Completion and Denoising of Depth Data from Kinect

Rishab Shah

December 14, 2017

1 Introduction

Microsoft's Kinect is a motion sensing device that uses an infrared projector and camera to generate a mesh depicting scene depth using structured light. It finds applications not only in video gaming but also in research such as robotics [6], human motion analysis [11], human-computer interaction [5], human pose recognition [13]. Kinect depth is measured by the generation of structured infrared light speckles in a specific pattern. The first limitation is that it is usually assumed that sensing is performed in an indoor environment. Furthermore, errors in depth arise because of occlusions in the scene, loss of data due to transparent objects, sensor noise, or insufficient resolution.

The computation generally required for depth processing of the Kinect data is high due to large amounts of raw information captured in real time. The range information is usually represented by 8-bit or 16-bit data images called depth maps. The depth maps are computed on regular 2D sampling grids, but each pixel represents the distance from the camera to a target object surface rather than color information. Building 3D maps of environments is a vital task in navigation, semantic mapping, manipulation and path planning. Tasks to study human motion analysis have also employed the use of depth maps, more commonly in multi-view settings, i.e, with multiple depth sensors.

The goal of this project is to complete and denoise the depth map obtained by a Kinect sensor.
We begin with a single-view variational formulation that re-implements current state-of-the-art techniques and extend them with novel types of regularization. We use a simplified computation for an objective function which is shown by Li and Osher in [9].

2 Literature review

There have been numerous methods[10] to solve traditional color image denoising some of which include the median filters, bilateral filters [18] and non-local means filter [1]. The bilateral filter,also called feature preserving filter, denoises an image using a kernel function from the differences of pixel positions and intensities between an interest pixel and its neighbors.

Several variations of the bilateral filter such as the joint bilateral filter [8] and the cross bilateral filter [4] which are well known edge preserving filters have also been used to denoise images [20]. Variational methods that employ the use of optical flow estimation have been used to perform denoising of depth maps [16]. Another common method used is hole-filling or in-painting method to remove artifacts in the depth maps using an exemplar-based method, which also reduces some spatial noise [2]. One other hole-filling technique uses background subtraction to smooth the data, where background subtraction ”extracts” moving objects from a sequence of frames [15]. Other methods include algorithms that use training data and prior knowledge of the environment to perform smoothing on the depth frames, for example MRFs (Markov Random Fields) [12], or adaptive color-guided auto-regression kernel filters [19] where the corresponding RGB image is used to predict weights for the depth map based on the non-similarity of both the depth map and the color image.

3 Problem

A captured depth map is a degraded version of the underlying ground truth. Denote by u the depth map of a scene. Our aim is to obtain a high quality joint 3-D reconstruction from the low quality depth maps in a consolidated framework that is robust to surface curvatures. We assume the sensors are subject to Gaussian noise and optimize over the depth values of the sensor. It is an iterative process that smoothens the 2D representation of the scene. The observation model assumes the depth maps to be fronto-parallel planes and the sensors are calibrated intrinsically. Another assumption we make is that it is a piecewise planar model. We propose a single view variational objective function as follows:

$$\min_{\hat{u}} \sum_{i,j}(\|u_{i,j} - \hat{u}_{i,j}\|^2 + \lambda \sum_{(k,l)\in\mathcal{N}_{(i,j)}} \xi_{(i,j)}|\hat{u}_{(i,j)} - \hat{u}_{(k,l)}|) \tag{1}$$

where,

$$\xi_{(i,j),(k,l)} = \frac{1}{\mathcal{E} + \|\nabla I_{RGB(i,j),(k,l)}\|^2} \tag{2}$$

is the regularization. λ and ξ are the weights that influence the median. The weights $\xi_{(i,j)}$ are used to preserve the discontinuities near the edges. These are weighed according to where the edges are found in the color image corresponding to the depth map. $\xi_{(i,j)}$ is inversely proportional to the gradient of values in the neighborhood for pixel (i, j). We penalize pixels and use priors to take care of inconsistencies where data is missing from the edges in the depth map. Li and Osher [9] provided a formal connection between median filtering and Eq. (1) and from the approach of [16], who show that minimizing this equation is related to a median computation and this simplifies computations.

$$\hat{u}^{(q+1)}_{(i,j)} = median(Neighbors^{(q)} \cup Data) \tag{3}$$

where

$$Neighbors^{(q)} = \{\hat{u}^{(q)}_{(k,l)}\} \; \forall(k,l) \in \mathcal{N}_{(i,j)} \tag{4}$$

and $\hat{u}^{(0)} = u$ as well as

$$Data = \{u_{(i,j)}, u_{(i,j)} \pm \lambda, u_{(i,j)} \pm 2\lambda \ldots, u_{(i,j)} \pm \frac{|\,\mathcal{N}_{(i,j)}\,|}{2}\lambda\} \tag{5}$$

with “q” being the number of iterations on the depth map. The set of data values is balanced with an equal number of elements on either side of the value $u_{(i,j)}$ and that information about the data term is included through $u_{(i,j)}$. Repeated application of Eq. (3) converges rapidly [9]. Observe that, as λ increases, the weighted data values on either side of $u_{(i,j)}$ move away from the values of Neighbors and cancel each other out.

3.1 Current Work

The work is done using ROS (Robot Operating System), a framework that uses operating system’s file system, user interface and programming utilities. It is used mainly for its modularity, resource handling and interoperability with various platforms. All coding is performed using C++. We have performed the denoising of a single depth map with the weight of 1 applied to the color image that corresponds to the depth map. Not that Eq. (1) can be minimized with repeated applications of Eq. (3); we use this approach with 5 iterations. We use a second-order neighborhood of size (3x3) to perform the median filtering. We perform 10 steps of alternating optimization and change λ and $\mathcal{E}$ from 10^{-4} to 10^2 and 10^{-2} to 10 respectively. In the end, we take $\hat{u}$ as the final depth map estimate and the parameters for our case to be $\lambda = 10$ and $\mathcal{E} = 0.1$.

We use ground truth images from the Middlebury dataset [7], to obtain objective results. We synthetically degrade these ground truth images to emulate Kinect-like problems with the depth maps. A conventional edge detector algorithm, the Canny edge detector [3], to remove 9 pixels along the edges and then introduce uniform holes of 4 pixels over the image to remove 60% of the pixels to degrade the image. We show the Structural Similiarity Index (SSIM), which is a measure to compare errors between images, for various methods and values. The algorithm takes around 1-2.5 seconds to denoise 1 frame on a 2.8 GHz Intel i7-5700HQ processor. In real-time applications, this is really slow but the code can be optimized further to use multi-threading techniques and speed up the algorithm.

We compared it to other methods in image processing such as morphological operations (Dilation and Erosion), and Inpainting [17]. Morphological image processing is a collection of non-linear operations based on the shape or the morphology of features in the image. Morphological operations can also be applied to grayscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest. Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. More information on grayscale morphology can be found in [14]. The inpainting is based on the paper [17]. It is based on Fast Marching Method. Consider a region in the image to be inpainted. The algorithm starts from the boundary of this region and goes inside the region gradually filling everything in the boundary first. It takes a small neighborhood around the pixel on the neighborhood to be inpainted. This pixel is replaced by normalized weighted sum of all the known pixels in the neighborhood.

SSIM for variation in λ					
Images	10^{-4}	10^{-2}	10^{-1}	10	10^{2}
Art	0.636	0.649	0.756	0.791	0.744
Books	0.645	0.667	0.768	0.843	0.754
Dolls	0.6398	0.654	0.778	0.837	0.746
Laundry	0.65	0.681	0.783	0.824	0.758
Moebius	0.619	0.665	0.786	0.818	0.763
Reindeer	0.652	0.688	0.791	0.849	0.789

SSIM for variation in ξ and $\lambda = 10$					
Images	10^{-2}	$5 * 10^{-2}$	10^{-1}	1	10
Art	0.698	0.7233	0.855	0.758	0.628
Books	0.703	0.748	0.9218	0.763	0.652
Dolls	0.715	0.737	0.9121	0.767	0.654
Laundry	0.6869	0.7389	0.9057	0.788	0.665
Moebius	0.7235	0.749	0.9091	0.766	0.677
Reindeer	0.736	0.7436	0.9233	0.791	0.693

SSIM for different iterations using Li and Osher [9]					
Images	2	3	4	5	7
Art	0.034	0.38	0.754	0.855	0.698
Books	0.038	0.426	0.783	0.9218	0.704
Dolls	0.0399	0.401	0.776	0.9121	0.716
Laundry	0.045	0.411	0.779	0.9057	0.707
Moebius	0.0387	0.405	0.782	0.9091	0.725
Reindeer	0.0431	0.429	0.786	0.9233	0.7301

SSIM for Morphological operations with different kernel shapes and sizes and Inpainting							
Images	Rectangular, 3	Rectangular, 5	Cross, 3	Cross, 5	Cross, 7	Cross, 9	Telea [17]
Art	0.059	0.541	0.501	0.601	0.698	0.667	0.032
Books	0.0614	0.598	0.543	0.612	0.829	0.691	0.039
Dolls	0.0624	0.587	0.557	0.626	0.803	0.68	0.035
Laundry	0.0629	0.583	0.56	0.623	0.791	0.678	0.0417
Moebius	0.0635	0.599	0.555	0.638	0.798	0.683	0.0408
Reindeer	0.065	0.615	0.568	0.641	0.809	0.699	0.04

4 Conclusion and future work

We have discussed a method to denoise degraded depth maps using the results shown in [9] and compared different methodologies commonly employed in image processing. The results show that the objective shown in Eq. (1) can be shown to be a median computation and we make our output faster. In the future, we would like to improve the speed even further by optimizing the code. Also, we would like to try other regularizations and exploit the fact that in a general scene, the depth increases as you go from bottom to the top. Finally, we would also like to extend the system to a multi-view system and obtain a 3D reconstruction for the scene.

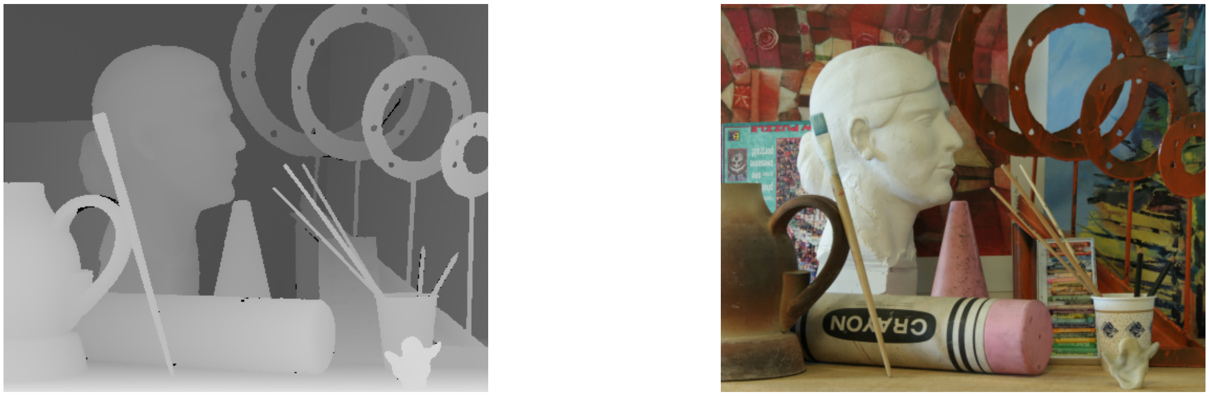

Figure 1: Ground-truth image and color image from the Middlebury dataset (Art).

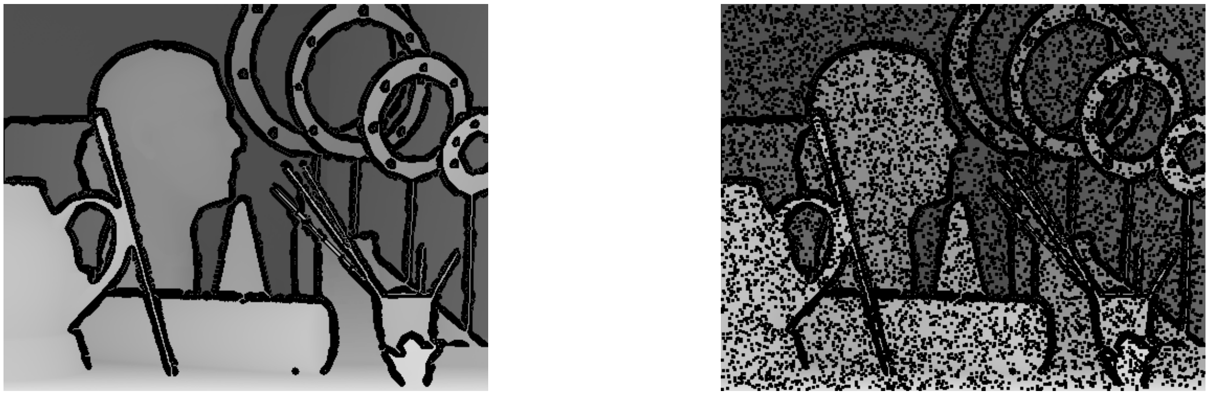

Figure 2: Degraded images (Art).

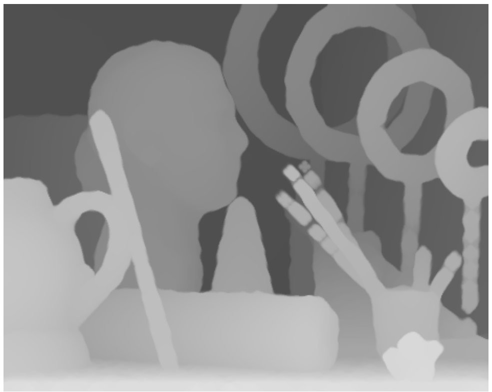

Figure 3: Denoised image without considering the weights in the color image(Art).

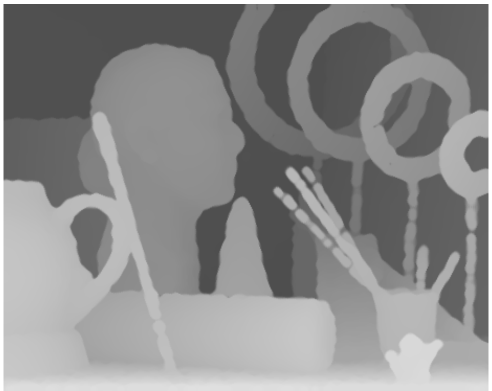

Figure 4: Denoised image (Art).

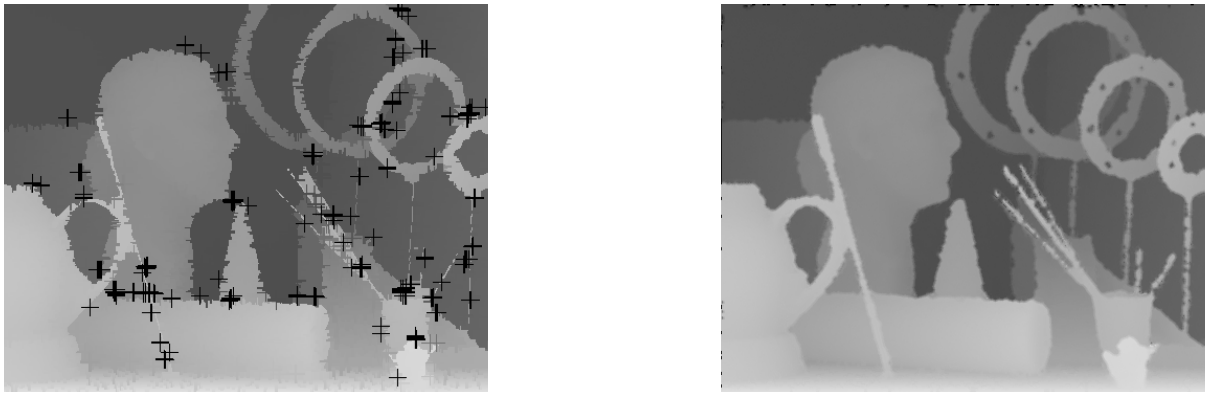

Figure 5: Denoised images using Morphological operations [Dilation and Erosion, Cross shaped kernel of size 7] (left) and Inpainting [Telea] (right) (Art).

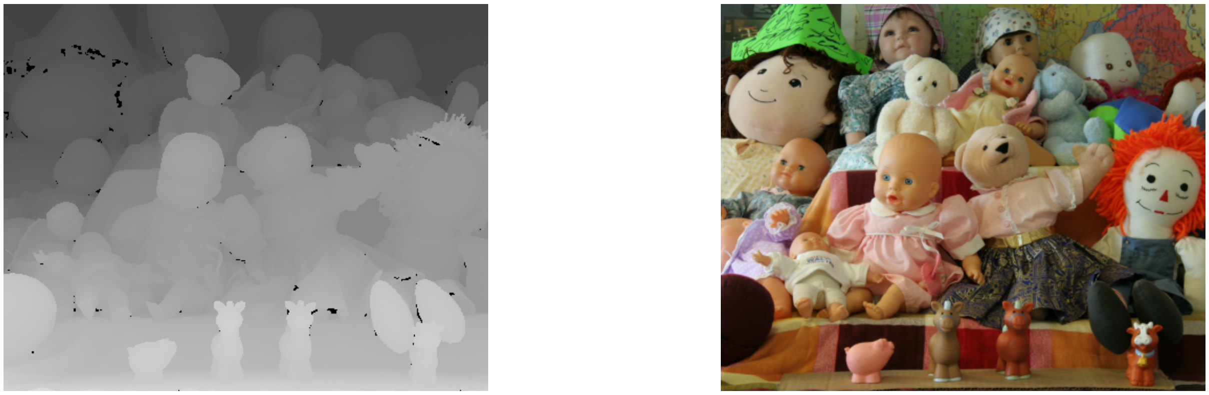

Figure 6: Ground-truth image and color image from the Middlebury dataset (Dolls).

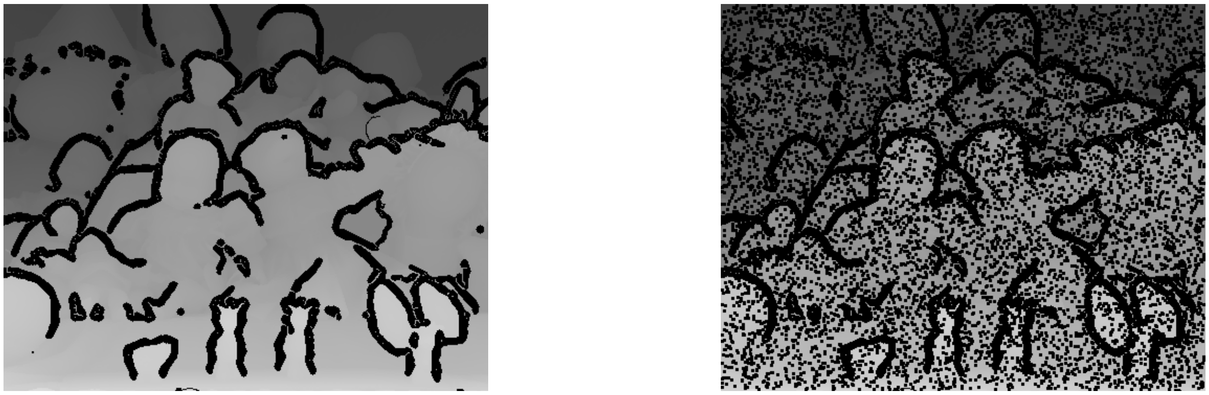

Figure 7: Degraded images (Dolls).

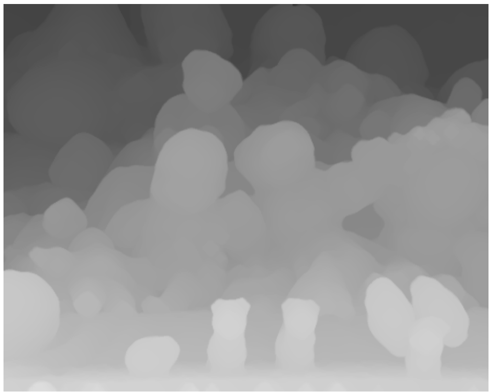

Figure 8: Denoised image without considering the weights in the color image(Dolls).

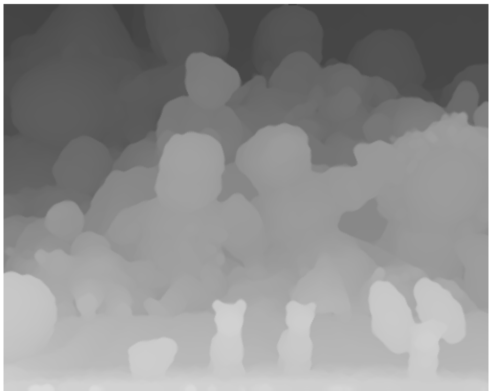

Figure 9: Denoised image (Dolls).

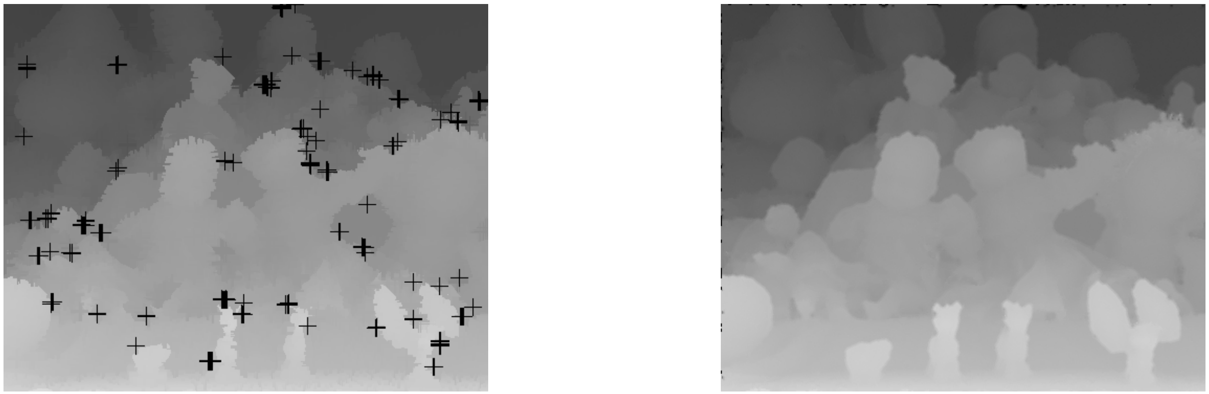

Figure 10: Denoised images using Morphological operations [Dilation and Erosion, Cross shaped kernel of size 7] (left) and Inpainting [Telea] (right) (Dolls).

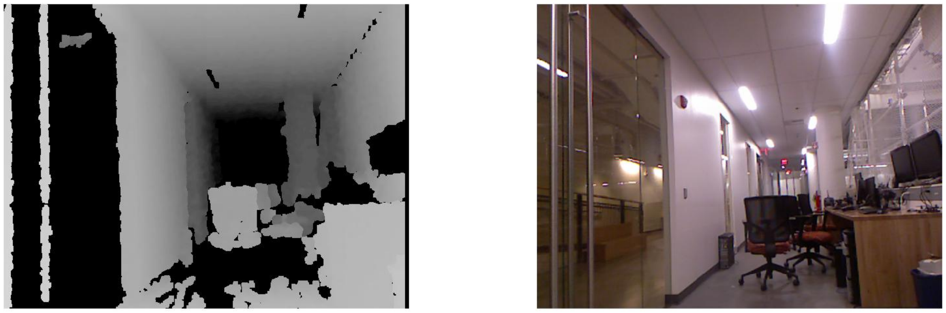

Figure 11: Depth map(left) and RGB (right) image.

Figure 12: Denoised depth map after 2 iterations (left) and 5 iterations (right).

References

[1] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[2] Massimo Camplani and Luis Salgado. Efficient spatio-temporal hole filling strategy for kinect depth maps. *Three-Dimensional Image Processing (3DIP) and Applications*, 8290, 2012.

[3] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[4] Laurent Caraffa, Jean-Philippe Tarel, and Pierre Charbonnier. The guided bilateral filter: When the joint/cross bilateral filter becomes robust. *IEEE Transactions on Image Processing*, 24(4):1199–1208, 2015.

[5] Luigi Gallo, Alessio Pierluigi Placitelli, and Mario Ciampi. Controller-free exploration of medical image data: Experiencing the kinect. In *Computer-based medical systems (CBMS), 2011 24th international symposium on*, pages 1–6. IEEE, 2011.

[6] Jan Hartmann, Dariush Forouher, Marek Litza, Jan Helge Klüssendorff, and Erik Maehle. Real-time visual slam using fastslam and the microsoft kinect camera. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–6. VDE, 2012.

[7] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[8] Anh Vu Le, Seung-Won Jung, and Chee Sun Won. Directional joint bilateral filter for depth images. *Sensors*, 14(7):11362–11378, 2014.

[9] Yingying Li, Stanley Osher, et al. A new median formula with applications to pde based denoising. *Communications in Mathematical Sciences*, 7(3):741–753, 2009.

[10] Bor-Shing Lin, Mei-Ju Su, Po-Hsun Cheng, Po-Jui Tseng, and Sao-Jie Chen. Temporal and spatial denoising of depth maps. *Sensors*, 15(8):18506–18525, 2015.

[11] Gemma S Parra-Dominguez, Babak Taati, and Alex Mihailidis. 3d human motion analysis to detect abnormal events on stairs. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 97–103. IEEE, 2012.

[12] Ju Shen and Sen-Ching S Cheung. Layer depth denoising and completion for structured-light rgb-d cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1187–1194, 2013.

[13] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[14] Stanley R Sternberg. Grayscale morphology. *Computer vision, graphics, and image processing*, 35(3):333–355, 1986.

[15] Martin Stommel, Michael Beetz, and Weiliang Xu. Inpainting of missing values in the kinect sensor's depth maps based on background estimates. *IEEE Sensors Journal*, 14(4):1107–1116, 2014.

[16] Deqing Sun, Stefan Roth, and Michael J. Black. Secrets of optical flow estimation and their principles. 2010.

[17] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004.

[18] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[19] Jingyu Yang, Xinchen Ye, Kun Li, Chunping Hou, and Yao Wang. Color-guided depth recovery from rgb-d data using an adaptive autoregressive model. *IEEE transactions on image processing*, 23(8):3443–3458, 2014.

[20] Qingxiong Yang, Ruigang Yang, James Davis, and David Nistér. Spatial-depth super resolution for range images. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.