Depth Image Enhancement using 1D Least Median of Squares

CONFERENCE PAPER · JUNE 2015

DOI: 10.13140/RG.2.1.3123.8244

4 AUTHORS, INCLUDING:

Abm Tariqul Islam
University of Rostock

Oliver Staadt
University of Rostock

CGI2015 manuscript No.
(will be inserted by the editor)

Depth Image Enhancement using 1D Least Median of Squares

ABM Tariqul Islam · Christian Scheel · Renato Pajarola · Oliver Staadt

Abstract We propose a new method to enhance the depth images from RGB-D sensors, such as Kinects, by filling the missing/invalid values which are reported by those sensors at certain pixels. We introduce a robust 1D least median of squares (1D LMedS) approach to accurately estimate the depth values of those invalid pixels. We use this approach for efficient traversal of each pixel's depth values over a sequence of frames and look for invalid depth values (considered as outliers), and finally, replace those values with stable valid depth values. Our approach solves the unstable nature of depth values in captured scenes that is perceived as flickering. Experimental results show good improvement both for static and moving parts of a scene.

Keywords Robust filtering · RGB-D sensor · 3D imaging · Virtual reality

1 Introduction

Microsoft's Kinect, as a low-cost depth camera for 3D content generation, has attracted considerable research attention in recent years. Quite a few areas in computer graphics, such as in human-computer interaction, virtual reality, remote collaboration, etc., have adapted Kinects as the primary acquisition device. Many of the works in those areas, such as in [5,7] etc., have used the 3D contents captured by Kinects. Kinects capture a scene with acceptable resolution and speed, but the generated depth data exhibits a considerable amount of artifacts (see Figure 1). So, the quality of 3D contents generated with Kinect's depth data is often too low for many applications such as telepresence, object tracking in a scene, forensic analysis of crime scene, etc. In case of telepresence, low quality 3D representation limits the sensation of a natural presence of remote users on the local site. Similarly, for tracking applications, the noise on an object's surface greatly deteriorates the tracking performance. Likewise, in forensic analysis, where every details of a crime scene is crucial, the part of a scene covered with noise makes the retrieval of correct information extremely difficult.

(a) (b)

Fig. 1 RGB-D frame from Kinect – (a) color image, (b) depth image. The depth image contains artifacts, such as black spots on the surface of the objects due to the invalid depth values reported by Kinect's depth sensor.

We propose an approach to enhance the depth frames of both static and dynamic parts of a scene. We extend the least median of squares (LMedS)[9] approach, a popular statistical regression technique for outlier handling. In our proposed 1D LMedS, for each pixel, we go through a sequence of frames, detect the invalid and unstable depth values for that pixel, identify the outliers among them by comparing them to a set of constraints (described in Section 4) and finally, replace them with valid stable depth values. Experimental results from our approach show good improvement in filling the invalid depth values and stabilizing the valid values for static and dynamic parts of a scene.

A.T. Islam, C. Scheel, O. Staadt
Visual Computing Lab, University of Rostock, Germany
E-mail: (tariqul.islam, christian.scheel, oliver.staadt)@uni-rostock.de

R. Pajarola
Visualization and MultiMedia Lab, University of Zürich, Switzerland
E-mail: pajarola@ifi.uzh.ch

2 Related work

Maimone et al., in [7], use a GPU-adapted median filter for enhancing the depth images. Their approach handles the issue of washed out edges of objects which occurs with standard median filter, and enhances the depth images by filling the black spots. But it occasionally reports wrong interpolated values for the invalid depth maps and does not always produce refined object boundary. Moreover, since it does not consider the fluctuation of valid depth values over the consecutive frames, the 3D representation still suffers from artifacts. Matyunin et al., in [8], uses motion information along with the median filter, which generates good results for some image areas, but suffers mainly from noisy object boundaries. Moreover, because of a large computational overhead, it might not be suitable for real-time applications. Amamra et al., in [1], use a GPU-adaptive Kalman filter, which produces smooth depth map with refined object edges in real-time. However, this method suffers when object's surface is sharply reflected. Another GPU-based approach [10], by Wasza et al., uses a combination of normalized convolution and guided filtering process to improve the depth map in real-time, but suffers from blurred borders for very noisy object borders.

Other works, such as [2,4], use bilateral filter to enhance the depth images. Chen et al., in [4], deploy a joint bilateral filter which uses color image as a guidance image to improve the depth image quality. Although, their method removes the holes efficiently, it cannot handle the region where color information is missing. Yang et al., in [11], have also obtained good depth frame enhancement by using such bilateral filter; but since it does not consider the temporal information, it is not suitable for processing dynamic scenes. Hui et al., in [6], use a combination of 2D median and bilateral filter where the bilateral filter removes the artifacts introduced by the median filter. Camplani et al., in [3], use an iterative approach of joint-bilateral filter which combines depth and color data by analyzing an edge-uncertainty map and pre-detected foreground regions. It performs well for the static scenes, but it is not suitable to work with dynamic scenes.

3 Least Median of Squares

The Least Median of Squares (LMedS) [9] is a statistical technique for robust regression of a p-dimensional sample set (x_i, y_i), which minimizes the median of the squared residuals, $r_i = y_i - x_{i1}\hat{\theta}_1 - \cdots + x_{ip}\hat{\theta}_p$ over the set of estimates, $\theta = (\theta_1, \theta_2, \cdots, \theta_p)^t$. LMedS is known to be very robust to false matches and outliers while fitting equations to observed data set. It estimates the data points by solving the nonlinear minimization problem i.e. r_is. The outliers from the data set can be filtered out by using a robust standard deviation $\hat{\sigma}$ which uses the minimum median of r_is. But, the

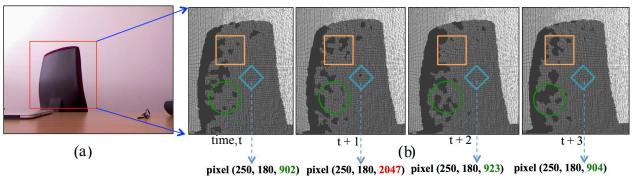

Fig. 2 RGB-D frame from Kinect with static objects – (a) color image, (b) depth images of four consecutive frames of square marked area of (a). The depth values of a pixel, inside the diamond marked area in (b) and shown by the four arrows, differ essentially over time and invalid value "2047" is also reported occasionally.

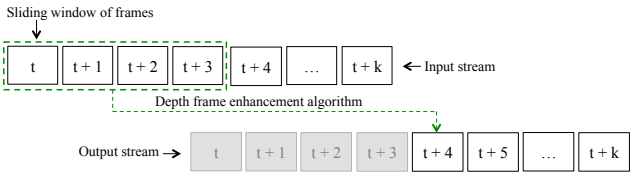

Fig. 3 Illustration of sliding window of frames in 1D LMedS.

LMedS estimator becomes computationally expensive when the dimension p of sample set(x_i, y_i) gets bigger. In our approach, we propose to use one dimensional LMedS which is a good fit for the nature of our problem domain and it is also computationally cheaper than higher dimensional data sets.

4 Proposed strategy

RGB-D sensors, such as Kinects, exhibit artifacts in the form of missing data and invalid values (see Figure 1). They exhibit such artifacts even if the objects in a scene remain static (see Figure 2); here, we can see that the valid depth values are showing an unstable nature for certain pixels. We take this unstable nature into consideration and apply 1D LMedS to replace the unstable and invalid values with valid stable depth values.

To enhance the depth frames by filling the invalid depth values, we introduce a 1D LMedS approach which is very robust to outliers. We apply the 1D LMedS on a per pixel basis over a sequence of frames inside a sliding window. Let us consider, for example, that we have k consecutive depth frames and we take n frames in a sliding window, see Figure 3. Due to Kinect's nature of depth value generation, we get different depth values d_is for any single pixel for the n previous consecutive frames inside the sliding window. We apply the 1D LMedS on this set of depth values which vary along the temporal domain. The name 1D LMedS (one dimensional LMedS) comes from the concept that the depth values d_is of a certain pixel (x,y) changes only in one dimension that is along the temporal domain i.e. over the consecutive frames. Our goal is to locate the invalid and unstable depth values (which we consider as outliers) from the set of depth values of each pixel and replace them with a stable valid depth value.

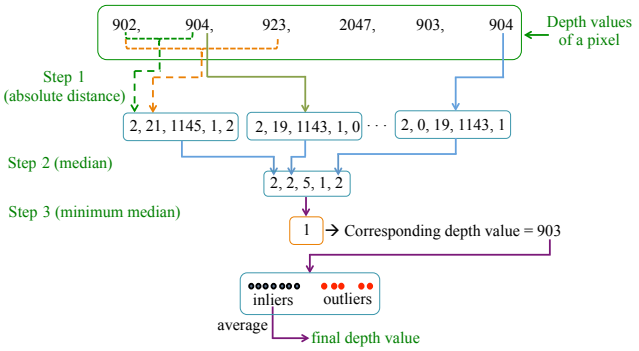

Fig. 4 Illustration of 1D LMedS.

Algorithm 1 Fast traversal for the depth values of one pixel

1: **procedure**
2: **Define** MAX 10000
3: `int idx, n` ▷ n = no. of frames in the window
4: `Float diff, depthVal[n]`
5: `Float preFinalDepthVal`
6: `min` ← `Float MAX`
7: **for** `int i = 0 to` $\frac{n-1}{2}$ **do**
8: `diff = depthVal`$\left[idx+\frac{n}{2}\right]$ − `depthVal`$[idx]$
9: **if** $diff < min$ **then**
10: $min = diff$
11: $idx \leftarrow i$
12: **end for**
13: `preFinalDepthVal =` $\frac{\texttt{depthVal}\left[i+\frac{n}{2}\right]-\texttt{depthVal}[i]}{2}$
14: **end procedure**

To filter out the outliers and fill the invalid depth values for a certain pixel, we calculate the minimum of the medians M of absolute residuals r_is for each depth value d_i of the set of depth values for n consecutive frames. In 1D LMedS, we use the absolute difference for getting the residuals r_is among the depth values d_is; in [9], r_i^2 is used to avoid negative values. The whole process of 1D LMedS, including the calculation steps of M, is depicted in Figure 4. Here, in step 1, for a set of depth values d_is, we calculate absolute difference for each depth value to the rest of the depth values for that pixel. So, we obtain a set of residuals for each depth values of d_is. In step 2, we calculate the medians from each set of residuals and finally, in step 3, we calculate the minimum of the medians M. Then, we retrieve the corresponding depth value whose index matches with the index of the minimum of the medians M. After that, we obtain inliers and outliers based on w_i of Equation 1 and finally, get the stable depth value by taking the average of the inliers. The necessary formulae, to obtain M, standard deviation $\hat{\sigma}$, and weight w_i which decides if a certain depth value is an inlier or outlier, are given in Equation 1:

$$\begin{aligned} M &= med_{i:1,\ldots,n} r_i\left(d_i\right) \\ \hat{\sigma} &= 1.4826\left[1+\frac{5}{n-1}\right]M \\ w_i &= \begin{cases} 1, & if \quad r \leq 2.5\hat{\sigma} \\ 0, & otherwise. \end{cases} \end{aligned} \tag{1}$$

The constants (1.4826, 5 and 2.5) of Equation 1 are explained in [9]. However, this process needs quite some time to traverse for each depth values of a pixel for these n frames. Therefore, we adapt a fast data traversal procedure to fit 1D LMedS. The fast data traversal procedure is stated in Algorithm 1. We apply this procedure for each of the depth values d_is of every pixels of the n frames. Algorithm 1 shows necessary pseudocode for fast traversal for one pixel over the n frames. Here, we sort the depth values d_is beforehand and split the traversal path to half of the window size and calculate differences among the depth values. We do the same for all the depth values inside the window and finally, take the minimum difference (see line 10 of Algorithm 1). Then we find the index `idx` of the minimum difference (see line-11) and retrieve the depth value which has this minimum index `idx` (see line 13). Based on this depth value and index `idx`, we calculate the $\hat{\sigma}$, and find the inliers and outliers according to w_i of Equation 1. We then take the average of the depth values of the inliers as the final depth value. In this way, 1D LMedS enhances a certain depth frame t_i by assigning stable valid depth values for each pixels of t_i. Then the sliding window moves one frame further to enhance the next frame t_{i+1}. It is worth to mention that, from the $(n+1)$th frame ($(t+4)$th frame on the output stream in Figure 3) we would observe the enhanced depth frame. For the $(n+2)$th frame ($(t+5)$th frame in Figure 3) and later ones, we just have to wait for one extra frame's calculation.

5 Results

We have applied 1D LMedS to both static and moving parts of the captured scene. 1D LMedS worked quite well with static parts of the scene and filled the missing/invalid depth values with valid ones (see Figure 5 (a) -(c)) and also made depth values more stable over consecutive frames (see Figure 5 (d) -(e)). We can compare between the frames of Figure 5(e) and Figure 2(b) to see the performance of our algorithm. Figure 5 (f) shows the result of applying 1D LMedS on 4 consecutive frames containing a static object; before applying 1D LMedS, the depth values at the pixels were fluctuating from one frame to another, whereas, after applying 1D LMedS, those pixels have stable depth value. For the moving parts of the scene, 1D LMedS also performed good in filling the missing depth values (see Figure 6), but the limitation is, when we increase the size of the window beyond 30 frames, we see ghosting effect. For the test, we used a window size of 15 for scenes with static objects and 30 for moving objects. Besides, we also observe that, when for a particular pixel, there is no valid depth value inside the window of frames, that pixel continues to remain as black spot (see Figure 6(b)). With our CPU implementation of 1D LMedS, we achieve a performance of 7 frames per second.

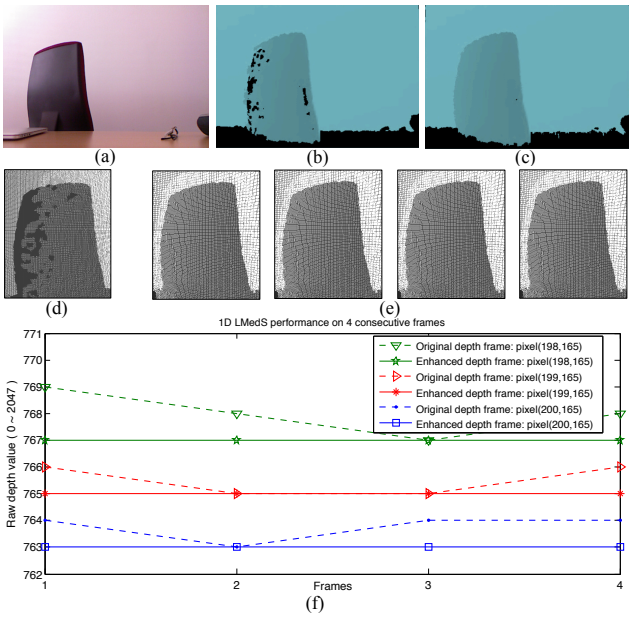

Fig. 5 Results with static object– (a) color image, (b),(d) raw depth image, (c) enhanced depth image with 1D LMedS, (e) 4 consecutive depth frames (zoomed for clear visibility) with stable depth values, (f) the depth values of 3 pixels, before and after applying 1D LMedS, on the 4 consecutive frames of (b) show that the enhanced frames have stable depth values for all of these 3 pixels over the consecutive frames.

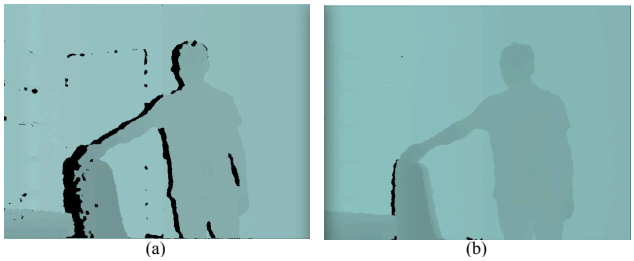

Fig. 6 Results with moving object – (a) raw depth image, (b) enhanced depth image with our approach. The black area near the edge of the object in (b) occurs because the depth sensor cannot visualize that area and hence, there is no depth information for that area.

We have also used test sequence from [3] and shown the performance of 1D LMedS on that sequence in Figure 7.

6 Conclusion

We have introduced a robust 1D LMedS approach which can efficiently enhance the depth images generated by Kinects. The output of our approach shows good performance in filling the invalid depth values and stabilizing the valid values for static and dynamic parts of a scene. Although, 1D LMedS exhibits less edge bleeding artifact, it can be further reduced by using an anisotropic diffusion approach along with structural similarity of depth and color frames. Moreover, a GPU implementation of 1D LMedS would provide a real-time speed to enhance the depth frames.

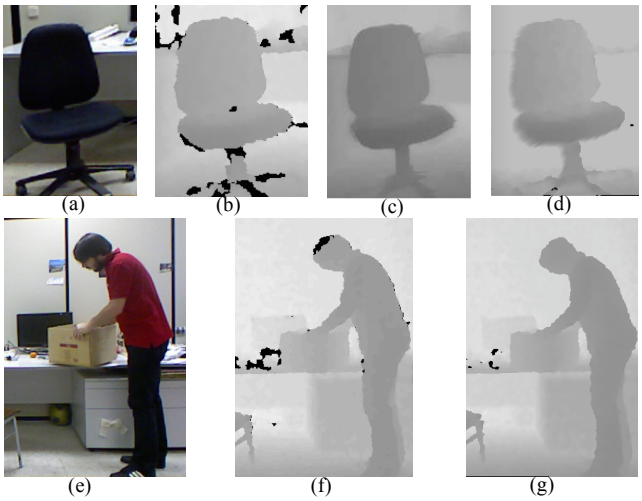

Fig. 7 Results with static and moving objects in test sequence from [3] – (a),(e): color image, (b),(f): raw depth image (static and moving objects), (c) enhanced depth image (static object) by [3], (d) enhanced depth image (static object) by our approach, (g) enhanced depth image (moving object) by our approach.

Acknowledgements This work was supported by the EU FP7 Marie Curie ITN "DIVA" under REA Grant Agreement No.290227 and by the German Research Foundation (DFG) within the GRK 1424 MuSAMA.

References

1. Amamra, A., Aouf, N.: Gpu-based real-time rgbd data filtering. Journal of Real-Time Image Processing pp. 1–18 (2014)
2. Beck, S., Kunert, A., Kulik, A., Froehlich, B.: Immersive group-to-group telepresence. IEEE Transactions on Visualization and Computer Graphics **19**(4), 616–625 (2013)
3. Camplani, M., Mantecon, T., Salgado, L.: Depth-color fusion strategy for 3-d scene modeling with kinect. IEEE Transactions on Cybernetics **43**(6), 1560–1571 (2013)
4. Chen, L., Lin, H., Li, S.: Depth image enhancement for kinect using region growing and bilateral filter. In: 21st International Conference on Pattern Recognition (ICPR), pp. 3070–3073 (2012)
5. Dou, M., Fuchs, H.: Temporally enhanced 3d capture of room-sized dynamic scenes with commodity depth cameras. In: IEEE Virtual Reality (VR), pp. 39–44 (2014)
6. Hui, T.W., Ngan, K.N.: Motion-depth: Rgb-d depth map enhancement with motion and depth in complement. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3962–3969 (2014)
7. Maimone, A., Fuchs, H.: Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras. In: 10th IEEE ISMAR, pp. 137–146 (2011)
8. Matyunin, S., Vatolin, D., Berdnikov, Y., Smirnov, M.: Temporal filtering for depth maps generated by kinect depth camera. In: IEEE 3DTV Conference, pp. 1–4 (2011)
9. Rousseeuw, P.J.: Least median of squares regression. Journal of the American statistical association **79**(388), 871–880 (1984)
10. Wasza, J., Bauer, S., Hornegger, J.: Real-time preprocessing for dense 3-d range imaging on the gpu: Defect interpolation, bilateral temporal averaging and guided filtering. In: International Conference on Computer Vision Workshops, pp. 1221–1227 (2011)
11. Yang, Q., Ahuja, N., Yang, R., Tan, K.H., Davis, J., Culbertson, B., Apostolopoulos, J., Wang, G.: Fusion of median and bilateral filtering for range image upsampling. IEEE Transactions on Image Processing **22**(12), 4841–4852 (2013)