DEPTH ENHANCEMENT USING ADAPTIVE AUTOREGRESSIVE MODEL

Rishab Shah, Fan Cao

5/5/2017

Boston University

Department of Electrical and Computer Engineering

Technical Report No. ECE-

BOSTON
UNIVERSITY

DEPTH ENHANCEMENT USING ADAPTIVE AUTOREGRESSIVE MODEL

Rishab Shah, Fan Cao

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

5/5/2017

Technical Report No. ECE-

1 Introduction

Microsoft's Kinect is a motion sensing device that uses a infrared projector and camera and a special sensing microchip that generates a mesh from the location of nearby objects in three dimensions. This 3D scanning is also known as *Light Coding*. It finds applications not only in video gaming but also in research where applications such as robotics [1], human motion analysis [2], human-computer interaction [4], human pose recognition [3] and computer vision where it is possible to capture a depth map of the environment based on the physical constraints of the sensor (field-of-view, sensor accuracy, etc.). Kinect depth is measured by the generation of strurctured infrared light speckles in a specific pattern. Then, the depth is computed by triangulation of each speckle between the observed light pattern and the reference light pattern which is obtained by capturing a plane at a noted distance and stored in the memory of the sensor. The first limitation is that is that it is usually assumed that sensing is performed in an indoor environment. Due to the problems that arise from but not restricted to occlusions (when there is no information due to sensor setup), or loss of data at transparent objects, distortions, structural data missing, or errors due to low resolutions and undersampling.
This project proposes an color-guided autoregressive (AR) model [10] for high quality depth map recovery from low quality measurements captured by depth cameras. We try to optimize the minimization problem of the AR prediction errors. The AR predictor for each pixel is constructed according to the local correlation in the initial depth map and non-local similarity in the accompanied color image. The proposed system is versatile for mainstream depth sensors, and Kinect, as we intend to demonstrate by experiments on the Kinect system.

2 Literature

There are quite a few methods that propose a solution for solving this problem and reconstruct/"recover" this missing data. One of the methods is proposed by Stommel et. al. [5] using an inpainting approach to fill the holes in the depth map using background estimates i.e., comparing specific properties of the Kinect itself. Another method is by using iterative diffusion and RGBD segementation [6] to recover depth information. Another method for recovery of depth maps is by using non-LSI (linear, shift invariant) filters such as the bilateral filter, joint bilateral filter or the non-local means filter and using both the color image and its corresponding depth map in conjunction to preserve the sanctity of the edges. More importantly, research that targets temporal denoising has to be weighted higher as this is more useful in a real-time application. One such method is by using a variation of the smoothed pointing algorithm [7] , which consists in scaling movements by using a velocity-based adaptive gain. Also, Lai et. al. [8] applied a recursive median filter in the construction of the RGB-D dataset, but blurring occurs for large occlusions. Finally, using a combination of spatial filtering and temporal denoising can be used as shown by

Camplani and Salgado [9] where they reduce the depth noise with a joint-bilateral filter on the spatial domain and the repair the depth value variation in the temporal domain. One problem that arises constantly when using spatio-temporal methods for recovering depth maps for Kinect is that the filtered dpeth maps are unexceptional around depth discontinuities.

3 Problem Statement

From our previous discussion, we know that depth capturing systems may suffer from different kinds of degradations, such as noise pollution, occluded regions around object boundaries, and suffers from degradation of structural depth missing. This method handles all four kinds of degradations in the proposed unified depth recovery framework.

3.1 AR model

An autoregressive model is a representation of a kind of random process and it specifies that the output depends linearly on its own previous values on a stochastic term i.e., the predictor, and the model is in the form of a stochastic difference equation. Depth-color pairs have been known have strong correlation in terms of geometrical structures and the locations of the edges in the depth maps can be correlated to the corresponding color images. This is the main motivation behind using a color-guided AR model for the recovery of depth maps.

3.2 Assumptions and constraints

The first assumption before any filtering of images and depth maps is that we have discretised all the depth maps from raw depth values in meters into a normalized range between zero and one. Depth maps from the Kinect have to be correctly calibrated with the accompanying color image for them to be ready to use in any filtering process. Else, we might lose important information along different structural data. Hence, we assume that the set of depth maps and their corresponding color images that we use are already calibrated to the best degree. We also do not use depth maps with areas that have transparent objects in the field of view or any shiny objects as the infrared sensor does not return the information and we lose large amounts of data from the depth map. Finally, in this project we are not looking at performing any temporal filtering or any real-time denoising and hence we are not looking at time constraints of the implementation.

3.3 Convex Optimization

Optimization problems are generally characterized by particular forms of objective or constraint functions. A vector is called optimal, or a solution of the problem if it has the smallest(or largest, depending on the type of optimization) objective that satisfy the constraints. A convex optimization problem is one in which the objective and constraint functions are convex, which means they satisfy the inequality

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \tag{1}$$

$\forall x, y \in \mathbb{R}^n$ and $\forall \alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1$, $\alpha \geq 0, \beta \geq 0$. We consider convex optimization to be a generalization of linear programming. The optimization problem proposed here is

$$\min_D E_{data}(D, D^0) + \lambda E_{AR}(D) \tag{2}$$

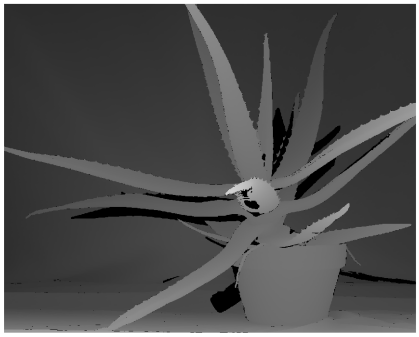
where $E_{data}(D, D^0)$ is the term required to rectify the recovered depth with the observed values, $E_{AR}(D)$ is the AR term to impose the AR model on the recovered depth map. The data term and the AR term are weighted by λ.

4 Solution

4.1 Getting Kinect-like data

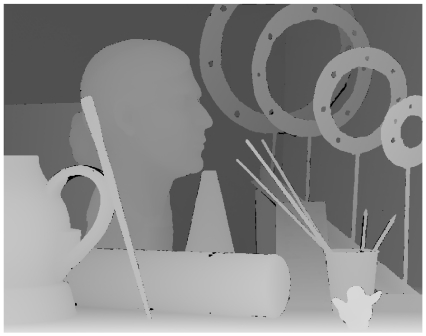
We used ground truth data for depth maps and their corresponding color images from the Middlebury dataset [11] and UCL dataset [8] that provides real data from Kinect with RGB colored image and depth map. In order to test the performance of the interpolation, we need color image, Kinect depth map, and ground truth depth map. Since most of the database provides either ground truth depth map, or Kinect like depth map, and color image, we have to generate one of the three. We decided to use the ground truth map to simulate the Kinect style degradation. There are four type of degradations: under sampling, random depth missing, structural depth missing, and pollution with additive noise. This method intends to deal with random depth missing and structural depth missing.

To simulate the structural missing around the edge of an object, we use the edge detection from MATLAB to extract the edge of ground truth map and use low pass filter to expand the area. For the random missing path, we use Photoshop to add black patches with different irregular shapes in the flat areas, then use MATLAB to make those patches black, since the Photoshop will smooth the edge which is undesirable. Through this we can get Kinect like depth map which we can later use to compare with the ground truth maps.

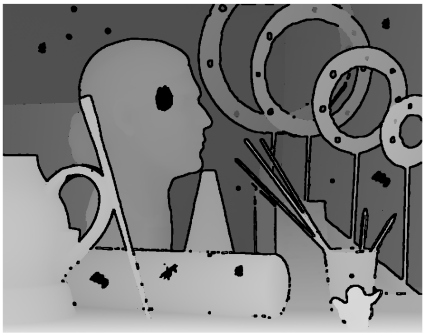


(a) Ground truth image

(b) Kinect-like image



(a) Ground truth image



(b) Kinect-like image

4.2 Interpolating the initial depth map

In order to generate the rough depth map, we needeed to interpolate the missing data (which we interpreted as scattered data in order to use it in MATLAB) in the Kinect depth map. We decided to use the **griddata()** function in MATLAB. **griddata()** is designed to interpolate the scattered data points, failing at arbitrary positions in the image. The function uses the Delaunay triangulation. A Delaunay triangulation [12] for a set P of points in a plane is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation, and it connects points in a nearest-neighbor manner. It dissects (tiles) the planar region into a set of non-overlapping triangles, so that any point to be interpolated inside the convex hull of the data must lie inside exactly one triangle. (Or, it may lie on a shared edge or vertex. Since the interpolant will be continuous, this is not a problem.)
The interpolated value at a query point is based on linear interpolation of the values at neighboring grid points in each respective dimension. For the nearest neighbor algorithm, the interpolated value at a query point is the value at the nearest sample grid point. Compared to other methods, the nearest neighbors have discontinuous,

while the other ones tend to be continues.

4.3 The AR predictor term for the Depth map

The data term is given by:

$$E_{data}(D, D^0) \triangleq \sum_{x \in \mathcal{O}} (D_x - D_x^o)^2, \tag{3}$$

and the AR model is shown as:

$$E_{AR}(D) \triangleq \sum_{x} \left(D_x - \sum_{y \in \mathscr{N}(x)} a_{x,y} D_y \right)^2, \tag{4}$$

where $\{x\} \in \mathcal{O}$ are the set of all pixels in the depth map, and $y \in \mathscr{N}(x)$ are the pixels in the neighborhood of x. The AR coefficient $a_{x,y}$ is defined as follows:

$$a_{x,y} = \frac{1}{S_x} a_{x,y}^{\hat{D}} a_{x,y}^{I} \tag{5}$$

where $\hat{D}$ is the rough estimated depth map obtained by nearest neighbor interpolation from D^0. The accompanied color image is represented with $I = \{I^i,\ i \in \mathcal{C}\}$, where I^i is the intensity of the color channel with index i and $\mathcal{C}$ is the index set of color channels in the color space. We will test various color spaces for their performance and implement it accordingly. S_x is the normalization factor, $a_{x,y}^{\hat{D}}$ and $a_{x,y}^{I}$ are the depth term and the color term respectively.
The depth term $a_{x,y}^{\hat{D}}$ is defined on the initial estimated depth map $\hat{D}$ by a range filter:

$$a_{x,y}^{\hat{D}} = exp\left(-\frac{(\hat{D}_x - \hat{D}_y)^2}{2\sigma_1^2} \right), \tag{6}$$

where σ_1 is the decay rate of the range filter. The AR predictor for that pixel x just gives a high predictor value for smooth regions in the depth map.
The color term for the AR predictor is a little more complicated. It uses the correlation between the depth mnap the color image about edges in the color images to preserve the edges in the depth map by assigning a heavier weight to smoother regions in each channel. This is possible by the use of an edge-preserving and a noise-reducing smoothing filter such as the bilateral filter. The color term $a_{x,y}^{\boldsymbol{I}}$ is given by:

$$a_{x,y}^{I} = exp\left(-\frac{\sum_{i \in \mathcal{C}} ||B_x \circ (\mathcal{P}_x^i - \mathcal{P}_y^i)||_2^2}{2 \times 3 \times \sigma_2^2} \right), \tag{7}$$

where σ_2 controls the decay rate of the exponential function, $\mathcal{P}_x^i$ denotes an operator that extracts a $\omega \times \omega$ patch centered at x in color channel i, "$\circ$"represents the element-wise multiplication. We use patches of size 3x3 for our implementation. The bilateral filter kernel is given by:

$$B_{x,y} = exp\left(-\frac{||x-y||_2^2}{2\sigma_3^2}\right) exp\left(-\frac{\sum_{i\in\mathcal{C}}(I_x^i - I_y^i)^2}{2\times 3\times \sigma_4^2}\right) \tag{8}$$

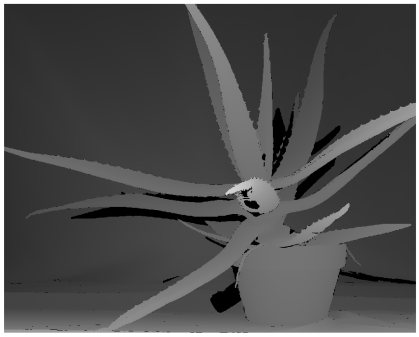
where σ_3 and σ_4 are parameters of the bilateral kernel to adjust the vitality of spatial distance and intensity difference, respectively. The way we approached this problem is that we take the difference of each pixel's patch with that of its neighbors' $y \in \mathcal{N}(x)$ where we take a neighborhood of 11x11 pixels. The values for all parameters we set are $\lambda = 0.01$, $\sigma_1 = 2$, $\sigma_2 = 9$, $\sigma_3 = 5$ and $\sigma_4 = 2$.

5 Results

	Resolution	PSNR	MAD
Plant	1282x1110	17.9533	13.57
Art(face)	1390x1110	15.6266	11.7

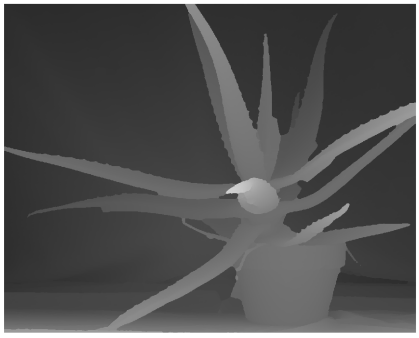
Figure 3: Quantitative depth recovery results from Kinect-like degradations

From the table, we can compare the PSNR and MAD (mean absolute difference) between the ground truth depth map and the filtered depth map and we see that these values are not good for recovered depth maps as compared to the same method used in [10] where they get a MAD of 0.58 for the ART set of images. This may be due to the selection of a heavy weight for the optimization or due to not running enough iterations to optimize the AR prediction errors. Our code took about two hours to run one iteration for the ART images.

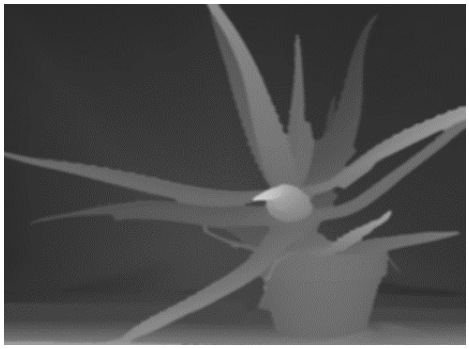


(a) Ground truth depth map

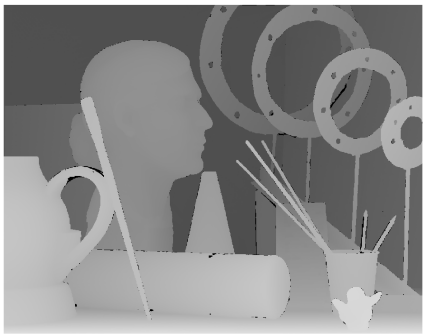
(b) Color image



(a) Nearest neighbor interpolation



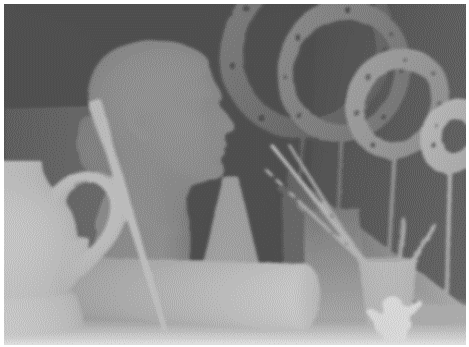
(b) Filtered depth map



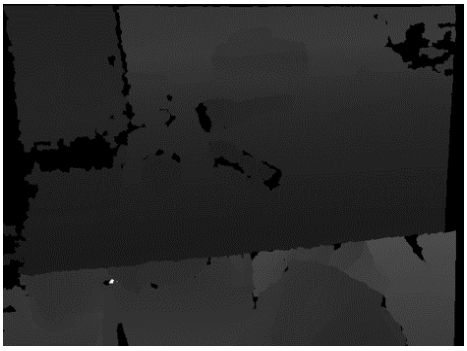
(a) Ground truth depth map

(b) Color image

(a) Nearest neighbor interpolation

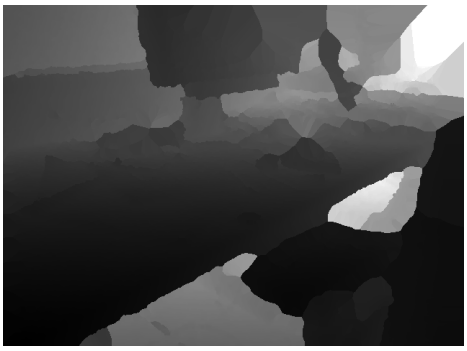


(b) Filtered depth map

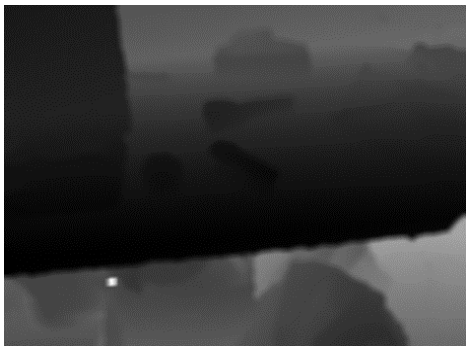


(a) Kinect Image of a table

(b) Color image



(a) Nearest neighbor interpolation



(b) Filtered depth map

6 Conclusion

In this project, we presented a new framework to recover depth maps from low quality measurements with mutiple types of degradations. We implemented a pixel-wise adaptive AR model using both the depth map and the color image. The depth map is recovered by minimizing AR prediction errors subject to the observation consistency.

6.1 Future Work

The code that implemented from scratch can be optimized easily for faster performance and also use of the GPU in running the code will reduce the runtime of the program. Also, in the future, we would like to work on temporal denoising in real-time for the Kinect so that it can be used in practical applications.

References

[1] J. Hartmann, D. Forouher, M. Litza, J.H. Kluessendorff, E. Maehle, "Real-time visual slam using fastslam and the microsoft kinect camera." *7th German conference on robotics*; proceedings of ROBOTIK 2012, pp 16.

[2] G. Parra-Dominguez, B. Taati, A. Mihailidis "3d human motion analysis to detect abnormal events on stairs.", *second international conference on 3D imaging, modeling, processing, visualization and transmission (3DIMPVT)*, 2012, pp 97103.

[3] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, "Real-time human pose recognition in parts from single depth images.", *Proceedings of the 2011 IEEE conference on computer vision and pattern recognition, CVPR 11.* IEEE Computer Society, Washington, DC, pp 12971304.

[4] L. Gallo, A. Placitelli, M. Ciampi, "Controller-free exploration of medical image data: experiencing the kinect.", *24th international symposium on computer-based medical systems (CBMS)*, 2011, pp 16

[5] M. Stommel and M. Beetz and W. Xu, "Inpainting of Missing Values in the Kinect Sensor's Depth Maps Based on Background Estimates", *IEEE Sensors Journal*, vol. 14, Pages 1107-1116, 2014

[6] A. Dakkak and A. Husain, "Recovering missing depth information from Microsofts Kinect", *in Proc. Embedded Vis. Alliance*, Boston, MA, USA, 2012.

[7] L. Gallo, M. Ciampi, A. Minutolo, "Smoothed pointing: a user-friendly technique for precision enhanced remote pointing. *In: International conference on complex, intelligent and software intensive systems (CISIS)*, 2010, pp 712717.

[8] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multiview RGB-D object dataset", *in Proc. Int. Conf. Robot. Autom.*, 2011, pp. 18171824.

[9] M. Camplani and L. Salgado, "Adaptive spatio-temporal filter for lowcost camera depth maps", *in Proc. Int. Conf. Emerg. Signal Process. Appl.*, Jan. 2012, pp. 3336.

[10] J. Yang, X. Ye, K. Li, C. Hou and Y. Wang., "Color-guided depth recovery from RGB-D data using an adaptive autoregressive model." *IEEE Transactions on Image Processing 23.8* (2014): 3443-3458.

[11] D. Scharstein, H. Hirschmller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling., "High-resolution stereo datasets with subpixel-accurate ground truth." *In German Conference on Pattern Recognition (GCPR 2014)*, Munster, Germany, September 2014.

[12] S. Lertrattanapanich and N. K. Bose, "High resolution image formation from low resolution frames using Delaunay triangulation." *IEEE Transactions on Image Processing* 2002: 1427-1441.