KINECT DEPTH STREAM PRE-PROCESSING FOR HAND GESTURE RECOGNITION

Muhammad Asad [*]

Dept. of Computer Science
City University London, United Kingdom

Charith Abhayaratne [†]

Dept. of Electronics & Electrical Engineering
University of Sheffield, United Kingdom

ABSTRACT

Over the recent years there has been growing interest to propose a robust and efficient hand gesture recognition (HGR) system, using real-time depth sensors like Microsoft Kinect. The performance of such HGR systems have been affected by the low resolution, noise and quantization error in the depth stream. In this paper, we propose a method to pre-process Kinect depth stream in order to overcome some of these limitations. The design approach utilizes the hand tracker from OpenNI SDK to perform distance invariant segmentation of hand region depth stream. This is followed by the construction of three different projections of hand in XY, ZX and ZY planes. These projections are then further enhanced using a combination of morphological closing and simple averaging based interpolation. The evaluation results show above 80% similarity with ground truth, and 1.45-5.35% increase in accuracy for gestures with recognition accuracy less than 90%.

***Index Terms*—** Kinect sensor, pre-processing, depth stream, hand gesture recognition, real-time

1. INTRODUCTION

With the recent introduction of Microsoft Kinect, there has been increasing interest in research community to use this inexpensive and powerful sensor in almost every domain in the field of Computer Vision. The main attraction of Kinect sensor in this field is the real-time provision of depth data. Combining this with the functionality of different Computer Vision libraries, a whole new category of applications has started to emerge [1–3].

Since Microsoft Kinect's introduction in 2010, there have been numerous attempts to use it to propose a number of HGR systems [4–13]. The depth stream acquired from Kinect sensor is low resolution (640x480) and contains random noise and quantization error [14]. This limits the operating range and functionality of most of the existing techniques, where some approaches also utilize RGB data stream to assist the depth stream based segmentation and HGR [5–10]. Another category of HGR systems utilize only the hand trajectory from depth stream to interpret gestures [11–13]. This paper proposes a Kinect depth stream pre-processing method for HGR, which extracts accurate 3D hand posture information and reduces quantization error.

A number of pre-processing methods for Kinect depth stream have been previously proposed. Most of these methods focus on de-noising [15–18] and filling the 'holes' discontinuity in the depth stream [19–21]. There are some methods which focus on increasing the resolution, hence reducing the quantization error [22, 23]. Most of these methods are complex for real-time execution and focus on the overall enhancement of the depth streams. There is still a need for a reliable and efficient Kinect depth stream pre-processing method for HGR.

In this paper we propose a Kinect depth stream pre-processing method for HGR systems. The proposed approach involves distance invariant segmentation of hand region from the depth stream (section 2). This segmented depth stream is used to construct projections of the hand region in three different planes. The quantization error in the extracted projections is reduced by using a combination of morphological operations and averaging based interpolation. Similarity measure and a neural network based ground truth model is used to evaluate the performance of this method (section 3).

2. PROPOSED DEPTH STREAM PRE-PROCESSING

The flowchart of the HGR system is presented in Fig. 1. This system consists of five parts. Distance invariant segmentation step involves segmenting the hand region using distance of the hand from the sensor. This is followed by projection extraction in which projections of the hand are extracted in XY, ZX, and ZY planes. Quantization error reduction is then performed and contour features of these projections are extracted. These features are used in the gesture recognition step, where a neural network based classifier recognizes each posture of the hand.

The proposed Kinect depth stream pre-processing approach consists of three main steps, which are (i) distance invariant segmentation, (ii) projection extraction and (iii) quantization error reduction. These methods are highlighted in Fig. 1 and discussed in detail below.

2.1. Distance Invariant Hand Segmentation

The method proposed in this step performs distance invariant segmentation on Kinect depth stream I to obtain hand region depth stream I_s by utilizing the hand tracker from OpenNI SDK. This hand tracker extracts a hand point $P = (P_x, P_y, P_z)$, defining the location of the hand in 3D coordinates, which is utilized throughout this paper. We collected a set of I with open hand pose at varying P_z. I_s was manually segmented from the collected set of I. For simplicity I_s is kept square in size, with side length S. Two sample frames with this hand pose at $P_z = 700$mm and 1700mm are shown in Fig. 2(a) and 2(c), respectively, in which the hand point location is marked with a white dot on the palm of the hand. It can be observed from these figures that the size of the hand region decreases with increased P_z. We use this inverse relationship between P_z and S along with their values extracted from the set of manually segmented I_s in the following equation to calculate the value of K.

$$S = \frac{K}{P_z} \quad . \tag{1}$$

[*] M. Asad was at the University of Sheffield and is now a PhD student at the City University London, UK muhammad.asad.2@city.ac.uk

[†] C. Abhayaratne is with the University of Sheffield, UK c.abhayaratne@sheffield.ac.uk

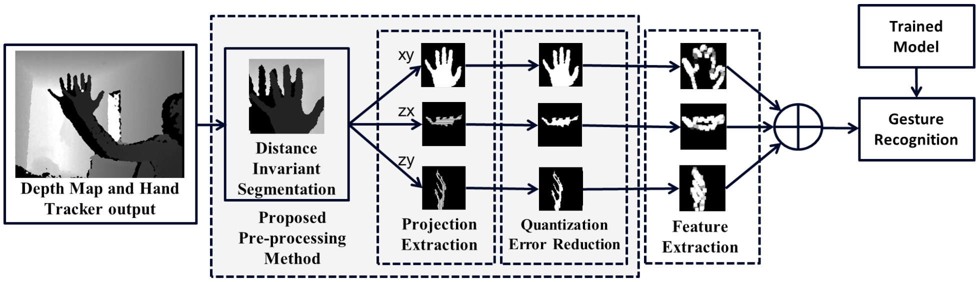
Depth Map and Hand Tracker output
Distance Invariant Segmentation
Proposed Pre-processing Method
xy
zx
zy
Projection Extraction
Quantization Error Reduction
Feature Extraction
Trained Model
Gesture Recognition

Fig. 1. Flowchart of the HGR system using our proposed pre-processing method.

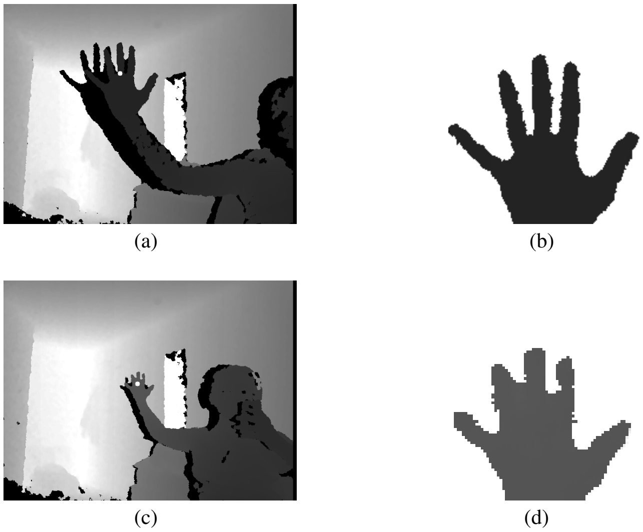

Fig. 2. Distance invariant segmentation at $P_z = 700mm$, (a) depth stream I, (b) Segmentated depth stream I_s with size 156x156 pixels, and at $P_z = 1700mm$, (c) depth stream I, (b) depth stream I_s with size 64x64 pixels.

The value of K comes out to be 108000. This value is independent of the sensor used, however it is dependent on the output resolution of I. We use 640x480 pixel I for our implementation. When using a different resolution I, the value for K can be easily recalculated using the above described method. Equation 1 is then used to segment the hand region in all three axis, taking hand point (P_x, P_y, P_z) as the origin for segmentation. Fig. 2(b) and 2(d) show the output of this distance invariant segmentation for Fig. 2(a) and 2(c) respectively.

2.2. Projection Extraction

The proposed approach uses a method similar to the construction of action graphs in [24] and silhouette image based 3D matching in [25]. The contribution of this work is to use a series of projections to define a 3D hand gesture. This involves the use of a segmented hand region depth stream to construct three silhouette projections in XY, ZX and ZY plane. These projections provide the front, top and side view of the hand respectively. Each of these projections are stored as a mask (m_{xy}, m_{zx}, m_{zy}) which are used in the next step for quantization error reduction and are given by

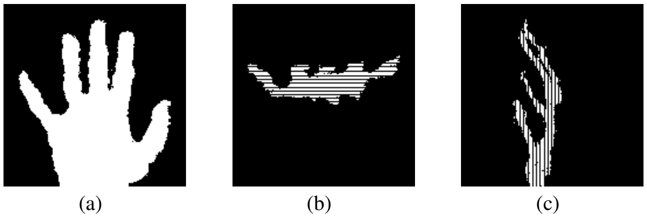

Fig. 3. Projections extracted from segmented hand region depth stream, (a) m_{xy} projection (front view), (b) m_{zx} projection (top view), (c) m_{zy} Projection (side view).

$$m_{xy}(x,y) = \begin{cases} 1, & \text{if } I_s(x,y) > 0 \\ 0, & \text{if } I_s(x,y) = 0 \end{cases}, \tag{2}$$

$$m_{zx}(z - c_z, x) = \begin{cases} 1, & \text{if } I_s(x,y) > 0 \\ 0, & \text{if } I_s(x,y) = 0 \end{cases}, \tag{3}$$

$$m_{zy}(z - c_z, y) = \begin{cases} 1, & \text{if } I_s(x,y) > 0 \\ 0, & \text{if } I_s(x,y) = 0 \end{cases}, \tag{4}$$

where $I_s(x,y)$ is the segmented depth stream containing depth value z at each location (x,y) and c_z is a shifting offset defined by $P_z - \frac{S}{2}$. The extracted projections are normalized to 64x64 pixels as each projection has different size due to dynamic segmentation size from previous step. Fig. 3 shows the constructed projections of hand region for an open hand pose, which includes noise and quantization errors. The next subsection introduces steps to reduce the quantization error.

2.3. Quantization Error Reduction

In [14], it was found that depth stream contains random noise and quantization error which increases quadratically with increased distance from the sensor. The depth stream is quantized for discrete values of x and y which gives rise to discontinuities when extracting ZX and ZY projection as shown in Fig. 4. The proposed method for this quantization error reduction consists of two different techniques which, when combined, are able to reduce these errors. These approaches are explained below.

2.3.1. *Morphological Filtering*

The first approach used to reduce the quantization error is to perform morphological closing operation using special structuring elements.

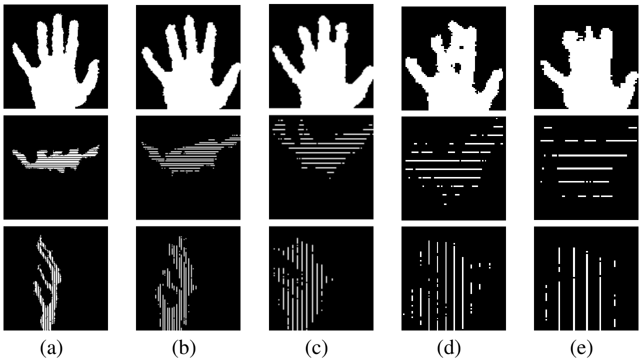

Fig. 4. Projections without quantization error reduction showing m_{xy} (top row), m_{zx} (middle row) and m_{zy} (bottom row) at $P_z =$ (a) 700mm, (b) 950mm, (c) 1200mm, (d) 1450mm, (e) 1700mm.

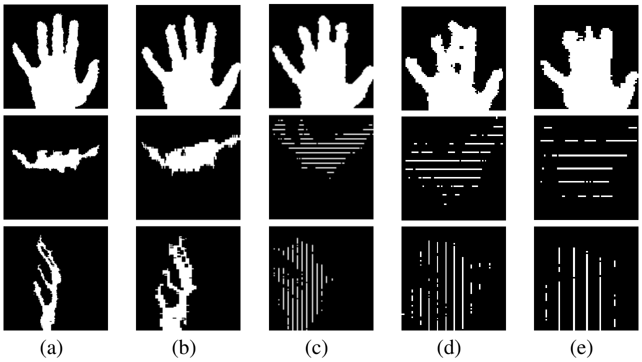

Fig. 5. Projections with morphological closing based quantization error reduction showing m_{xy} (top row), m'_{zx} (middle row) and m'_{zy} (bottom row) at $P_z =$ (a) 700mm, (b) 950mm, (c) 1200mm, (d) 1450mm, (e) 1700mm.

This approach uses two types of structuring elements, depending on the orientation of quantization error gap. These structuring elements are

$$V_{SE} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad (5) \qquad H_{SE} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad (6)$$

where V_{SE} and H_{SE} correspond to a vertical and horizontal line structuring elements respectively. The selection of these structuring elements for each projection is based on the orientation of discontinuities in that particular projection. The output m'_{zx} and m'_{zy} of this step is given by

$$m'_{zx} = (m_{zx} \oplus V_{SE}) \ominus V_{SE}, \quad (7)$$

$$m'_{zy} = (m_{zy} \oplus H_{SE}) \ominus H_{SE}, \quad (8)$$

where $\oplus$ is morphological dilation and $\ominus$ is morphological erosion. These output masks are shown in Fig. 5. It can be observed from this figure that the morphological closing operation only works for $P_z \leq 950mm$, which is due to lower magnitude of discontinuities at less distances. Therefore, we introduce an interpolation step before these operations to make them work with $P_z > 950mm$. This interpolation step is explained in detail in the next subsection.

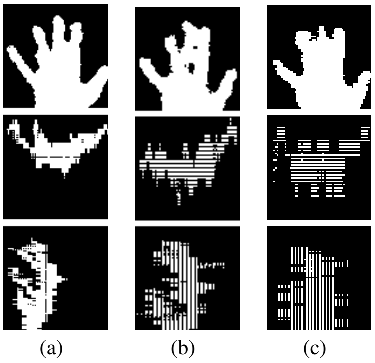

Fig. 6. Projections with interpolation based quantization error reduction: m_{xy} (top row), m_{zx} (middle row) and m_{zy} (bottom row) at $P_z =$ (a) 1200mm, (b) 1450mm, (c) 1700mm.

2.3.2. *Interpolation*

To reduce the quantization error for $P_z > 950mm$, we introduce a simple averaging based interpolation step. This step is further divided into two parts which are (i) searching for the regions with discontinuities, and (ii) interpolating them using a simple average of previous and next existing values. The method involving detection of these discontinued regions works by traversing through the projection masks in the opposite direction to the discontinuities. The location of the first value found is stored as $(prev_x, prev_y)$. These values are used along with the immediate next existing line coordinates $(next_x, next_y)$ to reconstruct the region in between. The reconstruction step involves finding the midpoint of both line coordinates and setting its value to interpolate the region. This reconstruction process is repeated recursively until the number of specified recursion steps is achieved. This interpolation method is applied throughout the projections. The equation for the average based interpolation operation is given by

$$m(\frac{prev_x + next_x}{2}, \frac{prev_y + next_y}{2}) = 1, \quad (9)$$

where $m = \{m_{zx}, m_{zy}\}$. The computation power required by this step is directly proportional to the size of the input projection and the number of recursive steps. This makes it computationally expensive to fill large quantization error gaps, *i.e.*, quantization error at greater P_z. To utilize these quantization error reduction techniques efficiently, our system uses interpolation steps only for $P_z > 950mm$. Furthermore, less number of recursive steps are performed, which are sufficient to reconstruct projection masks when followed by morphological closing operations. Fig. 6 shows the output, where the number of recursion steps performed is two. The overall output of the proposed method is shown in Fig. 7, combining both morphological filtering and interpolation steps. It can be observed that the reconstructed projections have increasing random noise with increasing distance P_z. Therefore, we impose a range limitation of distance $P_z \leq 1700mm$ on our proposed method.

3. EXPERIMENTAL RESULTS AND ANALYSIS

The choice of dataset collected to evaluate our proposed approach holds significance in showing the improved performance in HGR effectively. We divide a horizontal swipe gestures into 4 different stages and call them gesture stages. These stages are shown as projections in Fig. 9. From this figure, the significance of these projections is evident in defining the 3D posture of the four gesture stages.

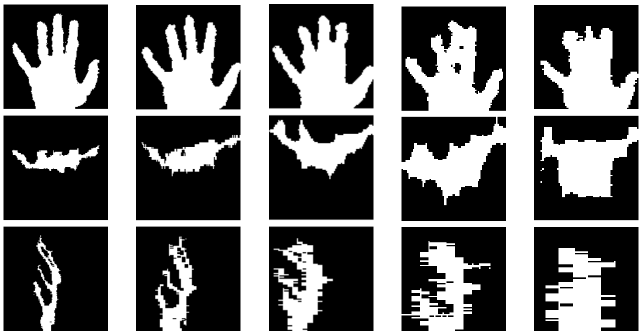
(a) (b) (c) (d) (e)

Fig. 7. Projections with the proposed quantization error reduction step: M_{xy} (top row), M_{zx} (middle row) and M_{zy} (bottom row) at P_z = (a) 700mm, (b) 950mm, (c) 1200mm, (d) 1450mm, (e) 1700mm.

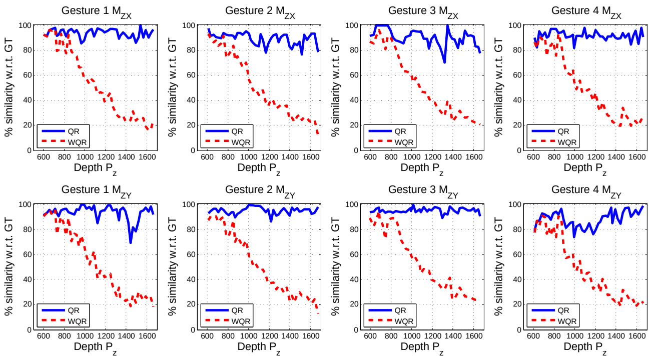

Fig. 8. Similarity measure of WQR and QR w.r.t. GT for four gesture stages: M_{zx} (top row) and m_{zy} (bottom row)

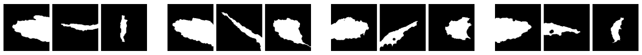
(a) (b) (c) (d)

Fig. 9. Projections of four gesture stages used for evaluation showing M_{xy} (left), M_{zx} (center) and M_{zy} (right) (a) stage 1, (b) stage 2, (c) stage 3, (d) stage 4.

When looking at these gestures from front, *i.e.*, XY projection, they are almost similar in shape. However together with other two projections, classifying them becomes an easy task.

We captured a dataset containing 2000 samples for each of the four gesture stages , with P_z varying between 600mm to 1700mm. Slight variations in the hand postures were also made while recording, making the dataset challenging for HGR. Two sets of projections based features were extracted, one using the proposed preprocessing quantization reduction method (QR), while other without quantization reduction (WQR). Evaluation was done using two different methods to quantify both the percentage increase in relevant pixel count and the accuracy of the HGR. These experimental techniques are presented in the subsections below, along with the analysis of the output.

Table 1. Evaluation Results.Key; QR - Pre-processed Depth stream with **Q**uantization **R**eduction; WQR - Pre-processed Depth stream **W**ithout **Q**uantization **R**eduction

Gesture	**Correct HGR**		**Accuracy (%)**	
	QR	WQR	QR	WQR
1	1972	1959	**98.60**	97.95
2	1985	1981	**99.25**	99.05
3	1664	1557	**83.20**	77.85
4	1763	1734	**88.15**	86.70

3.1. Similarity measure w.r.t. Ground Truth

The significance of this experimental technique is to quantify the increase in similarity of QR *w.r.t.* the ground truth (GT). To perform this experiment, we extracted 100 equally spaced gesture stages in the range $600 \leq P_z \leq 1700$ from QR and WQR of the captured dataset. The projections were manually edited for random error noise and quantization error to generate the GT. The percentage similarity of QR and WQR were calculated using the following formula

$$Similarity_{\ Two\ Images} = \frac{same\ white\ pixels}{total\ white\ pixels\ in\ GT}. \qquad (10)$$

The results from this similarity measure are presented in Fig. 8, with percentage similarity measure plotted against P_z for ZX and ZY projection of all four gesture stages. It can be observed that with increased P_z there is a significant decrease in similarity of WQR and GT. Whereas, using our quantization reduction technique the similarity measure remains above 80% for most values of P_z.

3.2. Accuracy of the HGR

To further validate the proposed quantization reduction approach, a neural network model for all four gesture stages was constructed using hand labelled ground-truth projections. Gesture stages from each of the two sets of projections (QR and WQR) were recognized using this model. The performance of the output was evaluated by calculating the percentage accuracy using the correct recognition results. The results are presented in Table 1. It can be observed that using quantization reduction steps encouraging results are achieved, particularly for gestures with accuracy less than 90%. This is due to the reason that in most cases the recognition rate is affected by high quantization error in ZX and ZY projections. Reconstructing these projections bring them closer in similarity to the actual ground truth projections, hence accounting for significant increase in performance.

4. CONCLUSION

In this paper, a Kinect depth stream pre-processing method for HGR related application was proposed. This method used a distance invariant segmentation step, which utilized the distance of hand from the sensor to segment only the hand region. The segmented hand region was used to construct projections in three different planes. The limitation of quantization error was overcome to some extent by using a combination of morphological closing operations and a simple averaging based interpolation technique. The proposed approach was evaluated using a similarity measure w.r.t. GT and a neural network based GT model. The results show improvement in accuracy of HGR by 0.2-5.35 %. Similarity measure shows that the proposed method is able to reconstruct the projections significantly, with above 80 % similarity with GT.

5. REFERENCES

[1] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[2] RA El-laithy, J. Huang, and M. Yeh, "Study on the use of microsoft kinect for robotics applications," in *IEEE/ION Position Location and Navigation Symposium (PLANS), 2012*, pp. 1280–1288.

[3] L. Cruz, D. Lucio, and L. Velho, "Kinect and rgbd images: Challenges and applications," *Conference on Graphics, Patterns and Images (SIBGRAPI), Tutorial*, 2012.

[4] J. Suarez and R.R. Murphy, "Hand gesture recognition with depth images: A review," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2012*. IEEE, 2012, pp. 411–417.

[5] M. Tang, "Recognizing hand gestures with microsofts kinect," *Palo Alto: Department of Electrical Engineering of Stanford University:[sn]*, 2011.

[6] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss, "Real-time 3d hand gesture interaction with a robot for understanding directions from humans," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2011*. IEEE, 2011, pp. 357–362.

[7] M. Van den Bergh and L. Van Gool, "Combining rgb and tof cameras for real-time 3d hand gesture interaction," in *IEEE Workshop on Applications of Computer Vision (WACV), 2011*. IEEE, 2011, pp. 66–72.

[8] T. Hongyong and Y. Youling, "Finger tracking and gesture recognition with kinect," in *IEEE 12th International Conference on Computer and Information Technology (CIT), 2012*. IEEE, 2012, pp. 214–218.

[9] Z. Ren, J. Meng, and J. Yuan, "Depth camera based hand gesture recognition and its applications in human-computer-interaction," in *2011 8th International Conference on Information, Communications and Signal Processing (ICICS)*. IEEE, 2011, pp. 1–5.

[10] Z. Ren, J. Meng, J. Yuan, and Z. Zhang, "Robust hand gesture recognition with kinect sensor," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 759–760.

[11] V. Tam and Ling-Shan Li, "Integrating the kinect camera, gesture recognition and mobile devices for interactive discussion," in *IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), 2012*, aug. 2012, pp. H4C–11 –H4C–13.

[12] C. Yang, Y. Jang, J. Beh, D. Han, and H. Ko, "Gesture recognition using depth-based hand tracking for contactless controller application," in *IEEE International Conference on Consumer Electronics (ICCE), 2012*. IEEE, 2012, pp. 297–298.

[13] D. Ramirez-Giraldo, S. Molina-Giraldo, A.M. Alvarez-Meza, G. Daza-Santacoloma, and G. Castellanos-Dominguez, "Kernel based hand gesture recognition using kinect sensor," in *IEEE Symposium of Image, Signal Processing, and Artificial Vision (STSIVA), 2012*. IEEE, 2012, pp. 158–161.

[14] 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.

[15] G.Y. Lee and Y.S. Ho, "Depth map boundary enhancement using random walk," in *International Workshop on Advanced Image Technology*, 2012, pp. 118–121.

[16] S. Milani and G. Calvagno, "Joint denoising and interpolation of depth maps for ms kinect sensors," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012*. IEEE, 2012, pp. 797–800.

[17] K.R. Lee, R. Khoshabeh, and T. Nguyen, "Sampling-based robust multi-lateral filter for depth enhancement," pp. 1124–1128, 2012.

[18] K. Essmaeel, L. Gallo, E. Damiani, G. De Pietro, and A. Dipanda, "Temporal denoising of kinect depth data," in *Eighth International Conference on Signal Image Technology and Internet Based Systems (SITIS), 2012*. IEEE, 2012, pp. 47–52.

[19] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov, "Temporal filtering for depth maps generated by kinect depth camera," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*. IEEE, 2011, pp. 1–4.

[20] K.R. Vijayanagar, M. Loghman, and J. Kim, "Refinement of depth maps generated by low-cost depth sensors," in *International SoC Design Conference (ISOCC), 2012*. IEEE, 2012, pp. 355–358.

[21] N.E. Yang, Y.G. Kim, and R.H. Park, "Depth hole filling using the depth distribution of neighboring regions of depth holes in the kinect sensor," in *IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC), 2012*. IEEE, 2012, pp. 658–661.

[22] C. Yang, H. Liu, R. Green, E. Song, and M. Fan, "Super-resolution for depth maps," pp. 224–230, 2011.

[23] M. Tallón, S.D. Babacan, J. Mateos, M.N. Do, R. Molina, A.K. Katsaggelos, and IL Urbana-Champaign, "Upsampling and denoising of depth maps via joint-segmentation," pp. 245–249, 2012.

[24] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010*. IEEE, 2010, pp. 9–14.

[25] Y. Okada, "3d model matching based on silhouette image matching," *Proc. of CSCC2002 (Recent Advances in Circuits, Systems and Signal Processing), WSEAS Press*, p. 3804385, 2002.