot{F}'_i, \ddot{T}'_i)$.

2.4 Random Rotation and 2D Projection

After we have obtained a colorized face $(\ddot{F}'_i, \ddot{T}'_i)$, we need to project it as a 2D image before we pass it to the identifier classifier $h(\cdot)$ for evaluation. For each individual, we would sample a random rotation $R \in \mathbb{R}^{3\times3}$ which will be applied to each individual:

$$\begin{cases} F_i^{(2D)} = Proj2DImg(R\dddot{F}'_i) \in \mathbb{R}^{H' \times W' \times 3} \\ \dddot{F}'_i = Reshape_{3 \times n'}(\ddot{F}'_i) \end{cases} \tag{14}$$

Note that the vector $\ddot{F}'_i \in \mathbb{R}^{3n' \times 1}$ will be reshaped as a matrix $\dddot{F}'_i \in \mathbb{R}^{3 \times n'}$ before doing matrix multiplication with $R \in \mathbb{R}^{3\times3}$. The same R_i (i.e. same rotation angle) will be used to rotate $(\ddot{F}'_i, \ddot{T}'_i)$ of different perturbation approach (i.e. different (q, Q) values) for the same individual i. Besides, $Proj2DImg(\cdot)$ of (14) is different to $Proj2D(\cdot)$ of (1), while the former refers to coordinate projection and the latter refers to the image rendering. $F_i^{(2D)}$ is expected to be a RGB image and thus belongs to $\mathbb{R}^{H' \times W' \times 3}$.

Since we do not want the face to be rotated too much (e.g. showing the back of face to the camera, or upside down), we would restrict the rotation angle by a small value. To implement this, we would

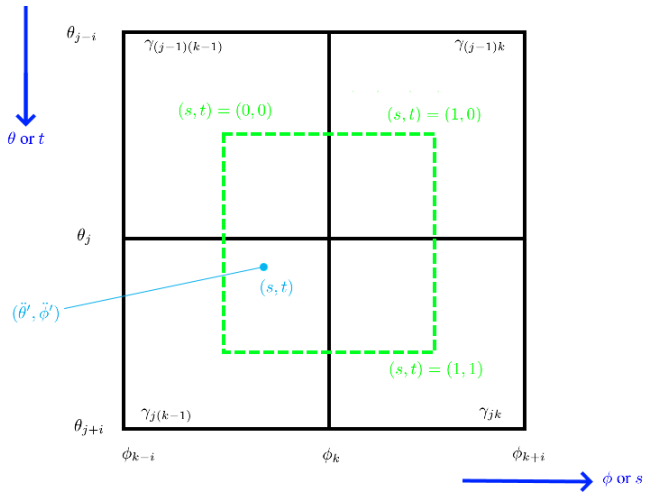


Figure 10: The weight $\begin{bmatrix} w_{(j-1)(k-1)} & w_{(j-1)k} \\ w_{j(k-1)} & w_{jk} \end{bmatrix} = \begin{bmatrix} 1-s-t+st & s-st \\ t-st & st \end{bmatrix}$ (where the sum of weight is 1) of bins to determine the color of vertex at $(\ddot{\theta}', \ddot{\phi}')$ by weight averaging the color of 4 adjacent bins using (13).

sample a random rotation vector $\varrho \in \mathbb{R}^3$ first (with $\|\varrho\|$ being a small value, i.e. little rotation), and then transform ϱ into $R \in \mathbb{R}^{3\times 3}$ by using the Rodrigues' rotation formula:

$$\begin{cases} R = Rodrigues(\varrho) = I + (sin\|\varrho\|)U + (1 - cos\|\varrho\|)U^2 \\ \text{where } U = \begin{bmatrix} 0 & -\varrho_z/\|\varrho\| & \varrho_y/\|\varrho\| \\ \varrho_z/\|\varrho\| & 0 & -\varrho_x/\|\varrho\| \\ -\varrho_y/\|\varrho\| & \varrho_x/\|\varrho\| & 0 \end{bmatrix} \end{cases}$$

i.e. U is the cross product matrix of the unit vector $\varrho/\|\varrho\|$. In this paper, we ramdomly sampled ϱ from $[0, \pi/8]^3$, and thus the maximum rotation possible would be $\sqrt{(\pi/8)^2 + (\pi/8)^2 + (\pi/8)^2} = \sqrt{3}\pi/8 = 38.97\ deg$ which is still reasonable for face recognition task.

To project $R\dddot{F}'_i$ as a 2D image, we would resize $R\dddot{F}'_i$ such that the width of the human face is ≈ 1, and then set the extrinsic matrix of the camera as:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1.8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that the spatial translation of direction-z in is 1.8 (i.e. $H[3,4]$) which is reasonable for a human face with width ≈ 1. For the intrinsic matrix of camera we would set $fov = \pi/3$ and $AspectRatio = 1$. We would also apply some ambient light and spotlight to the human face. To sum up:

$$Proj2DImg(\cdot) = Proj2DImg_{(Resize,\ H,\ fov,\ AspectRatio,\ light)}(\cdot)$$

After that, we would obtain $F_i^{(2D)}$.

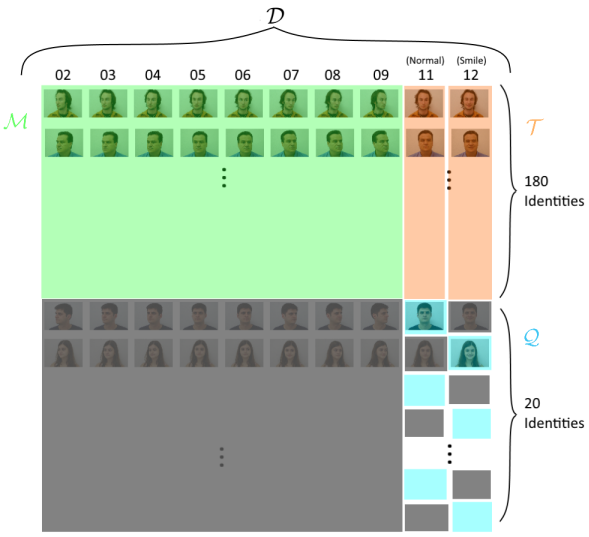


Figure 11: The $(180, 20)$ split of our 2000-image database $\mathcal{D}$ into $\mathcal{Q} \cup \mathcal{T} \cup \mathcal{M}$. Target set $\mathcal{T}$ are our targeted images to be perturbed at the embedding space (i.e. $\alpha' = \alpha + \delta$); Pivot set $\mathcal{Q}$ would be used as the candidates of pivots to be selected (i.e. candidates of the nearest neighbor selection and to compute α_{mid}). Classifier Training set $\mathcal{M}$ are images to be used to train the identity classifier $h(\cdot)$ in the evaluation (i.e. no need to do CHEP and SBC for these images). Note that we also have two other similar split of $(100, 20)$ and $(100, 100)$.

2.5 Evaluation

In the FEI database, there are 200 individual in the database (i.e. $|\mathcal{P}| = 200$). For each individual, image with indices "02" to "09" (i.e. 8 images for 8 different angles), "11" (i.e. normal expression) and "12" (i.e. smile expression) would be used for this paper. The angles of images "01" and "10" are too extreme, while the background of images "13" and "14" are too dark, so we would not use these images. Thus we would use $200 \times (8 + 1 + 1) = 2000$ images in this paper, and we call the collection of these 2000 images $\mathcal{D}$ (i.e. $|\mathcal{D}| = 2000$).

We would then split our 2000-image database $\mathcal{D}$ into three set (with some remaining images not used) which are mutually exclusive to each other, as illustrated in figure 11:

$$\mathcal{D} \supseteq \mathcal{Q} \cup \mathcal{T} \cup \mathcal{M}$$

Set $\mathcal{Q}$ is called the **pivot set** which we have encountered in section 2.2. Each identity in $\mathcal{Q}$ only contains one image in $\mathcal{Q}$, i.e. #(images in $\mathcal{Q}$) = #(identities in $\mathcal{Q}$), as we also cares about identity (not image) in the pivot-jumping perturbation at (11); Set $\mathcal{T}$ is called **target set**, which consists of all targeted images that we are going to perturb; Set $\mathcal{M}$ is called **classifier training set**, which is the training set of images to train the identity classifier $h(\cdot)$ in our final evaluation process. Note that each identity in $\mathcal{T}$ and $\mathcal{M}$ can have multiples images in $\mathcal{T}$ and $\mathcal{M}$.

As we are doing perturbation based on two different values of $Q = |\mathcal{Q}|$, namely $Q = 20$ and $Q = 100$, we have two different splits. Let's talk about the split for $Q = 20$ first. We would split the 200 identities into $(180, 20)$, while there are 180 identities in $\mathcal{M}$ and $\mathcal{T}$, and only 20 identities in $\mathcal{Q}$.

That is, image $x_{j,02}$ to image $x_{j,09}$ (i.e. 8 images) of 180 person would form $\mathcal{M}$, i.e. $|\mathcal{M}| = 8 * 180 = 1440$. We will then use image $x_{j,11}$ and image $x_{j,12}$ of the same 180 person to form $\mathcal{T}$, i.e. $|\mathcal{T}| = 2 * 180 = 360$. For the reserved 20 identities, we would pick image $x_{j,11}$ of 10 of them, and image $x_{j,12}$ of another 10 of them to form $\mathcal{Q}$, i.e. $|\mathcal{Q}| = 1 * 20 = 20$. There are 180 images in $\mathcal{D}$ that we would not use, i.e. $|\mathcal{D}| - |\mathcal{M}| - |\mathcal{T}| - |\mathcal{Q}| = 2000 - 1440 - 360 - 20 = 180$.

Similarly, for $Q = 100$, we can split the 200 identities into $(100, 100)$. For comparison purpose, we would use the same pivot identities from $(180, 20)$ and same target identities from $(100, 100)$ to form the hybrid split $(100, 20)$ (which used 120 identities only and did not use up all 200 identities

available). To sum up, we have 3 different split: $(180, 20)$, $(100, 20)$ and $(100, 100)$, while the latter two split share the same 100 target identities.

For the identity classifier $h(\cdot)$ we use in the evaluation process, we would pick another base model with embedding type different to Deep3DMM/ExpNet/PostNet. As mentioned before, our 3-in-1 bundle is based on coordinates oriented embeddings of (1), so we prefer to use another model with recognition oriented embeddings of (2), e.g. FaceNet. We would pick FaceNet as our base model, and train classifiers based on FaceNet + another machine learning model. Therefore our $h(\cdot)$ can be "FaceNet + SVM" or "FaceNet + logistic regression" under different setting (i.e. L1 vs. L2, "1 vs. all" vs. Multinomial, or different regularization parameters λ etc.). As mentioned above, $\mathcal{Q}$ would be used to train $h(\cdot)$, and to be precise we would only train the machine learning model attached to FaceNet while we would keep the parameters of the pre-trained FaceNet intact.

3 Result

We have three splits for the database $\mathcal{D}$: $(180, 20)$, $(100, 20)$ and $(100, 100)$. Splits $(180, 20)$ and $(100, 20)$ share the same 20 pivot identities, while plits $(100, 20)$ and $(100, 100)$ share the same 100 target identities. For the accuracy computation, we would cut some slack in the classification, and we would consider it a hit (i.e. classifier $h(\cdot)$ to be regarded as correct) if the true label is classified within the top κ identities (instead of the single top identity only), and we call this **top-κ accuracy**. We would pick $\kappa = 5$ for $(180, 20)$, while we would pick the same $\kappa = 10$ for both $(100, 20)$ and $(100, 100)$ so these two splits can be directly compared. Our result are listed at table 1.

Identity classifier $h(\cdot)$ for evaluation	λ	Our perturbation approaches							
		$\kappa = 5$			$\kappa = 10$				
		ID split = $(180, 20)$			ID split = $(100, 20)$		ID split = $(100, 100)$		
		α not perturbed	i.e. $Q = 20$		i.e. $Q = 20$		α not perturbed	i.e. $Q = 100$	
			$q = 2$	$q = 10$	$q = 2$	$q = 10$		$q = 2$	$q = 10$
Expression: Normal (i.e. to perturb images "11" of $\mathcal{T}$)									
FaceNet + SVM	1	42.7%	18.1%	13.6%	40.4%	31.3%	64.6%	44.4%	48.5%
FaceNet + Logistic Reg. (L2 "1 vs. all")	1	45.5%	18.1%	13.6%	38.4%	29.3%	62.6%	41.4%	43.4%
FaceNet + Logistic Reg. (L2 Multinomial)	1	43.8%	18.1%	13.6%	38.4%	27.3%	62.6%	40.4%	43.4%
FaceNet + Logistic Reg. (L1 Multinomial)	1	19.1%	9.0%	4.0%	24.2%	20.2%	44.4%	24.2%	25.3%
FaceNet + SVM	10	42.1%	18.1%	13.0%	39.4%	31.3%	64.6%	44.4%	47.5%
FaceNet + Logistic Reg. (L2 "1 vs. all")	10	42.7%	17.5%	13.6%	39.4%	26.3%	61.6%	43.4%	46.5%
Expression: Smile (i.e. to perturb images "12" of $\mathcal{T}$)									
FaceNet + SVM	1	33.9%	16.2%	10.6%	37.4%	36.4%	57.0%	37.4%	39.0%
FaceNet + Logistic Reg. (L2 "1 vs. all")	1	37.8%	17.9%	12.3%	39.4%	31.3%	59.0%	42.4%	38.0%
FaceNet + Logistic Reg. (L2 Multinomial)	1	37.2%	17.9%	12.3%	39.4%	30.3%	60.0%	42.4%	37.0%
FaceNet + Logistic Reg. (L1 Multinomial)	1	12.2%	6.7%	5.0%	21.2%	17.2%	38.0%	19.2%	21.0%
FaceNet + SVM	10	33.9%	16.2%	10.6%	37.4%	36.4%	57.0%	37.4%	39.0%
FaceNet + Logistic Reg. (L2 "1 vs. all")	10	37.2%	16.2%	10.6%	38.4%	30.3%	62.0%	42.4%	39.0%

Table 1: Our result: Top-κ accuracies of classifiers on $F_i^{(2D)}$. Note that the split $(100, 20)$ and $(100, 100)$ share the same accuracies of non-perturbed α. The trends of these figures are in line with the theoretical expectation.

We can observe several things. First of all, when the value of κ is increased from 5 to 10, the top-κ accuracies increases significantly, e.g. Increment for normal expression from $\sim 40\%$ to $\sim 60\%$ for non-perturbed faces, or from $\sim 20\%$ to $\sim 40\%$ for $(Q, q) = (20, 2)$. It is reasonable regardless of whether our CHEP method works because we are loosening our criteria of getting a hit. Second, the top-κ accuracies drops significantly when perturbation applies. For the normal expression, for $\kappa = 10$ it generally drops from the level of $\sim 60\%$ to the level of $30\% \sim 40\%$. For $\kappa = 5$ it drops from the level of $\sim 40\%$ to the level of $\sim 10\%$. This shows that our CHEP method works.

We can then pay a closer look. When we compare $Q = 100$ vs. $Q = 20$, the former is expected to have weaker perturbation effect. It is because we can choose a closer pivot for q to do the pivot-jumping perturbation if we have more pivot candidates to choose (i.e. $\mathcal{Q}^{(100)} \supseteq \mathcal{Q}^{(20)}$), leading to a smaller δ. From table 1, we can see that for $\kappa = 10$, the top-κ accuracies of $Q = 100$ drops less significantly than that of $Q = 20$. These results also match our theoretical expectation.

Further more, for $q = 2$ vs. $q = 10$, we should expect $q = 10$ are more likely to have a stronger perturbation effect since we are choosing more pivot-identities to perturb the original α. It is be-

cause α_{mid} of both cases are also located at the boundary of the convex hull $\mathcal{C}(\alpha)$ as we required $q \leq dim(\alpha)$ and therefore $\alpha_{mid}^{(q=10)}$ are more probable to be further than $\alpha_{mid}^{(q=2)}$ though it is not guaranteed. We can see that the top-κ accuracies of $q = 10$ drops more significantly than $q = 2$ in table 1 (except some cases in $Q = 100$) which is as expected.

There are some other observations. Values for smile expression are generally less recognizable than that of normal expression, which does not surprise us. And different classifiers $\{h(\cdot)\}$ lead to similar result, except "FaceNet + L1 Logistic" which gives very low top-κ accuracies. The increase of regularization factor from $\lambda = 1$ to $\lambda = 10$ would slightly worsen our result.

We have also illustrated examples of the 2D images $F^{(2D)}$ of 12 individuals at section 6.1 in the appendix.

4 Conclusion

In table 1, the top-κ accuracies are within a significant level before perturbation, and we can conclude that our Spherical Bins Coloring (SBC) method in section 2.3 can reinstate the color to a certain extent. Besides, the significant drops of top-κ accuracies reveals that the perturbed faces with embedding $\alpha' = \alpha + \delta$ are less recognizable than the original faces with embedding α. The trend of different (q, Q) approaches are also in line with our theoretical result. We can conclude that our Convex Hull Escape Perturbation (CHEP) method in section 2.2 using pivot-jumping perturbation with α_{mid} achieves controllable de-identification for 3D reconstructed faces.

5 Future Work

There are some potential research avenues.

Regarding the Spherical Bins Coloring (SBC) method, the average color of bins γ are computed from F (before applying expression), but the color should be extracted for $\ddot{F}'$ (after applying expression). Therefore the reconstruction can be improved further if the average color is computed from $\dot{F}$ instead of F.

Besides, we are using FaceNet + Machine learning model as our identity classifier $h(\cdot)$ in the evaluation. Due to human perceptual effect, our perturbation should be able to fool not only machine learning algorithm but also human. Human participants can be involved in the evaluation (i.e. play the role of $h(\cdot)$) in the manner of psychological experiment, e.g. similar to Human eYe Perceptual Evaluation (HYPE) [31] via crowd sourcing platforms such as Amazon Mechanical Turk (MTurk).

In our search of $\boldsymbol{q}_i$ in (10), we searched for nearest neighbors to perturb α. In fact, it is also possible to search for the furthest neighbors instead of the nearest neighbors:

$$\boldsymbol{q}_i = FurthestNB_{q,\mathcal{Q}}(i)$$

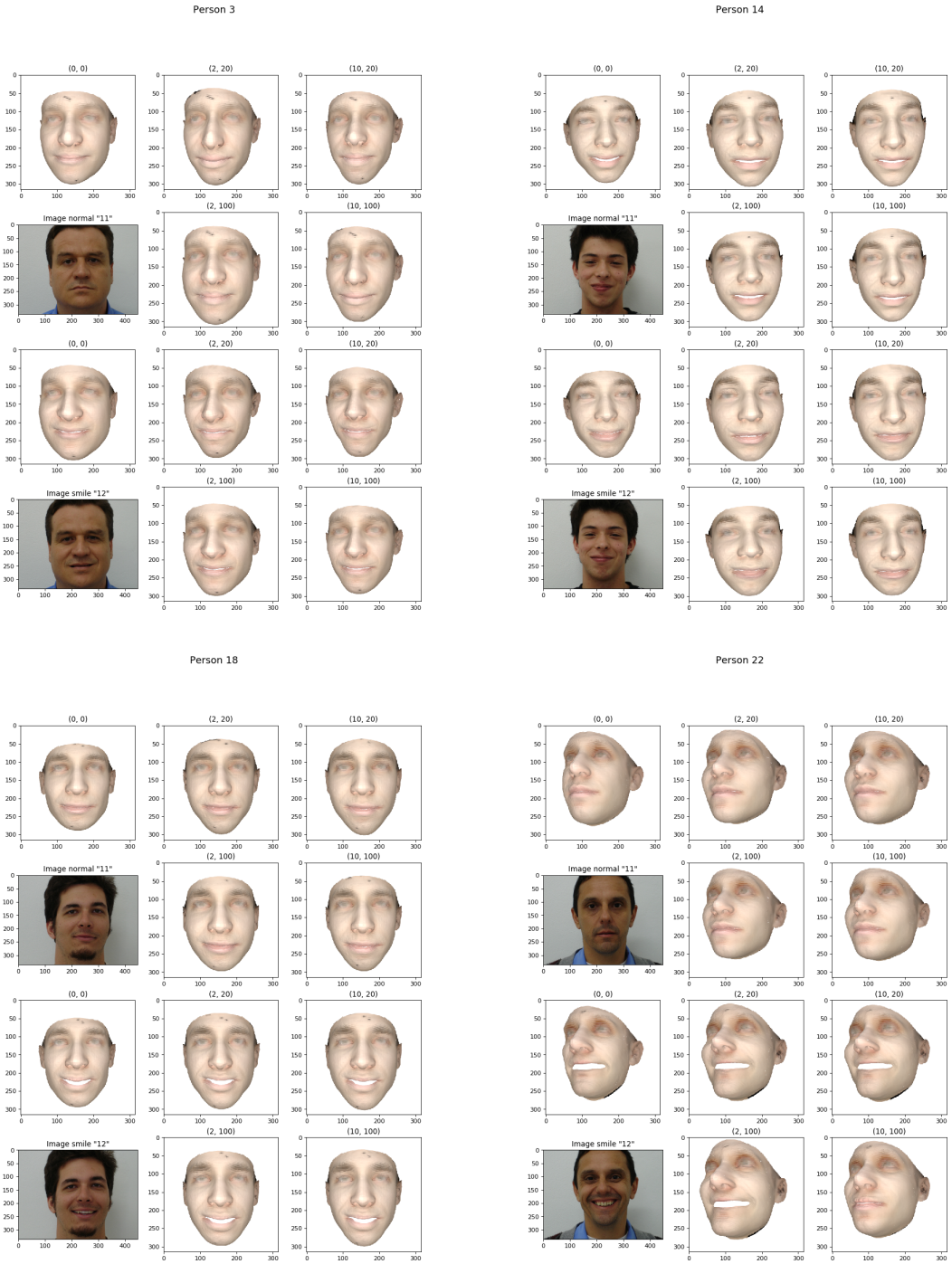
We can expect that the de-identification effect of $FurthestNB(\cdot)$ would be stronger than $NearestNB(\cdot)$. Future work may focus on how to balance the de-identification effect and while reserving the utility at certain extent.

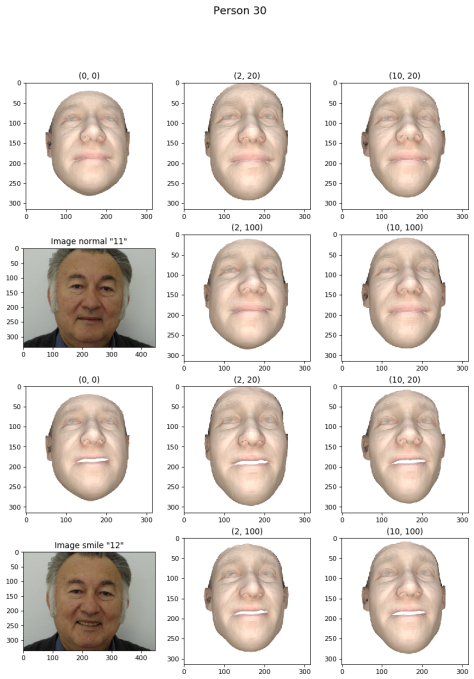
In our jumping process at (11), we picked $b = 2$ for $\delta = b(\alpha_{mid} - \alpha)$. In fact we can pick any values $b > 1$ other than $b = 2$ to escape the convex hull $\mathcal{C}(\alpha)$. These perturbation approaches can be studied further in the future.

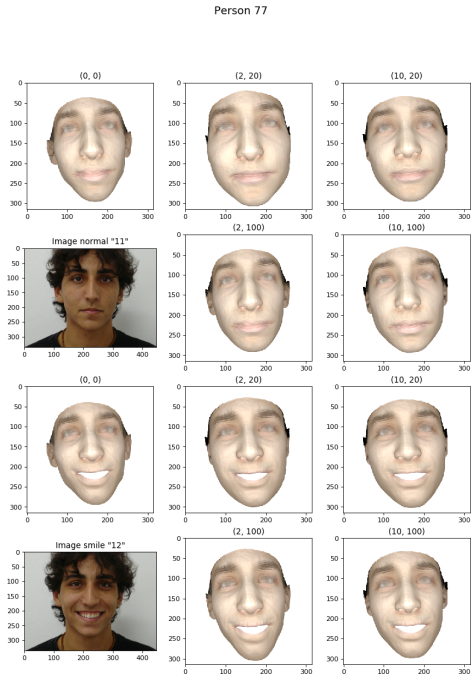
6 Appendix

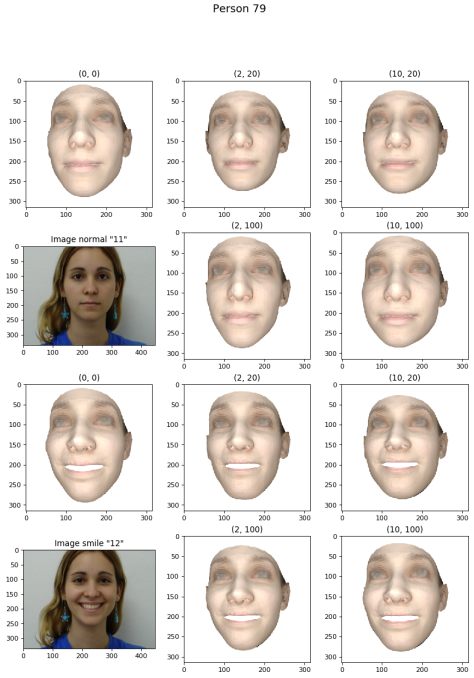
6.1 Result of $F^{(2D)}$ images

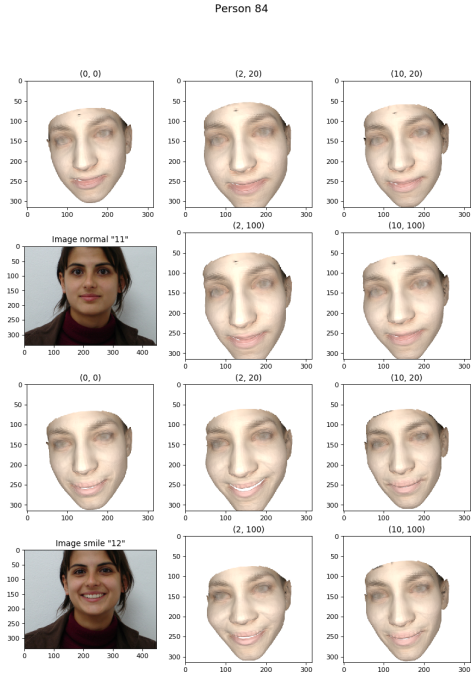
Images of 12 selected individuals with normal (upper half) and smile (lower half) expressions are perturbed by using 4 different (q, Q) perturbation approaches, namely $(2, 20)$, $(10, 20)$, $(2, 100)$ and $(10, 100)$. The $(0, 0)$ images mean no perturbation. Theoretically, $(10, 20)$ is expected to have the strongest perturbation, and $(2, 100)$ to have the weakest perturbation. Also note that the same random rotation angle (i.e. matrix R) are applied to different perturbation approach of the same individual.



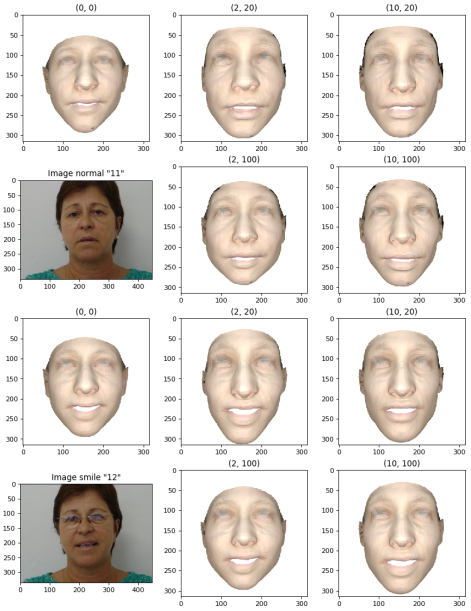

Person 30
(0, 0)
(2, 20)
(10, 20)
Image normal "11"
(2, 100)
(10, 100)
(0, 0)
(2, 20)
(10, 20)
Image smile "12"
(2, 100)
(10, 100)


Person 77
(0, 0)
(2, 20)
(10, 20)
Image normal "11"
(2, 100)
(10, 100)
(0, 0)
(2, 20)
(10, 20)
Image smile "12"
(2, 100)
(10, 100)

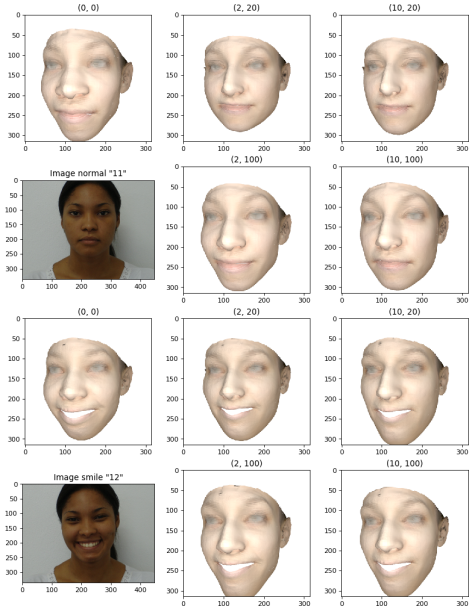

Person 79
(0, 0)
(2, 20)
(10, 20)
Image normal "11"
(2, 100)
(10, 100)
(0, 0)
(2, 20)
(10, 20)
Image smile "12"
(2, 100)
(10, 100)


Person 84
(0, 0)
(2, 20)
(10, 20)
Image normal "11"
(2, 100)
(10, 100)
(0, 0)
(2, 20)
(10, 20)
Image smile "12"
(2, 100)
(10, 100)

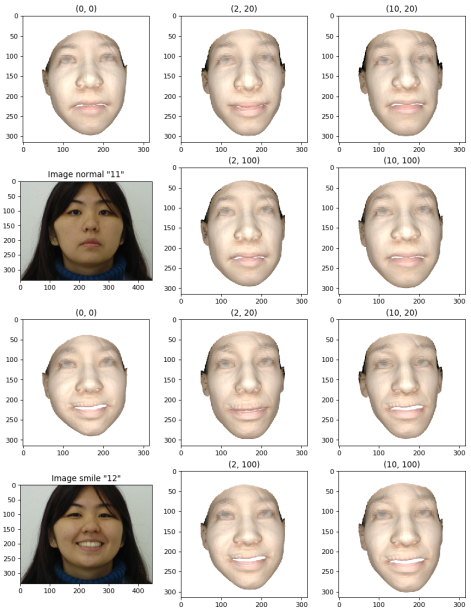
Person 143



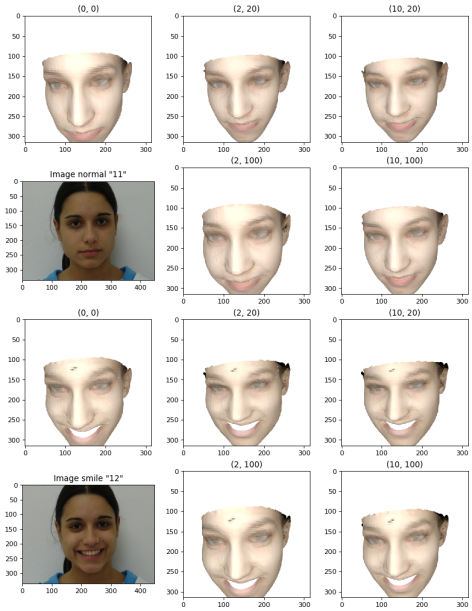
Person 157



Person 179



Person 200



6.2 Notations table

Notation	Space	Meaning
α	$\mathbb{R}^a$ or $\mathbb{R}^{99}$	Shape embedding of a human face
α'_i	$\mathbb{R}^a$ or $\mathbb{R}^{99}$	Perturbation of α_i
β	$\mathbb{R}^{99}$	Color embedding of a human face
η	$\mathbb{R}^{29}$	Expression embedding of a human face
ρ	$\mathbb{R}^6$	Pose embedding of a human face (i.e. rotation/translation)
$\mathbb{R}^{ab\times 1}$ (or $\mathbb{R}^{ab}$)	Sets	Set for vectors of length ab
$\mathbb{R}^{a\times b}$	Sets	Set for $a \times b$ matrices
x	$X = \mathbb{R}^{H\times W}$ or $\mathbb{R}^{H\times W\times 3}$	A B&W or RGB image which includes a human face
n	$\mathbb{R}$	Number of vertices of a human face under the 3DMM model
m	$\mathbb{R}$	Number of images
F	$\mathbb{R}^{3n\times 1}$	Shape array of a human face; 3 of $3n$ refers to 3D
T	$\mathbb{R}^{3n\times 1}$	Texture (i.e. color) array of a human face; 3 refers to RGB colors
S	$\mathbb{R}^{3n\times 99}$	Shape transformation from embedding space to 3D space
B	$\mathbb{R}^{3n\times 99}$	Color transformation from embedding space to color space
E	$\mathbb{R}^{3n\times 29}$	Transformation from expression embedding space to 3D space
$\dot{F}$	$\mathbb{R}^{3n\times 1}$	Face with expression
$\widehat{\dot{F}}$	$\mathbb{R}^{3n'\times 1}$	Dense version of $\dot{F}$
$\ddot{F}$	$\mathbb{R}^{3n'\times 1}$	Dense face with expression/wrinkles
F'	$\mathbb{R}^{3n\times 1}$	Perturbed face
$\ddot{F}'$	$\mathbb{R}^{3n'\times 1}$	Perturbed face with expression/wrinkles
$\dddot{F}'$	$\mathbb{R}^{3\times n'}$	Reshaped matrix from the vector $\ddot{F}'$
$F^{(2D)}$	$\mathbb{R}^{H'\times W'\times 3}$ (or $\mathbb{R}^{H'\times W'}$)	Projection of $\dddot{F}'$ onto a 2D image
$\ddot{T}'$	$\mathbb{R}^{3n'\times 1}$	Texture (i.e. color) for $\ddot{F}'$
$v(x)$	$\mathbb{R}^{68\times 2}$ (or $\mathbb{R}^{68\times 3}$ in 3D)	Landmarks detected in the image x
$e(x)$	$\mathbb{R}^{99}$ or other	Embedding of x encoded by an encoder, e.g. Deep3DMM or FaceNet
$h(\cdot)$	$X \to \mathbb{Z}_{\#(Person)}$	Identity classifier, especially the one used in evaluation
$\mathcal{L}^{LM}(\cdot)$	Embedding $\to \mathbb{R}$	Loss based on coordinates discrepancies of landmarks; Used by say Deep3DMM
$\mathcal{L}^{TP}(\cdot)$	Embedding $\to \mathbb{R}$	Triples loss among anchor, positive and negative images; Used by say FaceNet
$\mathcal{L}^{PX}(\cdot)$	Embedding $\to \mathbb{R}$	Loss based on discrepancies of pixels; Used by say MoFA
ω	Ω	Parameters of model
u_i	$\mathbb{R}$	Confidence provided by the CLNF landmarks detector
$\Pi_\rho(\cdot)$	$\mathbb{R}^{3n\times 1} \to \mathbb{R}^{68\times 2}$	$= Proj2D_\rho(Select_{68}(\cdot))$
p	$\mathbb{R}^2$	Position (x, y) of a point on the 2D image
$\dot{z}(p)$	$\mathbb{R}$	Depth of point p at the original z-buffer of face without wrinkles
$\ddot{z}(p)$	$\mathbb{R}$	Depth of point p at the z-buffer of face with wrinkles
$\Delta(p)$	$\mathbb{R}$	Bump map at point p
$\mathcal{X}(i)$	$Powerset(X)$	Set of images which belongs to the same person to image x_i
$\mathcal{P}$	Sets of individual	Set of all person (i.e. identities) in a database
$\mathcal{P}_i$	$Powerset(\mathcal{P})$	The i-th subgroup of identities
$\mathcal{P}(i)$	$Powerset(\mathcal{P})$	Subgroup that person i belongs to
P	$\mathbb{R}$	Number of elements in $\mathcal{P}$
k	$\mathbb{R}$	Refers to the k-anonymity to be achieved
q	$\mathbb{R}$	Number of pivots chosen for jumping, i.e. $q = \lvert\boldsymbol{q}\rvert$
Q	$\mathbb{R}$	Number of all candidate pivots, i.e. $Q = \lvert\mathcal{Q}\rvert$
$\boldsymbol{q}_\alpha$ (or $\boldsymbol{q}_i$)	$Powerset(\mathcal{Q})$	Perturbation set of α or α_i: Set of pivots chosen for perturbing α or α_i
$\mathcal{Q}$	Sets of individual	Pivot set: Set of all candidate pivots
$\mathcal{C}_G(\alpha)$	Regions	A convex hull of α
G	$\mathbb{R}$	Number of neighbors to form convex hull $\mathcal{C}_G(\alpha)$, i.e. $G = \lvert\mathcal{G}\rvert$
$\mathcal{G}_\alpha$	Sets of individual	Neighbor set: Set of neighbors to form convex hull $\mathcal{C}_G(\alpha)$
α_{mid}	$\mathbb{R}^a$ or $\mathbb{R}^{99}$	Centroid of $\boldsymbol{q}_\alpha$
δ	$\mathbb{R}^a$ or $\mathbb{R}^{99}$	The jump in the perturbation from α to α'

(Table continue next page)

(continue)

Notation	Space	Meaning
(x, y, z)	$\mathbb{R}^3$	Rectangular coordinate of a vertex
r	$\mathbb{R}$	Radius of a vertex in spherical coordinate
θ	$[0, \pi]$	Inclination angle of a vertex in spherical coordinate
ϕ	$[0, 2\pi]$	Azimuthal angle of a vertex in spherical coordinate
Γ	Regions	$= [0, \pi] \times [0, 2\pi]$, i.e. Set that (θ, ϕ) reside in
γ_{jk}	$Powerset(\Gamma)$	The bin $[\theta_j, \theta_{j+1}] \times [\phi_k, \phi_{k+1}]$ that separates Γ
Ξ_{ijk}	$\mathbb{R}^3$	The color of bin γ_{jk} for image i
w_{jk}	$\mathbb{R}$	The color interpolative extraction weight of γ_{jk} to compute color of $(\ddot{\theta}', \ddot{\phi}')$
ϱ	$\mathbb{R}^3$	Rotation vector
R	$\mathbb{R}^{3\times3}$	Rotation matrix
U	$\mathbb{R}^{3\times3}$	Cross product matrix of $\varrho/\|\varrho\|$
H	$\mathbb{R}^{4\times4}$	Extrinsic matrix of camera
$\mathcal{D}$	Sets of images	Database of images used in this paper
$\mathcal{M}$	Sets of images	Classifier training set: Images used to train the identity classifier $h(\cdot)$ in evaluation
$\mathcal{T}$	Sets of images	Target set: Images whose α would be perturbed
λ	$\mathbb{R}$	Regularization parameter
κ	$\mathbb{R}$	Criteria we use to define the top κ accuracy

References

[1] V. Blanz and T. Vetter. A Morphable Model For The Synthesis Of 3D Faces. *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 187–194, 1999. doi: 10.1145/311535.311556.

[2] J. F. Blinn. Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292, Aug. 1978. ISSN 0097-8930. doi: 10.1145/965139.507101. URL https://doi.org/10.1145/965139.507101.

[3] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniahy, and D. Dunaway. A 3D morphable model learnt from 10,000 faces. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:5543–5552, 2016. ISSN 10636919. doi: 10.1109/CVPR.2016.598.

[4] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014.

[5] F. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni. Faceposenet: Making a case for landmark-free face alignment, 2017.

[6] F. J. Chang, A. Tuan Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni. ExpNet: Landmark-Free, Deep, 3D Facial Expressions. *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, pages 122–129, 2018. doi: 10.1109/FG.2018.00027.

[7] H. Chi and Y. H. Hu. Facial image de-identification using identiy subspace decomposition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 524–528, 2014. ISSN 15206149. doi: 10.1109/ICASSP.2014.6853651.

[8] H. Chi and Y. H. Hu. Face de-identification using facial identity preserving features. *2015 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2015*, pages 586–590, 2015. doi: 10.1109/GlobalSIP.2015.7418263.

[9] N. H. Feldman and T. L. Griffiths. A rational account of the perceptual magnet effect. In *Proceedings of the annual meeting of the cognitive science society*, volume 29, 2007.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. Technical report, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[13] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[14] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[15] K. Kim, T. Baltrušaitis, A. Zadeh, L.-P. Morency, and G. G. Medioni. Holistically constrained local model: Going beyond frontal poses for facial landmark detection. In E. R. H. Richard C. Wilson and W. A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 95.1–95.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.95. URL https://dx.doi.org/10.5244/C.30.95.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] R. Or - El, G. Rosman, A. Wetzler, R. Kimmel, and A. M. Bruckstein. Rgbd-fusion: Real-time high precision depth recovery. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5407–5416, 2015.

[18] M. Pautov, G. Melnikov, E. Kaziakhmedov, K. Kireev, and A. Petiushko. On Adversarial Patches: Real-World Attack on ArcFace-100 Face Recognition System. *SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings*, pages 391–396, 2019. doi: 10.1109/SIBIRCON48586.2019.8958134.

[19] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. *6th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009*, pages 296–301, 2009. doi: 10.1109/AVSS.2009.58.

[20] A. Razavi, A. v. d. Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

[21] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. doi: 10.1109/cvpr.2015.7298682. URL http://dx.doi.org/10.1109/CVPR.2015.7298682.

[22] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. *Proceedings of the ACM Conference on Computer and Communications Security*, 24-28-Octo:1528–1540, 2016. ISSN 15437221. doi: 10.1145/2976749.2978392.

[23] Y. Shen, J. Gu, X. Tang, and B. Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[26] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez, and C. Theobalt. MoFA: Model-Based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2018-Janua:1274–1283, 2017. doi: 10.1109/ICCVW.2017.153.

[27] A. T. Tran, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3D morphable models with a very deep neural network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:1493–1502, 2017. doi: 10.1109/CVPR.2017.163.

[28] A. T. Tran, T. Hassner, I. Masi, E. Paz, Y. Nirkin, and G. Medioni. Extreme 3D Face Reconstruction: Seeing Through Occlusions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3935–3944, 2018. ISSN 10636919. doi: 10.1109/CVPR.2018.00414.

[29] A. Wood, M. Altman, A. Bembenek, M. Bun, M. Gaboardi, J. Honaker, K. Nissim, D. R. O'Brien, T. Steinke, and S. Vadhan. Differential privacy: A primer for a non-technical audience. *Vand. J. Ent. & Tech. L.*, 21:209, 2018.

[30] J. Zhang, Y. Yan, and M. Lades. Face recognition: eigenface, elastic matching, and neural nets. *Proceedings of the IEEE*, 85(9):1423–1435, 1997.

[31] S. Zhou, M. L. Gordon, R. Krishna, A. Narcomey, D. Morina, and M. S. Bernstein. HYPE: human eye perceptual evaluation of generative models. *CoRR*, abs/1904.01121, 2019. URL http://arxiv.org/abs/1904.01121.

[32] X. Zhu, X. Liu, Z. Lei, and S. Z. Li. Face alignment in full pose range: A 3d total solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):78–92, Jan 2019. ISSN 1939-3539. doi: 10.1109/tpami.2017.2778152. URL http://dx.doi.org/10.1109/TPAMI.2017.2778152.