ISSN: 2664-9640

Depth Image Inpainting Based on Improved Interpolation and Filtering Algorithm

Yingjie Xu

School of Mechanical Engineering, Heilongjiang University of Science and Technology, Harbin 150000, China.

Abstract

Large area holes and scattered noise points are easy to appear in the depth image collected by depth camera, I propose a nearest neighbor interpolation algorithm via diffusing from center to periphery to repair holes and present an improved mean filtering algorithm with weight to remove noise points. The experimental results show that my algorithm can well complete the depth image inpainting and preserve the depth information of the object edge, the inpainting effect and execution efficiency are better than some commonly used traditional interpolation algorithms and filtering algorithms.

Keywords

Depth Image, Image Inpainting, Interpolation Algorithm, Filtering Algorithm.

1. Introduction

In 2010, Microsoft launched Kinect V1 which is the first generation of depth camera, creating a precedent for consumer depth cameras to enter the market. Consumer depth camera has the advantages of mobility and low price, which has attracted a large number of scholars to apply it to the fields of virtual reality[1], three-dimensional reconstruction[2] and indoor robot[3]. Compared with color camera, it can not only provide color data, but also obtain the depth data from the object to the camera, which is expressed in the form of depth image. However, due to the influence of environmental noise, object surface material and limited measurement range, there are large area black holes and scattered noise points in the depth image. It is necessary to obtain more accurate depth data through hole repair and noise point removal, and then it is valuable to apply it to subsequent research.

Common holes repair methods include fast marching method[4](FMM) and curvature-driven diffusion(CDD) method[5], noise points removal algorithms include median filtering(MF) algorithm[6], bilateral filtering(BF) algorithm[7] and joint bilateral filtering(JBF) algorithm[8]. While doing depth image hole repairing and noise point removing, FMM needs to find the edge of the hole firstly and then fill it gradually from the edge to the center with the help of the effective depth data in the edge neighborhood, however, it is difficult to determine the edge of the hole; CDD method is predicated upon the effective depth data in the edge neighborhood of the hole, using it's isophote and curvature-driven factor to calculate the diffusion information for holes repairing, but for large area holes, the problem of edge information loss will occur; Although MF algorithm can excellently remove noise points by using the median value of effective depth data in the window, the effective depth data is changed and the problem of edge information loss is serious; BF algorithm comprehensively considers the image spatial information and pixel gray value information in the filter, which can better retain the edge information, but can not realize the repair of large area holes; On the basis of bilateral filter, JBF algorithm introduces the color image as the guide map, which can better retain the edge information and has a good repair effect for large area holes, however, the introduction of color map results in the execution efficiency of the algorithm is low.

ISSN: 2664-9640

For the depth image obtained by Kinect V2 camera, I propose a nearest neighbor interpolation algorithm that diffuses from center to periphery to quickly repair the large area hole in the depth image. In addition, I present an improved mean filter algorithm with weight, which can effectively retain the depth information of the object edge while removing the scattered noise points. Through holes repair and noise points removal, the depth image has more practical value in subsequent applications such as virtual reality, three-dimensional reconstruction, indoor robot and so on.

2. Hole Repair

The traditional nearest neighbor interpolation algorithm[9] is to find the effective pixel value with the smallest Euclidean distance around the hole to complete repair, providing an important reference for my hole repair algorithm, which is defined as follows:

$$P^{-}(x,y)|_{(x,y)\in R} = P(x^{+},y^{+}) \tag{1}$$

$$(x^{+},y^{+}) = \arg\min_{(u,v)\in R^{+}} (\sqrt{(u-x)^{2}+(v-y)^{2}}) \tag{2}$$

Where $P^{-}(x,y)$ is the pixel to be repaired, $P(x^{+},y^{+})$ is the effective pixel value in the hole neighborhood, R is set of inefficient point, R^{+} is set of efficient points in hole neighborhood.

2.1. Threshold Processing

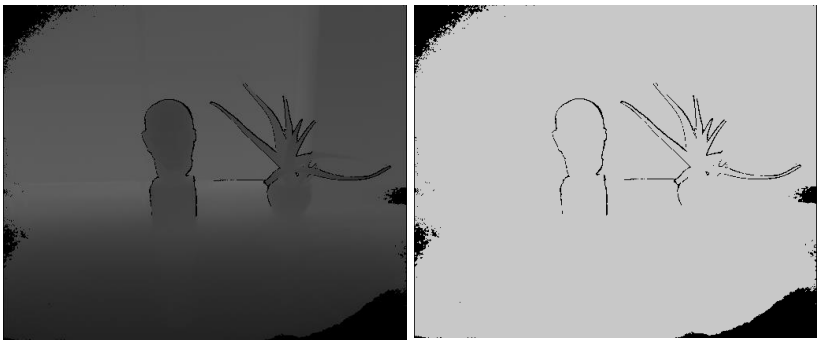
Each pixel in Kinect V2 depth image is saved as 16 bits in memory, the first 13 bits record depth data and the last 3 bits record user ID. If the depth data at a pixel is not available due to environment noise or measurement distance out of range, its pixel value is zero. The effective measuring range of Kinect V2 depth camera is 0.5 ~ 4.5m, while the depth value exceeds this range, the measurement error increases sharply, so I use the pixel threshold processing method to value the pixels beyond the effective range as zero, excluding pixels with large errors. The pixel value of 8-bit depth image which benefits to computer display is between 0~255, so I define the depth image pixel values as follows:

$$p = \begin{cases} 0, & P < 500 \\ \dfrac{P*255}{4500}, & 500 \le P \le 4500 \\ 0, & P > 4500 \end{cases} \tag{3}$$

where P is the pixel value of a point in the depth image before threshold processing and p is the pixel value after threshold processing of this point.

Using Kinect V2 camera to take a depth image, figure 1 shows the 8-bit depth image after threshold processing and highlights holes and noise points:

ISSN: 2664-9640



(a) (b)

Figure 1: Threshold processing results:(a)Image after threshold processing;(b)Holes and noise points

2.2. Nearest Neighbor Interpolation Algorithm via Diffusing from Center to Periphery

Because the depth camera of Kinect V2 is not at the same point as the infrared camera, there is an undetectable area caused by geometrical optics occlusion, furthermore it's measurement range is limited, causing the edge of the depth image is prone to black holes. Nearest neighbor interpolation algorithm and FMM are often used to repair such holes. The core idea of nearest neighbor interpolation algorithm is to use the non-zero pixels with the nearest Euclidean distance around the invalid area of depth image to fill the hole. The process of repairing holes by FMM is to find the boundary position of the hole, and then fill it layer-by-layer from outside to inside with the help of the pixel values in the boundary field until the hole is repaired.

Combining the core ideas of the above two algorithms, I use the nearest neighbor effective pixel value to fill the hole based on the similarity of pixels in the neighborhood, furthermore, the confidence of pixels at the boundary of the hole is higher than that of the inside so that we need to repair the hole from the outside to the inside. To sum up, I propose a nearest neighbor interpolation algorithm via diffusing from center to periphery to repair holes in depth images, the steps are:

(1)Select the center point of depth image as the starting point, whose coordinate is (212,256), at the same time, divide the depth image into four regions with the same size: I, II, III and IV;

(2)Starting from the center point, four areas are searched for holes at the same time by a 3*3 windows in each area, area I is from bottom to top and from right to left, area II is from bottom to top and from left to right, area III is from top to bottom and from right to left, and Area IV is from top to bottom and from left to right. If the confidence of the current window is zero, that is, the values of all nine pixels in the window are zero, it will be regarded as a hole area and then I will make the pixel value of the center point of the window 255 as a white mark point;

(3)Use the 3 * 3 window and repeat the window sliding step of the second step, when a white mark appears in the center of the window, I will find the pixel value that is not zero in the order of Euclidean distance from small to large to fill the hole in area I from the lower right of the marked point. Similarly, area II is searched from the lower left of the marked point, area III is searched from the upper right of the marked point, and Area IV is searched from the upper left of the marked point.

Figure 2 shows a schematic diagram of the sliding window during the execution of the algorithm, the black area represents the hole in the depth image, the white grid is the white mark of the center point of the window, the center point with the coordinate of (212,256) divides the depth image with the size of 414*512 into four areas as shown in the figure, four windows with the size of 3*3 slide simultaneously according to their respective rules in the direction indicated by the arrow.

ISSN: 2664-9640

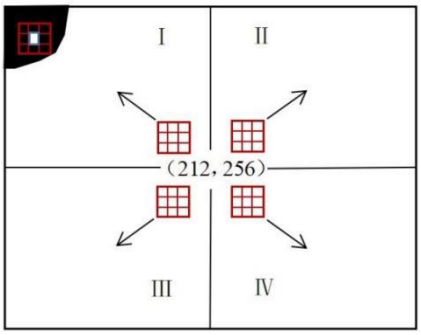
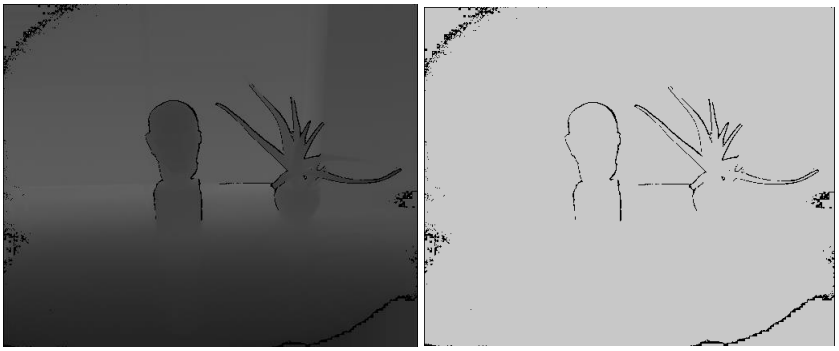


Figure 2: Schematic diagram of sliding window

2.3. Hole Repair Result and Analysis

The method of diffusing from center point of the depth image to the periphery can quickly find the hole area and ensure that the hole filling order is from the outside to the inside, which effectively solves the problem that it is difficult to determine the position of the hole area in the FMM algorithm, moreover, it is almost impossible for a large area of black hole to appear at the center of the depth image, so my algorithm can effectively repair all holes in the depth image. I mark the center point of the window with confidence of zero with a white dot, so I can accurately find the hole area for repair and ignore the other areas that do not need to be repaired in the process of hole repair, which can greatly reduce the processing time. In addition, the four regions are processed at the same time, which also achieves the effect of reducing the processing time and improving the efficiency of my algorithm. Figure 3 shows the image after hole repair by my algorithm:



(a) (b)

Figure 3: Hole repair results:(a)Image after hole repair;(b)Image residual defect

We can see from figure3 that my neighbor interpolation algorithm via diffusing from center to periphery can effectively repair the large area black holes in the depth image. However, when the nearest neighbor interpolation is adopted, the minimum euclidean distance is effective within the set threshold range, when the non-zero pixel value is not found within the maximum threshold range, the white mark points in the window will not be removed. As a result, there are very few white points in the depth image after hole repair, and I will clear them after noise points removal.

3. Noise Points Removal

3.1. Improved Mean Filtering Algorithm with Weight

Due to the influence of ambient light and other factors, there will be scattered noise points in the depth image, which often appear at the edge of the object. For such noise points, filtering algorithm is usually used to remove them. Although linear filtering algorithms such as mean filtering, median filtering and adaptive median filtering(AMF) have fast processing speed, they

ISSN: 2664-9640

are easy to cause object edge blur; Nonlinear filtering algorithms such as joint bilateral filtering algorithm can achieve the effect of noise points removal and edge preservation, but they have large amount of calculation and long processing time, can't meet the requirements of real-time, as a result, it is not suitable for real-time object recognition, three-dimensional reconstruction and other occasions.

In order to quickly and effectively remove scattered noise points and reduce object edge blur in depth image, I present an improved mean filtering algorithm with weight. The traditional mean filtering algorithm is a linear filtering algorithm, which uses the average value of all pixels in the sliding window to replace the pixel value in the center of the window, it can quickly and effectively remove scattered noise points, but change the value of effective pixels in the depth image in most cases.In other words, the mean filtering algorithm changes the real value of the distance from the object to the camera, and can not retain the depth information of the object edge in the depth image, resulting in edge blur. Based on the traditional mean filtering, my algorithm only replaces the invalid noise points in a certain size window, which can retain the effective depth information perfectly. According to the relationship between the pixel value and the mean value of all effective pixels in the window, my algorithm subtracts some pixel values that are easy to cause edge blur, at the same time, according to a certain calculation formula to give different weights to the effective pixel values participating in noise filtering, the results show that the filtered pixel value calculated by my algorithm can effectively improve the edge blur of depth image. The algorithm steps are:

(1)Take the point with the upper left coordinate of (0, 0) in depth image as the starting point, then adopt the 3 * 3 filtering window to traverse the entire 414 * 512 depth image according to the sliding rule that from top to bottom, from left to right, the row and column are increased by 3 for each slide;

(2)Make the number of pixels with non-zero pixel value in the current window m, and calculate the mean value of these m pixels;

(3)Compare the size relationship between each non-zero pixel value in the current window and the mean value calculated in the second step. If the absolute value of the difference between the two is greater than 0.5, the current pixel value is regarded as a point that has a great impact on the retention of edge information, so it is a invalid pixel value and does not participate in subsequent calculation. If the absolute value of the difference is less than or equal to 0.5, it will be regarded as the effective pixel value and participate in the subsequent calculation. I make the number of effective pixels n;

(4)Calculate the mean(α) and variance(σ) of n effective pixel values in the third step, then calculate the weight of the effective pixel value according to equation (4) and the pixel value of the final filled noise point according to equation (5):

$$\omega_i = \exp(-\frac{(v_i-\alpha)^2}{2\sigma^2}) \tag{4}$$

$$V_i = \sum_{i=1}^{m}(v_i \times \omega_i) \Big/ \sum_{i=1}^{m}\omega_i \tag{5}$$

Where v_i is the pixel value of the ith effective pixel selected in the third step, ω_i is the weight of the ith effective pixel value, V_i is the pixel value of the filled noise point;

(5)Adopt 5*5 filtering window instead of 3*3 window, and carry out another filtering process according to the above four steps.

ISSN: 2664-9640

3.2. Noise Point Removal Result and Analysis

In the process of noise points removal, I will first subtract the pixels with a difference of 0.5 from the mean value in the non-zero pixel value, which will not participate in the subsequent calculation. From equation (1), it can be calculated that the pixel value difference of 0.5 represents the actual distance difference of about 10mm, so that the pixel value in the window that may have a great impact on the loss of edge information can be excluded through my algorithm. In addition, after selecting the effective pixel value and calculating its mean and variance, I give different weights to each effective pixel according to the difference between it and the mean value. As can be seen from equations (2) and (3), the greater the difference between the effective pixel value and it's mean, the smaller its weight and the smaller its contribution to the final calculated pixel value, which can effectively solve the problem of edge information loss. To sum up, My algorithm can achieve the effect of noise point removal and edge protection.

My algorithm also has the following advantages: I only remove the invalid noise points in the window and do not change the size of the effective pixel value, retaining the original effective depth information of the depth image; When the sliding filter window is adopted, the distance added on the row and column each time is equal to the side length of the window, which ensures that all pixel values do not participate in the calculation repeatedly, so the operation efficiency of my algorithm is high; The 5*5 sliding window is used for the second filtering, which can remove more noise points in a larger range on the basis of 3*3 window. Figure 4 shows the image after noise point removal by my algorithm:

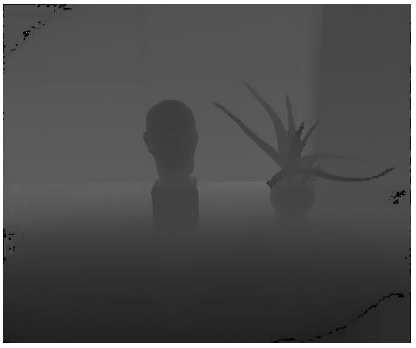


Figure 4: Noise point removal result

It can be seen from Figure 4 that my algorithm can effectively remove the scattered noise points in the depth image and the edge of the object is not smooth, which is more consistent with the actual situation, so I achieve a better effect of noise point removal and edge preservation. However, there are still some invalid areas with an area greater than 5 * 5 window in the depth image.

3.3. Final Result

For remaining invalid areas, I use a 5 * 5 sliding window to traverse the whole image according to the sliding rules from top to bottom, from left to right, and adding 5 to each row or column, using the nearest neighbor interpolation algorithm to fill the invalid region. In the meantime, the remaining white marker points in the process of hole repair are removed as noise points. Figure 5 shows the final processing results:

ISSN: 2664-9640

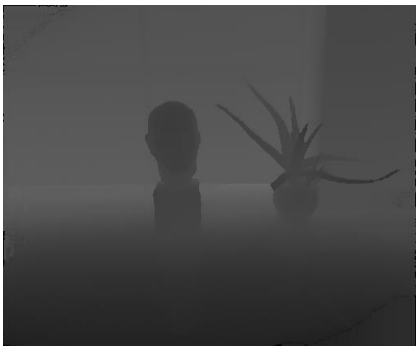


Figure 5: Final result

The depth image is first processed by threshold operation to make it easier for computer display. Then the nearest neighbor interpolation algorithm via diffusing from center to periphery is used to repair the hole and the improved mean filter algorithm with weight is used to remove noise points. Finally, the nearest neighbor interpolation algorithm is used to remove the remaining invalid areas and marked points, and the whole processing process of depth image is completed. Intuitively, my algorithm achieves a good image inpainting result.

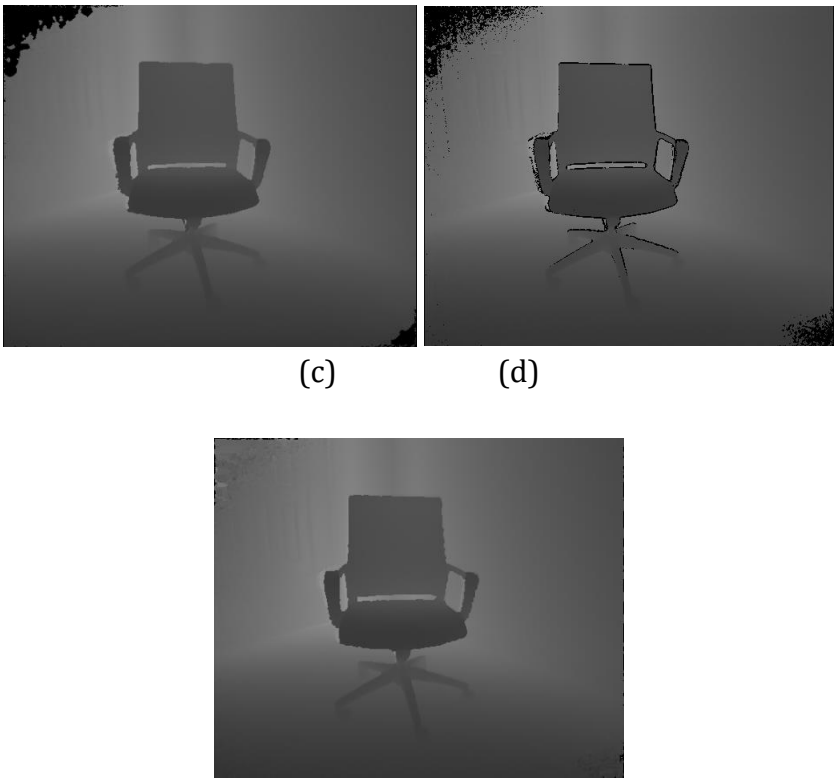
4. Experimental Results

In order to further verify the restoration effect of my algorithm on depth image, I use Kinect V2 camera to collect the real scene depth image, then take FMM, AMF, BF and our algorithm to process depth image separately. Firstly, we visually observe and compare the depth image restoration effect, and then quantitatively compare the number of point clouds and the running time of the algorithm. The experimental platform is: Windows 10 Operating System, AMD R5 5600H 3.30GHz Processor, Compilation environment is Visual Studio 2019, OpenCV4.1.1 Visual Library, PCL1.12.0 Point Cloud Library.

The first step is to observe and compare the depth image repair effect of each algorithm qualitatively. Since the 16 bit depth image collected by Kinect V2 cannot be displayed on the computer, I transform it into an 8-bit depth image to observe the processing results of each algorithm. When FMM selects different domain radius, its repair effect is different, so I select the domain radius as 25 after weighing the two factors of algorithm execution efficiency and repair effect. The minimum window side length of AMF algorithm is 3, the maximum length is 7, and the length increases by 2 each time. The pixel domain diameter of BF algorithm is taken as 55, the standard deviation of color space filter and filter in space coordinate are taken as 50. My algorithm combines the nearest neighbor interpolation algorithm via diffusing from center to periphery and improved mean filtering algorithm with weight. Figure 6 shows the original depth image of the real scene and the depth image repaired by each algorithm, Figure 7 shows a partial enlarged view:

(a) (b)

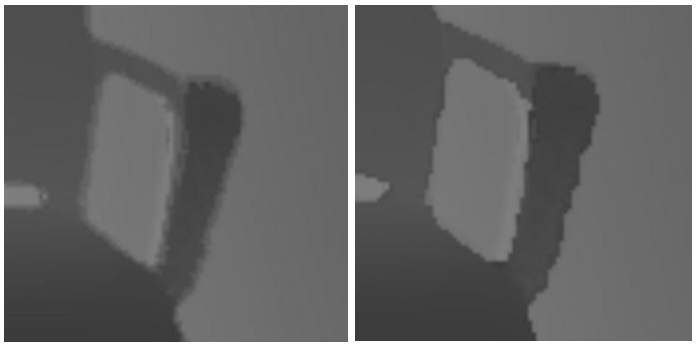
ISSN: 2664-9640



(c) (d)

(e)

Figure 6: Repair results:(a)Original depth image;(b)FMM;(c)AMF;(d)BF;(e)My algorithm



(a) (b)

Figure 7: Partial enlarged view:(a)FMM;(b)My algorithm

It can be seen from figure 6 that FMM can better realize the hole repair and noise points removal of depth image, but there are still some holes in the upper left corner that have not been repaired, and by observing the local enlarged view of figure 7, it can be found that the edge information retention degree is worse than my algorithm. AMF can remove noise points, but it can not completely repair large area holes. BF can not well complete the hole repair and noise points removal of our depth image.My algorithm can repair large area holes and remove noise points and retain the depth information of the object edge.

The second step is to quantitatively analyze the effect of depth image restoration. Convert the depth image into point cloud through Kinect V2 camera internal reference, and judge the depth image repair effect according to the number of point clouds. The more the number of point clouds, the better the repair effect.In addition, the running efficiency of the algorithm is reflected by the running time.Table 1 shows the number of point clouds and running time corresponding to each algorithm:

ISSN: 2664-9640

Table 1: Quantitative results

Different algorithms	Number of point clouds	Running time/ms
FMM	217079	575
AMF	212810	462
BF	211512	302
My algorithm	217019	91

We can see from table 1, the number of point clouds in FMM is similar to that in my algorithm, which is greater than AMF and BF, and my algorithm has the least running time. Through the comprehensive analysis of the experimental results of the first and second steps, it can be seen that my algorithm can well complete the depth image restoration and retain the object edge depth information, at the same time, the execution efficiency is high.

5. Conclusion

Aiming at large area holes and noise points in depth image, I propose a nearest neighbor interpolation algorithm that diffuses from the center to the periphery to repair holes and present an improved mean filtering algorithm with weight to remove noise points. The experimental results show that my algorithm can well complete the depth image restoration and retain the object edge depth information, the restoration effect is better than the traditional interpolation algorithm and filtering algorithm, and the execution efficiency has been significantly improved. My algorithm is not only applicable to Kinect V2 depth images, but also has application value for the repair of depth images collected by other devices, and it also has reference value for holes repair and noise points removal of other types of images.

Acknowledgements

This paper is the research result of the natural science foundation of Heilongjiang Province (LH2021F051).

References

[1] Kim Soo Kyun,Lee Chang Hee,Kim Sun Jeong,Song Chang Geun,Lee Jung. Implementation of Local Area VR Environment using Mobile HMD and Multiple Kinects[J]. Intelligent Automation and Soft Computing,2019,26(1).

[2] Xu Hantong,Xu Jiamin,Xu Weiwei. Survey of 3D modeling using depth cameras[J]. Virtual Reality & Intelligent Hardware,2019,1(5).

[3] Bhatt Meet,Joshi Hari,Vaghasiya Denil,Rohit R Parmar,Pradeep M Shah. Gesture-Based Robot Control with Kinect Sensor[J]. International Journal of Innovative Technology and Exploring Engineering (IJITEE),2020,9(7s).

[4] Alexandru Telea. An Image Inpainting Technique Based on the Fast Marching Method[J]. Journal of Graphics Tools,2004,9(1).

[5] Chan T F , Shen J . Nontexture Inpainting by Curvature-Driven Diffusions[J]. Journal of Visual Communication and Image Representation, 2001, 12(4):436-449.

[6] Yuan H D . Blind Forensics of Median Filtering in Digital Images[J]. IEEE Transactions on Information Forensics and Security, 2011, 6(4):1335-1345.

[7] Camplani M , Salgado L . Efficient Spatio-Temporal Hole Filling Strategy for Kinect Depth Maps[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2012, 8290:13.

ISSN: 2664-9640

[8] Le A , Jung S W , Won C . Directional Joint Bilateral Filter for Depth Images[J]. Sensors, 2014, 14(7):11362-78.

[9] Kim S W, Jung J Y, Lee S J, et al. Sensor fusion-based people counting system using the active appearance models[C]//2013 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2013: 65-66.