A Computationally Efficient Denoising and Hole-Filling Method for Depth Image Enhancement

Suolan Liu[a,b], Chen Chen[b], Nasser Kehtarnavaz[b]
[a]Changzhou University, Jiangsu, China;
[b]University of Texas at Dallas, Richardson, TX, USA

ABSTRACT

Depth maps captured by Kinect depth cameras are being widely used for 3D action recognition. However, such images often appear noisy and contain missing pixels or black holes. This paper presents a computationally efficient method for both denoising and hole-filling in depth images. The denoising is achieved by utilizing a combination of Gaussian kernel filtering and anisotropic filtering. The hole-filling is achieved by utilizing a combination of morphological filtering and zero block filtering. Experimental results using the publicly available datasets are provided indicating the superiority of the developed method in terms of both depth error and computational efficiency compared to three existing methods.

Keywords: Computationally efficient depth image enhancement, depth image denoising, depth image hole filling

1. INTRODUCTION

In the last few years, there has been a considerable increase in research works related to 3D action recognition. Some prominent applications of action recognition include intelligent surveillance, human-computer interaction and video analytics, e.g. [1-3]. Since the release of the Microsoft Kinect depth camera, the use of depth maps extracted from depth images has been growing for human action recognition, e.g. [4-6]. Features extracted from depth maps, such as histogram of oriented gradients (HOG) and histogram of optical flow (HOF) have been employed to recognize different actions [7-10]. However, depth maps provided by the Kinect depth camera are often noisy due to imperfections associated with the Kinect infrared light reflections. In addition, they exhibit missing pixels (i.e., pixels without any depth value which appear as black holes in depth maps), see Fig.1. The noise and holes can greatly affect the feature extraction outcome [8, 9, 11] and in turn the performance of action recognition. The noise-reduction and hole-filling enhancement algorithms presented in this paper are intended to serve as a pre-processing step for action recognition systems that use the Kinect depth camera.

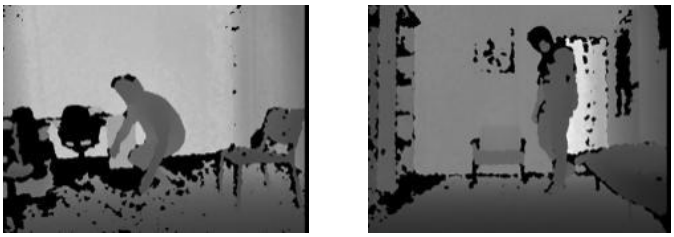

Fig.1 - Example depth images captured by a Kinect depth camera exhibiting depth imperfections (noise and black holes)

A number of methods have been proposed for noise smoothing and hole filling in depth images. Le et al. [12] proposed an adaptive directional filter by which depth pixels were classified into four groups: non-hole/non-edge, non-hole/edge, hole/non-edge, and hole/edge. In their method, color images were used to locate edge pixels in depth images. Tomasi et al. [13] used bilateral filtering (BF) [24] to denoise depth images. To fill holes while preserving edges, Camplani and Salgado [14] iteratively applied a joint bilateral filter (JBF) [12], which is a popular color-guided filtering method, and reported good performance for hole-filling [15]. However, it is noted that JBF does not perform well around

Real-Time Image and Video Processing 2016, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn, Proc. of SPIE Vol. 9897, 98970V · © 2016 SPIE
CCC code: 0277-786X/16/$18 · doi: 10.1117/12.2230495

Downloaded From: http://spiedigitallibrary.org/ on 05/04/2016 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

depth discontinuities where the foreground and background exhibit similar colors. Jung et al. [16] proposed a modified version of the joint trilateral filter (JTF) by using both depth and color pixels to estimate a filter kernel and by assuming the presence of no holes. Liu et al. [17] employed an energy minimization method with a regularization term to fill the missing regions and remove the noise in depth images. The linear regression model utilized in their method is based on both depth values and pixel colors. The examination of the above previous methods indicate that these methods are primarily based on different types of filters to smooth noise in depth images and to fill holes by using *color* images to guide the process. In other words, the previously developed methods are color-guided. In this paper, no color information is utilized leading to a computationally efficient solution.

Noting the computational limitations of the exiting methods in terms of the utilization of color or skeleton information, this paper provides a depth image recovery method that does not rely on any color image guidance [18], or skeleton information [26]. Denoising and hole-filling are performed purely based on depth images themselves and no other information is assumed to be available. For denoising, a discriminant approach is utilized to distinguish noise and non-noise depth pixels via gradient magnitude and orientation. For hole-filling, a zero block filter is utilized to fill them. The details of the developed method are mentioned next.

2. DEVELOPED DENOISING AND HOLE-FILLING METHOD

There exist several causes of noisy pixels and black holes in depth images as described in [4]. Noisy pixels are generated mostly due to background discontinuities at the contours of objects and the limitations of the sensor hardware. Holes are caused mostly by the infrared light reflectivity of different materials, fast movements, porous surfaces, and other similar effects.

2.1 Denoising

The noise generated by the Kinect depth camera is normally less distinguishable from the noise generated by movements, which can be effectively removed using a smoothing filter [4]. Therefore, a Gaussian kernel filter is first used here to smooth out the noise caused by movements. For other types of noises, a different approach needs to be considered to smooth them out while preserving the movement information. Motivated by the effectiveness of anisotropic theory in image filtering [19] and the work in [27], where anisotropic filtering is used for improving detection of object contours and region boundaries in natural scenes, and as well as a similar concept in in [9] and [28], an anisotropic filter is used here for noise discrimination.

For a point (x, y), let us consider the neighborhood $(x-a, y-b)_{o<|a|,|b|\le r}$, where r indicates the size of a local window centered at (x, y). A gradient orientation discriminant is defined as follows:

$$q_\sigma(x, y, x-a, y-b) = \left|\cos(\theta_\sigma(x, y) - \theta_\sigma(x-a, y-b))\right| \tag{1}$$

where σ denotes a scale size, and $\theta_\sigma(x, y)$ gradient orientation. If a point in a neighboring region has the same orientation as that of (x, y), it is expected to have an inhibitory effect. The strength value of $p_\sigma(x, y)$ for the point (x, y) is defined as follows:

$$p_\sigma(x, y) = \sum_a \sum_b M_\sigma(x-a, y-b) q_\sigma(x, y, x-a, y-b) \tag{2}$$

where M_σ denotes gradient magnitude. Next, a term for removing noise which uses a weighting factor λ to balance the effect of $M_\sigma(x, y)$ and $p_\sigma(x, y)$ is considered, that is

$$\begin{aligned} S_\sigma(x, y) &= F(M_\sigma(x, y) - \lambda p_\sigma(x, y)) \\ &= \begin{cases} 0 & , \text{if } M_\sigma(x, y) \le \lambda p_\sigma(x, y) \\ M_\sigma(x, y), & else \end{cases} \end{aligned} \tag{3}$$

As a result, if there is no noisy points in a neighboring region, the response will return the gradient magnitude $M_\sigma(x, y)$; otherwise, if there are noisy points in a neighboring region, the response will lower the influence of the gradient magnitude and smoothen out this point.

Downloaded From: http://spiedigitallibrary.org/ on 05/04/2016 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

2.2. Hole-filling

In depth maps, holes appear randomly at any place such as human bodies, walls, floor, door, shelf, bed, desk, chairs, etc. Some of these holes are small and isolated, but others are large and connected. For small holes, a morphological hole-filling operator is applied to fill them, that is a morphological closing operation with a 5×5 mask as experimentally obtained in [12]. As illustrated in Fig.2, holes are filled in the red highlighted rectangular regions by the morphological hole-filling operator. However, such operators do not perform well in the green rectangular regions.

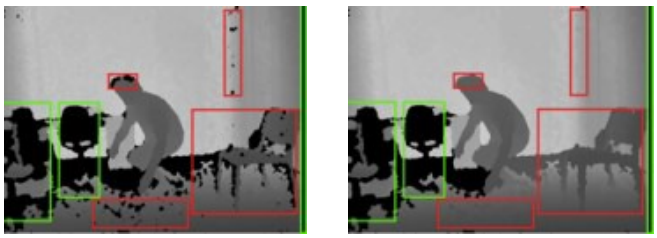

Fig.2 - Raw depth map (left) and hole filling result based on morphological filtering (right)

Here, a zero block filter is utilized to fill any remaining holes. This approach firstly searches zero pixels and labels them as holes. If $f(x,y)=0$, the pixel or point (x,y) is considered to be a hole. A small local window $(x \pm s, y \pm t)_{0<s,t\leq k}$ is defined on the point (x,y) with k denoting the window size. This pixel is then filled according to its neighboring pixels. If the pixel values of its neighboring pixels are all equal to 0, its value is not changed. Otherwise, its value is replaced by the maximum value of the neighbors as follows:

$$f(x,y) = \begin{cases} 0, & if \quad f(x \pm s, y \pm t) = 0 \\ \max\{f(x \pm s, y \pm t)\}, & else \end{cases} \tag{4}$$

where $0 < s,t \leq k$. The pseudocode of this computationally efficient hole-filling approach is provided in Fig.3.

Algorithm 1 Pseudocode for zero block filter mask(ZBFM)

```
Require: A depth image (M×N);
         A filter mask sized k×k; each element value is zero;
Ensure: fill zero fixels
for i=1 → M do
  for j=1 → N do
    if f(x,y)=0 then        // search every zero pixels f(x,y);
       filter_mask(s,t) ← neighbor pixels value of f(x,y) ;    // 0 < s,t ≤ k
       // search nonzero pixels in filter_mask;
       if search result is Null, then
          f(x,y)=0;
       else  search the maximal value m of all elements in filter_mask;
             replace zero pixels with m ;
       end if
    end if
  end for
end for
```

Fig.3 - Pseudocode of the hole-filling approach

Downloaded From: http://spiedigitallibrary.org/ on 05/04/2016 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

3. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the results of the experimentations conducted are presented to show the performance of the developed depth map recovery method. These results correspond to the two publicly available depth datasets: the UR fall detection dataset [22] and the Middlebury dataset [23].

3.1 Parameters setting

The outcome of our method is influenced by these four parameters: Gaussian scale size σ, window size *r*, balance factor λ and zero filter block size k. Appropriate values of these parameters were determined by the experimentations described next.

For a quantitative performance comparison, the Middlebury dataset [23] was utilized noting that this dataset provides the groundtruth disparity maps. Following the same experimental setting discussed in [17], [20], [21] and [25], depth images were generated by randomly removing some valid pixels and adding Gaussian white noise to the disparity maps. Example test images with randomly selected marked areas are shown in Fig.4. The average root-mean-squared error (RMSE) was computed for all the images in the dataset.

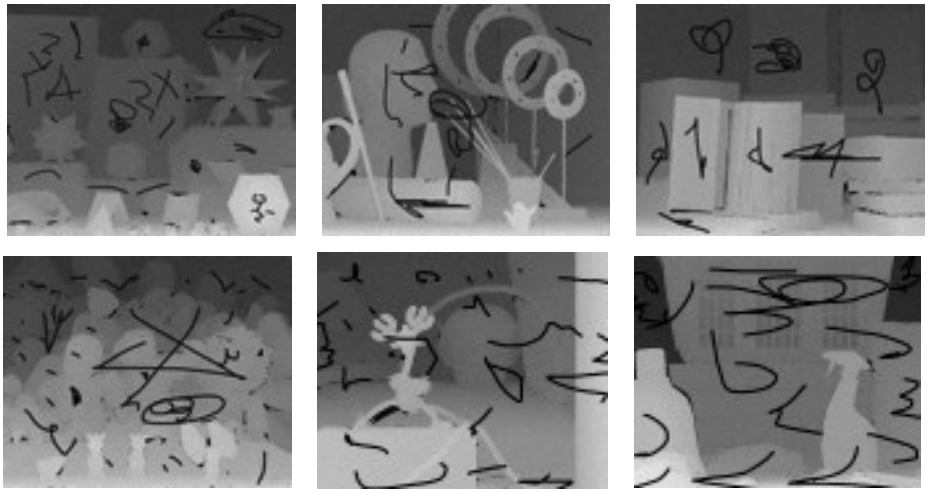

Fig.4 - Depth maps of Moebius, Art, Book, Dolls, Reindeer and Laundry (from left to right and top to bottom)

First, the effect of different balance factor λ was examined. As noted in [4], the scale size $\sigma = 5$ was considered with the denoising window size $w = 5$. Both RMSE and visual quality were examined. As shown in Fig.5, when λ was changed from 0.1 to 5.5, the average RMSE reached its minimum value at $\lambda = 2.5$. Fig.6 shows a closeup part of the image 'Moebius' exhibiting that when λ =2.5, the best visual quality was obtained. However, when λ =5.0, the denoised image became too blurry. By considering small values of λ, pixels were slightly impacted by their neighboring pixels based on Equations (2) and (3). That is, a large amount of the noise was preserved, leading to high RMSE values. On the other hand, while considering large values of λ, pixels were highly impacted by their neighboring pixels with a greater possibility of mistakenly taken as noise. As a result, a large amount of non-noise was smoothed out as noise and over-smoothing occurred on the images, which led to high RMSE values.

In another experiment, the filter block sizes of 3, 5, 7, 9, 11 and 21 were selected to examine the effect of the filter block size on the hole-filing performance. As can be seen from Fig.7, when the block size was set to 5, the average RMSE value reached a minimum. When the size was changed from 7 to 21, the average RMSE remained the same or about 4. Thus, the block size of 5 was selected as the zero pixel filter block size.

Moreover, RMSE values were computed to compare the performance of our method and the three existing methods [17,18]. As can be seen from Table 1 and Fig.8, our method outperformed these existing methods in terms of both RMSE and visual quality.

Downloaded From: http://spiedigitallibrary.org/ on 05/04/2016 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

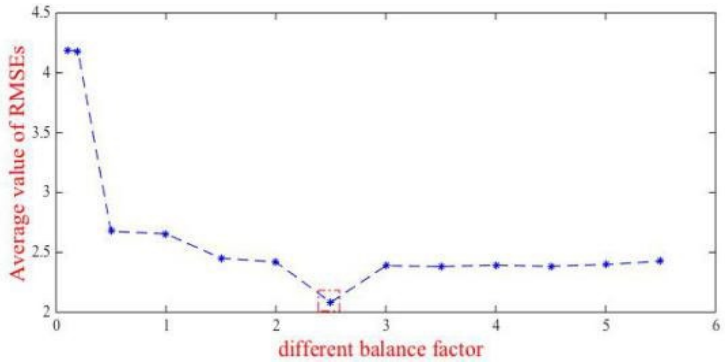

Fig.5 - Denoised RMSE vs. different balance factor λ

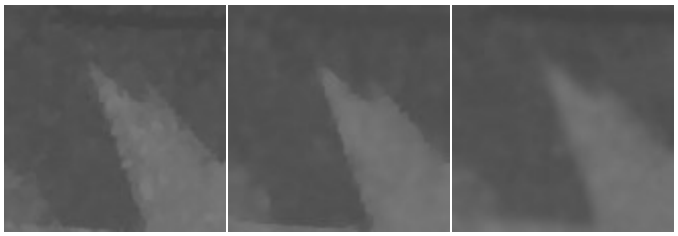

Fig.6 - Closeup of denoised outcome λ=1.0, 2.5, 5.0 (from left to right)

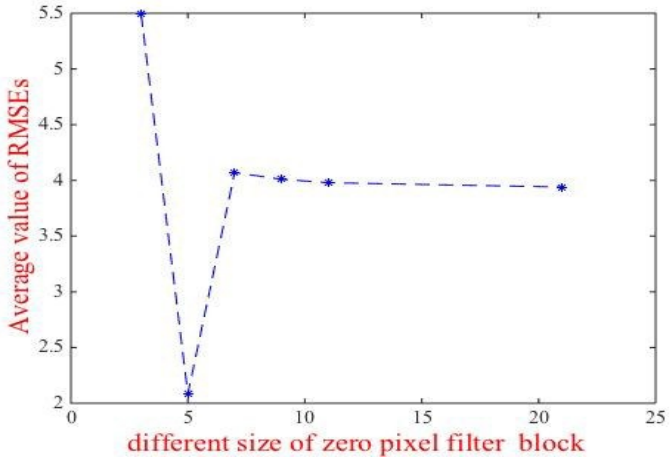

Fig.7 - Hole-filling RMSE vs. different sizes of zero pixel filter block k

Table.1 - RMSE of different methods on the Middlebury dataset

RMSE	Moebius	Art	Book	Dolls	Reindeer	Laundry	Average
JBF [12]	0.8290	2.7612	2.1040	3.0521	3.2886	4.2117	**2.7078**
FMM [17]	1.5139	3.2381	2.3935	3.1674	2.6838	3.8309	**2.8046**
GFMM [25]	1.1433	2.7381	2.2242	3.0026	2.5772	3.7786	**2.5773**
Ours	**0.6550**	**2.3768**	**2.0294**	**2.8347**	**2.2010**	**3.4906**	**2.2645**

Downloaded From: http://spiedigitallibrary.org/ on 05/04/2016 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

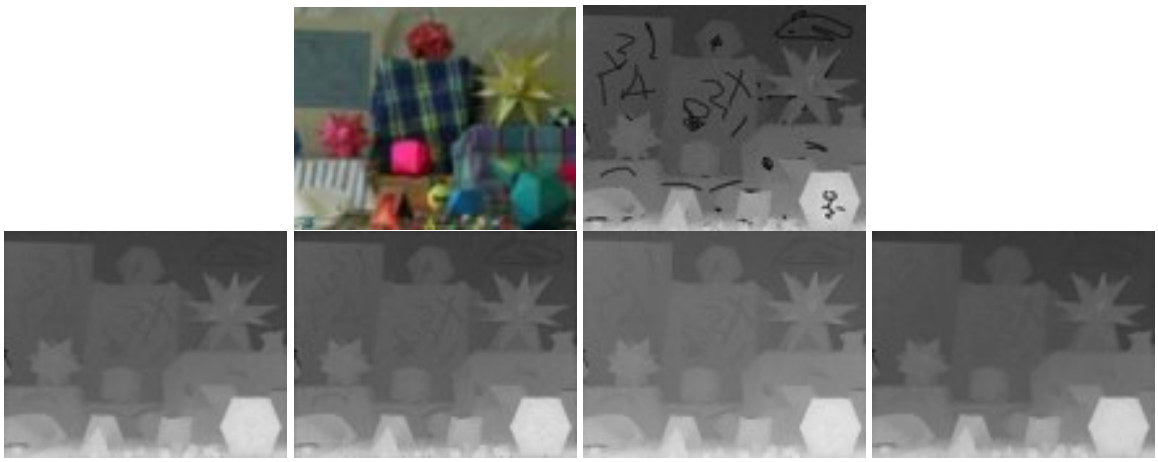

Fig.8 - Visual comparison on sample image 'Moebius' from the Middlebury dataset: top row displays the original color image and depth map; bottom row displays the recovered depth maps using from left to right JBF, FMM, GFMM, and the developed method.

In Fig.8, the top row displays the original color image and the depth map. The second row from left to right displays the results of JBF, FMM, GFMM and our method.

3.2 Results on depth images from Kinect depth camera

Our method was further applied to depth images captured by a Kinect depth camera. The UR fall detection dataset [22] contains raw depth images that were captured by a Kinect depth camera. The developed method was compared with three existing depth image enhancement methods (JBF [12], FMM [20], GFMM [25]). The depth sequences examined were of size 640 × 480 pixels. Two human activity sequences were randomly selected. Sequence 1 included frames #059, #099, and #139, and sequence 2 included frames #079 and #113. The results obtained are shown in Figs.9 through 11. In these figures, from left to right, the color images, the raw depth images, and the results of JBF, FMM, GFMM and our method are displayed.

Fig.9 - Three sample frames of sequence 1: each row from left to right corresponds to the original color image, original depth map, recovered depth maps using JBF, FMM, GFMM, and the developed method

Downloaded From: http://spiedigitallibrary.org/ on 05/04/2016 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

Fig.10 - Two sample frames of sequence 2: each row from left to right corresponds to the original color image, original depth map, recovered depth maps using JBF, FMM, GFMM, and the developed method

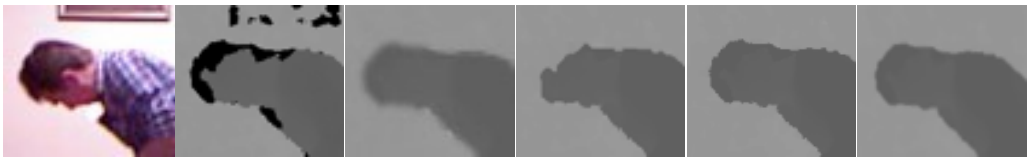

Fig.11 - Closeup of the second row frame in sequence 2 appearing in Fig.10

Color images were used in the JBF method [12, 13], and the weights were calculated based on a filter window of size 5□5. The standard deviations of this filter were 5 and 0.3. This method performed well in the body area but many large holes remained and the edges appeared blurry. For the FMM method, a binary image mask was produced by performing contour detection on the color images. The number of iteration was set to 100. Some parts of the body such as the head appeared over-smoothed and the holes were not completely filled. The results of the GFMM method appeared fine on the body but the background objects became oversmoothed, such as the two chairs on the left. In our method, the following parameter settings as described earlier were considered: $\sigma = 5$, $w = 5$, λ=1.5. As can be seen from these figures, all the holes in the images got filled and the noise was also reduced while preserving the boundaries. In the test sequence 2, there were many holes in the depth images and the upper part of the body was quite noisy. The comparison results showed that our method achieved improved performance compared to the other methods. More importantly, our method did not use any color image making it computationally efficient.

3.3 Computational efficiency

This subsection includes the computational complexity of the developed method. For denoising, the computational complexity for the Gaussian kernel filter is $O(2MNr_1)$, where MN denotes the depth image size & r_1 the filtering window size, and the computational complexity of the anisotropic filter is $O(MN\log(MN)) + O(MNr_2^2)$, where r_2 denotes the local window size. For hole filling, the computational complexity for the morphological closing filter used is $O(25MN)$, and the computational complexity of the zero block filter is $O(MNr_3^2)$, where r_3 denotes the zero mask size.

The computational efficiency or real-time aspect of our method for depth image enhancement is further presented here. For a depth image, the denoising is performed first and the hole-filling is done right after the denoising. Note that there are four major processing components based on four types of filters: Gaussian kernel filtering, anisotropic filtering, morphological filtering, and zero block filtering. The average processing time of each filtering component for the UR fall detection dataset is listed in Table 2. All the experiments were carried out using MATLAB on a PC equipped with Intel Xeon 3.4GHz CPU with 16GB RAM. As noted in Table 2, our method provided a real-time depth video processing rate of 30 frames per second. A videoclip of the enhancement running in real-time can be viewed at http://www.utdallas.edu/~kehtar/DepthEnhanced.avi.

Downloaded From: http://spiedigitallibrary.org/ on 05/04/2016 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

Table.2 - Processing times associated with the filtering components of our method

Depth image enhancement		Average processing time (ms/frame)
Denoising	Gaussian kernel filtering	0. 94
	Anisotropic filtering	19.63
Hole filling	Morphological filtering	1.17
	Zero block filtering	1.09

4. CONCLUSION

In this paper, a method for depth image enhancement has been developed for the purpose of reducing noise and filling holes. The main attribute of the developed method is that, despite most of the existing methods for depth image enhancement, it does not utilize any color information to achieve the depth image or map recovery in a computationally efficient manner. The experimentations carried out on the publicly available depth image datasets have revealed that the developed method provides enhanced images that come closer to the groundtruth depth images compared to three existing methods while achieving real-time processing rates. This method can be deployed as a preprocessing step in action recognition systems that utilize depth images.

REFERENCES

[1] Chen, C., Liu, K., Jafari, R. and Kehtarnavaz, N., "Home-based senior fitness test measurement system using collaborative inertial and depth sensors," Proc. *Engineering in Medicine and Biology Society (EMBC),* 4135-4138 (2014).

[2] Chen, C., Kehtarnavaz, N. and Jafari, R., "A medication adherence monitoring system for pill bottles based on a wearable inertial sensor," Proc. *Engineering in Medicine and Biology Society (EMBC),* 4983-4986 (2014).

[3] Chen, C., Jafari, R. and Kehtarnavaz, N., "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Transactions on Human-Machine Systems*, *45*(1), 51-61 (2015).

[4] Chen, C., Jafari, R., and Kehtarnavaz, N., "A real-time human action recognition system using depth and inertial sensor fusion," *IEEE Sensors Journal*, *16*(3), 773-781 (2015).

[5] Chen, C., Hou, Z., Zhang, B., Jiang, J. and Yang, Y., "Gradient local auto-correlations and extreme learning machine for depth-based activity recognition," In *Advances in Visual Computing*, Springer International Publishing, 613-623, (2015).

[6] Chen, C., Jafari, R., & Kehtarnavaz, N., "Action recognition from depth sequences using depth motion maps-based local binary patterns," Proc. *IEEE Winter Conference on Applications of Computer Vision*, 1092-1099 (2015).

[7] Ni, B., Moulin, P., Yang, X., and Yan, S., "Motion part regularization: improving action recognition via trajectory selection," Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 3698-3706 (2015).

[8]Abdul-Azim, H. A., and Hemayed, E. E., "Human action recognition using trajectory-based representation," *Egyptian Informatics Journal*, *16*(2), 187-198 (2015).

[9] Chakraborty, B., Holte, M. B., Moeslund, T. B., Gonzalez, J., and Roca, F. X., "A selective spatio-temporal interest point detector for human action recognition in complex scenes," Proc. *IEEE International Conference on Computer Vision*, 1776-1783 (2011).

[10] Laptev, I., Marszałek, M., Schmid, C., and Rozenfeld, B., "Learning realistic human actions from movies," Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 1-8 (2008).

[11] Saygili, G., van der Maaten, L., and Hendriks, E. A., "Hybrid kinect depth map refinement for transparent objects," Proc. *International Conference on Pattern Recognition*, 2751-2756 (2014).

[12] Le, A. V., Jung, S. W., and Won, C. S., "Directional joint bilateral filter for depth images," *Sensors*, *14*(7), 11362-11378 (2014).

[13] Tomasi, C., and Manduchi, R., "Bilateral filtering for gray and color images," Proc. *International Conference on Computer Vision*, 839-846 (1998).

[14] Camplani, M., and Salgado, L., "Efficient spatio-temporal hole filling strategy for kinect depth maps," Proc. *IS&T/SPIE Electronic Imaging*, 82900E-82900E (2012).

Downloaded From: http://spiedigitallibrary.org/ on 05/04/2016 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

[15] Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., and Toyama, K., "Digital photography with flash and no-flash image pairs," In *ACM transactions on graphics*, *23*(3), 664-672 (2004).
[16] Jung, S. W., "Enhancement of image and depth map using adaptive joint trilateral filter," *IEEE Transactions on Circuits and Systems for Video Technology*, *23*(2), 258-269 (2013).
[17] Liu, S., Wang, Y., Wang, J., Wang, H., Zhang, J., and Pan, C., "Kinect depth restoration via energy minimization with TV21 regularization," Proc. *IEEE International Conference on Image Processing*, 724-724 (2013).
[18] Chen, C., Cai, J., Zheng, J., Cham, T. J., and Shi, G., "A color-guided, region-adaptive and depth-selective unified framework for Kinect depth recovery," Proc. *IEEE 15th International Workshop on Multimedia Signal Processing*, 7-12 (2013).
[19] https://en.wikipedia.org/wiki/Anisotropic_filtering.
[20] Telea, A., "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, *9*(1), 23-34 (2004).
[21] Yang, J., Ye, X., Li, K., Hou, C., and Wang, Y., "Color-guided depth recovery from RGB-D data using an adaptive autoregressive model," *IEEE Transactions on Image Processing*, *23*(8), 3443-3458 (2014).
[22] Kwolek, B., & Kepski, M., "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Computer methods and programs in biomedicine*, *117*(3), 489-501 (2014).
[23] http://vision.middlebury.edu/stereo/data/scenes2005/
[24] He, K., Sun, J., and Tang, X., "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(6), 1397-1409 (2013).
[25] Liu, J., Gong, X., and Liu, J., "Guided inpainting and filtering for Kinect depth maps," Proc. *International Conference on Pattern Recognition*, 2055-2058 (2012).
[26] Xia, L., Chen, C. C., and Aggarwal, J. K., "View invariant human action recognition using histograms of 3d joints," Proc. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 20-27 (2012).
[27] Grigorescu, C., Petkov, N., and Westenberg, M. A., "Contour and boundary detection improved by surround suppression of texture edges," *Image and Vision Computing*, *22*(8), 609-622 (2004).
[28] Fan, J., Wu, Y., and Dai, S., "Discriminative spatial attention for robust tracking," Proc. *European Conference on Computer Vision*, 480-493 (2010).

Downloaded From: http://spiedigitallibrary.org/ on 05/04/2016 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx