

Article

A New Filtering System for Using a Consumer Depth Camera at Close Range

Yuanxing Dai ^{1,*}, Yanming Fu ², Baichun Li ³, Xuwei Zhang ¹, Tianbiao Yu ¹ and Wanshan Wang ¹¹ School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110004, China² Laboratory Management Center, Shenyang Sport University, Shenyang 110102, China³ College of Aeronautical Engineering, Civil Aviation University of China, Tianjin 300000, China

* Correspondence: yuanxing_dai@163.com; Tel.: +86-1662-088-7776

Received: 16 June 2019; Accepted: 1 August 2019; Published: 8 August 2019



Abstract: Using consumer depth cameras at close range yields a higher surface resolution of the object, but this makes more serious noises. This form of noise tends to be located at or on the edge of the realistic surface over a large area, which is an obstacle for real-time applications that do not rely on point cloud post-processing. In order to fill this gap, by analyzing the noise region based on position and shape, we proposed a composite filtering system for using consumer depth cameras at close range. The system consists of three main modules that are used to eliminate different types of noise areas. Taking the human hand depth image as an example, the proposed filtering system can eliminate most of the noise areas. All algorithms in the system are not based on window smoothing and are accelerated by the GPU. By using Kinect v2 and SR300, a large number of contrast experiments show that the system can get good results and has extremely high real-time performance, which can be used as a pre-step for real-time human-computer interaction, real-time 3D reconstruction, and further filtering.

Keywords: depth image filtering; point clouds filtering; Kinect v2; depth resolution; close range; hand pose

1. Introduction

The reasons for the success of consumer depth cameras are low price, acceptable accuracy, lower learning costs, extensive applicability, and excellent portability. It has been applied in the fields such as body and facial recognition, 3D motion capture, and has been developing very rapidly.

Most of the depth cameras are based on time-of-flight principle, such as Kinect v2 and SR300. It can collect the laser spots array reflected by surfaces, and works out the time difference between emission and reflection to get the distance array of the scene [1,2]. Generally, the array is expressed as a gray image, and the gray value of the pixel is generated by the depth value of the position according to certain rules.

According to the principle of perspective, within the unit area, the closer the surface is to the camera, the more laser points will be reflected, which means that higher measurement point density will result in higher surface resolution. Although the range could be changed by using Draelos's method [3], according to our observation, when a target surface gets close to the nearest limit, for example, 3D reconstruction of small objects [1], in the point cloud acquired by consumer depth cameras, there will be some irregular shape of noise areas surrounding or on the edge of the realistic surface (a surface consisting of laser points reflected by a real object, it is used to distinguish unrealistic surfaces formed by noise points that should not exist).

In the process of generating point clouds with depth cameras, we found that the closer the distance is, the more significant the phenomenon is. Figure 1 shows this phenomenon by using the depth

images of a human hand at different distances. In static applications, these low confidence noise areas could be filtered effectively in a post-processing stage [4]. However, any time consumption is undesirable for real-time interactive applications [5–7].

Using a consumer depth camera at close range is a double-edged sword. Developers often aim to use depth images or video stream for further development, such as reverse engineering [8], human pose recognition [9–12], and 3D scene reconstruction [13]. In order to obtain a pure depth image with high accuracy and low noise, one option is to select expensive, high learning cost, and precise optical equipment (3D time-of-flight (ToF) camera or LIDAR). However, for time and money savings, an easier way is to place the object closer to get a higher resolution on the surface of the object. In this case, how to eliminate the noise deterioration caused by close-range use has become a problem that must be solved.

Traditional methods for reducing or eliminating color image noise are usually based on window smoothing or sharpening, such as median filtering [14], non-local means filter [15], bilateral filtering [14, 16], joint bilateral filtering [17], etc. The principle is to make a window for each pixel in the image, and update the center pixel according to the value of other pixels in the window.

Different filters use different window selection methods and updated strategies [18]. However, the unmodified algorithm transplantation is not very suitable for depth image filtering. For edge noise of a depth image, joint bilateral filtering with reliable sources (usually from color images) can perform very well. It can better preserve the edge details of an object [19], but for human hand depth image filtering it also involves the lighting conditions [20–22] and the color difference of the foreground and background [23,24].

To fill this gap, in this article we proposed a composite filtering system for eliminating low confidence noise areas around or on the realistic surface and obtaining relatively pure point clouds of a human hand within close distance. In order to maximize the retention of raw data for further use, the system does not use a smoothing filtering algorithm. All the algorithms in the system are implemented by GPU-assisted parallel computing, thus making the system achieve very high real-time performance. Finally, the experiment results show that the proposed filtering system could eliminate the vast majority of noise areas.

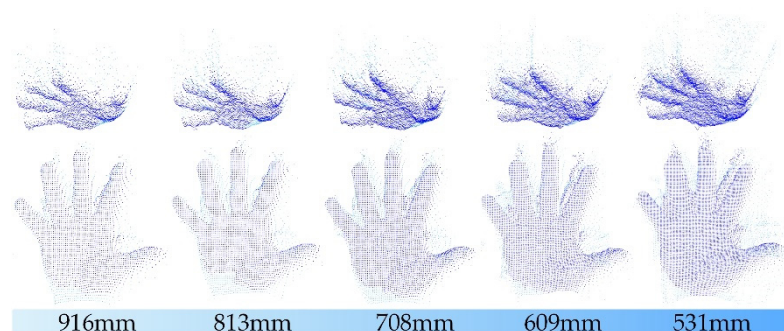


Figure 1. Point clouds of the human hand at different distances. The distance values are obtained by calculating the average of 25×25 pixels depth at the center of the hand.

2. Noise Characterization

Severely distorted noise of a close-range collected depth image tends to concentrate on the area of the image where the depth gradient is large. For a highly integrated device, the calibration method for laser scanner cannot be used [25], and as a result, the user cannot adjust the data generation process in most cases (it depends on the camera, SR300 could be allowed to adjust laser intensity and type of built-in filters), but only using the deep data acquired from the device. Hence an in-depth understanding of the noise characteristics within the depth images is the first task to build an effective filtering system.

The noise in the depth image is actually the sum of spatial noise and temporal noise. The former can be construed as an inaccurate depth measurement, which mainly includes the zero depth (ZD) that cannot be measured (like NaN [26]), and the wrong depth (WD) that is far from the actual depth value. The latter refers to the phenomenon that the measured value fluctuates with time when the depth of this point does not change, thus, multi-frames are needed to eliminate the temporal noise [27] that may cause input delay to the interface system. More detailed elaborations are made in [26–29].

For real-time interactive applications, any delay should be avoided, then the best way is to start with spatial noise and eliminate the low confidence WD areas. Therefore, in this section, according to shape and location, the noise areas that seriously affect the correctness of point cloud generation are of the hand surface classified into three types. Two of them are original noise which are shown in Figure 2, and another one is residual noise which will be described in Section 3.3.

- Outlier noise. This is the point with WD that exists away from the realistic surface, and is usually randomly distributed in the depth image spatially and temporally, which usually has impacts on the pass-through filtering [30].
- Edge noise. This kind of noise exists regionally. It can be an unrealistic surface composed of noise points surrounds the realistic surface, or it can be part of the edge of a realistic surface with WD. The closer to the blank area, the greater the depth gradient. It would eventually point to the z_+ or z_- (the positive or negative direction of the depth value).
- Plaque noise. This is a kind of residual noise. The filtering system may miss some plaque areas after filtering the first two kinds of noise. Most of the residual plaque areas are isolated and a few are connected to realistic surfaces.

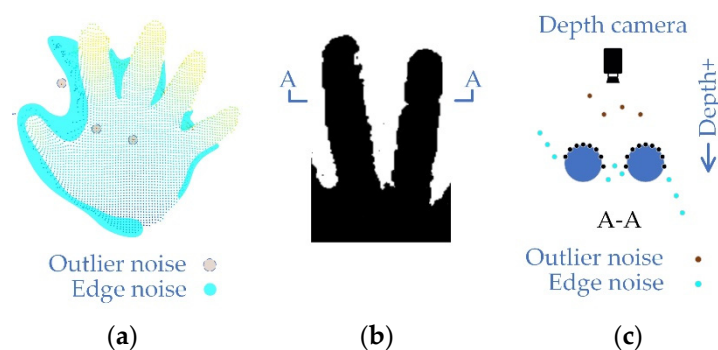


Figure 2. Noise classification. (a) A noisy point cloud image of the human hand at a distance of 800–900 mm. In order to show the details of the noise area, position A in (b) is cut and the section view is displayed in (c).

Together, these three types of noise constitute a low confidence noise area in depth images. It is worth noting that the characteristics of the first two noises are not independent of each other, if the gradient in the window is too large, the extreme points in the same window could be regarded as outliers. Therefore, in the next section, a composite filtering system will be proposed for these types of noise.

3. Proposed Filtering System

Based on the analysis of the noise types in the depth image acquired in close range, different noise characteristics make it difficult for a single filter to perform well. Therefore, a filtering system consisting of multiple detection modules is proposed in Figure 3. The CPU only needs to obtain the depth image from the device, and obtain the filtered point cloud data from the GPU and display them. All filtering algorithms are running in GPU, the calculation part such as standard deviation (SD) calculation, depth to 3D coordinate conversion follows the calculate-when-using principle to reduce the read and write frequency of the graphics card memory. By setting a reasonable number of

loops n_{loop} , it could eliminate most of the low confidence areas, and preserve realistic surfaces. Highly parallelized algorithms in the filtering system could save the computing resources of CPU and make the system run in real time.

3.1. Improved Dixon Test

The outliers seriously interfere with the depth value-based hand region truncation, often causing truncation failure. As for the adjacent points belonging to a same realistic surface, their depth values are usually very close. For any non-zero point (NZP) $p \in I_D$, both value range and the SD of $\delta(p)$ could not be too large, where $\delta(p) \subset I_D$ is the neighbor set of p .

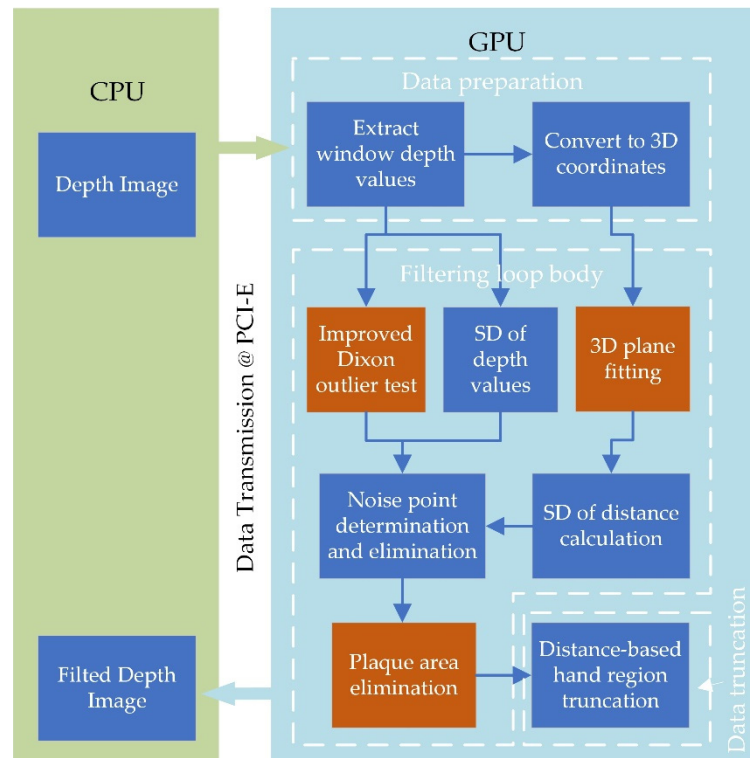


Figure 3. Proposed filtering system for hand depth image. The part with GPU algorithms of the system can be divided into three sub-parts. (1) Data preparation: extract the depth data of all points in the window corresponding to each thread, and work out their 3D coordinates. (2) Filtering loop body: as the core part of the filtering system, its main function is to identify the noise points and filter them out. (3) Data truncation: preserving the foreground and eliminating the depth image of the background. In addition, the three filtering algorithms proposed for different types of noise areas are marked in red boxes.

The central idea of the Dixon test is to determine whether extreme points are outliers by calculating the ratio between extreme point deviation and sample range. Equation (1) shows the Dixon outlier test method for up to 10 samples [31], where the Q_u and Q_l are used to identify the maximum and minimum sample respectively. x_1 and x_n are the extremes of the arranged samples, and the values of Q_u and Q_l could reflect how large the gap is. Different confidence levels (α) correspond to different limits of Q_u and Q_l , it can be obtained by looking up the table [31].

$$\begin{cases} 3 < n < 7 : Q_u = \frac{x_n - x_{n-1}}{x_n - x_1}, Q_l = \frac{x_2 - x_1}{x_n - x_1} \\ 8 \leq n \leq 10 : Q_u = \frac{x_n - x_{n-2}}{x_n - x_2}, Q_l = \frac{x_2 - x_1}{x_{n-1} - x_1} \end{cases}, x_1, \dots, x_n \in \delta(p) \quad (1)$$

However, to some extent, the ratio only reflects the deviation between the extreme point and cluster of other points spatially. Therefore, there are natural defects in applying it directly to outlier detection in depth images. The reason is that it cannot reflect the discreteness of depth values of all points in the window macroscopically. However, the SD, which can reflect the dispersion on the value of the samples, cannot microscopically reflect the positional relationship between each point and the cluster. Figure 4 and Algorithm 1 show the improvements of the Dixon test. There are three ways to determine whether p is an outlier.

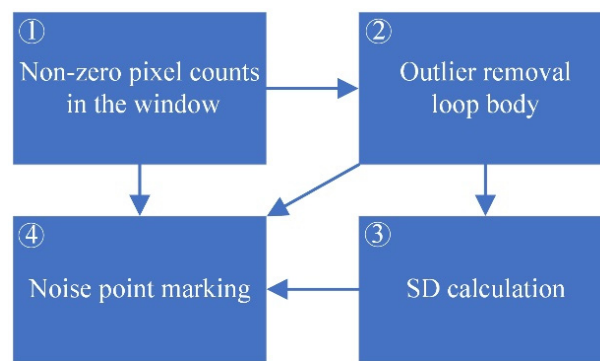


Figure 4. Improved Dixon test for depth image outlier detection. This algorithm combines the continuously draining outliers Dixon test and standard deviation (SD) calculation to perform macroscopic and microscopic identification of the center point.

Algorithm 1 Outliers Detection

Input: $\delta(p)$ **Output:** $Stat(tid)$, true for noise point

1: count NZPs: $n_{NZP} \leftarrow \text{counter}(\delta(p))$	10: if $p \in H$ then
2: if $n_{NZP} < k$ then	11: return: $Stat(tid) \leftarrow true$
3: return: $Stat(tid) \leftarrow true$	12: endif
4: endif	13: calculate SD: $SD \leftarrow \text{deviation}(H)$
5: ascending ordering: $H \leftarrow \text{sorter}(\delta(p))$	14: if $SD > SD_{max}$ then
6: $n_{remain} \leftarrow n_{NZP}$	15: return: $Stat(tid) \leftarrow true$
7: do	16: else
8: remove outliers: $\text{DixonTest}(H, n_{remain})$	17: return: $Stat(tid) \leftarrow false$
9: while $k < n_{remain} < n$	18: endif

- ① → ④, for any p -centred $k \times k$ size window, when the number of NZP $n_{NZP}(\delta(p)) < k$, that is, there are only at most $k - 2$ NZPs except p . As is shown in Figure 5, as an outlier, p locates on the jagged edges or in the blank area, and it can be directly identified as a noise point.
- ① → ② → ④, the loop body in ② eliminates the outliers in the sample repeatedly until there are no more points that can be identified as an outlier. If p is eliminated, then it could be defined as an outlier.
- ① → ② → ③ → ④, the remaining NZPs in ② will be sent to an SD calculator to make a macro evaluation of dispersion. p will be marked as an outlier if the SD is too large.

The improved algorithm not only checks whether the extreme point is an outlier, but the SD test is also carried out on the set after eliminating all outliers at the same time, which limits the dispersion of the cluster formed by remaining points. Thus, the macroscopic dispersion and the microscopic position distribution in the window could be simultaneously detected.

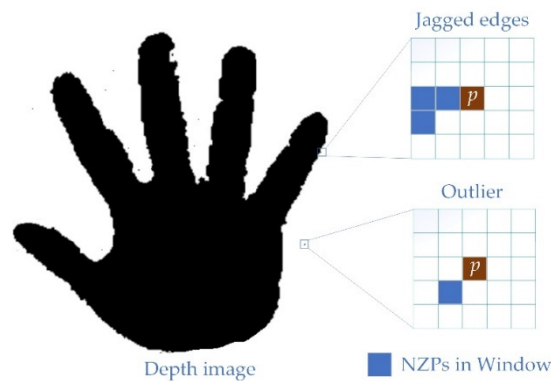


Figure 5. The locations of the points determined to be eliminated. For window areas with a small number of non-zero points, most of them are outliers or located on the jagged edge. This figure shows this case with $k = 5$. Other sizes of windows are similar.

3.2. Edge Noise Filtering Approach

For depth images, the maximum depth gradient within a window should be limited by the line of sight. Thus, if any p is determined as an edge noise point, the maximum gradient in any p -centered window should be larger than a specific value.

In addition, this type of noise area can be eliminated by the approach which is shown in Figure 6. According to $n_{NZP}(\delta(p))$, the determination of edge noise point will be based on the following three cases.

- $n_{NZP}(\delta(p)) < k$, when p locates at the jagged edges, p is defined as a noise point.
- $1.n_{NZP}(\delta(p)) = k$, if NZPs is arranged as a straight-line, p is defined as a noise point.
- $n_{NZP}(\delta(p)) > k$, in this case, fitting a plane π to $G(\delta(p))$ by using the least-square method, where $G(*)$ is the converted 3D global coordinates of $\delta(p)$. Let $S(G(\delta(p)), \pi)$ be the SD of the vertical distances from $G(\delta(p))$ to π , and $\alpha(\pi, \lambda)$ be the angle between π and sight plane λ . If $S(G(\delta(p)), \pi) > S_\pi$ or $\alpha(\pi, \lambda) > \alpha_p$, define p as a noise point.

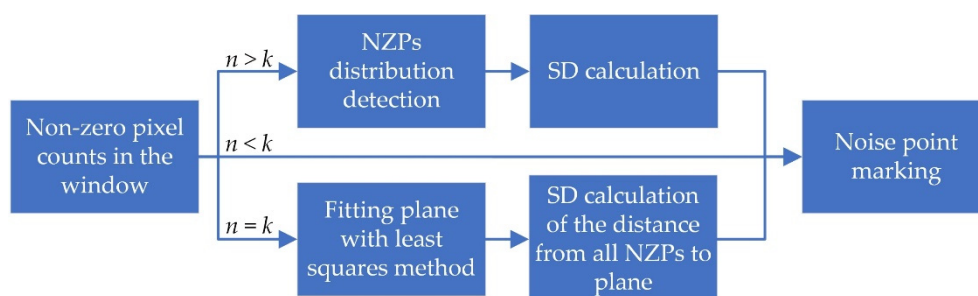


Figure 6. Edge noise filtering. Identification of this kind of noise points is mainly based on the relative position between non-zero points (NZPs).

When $n_{NZP}(\delta(p)) \leq k$, the algorithm makes noise point judgments for the number and distribution of NZPs. In another case, the algorithm fits NZPs into a three-dimensional plane, and the decision is made by the angle of the plane and the dispersion of the distance between the plane and NZPs. s_p and α_p are the given threshold values for judging p 's state. Selecting the appropriate threshold will help the system to separate the realistic area from the noise area.

3.3. Plaque Noise Filtering Approach

Part of the plaque noise areas is a kind of residual noise, and they come from the residual part filtered by the above steps. Most of them are isolated and located in the areas that are supposed

to be blank as an unrealistic surface. There are also some plaques connected to the realistic surface, which means that the serial algorithm for keeping the hand area by comparing the length of the chain code [32,33] will not always be effective. However, the use of a larger window is likely to cause excessive elimination of the hand area. Therefore, a method for orthogonally detecting the number of consecutive non-zero points is presented in Figure 7 to eliminate the plaque area.

As shown in Figure 7, threads are opened for each $p \in I_D$ to search the adjoining continuous NZPs along rows and columns respectively, and obtain the numbers n_{px} and n_{py} which stand for how many NZPs are connected to p in the direction of rows and columns (including itself). By putting limits on n_{px} , n_{py} , and their product m_p , respectively, all plaque areas could be marked. The pseudo-code for one thread is shown in Algorithm 2.

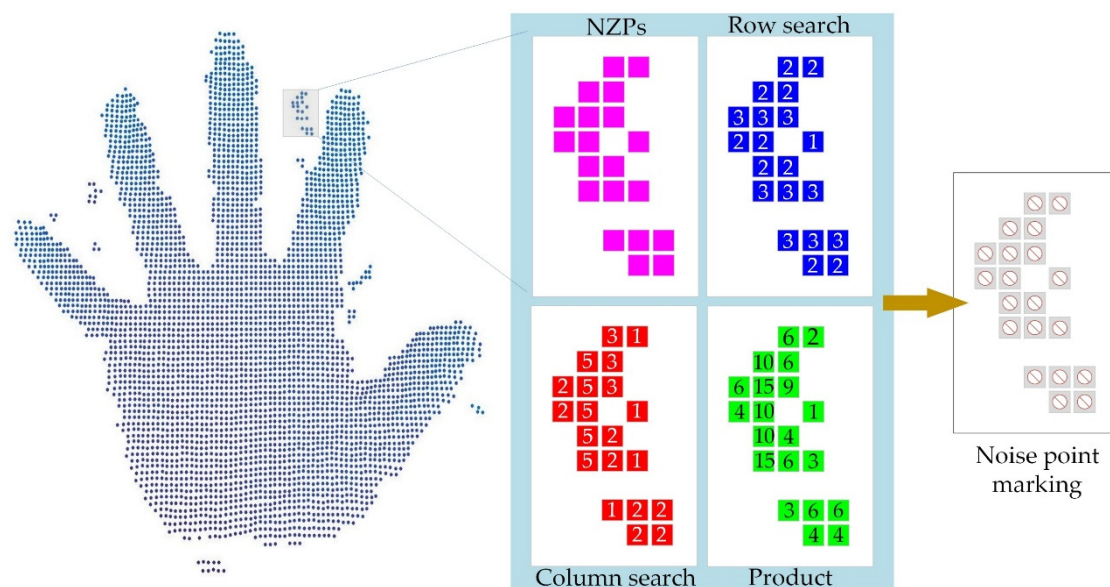


Figure 7. Orthogonally detecting the number of consecutive non-zero points of p .

Algorithm 2 Plaque Noise Area Detection

Input: $ID, n_{pymin}, n_{pymmin}, m_{pmin}$

Output: $Stat(tid)$, true for noise point

```

1:   row search: Xsearcher ( $n_{px}, I_D$ )
2:   col search: Xsearcher ( $n_{py}, I_D$ )
3:    $m_p = n_{px} \times n_{py}$ 
4:   if  $n_{px} < n_{pxmin} || n_{py} < n_{pymmin} || m_p < m_{pmin}$  then
5:     return:  $Stat(tid) \leftarrow \text{true}$ 
6:   else
7:     return:  $Stat(tid) \leftarrow \text{false}$ 
8:   endif

```

4. Experiments

To the best of our knowledge, studies based on a non-smooth filter specially used for filtering the high noise point clouds generated by consuming depth cameras when used at close range have not been reported. Therefore, for comparison, we employed the standard median filter (SMF) and skin color based depth image classification (SCBDIC) (a fused method presented in [34,35]). Its principle is to register the depth and color image, and then remove the unrealistic surface from the point cloud by recognizing the skin color region. However, to obtain universal experimental results, all the experiments were carried out in an indoor fluorescent light environment, and the lighting conditions were not deliberately improved.

All experiments were conducted on a computer with an Intel Core i7 4770 @ 3.6 Ghz CPU and a Nvidia GTX 1060-6 GB graphics card. The depth sequence was captured using a Kinect v2.0 with resolution 512×424 at 30 fps and a SR300 with resolution 640×480 at 30 fps. The programming environment was Visual Studio 2017 with CUDA 9.2 version.

To prove the validity of our filtering system, we experimented with Kinect v2 and SR300 on hand depth images at different distances. Figure 8 shows the comparison of the proposed filtering system with the other two filters when using Kinect v2. A large amount of edge noise (marked by the blue circle) exists in the original depth image, which constitutes the unrealistic surfaces in the point cloud. However, the part of the color image corresponding to these unrealistic surfaces was not the skin color area, so they could be well removed by SCBDIC. However, the outliers and edge noise located inside the realistic surface (marked by the gray and black circle) could not be filtered out. More seriously, under different light and different angles, the colors had different changes, which can cause many hand areas (marked by the red circle) to be incorrectly recognized, resulting in over-filtering. On the contrary, SMF seems to be ineffective for such large noise areas, and can only filter out some outliers. The change in window size could not provide a better filtering effect, so we present the filtering effect of SMF when $k = 3$ in Figure 8.

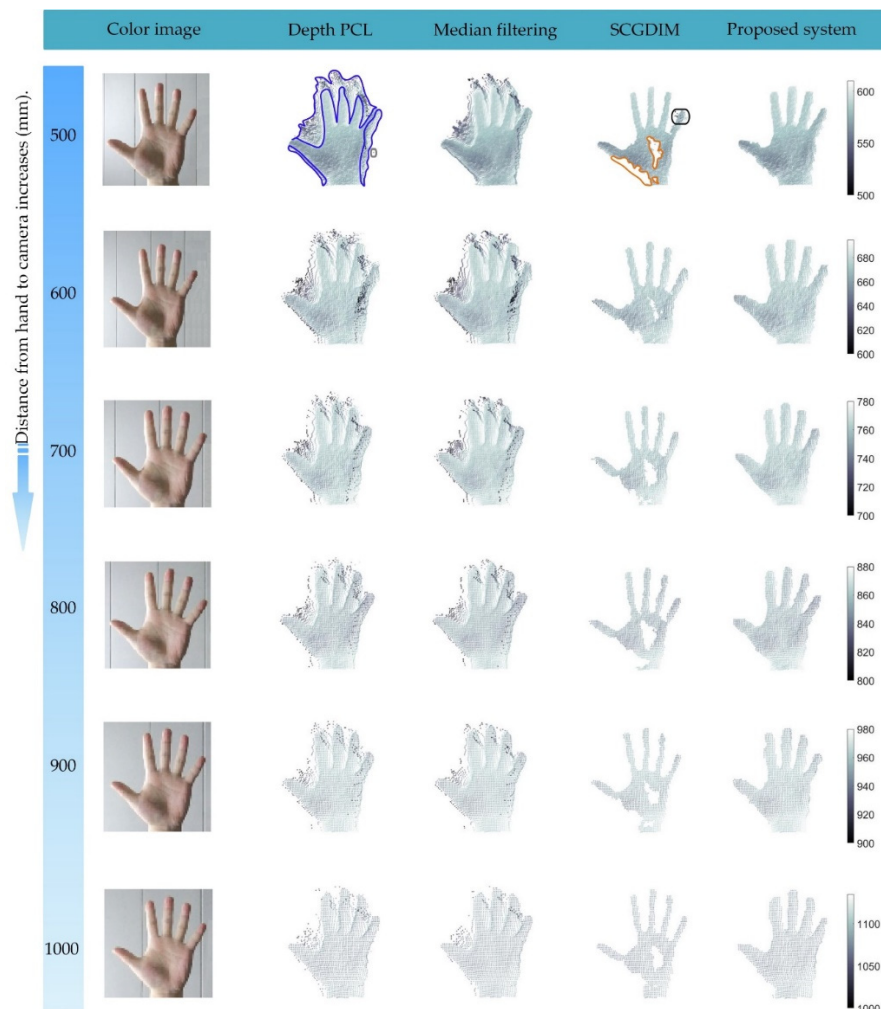


Figure 8. Comparisons at different distances by using Kinect v2.

The proposed filter does not depend on other input sources. As long as the parameters are set reasonably, it can recognize almost all the noise areas and outliers then eliminate them. Even at a close distance of 500 mm, it still produces good results. By querying the judgment status of each

sub-filtering algorithm, one type of noise area is not only recognized by its corresponding filtering algorithm, more often, it is recognized by both outlier filtering and edge noise filtering algorithms at the same time. This is because the region recognized as edge noise usually has high SD, which is also one of the characteristics of outlier noise. The reason why p is only recognized as edge noise is that the area where it is located is relatively smooth, but the angle between the fitted plane and the plane of view is too large. In addition, the reason why p is only recognized as an outlier is that in its window, the depth value of p is significantly different from other NZPs, and the values of other NZPs are not much different.

Since point clouds from SR300 hardly produce the realistic surfaces, the SMF that can filter out part of the outliers. Visually, this effect gets better as the filter window increases. As is shown in Figure 9, the edge noise area in orange circle and the outlier in gray circle are smoothed by SMF, and other noise areas in blue circles were also improved to some extent. However, at the same time, the gap between the two fingers (marked in red circle) was filled. At the same time, the depth values of almost all points were changed. This is equivalent to introducing a new error source. The proposed filtering system eliminated almost all edge noise regions without changing the depth value any point and preserved the raw depth data.

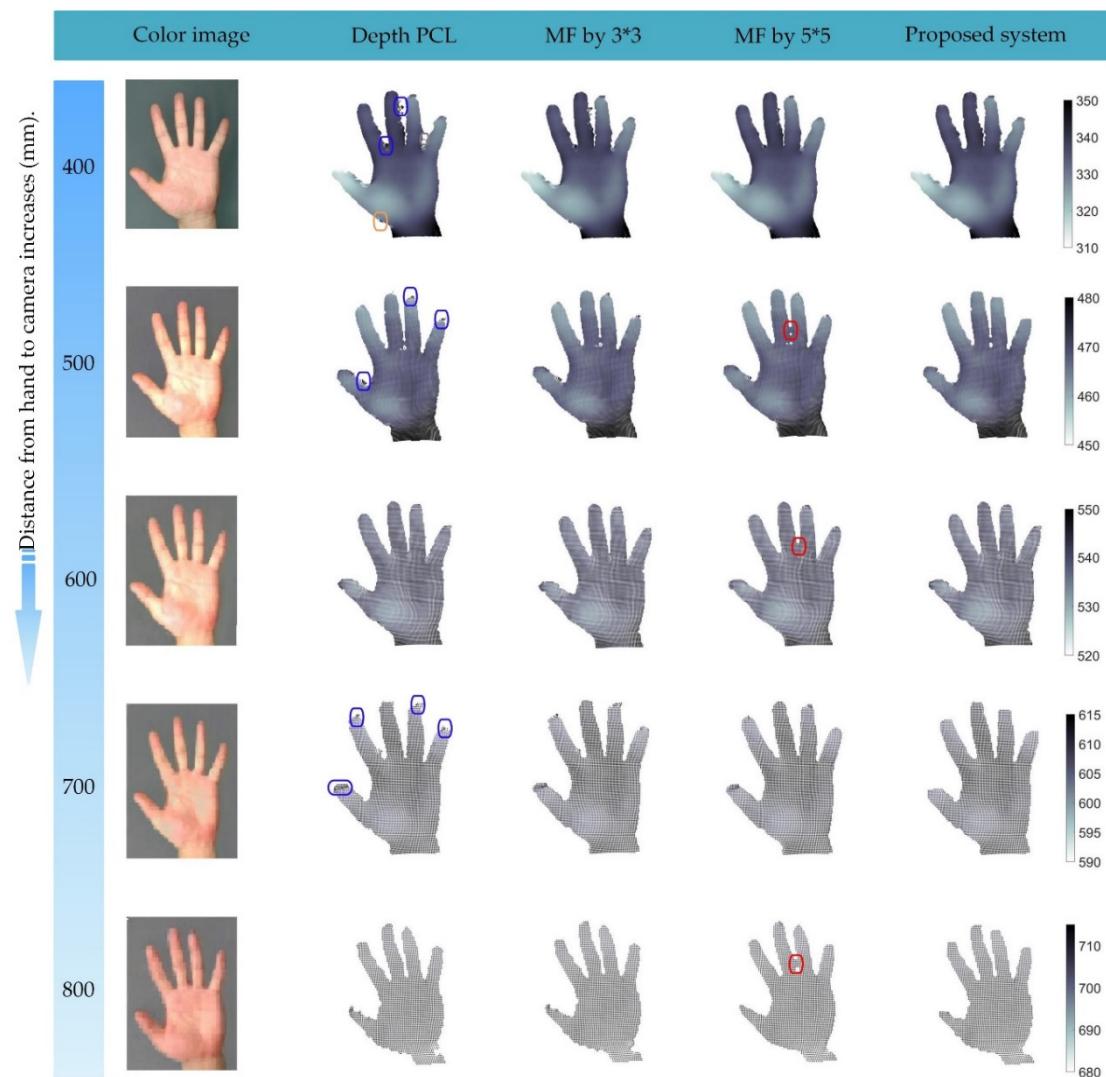


Figure 9. Comparisons at different distances by using SR300.

Figures 10 and 11 show three views of the filtering effect of point clouds by using Kinect v2 and SR300 respectively. For two kinds of equipment with totally different noise characteristics, the proposed system can maintain a good filtering effect. It means that the proposed filtering system has certain universality by setting appropriate parameters. To get better results globally, the determination of the parameters of the filtering system requires a lot of experiments. We present the parameters used in the experiments in this paper which are listed in Table 1.

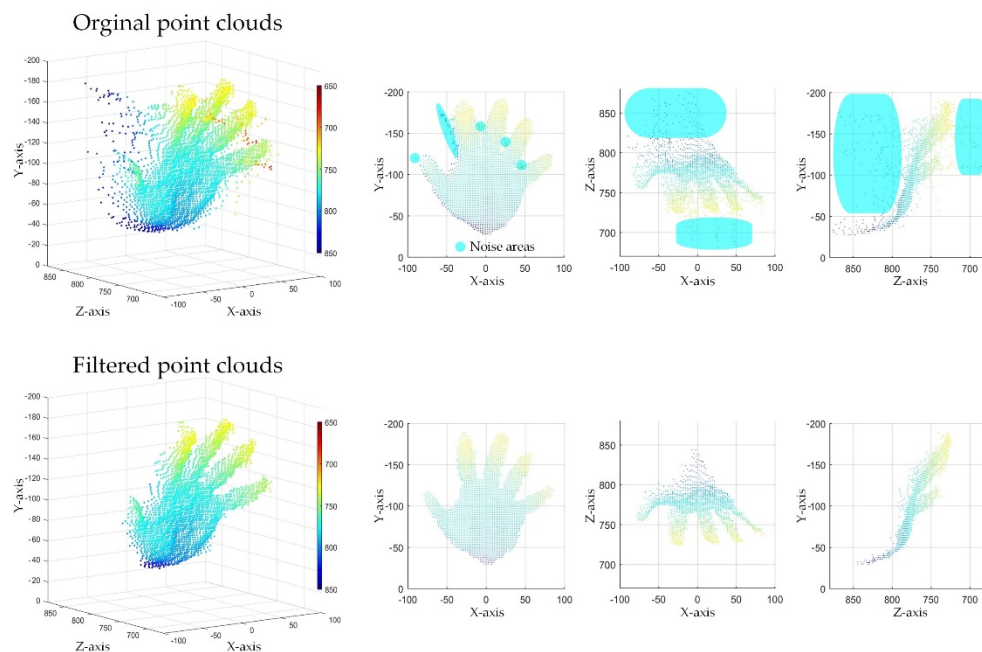


Figure 10. Three views of the filter result (Kinect v2).

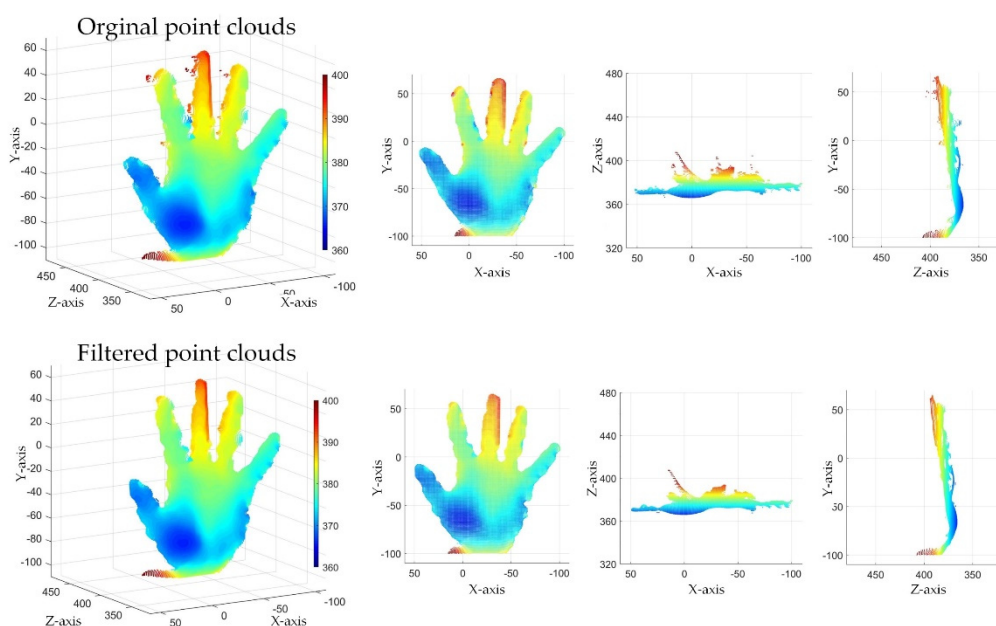


Figure 11. Three views of the filter result (SR300).

To evaluate the stability and real-time performance of the system, 1000 frames of continuous images were recorded as experimental material, and the run time for each frame was stable at 5 milliseconds, and the filtering effect video is updated as the supplementary material. Since all algorithms in this proposed filtering system adopt the parallel structure, the running speed of the system is not very sensitive to the resolution of depth image and more determined by GPU performance.

At the same time, it also has excellent performance in stability. In Figure 12, 18 frames of different hand postures are shown, and most of them have very good filtering effects. However, it is noteworthy that, when a gradient of a part of the realistic surface gets too large, the system may determine it as an edge noise area and eliminate it. The frame in a red box indicates this situation.



Figure 12. Comparisons of different hand gestures.

5. Conclusions

When collecting depth images with a consumer depth camera, the noise interference becomes more serious as the object approaches. In order to eliminate these noise areas and obtain a correct pure raw point cloud with high resolution of the object, we proposed a new filtering system for using consumer depth cameras at close range in this paper.

We classified the noise areas into three types, outlier noise, edge noise, and plaque noise. By analyzing the characteristics of these three noise types, we specially designed a filtering algorithm for each noise type: (1) an improved Dixon test algorithm for filtering outlier noise, (2) a three-dimensional plane fitting method to eliminate edge noise, and (3) an algorithm based on searching for the number of adjacent joints for the plaque noise. All algorithms adopted the parallel structure, which greatly improved the efficiency of the filtering system. The running speed of nearly 200 frames per second can meet the application of most real-time interactive systems. We tested the filtering system using two different depth cameras, and the filtering effects were much better than the other two filters involved in the comparison. This shows that the proposed filtering system has certain universality. At the same time, we also presented the system parameters that can achieve a better global filtering effect with the two cameras. Finally, in order to test the stability of the filtering effect, we used 1000 frame continuous hand depth images as experimental materials. The filtering effects show that the system can effectively eliminate most of the noise areas, and 18 of them were selected to present the filtering effect.

Excellent real-time, good filtering effect, and a certain degree of universality enables the proposed filtering system to be used as a pre-step for real-time human-computer interaction, real-time 3D reconstruction, and further filtering.

Table 1. Experimental parameters.

Dis.\Para.	k	n_{loop}	α	S_p	n_{p_x}	n_{p_y}	m_p	S_π	α_p
Kinect v2									
500 mm	3	20	0.1000	4.3	7	7	25	3	75
600 mm	3	20	0.1000	4.55	5	5	25	3	75
700 mm	3	15	0.1000	4.8	5	5	25	3	75
800 mm	3	15	0.1000	5.05	4	4	16	3	75
900 mm	3	10	0.1000	5.3	4	4	16	3	75
1000 mm	3	10	0.1000	5.55	4	4	16	3	75
SR300									
400 mm	3	10	0.1000	2.5	5	5	25	1.5	70
500 mm	3	10	0.1000	2.5	5	5	25	1.5	70
600 mm	3	10	0.1000	2.3	5	5	25	1.2	70
700 mm	3	10	0.1000	2.2	4	4	16	1	70
800 mm	3	10	0.1000	2.2	4	4	16	1	70

Future Works

In the future, on the one hand, we will try to develop a method to evaluate the filtering effect, which can be used to realize the automatic optimization of system parameters, and increase or modify some sub-algorithms by using other kinds of cameras to improve the universality of the filtering system. On the other hand, we will try to develop a new algorithm that replaces the points that are marked as noise instead of simply removing them, making the filtered point cloud image edges smoother.

Supplementary Materials: The following are available online at <http://www.mdpi.com/1424-8220/19/16/3460/s1>, Video S1: filtering effect at close range by using kinect v2.

Author Contributions: In this study, Y.D. is responsible for literature retrieval, charting, research and design, data collection, data analysis, manuscript writing and other practical work. In the process, we got some suggestions about experimental design from Y.F., adopted some research and design methods from B.L., and got the help of X.Z. in the process of data collection. Finally, after the completion of this paper, it was approved by Professor T.Y. and Professor W.W.

Funding: This research received no external funding.

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

References

1. Hansard, M.; Horaud, R.; Amat, M.; Evangelidis, G. Automatic detection of calibration grids in time-of-flight images. *Comput. Vision Image Understanding* **2014**, *121*, 108–118. [\[CrossRef\]](#)
2. Grzegorzec, M.; Theobalt, C.; Koch, R.; Kolb, A. Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications. *Lect. Notes Comput. Sci.* **2013**, *8200*, 354–360.
3. Draelos, M.; Deshpande, N.; Grant, E. The Kinect up close: Adaptations for short-range imaging. In Proceedings of the Multisensor Fusion & Integration for Intelligent Systems, Hamburg, Germany, 13–15 September 2012.
4. Han, X.-F.; Jin, J.S.; Wang, M.-J.; Jiang, W. Guided 3D point cloud filtering. *Multimedia Tools Appl.* **2018**, *77*, 17397–17411. [\[CrossRef\]](#)
5. Buttazzo, G.; Lipari, G.; Abeni, L.; Caccamo, M. *Soft Real-Time Systems: Predictability vs. Efficiency (Series in Computer Science)*; Plenum Publishing Co.: Pavia, Italy, 2005.
6. Gogouvitis, S.; Konstanteli, K.; Waldschmidt, S.; Kousiouris, G.; Katsaros, G.; Menychtas, A.; Kyriazis, D.; Varvarigou, T. Workflow management for soft real-time interactive applications in virtualized environments. *Future Gener. Comput. Syst.* **2012**, *28*, 193–209. [\[CrossRef\]](#)
7. Ma, Z.; Wu, E. Real-time and robust hand tracking with a single depth camera. *Vis. Comput.* **2014**, *30*, 1133–1144. [\[CrossRef\]](#)
8. Novak-Marcincin, J.; Torok, J. Advanced Methods of Three Dimensional Data Obtaining for Virtual and Augmented Reality. *Adv. Mater. Res.* **2014**, *1025*, 1168–1172. [\[CrossRef\]](#)
9. Nguyen, B.P.; Tay, W.L.; Chui, C.K. Robust Biometric Recognition From Palm Depth Images for Gloved Hands. *IEEE Trans. Hum. -Mach. Syst.* **2015**, *45*, 799–804. [\[CrossRef\]](#)
10. Deng, X.; Yang, S.; Zhang, Y.; Tan, P.; Chang, L.; Wang, H. Hand3D: Hand Pose Estimation using 3D Neural Network. *arXiv* **2017**, arXiv:1704.02224.
11. Sun, X.; Wei, Y.; Liang, S.; Tang, X.; Sun, J. Cascaded hand pose regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 824–832.
12. Qian, C.; Sun, X.; Wei, Y.; Tang, X.; Sun, J. Realtime and Robust Hand Tracking from Depth. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 1106–1113.
13. Morana, M. 3D Scene Reconstruction Using Kinect. In *Advances onto the Internet of Things: How Ontologies Make the Internet of Things Meaningful*; Gaglio, S., Lo Re, G., Eds.; Springer International Publishing: Cham, Switzerland, 2014. [\[CrossRef\]](#)
14. Maimone, A.; Bidwell, J.; Peng, K.; Fuchs, H. Enhanced personal autostereoscopic telepresence system using commodity depth cameras. *Comput. Graph.* **2012**, *36*, 791–807. [\[CrossRef\]](#)
15. Buades, A.; Coll, B.; Morel, J. A non-local algorithm for image denoising. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 60–65.
16. Zhang, B.; Allebach, J.P. Adaptive Bilateral Filter for Sharpness Enhancement and Noise Removal. *IEEE Trans. Image Process.* **2008**, *17*, 664–678. [\[CrossRef\]](#)
17. Petschnigg, G.; Szeliski, R.; Agrawala, M.; Cohen, M.; Hoppe, H.; Toyama, K. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.* **2004**, *23*, 664–672. [\[CrossRef\]](#)
18. Essmaeel, K.; Gallo, L.; Damiani, E.; De Pietro, G.; Dipanda, A. Comparative evaluation of methods for filtering Kinect depth data. *Multimed. Tools Appl.* **2015**, *74*, 7331–7354. [\[CrossRef\]](#)
19. Lo, K.-H.; Wang, Y.-C.F.; Hua, K.-L. Edge-Preserving Depth Map Upsampling by Joint Trilateral Filter. *IEEE Trans. Cybern.* **2018**, *48*, 371–384. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Yuan, L.; Sun, J.; Quan, L.; Shum, H.Y. Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph.* **2007**, *26*, 1. [\[CrossRef\]](#)
21. Le, A.V.; Jung, S.W.; Won, C.S. Directional Joint Bilateral Filter for Depth Images. *Sensors* **2014**, *14*, 11362–11378. [\[CrossRef\]](#)
22. Ran, L.; Li, B.; Huang, Z.; Cao, D.; Tan, Y.; Deng, Z.; Miao, X.; Jia, R.; Tan, W. Hole filling using joint bilateral filtering for moving object segmentation. *J. Electron. Imaging* **2014**, *23*, 063021.
23. Cai, Z.; Han, J.; Liu, L.; Shao, L. RGB-D datasets using microsoft kinect or similar sensors: A survey. *Multimedia Tools Appl.* **2017**, *76*, 4313–4355. [\[CrossRef\]](#)

24. Camplani, M.; Mantecon, T.; Salgado, L. Depth-Color Fusion Strategy for 3-D Scene Modeling With Kinect. *IEEE Trans. Cybern.* **2013**, *43*, 1560–1571. [[CrossRef](#)]
25. Reshetyuk, Y. *Terrestrial Laser Scanning: Error Sources, Self-Calibration and Direct Georeferencing*; VDM Verlag: Stockholm, Sweden, 2009.
26. International Organization for Standardization. ISO/IEC/IEEE 60559:2011, Information Technology—Microprocessor Systems—Floating-Point Arithmetic. 2011. Available online: <https://www.iso.org/standard/57469.html> (accessed on 7 August 2019).
27. Nazir, S.; Rihana, S.; Visvikis, D.; Fayad, H. Technical Note: Kinect V2 surface filtering during gantry motion for radiotherapy applications. *Med. Phys.* **2018**, *45*, 1400–1407. [[CrossRef](#)]
28. Yu, Y.; Song, Y.; Zhang, Y.; Wen, S. A Shadow Repair Approach for Kinect Depth Maps. In Proceedings of Computer Vision—ACCV 2012, Proceedings of the 11th Asian Conference on Computer Vision, Daejeon, Korea, 5–9 November 2012.
29. Accuracy (Trueness and Precision) of Measurement Methods and Results—Part 1: General Principles and Definitions. Available online: <https://www.iso.org/standard/11833.html> (accessed on 31 May 2019).
30. Filtering a PointCloud Using a PassThrough Filter. Available online: <http://pointclouds.org/documentation/tutorials/passthrough.php> (accessed on 20 May 2019).
31. Böhner, A. One-sided and Two-sided Critical Values for Dixon's Outlier Test for Sample Sizes up to $n = 30$. *Econ. Qual. Control* **2008**, *23*, 5–13. [[CrossRef](#)]
32. Kim, S.D.; Lee, J.H.; Kim, J.K. A new chain-coding algorithm for binary images using run-length codes. *Comput. Vis. Graph. Image Process.* **1988**, *41*, 114–128. [[CrossRef](#)]
33. Structural Analysis and Shape Descriptors. Available online: https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=findcon#cv2.findContours (accessed on 20 May 2019).
34. Khan, R.; Hanbury, A.; Stöttinger, J.; Bais, A. Color based skin classification. *Pattern Recognit. Lett.* **2012**, *33*, 157–163. [[CrossRef](#)]
35. Kang, S.I.; Roh, A.; Hong, H. Using depth and skin color for hand gesture classification. In Proceedings of the IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 9–12 January 2011.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).