JOINT DENOISING AND INTERPOLATION OF DEPTH MAPS FOR MS KINECT SENSORS

Simone Milani and Giancarlo Calvagno

Dept. of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova - Italy.
e-mail: {simone.milani,calvagno}@dei.unipd.it

ABSTRACT

Infrared structured light sensors are widely employed for control applications, gaming, acquisition of dynamic and static 3D scenes. Recent developments have lead to the availability on the market of low-cost sensors which prove to be extremely sensitive to noise, light conditions, materials, the surface nature of the objects, and their distance from the camera. As a matter of fact, accurate denoising and interpolation strategies are needed.

The paper presents a quality enhancement strategy for depth maps targeting low-cost IR structured light sensors. The approach has been tested using the MS Xbox Kinect device in both indoor and outdoor scenarios under different light conditions.

Index Terms— interpolation, denoising, MS Kinect, 3D scanning, structured light camera, infrared sensor.

1. INTRODUCTION

The recent availability of low-cost range cameras has shaken the ICT world leading to a flourishing of new object recognition applications, human-computer interfaces, and acquisition systems of dynamic 3D scenes. Time-of-Flight cameras [1], structured light 3D scanners [2], multicamera systems allow easy real-time acquisition of dynamic 3D scenes with both static and dynamic elements.

Among these, the Xbox Kinect sensor [2], which includes a standard RGB camera together with an infrared (IR) structured light scanner, has recently proved to be one of the most widely-used sensors thanks to its versatility and the limited cost (see Fig. 2). Unfortunately, despite the strong versatility and the wide range of new applications that these IR devices enable, the resulting depth signal is affected by a significant amount of noise. One of the main reasons for this inconvenience is that for most of the acquired scene there is no control over illumination, and therefore, IR sensors receive a significant amount of radiation that has not been provided by the 3D device itself. Depth sensors also present shot noise related to the radiation, A/D conversion quantization noise, and thermal noise. Moreover, the artifacts along object boundaries and the limited resolution of the acquired depth maps utterly increase the need for interpolating and denoising algorithms.

Authors want to thank Carlo dal Mutto for his help in building the acquisition set-up and his advising.

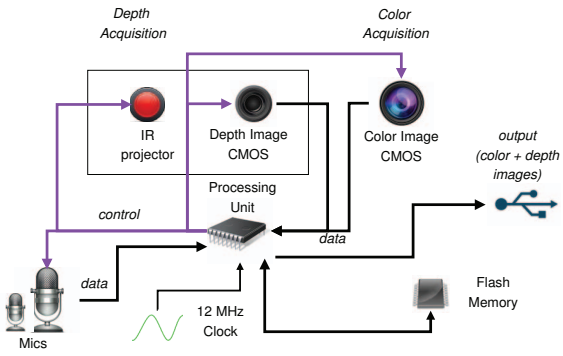

Fig. 1. Block diagram of the MS Kinect sensor.

Several works have proposed novel denoising algorithms to improve the quality of the acquired depth maps. One of these considers the confidence values to denoise depth acquired via a ToF camera, while other solutions rely on exploiting some side information obtained using a lateral color camera [3]. In addition, other techniques are employed to interpolate the depth maps in order to increase the resolution and fill missing data [4].

The paper presents a joint denoise-interpolation algorithm for MS Kinect sensor that aims at correcting the computed depth values and interpolate the depth map on those points where depth values are not available because of the noise conditions. The approach relies on an initial denoising performed by matching borders between range and color images. Then, a segmentation of the color image is employed to interpolate the data. Experimental results show that the quality of the processed range image improves both in terms of number of available points and quality of the warped views.

In the following, Section 2 presents the structure of the MS Kinect sensor. Section 3 describes the denoising and interpolation algorithm in detail, with Subsection 3.3 reporting the depth correction algorithm and Subsection 3.6 showing how segmentation is employed to interpolate data. Experimental results (Section 4) and conclusions (Section 5) end the paper.

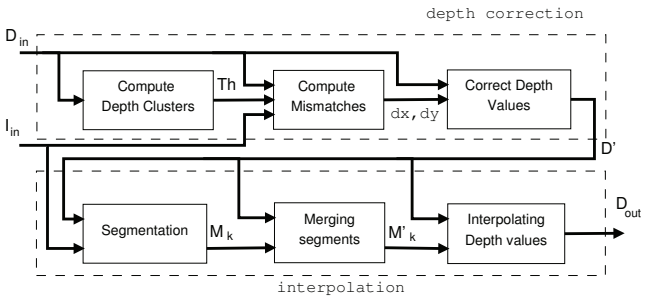

Fig. 2. Block diagram of the proposed algorithm.

2. A SHORT DESCRIPTION OF THE INFRARED STRUCTURED LIGHT SENSOR

To test the proposed approach we employed the MS Xbox Kinect device, a low-cost 3D sensor that is available on the market and exploits an IR structured light camera to estimate depth signals for the acquired scene. Despite this, the approach can be applied to any IR-based range camera. Figure 1 shows a simplified block diagram of the device. The implemented IR depth sensor consists in an IR projector, an IR CMOS camera, and a processing unit that controls them and elaborates the acquired signal. An IR pattern of dots is projected by the IR projector on the scene, and the IR CMOS camera acquires the reflected pattern, which will be distorted according to the geometry of the objects. The central processing unit estimates the distance of each point from the depth camera considering the distortions in the acquired dot pattern with respect to the projected one. Color information is available as well since an RGB CMOS camera permits obtaining a standard picture of the acquired scene.

This information permits building a pointcloud model of the 3D scene by mapping depth pixels into color pixels with a warping operation. The obtained 3D model presents several artifacts depending on possible calibration errors, lighting conditions and errors in depth estimation by the processing unit.

3. STRUCTURE OF THE PROPOSED ALGORITHM

The structure of the algorithm is summarized in Fig. 2 and consists in two main operating blocks: a denoising unit that corrects mismatches between the color image I_{in} and the warped depth map D_{in}, and an interpolating strategy that fills holes and missing pixels in the range image.

The following subsections will describe each step in detail.

3.1. Clustering depth values

At the beginning of the depth correction strategy, the depth values $D_{in}(x, y)$ (where (x, y) are the pixel coordinates) are clustered into a set of 20 classes C_k, $k = 0, \ldots, 19$, according to their distance from the IR camera using the k-means algorithm. The choice of using k-means algorithm and computing 20 classes was driven by the need of having a low complexity architecture .

Each class is characterized by its centroid and two threshold values that defines the upper and the lower bounds for depth values, that are grouped into the set of thresholds $\mathbf{Th}$.

3.2. Computing mismatches

At the beginning of the error correction unit, 3×3 Sobel operators S_x and S_y are applied to both the luminance component L of the color image and the warped depth image D_{in}. Let $S_x * L$ and $S_y * L$ be the convolutions of L with the horizontal and the vertical Sobel operators, respectively, and $S_x * D_{in}$ and $S_y * D_{in}$ be the convolution of the same operators with D_{in}. Then, the edge images E_L and E_D are computed as

$$\begin{aligned} E_L &= round\left(\frac{1}{64}\,\frac{|S_x * L| + |S_y * L|}{2}\right) \\ E_D &= round\left(\frac{1}{8}\,\frac{|S_x * D_{in}| + |S_y * D_{in}|}{2}\right) \end{aligned} \tag{1}$$

where quantization steps 8 and 64 have been chosen from a set of experimental trials. Coordinates (x, y) have been omitted for the sake of conciseness. For all the pixel positions (x, y) such that $D_{in}(x, y)$ is not valid, $E_D(x, y)$ is set to 0.

In a second step, the mismatches between I_{in} and D_{in} are computed for each class C'_k independently generating the pixel sets

$$C'_k = \{(x, y) \in C_k) : E_D(x, y) > 0\} \tag{2}$$

which comprises points in depth layer k with edge strength greater than 0. For each class C'_k, the algorithm computes the displacement vector $\mathbf{v}^* = [v_x^*, v_y^*]$ in the search window W_{SR} such that

$$\mathbf{v}^* = \arg \max_{\mathbf{v} \in W_{SR}} \sum_{(x,y) \in C'_k} E_L(x \boxplus v_x, y \boxplus v_y). \tag{3}$$

where the operator $\boxplus$ equals $+$ or $-$ depending on whether the coordinate x (or y) is higher or not with respect to the corresponding coordinate of the principal point $R = (R_x, R_y)$. In this way, the algorithm compensate the mismatch between edges of the color component and of the depth information (see Figure 3). Despite object profiles result irregular in the depth component, the algorithm assumes that errors vary symmetrically with respect to borders and maximizing edge matching permits a correct alignment.

3.3. Correcting depth values

The correction of the depth values obtained by the IR structured light camera can be obtained differentiating the equa-

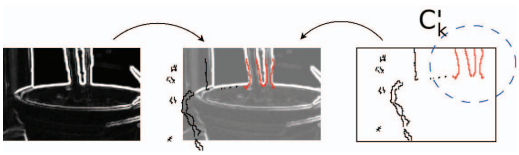

Fig. 3. Example of computation of $\mathbf{v}^*$ for the class C'_k (detail from the scene `bearbins`).

tions of the pinhole camera model

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 & C'_x \\ 0 & f_y & 0 & C'_y \\ 0 & 0 & 1 & C'_z \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \tag{4}$$

and combining it with the function that maps values $D_{in}(x, y)$ into distance values $z' = 100/(3.33 - 0.00307\ D_{in}(x, y))$. In our implementation, the algorithm corrects the positions of pixels in $\mathcal{C}_k$ and replaces the associated depth values $D_{in}(x, y)$ with the value

$$D'(x \boxplus v_x^*, y \boxplus v_y^*) = D_{in}(x, y) + \frac{\delta_x D(x, y) + \delta_y D(x, y)}{2} \tag{5}$$

where

$$\begin{aligned} \delta_x D(x, y) &= \frac{\boxplus v_x^* \cdot (1084 + D_{in}(x, y))}{(x - R_x)} \\ \delta_y D(x, y) &= \frac{\boxplus v_y^* \cdot (1084 + D_{in}(x, y))}{(y - R_y)}. \end{aligned} \tag{6}$$

At this point, the resulting depth map D' has to be extended in order to fill holes and gaps that lie wherever the depth values estimated by the sensor are not sufficiently reliable.

3.4. Segmentation

Like other algorithms for depth processing [5], our approach resorts to segmentation in order to partition the input depth map into segments where depth signal is assumed to be planar. As a matter of fact, it is necessary to oversegment the input images in order to avoid segments that include sharp edges or different objects inside themselves. In our approach we adopted the segmentation strategy in [6], which builds a grid of nodes associated to each pixel of the image. The weights of the edges equals the Euclidean distance including color and depth components (when depth is available).

The result of the algorithm is a map $\mathcal{M}$ of pixel regions M_k associated to each segment. The included depth values will be interpolated according to a planar model in order to fill all the holes. Before this operation, a further processing step is required.

3.5. Merging segments

In order to interpolate depth information, empty segments (i.e., not including a sufficient number of samples from D') need to be merged to one of the neighboring non-empty segments minimizing the MSE of the difference between the average color component of the two segments. The resulting segmentation map will be referenced as $\mathcal{M}'$.

3.6. Interpolating depth values

After refining the segmentation map, the valid depth values within segment M'_k can be interpolated in order to fill the missing values within the same segment. This operation relies on the assumption that the depth signal is approximately smooth within each segment M'_k. However, the borders of objects in the depth image are highly noisy and irregular, and therefore, some of the depth pixels could lie on a different segment M'_k.

The interpolation strategy has to select the pixel values concerning the object covered by M'_k and discard the extraneous ones. For each M'_k in $\mathcal{M}'$, the algorithm computes the variance of depth pixels and compares the resulting value with the threshold T_σ.

In case the variance is higher, it is possible that extraneous depth pixels are included in M_k and they have to be removed. To this purpose, k-means algorithm is run on depth pixels of M'_k partitioning the values $D'(x, y)$ into 3 classes. The algorithm will consider for the interpolation only those depth pixel within the most frequently chosen cluster. In case variance is lower, all the depth pixel in M'_k are used to fill holes.

After discriminating pixels to be interpolated, a polynomial regression is run on pertinent depth pixels, and the resulting coefficients are used to compute the missing pixel values. The resulting depth map will be named D_{out}.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, we built a stereo system consisting of an MS Kinect sensor and a side standard webcam that acquires images with resolution 320×240. Different 3D scenarios have been acquired under different light conditions. For each scene we performed 10 independent acquisitions in order to obtain different realizations for the noise signal on depth maps. The performance of the algorithm has been evaluated computing the average number of valid pixels in the final depth maps and the average PSNR obtained warping the pixels of Kinect color camera on the view corresponding to the side webcam. All the experimental data are available at [7].

Figure 4(a) shows the average PSNR values of the warped views for the original depth map D_{in}, the corrected depth map D', and the final depth map D_{out}, while Figure 4(b) reports the percentage of valid pixels for the different depth maps. The displayed results report also the performance of the interpolation strategy applied directly to the input depth map D_{in} (in this case the final depth map is referenced as D'_{out}). It is possible to notice that the interpolation strategy permits

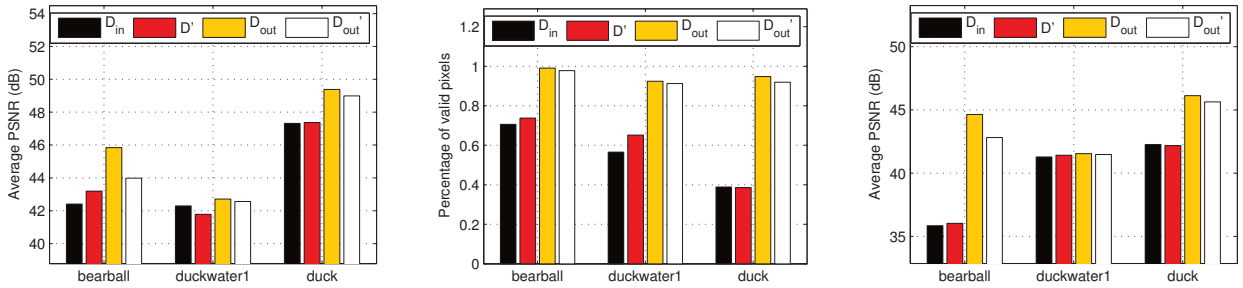

Fig. 4. Experimental results for `duck`, `duckwater`, `bearball` related to different depth maps.
a) Average PSNR of the warped view, b) percentage of valid depth pixels, c) Average PSNR obtained warping a common set of pixels.

Table 1. Experimental results for different scenes.

Scene	Ambient	Light	$\mathbf{E}[\Delta\text{PSNR}]$ (dB)	pix. incr. (%)
`coffee`	indoor	natural/reflections	+0.28	61.85
`director`	indoor	natural + neon	+0.45	51.04
`entering`	outdoor	natural/indirect	+0.02	67.70
`parking`	outdoor	natural/direct	+0.20	418.74
`peter`	indoor	natural + neon	+0.61	66.28
`soda`	indoor	natural/reflections	+0.70	78.36

filling most of the holes in the final depth map improving the quality of the depth values and reducing the amount of noise. The average PSNR for the warped view of the scene `bearball` is 3.7 dB higher using the depth map D_{out} with respect to D_{in}. No significant difference can be noticed for the scene `duckwater` since depth image is degraded by a strong amount of noise and the presence of transparent objects; as a matter of fact, no significant quality increment is possible. Note also that the interpolation strategy permits filling the holes within the image (from Fig. 4(b) it is possible to notice that the percentage of valid pixels is higher than 91 % for all the scenes). It is also possible to notice that the depth correction strategy proves to be effective for noisy sequences like `bearball`, where D_{out} permits improving the average PSNR of 1.7 dB with respect to D'_{out}. In the first case, corrected depth values are interpolated, while noisy depth values are employed for D'_{out}. This evidence can be noticed from the results in Figure 4(c), where the side view is obtained warping a common subset of depth pixels valid for all the maps. It is possible to notice that D_{out} permits obtaining significantly-higher PSNR values with respect to the other solutions.

In our experimental tests we evaluated the performance under diverse and uncontrolled light conditions. Table 1 reports the average PSNR increment $\mathbf{E}[\Delta\text{PSNR}]$ and the average relative increment of the percentage of valid pixels obtained by D_{out} with respect to D_{in}. The quality increment is lower since uncontrolled light introduces a stronger amount of noise on the acquired depth. However, the approach improves the quality of the final 3D model both in terms of quality and resolution (a visual evidence for the performance of the algorithm can be found at [7]).

5. CONCLUSION

The paper presents a joint denoising and interpolation approach for the MS Kinect sensor. The algorithm is based on an initial correction of depth values in the sequence, which then will be interpolated in order to fill holes and missing depth values. The proposed solution permits obtaining significant improvements for 3D models acquired under both controlled and uncontrolled light conditions.

6. REFERENCES

[1] S.B. Gokturk, H. Yalcin, and C. Bamji, "A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions," in *Proc. of CVPRW 2004*, June 27 – July 2, 2004, vol. 3, p. 35.

[2] IHS iSuppli, "The teardown: The kinect for xbox 360," *IET Engineering Technology*, vol. 6, no. 3, pp. 94 –95, Apr. 2011.

[3] B. Huhle, T. Schairer, P. Jenke, and Wolfgang W. Straíer, "Fusion of range and color images for denoising and resolution enhancement with a non-local filter," *Comput. Vis. Image Underst.*, vol. 114, pp. 1336–1345, Dec. 2010.

[4] E. Ekmekcioglu, M. Mrak, S.T. Worrall, and A.M. Kondoz, "Edge adaptive upsampling of depth map videos for enhanced free-viewpoint video quality," *Electronics Letters*, vol. 45, no. 7, pp. 353 –354, Mar. 2009.

[5] S. Milani and G. Calvagno, "A depth image coder based on progressive silhouettes," *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 711 –714, Aug. 2010.

[6] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167 – 181, Sept. 2004.

[7] S. Milani, "MS Kinect Denoising Test images - http://www.dei.unipd.it/∼sim1mil/materiale/kinect," 2011.