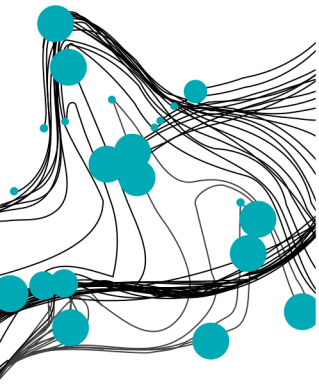
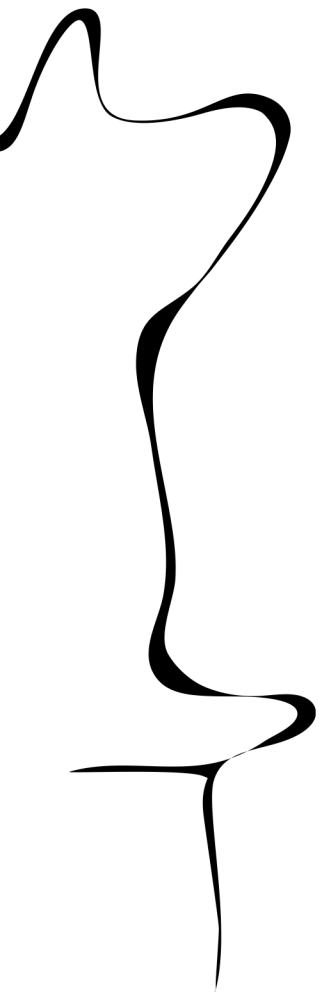


DMB

DATABASE MANAGEMENT
AND
BIOMETRICS



EXPLORING IDENTITY MATCHING FOR LOW QUALITY IMAGES WITH THE HELP OF A PIPELINE FOR SYNTHETIC FACE GENERATION

Marko Groffen BSc.

MASTER ASSIGNMENT

Committee:
dr.ir. L.J. (Luuk) Spreeuwers
dr. C.G. (Chris) Zeinstra
dr.ir. S.G.A. (Ghayoor) Gillani

December, 2022

Data Management and Biometrics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

UNIVERSITY OF TWENTE.

Master Thesis for the University of Twente

Exploring identity matching for low quality images with the help of a pipeline for synthetic face generation.

Marko Groffen BSc.

Abstract: In this paper we propose a pipeline for synthetic low resolution face generation as an alternative to image downsampling and real-world low resolution datasets for use in forensic cases. The goal of the pipeline is recreating physically accurate low resolution face images in a 3D space. We were able to incorporate a state-of-the-art conditioned machine learning algorithm to generate a realistic synthetic high resolution gallery dataset, by combining StyleGAN2 and attribute based latent space exploration. Using single image 3D reconstruction and a physically based renderer, an identity preserving pipeline was introduced that allows for one-to-one gallery to low resolution probe dataset generation, while enabling flexible pose, lighting, resolution and compression adjustment. Using volumetric path tracing, subsurface light scattering within human skin was emulated. To get further insight into our pipeline output, facial recognition experiments were conducted using state-of-the-art commercial and open-source facial recognition software and super-resolution upscaling using a convolutional neural network. Comparison to the real-world face image dataset SCFace was also conducted to test for potential applicability of our pipeline in forensic cases. Lack of accurate optical aberration and sensor characteristics resulted in a significantly different facial recognition performance on our synthetic dataset, making current application of our pipeline in forensic scenarios unfit. Thorough description of design choices and background make our research however an interesting stepping-stone for future research.

1 Introduction

The identification of people has been a problem for many years in the world of forensics. Many places nowadays have cameras that can help identifying people, for example security cameras in shops to help identify robbers, but a low image quality is often still a problem when it comes to identification. With the increased potential of artificial intelligence and machine learning, facial recognition can be a powerful aid for identification. However, the facial recognition is only as good as the training data fed to the system. And especially in the forensic casework, problems in training data can have large consequences. Currently, there are not a lot of real-world low resolution test databases and the ones that are available, for example SCFace [1], are a few years old. The lawful restrictions on gathering biometric data, for instance the EU restrictions on gathering privacy sensitive data [2] [3], makes it difficult to compose new training dataset. Some proposed methods for low resolution identification have therefore resorted to downsampling higher resolution images to create training and test datasets. However, this has influence on the reported performance compared to real low resolution databases, as discussed in [4]. Advances in machine learning, on the other hand, might give new opportunities in creating realistic and physically accurate low resolution test databases for forensic identification.

Powerful machine learning models for image generation have been proposed in recent years, where in particular the generation of face images has seen a lot of interest from this field of research. The generation of photo-realistic face images of non-existing people has also become a reality with the usage of for example Generative Adversarial Networks [5]. There is however a difference between photo-realistic and physically correct images. Certain physical phenomena, such as diffraction, lens distortion and sensor colour filters, have a larger impact on low resolution images and are therefore important to correctly model. When the successful generation of physically correct synthetic face images is however viable, it might be usable to create datasets with tailored specifications for identification matching a real forensic setting. To test the potential of such datasets, this research will try to propose a pipeline to generate these synthetic face images, using a 3D space to generate the low resolution images. The pipeline will have changeable components and the proposed pipeline will be a proof of concept that could form a basis for future research. Verification on low resolution images, based on the generated synthetic face images from the pipeline, is then performed through various methods to explore the effectiveness of the synthetic dataset.

1.1 Research Questions

To explore the creation of the image pipeline, we split the research into two parts. The first part focuses on the actual creation of the pipeline and the subsequential design choices, which are answered by means of literature research and practical implementations. This also includes the high resolution synthetic images that will be needed for training/facial recognition purposes:

- What steps/processes are of importance in a 3D pipeline for generating realistic synthetic face images comparable to images typically found in forensic use cases?

1. What are the advantages/disadvantages of using a 3D over a 2D pipeline to generate low resolution face images?
2. Which renderer is suitable for generating images with physically accurate lighting conditions? What is the influence of diffraction of light to create physically accurate low resolution images?
3. What are other influences of a lens, sensor characteristics and image storage on the degradation of realistic low resolution images? Can we implement these influences into our pipeline?
4. Which models can be used to generate 3D face structures?
5. How can subsurface scattering of light within human skin be accurately modeled?
6. Which model for synthetic face generation is suitable?

The second part has its focus on identity matching performance with the data created in the previous steps:

1. What is the impact of face images with an interpupillary distance of a few pixels on facial recognition?
2. How do low resolution and super-resolution biometric comparisons perform with our very low resolution synthetic dataset, while varying pose, lighting conditions and compression?

3. How do low resolution facial recognition methods perform with real-life low resolution face datasets compared to our synthetic datasets?

The results of the last sub-question can give an insight into an overarching question of our research: *Does the proposed pipeline generate low resolution face images that are comparable to a real-world dataset, making them usable in a forensic setting?*

2 Background

2.1 Low Resolution Face Images

As mentioned in the introduction, there is not a wide variety of low resolution facial image datasets. However, when talking about low resolution, there are of course different ways to interpret this term. Therefore, we will establish what we understand under low resolution face images.

Resolution in general describes the amount of image detail. In digital images, the resolution is often defined by a pixel count. This can be done by giving the width and height of an image in pixels, for example 240×300, which means an image that is 240 pixels wide and 300 pixels high. Another way to give the resolution pixel count measurement, is to quantify it for a specific unit of length. Often "pixels per inch" (PPI) or "dots per inch" (DPI) are used, where PPI describes the amount of pixels both horizontally and vertically per inch of an image [6]. For specifically face images, the pixel distance between the eyes, or more specific the pixel distance between the pupils, can be used to quantify the resolution. For example, the International Civil Aviation Organization (ICAO) uses the pixel distance between the centre of the eyes to give a resolution requirement for face images in machine readable travel documents [7]. Going forward, we will use this interpupillary distance (IPD) as our resolution metric.

In literature, there seems to be no generalized definition of "low", when low resolution face images are mentioned. For example, in [8] the low resolution images are described as having an IPD between 2 and 8 pixels, while in [9] low resolution is defined as images with an IPD of 6 pixels and [10] uses an IPD of 7 pixels. This inconsistency can also be observed with large face detection challenges, like WIDER FACE [11] and FDDB [12]. WIDER FACE labels faces with a height of 10 pixels or less, which roughly gives a IPD of 4 pixels for frontal faces, with an "ignore" flag in their test dataset, because they are too difficult to recognize [11]. FDDB excludes faces with a height of 20 pixels, roughly a IPD of 8 pixels for frontal faces, for similar reasons [12]. However, the performance of facial detection algorithms on these datasets has increased since their release, with the average precision (AP) on FDDB having reached 0.990 in 2017 and the AP on the WIDER FACE dataset with the lowest resolution reaching 0.921 in 2020. This means the solutions for detecting faces, even faces with a relatively low IPD value, have improved over the years. On the other hand, detection is only part of the recognition pipeline and the problem is not yet fully solved [13]. Also, the very low resolution images below the dataset thresholds still yield considerable deterioration in recognition performance [14]. Furthermore it can be noted, that besides the difference in definition, there also seems to be a significant difference in performance for datasets with self-proclaimed low resolution face images.

For this paper, we follow the same definition of low resolution as proposed in [15] by Yuxi et. al., where there is made a distinction between upper low resolution (ULR), moderately low resolution (MLR) and very low resolution (VLR). Figure 1 contains the proposed resolution scale. The scale includes four biometric standards. The first two are ISO/IEC 19794-5:2005 [16] and ANSI/INCITS 385-2004 [17]. They describe an example of proper face position in an image, where the IPD of the face is about 50 pixels [18]. This is used as the separation between high and low resolution. The separation between ULR and MLR is based on the European norm EN 50132-7 [19], which describes the recommended minimum resolution for CCTV objects as having an IPD of about 25 pixels. The ICAO requirement for machine readable document face images, as mentioned above, is also added [7]. Their required IPD is about 90 pixels. ULR will not pose a big challenge for most existing facial recognition methods, while MLR is more difficult and methods designed for low resolution should outperform high resolution facial recognition methods for MLR.

The transition point between moderately and very low resolution is

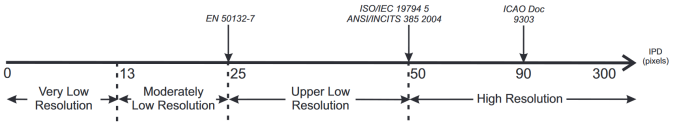


Fig. 1: Face image resolution scale

set at an IPD of 13 pixels. If the IPD gets lower than this, we expect poor results from most facial recognition methods to identify a person. Low resolution datasets with images from real-life uncontrolled settings like HumanID [20] and SCFace [1] do also contain images that have IPDs below this threshold.

The threshold 13 pixels for VLU might be slightly higher than expected, given the before mentioned papers in this chapter seem to use an IPD between 2 and 8 pixels. The goal of this paper is to create a pipeline that simulates real-life low resolution face images, and here resolution is not the only factor that degrades facial recognition performance. In other research, often down-sampled high resolution images are used to create low resolution test datasets. However, real low resolution images do perform significantly worse than down-sampled high resolution images when performing facial recognition [21] [22]. Therefore, simulating these real world effects is important if we want to compare our results to real world low resolution datasets, like HumenID and SCFace, and our separation between MLR and VLR is set at a slightly higher IPD value.

In literature, low quality is sometimes also used instead of low resolution. Low quality, however, is a broader term that takes both low resolution and other factors, like blur, heavy weather or (partial) occlusion, into account [23]. From this point forward, if we talk about low resolution face images, we mean VLR face images that either include or simulate realistic camera effects, but not motion effects or occlusion.

2.2 Facial Recognition Methods

Facial recognition is the automated recognition of an individual based on a digital face image. In facial recognition, there is a separation between two types of input image: gallery and probe images. The gallery image is a face image labelled with a known identity. The probe image on the other hand is an input image from someone with an unknown identity to the system. The goal of the facial recognition system is either verification, where a probe and gallery image are compared and the output is a decision whether the images are from the same person, or identification, where a probe image is compared to a list of gallery images and the output is the identity of a matching gallery image. The low resolution images that are discussed in the previous chapter are the probe images. Gallery images do not have to be of poor quality and are often not in the form of e.g. mug shots or passport photos.

In general, the facial recognition system can be split up in three main steps: face detection, feature extraction and face recognition [24]. The detection step often not only detects the existence of a face in an image, but also normalizes the face by e.g. landmark localization, image alignment and colour adjustments. In the next step, the useful features from the face are extracted. These features are abstract measures that need to be stable, so present in most images, but also distinctive so they vary from person to person. The final step, the recognition, is done by matching the extracted features and giving a score based on resemblance [25] [21].

Most facial recognition systems require both the probe and gallery image to have the same resolution at input. In our scenario however, we are dealing with low resolution probe images and high resolution gallery images. There are generally speaking three methods to solve this problem:

- Super-Resolution
- Low-Resolution
- Mixed-Resolution

Part of the main focus for our research is super-resolution, but all three methods are discussed in more detail below to give a full background.

2.2.1 Super-resolution

The first method solves the problem of different resolutions by upsampeling the low resolution probe image to a higher resolution. This technique is generally known as super-resolution (SR). Two main SR techniques can be distinguished: using a single image to predict the higher quality output, better known as single image super-resolution (SISR), or using multiple images to achieve this goal, known as multiple image super-resolution (MISR). MISR can often be used when multiple images from similar points of view are taken, either from multiple cameras or from a video. In the case of video, the separate frames can be used [26]. MISR often outperforms SISR under a proper inter-image alignment, since less information is estimated from the input images compared to the single input image used with SISR [27]. With MISR sometimes lost frequency components can be recovered, which is not possible with SISR [28]. SISR on the other hand has a high efficiency and is therefore more popular in practice, since MISR has a large computational demand [27] [29]. For this reason we will focus on SISR methods for this research. In further research MISR methods could be used to predict more refined potential of our image pipeline.

The typical SISR framework is presented in figure 2. Solving the recovery part of the framework is an ill-posed problem, since there are many possible high resolution solutions to a single low resolution input image.

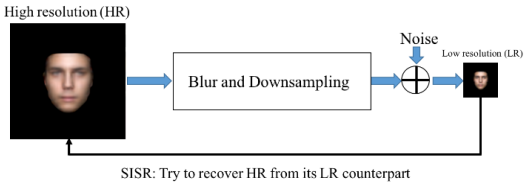


Fig. 2: Typical SISR framework

Interpolation

One of the more widely used (up)scaling algorithms for images is bicubic interpolation. It is for example the standard image scaling algorithm used in Photoshop. It is the more advanced algorithm compared to nearest neighbour and bilinear interpolation, which are both also widely available in image editing software. Nearest neighbour interpolation is the simplest of the three, where every new output pixel after the scaling is replaced by its nearest input pixel. Bilinear interpolation takes a slightly more advanced approach, where it uses the 4 neighbouring points (see figure 3). Linear interpolation between the top two known points, A and B, and the bottom two known points, C and D, is done. This creates two new points, AB and CD, between which linear interpolation is applied again to find the final value at point X. Bicubic interpolation is a further extension of bilinear interpolation. Instead of using 2 points to interpolate linearly, 4 points are used to draw a 3^{rd} degree polynomial, known as cubic spline interpolation in a 1-D space. For a 2D image, this requires 16 neighbouring points, as shown in figure 3. Cubic spline interpolation is performed for 4 rows, after which a final cubic spline interpolation is performed on the 4 new interpolated points.

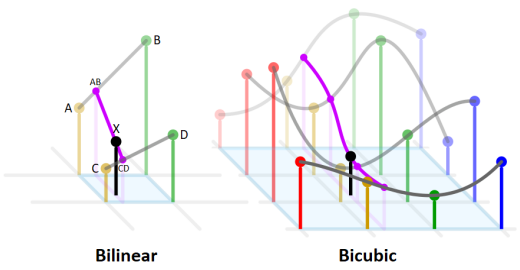


Fig. 3: Bilinear (left) and bicubic (right) interpolation for image scaling

Learning-based

Advances in machine learning opened the way to learning-based SISR algorithms that quickly outperformed the interpolation SR methods. Learning-based methods use machine learning algorithms to analyze relationships between corresponding low and high resolution images based on a sizeable image dataset. Early methods use for example sparse coding/representation, like the work by Yang et al. [30] [31]. Due to the rapid advances in deep learning, superior advantages over these early techniques have been obtained [29] [32]. Deep learning is a machine learning technique that is inspired by biological neural networks. The term "deep" is a reference to the multiple non-linear processing layers within the network that lie between the input and output layers.

CNN

In the early days of applying deep learning to SR, most methods were based on Convolutional Neural Networks (CNN), a deep learning model which uses convolution, often instead of matrix multiplication, in one of it's neural layers [32] [33]. The work by Dong et al. demonstrates that previous sparse-coding methods are practically equivalent to using a CNN, resulting in SRCNN [34].

The overall architecture of SRCNN is shown in figure 4. As can be seen in the figure, the SRCNN model first implements a pre-processing step. The pre-processing step consists of upscaling the input LR image to the desired spatial dimensions of the output, using bicubic interpolation. The goal during training is to learn a mapping from this upscaled LR image to a HR output that is as similar as possible to the ground truth HR image. The trained CNN model itself consists of 3 layers. The first layer, patch extraction, extracts (overlapping) patches from the upscaled LR input and represents each patch as a high-dimensional vector, which comprise a set of feature maps, or filters. The second layer nonlinearly maps each high-dimensional vector onto another high-dimensional vector, which conceptually represents a HR patch. These mapped vectors form another set of filters. The final reconstruction layer bundles the HR patches to create the HR output image [34].

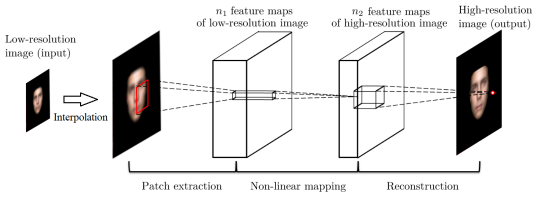


Fig. 4: SRCNN Architecture

SRCNN has a few significant attributes. First of all, the goal is not to obtain a high accuracy on a dataset during training, but to find the before mentioned filters. Once these filters are found, optimizing a loss function on a per-image basis is not required. a simple forward pass is enough to create the SR output images. This means the SRCNN is completely end-to-end, meaning no intermediate steps are needed to create the output. It is also fully convolutional, so an input image of any size (as long as patches can be extracted) can be fed to the model.

SRCNN is considered pioneering and a foundation on which many other CNN SR models are built [29] [32]. For example, changing the CNN to be bi-channel (BCCNN), where a separate information channel for the raw input image and the face representations is integrated to reconstruct face images [35]. SRCNN-IBP introduces iterative back projection as a post-processing step to the CNN to gain reconstruction performance [36]. Wavelet-SRNet uses wavelet transforms to avoid the smoothing out of high frequency details after SR [37]. In recent years, concepts like attention modules have also been introduced to the CNN to focus on more important feature information. For example SPARNet, where spatial attention designed for face SR is implemented [38].

GAN and Other Advanced Models

CNN methods utilize pixel-wise loss to reconstruct images. But maybe the exact reconstruction of pixel-level details, such as the direction of hairs, is not always necessary as long as the overall details are perceived as realistic. This led researches to introduce GAN-based SR methods, inspired by GANs with the ability to construct realistic face images from scratch with high detail, like StyleGAN [39]. Generative Adversarial Networks, GANs for short, are generative models that make use of an adversarial process. In a GAN, two models are trained simultaneously: a generative model and a discriminative model. They are in conflict with each other in the form of a zero-sum game. The generative model tries to capture a data distribution, while the discriminative model tries to estimate whether its input came from the generator or the true data distribution. An early example of this model being applied in SR can be found in UR-DGN [40], where the discriminative model tries to distinguish a real HR face image from a SR one, while the generative model generates SR images to fool this discriminative model and find a distribution of real HR face images. More recent examples include PCA-SRGAN, where firstly decomposed face images are progressively fed to the discriminator instead of the whole image [41], or SPGAN, which uses a supervised pixel-wise GAN that can also resolve VLR images [42]. Researchers have also directly made use of the facial generation networks like StyleGAN. By using a pretrained GAN like SytleGAN as a generative prior, by for example embedding the pretrained GAN into their own network like GPEN [43], the generation power of these facial generation networks can be used to improve SR reconstruction.

Instead of a generative prior, human face specific characteristics can also be used to support the deep learning model. Known as prior-guided face SR, information like facial boundaries, landmarks or heatmaps can be extracted before, during or after reconstruction to construct images with clearer facial structures [32]. ATSENet is such a model, which uses a facial boundary heatmap as prior to reconstruct especially VLR face images [44]. SR methods specifically catered towards the facial recognition application have also been introduced. Known as identity-preserving SR, the goal is to maintain consistency of the identity between the generated SR and a real HR image. Most methods consists of a pretrained facial recognition network and a SR model. The SR model reconstructs the input LR face image, which is fed into the facial recognition network to find identity features. Concurrently, a corresponding HR image is also fed into the facial recognition network, finding its identity features. The identity features are used to calculate a identity loss, for which a minimum is tried to be achieved during training [32]. An example of such a identity preserving SR model with the specific application for VLR images was presented by Jones et al [45]. A big disadvantage of the usage of facial recognition networks is the need for well labeled training datasets. A potential solution is so called pairwise data-based methods, which take advantage of the contrast

and similarity between identities to retain identity consistency, even for weakly-labeled datasets [32]. A combination of various deep learning models has also been put forward in order to combine the advantages of the different models. ATMFN is such a SR model, which combines a CNN, GAN and RNN (Recurrent Neural Network) to create an ensemble of deep learning-based methods [46].

It has to be noted that in practice, SR is not yet used in real forensic science. This is due to the addition of information to the original image in the SR process. Since this addition can not be guaranteed as fully accurate as of now, images enhanced with SR are seen as manipulated and are no longer authentic, even when the visual result is satisfying [47]. Work by Li et al. even demonstrates that bicubic interpolation might work better for upscaling than SR for highly degraded and VLR images, since various SR methods introduce artifacts in an attempt to sharpen details [22].

2.2.2 Low-resolution

The second method to compare a probe and gallery image of different resolutions, is to downsample the high resolution gallery image. However, the biggest problem here is that the high-frequency information from the image is lost in the downsampling process. And this high-frequency information is of use for facial recognition. Also, extracting features from low resolution images is more difficult because there is less information and therefore less features in general. Despite this, quality/resolution robust feature extraction methods have been proposed. These methods are mostly texture- or colour-based and are training free, which makes them faster to execute than learning-based methods [14]. Modifications of Local Binary Pattern (LBP) features [48], improving Local Phase Quantization (LPQ), a method based on quantizing the phase information of the local Fourier transform to characterize underlying image textures [49], and a method based on Histogram of Oriented Gradients (HOG) features and Gabor Filter [50] are amongst proposed methods. These methods are however very sensitive to pose or illumination occlusion and changes in expression, and features often have to be hand-crafted [14].

2.2.3 Mixed-resolution

A third method, besides super-resolution and LR robust features, is match LR to HR face images directly, by learning a common or unified space from both the LR and HR images. Most methods try to find discriminative and class-separable features when mapping the LR and HR images to the unified space. When projected into this learned space, similarity measurements can be performed to carry out the recognition task.

Practical implementations that illustrate this method include the work by Biswas et al., where a multidimensional scaling method is introduced. They simultaneously embed the LR and HR images into the unified space so that the distance between the LR and matching HR counterpart is approximately the same as the distance between equivalent HR images. The method is extended to work for difficult pose and lighting conditions too [51]. To bridge the deviation between LR and HR image domains, Wei et al. use a linear transformation with the constraint of sparsity for mapping [52]. The method by Moutafis et al. jointly learns two semi-coupled bases, using HR images to learn a basis and distance metric with increased class-separation and LR images to learn a basis and distance metric that can map LR images to the class-discriminated matching HR images [53]. Inspired by this method is the mixed resolution facial comparison presented by Peng et al. The MixRes classifier transforms the input probe and gallery samples to a common lower dimensional space, from which the log-likelihood ratio is determined for the probability that the samples originate from the same individual, over the probability the samples originate from different individuals. This method is particularly suitable for cases where the probe and gallery images have different resolutions, but can also be trained for same resolution inputs [54].

2.3 2D versus 3D

In order to create the synthetic face images, a method of generation has to be established. One of the first goals we set for ourselves, is that the images have to be physically accurate to properly model the real-world effects that impact the image quality. As describes in section 2.1, simply downsampling a high resolution image will gather different results in facial recognition than real low resolution images, due to various optical phenomena within a scene. The ability to change certain features in the scene, like lighting and the pose of the face, are also important criteria to allow for the creation of custom datasets. And since our research flows from the lack of real world datasets, mass generation of images is also required to create an actual dataset of substance.

The first distinction we can make, is whether we want to generate the images in a 2D or 3D space. Of course, in the end all images are 2D representations, but here we try to distinguish between creating certain features like pose and lighting in a 2D or 3D space.

Creating photo-realistic looking 2D images of synthetic faces has become a possibility in recent years due to advances in machine learning. For example the website Thispersondoesnotexist.com [55], a website that generates a new realistic image of a person that does not exist in the real world every time you refresh the page. The generated images where deemed so realistic, that the website was even featured in mainstream media [56][57][58]. The machine learning algorithm behind this website is StyleGAN2 [5], a Generative Adversarial Network (GAN) developed by NVidia to generate images of faces, cars or animals, depending on the training data. A more advanced explanation of the StyleGAN can be found in section 2.4.4. In order to use a 2D pipeline to generate the face images, a machine learning algorithm like StyleGAN2 could be used to generate a higher resolution image, while controlling certain attributes to make the GAN conditional. In a post-processing step, realistic aberration and other artifacts could be introduced in the form of e.g. filters while also downscaling the image. However, the first problem with this method is, that in this entire pipeline the effects of light on the scene and camera are not based in physics, but are merely visually approached. This is partly due to the usage of machine learning. The way light hits the face in the output images is based on how the model thinks light from a certain angle looks, not on the real behaviour of light. An extra problem can occur when face components are inaccurately estimated, causing lighting artifacts [59]. Using filters and image transformations to emulate light passing through a lens system and hitting a camera sensor, will on the other hand leave out more unexpected behaviours that do occur in real images. Another problem lies in changing the pose. StyleGAN is an unconditional GAN, so it requires latent space exploration to allow for an attribute manipulation like changing pose [60][61]. Editing along the latent space is however not always smooth, which will sometimes morph the relative location of features like the nose, mouth and eyes while changing pose. Visually this is often not directly noticeable, but it might be for facial recognition where localizing these features and subsequently matching them to a gallery picture are a big part of the recognition system. Conditional GANs are introduced that focus on identity preservation by disentangling identity and non-identity factors. But these are often limited to changing only one attribute, like pose [62][63], or lighting [59]. If we want to create a dataset that is comparable to real images, while also keeping as much control as possible on the image attributes, these problems are significant hurdles.

The other approach is creating a 3D scene to create the low resolution images. Here the 2D synthetic images described above can be used to generate 3D face models. These 3D models can be placed in a scene where pose and lighting can be changed without adjusting the 3D model's identity. Using a physically based renderer, realistic material, camera and light properties can be rendered. We therefore deem this 3D approach as more suitable for our goal, and will explore it in more depth in the next sections.

2.4 3D Pipeline

In order to describe the required components of a 3D image generation pipeline, we will split a 3D scene into four parts, as seen in figure 5. The 3D scene consists of a camera, that can be split up into a lens(system) and a sensor, a 3D model of a face and the renderer that will take all attributes and simulate the behaviour of light in this scene. For each part, the theoretical background and design choices will be discussed. The practical implementation of the components will be discussed in the methodology section.

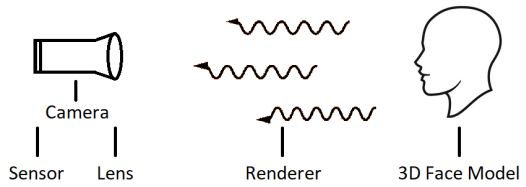


Fig. 5: Different components in our 3D scene

2.4.1 Renderer

As a basic definition, rendering is the projection of a scene from a certain viewpoint onto an imaginary image plane by means of a computer program. The scene is often described in a file, where information like object geometry, viewpoint, lighting, shading and textures are stored. The rendering program then applies a rendering algorithm to project all the information in the scene into a picture, or the render. Simply tracing all light particles in a scene is hugely impractical and would require huge amounts of time and computer power. Therefore, in modern computer graphics there are two basic methods to approach the rendering of a scene. The first one is rasterization or scanline rendering, where all objects of the scene are geometrically projected into the image plane and colour all the pixels accordingly. This method does not allow for advanced optical effects. The second method is ray tracing, where rays are traced from the viewpoint through each pixel and determine the objects that ray hit. The colour of the pixel is determined by shading the object the ray hits at that pixel. Often more than one ray is traced per pixel, after which all the rays are sampled in order to determine the final colour of an image pixel. Ray tracing can in turn be divided into ray casting and ray tracing. Ray casting does not trace rays further than the first object they hit. ray tracing on the other hand describes a more general method of tracing light paths in a scene and can be used to generate a photo-realistic render [64][65]. This is the method of interest for our research.

Rendering Equation

In general, the amount of light that reaches the viewpoint from an object is given by the sum of light emitted by the object and reflected light. This idea can be expressed in an integral equation generally known as the rendering equation. The equation is shown in equation 1. $L_\mathrm{o}(\mathbf{x}, \omega_\mathrm{o})$ is the outgoing radiance from point $\mathbf{x}$ in direction ω_o, $L_\mathrm{e}(\mathbf{x}, \omega_\mathrm{o})$ the emitted radiance from that point in the same direction and the integral represents all the incident radiance from all directions on the sphere Ω around point $\mathbf{x}$.

$$L_\mathrm{o}(\mathbf{x}, \omega_\mathrm{o}) = L_\mathrm{e}(\mathbf{x}, \omega_\mathrm{o}) \ + \int_\Omega f_\mathrm{r}(\mathbf{x}, \omega_\mathrm{i}, \omega_\mathrm{o}) L_\mathrm{i}(\mathbf{x}, \omega_\mathrm{i})(\omega_\mathrm{i} \cdot \mathbf{n}) \, \mathrm{d}\, \omega_\mathrm{i} \tag{1}$$

Solving the rendering equation analytically is only possible for very simple scenes, so most rendering algorithms rely on a numerical approach. And to render a scene realistically, a solution has to be found many times to find the overall lighting, or global illumination, of the scene. Particularly to simulate indirect lighting caused by light bouncing between objects. Local illumination on the other hand describes the responds of a single surface to light [64][65].

Solving the Rendering Equation

The two most common ways to find a solution to the rendering equation are the finite element method and Monte Carlo method. The finite element method is known as the radiosity method, where for global illumination each object surface is divided into small patches and the solution is found by modelling the transfer of light between these patches. The radiosity method, however, is very memory and time consuming and will only allow for diffuse reflections [64]. The Monte Carlo methods rely on repeated random sampling to find a solution for the integral part of the rendering equation. Finding the global illumination is inherently recursive with the rendering equation, as all the incoming light at one point is comprised of all previous light rays that hit that point, also described with the rendering equation. Instead of integrating over all incoming rays on sphere Ω, only certain points are sampled and only those rays traced to their origin. This drastically reduces the amount of rays that have to be traced, especially when the amount of samples is reduced the further the ray is traced. It has to be noted however, that a very large amount of samples is still necessary to create an accurate picture, with sometimes billions of rays being traced for moderately complex scenes.

Monte Carlo Methods

The first algorithm that was introduced using this Monte Carlo method was path tracing. Here the rays are traced from the viewpoint/camera into the scene until a light source is hit. When there is a lot of big light source in the scene, this is a viable solution. However, with darker scenes, tracing random rays is quite wasteful. One can also choose to trace the rays from the light source to the camera. But than a similar problem occurs, where you need to get enough rays that reach the camera. Using bidirectional path tracing, this problem is solved by tracing both rays from the camera and from the light source and the vertices are connected in the middle by a specific sampling strategy. Photon mapping is a similar method, but instead of trying to connect the paths from the light source and camera directly, the algorithm is split into two phases. First, rays from the light source and camera are traced independently until a certain criterion is met. Then in the second phase, the radiance at each point of a surface is determined based on all the rays that passed that point. Photon mapping is great for simulating caustics, concentrations of light due to refraction, and can simulate light scattering. But unlike path tracing, this algorithm is statistically biased. Light scattering can also be simulated with volumetric path tracing. For normal path tracing, the $f_\mathrm{r}(\mathbf{x}, \omega_\mathrm{i}, \omega_\mathrm{o})$ term of the rendering equation is the bidirectional reflectance distribution function (BRDF), which describes the reflection of light from ω_i to ω_o at point $\mathbf{x}$. This, however, only describes reflection on opaque surfaces. With volumetric path tracing, the BRDF is extended to model light entering at the surface and scatters internally before exiting at a different point. This way effects like fog and smoke can be rendered, but also subsurface scattering. Subsurface scattering is of particular interest to us, as this process also describes the reflectance of light on human skin [64][65]. Therefore, we chose the volumetric path tracing algorithm to render our scene. A more in depth description of subsurface scattering can be found in section 2.4.4.

2.4.2 Lens

In order to project a scene onto a much smaller image sensor, a lens system is used to converge the incoming rays towards the surface of the sensor. In this section, we discuss the real-world lens system we try to emulate and the effects of lens optics on the output image.

Lens System

There is of course a wide variety of camera types available, each with different optical specifications and therefore different effects on the output image. Hence we take a look back at the problem definition, where security cameras are a big source of (low quality) images within forensic cases. From this, it would logically follow that we try to model the camera of our pipeline after a security camera. So,

we use a real security camera as reference, the Axis P1435-LE [66]. Our choice for this camera mainly lies in the fact that the camera's datasheet gives a good overview of camera parameters, e.g. the F-number, lens focal length and field of view in degrees, that can be used to calculate more in-depth camera specifications. This is in no way an endorsement of this specific brand or camera, as it merely poses to give realistic camera parameters that are in ratio.

There are three main ways to model a camera that could be of interest for us: a pinhole camera, a thin lens camera and a realistic camera. The pinhole camera is a camera without a lens and a single point as aperture, and in renderers is often simulated as an ideal pinhole camera with an infinite depth of field. The scene can either be projected with a perspective transformation, based on a given field of view, or with a orthogonal transformation, were parallel lines within a scene are preserved. All real-world cameras with a lens give a perspective projection. The thin lens camera simulates a lens with a thickness of zero to purely simulate blurriness in a scene due to depth of field. Here the lens size and focal distance can be adjusted to change the part of the scene that is in focus. A realistic camera can also be modeled, where the different components of a lens, like lens thickness, radius of curvature and refractive index, could for example be described in a table [65].

Optical Aberration

Once a real lens is introduced into the pipeline, there will not be a ideal projection of the scene onto the camera sensor. One of the main problems is aberration. Aberration is an optical phenomenon, where not all rays converge to the same focal point once they passed through one or more lenses. There are different variants of aberration, shown in figure 6.

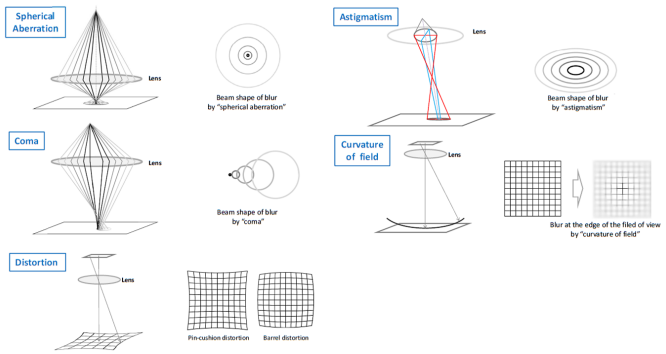


Fig. 6: Types of aberration and distortion

Spherical aberration occurs due to the sphere shape of a lens, causing the focal point to differ for rays that hit the outer part of a lens compared to the inner part of a lens. Coma aberration is similar, but is caused by the difference in refraction of different lens zones when the rays come in from a different angle of incidence. A lens can also have imperfections, that can cause the lens to have an unexpected refraction of rays. A larger, but common, imperfection is the lens not being perfectly spherical. This causes astigmatism, where the focal point from rays in the horizontal plane are different than in the vertical plane. These forms of aberration can be compensated with multi lens systems. And it is likely a security camera has a multi lens system, since even optical systems like cheap phone cameras consist of multiple lenses. Regardless, most camera lenses are catered to their application. And since specialized lenses can become expensive very quickly, these forms of aberration will most likely be present to at least some extent for cameras with the simple purpose of filming.

Even when spherical, coma and astigmatism aberration are compensated, the image is not projected into a straight plane but into a curved one, called Petzval Field curvature. When using a straight camera sensor, this will also cause a difference in focal point across the sensor. This can for example be solved with a curved sensor. Also, like lens imperfections, camera sensors can also have imperfections that cause aberration effects.

The last problem is distortion, that can occur even when all previously mentioned aberration effects are compensated. They cause a difference in focus, while distortion can happen regardless. Lines within a scene are not mapped with a linear radius, where relative distances between lines are distorted in the final image. This can result in either barrel distortion, where the center of the image is magnified more then the edges, or pincushion distortion, where the magnification becomes larger toward the edges of the image [67]. The camera we chose as a reference has barrel distortion.

Light as a Wave

When talking about rays, we try to describe the path of photons. Photons have both characteristics as massless particles carrying energy and propagating electromagnetic waves. Due to this wave nature, specific optical phenomena can be observed in lens systems. The first is chromatic aberration, where the focal point is different for photons with different wavelengths passing through a lens. A common way to reduce chromatic aberration is the usage of multiple lenses with different diffraction indices [68], but this is not observed in images made with our reference camera.

Another potentially significant impact on the image quality, especially on a per pixel scale, caused by the wave nature of light is diffraction. Diffraction is the bending or interference of waves when they encounter corners or pass through a narrow aperture. In the case of cameras, diffraction occurs because light travels through the round aperture within a lens system. Due to the round nature of the aperture, a specific interference pattern arises known as the airy disk (see figure 7). The airy disk is a result of far field diffraction. Given the small distance between the aperture and image sensor, normally near field diffraction would occur. But the lens of the camera will result the airy disk to appear at a finite distance, at the focal length to be precise, from the aperture. As can be seen in figure 7, the intensity of the interference pattern has both minima and maxima, with the minima having an intensity of 0. Once the diameter of the central peak is larger than a pixel on the image sensor, the interference pattern will start to have an impact on the image quality. Once this is the case, diffraction can for example be modeled into our pipeline by a convolution of the output image and a point spread function following the intensity of the airy disk. The intensity is given by equation 2.

$$I(\theta) = I_0 \left[\frac{2J_1(x)}{x} \right]^2 \tag{2}$$

I_0 is the intensity at the center peak, J_1 is a Bessel function of the first kind and x is given by equation 3.

$$x = \frac{2\pi a}{\lambda} \frac{q}{R} \tag{3}$$

a is the aperture radius, λ the wavelength of light, q the radius from the center of the airy disk and R the distance between the aperture center and q. The first minimum of $J_1(x)$ is at $x \approx 3.8317$. From this it can be deduced that q_1, or the radius of the first minimum of the airy disk, is given as:

$$q_1 \approx 1.22R\frac{\lambda}{2a} = 1.22\frac{\lambda}{2A} \tag{4}$$

Where A is the numerical aperture of the system. When the airy disk is projected onto the focal plane at focal distance f of a lens, A is related to the commonly given F-number, often denoted as N, of a lens with:

$$A = \frac{1}{\sqrt{4N^2 + 1}} \tag{5}$$

It has to be noted that q_1 is linearly related to the wavelength of the light passing through λ. This means the diameter of the center peak

is largest for red light, since this has the largest wavelength in the visible light spectrum [69] [70]. In order to quantify the impact of

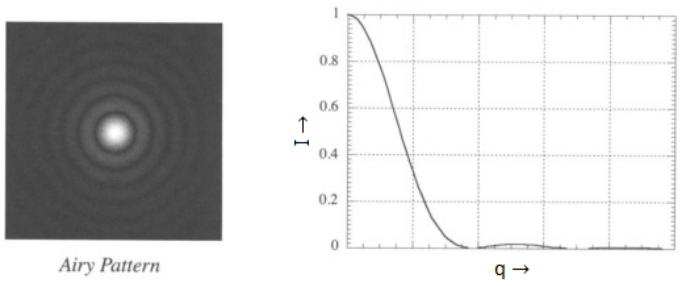


Fig. 7: Airy disk pattern and Intensity over the radius of the airy disk (q)

diffraction on our lens system, we need to find both q_1 and the size of a pixel for our reference camera. The F-number is given in the camera datasheet [66] as 1.4 for both the 3 mm and 10.5 mm lens options. So for red light with a wavelength of 700 nm, this results in a value for q_1 of 1.27 μm. As we recall, q_1 is the radius from the center peak of the airy disk to the first minimum, so the diameter of the center peak is twice q_1, or 2.54 μm.

To calculate the pixel size of our reference camera, we need the sensor size and pixel resolution. The Axis P1435-LE has a pixel resolution of 1920×1080 [66]. The sensor size can be found based on the Angle of View (AOV), interchangeably denoted as Field of View (FOV) in literature, and focal length with the following equation:

$$AOV = 2 \arctan \frac{d}{2f} \tag{6}$$

d is the sensor size in the same direction as the AOV. This equation holds for images with a rectilinear projection, which does not completely hold for our images as they contain barrel distortion. But the error is small and the values we find are realistic for this camera's sensor (see section 2.4.3) [71].

Using equation 6 and the horizontal and vertical FOV given for the 3mm lens, 95° and 51° respectively, we can find the horizontal and vertical sensor size. Dividing these sensor sizes by their corresponding pixel resolution, we find the smallest size of a pixel as being 2.65 μm in the vertical plane. Since this is larger than the airy disk's center peak diameter of 2.54 μm, diffraction does not have a significant effect on our camera. The second peak of the airy disk will however spill over into the neighbouring pixels. But since this peak only has a intensity of roughly 1.75% compared to the center peak (see figure 7), we consider this effect negligible.

2.4.3 Sensor

In real world digital cameras, after the light passes through the lens system it hits the camera sensor. Once a photons hit the sensor surface, they are converted into an electrical signal, often a voltage, by the sensor. The voltage is a measure of light intensity at that location of the sensor and can be used to construct the final image. There are two main types of image sensors that are used in modern cameras; the charge-coupled device (CCD) and the active-pixel sensor (CMOS). Both are based on MOS semi-conductor technology, where the CCD uses MOS capacitors to detect incoming photons, while a CMOS sensor uses a combination of photodetectors, typically photodiodes, and MOSFET amplifiers. CMOS sensors typically have a lower production cost and power consumption, so they are more widely used in consumer goods [72] [73].

The security camera we use as a reference, the AXIS P1435-LE, also uses a CMOS camera sensor. To be specific, it uses a 1/2.8" progressive scan CMOS sensor [66]. The fractional notation in inches is based on the outer diameter of old camera tubes, which were later replaced by digital sensors. The notation stuck, but there is no industry standard that directly translates this notation to a diagonal sensor size in mm. Values slightly differ per company and are more a general grouping of sensor types. The sensor sizes we calculated in 2.4.2 differ slightly from most 1/2.8" sensors we found online, but still lies within a realistic margin to be correct for a 1/2.8" sensor [74] [75] [76]. This revelation does not impact our diffraction calculations, as the aspect ratio is also different for the sensor sizes found online and therefore did not result in a smaller pixel size than we used for our calculations.

Bayer Filtering

A common way to represent colour in digital images is using the additive RGB colour model, where the final colour of a pixel is reproduced from adding the amount of red, green and blue light levels. CCD and CMOS sensors often operate with a variant of the RGB model, where professional cameras use three sensors, one for each colour channel. The cheaper solution, and most likely the solution used in the reference security camera, is using a single sensor in combination with a colour filter array. The most common colour filter array, is the Bayer filter mosaic, as seen in figure 8. This is a pattern of red, green and blue colour filters that lays on top of the square grid of photo sensors, so the light intensity at each photo sensor gives a colour value for the corresponding filter colour. In a Bayer filter pattern, half of the filters are green while the other half is split between blue and red filters. This layout roughly approximates the physiology of the human eye, where the L and M cones have a bias in the green spectrum when combined. In order to create the per pixel colour of the output image, a demosaicing algorithm is used. Simple algorithms use interpolation, like bilinear or bicubic interpolation, to find the pixel colour based on the neighbouring red, green and blue values, but also more advanced algorithms exist [77]. A disadvantage

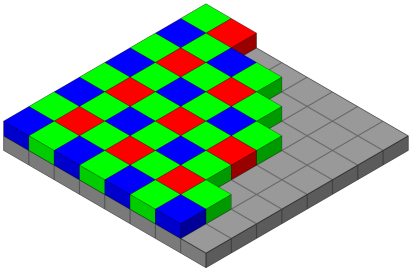


Fig. 8: Bayer filter mosaic

of the Bayer filtering is the small scale artifacts that can occur due to the demosaicing algorithm, like false colour artifacts or zippering artifacts, often along edges in the image. Since we are working on a small image scale of only a few pixels in our pipeline, these artifacts can have a significant impact when introduced.

Compression

In order to reduce the cost of storage and transmission, compression is applied on each image frame of a video instead of saving the raw image data for these frames. The AXIS P1435-LE uses either H.264 or motion JPEG for its video compression [66], which both use the lossy Discrete Cosine Transform (DCT) compression. Lossy means the compression is irreversible and uses partial data discarding and inexact approximations to create the compressed image. As a result of this inaccuracy, compression artifacts, like ringing, banding, aliasing and blockiness artifacts, are introduced [78].

2.4.4 3D Model

Since we are working in a 3D space, we need a 3D model to represent a human face. In brought terms, the 3D model of a human face in computer graphics can be divided into three components:

• *Mesh Structure* - the 3D structure describing the geometry of the face. Also known as a mesh structure, since that is a common way to visualize a 3D model without specific texture information.
• *Albedo Map* - the image texture of the 3D structure without any lighting information
• *Subsurface scattering* - the way a light ray reflects and scatters when it hits the model's surface

All 3 components are described below. But the goal of our pipeline is not simply producing a single model of a face, but for many different identities. These identities have to be generated synthetically, so this process is also described in this section.

Mesh Structure

As established, the goal of the pipeline is to generate a low resolution probe image with an identity that still corresponds to a high resolution gallery image. This means the 3D geometry of the face has to correspond completely with the gallery image, at least when seen from the same pose. This means the extraction, or reconstruction as it is more commonly known, of the 3D model has to be based on the high resolution gallery images. Like super-resolution, 3D face reconstruction can be based on either a single or multiple reference images. Multiple images from different angles will give more information about the original 3D structure of an individuals head. But, like expressed in section 2.3, creating synthetic images from the same individual with different poses is difficult and can lead to relative morphing of facial features. Also creating the 3D model from different images requires stitching and warping these images to fit the model from all angles, introducing extra room for errors in feature placement. This can be made even more difficult when expressions differ between images. Single image reconstruction on the other hand has a much stronger learning bias, as it has to predict much more about the 3D shape. It is therefore common to combine both a 3D morphable model of a face and a deep learning algorithm to restrict the amount of possible solutions [79]. Prediction of shape is however not necessarily a problem, as long as the features align with the reference 2D picture for the same pose. Using single image reconstruction also makes the amount of required synthetic images more straightforward; For each identity only one high resolution image is required for the pipeline to work. Therefore, we chose single image reconstruction for our pipeline.

To choose a single image reconstruction model that fits our application, we used the NoW (Not quite in the wild) benchmark as a starting point [80]. It is a benchmark that was introduced to give a standard evaluation metric to measure the robustness and accuracy of 3D face reconstruction methods. The performance of multiple methods is published on their website, so various well performing ones can be compared to find one fitting for our application. This resulted in a few interesting candidates:

• FOCUS (Face-autoencoder and OCclUsion Segmentation) [81] and 3D Deep Face Reconstruction by Microsoft [82] are two of the best performing methods. They are particularly good at reconstructing faces with occlusion. As a result however, they do not allow for uv mapped textures and use resembling per-vertex textures instead. This means the texture of the face is approximated and not taken from the source 2D image. It is therefore not identity preserving, which is not desired in our pipeline.
• **DECA** [83] and RingNet [80] use the FLAME [84] morphable model as basis for their reconstruction. The FLAME model is trained on 3800 real life 3D head scans to create a 3D shape space and in addition allows for dedicated pose and expression articulation. Both allow for uv mapped textures. DECA is the better performing of the two and is therefore considered for our pipeline.
• 3DDFA v2 [85] and **PRNet** [86] create face meshes unlike the full head 3D meshes of DECA and RingNet. Both are trained on the 300W-LP dataset of landmarked face images with large poses. They also both allow for uv mapped textures. 3DDFA v2 has the better performance of the two, but PRNet has a more straight forward execution of the code and can therefore be easily implemented in our pipeline.
• PIXIE [87] also performs well and even allows for full body 3D reconstruction. However, it uses DECA for face reconstruction without uv mapped textures.
• SADRNet [88] performs well on other benchmarks like REALY [89]. It does however only regress pose and shape and does not reconstruct the facial texture in any way.

So DECA and PRNet appear to be fitting candidates for our pipeline. Both models were therefore implemented and compared. The practical implementation and results of this comparison can be found in the methodology section.

Albedo Map

As mentioned before, an albedo map is the texture of a mesh model without any lighting information. For identity preservation, the albedo maps of our 3D models are extracted from the 2D input face images. The process of uv mapping can be used to map coordinates of a 2D image, refered to as U and V coordinates, to the 3D space with x, y and z coordinates. For a face, facial landmarks in the 2D image are used as reference points to distort the 2D image texture to fit the corresponding 3D landmark coordinates.

Subsurface Scattering

Realistic human skin is difficult to model, not only due to its complex texture both in colour and roughness, but also do to its interaction with light. Human skin does not behave as an opaque surface, but is rather translucent in reality. This can for example be seen when a hand is held in front of a bright light source, where a lot of light travels through the hand in a diffuse manner instead of simply reflecting back to the light source. This phenomenon, of light partly penetrating the surface and scattering through interaction with particles within the medium, before leaving the medium at a different point, is called subsurface scattering. A visual difference between opaque reflections and subsurface scattering can be seen in figure 9. As

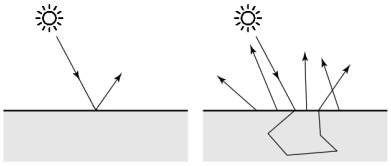


Fig. 9: Opaque reflection with BRDF (left) and subsurface scattering with BSSRDF (right)

mentioned in section 2.4.1, volumetric path tracing can be used to render subsurface scattering. Volumetric scattering of a larger volume like fog or clouds can be modeled with this path tracing model, but human skin too, since it is seen as simply a volume more dense with particles. To achieve this, the bidirectional reflectance distribution function (BRDF) term, $f_\mathrm{r}(\mathbf{x}, \omega_\mathrm{i}, \omega_\mathrm{o})$, in the rendering equation (eq. 1) has to be replaced. This is done with the bidirectional surface scattering reflectance distribution function (BSSRDF), which is a distribution function $S(\mathbf{x}_\mathrm{i}, \omega_\mathrm{i}, \mathbf{x}_\mathrm{o}, \omega_\mathrm{o})$ that describes the ratio of the exiting differential radiance at point $\mathbf{x}_\mathrm{o}$ in direction ω_o to the differential irradiance at point $\mathbf{x}_\mathrm{i}$ from direction ω_i. To generalize the BSSRDF, integration over both the incoming direction and surface area A in needed, changing the full rendering equation into equation 7.

$$L_\mathrm{o}(\mathbf{x}, \omega_\mathrm{o}) = L_\mathrm{e}(\mathbf{x}, \omega_\mathrm{o}) + \int_A \int_\Omega S(\mathbf{x}_\mathrm{i}, \omega_\mathrm{i}, \mathbf{x}_\mathrm{o}, \omega_\mathrm{o}) L_\mathrm{i}(\mathbf{x}, \omega_\mathrm{i})(\omega_\mathrm{i} \cdot \mathbf{n}) \,\mathrm{d}\,\omega_\mathrm{i} \,\mathrm{d}\, A \tag{7}$$

Generally, when the distance between point $\mathbf{x}_\mathrm{i}$ and $\mathbf{x}_\mathrm{o}$ increases, the value of S diminishes. For practical implementation of an subsurface scattering algorithm, this fact can be a substantial starting point [65].

An early approximation for S in the context of human skin is presented in the work of Wann Jensen et al. [90]. They construct the BSSRDF model by taking the sum of a diffusion approximation and single scattering term. This model accounts for light transport between surface locations and it simulates both the directional component and diffuse component. The diffusion approximation is based on the observation that, in a highly scattering medium like human skin, the light distribution tends to become isotropic, or uniform for all orientations. This even holds if the phase function, the function describing the angle between ω_i and ω_o, and initial light source distribution are highly anisotropic, or dependent on the orientation. The light distribution tends toward uniformity for a larger number of scattering events, since each scattering event blurs the light distribution. This leaves an approximation for S that is based on three main parameters: the absorption coefficient (σ_a), the scattering coefficient (σ_s) and the relative index of refraction of the medium with respect to a boundary medium (η). These parameters can be found through experiments, as also given in the work by Wann Jensen et al. σ_a and σ_s are found for red green and blue wavelengths by focusing a tight white beam of light on human skin and observing the radiant exitance over the entire surface [90]. The refractive index of skin varies throughout real skin, but only slightly and can be assumed as a constant for larger areas. The refractive index of skin compared to are is roughly 1.4 [91].

This method relies on a skin model of a single layer, which in not realistic when compared to the anatomical structure of skin. In real humans, the skin is build from multiple complex (cell) layers, like the stratum corneum surface layer, the epidermis and dermis, each reacting differently to light passing through them. More complex BSSRDF models have therefore been developed as well, like a two layer skin model by Donner et al. [91] or even a five layer model by Krishnaswamy et al [92]. Using a complicated multi-layer model can significantly improve the visual appearance skin [93]. However, especially given the small scale of our output images, using a single layer can still obtain a reasonable approximation [90]. Taking the significant increase in image generation time for more complex models into account, the simpler models can still be considered.

Identities

As input for our 3D reconstruction we need to generate synthetic 2D face images. We use StyleGAN2 by Nvidia [5], mentioned in sec 2.3, as a basis. StyleGAN is still considered state-of-the-art for face image generation and still under active development.

As mentioned before, GANs are generative deep learning models that make use of an adversarial process. The generative network is trained by maximizing the error-rate of the discriminative network, therefore passing as much synthesized data as true data. Independent backpropagation is also applied to both networks to improve both networks separately besides the adversarial loss [94]. StyleGAN uses the progressive GAN training method, where the GAN generator is grown from small to large in a pyramid shaped manner. First only a small generator and discriminator play the zero-sum game to generate 4×4 images. The generator and discriminator are then expanded and blended with a new layer to generate 8×8 images, and so on, until the system generates full resolution 1024×1024 images [5]. Like mentioned in the name, StyleGAN uses a style-based generator. Instead of using a traditional input layer to create the latent space, StyleGAN first uses a non-linear mapping network to transform the input latent space to a intermediate latent space creating a learned constant. Vectors from this intermediate latent space are split into styles in the synthesis network to generate the different features of the output image. This learned constant, compared to traditional style transfer networks, makes an input example image superfluous for generation [5].

As an unconditional GAN, StyleGAN's latent space is tangled and creates images with random pose, lighting etc. Untangling can be done with conditioned exploration of this latent space to allow for attribute based editing. Two promising methods of untangling are considered: StyleFlow [60] and the work by Colbois et al. [61].

StyleFlow uses conditional continuous normalizing flows in the GAN latent space to explore it non-linearly, conditioned by attribute features. This allows for both generation based on input attributes and editing a face image along its attributes, like pose and expression, while preserving its identity. The interface of StyleFlow is however not optimized for mass generation of output, and is therefore less suitable for large dataset generation [60].

The method by Colbois et al. projects a dataset with known attribute labels into the intermediate latent space of StyleGAN. Using support vector machines, linear separations are fit in the latent space based on single attributes. New synthetic identities are generated by random sampling the StyleGAN input latent space. It does not allow for real time attribute adjustment. Instead, it generates multiple images, where poses and lighting options are varied at given increments. This does allow for mass identity generation, as this method was specifically developed for large synthetic dataset generation, which is ideal for our pipeline [61]. So we pick StyleGAN2, in combination with the method by Colbois et al. to make StyleGAN conditioned, to generate our synthetic 2D face dataset.

3 Methodology

3.1 3D Pipeline

In this section, the used methods to practically implement the 3D pipeline are presented. This section will follow the same structure as the background counterpart, where the 3D scene is split into four parts (see figure 5). The 3D scene consists of a camera, that can be split up into a lens(system) and a sensor, a 3D model of a face and the renderer that will take all attributes and simulate the behaviour of light in this scene. For each part, the implemented components and software will be discussed.

3.1.1 Renderer

To execute the picked volumetric path tracing algorithm to solve the rendering equation, two software programs were considered; Physically Based Rendering, also known as PBRT [95], and Mitsuba Renderer [96]. Both are so called physically based renderers and allow for the incorporation of all Monte Carlo methods for solving the rendering equation described in section 2.4.1. PBRT is however the more well known one, since the authors received an Academy Award for the technical and scientific impact their work has had on film productions. While Mitsuba is updated more regularly and has slightly more features [97], PBRT has more in depth documentation due to the accompanying book they released [65]. Therefore, we chose to use PBRT as our renderer. We use the volumetric path tracing integrator, since it allows for simple implementation of subsurface scattering in PBRT. We can set a high sampling rate, which would normally cause a high computation time, since we are working on a low resolution scale of only a few pixels. Our sampling rate is 2048 samples per pixel and we use the Halton sampler since it gives a slightly better result with these sampling rates according to the PBRT documentation.

3.1.2 Lens

For our pipeline, we initially tried to approximate a security camera lens system in PBRT. PBRT can render a real lens by loading in a table describing the various lens components [65]. However, we could not find a schematic of a cheap security camera lens system with reasonable parameters to model in PBRT. We therefore tried to simplify the lens system to a single lens, finding the parameters with

the lensmaker equation:

$$\frac{1}{f} = (n-1)\left[\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n-1)d}{nR_1R_2}\right] \tag{8}$$

Here f is the focal length of the lens, n is the refractive index and R_1 and R_2 are the radii of curvature of both sides of the lens. For a convex lens surface, the sign convention gives that $R_1 > 0$ and $R_2 < 0$. For a concave lens surface the opposite is true. The diameter of aperture for the lens can be found with the F-number, as it is the ratio of the focal length to the aperture diameter. Creating such a lens, however, was found to produce unrealistic amounts of aberration, as can be seen in figure 10. Even though the face in this example is completely in focus, the face appears very fuzzy due to mainly spherical aberration. The spherical aberration can be particularly observed as to the circular glow around the face. It was also found that the aperture of our reference lens was only 2.14 mm. This means the head model, projected at a relatively large distance from the camera, appeared underexposed since most rays reflected from the head would not reach the sensor. Increasing the ISO, a measure of light sensitivity for the camera sensor, or exposure time solved this issue. Due to the excessive aberration, we finally opted to use the pinhole camera model. The thin lens model, a theoretical lens with zero thickness, did not produce a significant difference in quality when in focus compared to the pinhole camera. The rendered output of PBRT for both the pinhole and thin lens system, can also be observed in figure 10. Even when the real lens would have been implemented,

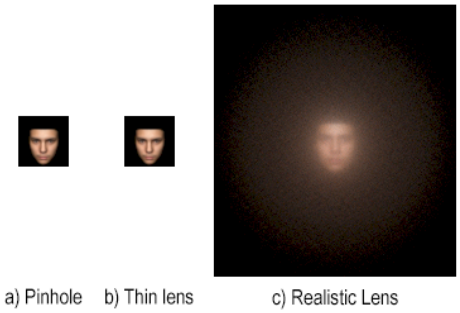


Fig. 10: Lens options PBRT

chromatic aberration can not be modeled with PBRT, since the path tracing algorithm of PBRT does only model rays as having the particle behaviour of photons. Spectral path tracing implementations do exist to introduce wave behaviour [98] [99] [100], but are not explored in this research.

3.1.3 Sensor

Sensor artifacts created by a Bayer filter could be reproduced by applying a Bayer filter pattern over the raw image output of PBRT, after which a demosaicing algorithm is applied to construct the final image. However, due to time constraints, the effects of the Bayer filter are not introduced in our actual pipeline. Instead, we use the method available within PBRT for image construction. Here, for each pixel of the image film, all the radiance values, represented as RGB spectrum values, of the sampled rays that hit this pixel are combined. A Gaussian filter is used over the pixel to combine the radiance values into a final pixel colour [65].

In order to replicate compression artifacts in our pipeline, we take the uncompressed image generated by PBRT and write the image to a compressed file using a Python script and the JPEG compression algorithm within the OpenCV Library [101].

3.1.4 3D model

DECA [83] and PRNet [86] were both found to be fitting single image 3D reconstruction models for our pipeline. To execute a proper comparison, both models were combined with PBRT to generate images. Going forward, the shortcomings and advantages of DECA and PRNet within our pipeline are compared, based on the generated mesh structure and albedo map. Example outputs for PRNet and DECA can be found in figure 11, both without realistic lighting conditions and after being rendered with PBRT.

PRNet versus DECA

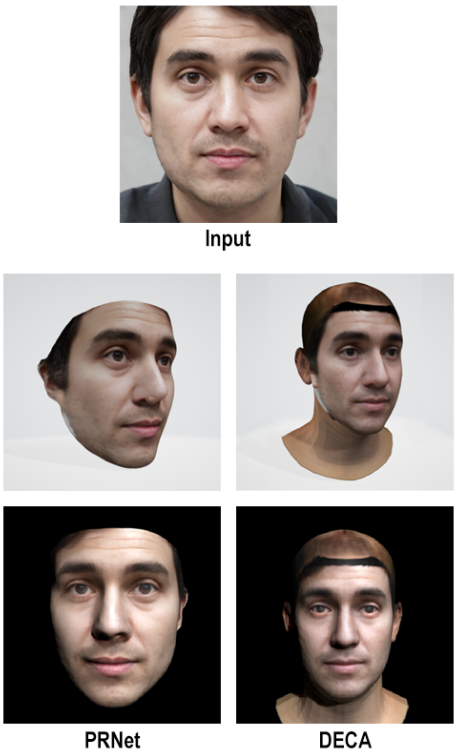


Fig. 11: 3D reconstruction of a synthetic face with PRNet (left) and DECA (right). The bottom two images are rendered using PBRT

PRNet generates a mesh of the face, excluding the hair and neck of an individual. When the structure is first generated, the mesh has a ribbed structure, which is unfavourable due to unexpected reflection patterns on the skin compared to a real face. To avoid this problem, a smoothing script is applied using the open3d library and Python.

DECA can generate both high and low poly-count meshes of a full head. The high poly-count mesh does allow expression modification, but not uv mapping. Since uv mapping is required for a realistic skin texture, the low poly-count output is compared to PRNet. In the mesh, the eyes are generated as separate objects from the rest of the head mesh. This results in unexpected reflection behaviour in PBRT between the eyelids and eyeball objects, even noticeable in low resolution outputs. In addition, DECA uses a 3D reference head space as its basis, as incorporated within the FLAME model [84]. This 3D head is then morphed to fit the identity of the input 2D reference image. It was observed that this morphing has some limitations, as a gap/ridge, on top of the output head mesh was often observed where the mesh was not properly connected. This too can result in undesired reflection effects in the rendering process.

PRNet and DECA both use uv mapping to create an albedo map. Both however have observable mapping inaccuracies. PRNet outputs have an alignment issue at the nose, where the texture of the nostrils from the 2D image is often aligned below the nostrils of the 3D mesh. DECA on the other hand has an alignment problem at the eyes. It uses the FAN [102] model for eye landmark predictions,

which are used for the mapping. FAN shows inaccuracy when detecting narrow eyes, so predicted eye landmarks are often off center, which in turn results in misaligned eye textures in the 3D model. We found that the dlib library's [103] landmark detector performs better, so we implemented a code change to overrule the FAN predictions, resulting in more accurate eye texture placement. DECA also crops the input 2D images strongly around the face in its pre-processing steps, causing an observable black bar at the forehead as a result of filler black pixels due to excluded texture information. Furthermore, DECA allows for texture reconstruction to create a LR estimation of the missing head textures, such as the hair and neck [84][104]. They are however not particularly accurate, likely due to the black bar at the forehead, which confuses the prediction algorithm. Finally, both PRNet and DECA have problems with interpreting textures of non-visible face parts in the input 2D images. This results in parts of the background from the input images appearing as face textures.

Altough DECA performs better in benchmark tests, the above listed limitations found after implementation of the reconstruction method were perceived as more impactful than the limitations of PRNet. PRNet generated less albedo map artifacts and the relative simplicity of the generated mesh allowed for less observable errors after rendering. We therefore choose PRNet as our preferred 3D face reconstruction method.

Subsurface Scattering

As discussed before, we use PBRT to render lighting conditions. The build-in volumetric path tracing algorithm of PBRT is used, which can be utilized to model subsurface scattering for the skin texture. PBRT uses a single layer model, similar to the method discussed in section 2.4.4. PBRT likewise takes σ_a, σ_s, both as RGB values, and η as input parameters. σ_a and σ_s are taken from the real-world experimental values found by Wann Jensen et al. [90], while η is approximated as the constant value 1.4 [90]. As seen in the bottom images in figure 11, this gives reasonable results for our application.

Identities

StyleGAN2 [5], in combination with the method by Colbois et al. [61] to untangle its latent space, is used to generate the synthetic identities. With this method, each identity can be generated with different poses in given increments. An example output for a single identity is given in figure 12. We use frontal images as input for our 3D reconstruction method, since this results in the least missing texture artifacts, unlike only one side of the face being visible. Therefore, we filter on frontal face images using the pre-trained pose estimation model HopeNet [105]. Since most real face datasets do not contain children due to complex privacy laws, we also filter out synthetically generated children from our dataset. Here a pre-trained model is used, an age estimation model trained by the Swiss Federal Institute of Technology [106]. Finally, our 3D reconstruction method does not anticipate glasses, which will result in the glasses being projected on the albedo map without having a 3D mesh. We use canny edge detection between the eyes, to detect a potential glasses nose bridge, as filtering method.

Fig. 12: Different poses for a single synthetic identity, as generated by the controlled StyleGAN2

The paper by Colbois et al. partly focuses on privacy concerns regarding face datasets. Therefore experiments are described to verify that StyleGAN does not simply reproduce identities from its training dataset (FFHQ), which would make the output not fully synthetic. This is confirmed to be the case, which is in line with our own requirements. It was however observed that the variability of the synthetic dataset is less than the real-world dataset. We therefore perform an extra check to verify the generated identities are unique from each other. The synthetic images are encoded into 128 element vectors with the face-recognition python library [107], which is based on a pre-trained model by Openface [108] and dlib [103]. The vectors are normalized and the angles between 1500 identities are plotted in fig13, where the smaller angles correspond to face pairs with similar attributes. No vectors were fully aligned, as the smallest angle found in this dataset was 11.2°. The pairs of images with the smallest angle between them where then manually inspected to check for similar identities. Some pairs had genuine attribute resemblance, but the overall identity was still confirmed to be different, as can be seen in figure 13.

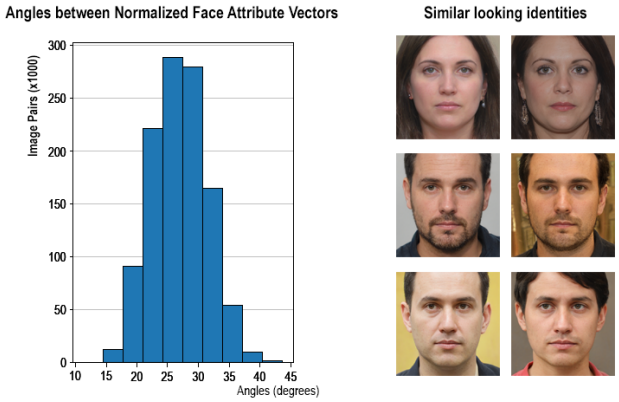


Fig. 13: Angles between encoded face images (in degrees) to verify uniqueness

3.2 Overview Pipeline

Now that all implemented components of our pipeline are discussed, a full pipeline overview can be presented. Figure 14 shows the flow diagram for full synthetic VLR image generation. The left side presents the generation of synthetic HR face images using StyleGAN2 [5] in combination with the latent space untangeling method by Colbois et al [61]. We filter the direct output of this combined model and only let frontally posed, adult and non-glasses images through. This creates a HR synthetic face dataset, that can also be used as gallery dataset in the context of facial recognition. The right side of the diagram presents the LR pipeline. We first use PRNet [86] to perform single image 3D reconstruction. The output of PRNet contains undesired ridges, which are smoothed out with a simple Python script. Now we have a 3D face mesh and a albedo map constructed from the HR input image. Next, they are placed in a 3D scene to be rendered by PBRT [65]. A 3D scene is described by a text file, describing all present 3D models and textures, as well as the used camera, camera position, light sources, material properties like scattering parameters and rendering algorithm. After running PBRT to render the described scene, a raw image file is generated as output. This file can either be converted to a .png output image, or JPEG compression can be applied using OpenCV [101] and Python. Both options result in a LR output image with the same identity as the HR input, either with our without compression. The LR pipeline is completely one-to-one, meaning one HR input images results in one LR output.

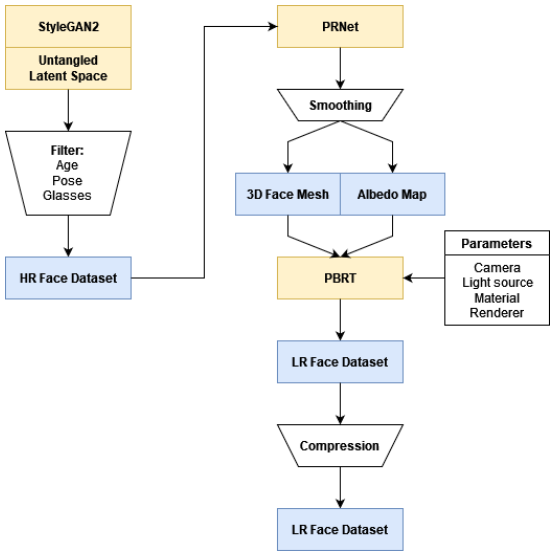


Fig. 14: Complete overview of our synthetic image generation pipeline

3.3 Facial Recognition Software and ROC

To answer the second part of our research questions, we need facial recognition software to compare the different output variations of our pipeline against each other and a real-world low resolution dataset. The facial recognition software of choice is FaceVACS [109] and Deepface [110]. FaceVACS is a closed-source facial recognition engine that presents itself as a state-of-the-art commercial competitor. Since FaceVACS is closed source, the inner workings are considered a black box. Batch matching of images is possible, with similarity scores being generated as outputs. Deepface on the other hand is an open-source library, that incorporates multiple commonly used and well performing open-source face detection models, facial recognition models and similarity metrics. From the available models, we choose RetinaFace [111] for detection of landmarks and cropping, since it also performances particularly well on low resolution faces. The ArcFace facial recognition model [112] is used to encode the face images as vectors. The euclidean distance is found between the L2-norms of a vector/face pair to find similarity scores. Euclidean L2 form seems to be the most stable similarity score metric according to experiments by the library authors [113].

The similarity scores generated by Deepface and FaceVACS are used to plot Receiver Operating Characteristic (ROC) curves. We generate probe and gallery images for 500 synthetic identities, which we compare against each other to generate similarity scores for 250.000 image pairs. The ROC curve plots the True Positive Rate (TPR), or probability of successful matching, against the False Positive Rate (FPR), or probability of false successful matching, for different thresholds. The similarity scores from Deepface and FaceVACS are used as probability scores with truth labels attached. Since the ROC plots the TPR against the FPR, the positive diagonal represents the situation where the probability of an image pair being flagged as a true match, is equal for both a real or false identity pair. With other words, this line represents a random classifier. The Area Under the Curve (AUC) can therefore be used as a measure of similarity score accuracy. A random classifier has a AUC of 0.5. An AUC higher than 0.5 means a classifier better than random and consequently an AUC less than 0.5 means a classifier worse than random. Yet, it has to be noted that the AUC has a probability distribution on its own when a finite, and in particularly limited, set of genuine and imposter scores is used to plot the ROC curve. This means the probability of a random system having an AUC significantly different from 0.50 is non trivial for a limited number of genuine and imposter scores [114]. In other words, AUC scores larger than, but close to 0.5 might still indicate a random classifier for these cases. In addition, an AUC score of say 0.7 might indicate a classifier better than random, but is still considered a poor performance for a facial recognition model. Especially compared to state-of-the-art accuracy scores for high resolution facial recognition, which are well above 0.9 [115] [113].

To properly compare results, we have to take the probability distribution of the AUC into account in case the AUC values are close. Based on the findings by Bamber, the AUC is closely related to the Mann-Whitney U statistic [116]. Based on the properties of the Mann-Whitney U statistic, DeLong et al. constructed a method to find a non-parametric approach of comparing AUC scores of two correlated ROC curves by generating an approximated covariance matrix [117]. Hanley et al. show how the DeLong method can also be used to find an estimate of the variance of the mean AUC score [118]. We will be implementing the method proposed by Sun et al. to find this variance of the mean AUC score, who have reduced the time complexity of DeLong's algorithm from quadratic down to linearithmic order [119]. From the variance of the mean we can also find the standard deviation of the mean by simply taking the square root. To compare two AUC scores, we would like to find a 95% Confidence Interval (CI). This estimated confidence interval will give the interval in which 95% of the AUC score means will lie. To achieve this, the distribution of the AUC score has to be known. It is shown that for large values of genuine and imposter scores, the estimated AUC is approximately normally distributed [120] [116] [114]. Given our dataset size (500 identities, 250.000 image pairs), enough scores are present to assume this normal distribution.

The ROC curves or AUC scores will not be used to compare between the two facial recognition models used. So the difference between compared ROC curves can be attributed to differences in the presented dataset, since the model is a constant. As a result, when two ROC curves or AUC scores are equal, the variables within the two corresponding datasets, like amount of noise or difference in colour, are perceived as equally challenging during the facial recognition process.

3.4 Super-resolution Software

In order to test the influence of applying super-resolution on probe images before facial recognition, we need to pick a super-resolution algorithm. Our choice fell on a relatively simple CNN, the SRCNN model as presented in section 2.2.1. As a benchmark for its influence, bicubic interpolation is also used to create SR images. SRCNN is not state-of-the-art, but should yield a significantly better SR performance compared to bicubic interpolation upscaling [34] [29]. SRCNN can also easily be combined with existing facial recognition software like FaceVACS or Deepface, as it can be introduced as a pre-processing step without a need to modify the facial recognition model. In addition, it requires relatively short training times. The model has to be trained just to find the filters instead of accuracy and the model is very small and compact.

Table 1 SRCNN Parameters

Filter sizes for each layer:	
Patch Extraction	64 x 1 x 9 x 9
Non-linear Mapping	32 x 64 x 1 x 1
Reconstruction	1 x 32 x 5 x 5
Optimizer:	Adam
Loss Function:	MSE
Input Patch Size:	33 pixels
Output Patch Size:	21 pixels
Stride:	14 pixels
Batch Size:	128

The used parameters during training are presented in table 1. We train for both $2\times$ and $4\times$ upscaling to compare between the two. Most parameters are based on the basic network settings as presented in the paper, like the filter sizes and the Mean Squared Error

(MSE) as the loss function. One place we deviate from the paper is the choice of optimizer. Using the Adam optimizer obtained better results with less hyperparameter tuning compared to the RMSprop optimizer used by Dong et al. For our training dataset, we use 1000 synthetic faces, each from a different identity, generated by our pipeline. Their resolution is either 2× or 4× larger compared to the expected VLR input, respective to the target upscaling. The corresponding VLR input images in our training data are also generated with our pipeline. Like explained in section 2.2.1, the input is first up-scaled with bicubic interpolation in the pre-processing. This is also done for our training data. The patch sizes for the first and last layer inputs are also shown in table 1. The size of the input layer patches are larger than the output because zero padding is avoided in the network, since it introduces border artifacts. As a result, the spatial dimensions reduce each convolutional layer, so the input patch has to be bigger. The stride parameter is the step size of our sliding window while creating patches of our full resolution input. It is smaller than the patch size to create overlapping, which improves the reconstruction step. We train for 160 epochs for 2× upscaling and 40 epoch for 4× upscaling. The best training parameters for number of epochs and dataset size were checked with the Peak Signal-to-Noise Ratio (PSNR) between HR references generated by our pipeline and the SR outputs. PSNR is used as comparison metric, since it was also used in the original paper [34]. It was found that an output peak exists, where more training data or epochs will reduce the quality of the SR output after a certain point. This is most likely due to the over-prediction of image information that is not present in the reference HR images.

Example outputs for are SRCNN model are presented in figure 15. Their PSNR relative to the HR reference images are also given, where a higher PSNR value means a more faithful reconstruction. For comparison, the outputs when bicubic interpolation is used for the same upscaling are also included. It can be seen that SRCNN has a perceptible increase in reconstruction abilities over bicubic interpolation, as expected.

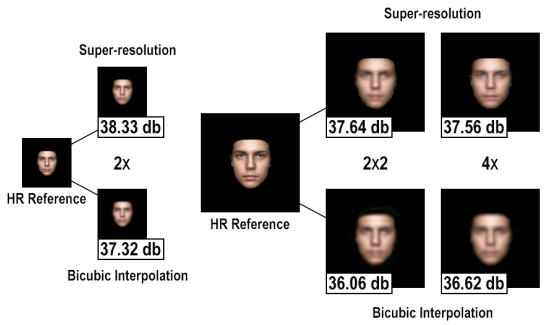


Fig. 15: Example SRCNN and bicubic interpolation outputs for 2×, 2×2 and 4× upscaling. Their PSNR values relative to their HR reference counterparts are also given

4 Experimental Setup

After thoroughly describing the different components of our pipeline, facial recognition models used, classifier comparison method and super-resolution model, we use them to find answers for our second set of research questions.

As mentioned in section 3.3, we use probe and gallery images from 500 synthetic identities for our experiments, unless indicated otherwise.

4.1 Influence of IPD

To test the influence of face images with a interpupillary distance of a few pixels on facial recognition, we generate images for three different IPD values that are frontally lit and posed. No compression is applied and the images are rendered with a pinhole camera, so these images are ideal, as in without distortion or artifacts. This way, the only variable is resolution or IPD. We chose an IPD of 8, 11 and 14 pixels for comparison. According to our definitions set in section 2.1, these face images are considered very low resolution or slightly moderately low resolution.

4.2 Variation in Pose, Lighting and Compression

To include the variation in pose, lighting and compression for our SR experiment, a lot of different options for each parameter can be generated. However, this gives an abundance of results. Therefore we try to narrow these options down by picking six parameter options that give a good representation of our pipeline output. We conducted a pre-experiment, with 10 identities, where we generated 48 options for each identity by combining the following parameters:

- 4 poses (frontal, looking up, looking down, looking sideways)
- 4 lighting positions (frontal, from above, from bottom, from side)
- 3 levels of compression (none, small, heavy)

From this we found that the six options in figure 16 give a good visual representation of the capabilities of our pipeline. We use an IPD of 8 pixels for these options to truly challenge our facial recognition models in the very low resolution domain. The cases where JPEG compression is applied, heavy compression is applied since this seemed to coincide with the observed compression in the real world dataset we will be comparing our pipeline output to. The examples of the six options shown in figure 16, are for a single identity. The synthetic gallery image is also shown as a reference.

Fig. 16: Six pipeline output combinations used for further experiments

4.2.1 Influence of Super-resolution

To quantify the influence of applying SISR to the low resolution probe images before performing facial recognition, we use the super-resolution algorithm as described in section 3.4. The pipeline output options, as described in figure 16, are used to find our reference or base AUC scores. Then, 2×, 2×2 and 4× upscaling is performed on these datasets using SRCNN. The bicubic interpolation algorithm, as described in section 2.2.1, is also used for 2× and 4× upscaling. This way, a distinction between super-resolution algorithm complexities can be made.

4.2.2 Pipeline vs SCFace

Finally, we will compare our pipeline output to a real-world low resolution dataset. The SCFace dataset [1] will be used as the real-world dataset. The SCFace dataset contains high resolution gallery images for 130 different identities. These 130 people are photographed with 5 different surveillance cameras at 3 different distances from the

camera. The cameras are placed at a height of 2.25 meters and the people stand at a distance of either 1.0, 2.6 or 4.2 meters from the camera. For our comparison, the images at a distance of 4.2 meters are of interest to us, since the closer images are no longer considered VLR or even MLR according to our definition set in section 2.1. These are the images in the dataset labeled with "distance 1".

In order to compare our pipeline output to SCFace, we first take the high resolution gallery images in the SCFace dataset and put them into our pipeline where normally the synthetic gallery images are used as input. We set our pipeline parameters, such as pose, lighting direction, illumination and compression amount to visually copy the real probe image datasets of SCFace. Two cameras, surveillance camera 1 (Bosch LTC0495/51) and 3 (J&S JCC-915D), are selected to be replicated. Camera 1 and 5 produce images with a similar yellow hue. Camera 5 has a higher contrast, while camera 1 produces images with a slightly higher IPD compared to all other cameras, probably due to a smaller field of view. The slightly higher IPD was regarded more interesting for a comparison. Images from camera 2, 3, and 4 all contain a similar blue hue and IPD, so the replication challenge would be similar for all three [1].

Objectively comparing the output of our pipeline and real cameras was considered, but found to be difficult. SCFace presents itself as an uncontrolled indoor dataset. In particular, the illumination conditions are uncontrolled and the participants were not asked to look at a fixed point. Therefore the illumination, pose and even IPD as a result of uncontrolled poses are slightly different between the images. Recreating every individual image from SCFace with our pipeline is possible, but is tedious work for 130 seperate images and defeats the point of generating a dataset. As a result, the slightly more artistic approach of replication was taken to tailor an output that visually resembles the camera dataset as a whole.

Figure 17 shows SCFace's camera 1 for four identities, the corresponding replication of the camera's specifications with our pipeline and the corresponding high resolution gallery images. Figure 18 shows the same for SCFace's camera 3.

Fig. 17: SCFace camera 1, our pipeline and high resolution references for four identities

Camera 1 was observed to create images with an average IPD of 13 pixels, while camera 3 puts out an average IPD of 9 pixels. The illumination was matched for both cameras by tweaking the RGB spectral distribution of the radiance emitted by the light source in PBRT. The SCFace paper [1] describes the only light source as being a window on one side of the room. In the images, the light source

Fig. 18: SCFace camera 3, our pipeline and high resolution references for four identities

appears to light the individuals quite uniformly, coming from above to create the shadow beneath the nose and slightly right since the left cheek appears to have a more shadow. So, the light source in our 3D scene was placed accordingly. The camera position, and therefore pose, can be modeled after the described position in the SCFace paper (2.25m up, 4.2m away). The images appear to have significant JPEG compression, since typical JPEG artifacts created by lossy DCT compression can be observed such as blockiness and ringing artifacts between the head and background. Therefore, heavy JPEG compression is also applied to our raw pipeline output. In particular camera 1 also appears to introduce colour/wavelength dependant artifacts, which could have been introduced by a Bayer filter or chromatic aberration, or the combination of the two. As discussed before, these forms of degradation can currently not be replicated with our pipeline.

To compare the effect of synthetic identity generation to the real world SCFace dataset, the exact same pipeline parameters found to recreate camera 1 and 3 are applied to our 500 synthetic identities. In addition, their is a significant difference between the background of the SCFace images and the background of our pipeline generated images. As can be seen in both figure 17 and 18, the SCFace images are taken in front of a white wall and the images do contain the full head, including hair and the neck, and upper body of every individual. Our pipeline, as described before, only depicts the face in a void, as generated by PRNet. While the pipeline images do contain the most common attributes used to find landmarks in the facial recognition process, like nose, eyes, mouth and jawline, their might be a difference in recognition performance when more of the body is visual. Therefore, the facial recognition software is also run a second time on the real SCFace datasets, but this time with the faces cropped manually to match our pipeline output. Examples of this cropping are shown in figure 19.

Fig. 19: Manual cropping of the SCFace dataset

5 Results and Discussion

In this section the results of the experiments from the previous section are presented and discussed.

5.1 Interpupillary Distance

Figure 20 shows the ROC curves and AUC scores for face images with an IPD of 8, 11 (VLR) and 14 (MLR), for both FaceVACS and Deepface. As described in section 4.1, these images are ideal. These AUC scores indicate that interpupillary distance has negative impact on the facial recognition software's ability to correctly verify identities at an IPD of 8 pixels. This appears to be especially the case for Deepface. However, once the IPD approaches the MLR range this impact quickly disappears as FaceVACS and Deepface perform as almost perfect classifiers for an IPD of 11 and certainly 14 pixels.

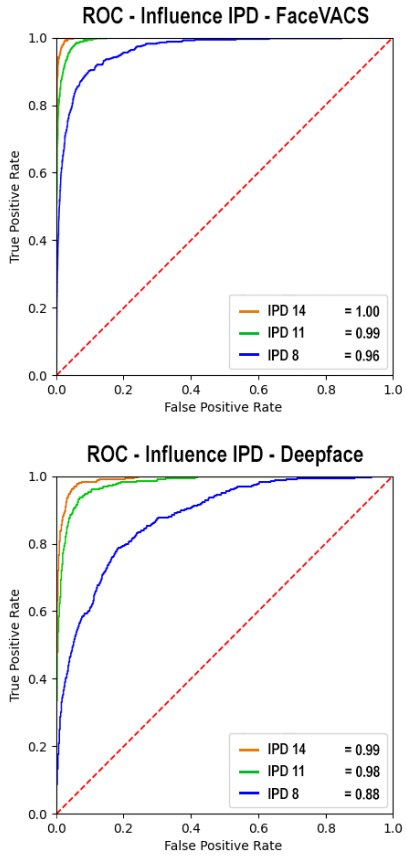


Fig. 20: ROC curves for face images with an IPD of 8, 11 and 14 pixels for both FaceVACS and Deepface

5.2 Super-resolution

Table 3 presents the AUC scores for the six pipeline parameter combinations described in table 2 and section 4.2 while using FaceVACS as facial recognition software. Table 4 presents the same results while using Deepface. Base represent the scores for the unaltered LR images. The results of our super-resolution experiment, as described in section 4.2.1, are also presented in these tables. For all AUC scores, their 95% confidence bounds, as substantiated in section 3.3, are also included.

The first thing that becomes apparent from our results, is the significant impact of compression on the verification capabilities of both FaceVACS and Deepface. For our base pipeline outputs with heavy

Table 2 Pipeline output options to test the influence of pose, lighting and compression

Option	1	2	3	4	5	6
Compression	No	No	No	Yes	Yes	Yes
Lighting	Frontal	From Above	From Above	Frontal	From Below	From Below
Pose	Frontal	Frontal	Looking Down	Looking Down	Looking Up	Looking Left

compression, Deepface basically perform as a random classifier with AUC scores slightly above 0.5. Although FaceVACS handles compression better, the performance loss is still significant compared to the non-compression cases. Applying either bicubic upscaling or super-resolution resulted in no meaningful improvement in classification capabilities for these compression cases. The opposite is even true for implementing SR before FaceVACS. The only significant, while taking the confidence bounds into consideration, change in AUC scores is negative compared to the base cases. A potential explanation is that FaceVACS uses a method to compare LR and HR images that is better resistant to compression than SR. A by-product of SRCNN upscaling is a sharpening process, which in cases with a lot of compression also means sharpening compression noise, as a result hurting verification. This would also explain why bicubic interpolation results in a slightly less prominent performance loss, as its algorithm results in less apparent sharpening compared to SR. Since FaceVACS is closed source, this phenomenon is however non-verifiable.

When the input images for FaceVACS do not contain heavy compression, their is a modest, but statistically significant improvement when $2\times$ upscaling is performed with SR. However, further upscaling does not result in further improvement, but rather loss in verification performance. Especially for $4\times$ SR upscaling. The initial improvement could be explained by the increasing of the IPD, which improves FaceVACS' performance as shown in our previous experiment. The subsequential performance loss for $4\times$ upscaling could be explained in a similar vain as for compression cases. But instead of sharpening compression noise, the sharpening introduces distinct facial feature details that are not present in the gallery HR images, making varification more difficult. Or at least compared to the method the closed source FaceVACS uses to compare LR and HR images. This could be substantiated by the fact that bicubic interpolation, which does not sharpen the up-scaled images, does not show this apparent drop in AUC scores for $4\times$ upscaling. This would be in line with research by Li et al. [22] as mentioned in section 2.2.1.

Deepface experiences positive improvement from upscaling when the input images contain no compression, where SR has a significant edge over bicubic interpolation. The overall improvement in facial recognition can likely be attributed to the increase in IPD. As seen in our previous experiment, this significantly improves Deepfake's ability to correctly match identities. For the base cases, it was observed that RetinaFace had sometimes difficulty with correctly placing all facial landmarks, even though this model performs particularly well with LR faces compared to the other available detection models in Deepface. Increasing facial details while upscaling helps to place landmarks on more realistic locations, which would explain why SR performed significantly better than bicubic interpolation. We find this likely, since most increases in AUC score correspond with the increase in PSNR value as presented with our example in figure 15. Looking at the SR cases, they contain the higher PSNR values, which relates to a better reconstruction of image details.

Changing the pose and lighting directions to something different than frontal also has an impact, although much less intense than the impact of compression. The change in pose and lighting direction, in particular the combination of the two, lowers the AUC score by roughly 0.09 for both FaceVACS and Deepface. Interestingly, super-resolution, whether being bicubic interpolation or SRCNN, does not close the gap on this loss of performance. The loss of performance can likely be attributed to making detection more difficult. Change of pose introduces and obscures facial features compared to frontal images, while change of lighting direction introduces hard shadows, mostly around important landmark locations such as the eyes and

Table 3 AUC with 95% confidence interval bounds for six pipeline parameter combinations using FaceVACS

	FaceVACS					
	1	2	3	4	5	6
Base	0.953 ± 0.008	0.870 ± 0.016	0.869 ± 0.017	0.786 ± 0.020	0.667 ± 0.026	0.653 ± 0.024
Bicubic 2	0.960 ± 0.007	0.884 ± 0.015	0.887 ± 0.015	0.781 ± 0.020	0.659 ± 0.026	0.648 ± 0.025
Bicubic 4	0.961 ± 0.007	0.885 ± 0.014	0.891 ± 0.014	0.763 ± 0.021	0.658 ± 0.025	0.644 ± 0.024
SR 2	0.969 ± 0.005	0.897 ± 0.013	0.902 ± 0.012	0.732 ± 0.022	0.651 ± 0.025	0.643 ± 0.024
SR 2x2	0.970 ± 0.005	0.882 ± 0.013	0.878 ± 0.014	0.701 ± 0.023	0.649 ± 0.025	0.630 ± 0.024
SR 4	0.945 ± 0.008	0.845 ± 0.017	0.852 ± 0.016	0.721 ± 0.023	0.648 ± 0.025	0.634 ± 0.025

Table 4 AUC with 95% confidence interval bounds for six pipeline parameter combinations using Deepface

	Deepface					
	1	2	3	4	5	6
Base	0.786 ± 0.019	0.698 ± 0.023	0.669 ± 0.025	0.579 ± 0.026	0.510 ± 0.025	0.525 ± 0.025
Bicubic 2	0.810 ± 0.019	0.711 ± 0.023	0.683 ± 0.025	0.579 ± 0.026	0.513 ± 0.025	0.523 ± 0.025
Bicubic 4	0.831 ± 0.017	0.734 ± 0.022	0.706 ± 0.024	0.591 ± 0.026	0.511 ± 0.025	0.523 ± 0.025
SR 2	0.907 ± 0.013	0.812 ± 0.019	0.751 ± 0.023	0.590 ± 0.027	0.511 ± 0.025	0.533 ± 0.025
SR 2x2	0.925 ± 0.011	0.820 ± 0.018	0.774 ± 0.022	0.592 ± 0.026	0.511 ± 0.026	0.534 ± 0.026
SR 4	0.917 ± 0.011	0.806 ± 0.019	0.793 ± 0.020	0.625 ± 0.025	0.517 ± 0.025	0.549 ± 0.025

nose. Both make the detection of landmarks more difficult, effects that stay relevant even after compression.

When looking at the main context of this research, real world VLR face images with realistic image degradation and thus heavy compression and pose/lighting variation, the application of SR does not appear to help, but rather hurt facial recognition performance. Facial recognition appears to perform poorly in general for these cases, with Deepface performing non better than a random classifier and FaceVACS showing performance significantly sub-par when compression is involved. Only for ideal cases with low amounts of noise, shadows or pose variation do FaceVACS and Deepface show good AUC scores. In these cases, SR upscaling might be meaningful, but the exact application really depends on the facial recognition software used. And since these cases are not really realistic in real-world settings, the significance can be questioned.

5.3 Comparison to SCFace

We also tried to compare the facial recognition performance on our pipeline output to the real low resolution dataset SCFace. The composition of the datasets are described in section 4.2.2. The AUC scores for SCFace's camera 1 and 3 (SCFace), the cropped SCFace images (SCFace Crop), their pipeline recreations (SCFace Syn) and the pipeline output using the same parameters but with the fully synthetic data (Pipeline) can be found in tables 5 and 6. Figure 21 shows the corresponding ROC curves for FaceVACS, while figure 22 shows the ROC curves for Deepface.

When looking at the ROC curves, two things immediately become clear. Firstly, FaceVACS does a much better facial recognition job than Deepface for the various datasets. FaceVACS even performs as a perfect classifier on the SCFace camera 1 dataset with an AUC of 1.000 and a very small confidence interval of ±0.0002. Deepface has much more difficulty, particularly with the real world data. Here, even for the camera 1 dataset with the higher IPD, the AUC score lies around 0.74. As mentioned in section 2.2, this is considered weak performance for a facial recognition model. This difference in performance between the models is not surprising, given the general better performance of FaceVACS in both previous experiments. Secondly, it is significantly easier to correctly match identities for camera 1 than for camera 3. The camera 1 datasets have a higher IPD, as discussed in section 4.2.2, than camera 3 datasets. And as shown with our first experiment, a higher IPD will result in better facial recognition performance for both Deepface and FaceVACS.

Another thing we notice, is that the synthetic data is matched better with Deepface than the real-world SCFace data, exactly the opposite of FaceVACS. This might be explained by the fact that FaceVACS is differently optimized for specific image artifacts and degradation,

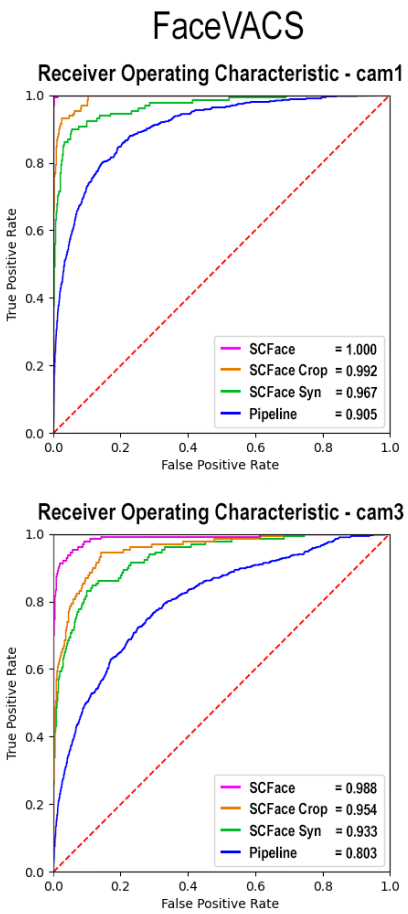


Fig. 21: FaceVACS ROC curves for SCFace camera 1 and 3

but this is hard to check because of the closed source nature of FaceVACS.

In general, taking our main goal into consideration of creating a VLR synthetic pipeline that generates face images comparable with a real-world dataset, we can see that we were not successful. Would we have been successful in recreating the image degradation of real world images, than in particular the SCFace cropped and SCFace synthetic dataset should have a equal ROC curves and AUC scores. As explained in section 2.2, since the facial recognition models are a constant, the difference in image information parameters will determine the difference in these scores. And the parameters of these two

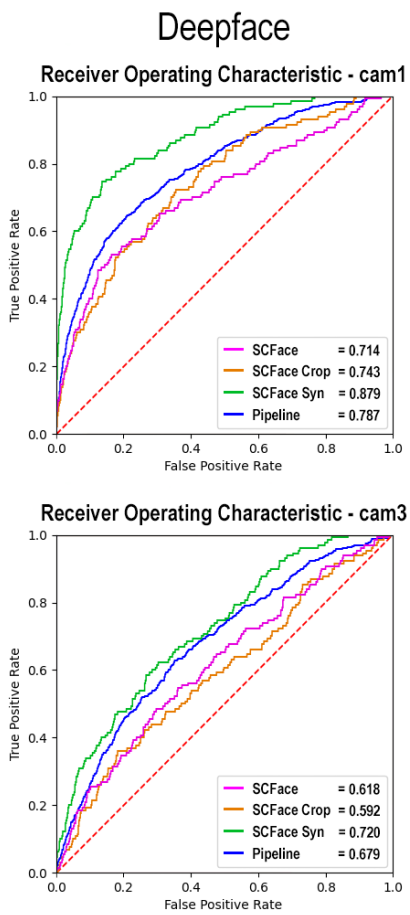


Fig. 22: Deepface ROC curves for SCFace camera 1 and 3

Table 5 AUC, standard deviation and 95% confidence interval for SCFace and pipeline comparison using FaceVACS

FaceVACS	cam1			cam3		
	AUC	σ	$\pm$CI 95%	AUC	σ	$\pm$CI 95%
SCFace	1.000	0.0001	0.0002	0.988	0.005	0.010
SCFace Crop	0.992	0.002	0.004	0.954	0.009	0.018
SCFace Syn	0.967	0.008	0.016	0.933	0.011	0.022
Pipeline	0.905	0.007	0.013	0.803	0.010	0.020

Table 6 AUC, standard deviation and 95% confidence interval for SCFace and pipeline comparison using Deepface

Deepface	cam1			cam3		
	AUC	σ	$\pm$CI 95%	AUC	σ	$\pm$CI 95%
SCFace	0.714	0.026	0.051	0.618	0.025	0.050
SCFace Crop	0.743	0.022	0.043	0.592	0.026	0.051
SCFace Syn	0.879	0.016	0.031	0.720	0.022	0.043
Pipeline	0.787	0.011	0.021	0.679	0.012	0.023

datasets are most similar, having visually similar appearances and cropping, and contain even the same identities. Across the board, so for both recreated cameras and facial recognition models, the scores and curves can not be called statistically significantly equal. Only for the camera 3 datasets when matched with FaceVACS, do the confidence bounds have overlap. But given the full set of results, we do not consider this enough evidence for dataset equality.

FaceVACS did the best job at facial recognition for the normal SCFace dataset, where manually cropping the dataset as a preprocessing step significantly hurts this performance. Maybe FaceVACS is better at finding features when a full head image is presented. Or maybe the manual cropping accidentally cut out features, especially around the jawline, that altered the identity information significantly. If this is the case, than comparing SCFace syntethic and the normal SCFace datasets might be better to cover for this alteration of feature information. However, the difference between the normal SCFace dataset and SCFace synthetic is even larger for FaceVACS, keeping our statement of unsuccessful real world image replication intact. For Deepface, cropping the SCFace images did not significantly alter Deepface's ability to match identities.

We think the shortcomings of our pipeline, especially the lack of a real lens aberration and Bayer filter, had too much impact to successfully recreate image degradation present in the SCFace datasets.

Introduction of fully synthetic identities ("pipeline" scores) made correct facial recognition considerably harder than on the SCFace synthetic dataset, in particular for FaceVACS. The difference in facial recognition performance on these datasets is somewhat counter-intuitive, as the exact same pipeline with the exact same scene parameters is used. Since these variables are equal, we have to look at the high resolution input datasets. The first big difference between the SCFace gallery dataset and our synthetic dataset is the diversity in ethnicity. SCFace overwhelmingly uses individuals with a Caucasian background for their dataset, while our synthetic dataset also contains a significant amount of individuals with an Asian, Latino or African ethnic background. In facial recognition models, their is often a positive bias toward the matching of Caucasian individuals, as a lot of face datasets used for training have a lack of other ethnic representation. Another reason might be the opposite, a lower variability of the synthetic dataset. As mentioned in the Identities section in 3.1.4, Colbois et al. observed less variability in the synthetic dataset generated by their method compared to a real-world dataset. Less variability in facial features between individuals makes the probability of false positive matching higher.

6 Conclusion

The goal of this research was to propose a proof of concept for a realistic but synthetic low resolution face image generation pipeline, which was tested for a potential application in forensic facial recognition settings. Improvements in machine learning in recent years have sprouted interesting techniques for photo-realistic synthetic face and image generation, that could potentially be combined to build such a pipeline. Our first set of research questions was focused on the creation of the pipeline, answered by literature research and practical implementations.

What steps/processes are of importance in a 3D pipeline for generating realistic synthetic face images comparable to images typically found in forensic use cases? The most important steps in our approach are high resolution synthetic face image generation, 3D face reconstruction, a physically based renderer and simulation of camera behaviour.

What are the advantages/disadvantages of using a 3D over a 2D pipeline to generate low resolution face images? At this moment in time, using 3D models appears to be the better approach to creating a synthetic pipeline, due to the possible implementation of physics based lighting and better identity preservation compared to 2D models when changing poses.
Which renderer is suitable for generating images with physically accurate lighting conditions? What is the influence of diffraction of light to create physically accurate low resolution images? PBRT can execute the volumetric path tracing needed for physically accurate particle behaviour of light, while being well documented and therefore easy to implement. However, PBRT is not a spectral renderer, which made the implementation of lights wave behaviour partly unexplored. The effects of diffraction are however shown to be negligible for realistic security camera lens system parameters, leaving only the absence of chromatic aberration due to PBRT's inability to render wave characteristics.
What are other influences of a lens, sensor characteristics and image storage on the degradation of realistic low resolution images? Can

we implement these influences into our pipeline?
A lens system introduces image degradation through wrong projection on the output image, known as optical aberration, which can be modeled with PBRT. However, the lack of realistic lens system schematics made the actual implementation difficult, leading us to use a much simpler pinhole camera model for this proof of concept. Two main contributors to sensor and storage based artifacts on a very low resolution level are Bayer filtering and subsequently compression of the raw image before storage. Only Compression was successfully implemented in our pipeline, by applying the OpenCV JPEG compression algorithm as a post-processing step to the raw pipeline output.
Which models can be used to generate 3D face structures? Single image reconstruction had our preference over multiple image reconstruction, mainly to enable one-to-one gallery to low resolution probe image generation with our pipeline. For identity preservation, the albedo map had to be constructed from the source image and not approximated, which limited the available state-of-the art 3D reconstruction methods. PRNet and DECA were deemed as suitable candidates, with both models having sometimes conflicting trade-offs and shortcomings. PRNet was preferred due to less observable errors and artifacts in the output when combined within the pipeline.
How can subsurface scattering of light within human skin be accurately modeled?
Subsurface scattering in human skin can be approximated by introducing the BSSRDF into the rendering equation. In practice, the volumetric path tracing algorithm can be used to render the subsurface scattering. Models with multiple scattering layers are introduced to approximate skin better, but a single layer model is implemented with PBRT which creates a reasonable approximation for VLR images.
Which model for synthetic face generation is suitable? We were able to incorporate a state-of-the-art conditioned machine learning algorithm to generate a realistic synthetic high resolution gallery dataset, by combining StyleGAN2 and attribute based latent space exploration as introduced by Colbois et al. The combination of the two models made for realistic image generation while making the output controlled. filtering the output of this combined model on pose, age and glasses was required and implemented to create a synthetic dataset compatible with the rest or our pipeline.

The second part of our research had its focus on identification performance with a state-of-the art and open-source facial recognition model, taking its input from our pipeline.

What is the impact of face images with an interpupillary distance of a few pixels on facial recognition? Once the face images have an interpupillary distance that enters the very low resolution domain, so an IPD < 11 pixels, even for ideal face images the facial recognition performance is significantly impacted.
How do low resolution and super-resolution biometric comparisons perform with our very low resolution synthetic dataset, while varying pose, lighting conditions and compression?
Introducing heavy compression into our dataset significantly reduced the performance of state-of-the-art and in particular open-source facial recognition. Super-resolution upscaling does not improve, but can rather hurt performance for these compression cases. Varying pose and lighting directions into something different than frontal does also introduce facial recognition performance loss that can not be bridged by applying super-resolution as a pre-processing step. Only for synthetic datasets with a low amount of noise, shadows or pose variation can super-resolution help recognition performance, but the exact application is really model dependant.
How do low resolution facial recognition methods perform with a real-life low resolution face datasets compared to our synthetic datasets? To make a substantiated comparison between a real-life and our synthetic dataset, we tried to visually approach two VLR SCFace datasets by tweaking our pipeline parameters before generating synthetic datsets with the same and fully synthetic identities. The facial recognition models performed significantly different with the real VLR datasets compared to our synthetic approximations. Our fully synthetic identities also resulted in worse facial recognition performance compared to datasets created with equal pipeline parameters but real identites.

This leads us to answering our overarching research question: *Does the proposed pipeline generate low resolution face images that are comparable to a real-world dataset, making them usable in a forensic setting?* From our experimental results we can conclude that we were not able to fully replicate real-world VLR image degradation, making the proposed pipeline unusable in a forensic setting.

7 Future Work

The pipeline introduced in this paper was not able to recreate real world digital image degradation on a very low resolution scale. But we think, given the extensive background research and presentation of shortcomings, this work can be used as a starting point for future research. As mentioned in our introduction, this research was started with the intention of proof of concept. And although not perfectly executed, as demonstrated with our results, we still believe expanding the general outline of a 3D pipeline, as presented in our background and methodology, could garner promising results in future work.

A key advantage for future research, is the fact that the components of our pipeline are backed by very active research fields. For example the generation of synthetic face images. The framework for a GAN was first introduced in 2014 [94], the first fully synthetic human face images were generated in 2017 [121], while the first groundbreaking version of StyleGAN was made public in 2019 [39]. This means that in a time-span of less than a decade, we went from the first introduction of the underlying techniques to mass generation of photo-realistic synthetic faces. An interesting publication during our own research was the work by Princeton U and Adobe, introducing 3D-FM GAN as a identity preserving and 3D-controllable synthetic face image generator [122]. Their work could for example replace our relatively roundabout way of combining two models to create a conditioned StyleGAN.

Other significant improvements of our pipeline would be the incorporation of a real lens system into our pipeline and extending the physically based renderer to include spectral rendering. Finding detailed lens system specifications, detailing things like exact lens shapes and glass types used, for cameras used in realistic forensic scenarios would be a must. For spectral rendering, software does exist, like the work by Stanford Vista Lab [100], and improved implementations of physically based rendering are still being introduced and developed. When spectral rendering within a pipeline is realized, implementing sensor characteristics like a Bayer filter would also give a great boost towards realism. For 3D face model construction, using multi image reconstruction could also be an interesting expansion. This would negate the ability of the pipeline to work in a one-to-one manner, but could potentially increase the realism of the 3D face/head models.

To get even better insight into the output of a low resolution synthetic face pipeline, the super-resolution experiment performed in our research could be expanded with state-of-the-art super-resolution models. Especially using super-resolution methods specifically intended for upscaling face images could garner interesting results. Also experiments incorporating resolution robust feature extraction methods or mixed-resolution methods are interesting for future work to cover all the bases for low resolution to high resolution image matching.

8 References

1 M. Grgic, K. Delac, and S. Grgic, "Scface - surveillance cameras face database," *Multimedia Tools Appl.*, vol. 51, pp. 863–879, 02 2011.

2 Nederlandse Overheid, "How to make your business gdpr compliant." available at: https://business.gov.nl/running-your-business/business-management/administration/how-to-make-your-business-gdpr-compliant/, accessed: 10-11-2022.

3 Politico and V. Manancourt, "Controversial us facial recognition technology likely illegal, eu body says." available at: https://www.politico.eu/article/clearview-ai-use-likely-illegal-says-eu-data-protection-watchdog/, accessed: 10-11-2022.

4 Y. Peng, L. Spreeuwers, B. Gökberk, and R. Veldhuis, "Comparison of super-resolution benefits for downsampled images and real low-resolution data," in *34th WIC Symposium on Information Theory in the Benelux and the 3rd Joint WIC/IEEE Symposium on Information Theory and Signal Processing in the Benelux 2013*, (Netherlands), pp. 244–251, Werkgemeenschap voor Informatie- en Communicatietheorie (WIC), May 2013. 34th WIC Symposium on Information Theory in the Benelux 2013 ; Conference date: 30-05-2013 Through 31-05-2013.

5 T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," 2019.

6 A. B. Zhang and D. Gourley, "5 - digitising material," in *Creating Digital Collections* (A. B. Zhang and D. Gourley, eds.), Chandos Information Professional Series, pp. 55–72, Chandos Publishing, 2009.

7 ICAO, "Portrait quality (reference facial images for mrtd) - technical report." available at: https://www.icao.int/Security/FAL/TRIP/Documents/TR%20-%20Portrait%20Quality%20v1.0.pdf, accessed: 01-09-2022.

8 Z. Lei, S. Liao, A. Jain, and S. Li, "Coupled discriminant analysis for heterogeneous face recognition," *Information Forensics and Security, IEEE Transactions on*, vol. 7, pp. 1707–1716, 12 2012.

9 B. Gunturk, A. Batur, Y. Altunbasak, M. Hayes, and R. Mersereau, "Eigenface-domain super-resolution for face recognition," *IEEE Transactions on Image Processing*, vol. 12, no. 5, pp. 597–606, 2003.

10 X. Yu, B. Fernando, R. Hartley, and F. Porikli, "Super-resolving very low-resolution face images with supplementary attributes," in *CVPR*, 2018.

11 S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

12 V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," Tech. Rep. UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

13 Y. Feng, S. Yu, H. Peng, Y.-R. Li, and J. Zhang, "Detect faces efficiently: A survey and evaluations," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 1, pp. 1–18, 2022.

14 P. Li, L. Prieto, D. Mery, and P. Flynn, "Face recognition in low quality images: A survey," 2018.

15 Y. Peng, L. J. Spreeuwers, and R. N. Veldhuis, "Low-resolution face recognition and the importance of proper alignment," *IET Biometrics*, vol. 8, no. 4, pp. 267–276, 2019.

16 ISO Central Secretary, "Information technology — biometric data interchange formats — part 5: Face image data," Standard ISO/IEC 19794-5:2005, International Organization for Standardization, Geneva, CH, 2012.

17 INCITS, "Information technology - face recognition format for data interchange," Standard ANSI/INCITS 385-2004, International Committee for Information Technology Standards, 2014.

18 T. Marciniak, A. Chmielewska, R. Weychan, M. Parzych, and A. Dabrowski, "Influence of low resolution of images on reliability of face detection and recognition," *Multimedia Tools and Applications*, vol. 74, 06 2013.

19 British Standards Institution, "En 50132-7: Alarm systems - cctv surveillance systems for use in security applications - application guidelines," Standard BS/EN 50132-7, British Standards Institution, 1996.

20 A. O'Toole, J. Harms, S. Snow, D. Hurst, M. Pappas, J. Ayyad, and H. Abdi, "A video database of moving faces and people," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, pp. 812–6, 06 2005.

21 Y. Peng, "Face recognition at a distance: low-resolution and alignment problems," 2019.

22 P. Li, L. Prieto, D. Mery, and P. J. Flynn, "On low-resolution face recognition in the wild: Comparisons and new techniques," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 2000–2012, aug 2019.

23 Y. Jin, Y. Zhang, Y. Cen, Y. Li, V. Mladenovic, and V. Voronin, "Pedestrian detection with super-resolution reconstruction for low-quality image," *Pattern Recognition*, vol. 115, p. 107846, 2021.

24 I. Marqués and M. Graña, "Face recognition algorithms - proyecto fin de carrera," 2010.

25 ISO Central Secretary, "Information technology — biometrics — overview and application," Standard ISO/IEC TR 24741:2018, International Organization for Standardization, Geneva, CH, 2018.

26 S. Villena, M. Vega, J. Mateos, D. Rosenberg, F. Murtagh, R. Molina, and A. K. Katsaggelos, "Image super-resolution for outdoor digital forensics. usability and legal aspects," *Computers in Industry*, vol. 98, pp. 34–47, 2018.

27 K. Sun, T.-H. Tran, J. Guhathakurta, and S. Simon, "Fl-misr: Fast large-scale multi-image super-resolution for computed tomography based on multi-gpu acceleration," 2021.

28 J. Yang and T. Huang, *Image Super-Resolution: Historical Overview and Future Challenges*, pp. 1–34. 12 2017.

29 W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, vol. 21, pp. 3106–3121, dec 2019.

30 J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

31 J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

32 J. Jiang, C. Wang, X. Liu, and J. Ma, "Deep learning-based face super-resolution: A survey," 2021.

33 O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.

34 C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," 2015.

35 E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Learning face hallucination in the wild," AAAI'15, p. 3871–3877, AAAI Press, 2015.

36 D. Huang and H. Liu, "Face hallucination using convolutional neural network with iterative back projection," pp. 167–175, 09 2016.

37 H. Huang, R. He, Z. Sun, and T. Tan, "Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1698–1706, 2017.

38 C. Chen, D. Gong, H. Wang, Z. Li, and K.-Y. K. Wong, "Learning spatial attention for face super-resolution," *IEEE Transactions on Image Processing*, vol. 30, pp. 1219–1231, 2021.

39 T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2018.

40 X. Yu and F. Porikli, "Ultra-resolving face images by discriminative generative networks," vol. 9909, pp. 318–333, 10 2016.

41 H. Dou, C. Chen, X. Hu, Z. Xuan, Z. Hu, and S. Peng, "Pca-srgan: Incremental orthogonal projection discrimination for face super-resolution," 2020.

42 M. Zhang and Q. Ling, "Supervised pixel-wise gan for face super-resolution," *IEEE Transactions on Multimedia*, vol. 23, pp. 1938–1950, 2021.

43 T. Yang, P. Ren, X. Xie, and L. Zhang, "Gan prior embedded network for blind face restoration in the wild," 2021.

44 M. Li, Z. Zhang, J. Yu, and C. W. Chen, "Learning face image super-resolution through facial semantic attribute transformation and self-attentive structure enhancement," *IEEE Transactions on Multimedia*, vol. 23, pp. 468–483, 2021.

45 E. Ataer-Cansizoglu, M. Jones, Z. Zhang, and A. Sullivan, "Verification of very low-resolution faces using an identity-preserving deep face super-resolution network," 2019.

46 K. Jiang, Z. Wang, P. Yi, G. Wang, K. Gu, and J. Jiang, "Atmfn: Adaptive-threshold-based multi-model fusion network for compressed face hallucination," *IEEE Transactions on Multimedia*, vol. PP, pp. 1–1, 12 2019.

47 E. Verolme and A. Mieremet, "Application of forensic image analysis in accident investigations," *Forensic science international*, vol. 278, p. 137—147, September 2017.

48 C. Herrmann, "Extending a local matching face recognition approach to low-resolution video," in *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 460–465, 2013.

49 Y. Xiao, Z.-G. Cao, L. Wang, and T. Li, "Local phase quantization plus: A principled method for embedding local phase quantization into fisher vector for blurred image recognition," *Information Sciences*, vol. 420, pp. 77–95, 12 2017.

50 O. Meslouhi, Z. Elgarrai, M. Kardouchi, and H. Allali, "Unimodal multi-feature fusion and one-dimensional hidden markov models for low-resolution face recognition," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, p. 1915, 08 2017.

51 S. Biswas, G. Aggarwal, and P. J. Flynn, "Pose-robust recognition of low-resolution face images," in *CVPR 2011*, pp. 601–608, 2011.

52 X. Wei, Y. Li, H. Shen, W. Xiang, and Y. Lu Murphey, "Joint learning sparsifying linear transformation for low-resolution image synthesis and recognition," *Pattern Recognition*, vol. 66, pp. 412–424, 2017.

53 P. Moutafis and I. Kakadiaris, "Semi-coupled basis and distance metric learning for cross-domain matching: Application to low-resolution face recognition," 10 2014.

54 Y. Peng, L. Spreeuwers, and R. Veldhuis, "Likelihood ratio based mixed resolution facial comparison," in *3rd International Workshop on Biometrics and Forensics (IWBF 2015)*, pp. 1–5, 2015.

55 "Thispersondoesnotexist.com." available at: https://thispersondoesnotexist.com/, accessed: 15-09-2022.

56 CNN Business and Rachel Metz, "These people do not exist. why websites are churning out fake images of people (and cats)." available at: https://edition.cnn.com/2019/02/28/tech/ai-fake-faces/index.html, accessed: 15-09-2022.

57 Volkskrant and Laurens Verhagen, "Eindeloos veel gezichten van mensen die niet bestaan." available at: https://www.volkskrant.nl/nieuws-achtergrond/eindeloos-veel-gezichten-van-mensen-die-niet-bestaan~b1018e41/, accessed: 15-09-2022.

58 Zondag met Lubach, "Deepfakes." available at: https://www.youtube.com/watch?v=VSi3o01hcvU, accessed: 15-09-2022.

59 H. Zhou, S. Hadap, K. Sunkavalli, and D. Jacobs, "Deep single-image portrait relighting," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7193–7201, 2019.

60 R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka, "Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows," *ACM Trans. Graph.*, vol. 40, May 2021.

61 L. Colbois, T. de Freitas Pereira, and S. Marcel, "On the use of automatically generated synthetic image datasets for benchmarking face recognition," in *International Joint Conference on Biometrics (IJCB 2021)*, 2021. Accepted for Publication in IJCB2021.

62 L. Tran, X. Yin, and X. Liu, "Disentangled representation learning gan for pose-invariant face recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1283–1292, 2017.
63 R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis," 2017.
64 N. Kurachi and M. Stark, *The Magic of Computer Graphics - Landmarks in Rendering*. USA: A. K. Peters, Ltd., 1st ed., 2011.
65 "Physically based rendering - from theory to implementation (third edition)," in *Physically Based Rendering - From Theory to Implementation (Third Edition)* (M. Pharr, W. Jakob, and G. Humphreys, eds.), p. IFC, Boston: Morgan Kaufmann, third edition ed., 2017.
66 Axis Communications, "Axis p1435-le network camera datasheet." available at: https://www.axis.com/dam/public/74/23/8f/datasheet-axis-p1435-le-network-camera--en-US-270676.pdf, accessed: 19-09-2022.
67 S. Ray, "Chapter 11 - aberration - defects in imaging systems," in *Applied Photographic Optics: Lenses and Optical Systems for Photography, Film, Video, Electronic and Digital Imaging*, pp. 82–98, Focal, 2002.
68 S. Ray, "Chapter 13 - colour correction of lenses," in *Applied Photographic Optics: Lenses and Optical Systems for Photography, Film, Video, Electronic and Digital Imaging*, pp. 112–121, Focal, 2002.
69 S. Ray, "Chapter 16 - resolving power of lenses and imaging systems," in *Applied Photographic Optics: Lenses and Optical Systems for Photography, Film, Video, Electronic and Digital Imaging*, pp. 145–154, Focal, 2002.
70 R. A. Schowengerdt, "Chapter 3 - sensor models," in *Remote Sensing (Third Edition)* (R. A. Schowengerdt, ed.), pp. 75–XIV, Burlington: Academic Press, third edition ed., 2007.
71 F. Bettonvil, "Fisheye lenses," *WGN, Journal of the International Meteor Organization*, vol. 33, pp. 9–14, 01 2005.
72 M. RadhaKrishna, M. Govindh, and P. Veni, "A review on image processing sensor," *Journal of Physics: Conference Series*, vol. 1714, p. 012055, 01 2021.
73 J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*. Optical Science and Engineering, CRC Press, 2017.
74 S. Anand and L. Priya, *A Guide for Machine Vision in Quality Control*. CRC Press, 2019.
75 Vincent Bockaert, "Sensor sizes." archived version available at: https://web.archive.org/web/20130125090640/http://www.dpreview.com/glossary/camera-system/sensor-sizes, accessed: 26-09-2022.
76 Digital Photography Review, "Making (some) sense out of sensor sizes." available at: https://www.dpreview.com/articles/8095816568/sensorsizes, accessed: 26-09-2022.
77 D. R. Bull, "Chapter 4 - digital picture formats and representations," in *Communicating Pictures* (D. R. Bull, ed.), pp. 99–132, Oxford: Academic Press, 2014.
78 A. Bovik, *The Essential Guide to Image Processing*. Elsevier Science, 2009.
79 E. Ramon, J. Escur, and X. G. i Nieto, "Multi-view 3d face reconstruction in the wild using siamese networks," in *ICCV 2019 Workshop on 3D Face Alignment in the Wild Challenge Workshop (3DFAW)*, (Seoul, South Korea), IEEE/CVF, IEEE/CVF, 11/2019 2019.
80 S. Sanyal, T. Bolkart, H. Feng, and M. Black, "Learning to regress 3D face shape and expression from an image without 3D supervision," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 7763–7772, June 2019.
81 C. Li, A. Morel-Forster, T. Vetter, B. Egger, and A. Kortylewski, "To fit or not to fit: Model-based face reconstruction and occlusion segmentation from weak supervision," *arXiv preprint arXiv:2106.09614*, 2021.
82 Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, and X. Tong, "Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set," in *IEEE Computer Vision and Pattern Recognition Workshops*, 2019.
83 Y. Feng, H. Feng, M. J. Black, and T. Bolkart, "Learning an animatable detailed 3D face model from in-the-wild images," vol. 40, 2021.
84 T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, pp. 194:1–194:17, 2017.
85 J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li, "Towards fast, accurate and stable 3d dense face alignment," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
86 Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3d face reconstruction and dense alignment with position map regression network," in *ECCV*, 2018.
87 Y. Feng, V. Choutas, T. Bolkart, D. Tzionas, and M. J. Black, "Collaborative regression of expressive bodies using moderation," in *International Conference on 3D Vision (3DV)*, 2021.
88 Z. Ruan, C. Zou, L. Wu, G. Wu, and L. Wang, "SADRNet: Self-aligned dual face regression networks for robust 3d dense face alignment and reconstruction," *IEEE Transactions on Image Processing*, vol. 30, pp. 5793–5806, 2021.
89 Z. Chai, H. Zhang, J. Ren, D. Kang, Z. Xu, X. Zhe, C. Yuan, and L. Bao, "Realy: Rethinking the evaluation of 3d face reconstruction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.
90 H. Wann Jensen, S. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," vol. 35, 09 2002.
91 C. Donner and H. Wann Jensen, "A spectral bssrdf for shading human skin," pp. 409–417, 01 2006.
92 A. Krishnaswamy and G. Baranoski, "A biophysically-based spectral model of light interaction with human skin," *Computer Graphics Forum*, vol. 23, pp. 331 – 340, 09 2004.
93 H. Nguyen, *Gpu Gems 3*. Addison-Wesley Professional, first ed., 2007.
94 I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.
95 M. Pharr, W. Jakob, and G. Humphreys, "Pbrt - physically based rendering." available at: https://www.pbrt.org/, accessed: 17-09-2022.
96 W. Jakob, S. Speierer, N. Roussel, and D. Vicini, "Dr.jit: A just-in-time compiler for differentiable rendering," *Transactions on Graphics (Proceedings of SIGGRAPH)*, vol. 41, July 2022.
97 W. Jakob, S. Speierer, N. Roussel, M. Nimier-David, D. Vicini, T. Zeltner, B. Nicolet, M. Crespo, V. Leroy, and Z. Zhang, "Mitsuba 3 - physically based renderer," 2022. available at: https://www.mitsuba-renderer.org/, accessed: 17-09-2022.
98 T. Cuypers, S. B. Oh, T. Haber, P. Bekaert, and R. Raskar, "Ray-based reflectance model for diffraction," 2011.
99 M. Radziszewski, K. Boryczko, and W. Alda, "An improved technique for full spectral rendering," *Journal of WSCG*, vol. 17, 01 2009.
100 Stanford Vista Lab, "Vista lab's spectral modification of pbrt, github code." available at: https://github.com/scienstanford/pbrt-v4, accessed: 26-09-2022.
101 G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
102 A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks)," in *International Conference on Computer Vision*, 2017.
103 D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
104 T. Gerig, A. Morel-Forster, C. Blumer, B. Egger, M. Luthi, S. Schoenborn, and T. Vetter, "Morphable face models - an open framework," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 75–82, 2018.
105 N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
106 R. Rothe, R. Timofte, and L. V. Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 144–157, 2018.
107 Adam Geitgey, "face-recognition - python library." available at: https://pypi.org/project/face-recognition/, accessed: 26-09-2022.
108 B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," tech. rep., CMU-CS-16-118, CMU School of Computer Science, 2016.
109 Cognitec, "Facevacs." available at: https://www.cognitec.com/facevacs-technology.html, accessed: 29-10-2022.
110 S. I. Serengil and A. Ozpinar, "Lightface: A hybrid deep face recognition framework," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 23–27, IEEE, 2020.
111 J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," 2019.
112 J. Deng, J. Guo, J. Yang, N. Xue, I. Cotsia, and S. P. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
113 S. I. Serengil and A. Ozpinar, "Deepface github." available at: https://github.com/serengil/deepface, accessed: 29-10-2022.
114 C. Zeinstra, R. Veldhuis, and L. Spreeuwers, "How random is a classifier given its area under curve?," in *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–4, 2017.
115 Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," 02 2015.
116 D. Bamber, "The area above the ordinal dominance graph and the area below the receiver operating characteristic graph," *Journal of Mathematical Psychology*, vol. 12, no. 4, pp. 387–415, 1975.
117 E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach," *Biometrics*, vol. 44, no. 3, pp. 837–845, 1988.
118 J. A. Hanley and K. O. Hajian-Tilaki, "Sampling variability of nonparametric estimates of the areas under receiver operating characteristic curves: An update," *Academic Radiology*, vol. 4, no. 1, pp. 49–58, 1997.
119 X. Sun and W. Xu, "Fast implementation of delong's algorithm for comparing the areas under correlated receiver operating characteristic curves," *Signal Processing Letters, IEEE*, vol. 21, pp. 1389–1393, 11 2014.
120 H. B. Mann and D. R. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50 – 60, 1947.
121 ALAgrApHY, "This person does not exist - neither will anything eventually with ai." available at: https://alagraphy.medium.com/this-person-does-not-exist-neither-will-anything-if-artificial-intelligence-keeps-learning-1a9fcba728f, accessed: 07-12-2022.
122 Y. Liu, Z. Shu, Y. Li, Z. Lin, R. Zhang, and S. Y. Kung, "3d-fm gan: Towards 3d-controllable face manipulation," 2022.