Layer Depth Denoising and Completion for Structured-Light RGB-D Cameras

Ju Shen Sen-ching S. Cheung
Center for Visualization and Virtual Environments, University of Kentucky
{jushen.tom, sen-ching.cheung}@uky.edu

Abstract

The recent popularity of structured-light depth sensors has enabled many new applications from gesture-based user interface to 3D reconstructions. The quality of the depth measurements of these systems, however, is far from perfect. Some depth values can have significant errors, while others can be missing altogether. The uncertainty in depth measurements among these sensors can significantly degrade the performance of any subsequent vision processing. In this paper, we propose a novel probabilistic model to capture various types of uncertainties in the depth measurement process among structured-light systems. The key to our model is the use of depth layers to account for the differences between foreground objects and background scene, the missing depth value phenomenon, and the correlation between color and depth channels. The depth layer labeling is solved as a maximum a-posteriori estimation problem, and a Markov Random Field attuned to the uncertainty in measurements is used to spatially smooth the labeling process. Using the depth-layer labels, we propose a depth correction and completion algorithm that outperforms other techniques in the literature.

1. Introduction

A typical structured-light stereo RGB-D system such as Microsoft Kinect consists of a projector and two camera sensors. Special light patterns, typically in the infra-red (IR) spectrum, are emitted by the projector and captured by the IR CMOS camera sensor. Depth information of the sensing environment can then be inferred based on how the light patterns are distorted in the IR images [5]. The RGB camera captures the color information and can be aligned with the IR camera by estimating the extrinsic parameters between them.

Depth images obtained by such an imaging system have two major problems: missing and distorted depth values. Figure 1 (row 1) shows a pair of typical RGB and depth images obtained by Kinect. All the black regions in the depth image contain no depth measurements. Some of these

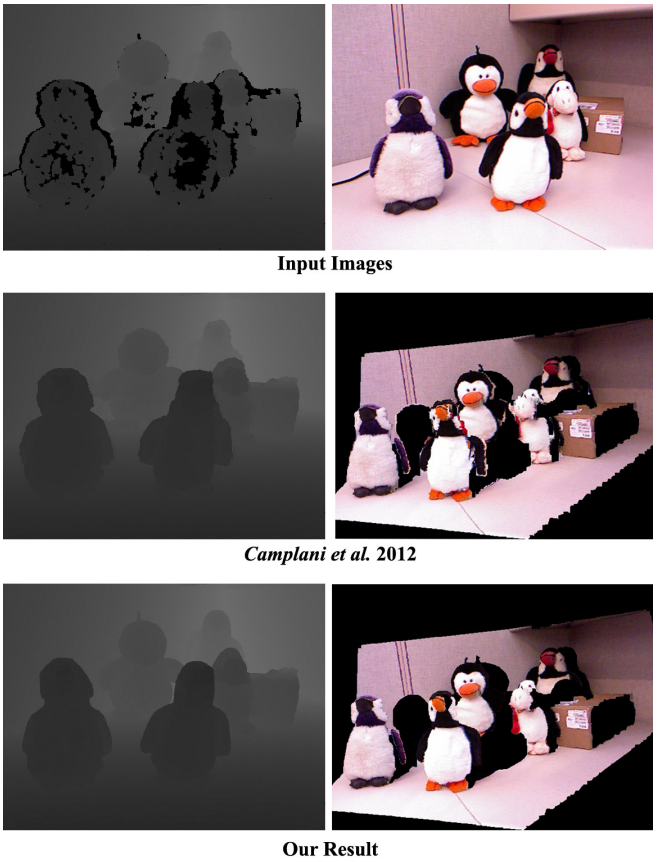


Figure 1. Depth Denoising and Completion. 1st column: input depth image; completed depth by Camplani et al. [2] and our method. 2nd column: input RGB image and two corresponding reconstructed virtual views

regions occur on object surface, such as the holes on the front penguin's belly. This is caused by the short distance between the object and the depth camera. Missing regions also present along the object boundaries. This represents a major source of depth error caused by the disparity between the projector and the sensor. Sometimes, they can be orders of magnitude different from their true values.

Missing and erroneous depth values can also be caused by absorption, poor reflection or even shadow reflection of

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.157

the light patterns. Objects with darker colors, specular surfaces, or fine-grained surfaces like human hair are prime candidates for poor depth measurements [3]. Surface orientation also plays a role in depth measurements – as the surface normal deviates from the principal axis of the IR camera, the accuracy of the depth measurement declines and becomes unreliable near depth discontinuities [10]. Random noise in depth images may also be a result of inadequate calibration, sensor noise, and round-off error during normalization [8]. Generic image denoising and complete algorithms fail to take into account the unique problems of structured-light RGB-D systems.

Our technical contribution is the use of depth "layer" in steering the completion process to produce well-defined depth edges. We describe a novel stochastic framework that separates the depth image into multiple layers, and combines multiple RGB-D system noise models to robustly determine the depth layer label. The goal is to denoise and complete missing values on the depth image that improve the quality for any subsequent RGB-D applications.

2. Related Work

Most of the works in depth image enhancement can be grouped into two categories: super-resolution [7] [11] [17] [4] and image in-painting [16] [15]. A common theme is to rely on information obtained from the companion color images to predict missing depth information. The use of color information for depth enhancement is based on the assumption that certain correlation exists between depth continuality and color image consistency [9]. While providing useful cue for interpolation, this assumption does not always hold as color edges and depth edges do not necessarily coincide with each other. In [7], Garro et al. presented an interpolation scheme for depth super-resolution. A high resolution RGB camera was used to guide the up-sampling process on the depth image. To interpolate the missing depth pixel, the scheme used neighboring depth pixels mapped into the same color segment as the target pixel. This method relied strongly on the extrinsic alignment between the color and depth image. A similar segmentation based method can also be found in [11] where a non-local means filtering based approach was used to regularize depth maps and maintain fine detail and structure. In [16], Wang et al. proposed a stereoscopic in-painting algorithm to jointly complete missing texture and depth by using two pairs of RGB and depth cameras. Regions occluded by foreground were completed by minimizing an energy function. The system required an additional pair of color and depth cameras to achieve the goal.

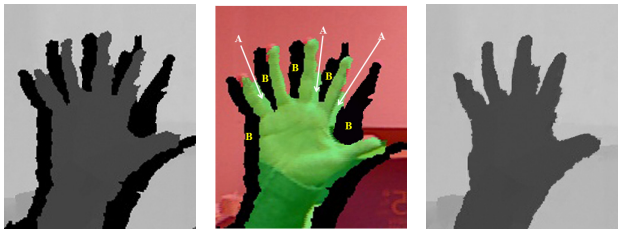
Various probabilistic frameworks are often used in modeling depth measurements, fusing depth and color information, and predicting missing values. In [4], Diebel et al. demonstrated the use of Markov Random Field in the super-resolution of depth data using high-resolution color data. However, their work provided little insight in modeling the sources of error in the depth sensor. Similarly, in [17], a low resolution depth image was iteratively refined through the use of a high resolution color image. Bilateral filter was applied to a cost function based on depth probabilities. A final high resolution image was produced by a winner-takes-all approach on the cost function. These approaches work well for the super-resolution problem where missing depth pixels are uniformly distributed. Depth images obtained by structured-light sensors often have large contiguous regions of missing depth measurements which cannot be handled by such approaches. In [14], the depth map was refined through foreground/background separation. Though the approach could well preserve foreground edges, it may fail when the captured scene has complex depth variation.

3. Problem Definition

Missing depth can be categorized into two types. The first type is the randomly distributed "small holes" on objects' surfaces. These missing values can usually be inferred by the available depth pixels in the neighboring regions. In addition, the corresponding RGB information can be used to steer the depth completion by using techniques such as Joint Bilateral Filter [9].

The second type is the large and contiguous missing depth patches that often present along the boundary between close (foreground) and far objects (background). An example can be found in Figure 2(a), where there are many missing depth (black) regions around the hand. This is caused by the disparity between the IR projector and the IR camera sensor. Part of background regions are visible to the IR camera but not to the projector and receive no structured light patterns. Thus, the depth values in those regions cannot be measured.

Missing and incorrect depth values due to disparity cannot be easily and correctly inferred by many existing works including Joint Bilateral Filter based schemes [9] [2] and probabilistic based schemes [4]. Figure 2 explains why these methods yield unsatisfactory results. The raw depth image shown in Figure 2(a) clearly indicates two distinct depth layers: foreground in dark gray and background in light gray. No depth measurements are obtained in the black region. In Figure 2(b), we overlay semitransparent green (foreground) and red (background) layers over the RGB image as an indicator of its corresponding depth information. There are a number of background pixels wrongly labeled as foreground along the boundaries of the hand (annotated as "A"). The missing depth patches (annotated as "B") are often adjacent to "A". Most existing spatial approaches complete these missing values by using the erroneous depth in the neighboring areas. Using the color channel provides



(a) (b) (c)

Figure 2. (a) Depth Image; (b)Labeled Image by depth information; (c) Depth Completion by Camplani et al. [2]. The alignment between the depth and color images is based on the calibration result from Mircrosft Kinect diver

little to rectify this problem because the RGB values from "A" are very similar to the ones from "B", both of which are from the background color. The depth completion result using a joint color-depth bilateral [2] shown in Figure 2(c) is indeed quite poor.

4. Proposed Method

We assume that the dynamic 3D environment can be separated into a static background and a number of dynamic foreground objects. This is a configuration typically seen in a home or office environment with a small number of individuals moving in front of the device. The RGB and depth cameras are assumed to be extrinsically aligned and temporally synchronized.

After an initial step of offline training on background-only frames, our online algorithm consists of two main phases: layer labeling followed by depth denoising and completion. In the first phase, each pixel of the incoming frame is labeled by different layers via a probabilistic framework that incorporates a data measurement model and a smoothing neighborhood model based on available observations. Maximum A Posteriori (MAP) estimation is used in the labeling to prevent blurring along depth discontinuities.

In the second phase, the labels estimated in the first phase are used to steer the removal of outlier and the completion of missing depth values, from either the background model or from neighboring depth values with the same labels. The robust labeling allows us to preserve the shape of object boundary and prevent noise propagation across objects with significant depth differences.

4.1. Layer-driven Stochastic Models of Depth Measurements

The probabilistic graphical model that describes the multi-layered RGB-D measurement process is shown in Figure 3. Let G be the support of the 2D color and depth images. At each pixel location $s \in G$, X_s denotes the latent random integer variable indicating which layer the pixel belongs to. X_s can assume any value in the set $\{-n', -n'+1, \ldots, -1\} \cup \{1, 2, \ldots, n\}$. Negative numbers are layers from the static background while positive numbers refer to layers in the foreground. The larger the layer number is, the closer it is to the camera. The number of background layers n' and the number of foreground layers n are determined based on the observed data and are the same for all pixels. The approaches to estimate n and n' will be described in Section 4.1.1.

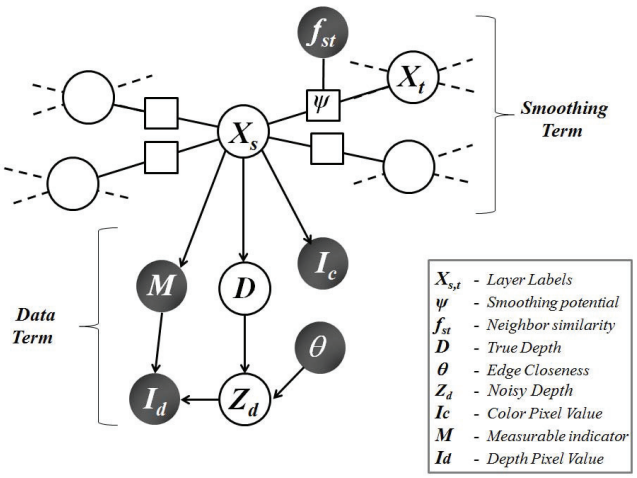


Figure 3. RGB-D model: the smoothing term on top specifies the constraints between neighboring labels while the data term below describes the measurement process. Darken nodes are actual measurements, white nodes are hidden variables, and square nodes are factors.

Spatially, a layer pixel is connected to its four closest neighbors, generically referred to as X_t. All the labels over the entire image thus form a Markov Random Field (MRF) and the spatial relationship between adjacent labels is governed by an edge factor $\psi(X_s, X_t, f_{st})$, where f_{st} is the measured similarity between the two pixels. Each layer label X also has its evidence potential function $\phi(X_s)$ based on the measurement Bayesian Network (BN) shown in the lower half of Figure 3. As all the measurements are made at the same pixel, the subscript s is omitted and $\phi(X)$ is defined as follows:

$$\phi(X) \triangleq P(X) \cdot P(I_c|X) \cdot P(M|X) \cdot P(I_d|M, \theta, X) \quad (1)$$

where

$$P(I_d|M, \theta, X) = \int_D P(D|X) \int_{Z_d} P(I_d|Z_d, M) P(Z_d|D, \theta) \quad (2)$$

I_c represents the observed color values. D represents the true but unobserved depth values. D is corrupted by an additive Gaussian noise which produces a noisy measurement

Z_d. The noise variance is determined by the closeness measurement θ to the nearest edge. Due to the missing depth problem, Z_d may not be directly observable. We thus introduce an observable depth indicator random variable M which is 1 if the depth value is observed and 0 otherwise. Combining these two random variables results in the observable depth value $I_d = MZ_d$. Using this probabilistic model, layer labeling can be formulated as a Maximum a posteriori or MAP problem:

$$X_G^{MAP} \triangleq \arg\max_{x_G} \sum_s \log \phi(x_s) + \sum_{(s,t)\in G} \log \psi(x_s, x_t, f_{st}) \tag{3}$$

Our choice of parametrization allows $\psi(x_s, x_t)$ and $\phi(x_s)$ to be computed. While the complexity of the exact solution to the MAP problem is exponential in the image size, which is known to be NP hard. To improve the computation speed, various approximation algorithms have been proposed for a global optimization, such as Graph Cuts, or Loopy belief propagation [6]. In this paper, we used max-product based loopy belief propagation to approximate the inference. One major reason of choosing it than graph cuts is that belief propagation is more efficient in processing video sequence: the output of previous frame can be used as the initial value of messages for current frame, which makes the convergence speed improved.

4.1.1 Data Term

In building the data term for the graphical model, the four layer distributions $P(X)$, $P(M|X)$, $P(D|X)$, and $P(I_c|X)$ are estimated based on both offline training data and online data. The depth distribution $P(D, X) = P(X)P(D|X)$ is modeled as a mixture of Gaussian (MOG) model while the color distribution $P(I_c, X) = P(X)P(I_c|X)$ is modeled as multiple color histograms on the quantized HSV space, one for each layer. The observable depth indicator distribution $P(M, X) = P(X)P(M|X)$ is based a simple Bernoulli distribution for each layer. In fact, the parameter estimation for the depth indicator is a simpler version of the color distribution. As such, our discussion will focus on the depth and color distributions. The parameter estimation process is summarized in Figure 4.

During the offline training phase, we estimate the parameters for the negative (background) depth layers based on a set of training RGB-D frames of the static background. There are two phases of the estimation: global estimation and local adaptation. During the global estimation, all the pixels with both color and depth measurements will be aggregated to estimate a single pair of $P(D, X)$ and $P(I_c, X)$ using the Expectation-Maximization (EM) approach. $P(D, X)$ is initialized by K-means and $P(I_c, X)$ is initialized as an uniform distribution. It is important to

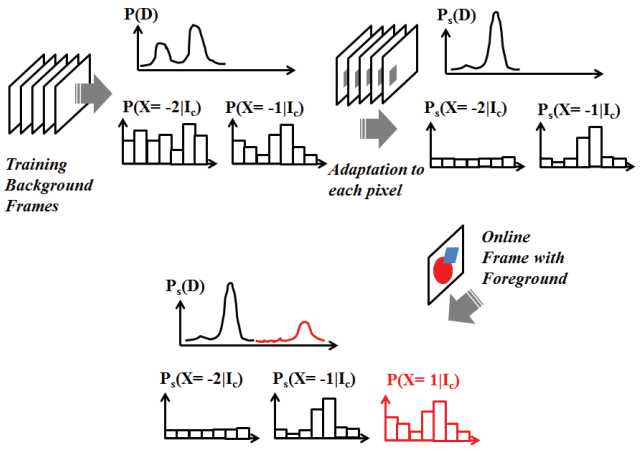


Figure 4. Parameter estimation for depth and color layer distributions

note that the concept of layers is based only on depth but not on color. As such, the EM process is primarily driven by the depth data in the sense that the E-step only estimates the layer posterior $P(X|D)$ for the depth but not the color. During the M-step, we use the depth data to update the estimates for the layer prior $P(X)$ and the depth layer conditional $P(D|X)$, only use the color data to update the color layer conditional $P(I_c|X)$ using the posterior probability $P(X|D)$ of the co-located depth pixel. As a result, two pixels with the same color value can have different contributions to different layers. Different number of layers are tested and the optimal number n' is determined by using the Bayesian Information Criterion (BIC) on the depth data [1]. The example in Figure 4 shows this first step to have two separate background layers and obtain the global color and depth models. In the second phase, the global distributions are adapted to each individual pixel by using only the temporal data at that pixel location. Sequential exponential weighing scheme is used for the adaption. For example, the local mean for layer $X_s = i$ at location s is updated by a new depth value d_{new} as follows:

$$\mu_{s,i}^{(t+1)} := \lambda P^{(t)}(X_s = i|D_s = d_{new}) \cdot d_{new} + \Big(1 - \lambda P^{(t)}(X_s = i|D_s = d_{new})\Big) \cdot \mu_{s,i}^{(t)} \tag{4}$$

t represents the iteration step and λ controls the rate of adaptation which is empirically set to 0.3. All the other parameters are updated in a similar fashion. Similar to the global distributions, the two layer conditional probabilities are updated based on the corresponding color or depth data. The layer prior is updated using the depth data if available, or the color data if the depth data is missing. After the adaptation, the parameters can better describe the characteristics of the local pixel – Figure 4 illustrates this idea where the local

depth distribution $P_s(D)$ and the color posterior $P_s(X|I_c)$ clearly indicate that this pixel is more likely to belong to layer -1.

The foreground layer distributions are estimated online. To deliver a real-time response and cope with fast moving objects, the foreground distributions are estimated for every frame. As no temporal data is maintained, we only estimate global distributions using an approach identical to that of the background global distribution. The training data are obtained based on only those pixels with valid depth measurements and very low background posterior probability, i.e. $\max_{i=1,\ldots,n'} P_s(X_s = -i|D_s) < \epsilon$ for a small fixed ϵ. To obtain the full range of $P(X)$, $P(D|X)$, and $P(I_c|X)$, we also need prior probabilities for foreground and background. We simply set them to be equally likely for our experiments though better performance may be possible with more foreground training data. In Figure 4, the foreground components of the layer distributions are in red color.

Next, the noisy depth measurement Z_d is modeled based on an additive Gaussian model:

$$Z_d \triangleq D + N, \quad \text{with } N \sim \mathcal{N}(0, \sigma_\theta^2) \text{ and } N \perp D. \quad (5)$$

The noise standard deviation σ_θ reflects the uncertainty in the depth measurement. As argued in Section 3, erroneous depth measurements occur predominantly near object boundaries. To model this effect, we apply an edge detector on the depth map and use the spatial distance θ to the closest depth edge as a reliability measure. The noise variance σ_θ is modeled as a deterministic logistic function given below:

$$\sigma_\theta := \frac{a}{1 + e^{-(b\theta - c)}} \quad (6)$$

a and b are constant scaling parameters. c/b is a distance threshold beyond which the depth value is relatively noise free. This simple model is easy to compute, though a more sophisticated one incorporating surface normal, texture, and color can be used in a similar fashion.

Finally, we present a simple approach to evaluate the integrals in Equation 2. Note that the actual depth measurement $I_d = MZ_d$ implies that $P(I_d = i_d|Z_d = z, M = m) = \delta_{mi_d}(z)$, the dirac delta function with the only non-zero value at $z = mi_d$. Substituting this into (2) results in the following simplification:

$$\begin{aligned} &P(I_d = i_d|M = m, \theta, X = x) \\ &\quad = \int_e P(D = e|X = x) P(Z_d = mi_d|D = e, \theta)\, de \\ &\quad = P(Z_d = mi_d|\theta, X = x) \end{aligned} \quad (7)$$

Given $X = x$, Z_d and D are multivariate Gaussian with the following distribution:

$$\begin{bmatrix} Z_d \\ D \end{bmatrix} \Bigg| X = x \;\sim\; \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_x \end{bmatrix}, \begin{bmatrix} \sigma_x^2 + \sigma_\theta^2 & \sigma_x^2 \\ \sigma_x^2 & \sigma_x^2 \end{bmatrix} \right) \quad (8)$$

Thus, $Z_d|X = x \;\sim\; \mathcal{N}(\mu_x, \sigma_x^2 + \sigma_\theta^2)$ and (7) can be numerically evaluated.

4.1.2 Smoothing Term

For the spatial MRF, the edge potential $\psi(X_s, X_t)$ is defined based on the similarity in color and depth between the neighboring pixels:

$$\psi(X_s, X_t, f_{st}) \triangleq \begin{cases} f_{st} & \text{for } X_s = X_t, \\ \frac{1}{n+n'}\left[1 - f_{st}\right] & \text{otherwise} \end{cases} \quad (9)$$

The similarity strength f_{st} is a feature based on how close the color and depth of the neighboring pixels are. It is modeled using the following equation:

$$f_{st} = \max\{\alpha \exp(-C_{st}) + (1-\alpha)\exp(-D_{st}), n_f\} \quad (10)$$

The color similarity ratio C_{st} and the depth similarity ratio D_{st} are defined in a similar fashion [13]:

$$C_{st} = \frac{\| I_c(s) - I_c(t) \|^2}{k_c \left\langle \| I_c(s) - I_c(\tilde{t}) \|^2 \right\rangle} \quad (11)$$

$$D_{st} = \frac{|I_d(s) - I_d(t)|^2}{k_d \left\langle |I_d(s) - I_d(\tilde{t})|^2 \right\rangle} \quad (12)$$

$\langle \cdot \rangle$ denotes the average operator. $\tilde{t}$ represents any one of the eight neighboring pixels of s (including pixel t). k_c and k_d are normalization constants so that the two terms are of the same range. n_f is the minimum similarity to prevent the potential function from becoming zero. If either depth measurement is not present, n_f will be used.

The parameter $\alpha \in [0, 1]$ is a trade-off between depth and color information. If the depth measurements are reliable, most of the weight should be assigned to depth values as they are more reliable for foreground/background labeling; if the depth measurements are unreliable, they should not be used at all in computing the edge potential. Similar to the approach used in Section 4.1.1, α is defined as the logistic function of the distances θ_s and θ_t of the two neighboring pixels to the closest edge:

$$\alpha \triangleq f\left(\tfrac{\theta_s + \theta_t}{2}\right) \text{ with } f(\theta) \triangleq \tfrac{1}{1 + e^{-(b\theta - c)}} \quad (13)$$

4.2. Depth Image Completion

After assigning each pixel with a layer label, we need to identify erroneous depth measurements and complete missing depth values. The erroneous depth values are essentially outliers that are significantly different from other depth values in the neighborhood. However, as most measurement errors occur around object boundaries, it is imperative not to mistake true depth discontinuities as wrong depth values. The layer labels allow us to separate pixels that are likely to have come from objects at completely different depths. To

Component	Time (seconds)
Data Term	0.115
Smoothing Term	0.032
EM Iteration	0.095
LBP Iteration	0.641

Table 1. Time Performance Evaluation

determine if a depth pixel is an outlier, we robustly estimate the depth distribution in the neighborhood around the pixel via a RANSAC-like procedure. First, we only consider depth values in the neighborhood that share the same label as the target pixel. Then, multiple small sets of random sample pixels are drawn and a Gaussian distribution is estimated for each set. If only a small fraction of the neighborhood can be fit within two standard deviations from the mean of a sample distribution, this distribution is likely to contain outlier samples and is thus discarded. Among those that survive the robustness test, the one with the smallest variance is used and the target depth pixel is declared an outlier if it is beyond two standard deviations from the mean. The outlier depth pixel will join the rest of the missing depth pixels and will be completing using a joint color-depth bilateral filtering scheme similar to that in [9]. The only difference is that we only consider the contributions from neighboring depth pixels that have the same layer label as the center pixel.

5. Experimental Results

We use a Microsoft Kinect (model LPF-00004) with the PrimseSense driver [12] to capture depth and RGB images at a resolution of 640×480. The proposed algorithm is implemented in C++ with the OpenCV library. The experiment is conducted on a computer with the following hardware settings: Intel Core(TM) i7-2820QM CPU at 2.30 GHz and 8.0Gb of RAM. For the static background training, we captured 100 images for each scene to obtain the background color and depth statistics. The speed performance for each component of our system is shown in table 1.

Figure 5 shows a simple example with one foreground and one background layer. We can see that the background wall is clearly visible between the foreground leaves and the rapid spatial changes of depth layers lead to significant measurement error shown in the original depth image in the top right. Our algorithm correctly identifies the number of layers and produces an accurate segmentation mask in the bottom left. Guided by this segmentation mask, our algorithm corrects and completes the depth values in the bottom right.

Figure 6 presents a more complex example with two foreground and two background layers. The two foreground layers are the head model (layer 2) as well as the books and the ball (layer 1). The two background layers are the white board and the table (layer -1) as well as the rest of the region (layer -2). Notice that each layer can have objects of varying depths. The curves and straight lines from different objects are well preserved in the output result shown in the bottom right.

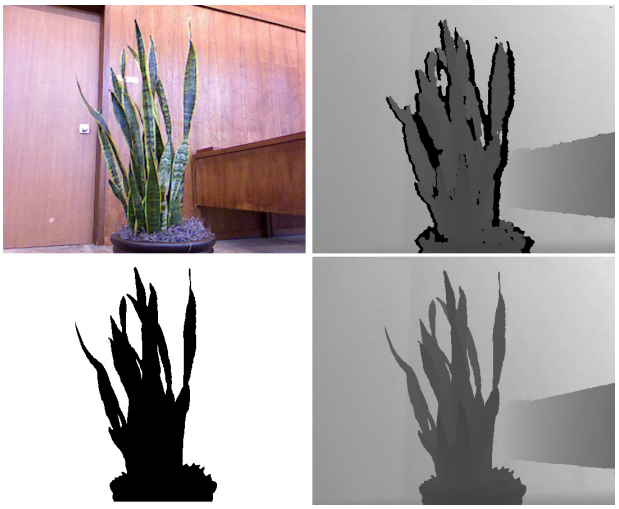


Figure 5. Scene with Two Layers: input RGB and depth images (top); layer labeling result and completed depth steered by layers (bottom).

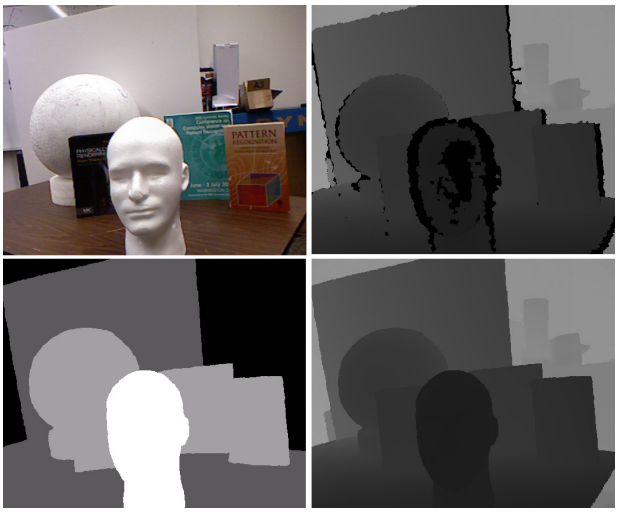


Figure 6. Scene with Multiple Foreground and Background Layers: input RGB and depth images (top); layer labeling result and completed depth steered by layers (bottom).

We have also compared our scheme with other works. In particular, we have chosen [2] which represents one of the most recent efforts in using joint color-depth bilateral filtering for depth completion, and [4] which uses a similar MRF as ours in providing spatial smoothing. Figure 7 presents a side-by-side comparison between these methods and ours. We have selected two sequences for comparison. Except for the last row, the original data is shown in the first column, results from [2] in second, [4] in third, and ours in

the last column. Two scenes are used in the experiments. The depth maps of the two scenes are shown in the first and third rows. To highlight the differences in depth completion, we generate a set of arbitrary views based on the produced depth image and the RGB image, which are presented in the second and fourth rows. In the last row, we zoom in on the rendered views and add pre-captured static background to fill the occluded regions.

The results from [2] are shown in the second column. As shown in the depth maps, this approach enlarges the foreground shape by attaching unrelated pixels around objects' boundaries. The wrongly assigned depth values move some of the background pixels to the foreground or vice versa. This error is clearly noticeable in the rendered virtual views – in the second row, a significant number of wall pixels present around the boundaries of the person's hair and arm. In the fourth row, the contours of the fingers are poorly inferred. Small gaps between fingers are naively wiped out due to erroneous color similarity or depth. Results from [4] in the third column have better contour than [2]. However, the MRF blurs the boundaries between foreground and background by generating intermediate depth values. From the virtual views, these intermediate depth values can be seen spreading across the space between the foreground and background. In contrast with these two schemes, our proposed method produces superior results. The depth completion steered by layers can better preserve the shape of object boundary and prevent noise propagation across objects with significant depth differences.

6. Conclusions

In this paper, we have proposed a novel depth denoising and completion scheme by combining color-depth correlation, background modeling, spatial smoothness, and measurement error models pertinent to structured-light systems. A probabilistic graphical model is used to fuse all these different factors together with the key latent variable being the depth layer at each pixel. The depth layer labeling is formulated as a MAP problem and a MRF attuned to the uncertainty in depth measurements is used to spatially smooth the labeling process. Driven by the obtained depth layer labels, depth noise is removed and depth interpolation is performed using a bilateral filter. There are some weaknesses to our proposed design. For example, additional depth features such as surface normal and texture may provide a more accurate uncertainty model in depth measurements. We are currently investigating more computationally efficient approaches for depth layer labeling, and extending our model to support multiple Kinects which can provide more complete 3D scanning but may introduce new errors due to interference.

References

[1] H. S. Bhat and N. Kumar. On the derivation of the bayesian information criterion. Technical report, University of California, Merced, 2010.

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

[3] J. Cho, S. Kim, Y. Ho, and K. Lee. Dynamic 3d human actor generation method using a time-of-flight depth camera. In *IEEE Transactions on Consumer Electronics*, volume 54, pages 1514–1521, 2008.

[4] J. Diebel and S. Thrun. An application of markov random fields to range sensing. In *Advances in neural information processing systems*, volume 18, pages 291–295, 2006.

[5] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. Depth mapping using projected patterns. Technical report, U.S. Patent No. 20100118123, 2010.

[6] W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. In *International journal of computer vision (IJCV)*, volume 40, pages 25–47. Springer, 2000.

[7] V. Garro, C. dal Mutto, P. Zanuttigh, and G. Cortelazzo. A novel interpolation scheme for range data with side information. In *Conference for Visual Media Production (CVMP)*, 2009.

[8] K. Khoshelham. Accuracy analysis of kinect depth data. In *ISPRS Workshop Laser Scanning*, 2011.

[9] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3), 2007.

[10] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[11] J. Park, H. Kim, Y. Tai, M. Brown, and I. Kweon. High quality depth map upsampling for 3d-tof cameras. In *International Conference on Computer Vision (ICCV)*, 2011.

[12] PrimeSense Inc. *Prime Sensor NITE 1.3 Algorithms notes*, 2010.

[13] C. Rother, A. Blake, and V. Kolmogorov. Grabcut - interactive foreground extraction using iterated graph cuts. In *In Proceedings of ACM SIGGRAPH*, 2004.

[14] J. Shen, C. S. C., and J. Zhao. Virtual mirror by fusing multiple rgb-d cameras. *Asia-Pacific Signal Information Processing Association Annual Summit and Conference (APSIPA)*, 2012.

[15] L. Torres-Méndez and G. Dudek. Reconstruction of 3d models from intensity images and partial depth. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2004.

[16] L. Wang, H. Jin, R. Yang, and M. Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[17] Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-depth super resolution for range images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

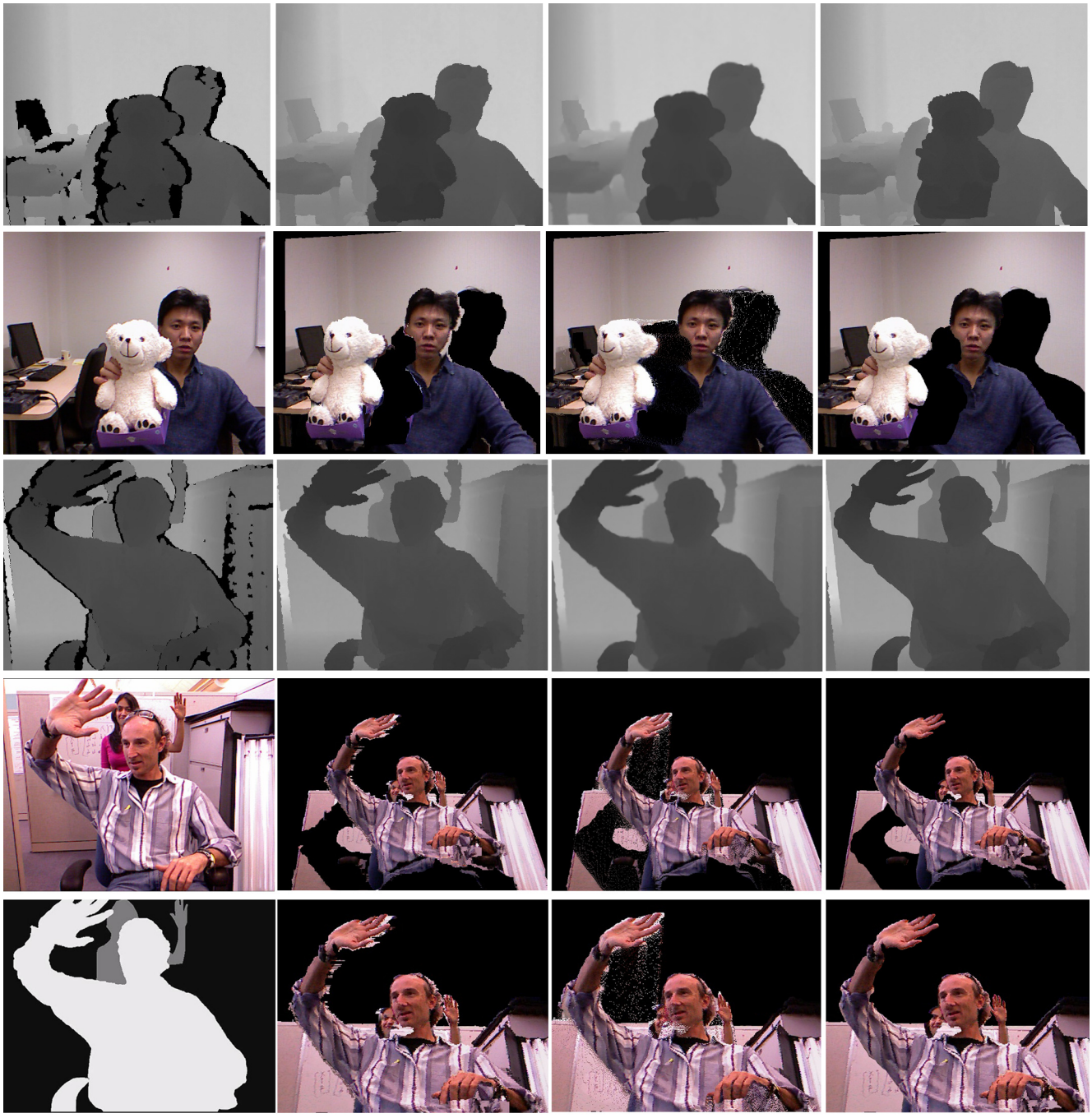


Figure 7. Comparison with other schemes. From left to right: input images together with our layer labeling, results from Camplani et al. [2], results from Diebel. et al. [4], and our results.