UPSAMPLING AND DENOISING OF DEPTH MAPS VIA JOINT-SEGMENTATION

Miguel Tallón[1], S. Derin Babacan[2], Javier Mateos[1], Minh N. Do[2], Rafael Molina[1], Aggelos K. Katsaggelos[3]*

[1]Departamento de Ciencias de la Computación e I.A. Universidad de Granada, Granada, Spain {mtallon,jmd,rms}@decsai.ugr.es

[2] ECE Department, and Beckman Institute, University of Illinois at Urbana-Champaign, IL USA {dbabacan,minhdo}@illinois.edu

[3]Electrical Engineering Computer Science Department Northwestern University, Evanston, IL USA aggk@eecs.northwestern.edu

ABSTRACT

The recent development of low-cost and fast time-of-flight cameras enabled measuring depth information at video frame rates. Although these cameras provide invaluable information for many 3D applications, their imaging capabilities are very limited both in terms of resolution and noise level. In this paper, we present a novel method for obtaining a high resolution depth map from a pair of a low resolution depth map and a corresponding high resolution color image. The proposed method exploits the correlation between the objects present in the color and depth map images via joint segmentation, which is then used to increase the resolution and remove noise via estimating conditional modes. Regions with inconsistent color and depth information are detected and corrected with our algorithm for increased robustness. Experimental results in terms of image quality and running times demonstrate the high performance of the method.

Index Terms— Time-of-flight cameras, depth enhancement, color segmentation, multisensor image fusion

1. INTRODUCTION

Obtaining accurate high resolution (HR) depth maps is crucial in a number of applications, including image based rendering, 3DTV, automotive applications, human-machine interfaces and gaming, robotics, among many others. Conventional methods to acquire depth information such as laser range scanners [1] or stereo vision algorithms [2, 3] either require static scenes or require long computation times preventing their use in real time applications.

An important recent development in depth map acquisition is the emerging low-cost and fast cameras for measuring depth [4]. With the development of these cameras, depth information can be captured at high speeds and can be incorporated in many applications due to their mobility. Unfortunately, their imaging capabilities are very limited compared to conventional color sensors: The acquired depth maps are low-resolution with a high noise level. For instance, the current ToF sensor 'Mesa Imaging SR4000' [5] offers a 176×144 depth map up to 50 frames per second (fps) and the structured light sensors included in the Microsoft Kinect [6] and Asus Xtion PRO LIVE [7] offer 640×480 at 30 fps captured simultaneously with a 1280×1024 RGB color image.

A number of post-processing methods have been developed to overcome these limitations. In [8] the authors proposed an extension of the bilateral filter, named Joint Bilateral Upsampling (JBU), to upsample the LR depth map with the guidance of the HR color image. The JBU filter operates simultaneously on the high and low resolution images. However, using the color image as a guide may in some regions lead to blurry edges and texture transfer from the color image to the depth map. NAFDU (Noise Aware Filter for Depth Upsampling) [9] tries to overcome these undesirable effects with a noise-aware filter which combines the original JBU filter with a filter designed to prevent artifacts in regions where JBU is likely to produce poor results. Joint Global Mode Filtering (JGMF) is proposed in [1] based on the joint histogram of color and depth images. It is shown that the solution is optimal with respect to l_1-norm minimization, and it can also be used to enforce temporal consistency for video depth enhancement. A Markov random field (MRF)-based approach is proposed in [10], where a HR depth map is obtained by finding the mode of the posterior distribution defined by the MRF. MRFs are also used in [11], reformulating both the data fidelity and smoothness terms and using loopy belief propagation to minimize the energy function. Although the image quality is high, the method requires manual tuning of a number of parameters.

In this paper, we propose a new method to obtain an enhanced HR depth map from a pair of a LR noisy depth map and a HR color image of the same scene. By applying joint segmentation on the color and depth images into regions of homogeneous characteristics (color and depth), we estimate the pixels in the HR depth image by the conditional modes within these regions. This leads to very effective denois-

Miguel Tallón performed the work during a visit at the Beckman Institute. This work was supported in part by the "Comisión Nacional de Ciencia y Tecnología" under contract TIC2010-15137.

ing while preserving object boundaries. Regions where the color information is inconsistent with the information from the depth map are detected and corrected. In addition, the method mitigates the undesirable effects caused by texture transfer and blurred edges.

The rest of the paper is organized as follows. In Sec. 2 we describe in detail the proposed algorithm. In Section 3 we compare our algorithm with other state-of-the-art methods and assess its quality and, finally, section 4 concludes the paper.

2. PROPOSED ALGORITHM

Let us denote by $\mathbf{Y}$ the HR color image and by $\mathbf{X}_\mathrm{L}$ the low-resolution depth map upsampled to the size of $\mathbf{Y}$ and aligned to it. Our goal is to obtain an enhanced depth map, $\mathbf{X}$, by applying a spatially varying denoising filter on $\mathbf{X}_\mathrm{L}$ which is designed via jointly segmenting the color image $\mathbf{Y}$ and the upsampled depth map $\mathbf{X}_\mathrm{L}$.

The proposed algorithm is summarized as follows. Both input images are divided into overlapping patches that will be processed independently in two stages. First, a joint segmentation is performed on the color and depth patches, denoted by $\mathbf{Y}^p$ and $\mathbf{X}_\mathrm{L}^p$ and a single depth value is assigned to each region of $\mathbf{Y}^p$. In the second stage, a refinement is applied to regions that contain multiple objects with similar colors but at different depth. These regions are detected and divided into subregions with different depth. Once all the patches are processed, they are merged together to form the final HR depth map.

Each step of the proposed algorithm, summarized in Alg. 1, will be described in the following sections.

Algorithm 1: Proposed Algorithm

Input: HR color image $\mathbf{Y}$, and upsampled depth map $\mathbf{X}_\mathrm{L}$.
$K_1 = 8$: number of classes in each color image patch.
$K_2 = 3$: number of classes in each depth patch.
σ^2_{tol}: depth refinement threshold.
Output: $\mathbf{X}$: HR depth map

```
// Preprocessing
```
$\mathbf{X}_G$ = Gaussian filtering of $\mathbf{X}_\mathrm{L}$
Divide images in overlapping patches.
```
// Main Algorithm
```
for *each patch* p **do**
  $\mathbf{C}^p_\mathbf{Y}$ = segmentation of $\mathbf{Y}^p$ into K_1 classes, representing each class by its centroid.
  $\mathbf{C}^p_{\mathbf{X}_G}$ = segmentation of $\mathbf{X}^p_G$ into K_2 classes, representing each class by its centroid.
  `// Joint-Segmentation`
  for *each region* $\mathbf{r}$ *in* $\mathbf{C}^p_\mathbf{Y}$ **do**
    $\mathbf{X}^p(i) = \mathrm{mode}_{i\in\mathbf{r}}(\mathbf{C}^p_{\mathbf{X}_G}(i))$, $\forall i \in \mathbf{r}$.
  end
  `// Depth refinement`
  for *each region* $\mathbf{r}$ *in* $\mathbf{X}^p$ **do**
    if $\mathrm{variance}_{i\in\mathbf{r}}(\mathbf{X}^p_\mathrm{L}(i)) > \sigma^2_{tol}$ **then**
      $\mathbf{X}^p(i)$ = segmentation of $\{\mathbf{X}^p_\mathrm{L}(j), j \in \mathbf{r}\}$ into K_2 classes, representing each class by its centroid, $\forall i \in \mathbf{r}$.
    end
  end
end
Postprocessing: Blend the obtained patches to form the final HR depth map $\mathbf{X}$.

2.1. Preprocessing

Our algorithm starts from an initial HR depth map $\mathbf{X}_\mathrm{L}$ obtained by bicubic interpolating the LR depth map to the size of $\mathbf{Y}$. We assume that, in small regions, the number of different objects or textures will be small so we will divide $\mathbf{Y}$ and $\mathbf{X}_\mathrm{L}$ into small overlapped square patches of size $B_x \times B_y$, denoted respectively by $\mathbf{Y}^p$ and $\mathbf{X}^p_\mathrm{L}$, and process them independently. Using overlapping patches will allow us to adapt to smoothly varying depths without creating false contours or blocking artifacts and, also, will allow to process each block in parallel with a considerable improvement in running time. Given an image of size $N_x \times N_y$ that will be divided in $p_x \times p_y$ patches with an overlapping factor of $overlap$, $0 \leq overlap < 1$, the size of each block can be computed as

$$B_z = \left\lfloor \frac{N_z}{(1-overlap)p_z + overlap} \right\rfloor, z \in \{x, y\}.$$

The size of the block is relatively important to the algorithm. If the size of the block is too big, texture transfer from the color image to the depth image might occur. On the other hand, if the block size is too small the obtained depth map will be noisy and not accurate. In our experiments we found that a block size around 20×20 pixels for an image size of 420×378 gives very good results. Examples of a block of the color image and the upsampled depth map are shown in Fig. 1a and 1b, respectively. The real unknown HR depth map patch, from Moebius dataset available in the Middlebury web site [3], is shown in Fig. 1h for reference.

Since the input depth map is quite noisy, before decomposing it into patches, we apply a Gaussian low pass filter on it to obtain a smoothed depth map, $\mathbf{X}_G$. The variance of the filter is obtained by searching for a flat area in the luminance of the color image, that is expected to conform a single object, and computing the variance in the corresponding area of the depth map. Taking advantage of the patch decomposition, we select, as flat area, the patch of the color image with the lowest variance. Fig. 1c shows the upsampled block in Fig. 1b after filtering. Note that the objective of this step is not to obtain a high quality depth map but to prevent noise from adversely affecting the segmentation process, which is described next.

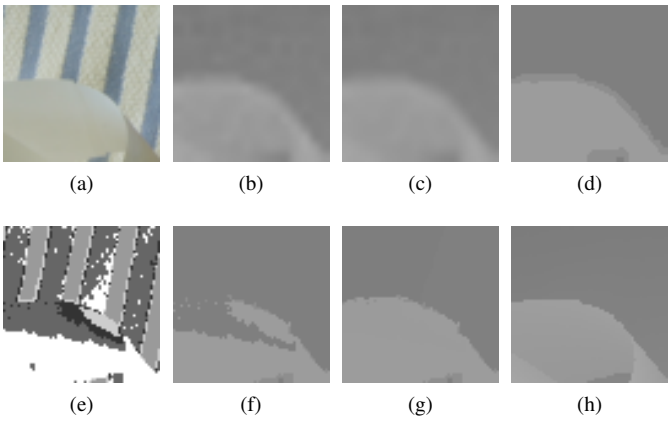


Fig. 1. (a) HR color image patch. (b) LR noisy depth map patch upsampled by bicubic interpolation. (c) Gaussian low pass filtered version of the patch in (b). (d) Depth map patch segmentation. (e) Color image patch segmentation. (f) HR depth map patch after initial classification. (g) HR depth map patch after refinement. (h) Ground truth depth map.

2.2. Joint segmentation

Each patch in $\mathbf{Y}$ and $\mathbf{X}_G$ is then processed to obtain areas with a homogeneous color that, usually, correspond to different objects. In most cases, each of those areas will be a part of an object. Areas of homogeneous color which are not part of only one object will be detected later. It is well known that there is not a direct correspondence between color and depth, that is, objects with the same color may be at different distances. However, in vast majority of the cases, regions of the same color correspond to the same object. Therefore the information on shape of the region from the color image can be used to accurately form the same region in the depth map with crisp edges and smooth values inside each region.

In order to obtain the areas with homogeneous color, we apply the standard k-means [12] clustering technique with the Euclidean distance metric to $\mathbf{Y}^p$, obtaining the segmentation of the patch $\mathbf{C}^p_{\mathbf{Y}}$. We use the HSV color space since we found that it provides better results than RGB, YCbCr and CIELab, especially in the object boundaries. An example of the segmentation of the color image patch in Fig. 1a is displayed in Fig 1e. The algorithm is not very sensitive to the number of classes if it is large enough to capture the different colors that include objects and textures. It is important to note that the k-means clustering algorithm will deliver empty clusters so that the final number of clusters may be smaller than selected. Experimentally we found that $K_1 = 8$ classes is appropriate.

We also apply the k-means algorithm to cluster the smoothed depth map patch, $\mathbf{X}^p_G$, into at most $K_2 = 3$ depth classes. Since we are working on small patches, we assume that the number of objects within a patch at different depths is small. After the segmentation, each pixel in $\mathbf{X}^p_{\mathrm{L}}$ is assigned the value of the centroid of its corresponding class, obtaining a depth patches $\mathbf{C}^p_{\mathbf{X}_G}$ with reduced noise. However, the borders of the objects may not be precise and they may present artifacts due to the noise and the blur introduced by the upsampling and filtering performed in the preprocessing stage as can be seen in Fig 1d.

To obtain a high resolution depth map with detailed edges, we make use of $\mathbf{C}^p_{\mathbf{Y}}$. For each segment $\mathbf{r}$ in $\mathbf{C}^p_{\mathbf{Y}}$, the most frequent depth value of the segment in $\mathbf{C}^p_{\mathbf{X}_G}$ is calculated and assigned to the set of pixels $\mathbf{r}$ in $\mathbf{X}^p$, that is,

$$\mathbf{X}^p(i) = \mathrm{mode}_{i\in\mathbf{r}}(\mathbf{C}^p_{\mathbf{X}_G}(i)),\ \forall i \in \mathbf{r}.$$

This process has two important properties. First, it is essentially a spatially-selective denoising filter, where the value of each pixel is estimated using its spatial neighborhood in both color and depth images. Second, it effectively merges small regions created during the segmentation of the color image due to textures or differences of color in the same objects. The merging occurs due to the similar depth values at these segments.

The resulting HR depth map patch, $\mathbf{X}^p$, will be accurate in most of the regions of the image. However, regions with homogeneous colors but different depths may be classified as a single depth region. An example is shown in the central part of Fig. 1f. In this case, the information of the color image does not help in the segmentation of the depth image and thus another step is necessary to detect and process these regions.

2.3. Depth map refinement

In order to check if each one of the above found regions $\mathbf{r}$ in $\mathbf{X}^p$ corresponds to a single depth, we calculate the variance of the pixels in $\mathbf{r}$ in the upsampled depth map $\mathbf{X}^p_{\mathrm{L}}$. Regions with objects at different distances are expected to have a higher variance. Thus, if the variance within a region is greater than a given threshold σ^2_{tol}, we further segment it in different regions, each one corresponding to a different depth. Note that, in this case, color image does not provide any information since the object color in the region will be very similar.

To further segment these regions, we apply a k-means clustering on the three dimensional space composed of the depth value obtained from $\mathbf{X}^p_{\mathrm{L}}$ and the horizontal and vertical coordinates of the pixels in the region $\mathbf{r}$. To avoid problems with the different ranges, the depth values are normalized to the interval $[0, 1]$ and the spatial coordinates x, y are normalized as $(x - min_x)/\sqrt{((max_x - min_x)^2 + (max_y - min_y)^2)}$ and $(y - min_y)/\sqrt{((max_x - min_x)^2 + (max_y - min_y)^2)}$, where max_x, min_x, max_y and min_y refer to the corresponding coordinates of the bounding box of the region. The intuition behind this clustering is to create clusters that have similar values but they are also spatially close to each other. This process creates compact regions, each one corresponding to one of the depth values present in the region. An

example of the result of this process, applied to the patch depicted in Fig. 1f, is shown in Fig. 1g.

2.4. Postprocessing

After all image patches are processed, we merge the HR depth map patches $\mathbf{X}^p$ using a normalized windowing function $\mathbf{w}^p$ as

$$\mathbf{X} = \sum_{p=1}^{P} \mathbf{w}^p \mathbf{X}^p,$$

where $P = p_x \times p_y$ is the number of patches and $\sum_p^P \mathbf{w}^p(j) = 1$, for $1 \leq j \leq (N_x \times N_y)$. We evaluated Gaussian, rectangular, triangular, and Hann windowing functions and empirically found that Hann windowing produces the best results without noticeable blocking artifacts and smooth depth values within the objects. This is particularly important to prevent the staircase effect that may appear if an object is not parallel to the image plane.

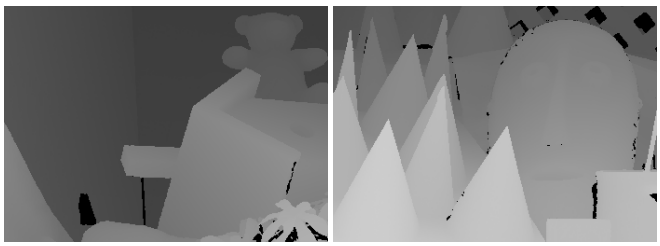
3. EXPERIMENTS

We evaluated the performance of the developed algorithm on the HR depth map and color image pairs from the Middlebury stereo database [2, 3]. Results are reported on the Teddy and Cones images, shown in Fig. 2, supplementary material can be found at http://decsai.ugr.es/pi/computationalphotography/depth_upsampling/. The LR depth maps, depicted in Fig. 3(a), are simulated by downsampling the ground truth depth map, using bicubic interpolation, by a factor of 4 in each direction and adding white Gaussian noise to obtain a signal-to-noise ratio (SNR) of 20 dB.

In all experiments, the number of patches was fixed to 26 and 34 in the vertical and horizontal directions, respectively, with an overlap percentage of 50%. We used eight classes for the color image segmentation and three for the depth map segmentation both in the initial segmentation and in the refinement stage. The variance threshold σ_{tol}^2, used to classify a region as homogeneous, is set to 100 in all experiments. We found that the algorithm is not sensitive to the value of this parameter and that selecting a threshold higher than the noise variance estimation is sufficient.

We compare the proposed algorithm with the state-of-the-art algorithms named JBU [8], NAFDU [9] and MRF [10]. All experiments are performed using non-optimized MATLAB implementations of the methods running on a Core 2 Duo laptop without parallel execution using GPUs. The resulting HR depth maps, depicted in Fig. 3(b) to 3(f), show that the proposed method removes the noise better than the competing methods and produces fewer artifacts around the object edges. In addition, it does not cause texture transfer between color and depth images. Quantitative evaluation of the results in terms of PSNR and SSIM are shown in Table 1.

(a)



(b)

Fig. 2. (a) Original color images. (b) Ground truth depth maps.

Table 1. Numerical comparison of the different methods.

Teddy				
Method	Time	Errors	PSNR	SSIM
JBU [8]	81.8	0.594	34.63	0.904
NAFDU [9]	115	0.580	34.74	0.915
MRF [10]	40.4	0.566	34.09	0.913
Proposed	38.4	0.423	35.56	0.954
Cones				
Method	Time	Errors	PSNR	SSIM
JBU [8]	83.3	0.709	35.48	0.845
NAFDU [9]	114.5	0.692	35.91	0.874
MRF [10]	40.7	0.681	35.26	0.885
Proposed	38.1	0.542	36.94	0.940

We also show the percentage of bad pixels (pixels with an absolute error in depth greater than 1), denoted by "Errors" in Table 1. Results show that the proposed method outperforms the competing methods by providing much fewer bad pixels and a higher SSIM, especially for the Cones image. Table 1 also shows the running times in seconds for each method. Our method is much faster than JBU and NAFDU and is comparable to MRF. The proposed method can also easily make use of parallel architectures to reduce the running times.

4. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method to efficiently combine high resolution color images and low resolution depth maps to obtain HR depth maps with suppressed noise and upsampling artifacts. The proposed method provides high quality depth maps and compares favorably to other state-of-the-art methods both in terms of image quality and running speed.

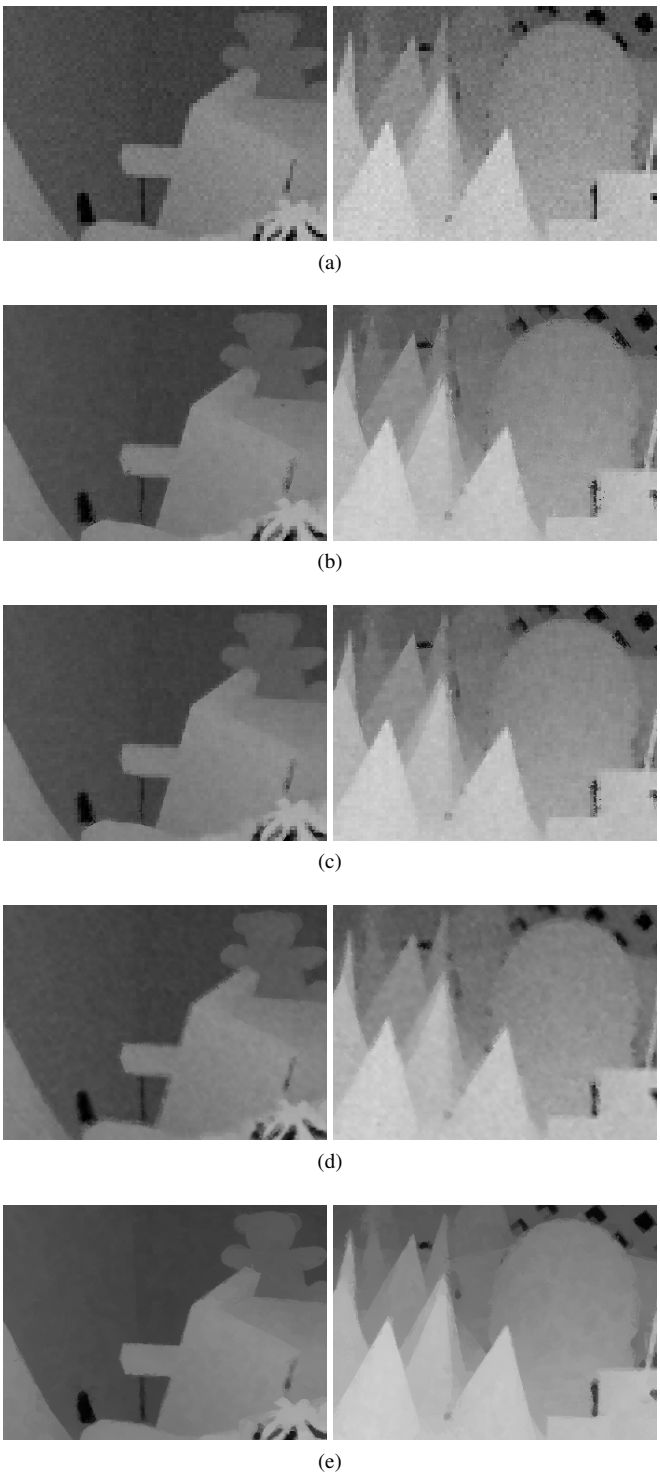


Fig. 3. (a) LR depth maps, upsampled by pixel replication. (b) Results with JBU [8]. (c) Results with NAFDU [9]. (d) Results with MRF [10]. (e) Results with the proposed algorithm.

Future work will focus on developing a GPU implementation of the method for real video processing.

5. REFERENCES

[1] D. Min, J. Lu, and M. N. Do, "Depth video enhancement based on joint global mode filtering," *IEEE Trans. on Image Processing*, Accepted for publication, 2012.

[2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. Journal of Computer Vision*, vol. 4, pp. 7–42, 2002.

[3] "Middlebury stereo web page," http://vision.middlebury.edu/stereo/.

[4] A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-Flight Sensors in Computer Graphics," in *Proc. Eurographics (State-of-the-Art Report)*, 2009.

[5] "Mesa Imaging SR4000 overview," http://www.mesa-imaging.ch/prodview4k.php.

[6] "Microsoft Kinect for Windows," http://kinectforwindows.org/.

[7] "Asus Xtion PRO LIVE," http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO.

[8] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. on Graphics*, vol. 26, pp. 96, 2007.

[9] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, "A noise-aware filter for real-time depth upsampling," in *ECCV Workshop on Multicamera and Multimodal Sensor Fusion Algorithms and Applications*, 2008, pp. 1–12.

[10] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *Conf. on Neural Information Proc. Systems (NIPS)*, 2005, pp. 291–298.

[11] Ji. Lu, D. Min, R. S. Pahwa, and M. N. Do, "A revisit to MRF-based depth map super-resolution and enhancement," in *Int. Conf. in Audio, Speech and Signal Proc. (ICASSP)*, 2011, pp. 985–988.

[12] G. F. Seber, *Multivariate Observations*, John Wiley & Sons, Inc., 1984.