This CVPR2014 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Depth Enhancement via Low-rank Matrix Completion

Si Lu[†] Xiaofeng Ren[‡] Feng Liu[†]
[†]Department of Computer Science Portland State University
[‡]Computer Science and Engineering University of Washington
lusi@pdx.edu xren@cs.washington.edu fliu@cs.pdx.edu

Abstract

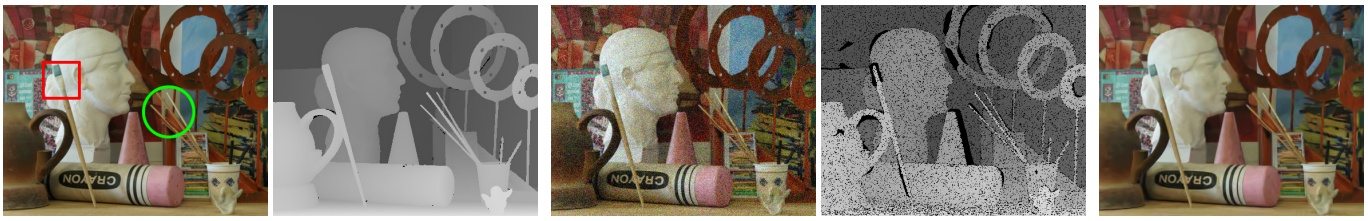
Depth captured by consumer RGB-D cameras is often noisy and misses values at some pixels, especially around object boundaries. Most existing methods complete the missing depth values guided by the corresponding color image. When the color image is noisy or the correlation between color and depth is weak, the depth map cannot be properly enhanced. In this paper, we present a depth map enhancement algorithm that performs depth map completion and de-noising simultaneously. Our method is based on the observation that similar RGB-D patches lie in a very low-dimensional subspace. We can then assemble the similar patches into a matrix and enforce this low-rank subspace constraint. This low-rank subspace constraint essentially captures the underlying structure in the RGB-D patches and enables robust depth enhancement against the noise or weak correlation between color and depth. Based on this subspace constraint, our method formulates depth map enhancement as a low-rank matrix completion problem. Since the rank of a matrix changes over matrices, we develop a data-driven method to automatically determine the rank number for each matrix. The experiments on both public benchmarks and our own captured RGB-D images show that our method can effectively enhance depth maps.

1. Introduction

The emerging consumer depth cameras, such as Microsoft Kinect and ASUS Xtion Pro, can now capture scene depth in real time, and enable a variety of applications that used to be challenging for computer vision and graphics. However, depth maps captured by these devices are often noisy and miss values at some pixels. For example, there are often missing depth values around object boundaries caused by the viewpoint disparity between multiple sensors in the depth camera. Moreover, for the depth cameras that use structured light, they often cannot provide accurate depth for dark surfaces due to light absorption.

Since depth cameras often capture color and depth (RGB-D) information of a scene, various methods have been recently developed to use the color image to enhance the corresponding depth map [30, 21]. These methods are developed based on two assumptions. First, the quality of the color image is high. Compared to the depth map, the color image obviously has no missing pixel values. Furthermore, it is less noisy than the depth map. Second, the depth image and its corresponding color image are highly correlated. For example, joint bilateral filter-based methods assume that pixels with similar colors tend to have similar depth values. These two assumptions, however, do not always hold in practice. When these depth cameras are used in a poor illumination environment, which is not uncommon in indoor scenes, the color image is often very noisy. The noisy color image can often mislead depth map enhancement and damage fine depth details. As shown in the green circle of Figure 1 (d), the smooth edges of the small sticks are damaged. One may suggest first de-noising the color image using a state-of-the-art color image denoising methods like BM3D [8], as shown in Figure 1 (c), and then using the denoised color image to guide depth map enhancement. However, such a method may still fail in practice because the correlation between color and depth does not always exist. For example, the white part of the brush head shares a similar color to the background, which misleads the joint bilateral filter to use the depth values from the background to fill in the missing depth values in the brush, as indicated in the red square area in Figure 1 (e).

In this paper, we present an approach to depth map enhancement that can use the noisy color image to simultaneously complete missing depth values and denoise the depth map. Our method is based on an observation that similar RGB-D patches approximately lie in a low-dimensional subspace. We can then assemble these RGB-D patches into a matrix and enforce the low-rank subspace constraint to achieve both de-noising and missing depth value completion. The subspace constraint essentially captures the potentially scene-dependent image structures in the RGB-D patches in both color and depth domain and thus our method is more robust than those methods that are based on the correlation between color and depth. Our method enforces



(a) Ground truth (b) Noisy RGB-D image (c) BM3D color denoising

(d) Joint bilateral filter (e) BM3D+Joint bilateral filter (f) Our depth enhancement result

Figure 1. Depth map enhancement.

the low-rank subspace constraint through incomplete matrix factorization. One technical challenge is that the actual rank number varies over the RGB-D patches depending on the image structures. We use a data-driven method to automatically estimate a right rank number for each patch matrix.

The main contribution of this paper is a low-rank matrix completion-based depth map enhancement method. This method can use noisy color images to guide depth map denoising and completion. We also contribute a data-driven approach for automatic rank estimation. We experimented with our method on public benchmarks and our own captured RGB-D images, and compared it to a range of depth map enhancement methods. Our experiments show that our method outperforms the state-of-the-art methods, even after we use the state-of-the-art color image de-noising method to clean the color images for these methods beforehand.

2. Related Work

A variety of methods have been developed for depth enhancement. These methods can be roughly categorized into depth map completion and depth super resolution. Denoising is often simultaneously handled during each of the two processes. This section first gives an overview on depth image completion, which is most relevant to this work. We then briefly overview depth super resolution.

Most existing methods compute the missing depth values using the corresponding color image based on the observation that there exists correlation between the depth map and the corresponding color image. A variety of joint bilateral filter-based methods have been developed based on this observation to use color images for hole filling [5, 20, 21]. Median filter has also been extended for depth image completion guided by the color image [18]. The fast marching algorithm was extended to incorporate the color image for filling depth values [17]. Wang *et al*. developed a joint color and depth in-painting method for stereo images [29]. Their method segments an image into small regions and estimates a depth plane for each region to fill in missing disparity (depth) values. Shen and Cheung presented a probabilistic model to capture various types of uncertainties in depth acquisition of a structured-light camera and used it to denoise and complete the depth map [27]. These methods can robustly fill in missing values in a depth map; however, their performance depends on the quality of the associated color images. The KinectFusion method from Izadi *et al*. uses depth maps from neighboring frames to fill in missing information during realtime 3D reconstruction [13]. When the camera is static, this fusion method cannot work well. Yu *et al*. recently developed a method that combines color and partial depth information to refine the depth map in a way that preserves structure [31].

A significant amount of research has been focused on depth super resolution. These methods typically simultaneously handle noise in the input. Similar to depth map completion, joint bilateral filter-based methods have been employed to make use of color images for depth super resolution and denoising [14, 30, 21]. Park *et al*. further extended the nonlocal means filtering algorithm to up-sample a low-resolution and noisy depth map guided by an auxiliary high-resolution color image [19]. Li *et al*. designed a hybrid camera system that captures color and depth images at different spatial-temporal resolution and used it for both color image motion deblurring and depth super resolution [15]. The recent method from Dolson *et al*. can upsample depth maps from dynamic scenes when combining the input color and depth images [9].

Schuon *et al*. fused multiple low-resolution depth maps of the same scene captured from different viewpoints into

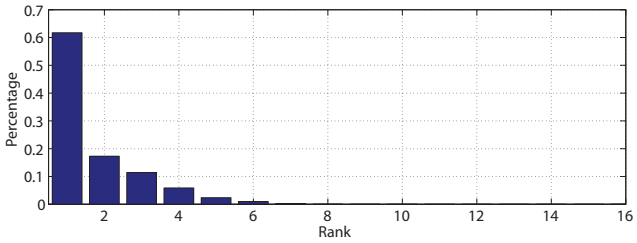


Figure 2. Histogram of the ranks of 30,000 RGB-D patch matrices that were randomly sampled from 3 RGD-D images.

a high-resolution one [26]. Multiple depth acquisition devices/methods have been combined to obtain high-quality depth maps. Zhu *et al*. combined passive stereo and time-of-flight sensors [32]. Castaneda *et al*. used two aligned time-of-flight cameras to capture two slightly different depth maps and merged them to achieve high depth quality [6]. Aodha *et al*. developed a data-driven approach to single depth image super resolution [1]. Hornáček *et al*. used a 3D patch matching method to find similar patches from a single depth image for depth super resolution [11].

3. Depth Map Denoising and Completion

Our method takes a noisy depth image D and its corresponding color image I as input. The noisy depth image D also misses values at some pixels. Our method both denoises the depth image and completes the missing depth values. Our method is based upon the observation that similar image patches lie in a low-dimensional subspace. Below we first examine this observation empirically, and then describe our depth enhancement method.

3.1. Low-rank Subspace Constraint

Given a reference RGB-D image patch P_r with size $m \times m$, we find its $k-1$ nearest neighbors $\{P_i\}$ in the same input image using the distance metric defined in Equation 2. We then subtract each patch with the average of all these k patches and assemble the resulting k patches into a patch matrix M of size $4m^2 \times k$, where each column corresponds to one of the patches. We observe that this patch matrix can be well approximated by a matrix with a very low rank. That is, the similar matches approximately lie in a very low-dimensional subspace.

To empirically verify this observation, we collected a set of images from the Middlebury stereo vision benchmarks [25, 23, 10] and randomly sampled 30,000 reference patches, each with size 7×7. Here, we considered disparity from this benchmark as depth. We built a corresponding patch matrix with $k = 40$ patches for each reference patch as described above and normalized all its values to [0, 1]. As there is noise in the images, we estimate the rank of each patch matrix using SVD as follows,

$$\hat{r} = \arg\min_r f_e(r-1) - f_e(r) \leq \epsilon \qquad (1)$$

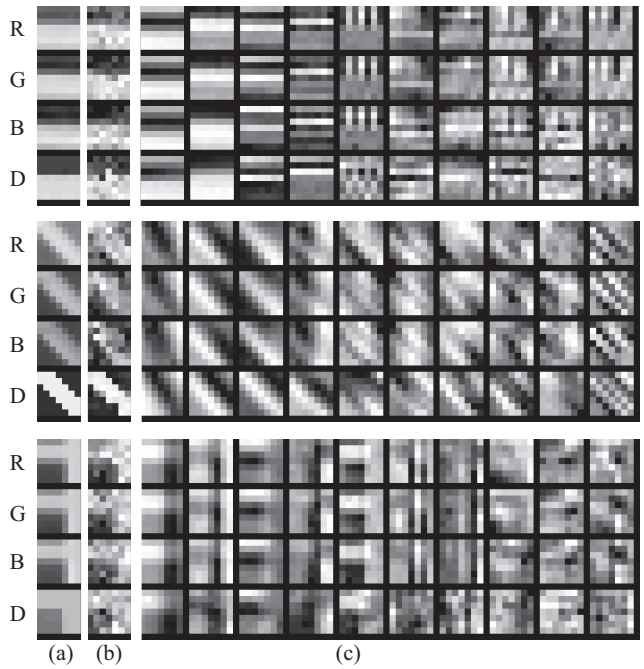


Figure 3. Patch samples. (a): clean patch. (b): noisy patch. (c): the top ten eigen vectors of the patch matrix formed by similar patches to each noisy patch. These eigen vectors are ordered according to the decreasing order of the corresponding singular values. For each patch, we show its **R**, **G**, **B**, and **D** channels separately, each as one row. We can see that the first few eigen vectors capture the structure of each patch matrix.

where $f_e(r)$ is the root mean square error between the patch matrix and its rank-r approximation obtained using SVD and ϵ is the error-decreasing threshold with value 0.015%. Figure 2 shows the rank histogram of these 30,000 patch matrices. 99.7% of these matrices can be approximated by a matrix with rank below 7. We also find that the rank varies over individual patch matrices. We develop a data-driven method to determine the rank, as described in Section 3.3.

The subspace constraint essentially captures the underlying structure in the RGB-D patches. Figure 3 shows three reference patches and the first ten eigen vectors of their patch matrices. While each reference patch shown in Figure 3 (b) is very noisy, the first few eigen vectors of the patch matrix formed by its similar patches can well capture the underlying structures, as shown in Figure 3 (c).

3.2. Depth Enhancement

Our method explores the low-rank subspace constraint of the patch matrix for depth image denoising and completion. Consider a patch P_r with missing depth values at some pixels. We first find its $k-1$ nearest neighbors $\{P_i\}$. We could compute the L2 norms-based distance between two patches. However, this method has a problem as it treats depth in the same way as color. A large difference between two patches in their color components indicates that these

(a) Image (b) Depth

Figure 4. Patches with similar local 3D structures. Although the three patches marked with green squares are at three different depth layers, they share a similar local 3D structure and thus should be considered as similar patches.

two patches have different colors. A large difference between two patches in the depth component, however, does not necessarily mean that they have different local 3D structures. As shown in Figure 4, while the three patches marked with green squares are at different depth layers, their 3D structures are very similar to each other and thus can be considered as similar patches.

We therefore shift each patch in depth when we compute the patch distance as follows,

$$d(P_r, P_i) = \alpha\|I_r - I_i\| + \beta\|(D_r - \bar{D}_r) - (D_i - \bar{D}_i)\| \quad (2)$$

where I_r and D_r are the color and depth components of patch P_r. $\bar{D}_r$ is the median depth value of patch D_r. α and β are the parameters to weight the color and depth distance, with default value 0.4 and 30, respectively. Since the depth component of a patch is often incomplete, we first assign an initial depth value $D^{pre}(p)$ for each depth-missing pixel p using a nonlocal means method before we compute the patch distance as follows,

$$D^{pre}(p) = \frac{1}{\sum_{q\in\Omega_p} w_{p,q}} \sum_{q\in\Omega_p} w_{p,q} D(q) \quad (3)$$

where Ω_p is the neighborhood of pixel p, and $w_{p,q}$ is a weight consisting two components

$$w_{p,q} = \exp - \frac{\|G \odot (P_p - P_q)\|_F^4}{2\sigma^2 n_G^2} \quad (4)$$

where $\|\cdot\|_F$ is the Frobenius norm. $\odot$ means component-wise multiplication. P_p and P_q are two patches centered at p and q. G is a mask matrix. It takes value 0.84 for elements corresponding to the color channels. For the depth channel, it takes value 0 when either P_p or P_q has a missing depth value and 0.55 otherwise. n_G is the number of none-zero values in G. σ is a parameter with default value 100.

Once we find $k-1$ nearest neighbors in total, we assemble them together with the reference patch into a patch matrix $M_{4m^2\times k}$. M can be incomplete and the pre-inpainting step above is only used to find the nearest neighbors. Denote a clean and complete version of this matrix as $\hat{M}_{4m^2\times k}$. Since $\hat{M}$ is of low rank, it can be decomposed into two low-rank sub-matrices A and B as follows,

$$\hat{M}_{4m^2\times k} = A_{4m^2\times r} B_{r\times k} \quad (5)$$

where r is the rank of $\hat{M}$ and can be determined using the method described in Section 3.3. We can then formulate depth completion and denoising as an incomplete matrix factorization problem that aims to recover A and B from the noisy and possibly incomplete input M. We formulate this matrix factorization problem as follows,

$$\min_{A\in\mathcal{R}^{4m^2\times r}, B\in\mathcal{R}^{r\times k}} \|W \odot (M - AB)\|_F^2 \quad (6)$$

where W is a mask matrix, with 1 and 0 indicating existing and missing data, respectively. If M is complete, the above factorization problem can be solved by SVD. Otherwise, it can be solved using incomplete matrix factorization methods [4, 7]. We experimented with these algorithms and found that the alternation method [3] works well here. It can produce comparable results to those non-linear methods such as the Damped Newton algorithm [4]. One reason is that M has only a very few number of missing values. We therefore use the alternation algorithm in our method for its efficiency. Once we find A and B, we can recover the clean and complete patch matrix $\hat{M}$ as their product. Note, this matrix factorization step cleans and completes all the patches in the patch matrix M. All these patches can be used as depth enhancement results collaboratively.

For depth map completion, we only need to assemble a patch matrix and run matrix factorization at pixels with missing values. We can also run this process at each pixel to achieve depth map denoising. Since a pixel can be covered by multiple patches, which are all enhanced, we aggregate the enhanced patches in a similar way to collaborative filtering [8]. Specifically, to obtain the final depth enhancement result at each pixel, we retrieve its values in all the patches that cover it and have been used in and recovered by matrix factorization, and average them as the final depth value.

3.3. Rank Estimation

As described in Section 3.1, the patch matrix rank varies over matrices. We observe that the rank typically depends on the RGB-D patch structure. We therefore design a series of features to capture the patch structure and then use a regression method to estimate the rank for each patch matrix.

Our method detects edges in both the color and depth image to capture the patch structure. Since the input images are noisy, we use a robust edge detector [16]. Then for each patch P_i, we obtain a color edge map E_i^c and depth edge map E_i^d. For simplicity, we arrange these edge maps into vectors and use the same notations to represent them. We use the following features to capture the patch structure.

Method	Correct	Error ≤ 1	Error ≤ 2
Parabola regression	69.2%	93.9%	98.6%
Linear regression	68.3%	93.6%	98.6%
SVM	53.0%	84.0%	91.8%
KNN	67.3%	88.3%	95.6%

Table 1. Rank estimation.

a. Number of edge pixels.

$$F_1 = \sum_{i=1}^{k} |E_i^c|_1 \text{ and } F_2 = \sum_{i=1}^{k} |E_i^d|_1$$

b. The edge difference between the nearest patch and the furthest patch.

$$F_3 = |E_1^c - E_k^c|_1 \text{ and } F_4 = |E_1^d - E_k^d|_1$$

c. The variance of the edge differences.

$$F_5 = var(|E_2^c - E_1^c|_1, |E_3^c - E_1^c|_1, \cdots, |E_k^c - E_1^c|_1)$$
$$F_6 = var(|E_2^d - E_1^d|_1, |E_3^d - E_1^d|_1, \cdots, |E_k^d - E_1^d|_1)$$

d. The pixel value variance within the average patch.

$$F_7 = var(\bar{P}^r(1), \bar{P}^r(2), \cdots, \bar{P}^r(m^2))$$
$$F_8 = var(\bar{P}^g(1), \bar{P}^g(2), \cdots, \bar{P}^g(m^2))$$
$$F_9 = var(\bar{P}^g(1), \bar{P}^b(2), \cdots, \bar{P}^b(m^2))$$
$$F_{10} = var(\bar{P}^d(1), \bar{P}^d(2), \cdots, \bar{P}^d(m^2))$$

where $\bar{P}$ is the average patch, and the superscript r, g, b, and d indicate the color and depth channels.

We assemble the above features into a feature vector F for each patch matrix and train a parabola regression model to predict the rank. We collected a set of 43,200 patch matrices. We randomly partitioned half of the set into a training set and the other half as a testing set, and used them to test our prediction method. We repeated this process 100 times, and reported the performance of our method in Table 1. We also compared the parabola regression model with the linear regression model, the K-Nearest-Neighbor Multi-Classifier, and the multi-class SVM. We found that the parabola regression model works best. It can predict the rank with error no more than 1 for 93.9% of the patch matrices.

4. Experiments

We experimented with our depth completion and denoising method on two public dataset, namely the Middlebury stereo dataset [24, 25, 23, 10], and the RGBZ dataset from [21]. For the Middlebury dataset, we used the left color images and the corresponding disparity images of 30 stereoscopic images. In our experiment, we simply used disparity as depth. From the RGBZ dataset, we randomly chose one frame from each of its 9 RGBZ videos. This RGBZ dataset does not provide the ground truth, and we took the enhancement result from [21] as the ground truth for our experiments. In our experiments, we added Zero-Mean-White-Gaussian noise with stand deviation 25 and 5 to the color image and depth image respectively. We also set around 13.0% pixels with unknown depth values. To simulate the spatial distribution of unknown depth values from a consumer RGB-D camera, we used importance sampling to select more pixels around edges with unknown depth values. We also manually selected pixels around depth edges and dropped their depth values.

We compared our method to various methods that focus on depth completion for RGB-D images, including Joint bilateral filter (JBF) [14, 21], Nonlocal means filter (NLM) [12],Structure-guided fusion (SGF) [20], Spatio-temporal hole filling (SHF) [5], and Guided inpainting and filtering (GIF) [17]. For JBF and SHF, we modified their method to handle single input image. We found that the performance of these other methods was significantly compromised by the noisy color images. To have a more meaningful comparison, we first employed a state-of-the-art color image denoising method, BM3D [8], to pre-process the color images and then apply these methods for depth enhancement. Note that in this test, our method still directly took the noisy color images as input. We report the quantitative results in Figure 5. Overall, our method performs best among all these algorithms with respect to the PSNR values. The average PSNR value of our method is 39.3 while those for the other five methods are 37.9, 37.2, 33.9, 36.5 and 37.0, respectively.

Figure 6 shows some depth enhancement results. As shown in Figure 6 (h) and (p), our method can better preserve or recover image details, such as depth edges. Our method performs particularly well in recovering small objects, such as the slim leaf indicated by the red rectangles in the second example. The leaf has a very similar green color to its neighboring background, which is challenging for other methods. As shown in Figure 6 (p), our method can better recover it. We show more qualitative comparisons on these testing images as well as some RGB-D images captured by ourselves in our project website[1].

4.1. Discussion

The selection of the rank is important for our method. While Table 1 reported the performance of our rank estimation method, it is more important to examine how our automatic rank estimation helps depth enhancement. We therefore experimented with our method with the predicated ranks and with fixed ranks. Since the majority of patch matrices have a rank smaller than 8 as shown in Figure 2, we tested our method with fixed ranks from 1 to 8. We reported the average PSNR on all the 39 testing images in Figure 7. This experiment shows that our automatic rank selec-

[1] http://graphics.cs.pdx.edu/project/depth-enhance

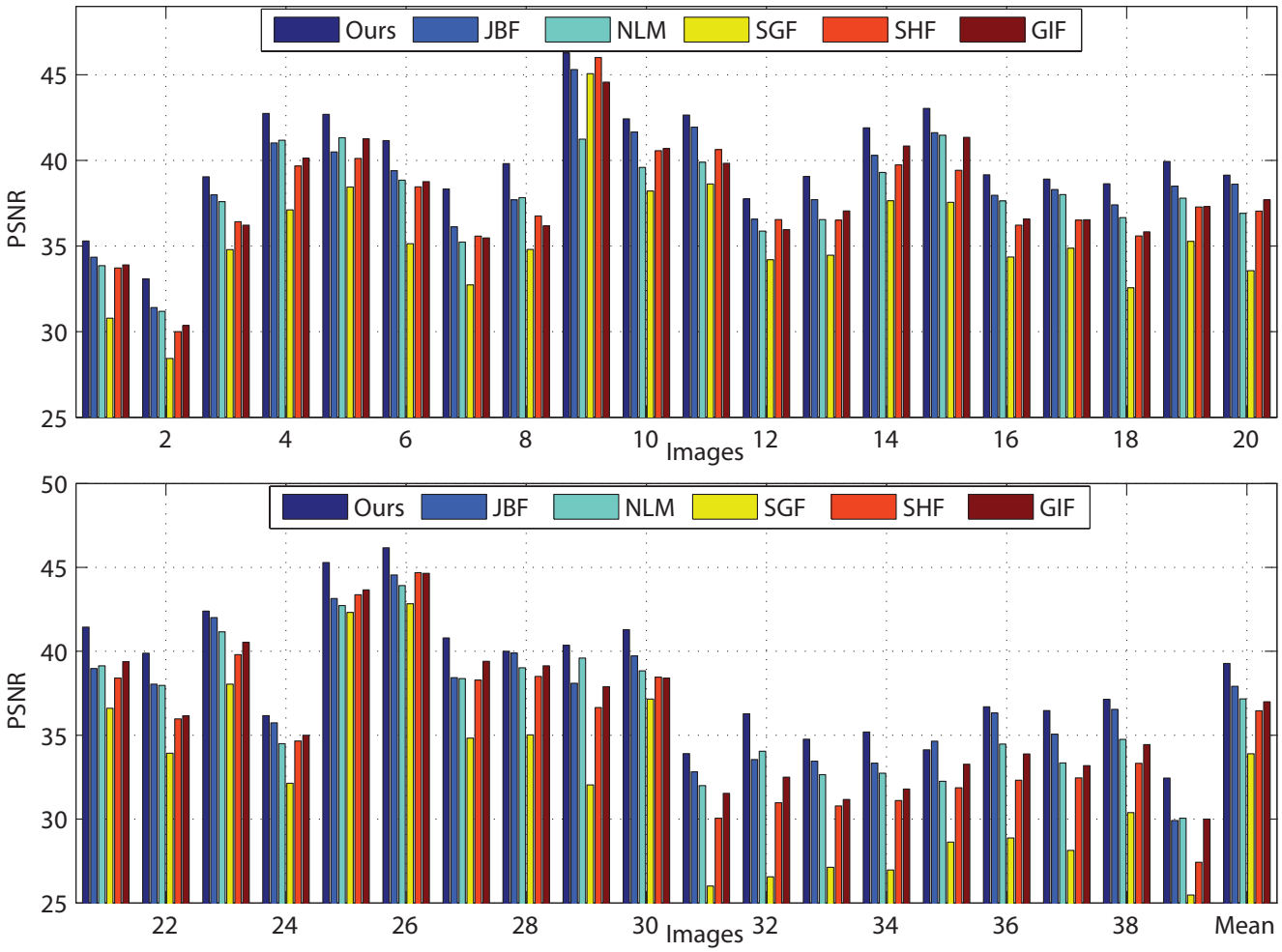


Figure 5. Comparisons among depth enhancement methods. We show the PSNR values of these methods on each of the testing images as well as the average PSNR value (the last column in the bottom figure).

tion method helps depth enhancement. Since most patches take rank 1, 2 or 3 and the small difference in rank leads to small PSNR change, automatic rank estimation only improves fixed ranks 2 and 3 marginally. On the other hand, our method has already achieved a PSNR value (39.32) very close to that using optimal (groundtruth) ranks (39.34).

The number of patches and the patch size in our experiments are 40 and 7x7. Our experiments show that our method is stable with reasonable parameter values. The average PSNR values for k=30, 40, and 50 are 39.25, 39.32, and 39.31. The PSNR values for the patch size 5x5, 7x7, and 9x9 are 38.93, 39.32, and 39.33.

Our method directly takes a noisy color image as input to help depth enhancement. It is interesting to test if the performance of our method can be further improved by denoising the color images using a state-of-the-art method before using them in our method. In our test, we used BM3D [8] to denoise the color images and then used the denoising results for depth enhancement in our method. Figure 8 shows the depth enhancement results using our method with and without applying BM3D to color images. Interestingly, BM3D helps our method insignificantly. This is consistent with our further tests that show the PSNR value of our result drops slowly with higher noise levels until the color and depth noise level are over 50 and 10, respectively.

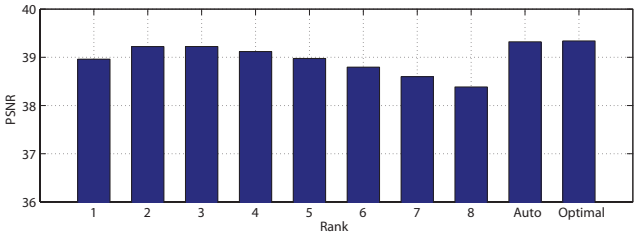
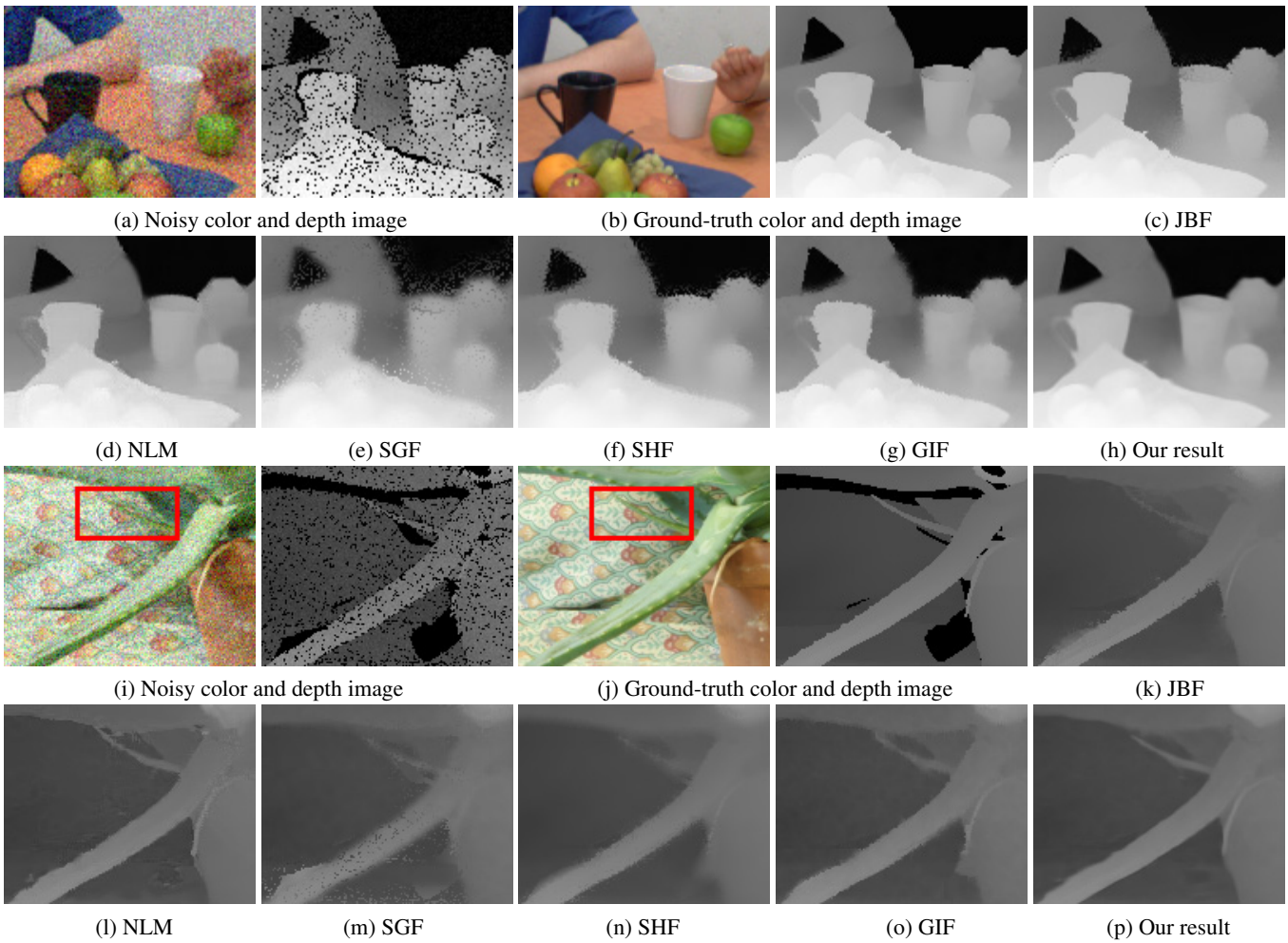


Figure 7. Automatic rank selection vs. Fixed rank.

Since our method denoises R,G,B,D channels simultaneously through (incomplete) matrix factorization, a by-product is color image de-noising although this is not intended in our work. We compared our color image denoising results to the state-of-the-art color denoising methods on the same set of RGB-D images used above. As reported



(a) Noisy color and depth image (b) Ground-truth color and depth image (c) JBF

(d) NLM (e) SGF (f) SHF (g) GIF (h) Our result

(i) Noisy color and depth image (j) Ground-truth color and depth image (k) JBF

(l) NLM (m) SGF (n) SHF (o) GIF (p) Our result

Figure 6. Comparisons among depth enhancement methods. All these images were cropped from the full-size versions for clarity.

Method	BM3D	BF	NLM	EPLL	FOE	Ours
PSNR	32.3	28.8	30.2	31.3	29.9	31.2

Table 2. Color image denoising.

in Table 2, while our method does not perform as well as BM3D, it is comparable to EPLL [33], and outperforms bilateral filter (BF) [28], nonlocal means filter (NLM) [2], and the fields of experts method (FOE) [22].

Our method estimates missing depth values from sufficient similar pathes with available depth values. When a large region misses depth values completely, few repetitive patches are available for a rare image structure and then our method may fail.

The computational cost of our method mainly consists of two parts, similar patch searching and patch matrix factorization, which need to be run at every pixel. To improve the speed, our implementation did not execute these two steps at every pixel. Instead, we uniformly selected 1 out of 16 pixels to process as the patches recovered from processing these pixels also cover the pixels that are not sampled. Then the final collaborative filtering step described in Section 3.2 can already compute the results for those pixels that are not sampled. Our system took about 1.5 minutes to process an image with size 430×370 on a desktop machine with Intel i7 CPU using a single thread. Since matrix formulation and factorization at pixels can be parallelized, our algorithm can be speeded up significantly with parallel computing.

5. Conclusion

In this paper, we presented a method for depth enhancement that can simultaneously achieve missing depth value completion and depth denoising. Our method is based on an observation that similar RGB-D patches lie in a low-dimensional subspace. Our method assembles similar patches into a patch matrix and formulates depth enhancement as an incomplete matrix factorization problem. Our method shows that our method can enhance noisy and incomplete depth maps even with noisy color images.

Acknowledgements. This work was supported by NSF grants IIS-1321119, CNS-1205746, and CNS-1218589.

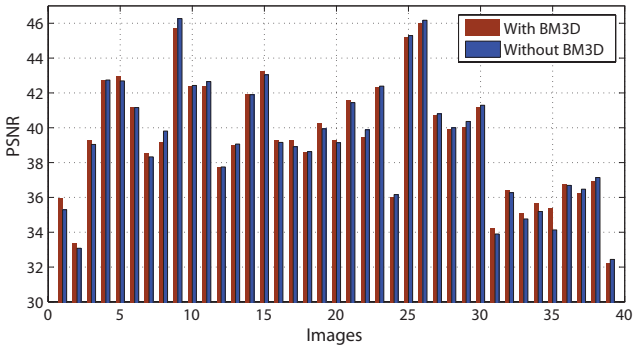


Figure 8. Our method with and without BM3D.

References

[1] O. M. Aodha, N. D. F. Campbell, A. Nair, and G. J. Brostow. Patch based synthesis for single depth image super-resolution. In *ECCV*, pages 71–84, 2012.

[2] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *IEEE CVPR*, pages 60–65, 2005.

[3] A. M. Buchanan. Investigation into matrix factorization when elements are unknown. Technical report, University of Oxford, 2004.

[4] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *IEEE CVPR*, pages 316–322, 2005.

[5] M. Camplani and L. Salgado. Efficient spatio-temporal hole filling strategy for kinect depth maps. In *SPIE Electronic Imaging*, 2012.

[6] V. Castaneda, D. Mateus, and N. Navab. Stereo time-of-flight. In *IEEE ICCV*, pages 1684–1691, 2011.

[7] P. Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *Int. J. Comput. Vision*, 80(1):125–142, 2008.

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Imgage Processing*, 16(8):2080–2095, 2007.

[9] J. Dolson, J. Baek, C. Plagemann, and S. Thrun. Upsampling range data in dynamic environments. In *IEEE CVPR*, pages 1141–1148, 2010.

[10] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *IEEE CVPR*, 2007.

[11] M. Hornáček, C. Rhemann, M. Gelautz, and C. Rother. Depth super resolution by rigid body self-similarity in 3d. In *IEEE CVPR*, pages 1123–1130, 2013.

[12] B. Huhle, T. Schairer, P. Jenke, and W. Strasser. Robust non-local denoising of colored depth data. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2008.

[13] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *ACM UIST*, pages 559–568, 2011.

[14] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM Trans. Graph.*, 26(3):96:1–96:6, 2007.

[15] F. Li, J. Yu, and J. Chai. A hybrid camera for motion deblurring and depth map super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[16] D. H. Lim. Robust edge detection in noisy images. *Comput. Stat. Data Anal.*, 50(3):803–812, 2006.

[17] J. Liu, X. Gong, and J. Liu. Guided inpainting and filtering for kinect depth maps. In *IEEE International Conference on Pattern Recognition*, pages 2055–2058, 2012.

[18] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov. Temporal filtering for depth maps generated by kinect depth camera. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, 2011.

[19] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon. High quality depth map upsampling for 3d-tof cameras. In *IEEE ICCV*, pages 1623–1630, 2011.

[20] F. Qi, J. Han, P. Wang, G. Shi, and F. Li. Structure guided fusion for depth map inpainting. *Pattern Recogn. Lett.*, 34(1):70–76, 2013.

[21] C. Richardt, C. Stoll, N. A. Dodgson, H.-P. Seidel, and C. Theobalt. Coherent spatiotemporal filtering, upsampling and rendering of rgbz videos. *Comp. Graph. Forum*, 31(2):247–256, 2012.

[22] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *IEEE CVPR*, pages 860–867, 2005.

[23] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *IEEE CVPR*, 2007.

[24] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002.

[25] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE CVPR*, pages 195–202, 2003.

[26] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. Lidarboost: Depth superresolution for tof 3d shape scanning. In *IEEE CVPR*, pages 343–350, 2009.

[27] J. Shen and S.-C. S. Cheung. Layer depth denoising and completion for structured-light rgb-d cameras. In *IEEE CVPR*, pages 1187–1194, 2013.

[28] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *IEEE ICCV*, pages 839–846, 1998.

[29] L. Wang, H. Jin, R. Yang, and M. Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *IEEE CVPR*, 2008.

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

[31] L.-F. Yu, S.-K. Yeung, Y.-W. Tai, and S. Lin. Shading-based shape refinement of rgb-d images. In *IEEE CVPR*, pages 1415–1422, 2013.

[32] J. Zhu, L. Wang, R. Yang, J. E. Davis, and Z. pan. Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(7):1400–1414, 2011.

[33] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *IEEE ICCV*, pages 479–486, 2011.