

## Research Article

# Effective Key Region-Guided Face Detail Optimization Algorithm for 3D Face Reconstruction

Meihua Xiao,<sup>1</sup> Hanxiao Yi,<sup>1</sup> Ying Huang ,<sup>2</sup> Guoliang Luo ,<sup>3,4</sup> and Qian Xiao<sup>3</sup>

<sup>1</sup>Software Institute, East China Jiaotong University, Nanchang 330013, China

<sup>2</sup>Alibaba Business School, Hangzhou Normal University, Hangzhou 311121, China

<sup>3</sup>Virtual Reality and Interactive Techniques Institute, East China Jiaotong University, Nanchang 330013, China

<sup>4</sup>Jiangxi Booway New Technology Co., Ltd, Nanchang 330096, China

Correspondence should be addressed to Ying Huang; yw155@outlook.com and Guoliang Luo; luoguoliang@ecjtu.edu.cn

Received 27 August 2021; Accepted 27 December 2021; Published 27 January 2022

Academic Editor: Carlos Marques

Copyright © 2022 Meihua Xiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Low-cost and highly efficient 3D face reconstruction is a significant technique for developing virtual interaction applications. While the Kinect, a typical low-cost 3D data collection device, has been widely used in many applications, in this work, we propose a highly efficient 3D face point cloud optimization method for face reconstruction based on Kinect. Based on the different characteristics of the Kinect point cloud, diverse optimization strategies were delicately applied to improve the reconstruction quality. Our extensive experiments clearly show that our approach outperforms traditional approaches in terms of accuracy.

## 1. Introduction

The human face displays unique human biological characteristics and conveys a wide range of emotions. With the development of science and technology, the demand for digital human faces is no longer limited to two-dimensional images. Digital 3D face techniques are receiving increasing attention because they can convey more facial information, such as spatial shape, facial contour, and wrinkles. Many applications use 3D face technology to create vivid visual effects. For instance, virtual teleconference software uses reconstructed heads to compress the transmission bandwidth [1]. Security and electronic payment systems utilize facial depth characteristics to validate user identities [2]. In the medical field, surgical planning and analysis, as well as communication between doctors and patients, have become more efficient with the interactive assistance of virtual faces [3, 4]. 3D face techniques have incredible application prospects in our society.

The physiological structure and geometry of the face are very complex, so describing the face is a key problem. The classical model-based face construction method 3D Morph-

able Model (3DMM) [5], which uses a single RGB image, reconstructs a face model by solving the model parameters and building a linear combination of 3D faces. It can reconstruct a complete face as well as its topological structure, but it lacks deformation expression and is not flexible enough to represent high-frequency deformations, resulting in poor distinction between different human faces.

The depth information contains more details, which can enhance the realism of the face model. High-precision depth information is usually obtained by high-resolution lasers or structural light scanning equipment. For the high-precision instrument Go!SCAN, each scanned frontal face model has approximately 8,500 to 9,500 vertices. However, this equipment is often cumbersome and expensive, has strict usage conditions, and is unfriendly to measured objects. To address these problems, several low-cost RGB-D devices have been developed. For example, Microsoft Kinect can obtain colour and depth images in real time [6]. Due to its convenience and low cost, it has become the most popular 3D data collection device. Kinect HD Face [7] is a standard built-in API of Kinect SDK for face reconstruction. It can detect and track human faces at a rate of 30

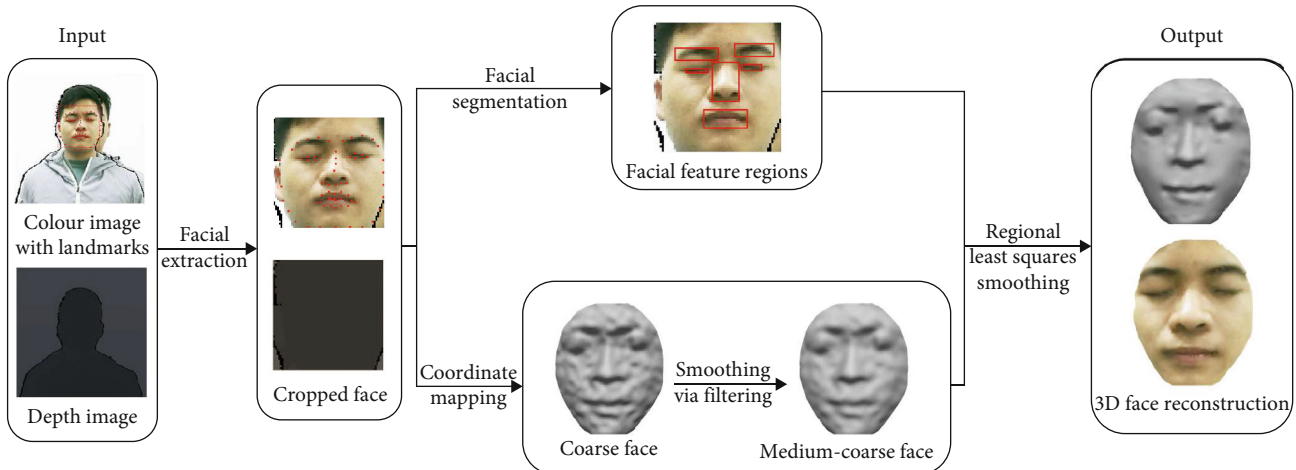


FIGURE 1: Overview of our 3D face reconstruction scheme.

frames per second, but it only generates 1,347 facial vertices. Too few point cloud vertices constrain the ability to describe the rich information of the human face. Its face tracking is based on body skeleton detection. If only the head, but no body, exists, there will be undetectable problems or data loss. In [8], Liang et al. used high-precision database matching to improve the quality of face models. However, due to the long time spent on nonrigid alignment and retrieval, the whole algorithm is very slow, requiring approximately 92.16 seconds per frame. In addition, texture information cannot be obtained.

In summary, model-based methods that use RGB information tend to produce results that are too smooth. The depth frame that is efficiently captured by the low-cost device Kinect contains more information. However, the low precision and weak anti-interference ability of Kinect does not meet the requirements of detailed facial expressions. The purpose of this paper is to generate high-quality face models efficiently and at a low cost while restoring as many different shapes and details of human faces as possible.

Our scheme is shown in Figure 1. With the acquired data from Kinect, we used face recognition and landmark detection to extract the face region and obtained 68 key facial feature landmarks. Then, we utilized the landmarks to distinguish a number of facial feature regions, which were then mapped to the 3D face point cloud. Based on the different characteristics of the region's point cloud, various optimization strategies were applied. The proposed algorithm allowed us to improve the quality of face reconstruction with a low-cost and highly efficient method.

The main contributions of our scheme can be summarized as follows:

- (i) We propose a face detail optimization method that is guided by key regions from a single RGB-D image to smooth faces while retaining more face details. In addition, the final face model includes texture information
- (ii) We develop high-precision face models by using optimized algorithms, overcoming the inevitable

low-quality problems caused by the low-cost Kinect equipment. Compared with existing methods, our method performs better in terms of both qualitative and quantitative results for seven expressions

The remainder of the paper is organized as follows. First, we summarize the related works in Section 2. Then, we present our 3D face reconstruction scheme in Section 3. After showing the experimental results in Section 4, the conclusions are presented in Section 5.

## 2. Related Works

In this section, we briefly summarize state-of-the-art 3D face reconstruction techniques and Microsoft (MS) Kinect-based research.

**2.1. 3D Face Reconstruction.** Currently, 3D face reconstruction methods can be classified as active or passive based on the information acquisition mode. For the active mode, depth data are extracted by dedicated scanning instruments, whereas for the passive mode, depth data are learned on 2D information such as videos and pictures.

Among the active methods, contact style and noncontact style are popular 3D scanning techniques. Contact instruments usually have high precision and do not have strict scanning condition requirements, such as illumination and object colour, but the scanning speed is slow and the usage is inconvenient. Noncontact instruments can actively emit infrared light to scan the environment. Noncontact instruments are popular due to their lightweight structure and fast scanning speed, but their accuracy is low. The scanning results are often affected by ambient conditions and frequently contain considerable noise. To improve accuracy, Hernandez et al. proposed a real-time 3D face reconstruction algorithm that used the first depth image as a reference frame and then used a GPU-based ICP algorithm to fuse the subsequent consecutive frames to the reference frame [9]. The algorithm can restore more facial details, but the texture mapping is unstable. Meyer and Do proposed a fast face segmentation method for separating the face from the depth

image, in which the segmented depth image was registered to a 3D model [10]. However, incomplete depth data may result in a partial 3D face, and the texture cannot be mapped. Point cloud data scanned or calculated by instruments often have problems such as noise and holes due to missing data [11, 12], which are not conducive to processing [13, 14]. Therefore, many scholars have studied the problem of hole repair in point clouds. Kumar et al. used the point cloud data around the hole to fit a nonuniform B-spline surface [15]. Leong et al. directly connected the boundary of the hole [16]. Since the boundary was randomly distributed, the result was not satisfactory. He et al. proposed a point cloud hole repair method based on a cubic B-spline surface [17] that performed better for large holes. However, the repair effect on the local area is greatly affected by noise.

For passive methods, face reconstructions are generated directly from 2D images [18–21] according to the shape-from-shading principle, which is an inverse photography process that uses boundary conditions, reflecting properties, and lighting and depth values to determine the 3D shape of the object. Boundary localization requires accurate two-dimensional feature point detection [22, 23]. To achieve this, Jiang et al. proposed a method for inferring geometric details from a single image [20]. Gecer et al. proposed a generative adversarial network model to fit feature points [24]. Hernandez et al. proposed a motion-based prior constrained structure [25]. Roth et al. proposed a reconstruction method using albedo information and personalized templates [26]. Another 3D face reconstruction trend is based on dense alignment. Fan et al. proposed an approach for recovering realistic facial textures that used facial feature points to adjust the overall and partial general frame models [21]. In [27], Liu et al. utilized multiangle 2D pictures with marked feature points to regress the 3D model. In [28], Feng et al. developed a UV position map to record the 3D shape of a complete face and trained a convolutional neural network to iteratively learn this UV map from a single 2D image. Liu et al. proposed learning 3D-to-2D mapping by computing the adjustment between the 3D face shape and the 2D landmarks [29, 30]. Tu et al. proposed a self-supervised learning method for overcoming noisy values that used sparse two-dimensional face landmarks as supplementary information [31].

Some methods combine the two types. For example, in [32], Zollhöfer et al. used colour images captured by Kinect to detect facial features and then used the features to adjust the general model and register it with the depth image. Macedo et al. improved the KinectFusion algorithm [33]. The algorithm uses head pose estimation to address occlusion problems, which provides the ICP algorithm with a better initial guess and allows it to reregister to obtain a complete face model.

**2.2. MS Kinect-Based Applications.** Microsoft Kinect can collect depth information from the environment and objects in a convenient and low-cost manner [6], supporting the expansion of applications such as postural control assessment [34], medical assistance [35–37], indoor mapping [38], and 3D scene reconstruction.

Newcombe et al. used a handheld Kinect to reconstruct room-sized scenes [39]. This was pioneering work for real-time dense 3D reconstruction using an RGB-D camera. Based on this work, many optimized reconstruction algorithms have been developed [40–42]. Zhang et al. proposed an integral imaging display based on KinectFusion that fuses multiframe depth information to eliminate the inherent noise in a single frame [43]. To restore and denoise depth maps, Esfahani and Pourreza proposed a method based on Gaussian filtering to repair small-scale noise and holes in depth images [44]. Lee et al. used asynchronous cellular automata and combined the depth and colour information to enhance the reconstruction accuracy [45].

Kinect has been used extensively for face recognition [46–50], emotion recognition [51, 52], wearable face recognition [53], facial recognition databases [54], and pose estimation [55–57]. Ren et al. applied the Kinect sensor for part-based hand gesture recognition, using a novel distance metric known as the Finger-Earth Mover's Distance to assess the dissimilarity of handshapes [58]. Saeed et al. proposed a head pose estimation method based on Kinect and a new depth-based feature descriptor that provided competitive estimation accuracy while requiring less computation time [59]. In [60], Southwell and Fang proposed a human object recognition approach using Kinect and a depth information mask matching model. Kinect has also been used to detect humans with a 3D human segmentation scheme [61, 62]. Siv et al. used Kinect to create 3D facial avatars [63], and Nose and Igarashi created talking avatars for contact on the Internet [64]. Bondi et al. used Kinect as a low-resolution scanner to reconstruct high-resolution face models by computing the 3D face registration among depth face sequences [65].

### 3. Proposed Method

Our algorithm consists of two modules: coarse face point cloud generation and optimization. In the first module, we use Kinect to capture depth and colour images of humans in real time and align them to the same size. Then, the face region and landmarks are extracted from the colour image with face detection and mapped to the depth image regions. Finally, we obtain a coarse 3D face model by transforming the depth map to a point cloud.

In the second part, we perform a key region-guided detail optimization algorithm. Specifically, for different face regions, we use various strategies based on the characteristics of the face subregion to overcome the problems of reserving face details and noise smoothing.

**3.1. Aligning Depth and Colour Images.** We used the Kinect for Windows V2 Sensor to capture the colour and depth images. Kinect V2 is a 3D somatosensory camera manufactured by Microsoft that includes a built-in colour camera, an infrared transmitter, and a depth (infrared) camera. The colour camera can capture colour image frames with a resolution of  $1920 * 1080$ . The infrared transmitter emits a near-infrared spectrum, which is reflected off the surface of the camera. The depth information can be calculated based on the flight time, and a depth image with a resolution of  $512 * 512$ .

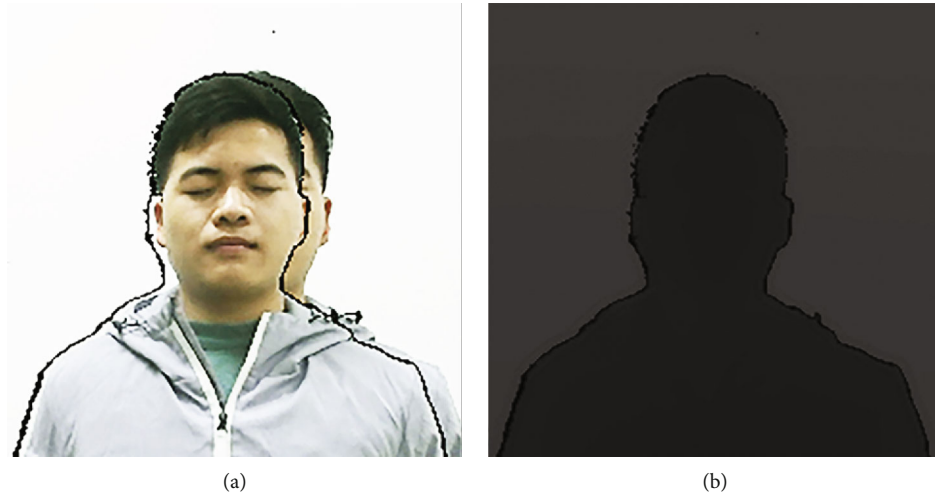


FIGURE 2: The aligned colour image (a) and the depth image (b). Some repeated pixels and shadows are generated due to the parallax of the depth camera and colour camera.

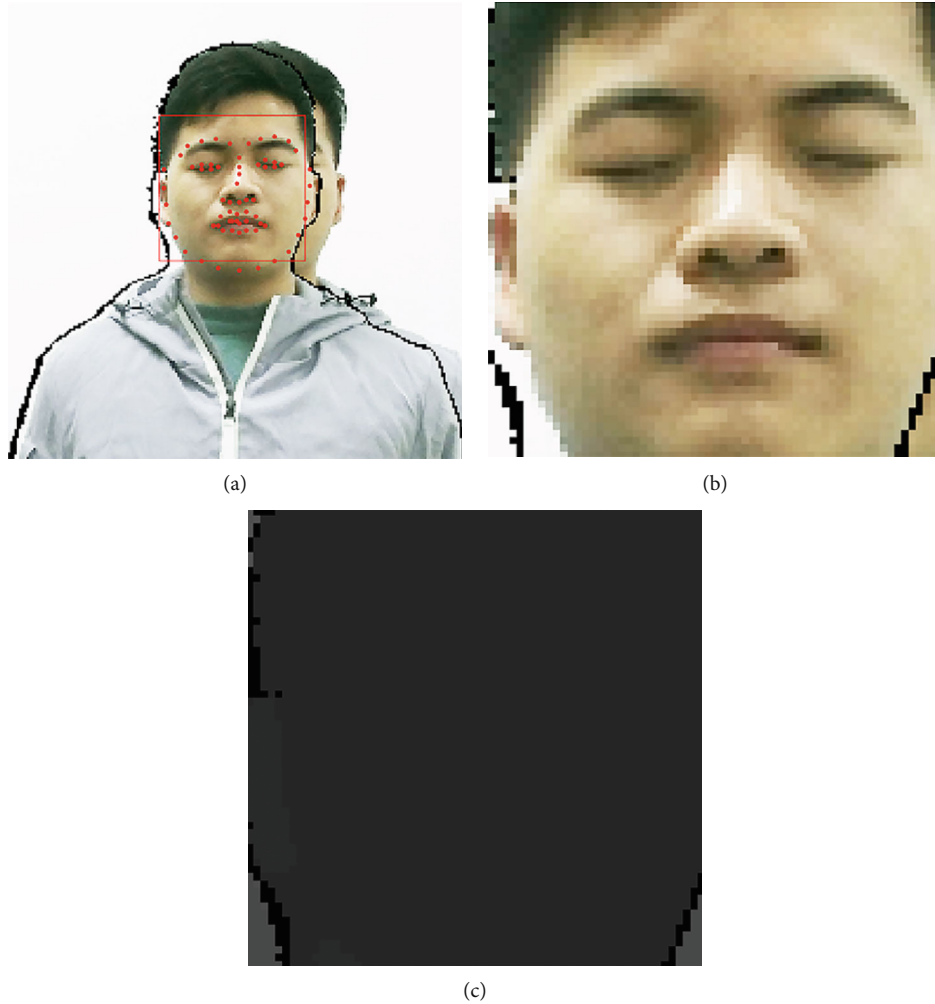


FIGURE 3: (a) Recognized face region and 68 face landmarks. (b) Colour image of a cropped face. (c) Depth image of a cropped face.

424 is generated. The depth and colour images are captured simultaneously. Since the depth and colour information should be aligned in the reconstruction, we use an efficient

built-in function of Kinect SDK to align the two kinds of images, obtaining images with the same resolution of  $512 * 424$ . Due to the parallax of the colour and depth cameras,

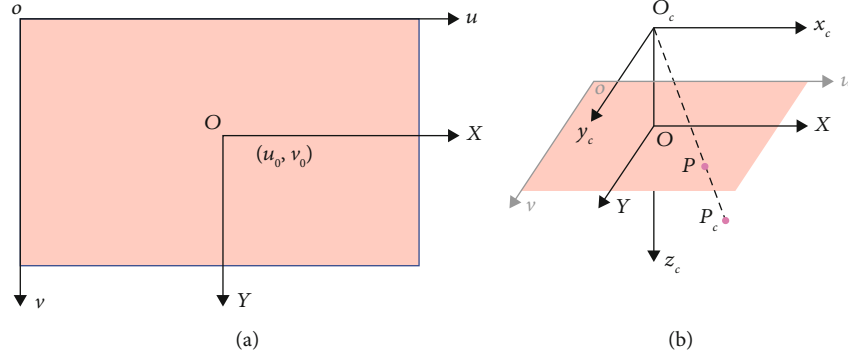


FIGURE 4: The conversion between the pixel coordinate system and the image coordinate system (a). The conversion between the camera coordinate system and the world coordinate system (b).

some pixels captured by one camera may not exist in the other, resulting in repeated pixels and shadows (see Figure 2).

**3.2. Face Region Extraction and Landmark Detection.** To calculate the face point cloud, we need to extract the face region from the image. Following previous works, we use the Dlib [66] face detection module to obtain the face region and the corresponding 68 face landmarks (see Figure 3(a)), which are used as indicators of key face regions. The face region is represented by a classical bounding box. Since we aligned the depth image and the colour image, the coordinates of the face region in the two images are the same. We can directly extract the face region from the colour image and the depth image.

**3.3. Face Point Cloud Generation.** Face point cloud generation is the process of converting the pixel coordinates of the face region in the depth image into world coordinates. This process includes converting among four coordinate systems: the pixel, image, camera, and world coordinate systems [34]. The units of the pixel coordinate axis and the image coordinate axis are pixels and millimetres (mm), respectively. The pixel coordinate system and the image coordinate system are on the same imaging plane. These two coordinate systems are translationally related (Figure 4(a)). This transformation can be expressed as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \quad (1)$$

where  $dx$  and  $dy$  are the lengths that pixels occupy in the  $X$  and  $Y$  directions, respectively, and  $(u_0, v_0)$  is the origin of the image.

The conversion of the image coordinate system to the camera coordinate system involves a perspective projection (Figure 4(b)), and it can be expressed as Equation (2).

$$z_c \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}, \quad (2)$$

where  $f$  is the focal length (the distance from the optical centre to the image plane),  $(x_c, y_c, z_c, 1)^T$  is the homogeneous coordinate of a point in the camera coordinate system, and  $(X, Y, 1)^T$  is the homogeneous coordinate of a point in the image coordinate system.

The transformation of the camera coordinate system to the world coordinate system can be expressed as follows:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad (3)$$

where  $\mathbf{R}$  and  $\mathbf{t}$  are the camera extrinsics that represent the rotation matrix and the translation matrix, respectively. In the experiment, the camera coordinate system coincides with the origin of the world coordinate system, so  $\mathbf{R}$  and  $\mathbf{t}$  are set to the following:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

$$\mathbf{t} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The camera coordinate system has the same depth as the world coordinate system. In summary, the conversion from the pixel coordinates to the world coordinates can be expressed as follows:



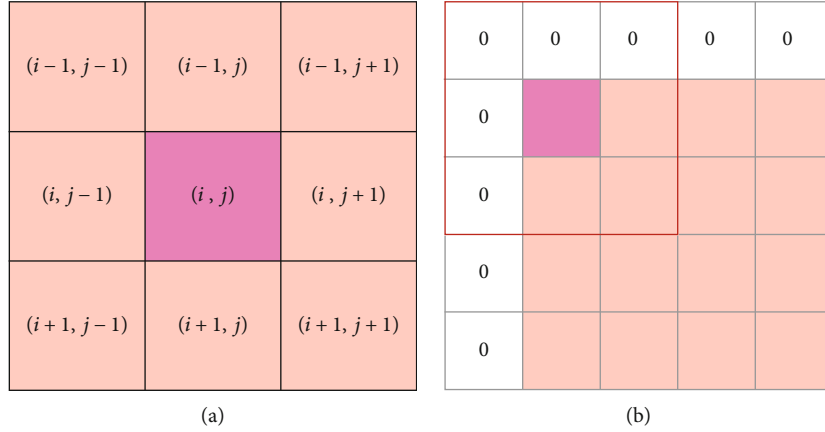


FIGURE 5: We use the mean smoothing method to calculate the smoothed value of each point (a). The depth values of the eight points around the centre pink point are averaged and used as the new value of the centre point. For boundary points, a zero-padding strategy is used to fill the template.

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{dx} & 0 & u_0 \\ 0 & \frac{f}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \quad (5)$$

Then,

$$\begin{cases} x_w = z_c \cdot (u - u_0) \cdot \frac{dx}{f}, \\ y_w = z_c \cdot (v - v_0) \cdot \frac{dy}{f}, \\ z_w = z_c. \end{cases} \quad (6)$$

After we calibrate the intrinsics of the depth camera, including the  $X$  focal length of the camera  $f_x = f/dx$ , the  $Y$  focal length of the camera  $f_y = f/dy$ , the principal point of the camera in the  $X$  dimension  $u_0$ , and the principal point of the camera in the  $Y$  dimension  $v_0$ , we can perform the coordinate conversion.

The final point cloud is an organized point cloud. An organized point cloud is a 2D array that stores vertices based on their spatial relationships. Adjacent points in the 2D array are also adjacent in three-dimensional space, allowing the index values to quickly locate adjacent points. Moreover, organized point clouds enable us to quickly map 2D points in the colour image to 3D points in the point cloud.

**3.4. Landmark-Guided Face Point Cloud Optimization.** Due to the limitations of environmental factors and Kinect sensor precision, the depth data may have holes formed by invalid points, which are represented as NaN values. We need to fill the holes to recover the face regions. In addition, we observe that point clouds often have rough burrs. For these two reasons, preliminary smoothing is performed. Because the organized point cloud is sequential, the smoothing traversal method is similar to traversing a two-dimensional array. We traverse the point cloud sequentially from the top left

corner to the bottom right corner, smoothing each point with a mean filter. Mean smoothing calculates the average depth values of the eight neighbouring points around the current point (see Figure 5) and uses the mean result as the smoothed value for this point. When an invalid point is encountered within these eight points, we ignore it and use the remaining points to compute the average. Hole filling is similar to mean smoothing. We replace the invalid point with the average of the surrounding eight points. For boundary points, we use a zero-padding strategy to fill the template. This operation yields a point cloud with no holes and few burrs.

Experiments show that the point cloud is still not smooth enough, even after mean smoothing. Thus, we use bilateral filter to smooth the point cloud. Bilateral filter is used frequently in depth image denoising. In [67], Li et al. used bilateral filter to denoise depth images and achieved good results. Bilateral filter is a nonlinear and noniterative extension of Gaussian filter [68, 69]. In Gaussian filter, weight is directly related to spatial distance. The closer the spatial distance is, the greater the weight is. Bilateral filter uses both spatial domain and range domain (greyscale) information, the points with a large greyscale difference are assigned with small weight, and the points with a small greyscale difference are assigned with large weight, allowing edge information to be preserved while smoothing. For both the spatial domain  $S$  and the range domain  $R$ , the kernel functions of bilateral filtering are as follows:

$$\begin{aligned} G_s &= \exp \left( -\frac{(x-u)^2 + (y-v)^2 + (z-w)^2}{2\sigma_s^2} \right), \\ G_r &= \exp \left( -\frac{\|\text{Gray}(x, y, z) - \text{Gray}(u, v, w)\|^2}{2\sigma_r^2} \right), \end{aligned} \quad (7)$$

where  $(x, y, z)$  is the centre point of the template, with a size of  $13 * 13$  and  $(u, v, w)$  represents the other points in the template. The gray value is converted from RGB values according to  $\text{Gray} = 0.299R + 0.587G + 0.114B$ .  $\sigma_s$  and  $\sigma_r$

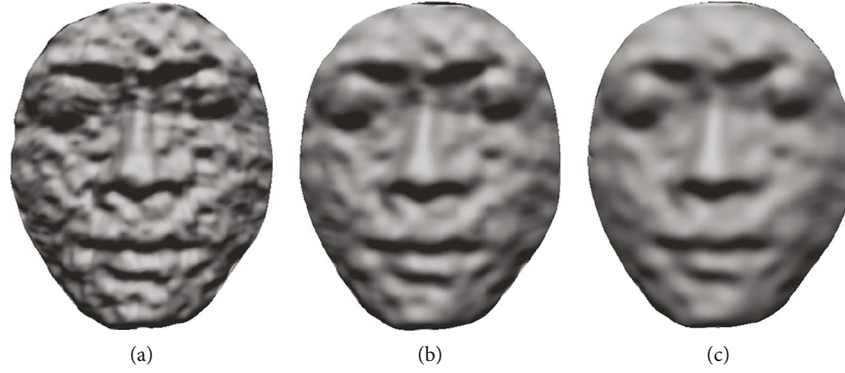


FIGURE 6: The original 3D face model (a). The 3D face model after applying the mean smoothing and hole filling operations (b). The 3D face model after applying bilateral smoothing (c).

are the standard deviation of the spatial domain and the range domain, respectively. The bilateral filter can be denoted as BF:

$$\text{BF}_{(x,y,z)} = \frac{1}{W_{(x,y,z)}} \sum_{p \in S} G_s * G_r * \text{Gray}_{(u,v,w)}, \quad (8)$$

where  $W_{(x,y,z)} = G_s * G_r$  is the weight sum of each point in the template used for the standard normalization operation. Figure 6 shows that the smoothing has noticeably improved. The face becomes smoother after bilateral smoothing, and the edge information is preserved as much as possible.

The point cloud after bilateral filtering is smooth in a small range, but it is still uneven as a whole. The least squares method can be used to fit the scattered points in a larger area than in the previous steps. We traversed the point cloud sequentially. Taking the current point as the centre, the surface was fitted with the least squares method in a range of  $13 * 13$ . Then, we calculated the value of the points on the smooth surface to replace the original value. The quadric surface equation to be fitted can be expressed as follows:

$$z = b_{20}x^2 + b_{11}xy + b_{02}y^2 + b_{10}x + b_{01}y + b_{00}, \quad (9)$$

where  $b$  represents the surface coefficient to be fitted. The total error between the surface and the original point can be expressed as follows:

$$Q = \sum_{i=1}^n [(b_{20}x_i^2 + b_{11}x_i y_i + b_{02}y_i^2 + b_{10}x_i + b_{01}y_i + b_{00}) - z_i]^2, \quad (10)$$

where  $z_i$  is the depth value of the  $i$ -th point. Fitting the surface minimizes the total error  $Q$ , which can be written in a matrix form as follows:

$$\min_{x \in R} (\|\mathbf{AB} - \mathbf{Z}\|_2)^2, \quad (11)$$



FIGURE 7: 3D face model after using least squares smoothing (without landmark guidance).

where

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{20} \\ b_{11} \\ \cdots \\ b_{02} \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} z_1 \\ z_2 \\ \cdots \\ z_n \end{bmatrix}, \quad (12)$$

where  $\mathbf{Z}$  is an  $n * 1$  matrix of depth values and  $n = 13 * 13 = 169$  indicates the number of points in the template.  $\mathbf{A}$  is an  $n * m$  matrix, where  $m = 6$  indicates the 6 bases constituted by  $x^2, xy, y^2, x, y$  and 1. The  $i$ -th row of  $\mathbf{A}$  can be calculated by the  $x$  and  $y$  values of the  $i$ -th point.

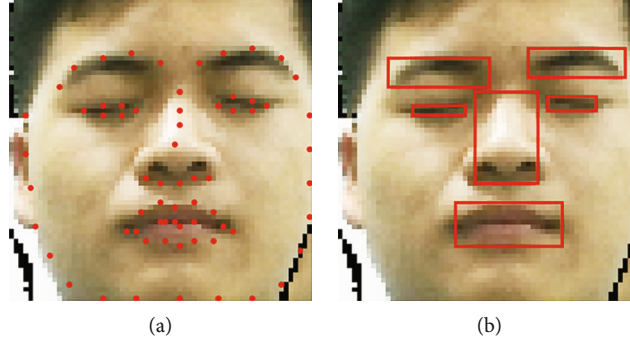


FIGURE 8: The results of landmarks in this experiment (a) and facial feature regions (b).

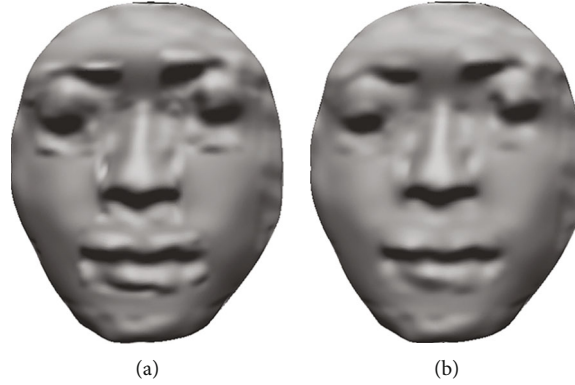


FIGURE 9: The 3D face model after applying regional least squares smoothing (a) and the final 3D face model (b).

$$\begin{aligned} (\|\mathbf{AB} - \mathbf{Z}\|_2)^2 &= (\mathbf{AB} - \mathbf{Z})^T (\mathbf{AB} - \mathbf{Z}) \\ &= \mathbf{B}^T \mathbf{A}^T \mathbf{AB} - \mathbf{Z}^T \mathbf{AB} - \mathbf{B}^T \mathbf{A}^T \mathbf{Z} + \mathbf{Z}^T \mathbf{Z}. \end{aligned} \quad (13)$$

Since  $\mathbf{Z}^T \mathbf{AB}$  and  $\mathbf{B}^T \mathbf{A}^T \mathbf{Z}$  can be transposed to each other and both results are scalars, noting that the transpose of a scalar is equal to itself, then

$$(\|\mathbf{AB} - \mathbf{Z}\|_2)^2 = \mathbf{B}^T \mathbf{A}^T \mathbf{AB} - 2\mathbf{B}^T \mathbf{A}^T \mathbf{Z} + \mathbf{Z}^T \mathbf{Z}. \quad (14)$$

The above equation can be solved when the partial derivative of  $\mathbf{B}$  is equal to 0:

$$\begin{aligned} \frac{\partial (\|\mathbf{AB} - \mathbf{Z}\|_2)^2}{\partial \mathbf{B}} &= \frac{\partial (\mathbf{B}^T \mathbf{A}^T \mathbf{AB} - 2\mathbf{B}^T \mathbf{A}^T \mathbf{Z} + \mathbf{Z}^T \mathbf{Z})}{\partial \mathbf{B}} \\ &= \frac{\partial (\mathbf{B}^T \mathbf{A}^T \mathbf{AB})}{\partial \mathbf{B}} - 2\mathbf{A}^T \mathbf{Z} = 2\mathbf{A}^T \mathbf{AB} - 2\mathbf{A}^T \mathbf{Z} = 0. \end{aligned} \quad (15)$$

The above equation can be simplified as follows:

$$\mathbf{B} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Z}. \quad (16)$$

The quadratic expression of the surface can be obtained by  $\mathbf{B}$ . Finally, the new depth value can be obtained by combining the  $X$  and  $Y$  coordinates of the point. The result is shown in Figure 7.

The face is overly smooth, and some important detail features are lost. Most of the human face is smooth,

except for key facial features such as the eyes and nose. Our goal was to use least squares smoothing to smooth only the areas without facial features while maintaining the key facial regions established in the previous step. Thus, we utilize the 68 previously obtained face landmarks to guide the smoothing procedure. Figure 8(a) shows the 68 landmarks marked with red dots on the colour image. Then, we divide the areas of the facial features into six key regions based on the landmarks. For example, for the mouth, the  $x$ -coordinates of the left and right corners of the mouth serve as the left and right boundaries of the region, while the  $y$ -coordinates of the upper lip and lower lip points serve as the upper and lower boundaries. The six obtained key regions are represented by bounding boxes in Figure 8(b). Since the points in the colour image and the depth image are aligned, we can use 2D landmark information to extract the depth information directly.

For the key facial regions, we perform least squares smoothing on the area outside the boxes while maintaining the values inside the boxes. This produces a smooth face model that does not blur the detailed facial features (Figure 9(a)). However, due to the different smoothing operations performed on the inside and outside of an area, an obvious abruptness appears on the edges of the area. Therefore, we continue to perform mean smoothing on the edge regions. Finally, we use two efficient methods (Delaunay [70] and Trimesh [71]) to calculate the triangular mesh and display the final face model (Figure 9(b)).



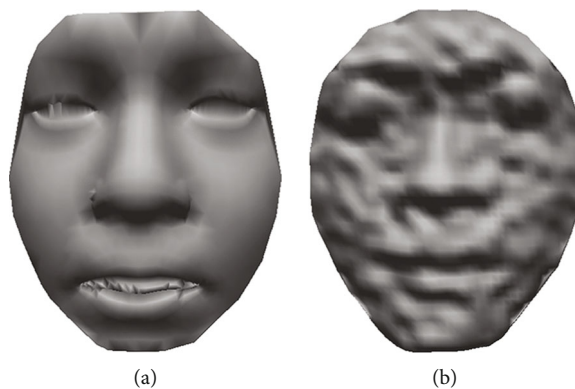


FIGURE 10: The results of Kinect HD Face (a) and GRIDFIT (b).

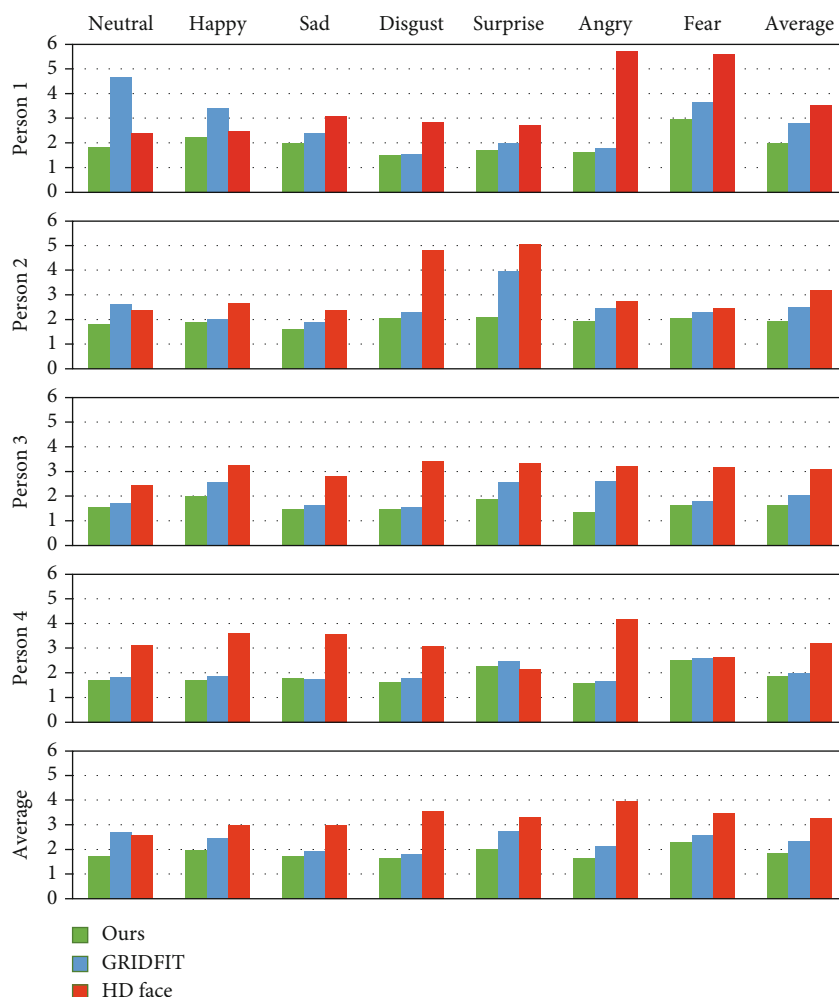


FIGURE 11: Histogram analysis of the reconstruction error for our method, the GRIDFIT method, and the HD Face method.

### 4. Experiment

In the experiments, the face scanned by Go!SCAN was regarded as the ground truth. The Hausdorff distance was used to measure the difference between the two meshes. MeshLab is a 3D geometric processing system that integrates the Hausdorff distance function [72]. Our implementation

was constructed in MATLAB and run on a PC with a 2.2 GHz Intel Core I7-8750h CPU and 16.0 GB RAM. In Section 3.4, both  $\sigma_s$  and  $\sigma_r$  were set to 1. We reconstructed 3D face models of four people with seven different expressions (neutral, happy, sad, disgust, surprise, angry, and fear). We compared our method with two popular approaches: Kinect HD Face and GRIDFIT.



FIGURE 12: The reconstruction results of four subjects for seven expressions. We show the ground truth (GT) as well as the results using our method, the GRIDFIT (GF) method, and the HD face (HD) method.

Kinect HD Face has two functions [7]. First, it can capture faces and display them in three dimensions. Second, it can track faces in real time and capture facial movements. In the first stage, Kinect uses 16 frames to fully capture our faces: four on the left, four on the right, four in the front, and four on the top of the face. In the second stage, the face model is created by deforming the mean face based on the captured face. Finally, the generated face mesh can be used in other animation applications. Moreover, no colour information is used in the final face model. We show the results in Figure 10(a).

In GRIDFIT, the surface is represented as a ductile film. The film is stretched only in the  $z$  direction of the real data point and is fixed in all other directions. In contrast to interpolation methods, it does not predict the data of the point but instead pushes the obtained surface as close as possible to the data point to obtain an approximate solution [73]. There are some lattices that cover all the points, forming a matrix unit. All points fall into one of the matrix units. The fitting problem can be written as  $PX = Y$ . We define

the number of points in the  $x$  direction as  $nx$  and the number of points in the  $y$  direction as  $ny$ , so the length of  $X$  is  $nx * ny$ . Thus,  $P$  has  $n$  rows and  $nx * ny$  columns. Because this is often an ill-posed problem in practice, we added a regularization term. Then, the goal of the objective function is to minimize  $\|PX - Y\|^2 + \lambda\|RX\|^2$ . GRIDFIT allows the user to control the stiffness through  $\lambda$  [73]. In comparative experiments,  $\lambda$  is selected as 1. We show the results in Figure 10(b). There were still many bulges on the face, and the facial features were not clear.

Figure 11 shows the 3D face reconstruction results of four people using our methods: the GRIDFIT method and the Kinect HD Face method. We also compare the results with the ground truth faces (Go!SCAN results). The mean and standard variation of the RMS are  $1.85 \pm 0.33$  for our method,  $2.33 \pm 0.76$  for the GRIDFIT method, and  $3.26 \pm 0.96$  for the HD Face method. The figure shows that our results always had lower errors than the GRIDFIT and HD Face methods. The results show that our method can express human faces more accurately. We preserved as many face details as possible while ensuring smoothness. This is verified in Figure 12. In terms of speed, our method requires approximately 1.31 seconds to load the face landmark detection model in the first frame. After the first frame, our method takes approximately 0.23 seconds per frame. The GRIDFIT method takes 0.63 seconds per frame on average. The Kinect HD Face method is divided into two stages. The first stage is human detection, which detects the body skeleton and the dynamic face. This stage takes approximately 3.67 seconds. The second stage is the face tracking stage, which takes approximately 0.23 seconds per frame. We calculated the time consumption of the three models for ten frames. Our method, the GRIDFIT method, and the Kinect HD Face method take approximately 3.61 seconds, 6.30 seconds, and 5.97 seconds for ten frames, respectively. Our method not only produces higher-quality models but also takes less time.

## 5. Conclusion

In this paper, we propose a key region-guided detail optimization approach for 3D face reconstruction from a single RGB-D image. We extract the face region from an image frame and use face landmarks to guide the optimization of the face point cloud. For different face regions, we design various strategies for smoothing noise and holes while retaining necessary face details, resulting in a more realistic face model. When comparing our approach with existing popular methods on a real face reconstruction dataset, our algorithm performs better than the other methods, with a clear advantage in both quantitative and qualitative terms. The Kinect HD Face method performed poorly on disgust, surprise, anger, and fear expressions, while our method was consistent across the seven expressions. Especially in the eyebrow and mouth regions, the Kinect HD method had difficulty expressing the real emotion. The speed comparisons also show that our method is effective for high-quality 3D face generation.

One limitation of our method is that the effective depth range of Kinect is approximately 0.5~4.5 m, and the measurement data beyond this range have a large error. This leads to a high modelling error when there is a large distance between the subjects and the Kinect. In addition, in this study, we used Kinect to obtain RGB-D images quickly and easily. Many mobile phones, such as the iPhone X, now include RGB-D cameras, which makes it possible to extend our method to mobile terminals. In future work, we plan to adapt the method for mobile phones to apply the method to a broader range of scenarios.

## Data Availability

The data used to support the findings of this study are currently under embargo. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work has been in part supported by the National Natural Science Foundation of China (Nos. 61962021, 61602222, 61972344, and 61732015), the Natural Science Foundation of Jiangxi, China (20202ACBL202008), the Postgraduate Course (Virtual Reality Technology) Construction Project of Jiangxi, China, the Higher Education Reform Research Provincial Project of Jiangxi, China (JXJG-20-5-20), and the Technology Project of Jiangxi Education Department, China (GJJ210623).

## References

- [1] J. Thies, M. Zollhöfer, M. Stamminger, M. Nießner, and M. Nießner, "FaceVR: real-time gaze-aware facial reenactment in virtual reality," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 2, pp. 1–15, 2018.
- [2] C. W. Wang and C. C. Peng, "3D face point cloud reconstruction and recognition using depth sensor," *Sensors*, vol. 21, no. 8, p. 2587, 2021.
- [3] S. Rasteau, A. Louvrier, P. Bouletreau, and N. Sigaux, "Three-dimensional acquisition technologies for facial soft tissues - applications and prospects in orthognathic surgery," *Journal of Stomatology, Oral and Maxillofacial Surgery*, vol. 121, no. 6, pp. 721–728, 2020.
- [4] Z. Fishman, J. Liu, J. Pope, J. A. Fialkov, and C. M. Whyne, "Validating 3D face morphing towards improving pre-operative planning in facial reconstruction surgery," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 9, no. 5, pp. 480–487, 2021.
- [5] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 187–194, New York, USA, 1999.
- [6] Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, 2012.
- [7] "Microsoft, Kinect for Windows: human interface guidelines v1.8," 2013, <http://go.microsoft.com/fwlink/?LinkID=247735>.
- [8] S. Liang, I. Kemelmacher-Shlizerman, and L. G. Shapiro, "3D face hallucination from a single depth frame," in *2nd International Conference on 3D Vision*, vol. 1, pp. 31–38, Tokyo, Japan, 2014.
- [9] M. Hernandez, J. Choi, and G. Medioni, "Laser scan quality 3-D face modeling using a low-cost depth camera," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pp. 1995–1999, Bucharest, Romania, 2012.
- [10] G. P. Meyer and M. N. Do, "Real-time 3D face modeling with a commodity depth camera," in *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 1–4, San Jose, California, USA, 2013.
- [11] M. Vieira and K. Shimada, "Surface mesh segmentation and smooth surface extraction through region growing," *Computer Aided Geometric Design*, vol. 22, no. 8, pp. 771–792, 2005.
- [12] S. Gai, F. Da, L. Zeng, and Y. Huang, "Research on a hole filling algorithm of a point cloud based on structure from motion," *Journal of the Optical Society of America A*, vol. 36, no. 2, pp. A39–A46, 2019.
- [13] W. Y. Zhong, "Direct integration of reverse engineering and rapid prototyping based on the properties of NURBS or B-spline," *Precision Engineering*, vol. 28, no. 3, pp. 293–301, 2004.
- [14] E. Bagci, "Reverse engineering applications for recovery of broken or worn parts and re-manufacturing: three case studies," *Advances in Engineering Software*, vol. 40, no. 6, pp. 407–418, 2009.
- [15] A. Kumar, A. Shih, Y. Ito, D. Ross, and B. Soni, "A hole-filling algorithm using non-uniform rational b-splines," in *Proceedings of the 16th International Meshing Roundtable*, pp. 169–182, Washington, USA, 2008.
- [16] K. F. Leong, C. K. Chua, and Y. M. Ng, "A study of stereolithography file errors and repair. Part 2. Special cases," *The International Journal of Advanced Manufacturing Technology*, vol. 12, no. 6, pp. 415–422, 1996.
- [17] D. J. He, J. Y. Niu, and Z. R. Zhang, "Repairing method of missing area of dairy cows' point cloud based on improved cubic B-spline curve," *Transaction of the Chinese Society for Agricultural Machinery*, vol. 49, no. 6, pp. 225–231, 2018.
- [18] I. Kemelmacher-Shlizerman and R. Basri, "3D face reconstruction from a single image using a single reference face shape," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 394–405, 2011.
- [19] M. Zollhöfer, J. Thies, P. Garrido et al., "State of the art on monocular 3D face reconstruction, tracking, and applications," *Computer Graphics Forum*, vol. 37, no. 2, pp. 523–550, 2018.
- [20] L. Jiang, J. Zhang, B. Deng, H. Li, and L. Liu, "3D face reconstruction with geometry details from a single image," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4756–4770, 2018.
- [21] X. Fan, P. Qiang, and Z. Ming, "3D face reconstruction from single 2D image based on robust facial feature points extraction and generic wire frame model," *International Conference on Communications & Mobile Computing*, vol. 3, pp. 394–400, 2010.
- [22] X. Jin and X. Tan, "Face alignment in-the-wild: a survey," *Computer Vision and Image Understanding*, vol. 162, pp. 1–22, 2017.

- [23] N. Wang, X. Gao, D. Tao, H. Yang, and X. Li, "Facial feature point detection: a comprehensive survey," *Neurocomputing*, vol. 275, pp. 50–65, 2018.
- [24] B. Gecer, S. Ploumpis, I. Kotsia, and S. Zafeiriou, "Ganfit: generative adversarial net-work fitting for high fidelity 3D face reconstruction," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1155–1164, Long Beach, CA, USA, 2019.
- [25] M. Hernandez, T. Hassner, J. Choi, and G. Medioni, "Accurate 3D face reconstruction via prior constrained structure from motion," *Computers & Graphics*, vol. 66, pp. 14–22, 2017.
- [26] J. Roth, Y. Tong, and X. Liu, "Adaptive 3D face reconstruction from unconstrained photo collections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4197–4206, Las Vegas, NV, USA, 2016.
- [27] F. Liu, D. Zeng, J. Li, and Q. J. Zhao, "On 3D face reconstruction via cascaded regression in shape space," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 12, pp. 1978–1990, 2017.
- [28] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3D face reconstruction and dense alignment with position map regression network," in *Proceedings of the European Conference on Computer Vision*, pp. 557–574, Munich, Germany, 2018.
- [29] F. Liu, Q. Zhao, X. Liu, and D. Zeng, "Joint face alignment and 3D face reconstruction with application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, pp. 664–678, 2020.
- [30] F. Liu, D. Zeng, Q. Zhao, and X. Liu, "Joint face alignment and 3D face reconstruction," in *European Conference on Computer Vision*, pp. 545–560, Amsterdam, The Netherlands, 2016.
- [31] X. Tu, J. Zhao, M. Xie et al., "3D face reconstruction from a single image assisted by 2D face images in the wild," *IEEE Transactions on Multimedia*, vol. 23, pp. 1160–1172, 2019.
- [32] M. Zollhöfer, M. Martinek, G. Greiner, M. Stamminger, and J. Süßmuth, "Automatic reconstruction of personalized avatars from 3D face scans," *Computer Animation and Virtual Worlds*, vol. 22, no. 2-3, pp. 195–202, 2011.
- [33] M. C. de Farias Macedo, A. L. Apolinário, and A. C. dos Santos Souza, "A robust real-time face tracking using head pose estimation for a markerless AR system," in *2013 XV Symposium on Virtual and Augmented Reality*, pp. 224–227, Cuiaba, Mato Grosso, Brazil, 2013.
- [34] R. A. Clark, Y.-H. Pua, K. Fortin et al., "Validity of the Microsoft Kinect for assessment of postural control," *Gait & Posture*, vol. 36, no. 3, pp. 372–377, 2012.
- [35] B. Galna, G. Barry, D. Jackson, D. Mhiripiri, P. Olivier, and L. Rochester, "Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease," *Gait & Posture*, vol. 39, no. 4, pp. 1062–1068, 2014.
- [36] Y. Yuminaka, M. Fujii, A. Manabe, M. Hasegawa, and N. Wada, "Rehabilitation assistance using motion capture devices and virtual reality feedback," *Applied Mechanics and Materials*, vol. 888, pp. 37–42, 2019.
- [37] M. Ababneh, H. Sha'ban, D. AlShalabe, D. Khader, H. Mahameed, and M. AlQudimat, "Gesture controlled mobile robotic arm for elderly and wheelchair people assistance using Kinect sensor," in *2018 15th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 636–641, Hammamet, Tunisia, 2018.
- [38] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of Kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.
- [39] R. A. Newcombe, S. Izadi, O. Hilliges et al., "KinectFusion: real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136, 2011.
- [40] T. Whelan, H. M. Kaess, M. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 598–626, 2015.
- [41] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.
- [42] S. Choi, Q. Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5556–5565, Boston, MA, USA, 2015.
- [43] J. Zhang, X. Wang, Q. Zhang, Y. Chen, J. du, and Y. Liu, "Integral imaging display for natural scene based on KinectFusion," *Optik-International Journal for Light and Electron Optics*, vol. 127, no. 2, pp. 791–794, 2016.
- [44] M. A. Esfahani and H. Pourreza, "Kinect depth recovery based on local filters and plane primitives," *Integral Methods in Science and Engineering*, vol. 2, pp. 53–63, 2017.
- [45] K. Lee, Y. Ban, and S. Lee, "Efficient depth enhancement using a combination of color and depth information," *Sensors*, vol. 17, no. 7, p. 1544, 2017.
- [46] B. Y. Li, A. S. Mian, W. Liu, and A. Krishna, "Face recognition based on Kinect," *Pattern Analysis and Applications*, vol. 19, no. 4, pp. 977–987, 2016.
- [47] B. Y. Li, M. Xue, A. Mian, W. Liu, and A. Krishna, "Robust RGB-D face recognition using Kinect sensor," *Neurocomputing*, vol. 214, pp. 93–108, 2016.
- [48] M. Hayat, M. Bennamoun, and A. A. El-Sallam, "An RGB-D based image set classification for robust face recognition from Kinect data," *Neurocomputing*, vol. 171, pp. 889–900, 2016.
- [49] B. Y. L. Li, A. S. Mian, W. Liu, and A. Krishna, "Using Kinect for face recognition under varying poses, expressions, illumination and disguise," in *2013 IEEE Workshop on Applications of Computer Vision*, pp. 186–192, Clearwater Beach, FL, USA, 2013.
- [50] F. T. Zohra and M. L. Gavrilova, "KINECT face recognition using occluded area localization method," *Transactions on Computational Science XXX*, vol. 10560, pp. 12–28, 2017.
- [51] T. Mallick, P. Goyal, P. P. Das, and A. K. Majumdar, "Facial emotion recognition from Kinect data-an appraisal of Kinect face tracking library," in *VISIGRAPP*, pp. 525–532, Rome, Italy, 2016.
- [52] Q. R. Mao, X. Y. Pan, X. J. Shen, and Y. Z. Zhan, "Using Kinect for real-time emotion recognition via facial expressions," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, no. 4, pp. 272–282, 2015.
- [53] L. B. Neto, F. Grijalva, V. R. M. L. Maibe et al., "A Kinect-based wearable face recognition system to aid visually impaired users," *IEEE Transactions on Human-Machine Systems*, vol. 47, pp. 1–13, 2016.
- [54] R. Min, N. Kose, and J.-L. Dugelay, "KinectFaceDB: a Kinect database for face recognition," *IEEE Transactions on Systems*,



- Man, and Cybernetics: Systems*, vol. 44, no. 11, pp. 1534–1548, 2014.
- [55] J. Shotton, A. Fitzgibbon, M. Cook et al., “Real-time human pose recognition in parts from single depth images,” *Machine Learning for Computer Vision*, vol. 411, 2013.
- [56] J. C. Núñez, R. Cabido, A. S. Montemayor, and J. J. Pantrigo, “Real-time human body tracking based on data fusion from multiple RGB-D sensors,” *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4249–4271, 2017.
- [57] A. Martínez-González, M. Villamizar, O. Canévet, and J. Odobez, “Real-time convolutional networks for depth-based human pose estimation,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 41–47, Madrid, Spain, 2018.
- [58] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, “Robust part-based hand gesture recognition using Kinect sensor,” *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.
- [59] A. Saeed, A. Al-Hamadi, and A. Ghoneim, “Head pose estimation on top of Haar-like face detection: a study using the Kinect sensor,” *Sensors*, vol. 15, no. 9, pp. 20945–20966, 2015.
- [60] B. J. Southwell and G. Fang, “Human object recognition using colour and depth information from an RGB-D Kinect sensor,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 3, pp. 171–178, 2013.
- [61] L. Xia, C.-C. Chen, and J. K. Aggarwal, “Human detection using depth information by Kinect,” in *CVPR 2011 Workshops*, pp. 15–22, Colorado Springs, CO, USA, 2011.
- [62] V. Gulshan, V. Lempitsky, and A. Zisserman, “Humanising grabcut: Learning to segment humans using the kinect,” in *2011 IEEE International conference on computer vision workshops (ICCV workshops)*, pp. 1127–1133, Barcelona, Spain, November 2011.
- [63] R. Siv, I. Ardiyanto, and R. Hartanto, “3D human face reconstruction using depth sensor of Kinect 2,” in *2018 International Conference on Information and Communications Technology (ICOIACT)*, pp. 355–359, Yogyakarta, Indonesia, 2018.
- [64] T. Nose and Y. Igarashi, “Real-time talking avatar on the Internet using Kinect and voice conversion,” *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 12, 2015.
- [65] E. Bondi, P. Pala, S. Berretti, and A. D. Bimbo, “Reconstructing high-resolution face models from Kinect depth sequences,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2843–2853, 2016.
- [66] D. E. King, “Dlib-ml: a machine learning toolkit,” *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
- [67] H. Li, D. Kong, S. Wang, and B. Yin, “Hand depth image denoising and superresolution via noise-aware dictionaries,” *Journal of Electrical and Computer Engineering*, vol. 2016, Article ID 6587162, 12 pages, 2016.
- [68] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth International Conference on Computer Vision*, pp. 839–846, Bombay, India, 1998.
- [69] S. Paris and F. Durand, “A fast approximation of the bilateral filter using a signal processing approach,” in *European Conference on Computer Vision*, pp. 568–580, Graz, Austria, 2006.
- [70] “Delaunay: Delaunay triangulation,” <https://ww2.mathworks.cn/help/matlab/ref/delaunay.html?lang=en>.
- [71] “Trimesh: triangular mesh plot,” <https://ww2.mathworks.cn/help/matlab/ref/trimesh.html?lang=en>.
- [72] P. Cignoni, C. Rocchini, and R. Scopigno, “Metro: measuring error on simplified surfaces,” *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998.
- [73] J. D’Errico, “Surface fitting using gridfit,” *MATLAB Central File Exchange*, vol. 8998, 2010.