applied sciences

MDPI

Article

Robust and High-Fidelity 3D Face Reconstruction Using Multiple RGB-D Cameras

Haocheng Peng [1], Li Yang [1] and Jinhui Li [2,*]

[1] Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou 310018, China
[2] Science and Technology on Near-Surface Detection Laboratory, Wuxi 214035, China
* Correspondence: jinhui_lab@126.com

Abstract: In this paper, we propose a robust and high-fidelity 3D face reconstruction method that uses multiple depth cameras. This method automatically reconstructs high-quality 3D face models from aligned RGB-D image pairs using multi-view consumer-grade depth cameras. To this end, we mainly analyze the problems in existing traditional and classical multi-view 3D face reconstruction systems and propose targeted improvement strategies for the issues related. In particular, we propose a fast two-stage point cloud filtering method that combines coarse filtering and fine filtering to rapidly extract the reconstructed subject point cloud with high purity. Meanwhile, in order to improve the integrity and accuracy of the point cloud for reconstruction, we propose a depth data restoration and optimization method based on the joint space–time domain. In addition, we also propose a method of multi-view texture alignment for the final texture fusion session that is more conducive for fusing face textures with better uniformity and visual performance. The above-proposed methods are reproducible and can be extended to the 3D reconstruction of any subject. The final experimental results show that the method is able to robustly generate 3D face models having high geometric and visual quality.

Keywords: face reconstruction; point cloud filtering; texture fusion; 3D simulation

Citation: Peng, H.; Yang, L.; Li, J. Robust and High-Fidelity 3D Face Reconstruction Using Multiple RGB-D Cameras. *Appl. Sci.* **2022**, *12*, 11722. https://doi.org/10.3390/app122211722

Academic Editor: Luis Javier Garcia Villalba

Received: 23 October 2022
Accepted: 15 November 2022
Published: 18 November 2022

1. Introduction

3D face reconstruction has become a basic and important part of virtual reality tasks owing to the increasing demand for virtual human–computer interaction, smart medical applications, and 3D movies and games. In recent years, considerable research work has been conducted on 3D face reconstruction from a single RGB image or RGB-D image using the powerful capabilities of deep convolutional neural networks (DCNNs). Researchers generally use DCNNs to learn the mapping relationship between the original image data and the parameterized 3D morphable model (3DMM), thereby transforming the problem of face reconstruction into an optimization problem of solving parameters [1–5]. Although these methods have produced many impressive results in recent years, they are still not widely applicable owing to most of them requiring good initialization parameters, and the reconstruction quality is largely dependent on the training dataset. Similarly, high-precision face reconstruction equipment and methods, such as the laser ranging scanner [6], are greatly limited in practical application due to the high cost of special hardware and the cumbersome process of manual post-processing.

With the latest advancements in 3D scanning technology and the progress in consumer depth cameras, it is possible to obtain RGB-D image pairs in real time at an acceptable cost. The availability of depth and RGB data dramatically strengthens the possibility of wide applications of 3D face reconstruction using multi-view depth cameras in practical scenes.

Therefore, our goal is to build a high-quality face reconstruction system with the consumer-grade depth camera and ensure that the method is robust enough (unlike the

abovementioned DCNNS method, which is deeply affected by sample differences) while keeping the low cost and efficient, so that face reconstruction can achieve actual application viability.

However, in practice, high-quality face reconstruction tasks that rely on aligned RGB-D image pairs are not as easy as described. Low-cost RGB-D cameras, including Microsoft Kinect, Asus Xtion, and Intel RealSense, share the common characteristic that the single depth frames acquired contain severe data noise, especially at the edges of the discontinuities in the depth domain. The point cloud generated from the RGB-D image acquired by the depth camera mainly contains two noise parts. One is the accidental noise caused by the acquisition environment, such noise is minor and contingent, and the impact of this kind of noise will be as small as possible under the premise that the camera measurement quality is stable enough. The second is the edge noise (as shown in Figure 1) at the discontinuity in the depth domain, which is caused by inaccurate depth measurement of the background due to the depth measurement principle and ambiguity of boundaries. This noise is attached near the edge of the foreground and densely distributed. There is no doubt that noise hinders the accurate reconstruction of a 3D model of the observed subject unless it can be removed by an efficient filtering algorithm. However, the regular point cloud filtering methods, such as statistical filtering, radius filtering, and voxel filtering, all have obvious shortcomings in removing the abovementioned edge noise. In addition, when they are used together, they often ensure better performance but have an issue with efficiency. At the same time, due to the differences in lighting from different views and the differences in manufacturing between the cameras, the color inconsistency of the RGB images captured by the cameras from different views is also quite severe, this certainly increases the difficulty of recovering the overall uniformity of the textures for the 3D model from the images.

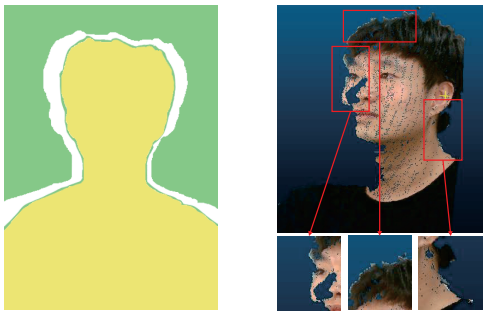

Figure 1. Edge noise at discontinuities in the depth domain. (**a**) Schematic diagram of edge noise. The green area is the background and the yellow area is the foreground. The white area is the depth discontinuity area where the foreground and background are separated, and the green area attached to the outer contour of the yellow foreground is the edge noise, which is caused by the depth camera measurement principle. (**b**) Visualization of edge noise on a point cloud of real data. The detailed image of the edge noise is contained in the red box.

In this paper, we propose a new and detailed method for efficient and high-quality face reconstruction based on commercial, low-cost depth cameras that are robust and can easily be scaled to reconstruct arbitrary subjects. In this multi-view face reconstruction framework, we do not rely on significant data priors, which frees our method from the problem of different samples behaving differently. The fast and powerful noise filtering capability of our method ensures the removal of the deep-rooted edge noise (which severely hinders the point cloud stitching and fusion) and the accuracy of face reconstruction. In addition, our method has a firm texture recovery capability, which guarantees that the reconstructed face meshes with natural texture even in extreme cases. We hope that the mature process and robustness of our method will improve the applicability of 3D face reconstruction in future scenarios.

In summary, the main contributions of this study are as follows:

- We propose a novel, fast two-stage point cloud filtering method, in which we simplified the filtering process through this two-stage process and employed image matting technology based on deep learning for the first time to effectively remove the edge noise at discontinuities in the depth domain.
- We propose an improved time–space joint depth data restoration and optimization method, which ensures the accuracy and integrity of the point cloud.
- We propose a novel method of multi-view texture alignment, which reduces the burden of texture fusion and achieves overall unity texture recovery for 3D models.
- We demonstrate the excellent facial shape and texture reconstruction performance of this method and verify the practicality of the above approaches with both qualitative and quantitative experiments.

2. Related Work

Our system uses multi-view depth cameras to reconstruct a high-quality 3D face model; so, in this chapter, we discuss and analyze methods and systems for acquiring 3D face models using depth data. Generally speaking, such methods can be classified into two categories: data-driven methods and model-based methods.

Data-driven methods: Data-driven 3D face reconstruction methods usually achieve complete 3D face reconstruction by integrating depth images acquired in real time into a final global 3D model. Ref. [7] proposed a real-time reconstruction of a 3D face model using a low-cost and low-resolution depth camera with frame tracking from the current frame to the reference frame (a similar approach was used in Ref. [8]). Specifically, the depth image of the first frame is initialized and regarded as a reference frame, and then the iterative closest point (ICP) algorithm is used to align the point cloud corresponding to the subsequent depth frame to the reference frame. They also introduced the bump image framework to parameterize the 3D facial surface in a so-called cylindrical coordinate system. Although this method can generate a high-quality 3D face model, it also has obvious shortcomings. If the pose of the current frame relative to the reference frame changes too much, the camera tracking system will fail due to the inaccurate pose estimate, and the cylindrical bump images will only record geometric information and not texture information. Ref. [9] proposed a new method of Kinect fusion, which fuses all the depth data streams of the Kinect depth sensor into the truncated signed distance function (TSDF) volumetric scene by using a coarse-to-fine ICP algorithm (acting on all available depth data) to track the pose of the current sensor. This framework is robust to drift. Ref. [10] further extended this method to capture dynamic action in the foreground. Although the Kinect fusion approach is generic and its application to face reconstruction is optimized for noise in a single depth frame, there is still a significant gap compared with high-quality scans. Ref. [11] introduced a method for generating accurate 3D faces based on RGB-D sensors. They extend and use bump images to reconstruct faces with high accuracy while keeping memory consumption low. Although this method still works for unknown face profiles, the validity of the method by Anasosalu et al. is based on the assumption that facial expressions do not change with small head movements. Ref. [12] proposed a solution for the marker-less reconstruction of non-rigidly deformed subjects using a GPU pipeline for non-rigidly aligned, real-time RGB-D images with a smoothed template. However, the solution requires specific stereo matching algorithms to estimate the real-time depth data. Ref. [13] proposed a 3D face reconstruction algorithm with continuous face geometry and texture. However, their method requires special settings, such as two mirrors placed on both sides of the face and image calibration using the mirror. Ref. [14] proposed a method for reconstructing full-body meshes with high-resolution texture from RGB-D video. They used incident illumination to accurately estimate the surface geometry and albedo based on the coarse human mesh from RGB-D tracking, and further used photometric constraints to perform more detailed geometric and texture estimation in a self-supervised manner.

However, their approach is based on the assumption that the deformed invisible areas are the same or similar as in the keyframes, which is not always correct.

Model-based methods: These methods mainly use a standard variability model as a template and learn the corresponding texture and shape parameters from input data; thus, they adapt the standard template to the input face. In fact, in many applications, 3DMM is suitable for RGB images [15–17] as original data for face rendering, expression simulation, etc. However, in this section, we focus on applications that use low-cost depth cameras with depth images or RGB-D image pairs as input data to generate the corresponding 3D face models with 3DMM. Ref. [1] described a variability model for depth images acquired by a depth camera. Specifically, the features detected by the aligned RGB images allow the template model to complete a rough registration with the depth frame. The template is then non-rigidly aligned with the depth frame data. Unfortunately, this method eventually produces results that are biased towards the template and away from the input data. Ref. [18] proposed a novel facial geometry modeling and reflectivity synthesis procedure to reconstruct the complete head in less than half a minute by recording an RGB-D video of the user rotating the head. The method first selects high-quality frame data available for reconstruction through a two-stage frame screening procedure; then, it recovers facial shapes by fitting 3DMM parameters from multi-view RGB-D data in a learning-based manner. Although this method can recover accurate geometric shapes and synthesize realistic detailed textures, the method of recording depth video is cumbersome for practical applications, and its input data are redundant. Ref. [19] proposed a fine-grained reconstruction network for the problem of fine-grained geometric loss in 3DMM template reconstruction; they focused on shape modification by distorting the network input and output to UV space and constructing a new, fine-grained 3D face dataset from the RGB-D data. Ref. [20] proposed an efficient face reconstruction and real-time facial expressions based on RGB-D videos for VR interaction. Specifically, the depth image is first captured using an RGB-D camera, and a rough face model is then quickly reconstructed. The user's specific avatar is generated using standard facial model templates and shape morphing techniques. In addition, RGB data is used to track the user's head movements and localize facial features. The facial features are automatically linked to the facial model. Finally, the user's avatar can be driven by the facial features using Laplacian deformation. However, it still cannot eliminate the algorithm robustness problem caused by sample differences.

Among all the above methods, the first type of strategy offers real-time processing and high accuracy but the visual performance is generally not good enough; it requires redundant data to generate a complete face and does not perform detailed processing of textures. Meanwhile, the second type of strategy does not require any post-processing step, but usually depends on a large amount of training data and differences in accuracy between different samples due to the fitting principle, which is not conducive to practical applications. Therefore, we hope to combine the advantages of these two types of methods and adopt a classical multi-view method to achieve considerable reconstruction performance while maintaining high accuracy and strong generalization so that 3D face reconstruction can be more widely applied.

3. Materials and Methods

3.1. Overview

The system we designed uses three depth cameras positioned for left, middle, and right views to acquire aligned RGB-D image pairs for a complete 3D face reconstruction. The relative positioning of depth cameras ensures that a person's face (from left ear to right ear) can be fully captured with a sufficiently overlapping field of view, as shown in Figure 2.

Finally, the Intel RealSense D435 camera is adopted with comprehensive consideration for measurement accuracy and the available depth range. In practice, the RealSense D435

is able to capture both 1280 × 720 depth and RGB images at 30 frames per second (fps) on our machine, which is the final resolution we used for reconstruction.

Figure 2. Schematic diagram of the acquisition platform. (**a**) The acquisition platform contains three calibrated RealSense D435 cameras and a light source. (**b**) The acquisition platform with the users.

Figure 3 shows a diagram of the software system. First, we jointly calibrate the three relatively fixed cameras, using the middle camera as the reference coordinate system, and aligning the left and right cameras with the middle camera, respectively, as the initial parameters of the point cloud registration. The denoising process takes the point cloud of each viewpoint as input and generates a clean and accurate point cloud by filtering out the noise and repairing the depth. The denoised and restored point clouds are used for fine registration by the ICP algorithm and fusion. The output is further processed by operations such as surface triangulation to complete the geometric reconstruction of the 3D face. Finally, texture mapping takes as input the RGB images after multi-view alignment to recover the overall uniform face texture for the geometric model. Sections 3.2–3.4 present the blocks of denoising and multi-view texture alignment (the orange block in the software diagram) in detail.

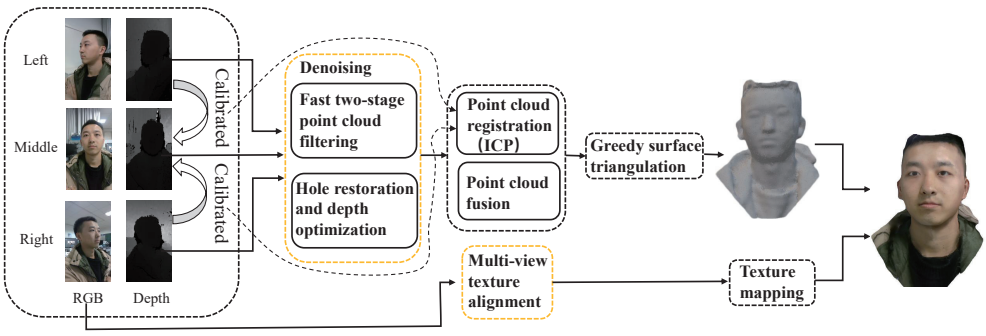

Figure 3. Overview of the face reconstruction system.

3.2. Fast Two-Stage Point Cloud Filtering

In this section, we mainly describe a fast two-stage point cloud filtering method of coarse filtering based on a distance prior and fine filtering based on image matting, to ensure that the point cloud of the reconstructed subject was clean enough. The stages are, respectively, used to remove the overall and edge noise at the discontinuity in the depth domain. The schematic diagram of the whole process is shown in Figure 4.

Specifically, we ensure that there is no other object between the face and the camera when we capture the face with the device shown in Figure 2. Otherwise, the complete 3D face reconstruction task would not be possible. This way, the face is the closest foreground subject to the camera; so, we first extract the current minimum value of depth $depth_{min}$ by traversing the Z coordinates of all points in the point cloud, and the point to which the minimum depth value belongs is the closest point in the face to the XOY plane of the camera coordinate system. Meanwhile, based on the assumption that the maximum front-to-back diameter of a normal human head should not exceed β millimeters (mm), we

perform the first coarse filtering of the reconstructed point cloud based on the following distance interval:

$$\{P|\forall p \in P, depth_{min} \leq p_z \leq depth_{min} + \beta\} \quad (1)$$

where P is the point cloud after coarse filtering for each view, p is a point in the P, and p_z denotes the Z coordinate of p. In this way, we only need a simple traversal to obtain knowledge of the face position, which can remove most of the noise in the non-face areas of the point cloud. In order to effectively remove the remaining noise, especially the edge noise at the discontinuities in the depth domain, we combine the image matting method with deep learning for the first time to guide the fine filtering completely and accurately filter this noise.

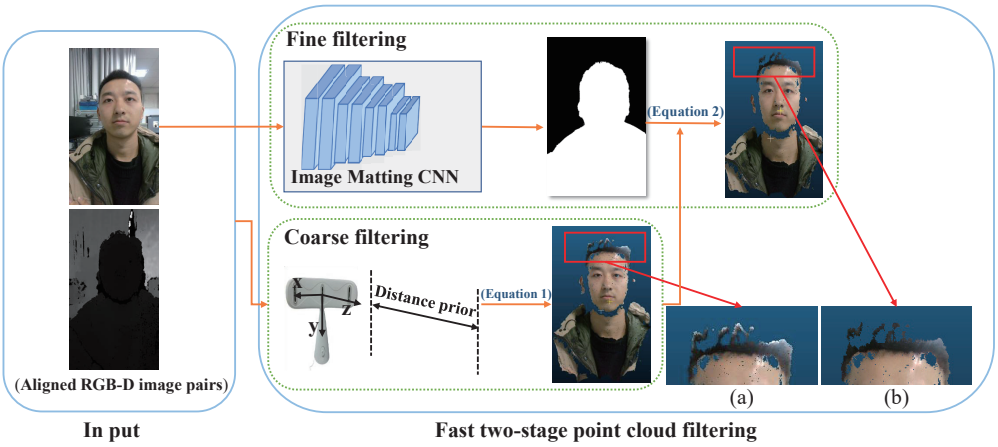

Figure 4. Schematic diagram of two-stage point cloud filtering. (**a**) Detailed image after coarse filtering. (**b**) Detailed image after fine filtering. It can be seen that the noise at the edges (curtains and lights in the background) of the hair is filtered out well after fine filtering.

In this module, an advanced work called Robust High-Resolution Video Matting (RVM) [21], which adopts MobileNetV3-Large [22] as an effective backbone, is used in our image matting tasks that have a human as the foreground. The whole architecture consists of an encoder that extracts the features of an individual frame, a recurrent decoder that aggregates temporal information, and a deep guided filter module for high-resolution upsampling. RVM is designed explicitly for robust human video matting; however, it can still show excellent performance in image matting and can handle matting in real-time without additional input, which is why we chose it. To enhance the robustness of the image matting task, we combine multiple data sets, including VideoMatte240K (VM) and Adobe Image Matting (AIM), to train the network according to the training strategy in this study. The VM contains 484 4K/HD video clips; in the AIM, we select only 420 images containing humans for training. The MobileNetV3 backbone is initialized with pretrained weights and uses a 1×10^{-4} learning rate, while the rest of the network uses 2×10^{-4}. The final testing efficiency is about 10 ms. Please refer to Section 4 of the original article for specific training strategies.

Its network output, which is the transparency image first proposed by [23], represents the mixing ratio of foreground and background. We know that a larger transparency value means that the foreground is more visible than the background in the current pixel. On the contrary, the background is relatively more obvious. The transition region of transparency should theoretically appear in the region of foreground and background separation. Thus, we use the transparency value as the confidence level of the pixel belonging to the foreground (i.e., the higher the transparency value, the higher the probability that the pixel belongs to the foreground) to filter out the uncertain pixels in the region of foreground and background separation. This is not possible with conventional filtering and segmentation algorithms.

In particular, we overlay the result of the coarse filtering with a fine filter based on matting, while using a transparency value of $\alpha \geq C_{th}$ as the criterion for edge points retention:

$$\{P|\forall p \in P, projection(p)_{\alpha} \geq C_{th}\} \tag{2}$$

where $Projection(\cdot)$ is the projection function, which indicates the two-dimensional pixel point in the image that corresponds to the point in the point cloud, C_{th} indicates the confidence threshold for retention, and $Projection(p)_{\alpha}$ indicates the transparency value corresponding to the projection point of p. In fact, in the final experiment, we set C_{th} to 0.8; filtering by this criterion allows for some loss of the data that belong to the foreground object at the edge. The lost edge data at the overlap of the point clouds are redundant. For the edge data lost at the non-overlapping region, the reason why we adopt this strict limiting value is that we hope the edge parts are kept as accurately as possible, and we do not want the noise of non-face parts to affect the smoothness and accuracy of the final model.

3.3. Hole Restoration and Depth Optimization

To perform dense, high-precision reconstruction, we also expect the data for the target point cloud to be complete and sufficiently accurate. Therefore, this section focuses on the hole restoration and depth optimization of the current point cloud by combining the redundant depth data in the temporal and spatial neighborhoods of the depth frame.

Specifically, considering the performance of the RealSense D435 at frame rates up to 30 fps, we take N frames of depth data at the moment of acquisition and its subsequent time domain, and we define the final depth constraint as follows:

$$D^*(i,j) = mid(D^{(1)}(i,j), D^{(2)}(i,j) \cdots D^{(N)}(i,j)) \tag{3}$$

where $D^*(i,j)$ denotes the optimized depth value at pixel coordinate (i,j) in the depth image, $D^{(k)}(i,j)$ denotes the depth value at pixel coordinate (i,j) of the k-th $(1 \leq k \leq N)$ frame depth image in the captured N frames of depth data, and $mid(\cdot)$ denotes the median operation performed on the current set.

In addition, based on the results of matting in Section 3.2, we can obtain the face region to be repaired; further, for the points that are still invalid (points without depth value) in the face region after the median operation, we take the depth value of the spatial neighborhood to further repair it. Let $D^{hole}(x,y)$ be the hole point of the face at the (x,y) position in the depth image; we repair this type of point by the available depth in its spatial neighborhood and its weight. The specific constraints are defined as follows:

$$D^{hole}(x,y) = \frac{\sum_{(i,j)\in R}(W(i,j) * D(i,j))}{\sum_{(i,j)\in R} W(i,j)} \tag{4}$$

$$W(i,j) = \frac{1}{(i-x)^2 + (j-y)^2} * S(i,j) \tag{5}$$

$$S(i,j) = \begin{cases} S(i,j), S(i,j) \geq \alpha_{th} \\ 0, otherwise \end{cases} \tag{6}$$

where R is the set of neighborhood points of the hole point at the (x,y) position; $W(i,j)$ and $D(i,j)$ represent, respectively, the weight and depth value of the neighborhood point at the (i,j) position; and the specific definition of the weight is shown in Equation (5). $S(i,j)$ is the transparency value at the (i,j) position in the transparency image described in Section 3.2. It is important that the spatial neighborhood used for repair should have explicit properties, which means that the holes of a certain subject should be repaired with the neighborhood data of the same subject. The cut-off value α_{th} is to ensure that the spatial neighborhood used for restoration mentioned in this section belongs to the face.

3.4. Multi-View Texture Alignment

After we solve the problem of geometric accuracy in face reconstruction, we cannot ignore the problem of texture fusion if we want to ensure 3D face model reconstruction with high-quality geometric and visual performance. Although an overlap of textures can be made almost unnoticeable by the pixel weighting, it is not enough to solve the problem of color non-conformity in the captured RGB images caused by different views and individual camera differences.

Inspired by professional color calibration methods [24], we assume that colors observed from the same spatial point at different views are linear, as shown in Equation (7):

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = ColorMatrix * \begin{bmatrix} R \\ G \\ B \\ 1 \end{bmatrix} \tag{7}$$

where the input and output of the expression are the color representation of different pixels corresponding to the same spatial point in images acquired from different views, and $ColorMatrix \in R^{3\times4}$ is the color mapping matrix.

We design the solution as follows. First, we take a captured RGB image of a human face with standard color as a template; then, we use the color transfer [25] in traditional image processing algorithms to align all the image colors to the template to prevent excessive differences between the colors of images captured by different cameras. Then, because we have the aligned RGB-D image pairs, each point in the point cloud has not only its spatial location information but also its corresponding color information. Next, we use the results of the ICP algorithm to obtain the matching point pairs in different views of the point cloud using the following point-matching constraints:

$$\left\{ (p_i, q_{j^*}) \middle| \left\| Rp_i + t - q_{j^*} \right\|^2 \le r^2 \right\} \tag{8}$$

$$j^* = \underset{1\le j\le N}{\operatorname{argmin}} \left\| Rp_i + t - q_j \right\|^2 \tag{9}$$

where R, t are the results obtained by the ICP algorithm; p_i is the i-th point in the source point cloud (the source point cloud is the point cloud in the left and right views); q_j is the j-th point in the target point cloud (the target point cloud is the point cloud in the middle view); N is the number of points in the target point cloud; j^* is the label of the nearest point in the target point cloud to p_i in the source point cloud; and r is the hyperparameter used as the upper limit for the distance between p_i and q_{j^*}, above which the points cannot be regarded as matching points. Based on the matched point pairs obtained by Equation (8), the corresponding color matching pairs $(p_i^{rgb}, q_{j^*}^{rgb})$ are obtained. p_i^{rgb} represents the color information (RGB value) of the i-th point in the source point cloud, and $q_{j^*}^{rgb}$ represents the color information of the j^*-th point in the target point cloud. Finally, the corresponding color matching pairs are used to solve the following optimization problem to determine the optimal color matrix $ColorMatrix^*$:

$$ColorMatrix^* = \underset{ColorMatrix}{\operatorname{argmin}} \sum_{i,j^*} \left\| ColorMatrix * p_i^{rgb} - q_{j^*}^{rgb} \right\|^2 \tag{10}$$

This is a classical least-squares problem [26], which can be efficiently solved with the orthogonal Procrustes algorithm [27,28] using singular value decomposition (SVD). $ColorMatrix^*$ is then used to adjust the source photo to a new color. This ensures that the left and right view textures are the same color as the middle view textures.

4. Experiments

In this section, we describe the experiments performed to evaluate our method. The experiments mainly consisted of quantitative and qualitative experiments. Note that there were no public datasets available that could be used as our input. Thus, in the first part, we quantitatively evaluate our system and the methods proposed above using synthetic data. In the second part, we qualitatively analyze the final reconstruction results with a real dataset.

Test data. In our experiments, we used both synthetic and real test data. The real data were acquired by our system using the RealSense D435 camera, and the synthetic data were generated from the high-fidelity face data acquired by [29]. We generated synthetic data for each view by setting up three virtual depth cameras and recording the relative positions between virtual depth cameras as the original calibration parameters in the system, which are consistent with the real system settings. In addition, to simulate the real photography situation, we also added random noise (including accidental and edge noise) to the depth data, especially the edge noise similar to the real data, to prove the effectiveness of our method. In the synthetic data, each depth image is disturbed by random noise in the interval $[-a, a]$, and the value of a is taken from 1 mm to 10 mm at an interval of 1 mm. Note that the accidental error of the depth data acquired by the RealSense D435 camera after calibration at short distances should be less than 2 mm.

4.1. Quantitative Comparison

We first compared our filtering algorithm with other classical filtering algorithms by mesh reconstruction error. We used the ICP algorithm to densely align each of our generated 3D face meshes with the ground truth meshes corresponding to the synthetic data, and we followed [30] using the average symmetry point-to-plane distance as the evaluation metric for this error measure. The normalized point-to-plane errors on reconstructed face mesh with different filtering methods are shown in Table 1 in mm. We can see in the final results that our method is on par with the best-performing method (radius filtering) for mean error, and very close to the best-performing method (statistical filtering) for standard deviation. The reason our method is not significantly better than the other methods we analyzed is that the accidental noise we added to the synthetic data was not explicitly filtered in our fast two-stage filtering method, allowing other methods to narrow the gap in accuracy with our method. However, in practice, the range of accidental noise of the depth camera we used is less than 2 mm under calibration, which means that the accidental noise caused by the regular depth camera is minimal (much smaller than the interval of ±10 mm we set); so, this experimental condition is actually unfavorable for us. In other words, the smaller the interval of the level of accidental noise, the higher the accuracy of our method will be. Even so, our method still achieves the best performance. The reason why we chose such a considerable error interval (beyond the conventional error range) is that we feel if we want to promote the practical application of our face reconstruction method, we should overcome the drawbacks of the lower cost and worse measurement quality of the consumer-grade depth cameras that bring more accidental noise. We set a considerable error interval that all the depth cameras we know can satisfy.

Table 1. Comparison of reconstruction accuracy of mesh on synthetic data using point-to-plane distance with different filtering methods. The table reports the mean error (Mean) and the standard deviation (Std).

Method	Mean	Std
Voxel filtering	1.75	0.272
Statistical filtering	1.46	0.147
Radius filtering	1.42	0.165
Fast two-stage filtering	1.42	0.153

We also tested the speed of our filtering method and compared it with voxel filtering, radius filtering, and statistical filtering. Fast face reconstruction can now be effectively applied in virtual reality interaction scenarios, such as future 3D digital conferences and the hot metaverse games of today. A real-time or quasi-real-time 3D reconstruction of the user's face is required to achieve real-time face view correction and rendering of the digital face after correction. Thus, time consumption must be taken into consideration for the vision of the practical application of face reconstruction; as point cloud filtering is an essential component of the face reconstruction process, the evaluation of the filtering speed is also necessary. We used the PCL library implementations of statistical filtering, radius filtering, and voxel filtering. The machine used in the experiment is equipped with an Intel Core i7 2.80 GHz CPU. Speed is measured by the average time for point cloud filtering, in seconds per frame (point cloud), and the results are shown in Table 2. It can be seen that our method outperforms other filtering algorithms in terms of filtering efficiency; it is also nearly twice as fast as the best-performing method (voxel filtering). Although the statistical filtering performed best on the reconstructed standard deviation, it is also the slowest. One possible reason is that the statistical filtering method requires information about the nearest neighbors of each point and this search process is time-consuming.

Table 2. Comparison of the speed of different filter methods. Speed is measured in seconds-per-frame (point cloud).

Method	Time
Voxel filtering	0.088
Statistical filtering	1.77
Radius filtering	0.776
Fast two-stage filtering	0.046

We also tested the results of hole restoration and depth optimization combined with fast two-stage filtering, shown in Table 3, which can be used to evaluate the performance of our method for shape reconstruction. We can see that the reconstruction error is further reduced compared with the previous fast two-stage filtering, which shows that the depth optimization and hole restoration methods are effective for improving shape reconstruction. The reason for further improvement is that the method effectively responds to the noise and holes in the inner area of the face. Although we removed as much non-face noise as possible in the filtering stage, the depth data within the face still contain a part of random noise; the hole restoration and depth optimization methods are precisely aimed at solving this problem. The reconstructed error relative to the ground truth meshes in Table 3 shows that our method exhibits robust shape reconstruction performance that fully meets the 3D face reconstruction requirement in relevant fields.

Table 3. Reconstruction accuracy of mesh on synthetic data using point-to-plane distance with fast two-stage filtering combined with hole restoration and depth optimization (F + H). The table reports the mean error (Mean) and the standard deviation (Std).

Method	Mean	Std
F + H	1.26	0.125

Figure 5 shows the reconstruction capability of our method at different noise levels, and it can be seen that the total reconstruction error (both mean error and standard deviation) of the shape reconstruction of our method is always less than 1 mm within the actual noise range (2 mm) captured by the RealSense D435 camera.

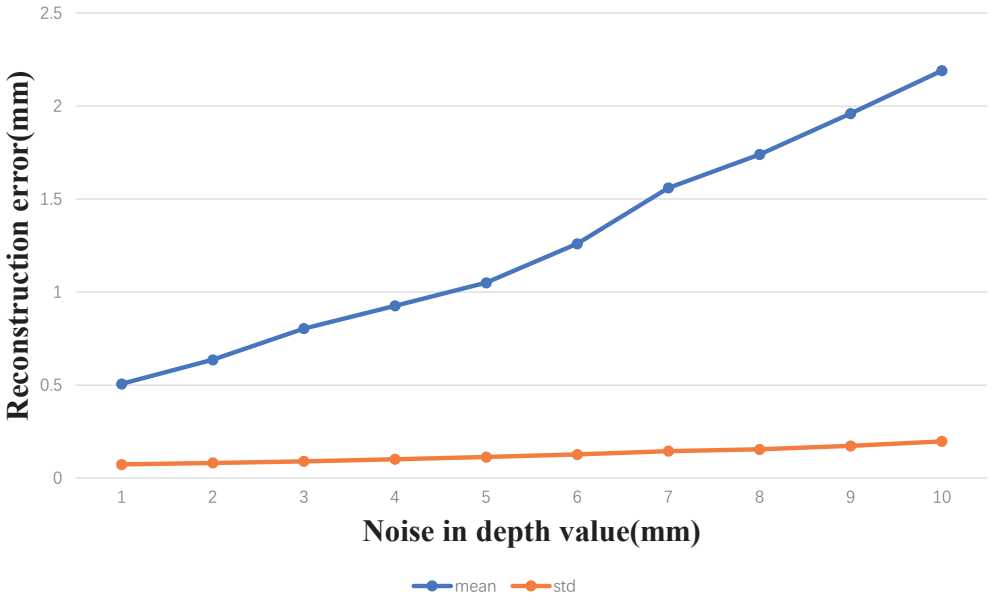

Figure 5. Reconstruction accuracy of our method on synthetic data with different noise levels.

As the noise increases, the reconstruction accuracy of our method decreases, but our method still maintains a high accuracy level.

4.2. Qualitative Evaluation

In this section, we qualitatively evaluate the methods above and the 3D face reconstruction capability of our system on the real dataset we captured.

Figure 6 shows a comparison of our point cloud filtering method with other filtering methods; in this experiment, we first performed the same depth filtering on all the method groups to filter out the approximate face results, and then conducted the comparison experiment. It is obvious from the comparison results that our filtering algorithm is excellent in removing edge noise. The statistical filtering excessively filters out the data that originally belonged to the face based on the core idea that the characteristics of apparent noise are sparsely distributed in space. The distribution of edge noise is not in isolated areas but closely attached to the edge area of the reconstructed subject, making the statistical filtering unable to distinguish the edge noise and remove it effectively. Radius filtering is better than the other two filtering algorithms, except for our method, in terms of results. However, radius filtering is also to determine whether the point is noise by its outlier; so, it still cannot effectively remove the edge noise attached to the face. Voxel filtering is used for noise removal by downsampling and is also ineffective for edge noise. Compared with the other methods, the point cloud after voxel filtering is the most sparse. Our fast two-stage filtering method does not consider the distribution characteristics of the edge noise but accurately segments the face region (including the edges) through human matting on the RGB image. It then combines the connection between the 2D RGB image and the 3D point cloud to make an accurate guide for point cloud filtering, thus achieving excellent performance.

Figure 7 shows a comparison of the results of our multi-view texture alignment method for the real image set, including the original data and the comparison results of the generated facial model before and after the method was adopted. For the left image set in Figure 7, the overall color of the face captured in the right view is darker and more reddish than the other views, and after the multi-view texture alignment, the right face is closer to the middle face than before. Even in an extreme case where the hues of the right image set in Figure 7 are completely inconsistent, our texture alignment method is still able to ensure uniform face color from all views. This is very important for texture fusion.

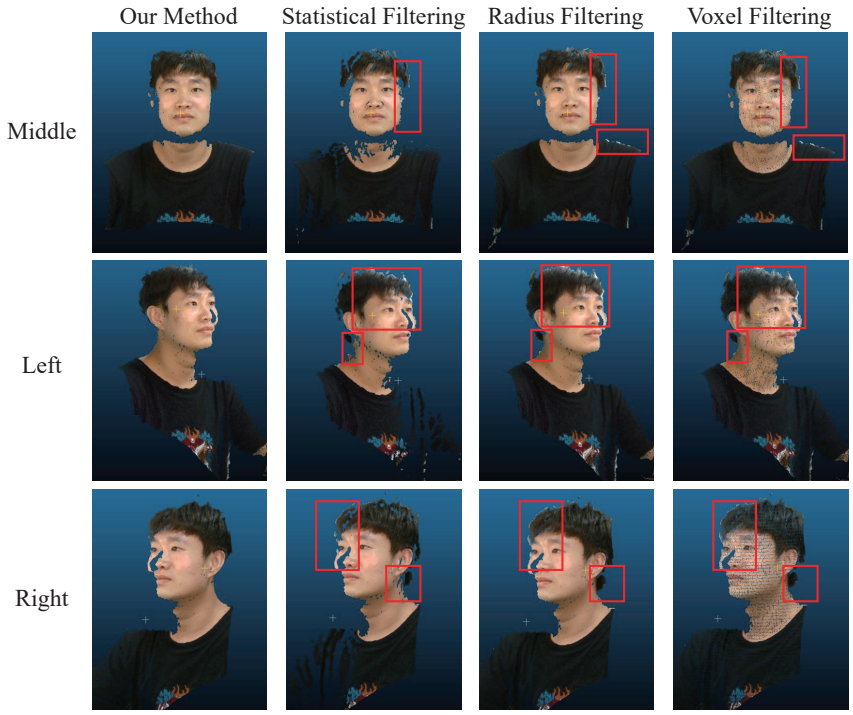

Figure 6. Comparison of point cloud filtering results under different camera views. The red box contains the noise that the other filtering method failed to filter out.

Figure 7. Multi-view texture alignment results.

Figure 8 shows the results of our face reconstruction on the real data acquired by RealSense D435, which shows that our method can accurately recover the 3D facial model of the acquisition users with excellent results for both geometry and texture.

The shape reconstruction details can be seen in Figure 9, from which we can see that we have clearly recovered the geometric shape of facial features even for slight differences in depth. Note that all of these results were achieved automatically without any user intervention, and our data acquisition was performed in a single moment, which makes for a good user experience and practical applications. In summary, our approach performs face reconstruction tasks with high standards for geometry and visual performance, and it can be extended to other tasks using arbitrary depth sensors for arbitrary subject reconstruction. At the same time, our algorithm maintains high efficiency to meet the requirements of practical applications while relying only on a CPU. The current system maintained an average time of 30 s on a machine equipped with an Intel Core i7 2.80 GHz CPU to complete the entire 3D face reconstruction process described above.

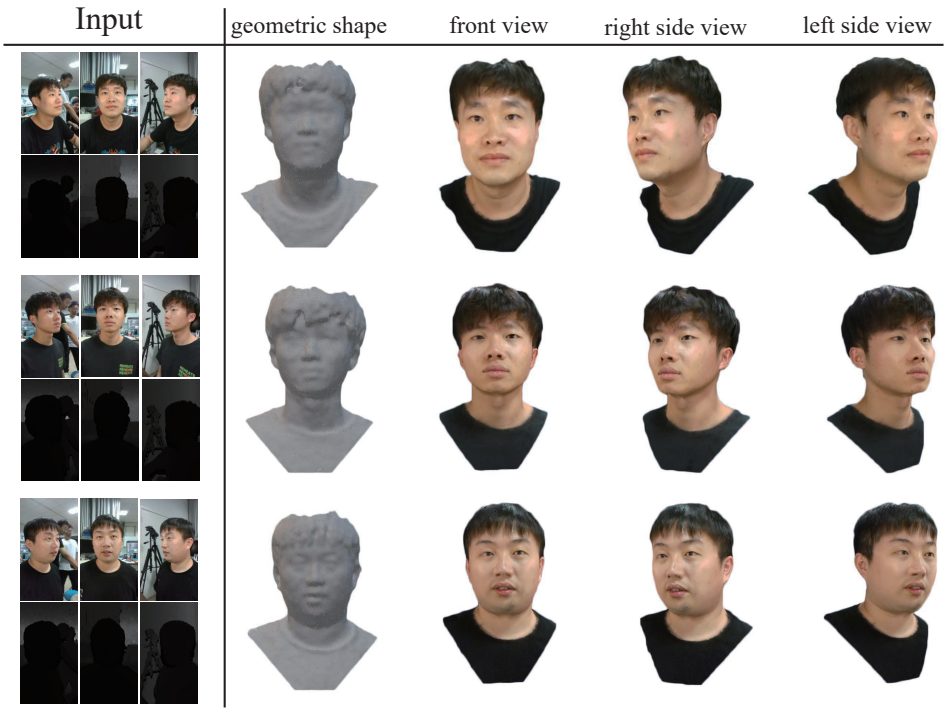

Figure 8. Face reconstruction results from aligned RGB-D image pairs using our approach. The first column contains the input RGB and depth image pairs, and our geometric shapes are shown in the second column. Columns 3–5 show our texture geometry results from different views.

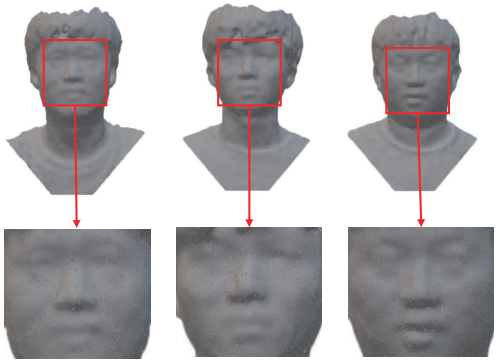

Figure 9. Geometric shape detail of facial reconstruction. We can see that we have clearly recovered the geometric shape of facial features even for small differences in depth.

5. Conclusions

In this paper, we present an algorithm for the accurate reconstruction of 3D faces using multiple depth cameras. Our method ensures the geometric accuracy and textural visualization of the reconstruction. The strength of our method is the use of the distance prior and matting technology to remove the non-subject noise from the point cloud, and the optimization and restoration of the depth data for the current reconstructed subject using redundant depth data in the spatiotemporal neighborhood. In addition, we also propose a new method of texture alignment through the idea of professional color correction to align the facial textures in different views globally. The experimental results show that our method has the ability to reconstruct face shapes stably and its recovered textures are highly uniform without the use of additional, prior, and strong 3D facial knowledge. Of course, the main limitation of our method is that it needs to combine multi-depth cameras to complete the data acquisition task synchronously. This also means that the data of our method are specific and no such public dataset exists on the web; so, we can not compare it

with the related methods we mentioned. However, this result is a compromise to ensure the integrity of the face and the most simplified data acquisition operations.

Author Contributions: Conceptualization, H.P. and L.Y.; methodology, H.P.; software, H.P.; validation, H.P.; formal analysis, H.P.; investigation, H.P.; resources, L.Y.; data curation, H.P.; writing—original draft preparation, H.P.; writing—review and editing, H.P. and L.Y.; visualization, H.P.; supervision, L.Y. and J.L.; project administration, L.Y.; funding acquisition, L.Y. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Foundation of Science and Technology on Near-Surface Detection Laboratory (grant number TCGZ2019A004) and Zhejiang Science and Technology Plan Project (grant number 2022C01204).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Acknowledgments: I would like to thank other academics, Duyang Wang and Bin Wu, for their valuable guidance throughout my studies.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zollhöfer, M.; Martinek, M.; Greiner, G.; Stamminger, M.; Süßmuth, J. Automatic reconstruction of personalized avatars from 3D face scans. *Comput. Animat. Virtual Worlds* **2011**, *22*, 195–202. [CrossRef]
2. Ferrari, C.; Berretti, S.; Pala, P.; Del Bimbo, A. 3D Face Reconstruction from RGB-D Data by Morphable Model to Point Cloud Dense Fitting. In Proceedings of the ICPRAM, Prague, Czech Republic, 19–21 February 2019; pp. 728–735.
3. Richardson, E.; Sela, M.; Or-El, R.; Kimmel, R. Learning detailed face reconstruction from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1259–1268.
4. Kemelmacher-Shlizerman, I.; Basri, R. 3D face reconstruction from a single image using a single reference face shape. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 394–405. [CrossRef] [PubMed]
5. Gecer, B.; Ploumpis, S.; Kotsia, I.; Zafeiriou, S. Fast-GANFIT: Generative adversarial network for high fidelity 3D face reconstruction. *arXiv* **2021**, arXiv:2105.07474.
6. Deli, R.; Galantucci, L.M.; Laino, A.; D'Alessio, R.; Di Gioia, E.; Savastano, C.; Lavecchia, F.; Percoco, G. Three-dimensional methodology for photogrammetric acquisition of the soft tissues of the face: A new clinical-instrumental protocol. *Prog. Orthod.* **2013**, *14*, 1–15. [CrossRef] [PubMed]
7. Hernandez, M.; Choi, J.; Medioni, G. Laser scan quality 3-d face modeling using a low-cost depth camera. In Proceedings of the 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012; pp. 1995–1999.
8. Blanz, V.; Vetter, T. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 1063–1074. [CrossRef]
9. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. Kinectfusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011; pp. 127–136.
10. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568.
11. Anasosalu, P.; Thomas, D.; Sugimoto, A. Compact and accurate 3-D face modeling using an RGB-D camera: Let's open the door to 3-D video conference. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013; pp. 67–74.
12. Zollhöfer, M.; Nießner, M.; Izadi, S.; Rehmann, C.; Zach, C.; Fisher, M.; Wu, C.; Fitzgibbon, A.; Loop, C.; Theobalt, C.; et al. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Trans. Graph. (Tog)* **2014**, *33*, 1–12. [CrossRef]
13. Hwang, J.; Yu, S.; Kim, J.; Lee, S. 3D face modeling using the multi-deformable method. *Sensors* **2012**, *12*, 12870–12889. [CrossRef] [PubMed]
14. Zhi, T.; Lassner, C.; Tung, T.; Stoll, C.; Narasimhan, S.G.; Vo, M. Texmesh: Reconstructing detailed human texture and geometry from rgb-d video. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 492–509.
15. Lee, Y.J.; Lee, S.J.; Park, K.R.; Jo, J.; Kim, J. Single view-based 3D face reconstruction robust to self-occlusion. *EURASIP J. Adv. Signal Process.* **2012**, *2012*, 176. [CrossRef]

16. Gecer, B.; Ploumpis, S.; Kotsia, I.; Zafeiriou, S. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1155–1164.
17. Tran, L.; Liu, X. Nonlinear 3d face morphable model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7346–7355.
18. Bao, L.; Lin, X.; Chen, Y.; Zhang, H.; Wang, S.; Zhe, X.; Kang, D.; Huang, H.; Jiang, X.; Wang, J.; et al. High-Fidelity 3D Digital Human Head Creation from RGB-D Selfies. *arXiv* **2020**, arXiv:2010.05562.
19. Zhu, X.; Yang, F.; Huang, D.; Yu, C.; Wang, H.; Guo, J.; Lei, Z.; Li, S.Z. Beyond 3dmm space: Towards fine-grained 3d face reconstruction. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 343–358.
20. Ren, H.; Zhang, X. *Efficient Facial Reconstruction and Real-time Expression for VR Interaction Using RGB-D Videos*; Virtual and Augmented Reality, Simulation and Serious Games for Education; Springer: Singapore, 2021; pp. 177–188.
21. Lin, S.; Yang, L.; Saleemi, I.; Sengupta, S. Robust High-Resolution Video Matting with Temporal Guidance. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 238–247.
22. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
23. Porter, T.; Duff, T. Compositing digital images. In Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, Minneapolis, MN, USA, 23–27 July 1984; pp. 253–259.
24. Bimber, O.; Emmerling, A.; Klemmer, T. Embedded entertainment with smart projectors. *Computer* **2005**, *38*, 48–55. [CrossRef]
25. Reinhard, E.; Adhikhmin, M.; Gooch, B.; Shirley, P. Color transfer between images. *IEEE Comput. Graph. Appl.* **2001**, *21*, 34–41. [CrossRef]
26. Horn, B.K. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A* **1987**, *4*, 629–642. [CrossRef]
27. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; JHU Press: Baltimore, MD, USA, 2013.
28. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *5*, 698–700. [CrossRef] [PubMed]
29. Huang, H.; Chai, J.; Tong, X.; Wu, H.T. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. In *ACM SIGGRAPH 2011 Papers*; Association for Computing Machinery: Vancouver, BC, Canada, 2011; pp. 1–10.
30. Genova, K.; Cole, F.; Maschinot, A.; Sarna, A.; Vlasic, D.; Freeman, W.T. Unsupervised training for 3d morphable model regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8377–8386.