ASYMMETRIC STEREO VIDEO SUPER-RESOLUTION USING CONVOLUTIONAL NEURAL NETWORKS

Weiyuan Bao, Shahram Shirani

ABSTRACT

In the asymmetric stereo video, the resolution of the left view and the right view are different. One of the main motivations for utilizing asymmetric stereo video is to reduce the bit rate required to transmit stereo video. However, during playback, both views ideally should be at the same resolution. This is achieved by up-scaling the low-resolution view to the resolution of the other view. In this paper, we proposed a learning-based approach for the asymmetric stereo video super-resolution. Unlike other methods, our proposed method does not require any environment parameters or depth information. After view adjustment, we feed the asymmetric stereo frame pairs to the convolutional neural network simultaneously to establish pixel correlation between them and to recover the missing high-frequency details of the low-resolution view. The test on stereo frames shows promising results on asymmetric stereo video super-resolution.

Index Terms— super resolution, mix-resolution, disparity map, convolutional nerual network

1. INTRODUCTION

The asymmetric stereo video super-resolution is the process of taking low-resolution frames of one view and their stereo high-resolution frames of the other view to upscale the low-resolution frames. Asymmetric stereo video has proven to be an effective way of stereo video compression[1]. When the videos are encoded, frames of one view are down-sampled in order to reduce the overall size of the file. The low-resolution frames will be up-scaled to full-resolution at the decodior side. While displaying, human eyes fuse the two images and the resulting quality is similar to the full-resolution view[2][3].

One of the solutions for asymmetric stereo video super-resolution, which actually is a solution for a more general problem, is performing the single image super-resolution(SISR) on the low-resolution image alone. When the low-resolution image is downsampled heavily from its native resolution, high-frequency details are hugely eliminated. As a result, high-resolution pixels recovery presents a more challenging problem. The result high-resolution image from upscaling can display distorted and inaccurate when compared to the ground truth image.

There are early methods which are developed for SISR using sparse-coding techniques [4] [5]. More recently, there are break-throughs in the deep learning field to utilize learning-based approaches for SISR. The early attempts are using neural networks, which leads to the use of multi-layer cascading autoencoders. Another successful approach, provided by Kim [6], is embedding sparse-coding in the layers of a neural network. A state-of-art method, proposed by Dong et al. [7], is utilizing a fully convolutional neural network to establish the relationship between the low-resolution pixels with the high-frequency pixel of the ground truth image.

These methods developed for SISR have been extended to stereo and multi-view video super-resolution. Inspired by Dong et al.'s approach, fully convolutional neural network is used on multi-view image and video super-resolution. [8] proposes to use a fully convolutional neural network to solve the mapping between low-resolution pixels and high-resolution pixels from the neighboring full-resolution image. Based on Depth-Image-Based-Rendering (DIBR) technique proposed by [9], they are able to project the high-resolution view to the viewpoint of the low-resolution image and construct the full-resolution virtual view. Then, both images are fed into the CNN to learn the mapping between the low-resolution image and the neighboring high-resolution image. They are using only 17 pairs of multiview images as the training set, which us .

However, sometimes, the depth information and camera parameters are not available or can be inaccurate. We believe there is a method to do stereo frames super-resolution without using any depth information or camera parameters.

In our approach, inspired by the infrastructure of the SR-CNN and the model constructed to perform super-resolution on stereo video, we craft a similar CNN with according pre-processing steps. We add one extra channel to the model of [7] for the input of the high-resolution stereo image. Our model is trained on 65 stereo image pairs and tested on other images and video sequences. After testing the test set with images/video sequences, we observed promising and stable results.

The remaining of the paper is structured as the following: section II lists related work done by other people; section III demonstrates and discusses our proposed methods in detail; section IV provides experimental procedures and results; section V concludes our paper.

2. RELATED WORK

2.1. Single image SR

The previous methods used to solve SISR problems are using example-based mapping based on self-similarities of the images [6] [10] [11]. To be able to build the dictionary to solve such a mapping from low-resolution pixels in RGB or YCbCr, they all need a relatively large amount of training data. Certain image priors will be needed as well. Methods via Sparse Representation [6] have later been used and achieved a good result on SISR problems, which can be treated as an early learning-based method. By using CNN, [7] achieves the state-of-art result for SISR by using the powerful concept of deep learning. By them, a three-layer neural network is proposed for three operations(1.Patch extraction and representation; 2.Non-linear mapping; 3.Reconstruction.).

2.2. Stereo and multiview image SR using depth information

An approach to stereo image SR is to enhance the low-resolution image by analyzing the neighboring high-resolution views. Diogo et al. [12] found a way to do this by using the depth information to generate a virtual view from the full resolution viewpoint. Then, following a series of steps, the corresponding high-frequency details can be extracted and incorporated into the interpolated low-resolution image. Motivated by [12] and the state-of-art CNN approach [7], Yanchun et al. [8] decided to generate the full-resolution virtual view using DIBR from the viewpoint of the low-resolution image and then feed in both images to the 3-layer CNN. The virtual view is established through 3D-wraping:

$$X_{warp} = X_{origin} + b * \frac{f}{Z}. \quad (1)$$

where X_{warp} and X_{origin} are the horizontal coordinates of the reference point and the virtual viewpoint. b is the baseline distance, f is the focal length of the camera, and Z represents the pixel's depth information. The three layers' operations are similar with the SRCNN model [7], but the first layer also takes charge of fusing the two views together(full-resolution virtual view and low-resolution view). After trained on a small set of the image pairs, their model can provide promising results of asymmetric stereo image SR.

3. PROPOSED METHOD

Let us denote the low-resolution left view frame as $\mathbf{Y}_{LR}$ and the high-resolution right view frame as $\mathbf{Y}_{HR}$. The two frames $\mathbf{Y}_{LR}$ and $\mathbf{Y}_{HR}$ have different viewpoints. Our method here is aiming to recover $\mathbf{Y}_{LR}$'s full-resolution frame $\mathbf{X}$ with $F(\mathbf{Y}_{LR}, \mathbf{Y}_{HR})$. The mapping F consists of four steps:

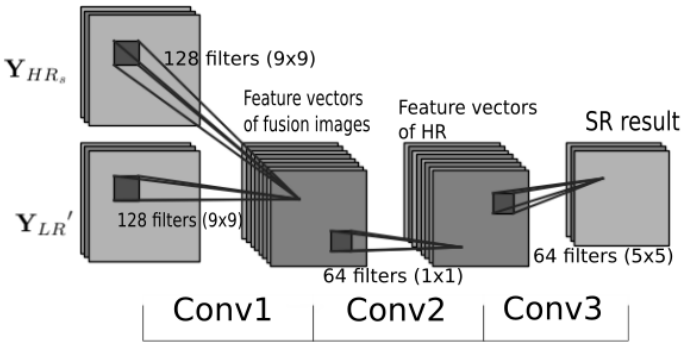

Fig. 1. The general architecture of our CNN. We put $\mathbf{Y}_{LR}{}'$ and $\mathbf{Y}_{HR_s}$ to our 3-layer convolutional nerual network. Layer 1: image fusion and feature extraction. Layer 2: non-linear mapping. Layer 3: high-resolution result reconstruction.

1. $view\ \ adjustment$. this operation adjusts the low-resolution left view image to the same size of the high-resolution right view. Let us denote the upscaled low-resolution image to be $\mathbf{Y}_{LR}{}'$. Then, this operation maps $\mathbf{Y}_{HR}$'s viewpoint to relatively close to the $\mathbf{Y}_{LR}{}'$'s. This shift can be achieved based on the disparity compensation. The result image from the shifting is denoted as $\mathbf{Y}_{HR_s}$.

2. $stereo\ \ view\ \ fusion$. this operation fuses both views $\mathbf{Y}_{LR}{}'$ and $\mathbf{Y}_{HR_s}$ and extracts features from each fusion using one single convolutional layer. The feature will be represented as a high-dimensional vector.

3. $non-linear\ \ mapping$. this operation builds a non-linear mapping between the fused mixed resolution high-dimensional vector and the other high-dimensional vector. The result vector is conceptually the representation of a patch of high-resolution pixels.

4. $result\ \ reconstruction$. this operation patch-wisely converts the high-dimensional vector from the previous step to the final high-resolution image.

The structures of the last three steps(CNN) is shown in Figure 1.

3.1. View adjustment

The initial upscaling of $\mathbf{Y}_{LR}$ is done by bicubic interpolation. We denote the interpolated image as $\mathbf{Y}_{LR}{}'$. As $\mathbf{Y}_{LR}{}'$ and $\mathbf{Y}_{HR}$ have the same dimension, we can get the disparity map $\mathbf{D}$ based on the H. Hirschmuller algorithm [13] from the two images.

By taking $\mathbf{Y}_{LR}{}'$ as the base image and $\mathbf{Y}_{HR}$ as the disparity image, the disparity function will do the dissimilation calculation pixel-by-pixel between the two images. We are

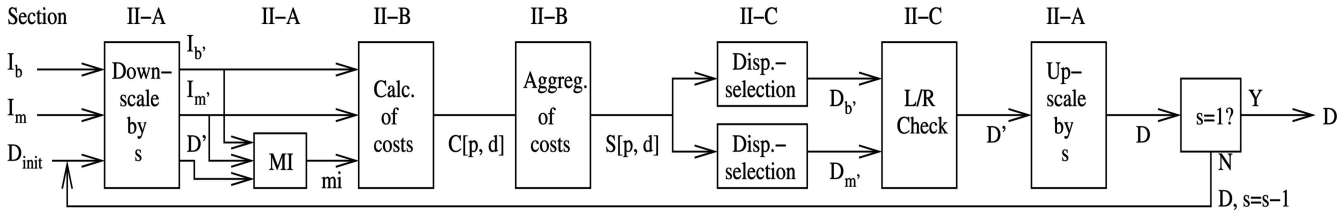

Fig. 2. The process flow of the H. Hirschmuller algorithm [13].

considering only 5 directions instead of 8 from $\mathbf{Y}_{LR}{}'$ to speed up the process. As shown in formula (2), the aggregated cost S($\mathbf{p}$, d) for a pixel $\mathbf{p}$ from the image and its disparity d is calculated by summing up the costs from each path that ends in block $\mathbf{p}$ at disparity d. The cost along one of the 5 paths $L_r(\mathbf{p}, d)$ can be calculated following the formula (3). The pixelwise matching cost C can be C_{BT} [14]. P_1 and P_2 are constant penalties for all pixels q in the neighborhood N_p of p as described in [13].

$$S(\mathbf{p}, d) = \sum_r L_r(\mathbf{p}, d). \tag{2}$$

$$\begin{aligned} L_r(\mathbf{p}, d) = C(\mathbf{p}, d) + min(&L_r(\mathbf{p} - \mathbf{r}, d), \\ &L_r(\mathbf{p} - \mathbf{r}, d - 1) + P_1, \\ &L_r(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \\ &\min_i L_r(\mathbf{p} - \mathbf{r}, i) + P_2) - \min_k L_r(\mathbf{p} - \mathbf{r}, k). \end{aligned} \tag{3}$$

The disparity map $\mathbf{D}$ based on $\mathbf{Y}_{LR}{}'$ and $\mathbf{Y}_{HR}$ is determined by selecting for each pixel $\mathbf{p}$, the disparity d that corresponds to the minimum cost, which is $min_d S(\mathbf{p}, d)$. The whole process of disparity map generation is described in Figure 2.

Based on the average value of the disparity map $\mathbf{D}$, we shift the $\mathbf{Y}_{HR}$ towards the $\mathbf{Y}_{LR}{}'$ to make the viewpoint of the two image as close as possible. We denote the shifted $\mathbf{Y}_{HR}$ as $\mathbf{Y}_{HR_s}$. The offset blank part resulting from the image shift is filled with the same area from $\mathbf{Y}_{LR}{}'$.

3.2. Stereo view fusion

After the view adjustment step, we divid the image pairs $\mathbf{Y}_{HR_s}$ and $\mathbf{Y}_{LR}{}'$ into small subimage pairs (i.e., 82-pixel x 82-pixel). Using sub-images here will speed up the training process for our network. Among all the pairs, we eliminate all the pairs that have low correlation. The correlation between two segments $\mathbf{I}_{LR}$ and $\mathbf{I}_{HR}$, is calculated by mean square difference $\mathbf{MSE}(\mathbf{I}_{LR}, \mathbf{I}_{HR})$ as shown as following:

$$\mathbf{MSE}(\mathbf{I}_{LR}, \mathbf{I}_{HR}) = \frac{1}{3HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{k=1}^{3} \left(\left| \mathbf{I}_{LRijk} - \mathbf{I}_{HRijk} \right| \right)^2, \tag{4}$$

where H and W are the height and width of the image segment. We drop all the segment pairs which have $\mathbf{MSE}$ lower than 2500.0.

After the pruning, we use a convolutional layer to fuse $\mathbf{I}_{LR}$ and $\mathbf{I}_{HR}$ and extract features. We use convolutional filters (9 x 9) to obtain the set of feature maps from the two input. We add a bias term to the result of the first layer. The activation function used for each output here is the ReLU function [15], $max(0, V_i)$, where V_i is the ith vector output from this convolutional layer.

Equation (5) shows the operation in this layer:

$$V_i = W_{1,i} \cdot \mathbf{I}_{LR} + W_{2,i} \cdot \mathbf{I}_{HR} + b_i, \tag{5}$$

where $W_{1,i}$ and $W_{2,i}$ are the ith filter, and b_i is the ith bias term that we add to the output.

3.3. Non-linear mapping and result construction

We have two convolutional layers to do the non-linear mapping and the result construction operation. The second convolutional layer, which is also the only hidden layer, will establish non-linear mapping of V_i. The final convolutional layer will reconstruct the high-resolution image of $\mathbf{Y}_{LR}{}'$.

3.4. Training procedure

The goal of the training procedure is to train the 3-layer convolutional nerual network to achieve the minimum error while estimating the high-resolution image of $\mathbf{I}_{LR}$, given training dataset $[\mathbf{I}_{LR,i}, \mathbf{I}_{HR,i}, \mathbf{L}_i]_{i=1}^N$. The input data $\mathbf{I}_{LR}$ and $\mathbf{I}_{HR}$ are patches chopped from the interpolated low-resolution image and the disparity compensated high-resolution neighboring view. $\mathbf{L}_i$ is the ground-truth(label). The loss function we use here is Mean Squared Error (MSE) of the estimated image, denoted as $\hat{L}$:

$$\mathbf{Loss}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left(||f(\mathbf{I}_{LR,i}, \mathbf{I}_{HR,i}, \theta) - \mathbf{L}_i|| \right)^2, \tag{6}$$

where θ represents all the filters' parameters and bias terms of our CNN.

In addition to the CNN, stochastic gradient descent (SGD) and Adam optimizer are used here. The weights of our network are initialized to be the random Gaussian distribution with zero mean. The learning rate for all the layer is 10^{-3}.

We found using small patches pairs from the original image to train is faster than feeding the whole image to the CNN. Also, chopping sub-images can give us more data to improve the accuracy of the CNN. Therefore, we decide to use the patch pairs from the full-size image as input to our CNN.

4. EXPERIMENTS

CNN usually requires a large amount of training dataset. For example, SRCNN[7] uses a large dataset from ImageNet. However, [8] gets a really good result by just using 17 image pairs. So, we decide to use 65 image pairs from Middlebury Stereo Dataset [16] for training. Each stereo pair of the training dataset has 2 neighboring views. After sub-image formating and pruning, we have 20194 pairs(82 x 82) for training.

For testing data, we use test image pairs "bicycle", "sticks", "storage", and "backpack" from Middlebury Stereo Datasets, "ballet scene" stereo frames from dataset created by [17], and "Akko & Kayo100" stereo frames from dataset [18].

Our CNN has 3 convolutional layers: 6 x 128 filters (9 x 9) as conv1, 128 x 64 fileters (1 x 1) as conv2, and 64 x 3 filters (5 x 5) in conv3, all with convolution stride of 1.

4.1. Performance

While testing, we compare our performance with SRCNN [7]. It can be found in Table 1 that our proposed method outperforms [7]. The numerical evaluation used here is Peak Signal-to-Noise Ratio (PSNR):

$$\mathbf{PSNR}(\hat{Y}, Y) = 20 * log_{10}(\frac{255}{\sqrt{\mathbf{MSE}(\hat{Y}, Y)}}), \quad (7)$$

where Y is the ground truth image and $\hat{Y}$ is the estimation image.

To compare, we trained the 9-1-5 model of SRCNN on over 40000 patches generated from the 91 images they provided and get a stable error loss. Then, we feed the low-resolution image dataset to the SRCNN and the mixed-resolution image pair dataset to our model. The result can be found in table 1. All the test is running after the ground truth image is downsampled by 4.

Our method outperforms the 9-1-5 SRCNN model by 0.487dB on average. In "bicycle", our proposed method performed poorly and we conclude that this is due to the amount of detail there is large and the disparity value for this image pair is larger than other test sequences.

Table 1. Comparison with 9-1-5 SRCNN [7], with downsampling scale 4

Dataset	Scale	SRCNN [7](dB)	Proposed(dB)
backpack	4	32.3378	32.7189
bicycle	4	31.6809	30.9519
sticks	4	33.7455	34.3726
storage	4	39.5527	41.4968
ballet scene	4	31.8805	32.3132
Akko & Kayo100 frame 1	4	30.7760	31.0459
Average	4	33.3289	33.81655

Table 2. Comparison with 9-5-5 SRCNN [7], with downsampling scale 4

Dataset	Scale	SRCNN [7](dB)	Proposed(dB)
backpack	4	33.2802	33.2179
bicycle	4	32.3124	31.8253
sticks	4	34.6773	34.7498
storage	4	42.1589	42.2955
ballet scene	4	32.1652	32.2018
Akko & Kayo100 frame 1	4	31.5810	31.4178
Average	4	34.4625	34.2847

5. CONCLUSION

We have presented a CNN approach to asymmetric stereo video super-resolution. Our proposed approach outperforms the existing SISR methods on our testing image pairs and stereo video frame sequences. However, because of not using depth information, the outperforming margin is not stable and can vary on the correlation between the two views. Also, a larger training dataset could help our model to achieve a better result.

6. REFERENCES

[1] Michal Joachimiak, Payman Aflaki, Miska M. Hannuksela, and Moncef Gabbouj, *Evaluation of Depth-Based Super Resolution on Compressed Mixed Resolution 3D Video*, pp. 227–237, Springer International Publishing, Cham, 2015.

[2] P. Aflaki, M. M. Hannuksela, J. Hkkinen, P. Lindroos, and M. Gabbouj, "Impact of downsampling ratio in mixed-resolution stereoscopic video," in *2010 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video*, June 2010, pp. 1–4.

[3] L. Stelmach, Wa James Tam, D. Meegan, and A. Vincent, "Stereo image quality: effects of mixed spatio-

temporal resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 2, pp. 188–193, Mar 2000.

[4] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, Nov 2010.

[5] G. Sun and C. Qin, "Single image super-resolution via sparse representation in gradient domain," in *2011 Third International Conference on Multimedia Information Networking and Security*, Nov 2011, pp. 24–28.

[6] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, June 2010.

[7] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb 2016.

[8] Y. Xie, J. Xiao, T. Tillo, Y. Wei, and Y. Zhao, "3d video super-resolution using fully convolutional neural networks," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, July 2016, pp. 1–6.

[9] Z. Jin, T. Tillo, C. Yao, J. Xiao, and Y. Zhao, "Virtual-view-assisted video super-resolution and enhancement," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 467–478, March 2016.

[10] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 349–356.

[11] R. Timofte, V. De, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 1920–1927.

[12] E. M. Hung, C. Dorea, D. C. Garcia, and R. L. de Queiroz, "Transform-domain super-resolution for multiview images using depth information," in *2011 19th European Signal Processing Conference*, Aug 2011, pp. 398–401.

[13] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb 2008.

[14] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Jan 1998, pp. 1073–1080.

[15] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, USA, 2010, ICML'10, pp. 807–814, Omnipress.

[16] Daniel Scharstein, Heiko Hirschmller, York Kitajima, Greg Krathwohl, Nera Nei, Xi Wang, and Porter Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," 09 2014.

[17] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, Aug. 2004.

[18] Nagoya University, "Nagoya university sequences," 2008, http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/.[Online; accessed 1-November-2017].