Multi-Exposure Image Fusion with Denoising

Xinyu Guo (20847775)

April 2020

1 Introduction

The range of luminance levels in the natural scenes are usually high dynamic, the irradiance across a scene varies a lot, when taking an image of a scene; no matter what exposure time is used, there will usually exists some area of the image to be over- or under-exposed. On the other hand, there is a big gap between the luminance range of low dynamic standard display devices. The Multi-Exposure Image Fusion (MEF) algorithm try to solve these problems, by fusing a sequence of differently exposed images to one single image.

Many algorithms are proposed to solve this problem. Most of them are pixel or patch weighting based method, in these methods, the weight for each pixel or each patch is calculated, then the fused image is generated by combining these weighted pixels or patches. The weights are computed based on different hypothesis which result in different MEF algorithms. For example, Laplacian pyramid is used to calculate pixel weights[1], in this method, Laplacian pyramid is used to decompose an image into different component. The one with fine details can be served as weighting map. Also, entropy based patch selection method is adopted to fuse input sequence[2]. One of the state of the art MEF algorithms proposed recently is MEF-SSIM $_c$ [3], which is a numerical optimization based method. Taking the a sequence of color images as input, MEF-SSIM $_c$ aim to produced a fused image by comparing the similarity between the input and the fused image. The fused image is generated by an optimization process.

However, the MEF algorithms usually applied on clean input images, which is diffenerent from the practical situations. In this work, the aim is to make innovation based on the above MEF-SSIM $_c$ algorithm and the innovation is made by generalization. In the real environment, a image will go through several process before reaching the consumer. Such as image acquisition and transmission. Noise can be added to image during such processes. For example, the sensor in the camera may introduce noise to image data. Thus, this work aim to generalize the original solution to a more complex setting. A CNN-based denoising module is added to the MEF algorithm,this denoising alogorithm is applied to input noisy sequence before they enter the MEF algorithm. The denoising CNN [4] is trained on different noise levels, and the fused image is generated using the numerical optimization method operating on the joint denoising and similarity comparison pipeline.

2 Literature Review

Over the past decades, lots of MEF algorithms are proposed, ranging from simple weighted summation methods to complex statistical methods. A simple algorithm to determine weights are local and global energy weighting approaches. In which, local energy and global energy are used for computing the weights. Some pixel based weighting methods are proposed, Laplacian pyramid is used by Burt to completing binocular image fusion [1], in which, pyramid parameters are used to compute the weight used for fusion. A multi-explosure fusion algorithm is later developed by Mertens [5]. In this method, contrast, saturation and well-exposedness are quantified at pixel level. Then these values are multiplied together which yield weights for pixels. Bilateral filter is adopted by Raman to create weighting map[6], in this method, utilizing the edge preserving and smoothing feature of bilateral filter, the weighting map contains the texture and weak edge of original image, which is served to detect over- or under-exposure. A improvement method named boosting Laplacian pyramid is proposed later, in which a hybrid exposure quality weight is used to fused a sequence of input to one output[7]. The hybrid exposure quality weight is calculated by considering the global exposure weight, local exposure weight and saliency weight. Also, there are some patch based weighting methods. A entropy based patch selection method is proposed, the patch with the higheset entropy is consider to construct the fused image[2], however, because the patch position is determined in a non-overlapping fashion, blocking artifacts are appeared in this method. Later, a structural comparison method for patch decomposition is proposed, the patch is decomposed to signal strength, signal structure and mean, and the position is determined using a sliding window approach, then a new patch is reconstruction for image fusion [8]. Because patches are chosen by applying a sliding window, the blocking artifact can be reduced. In [9], contrast and gradient are first calculated, then the fused image is synthesised by suppressing reversals in image gradient.

A important problem in this domain is the misalignment problem. When the object in the scence is moving or the camera is moving, the sequence of the input images can be misaligned. A gradient guided fusion method is proposed, the gradient is computed and used to distinguish the moving object from the background [10]. Later on, the dynamic scene problem is solved by using a ghost removal pipeline. Firstly dectcting the non-consistency pixels, then the detected pixels are corrected [11]. Another method to tackle this problem is proposed by Qin et al. [12], in their work, a matching algorithm which operate at image patch level is developed to to keep track of the motion in the input sequence.

A numerical optimization based method for multi-exposure image fusion is proposed recently [3]. In this method, patches with highest contrast is consider to be the well-exposed, and a optimal exposure image is constructed by combining all selected patches at different position together. This optimal image and a initial image is measured using color image structure simiilarity index. The aim is to maximize the output structure similarity index by keep updating the given initial image.

3 Proposed Solution

As mentioned above, a denoising-based MEF method is proposed to generalize the original solution to a more general setting. For denoising purpose, A CNN-based denoising module is trained and added to the MEF-SSIM $_c$. The overall algorithm flow can be seen in Figure 1. Firstly, the input noisy sequence is put in the denoising CNN, the denoising CNN will reduce the noise and give the noise-reduce sequence to MEF-SSIM $_c$ algorithm. The patch selection is the first step of MEF-SSIM $_c$ algorithm, at each location, a patch with highest contract is selected and go through the structure similarity comparison with the corresponding patch in random initial image. This process gives quality score for each patch. Then we use gradient ascent algorithm to increase the quality score by adjusting each patch in random initial image. Finally, the fused image is created.

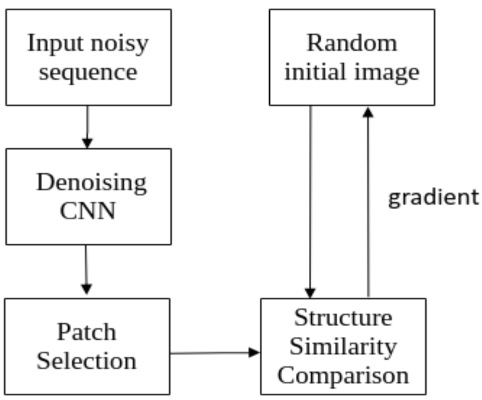

Figure 1: Block diagram of proposed solution

In this section, firstly, the denoising based MEF algorithm is introduced. Then, some of the results are presented. Finally, methods used in comparative study are briefly introduced.

In the proposed MEF algorithm, a CNN-based denoising module is firstly applied to the input sequence. For the CNN, We adopted the network structure design in [4].

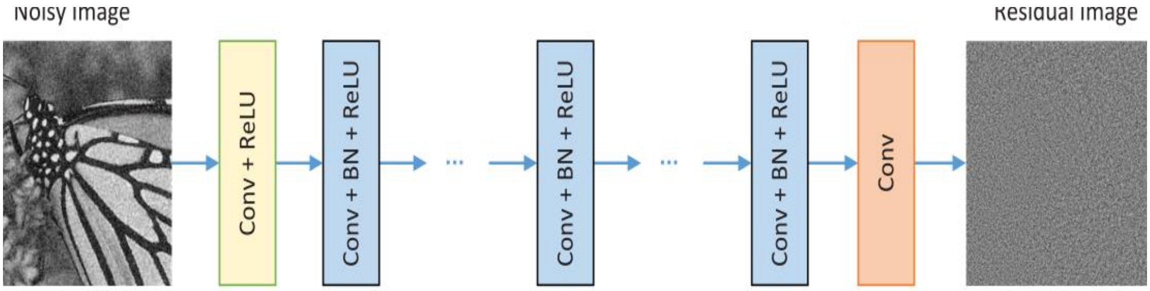

Figure 2: Network structure of the denoising CNN [4]

As shown in Figure 2. The denoising CNN consist of 17 layers. For the first layer, 64 convolution kernels with size 3×3 are applied to the input images, followed by a ReLU function. The middle 15 layers follow a same structure. These layers are composed of a convolutional layer with 64 3×3 kernels, a batch nomalization layer and a ReLU layer. The last layer of the network is a convolution later with one 3×3 kernel which output a residual image. The input of the network is a grayscale image. The network try to predict the residual which is the difference between the noisy image and the reference image.

The CNN is applied to the input noisy images. Since the input noisy images are color images, we firstly split the color images by R, G, B three channels, and our model is applied to each channel respectively. The output image is obtained by concatenating three channels together. Let X_k represents the k^{th} noisy image in the input noisy sequence, then the denoising process can be express by $\bar{X}_k = f(X_k)$. The function of denoising CNN is represented by f

The following MEF-SSIM$_c$ algorithm takes a sequence of noise-reduced color images and a initial images created at a random starting point as inputs, this algorithm create the final fused image by a comparison and optimization process. The comparison is done at patch level. Firstly, each image in the image sequence is divided into patches using a sliding window. For each patch, the patch strength is computed by calculating the standard deviation of the patch, and the patch with the highest patch strength is selected. Then, the patch intensity is obtained by following equations:

$$\hat{l} = \frac{\sum_{k=1}^{K} u(\mu_k, l_k)\, l_k}{\sum_{k=1}^{K} u(\mu_k, l_k)}$$

where μ_k is the mean intensity of whole color image $\bar{\mathbf{X}}_k$, l_k is the mean intensity the patch. The weighting function is a 2 dimensional gaussian function:

$$u(\mu_k, l_k) = \exp\left(-\frac{(\mu_k - \mu_c)^2}{2\sigma_g^2} - \frac{(l_k - l_c)^2}{2\sigma_l^2}\right)$$

where μ_c and l_c are constants representing the middle intensity. For normalized images, the values are 0.5. σ_g and σ_l control the shape of the function along two dimensions. Finally, the patch-wise comparison is done by computing the structural similarity index, and the structural similarity for the whole between the whole sequence and the initial image is done by averaging the patch level comparison results:

$$S(\{\bar{\mathbf{x}}_k\}, \mathbf{y}) = \frac{(2\mu_{\hat{\mathbf{x}}}\mu_{\mathbf{y}} + C_1)(2\sigma_{\hat{\mathbf{x}}\mathbf{y}} + C_2)}{(\mu_{\hat{\mathbf{x}}}^2 + \mu_{\mathbf{y}}^2 + C_1)(\sigma_{\hat{\mathbf{x}}}^2 + \sigma_{\mathbf{y}}^2 + C_2)}$$

$$Q(\{\bar{\mathbf{X}}_k\}, \mathbf{Y}) = \frac{1}{M}\sum_{j=1}^{M} S(\{\bar{\mathbf{x}}_k(j)\}, \mathbf{y}(j))$$

The aim of the proposed solution is to get the optimal fused images when the inputs is a noisy sequence, We can obtain the optimal solution by keep adjusting Y to let cost function Q reach its maximum value:

$$Y_{opt} = \arg\max_{Y} Q\left(\{f(X_k)\}, Y\right)$$

The cost function here is different from the original one since the denoising CNN is combined. We solve it using the gradient ascent-based method adopted in the original MEF-SSIM $_c$ paper:

$$Y_{i+1} = Y_i + \lambda \nabla_Y Q\left(\{f(X_k)\}, Y\right)$$

We implement out proposed solution using PyTorch. For the denoising part, The training data, parameters and hyper-parameters are same with the ones used in [4]. This including the learning rate $lr = 1e-3$, batchsize=128. However, different epochs are used for training on noisy images with different noise levels. Here, we use 2 noise levels: $\sigma = 5, 15$, and epochs are set to 15, 50 respectively. This is because for smaller noise level, it is easier to train the neural network. Thus, we adopt fewer epochs which can save some time. For creating the fused image, same here, most of the parameters are same with previous publication[3]. These parameters include two parameters for controlling shape of gaussian function $\sigma_g = 0.2, \sigma_l = 0.2$, the parameters for structure similarity comparison $C1 = (K_1L)^2, C2 = (K_2L)^2$. $L = 255$ for 8 bits sequence, $K1 = 0.01, K2 = 0.03$, step size $\lambda = 150$ for gradient ascent method. The iteration is set to 1500 for a easier implementation. After the optimizing, some example results can be seen in Figure 3. The proposed solution can preserve the vivid color in the fused image produced by MEF-SSIM$_c$ algorithm, while the noise in the fused image is reduced.

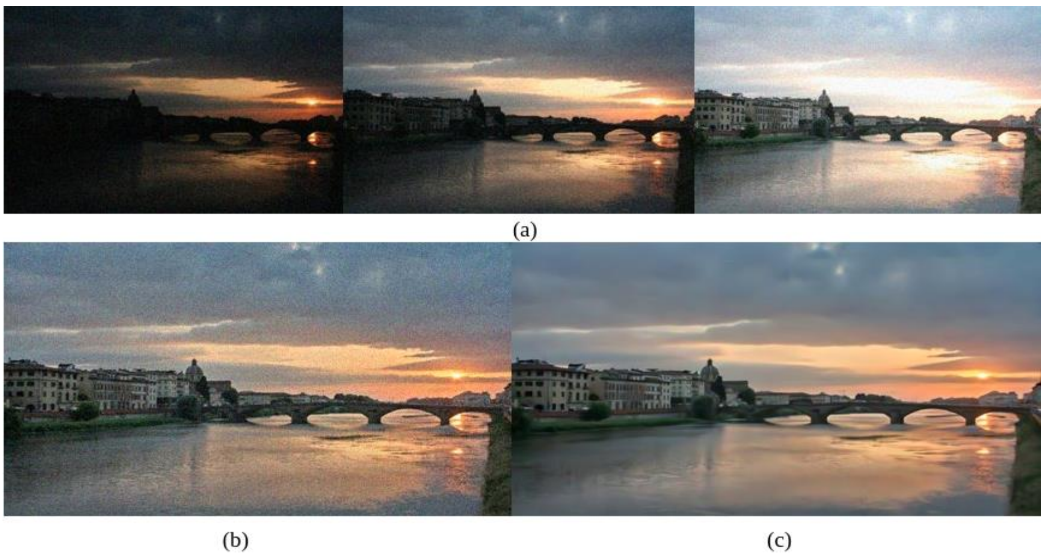

Figure 3: Visual comparison of MEF algorithms. (a) Source noisy sequence "Arno" ($\sigma = 15$). (b) Result of MEF-SSIM$_c$. (c) Result of proposed solution

The MEF algorithm developed in [6] is adopted in comparative study. In this algorithm, the luminance of each pixel is computed using the RGB values, then a bilateral filter is applied on the luminance image:

$$f^{BF}(x,y) = \frac{\sum_{y'} \sum_{x'} f(x',y')\, G_{\sigma_s}(x-x', y-y')\, G_{\sigma_r}(f(x,y) - f(x',y'))}{\sum_{y'} \sum_{x'} G_{\sigma_s}(x-x', y-y')\, G_{\sigma_r}(f(x,y) - f(x',y'))}$$

where $G_{\sigma_s}(x,y) = \exp\left(-\frac{(x^2+y^2)}{2\sigma_s^2}\right)$ is a spatial domain gaussian filter, $G_{\sigma_r}(a) = \exp\left(-\frac{a^2}{2\sigma_r^2}\right)$ is an intensity domain gaussian filter, (x',y') is the neighbour of (x,y).

A entropy based MEF algorithm is also adopted in the comparative study [13]. Firstly, the image sequence is transformed to log domain, in circular region defined by parameter R, the cross exposure normalized log-scale local entropy is computed using:

$$H_k^{\text{norm}}(i,j) = \frac{H_k^{\log}(i,j)}{\sum_{k=1}^{N} H_k^{\log}(i,j)}$$

Based on the local entropy, the weight of each pixel in the exposure sequence is computed, the weights are further adjusted using the exponential function. Finally, weights are normalized and applied to multi-exposure input sequence.

4 Experimental Results

Twenty four source image sequences used in [3] are adopted here. The visual demonstration in Figure 4 shows some of the source sequence and noisy source sequences.

Figure 4: Visual comparison source sequence and noisy source sequence ($\sigma = 15$)

The proposed solution is compared with MEF-SSIM$_c$[3], a entropy based MEF algorithm [13], a bilateral filter based MEF algorithm [6]. These algorithms are compared using the same input noisy source sequences. For the MEF-SSIM$_c$ algorithm and the proposed solution, the initial images are both random gaussian noise with $\mu = 0, \sigma = 1$.

Firstly, the visual comparison of fused images created by different MEF algorithms are conducted. The inputs is the "Farmhouse" image sequence, the demonstration is provide in Figure 5.

From Figure 5 we can see that the fused images produced by Raman09 and Bruce14 are darker and there are noise inside the fused images. These two weakness points reduce the visual quality of the fused image created by them. The color produced by Raman09 and Bruce14 is not as good as that produced by MEF-SSIM$_c$ algorithm. The proposed method can produce the color as vivid as MEF-SSIM$_c$ algorithm and reduce the noise using the CNN component in the algorithm.

Next, the algorithms are compared by quantized values. The color version structural similarity index betweeen the fused images created by four different algorithms and the source sequence is calculated. The reference image is create from the source sequence using the patch selection method. The patches with highest contrast at each spatial location is selected to be the representative at that location. The select patch is compared with patch extracted from the fused images created by each algorithm. Then for each algorithm, we can get a SSIM$_c$ score for each image content. The algorithms are compared at two different noise levels using 24 difference image contents. The results are presented in the table.

From Table 1 we can the proposed solution gives better results compared with other algorithms on noisy input sequences. In the "Average" row in the table, the SSIM$_c$ score given by proposed solution at two noise levels outperform the scores given by other algorithms. When the noise level raise from 5 to 15, all other algorithms performance drops significantly, however, with the denoising module the proposed solution just drops a little. Also, the influence of noise to different algorithm can be different.

Image	Noise	Raman09	Bruce14	MEF-SSIM$_c$	Denoising MEF-SSIM$_c$
	$\sigma = 0$	0.8156	0.8633	0.9583	-
Room	$\sigma = 5$	0.7971	0.8315	0.9340	0.9516
	$\sigma = 15$	0.6768	0.6995	0.8143	0.9349
	$\sigma = 0$	0.7248	0.8240	0.9915	-
Church	$\sigma = 5$	0.7167	0.7945	0.9807	0.9868
	$\sigma = 15$	0.6585	0.7095	0.9062	0.9697
	$\sigma = 0$	0.5008	0.5704	0.9902	-
Balloons	$\sigma = 5$	0.4781	0.4913	0.9526	0.9836
	$\sigma = 15$	0.3818	0.3717	0.7867	0.9673
	$\sigma = 0$	0.8579	0.9196	0.9845	-
Kluki	$\sigma = 5$	0.8196	0.8676	0.9441	0.9727
	$\sigma = 15$	0.6651	0.6860	0.7691	0.9404
	$\sigma = 0$	0.9557	0.9617	0.9980	-
Yellow	$\sigma = 5$	0.9494	0.9492	0.9929	0.9971
	$\sigma = 15$	0.9070	0.9021	0.9553	0.9947
	$\sigma = 0$	0.6771	0.8232	0.9687	-
House	$\sigma = 5$	0.6558	0.7634	0.9444	0.9629
	$\sigma = 15$	0.5683	0.6176	0.8223	0.9461
	$\sigma = 0$	0.5178	0.5331	0.9800	-
Lamp	$\sigma = 5$	0.4984	0.4626	0.9472	0.9731
	$\sigma = 15$	0.4177	0.3808	0.7975	0.9596
	$\sigma = 0$	0.5384	0.5692	0.9813	-
Belgium	$\sigma = 5$	0.5189	0.5268	0.9526	0.9740
	$\sigma = 15$	0.4338	0.4269	0.7963	0.9569
	$\sigma = 0$	0.8681	0.9517	0.9946	-
Lighthouse	$\sigma = 5$	0.8454	0.9175	0.9756	0.9909
	$\sigma = 15$	0.7337	0.7876	0.8625	0.9804
	$\sigma = 0$	0.8700	0.9131	0.9833	-
Ostrow	$\sigma = 5$	0.7928	0.8088	0.8945	0.9647
	$\sigma = 15$	0.5229	0.5156	0.5984	0.9311
	$\sigma = 0$	0.7741	0.8659	0.9919	-
Mask	$\sigma = 5$	0.7355	0.7962	0.9503	0.9767
	$\sigma = 15$	0.5775	0.6071	0.7501	0.9420
	$\sigma = 0$	0.4244	0.8689	0.9871	-
Cave	$\sigma = 5$	0.4261	0.7231	0.9471	0.9671
	$\sigma = 15$	0.3731	0.4836	0.7746	0.9026
	$\sigma = 0$	0.9223	0.9610	0.9948	-
Set	$\sigma = 5$	0.8825	0.9158	0.9595	0.9884
	$\sigma = 15$	0.7248	0.7512	0.8064	0.9799
	$\sigma = 0$	0.8672	0.9029	0.9928	-
Arno	$\sigma = 5$	0.7970	0.8138	0.9222	0.9765
	$\sigma = 15$	0.5534	0.5523	0.6532	0.9464
	$\sigma = 0$	0.8171	0.9121	0.9867	-
Lauren	$\sigma = 5$	0.7894	0.8629	0.9584	0.9783
	$\sigma = 15$	0.6655	0.7056	0.8104	0.9520
	$\sigma = 0$	0.8374	0.9033	0.9889	-
Office	$\sigma = 5$	0.7944	0.8354	0.9449	0.9785
	$\sigma = 15$	0.6241	0.6282	0.7566	0.9546
	$\sigma = 0$	0.6800	0.8437	0.9893	-
Farmhouse	$\sigma = 5$	0.6617	0.7661	0.9576	0.9835
	$\sigma = 15$	0.5742	0.6354	0.8192	0.9705
	$\sigma = 0$	0.7865	0.9102	0.9911	-
Tower	$\sigma = 5$	0.7349	0.8179	0.9375	0.9722
	$\sigma = 15$	0.5471	0.5896	0.7059	0.9190
	$\sigma = 0$	0.5487	0.5767	0.9796	-
Madison	$\sigma = 5$	0.5375	0.5582	0.9651	0.9727
	$\sigma = 15$	0.4873	0.4921	0.8822	0.9482
	$\sigma = 0$	0.7624	0.8647	0.9876	-
Window	$\sigma = 5$	0.7195	0.7698	0.9362	0.9725
	$\sigma = 15$	0.5548	0.5758	0.7289	0.9501
	$\sigma = 0$	0.8784	0.9584	0.9954	-
Landscape	$\sigma = 5$	0.8126	0.8624	0.9319	0.9744
	$\sigma = 15$	0.5540	0.5611	0.6542	0.9231
	$\sigma = 0$	0.5844	0.7295	0.9791	-
Studio	$\sigma = 5$	0.5697	0.6396	0.9554	0.9738
	$\sigma = 15$	0.4950	0.5311	0.8235	0.9608
	$\sigma = 0$	0.7634	0.8884	0.9743	-
Venice	$\sigma = 5$	0.7316	0.8305	0.9395	0.9598
	$\sigma = 15$	0.5842	0.6272	0.7592	0.9163
	$\sigma = 0$	0.7694	0.8772	0.9916	-
Garden	$\sigma = 5$	0.7422	0.8233	0.9607	0.9799
	$\sigma = 15$	0.6296	0.6698	0.8129	0.9426
	$\sigma = 0$	**0.7393**	**0.8331**	**0.9859**	-
Average	$\sigma = 5$	**0.7086**	**0.7678**	**0.9494**	**0.9755**
	$\sigma = 15$	**0.5797**	**0.6045**	**0.7853**	**0.9496**

Table 1: Comparison between MEF algorithms

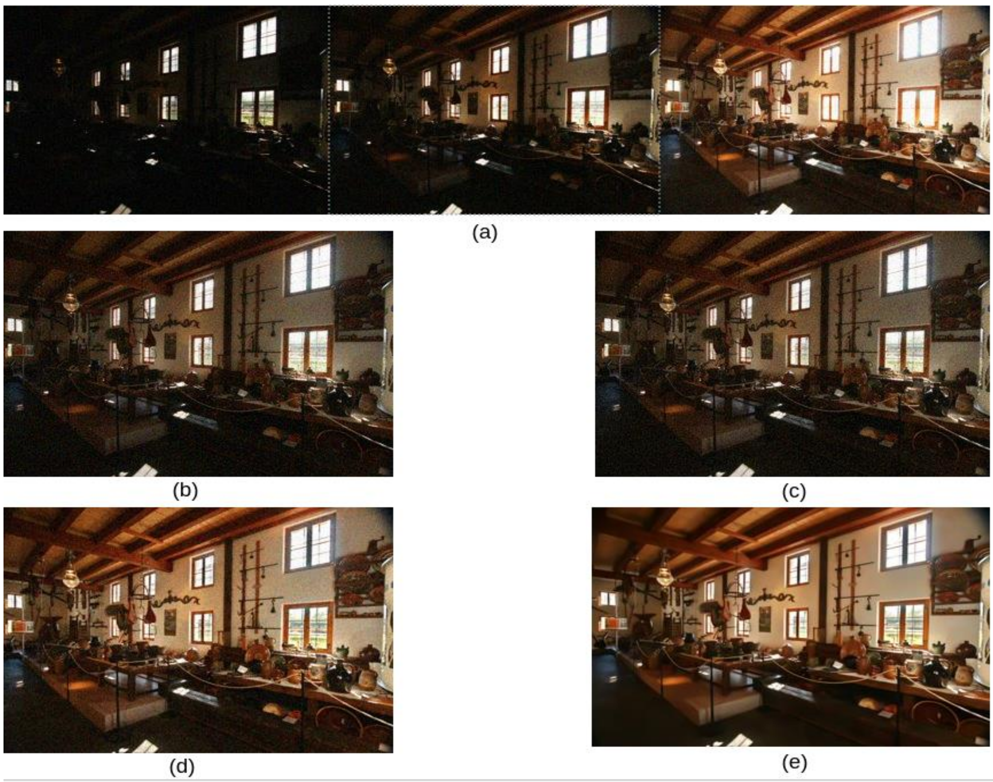

Figure 5: Visual comparison of different MEF algorithm (a) Input noisy source sequence ($\sigma = 15$) (b) Fused image produced by Raman09[6] (c) Fused image produced by Bruce14[13] (d) Fused image produced by MEF-SSIM$_c$ (e) Fused image produced by proposed solution

As noise level raise from 5 to 15, score for Bruce14 and MEF-SSIM$_c$ drop around 0.16, but Raman09 drops about 0.13. Another point to mention here is that the performance of MEF algorithm can vary from sequence to sequence. For example, Raman09 and Bruce14 perform really poor on input sequence with lots of exposure images, such as "Balloons", "Lamp", "Belgium", "Madison" which have over 9 images in input sequence. However, the other two algorithms have better performance.

5 Discussion and Conclusions

As mentioned above, the denoising feature of MEF algorithm is quite important since in the real environment, a image will go through several process before reaching the consumer. Image acquisition and transmission are some examples. As a result, noise can be added to image during such processes. Noise can significantly drop the performance of MEF algorithm by degrading the fused image quality as shown in Figure 3. In Figure 3, it is obvious that the scene in (c) is more visual appealing to people than (b). This can be also verified by the figures in Table 1, denoising based MEF-SSIM$_c$ algorithm has much better performance that the algorithms without denoising module. The combination of denoising module and MEF algorithm is based on the real situation and can help to reduce the influence of noise to some extent.By combining denoising module into a MEF algorithm, there are some improvements. On the one hand, the denoising algorithm successfully reduce the noise in the input sequence, as a result, the MEF algorithm can take the noise-reduced sequence as its input and create more visual appealing fused image. On the other hand, this combination let the whole algorithm take advantage of the state of the art MEF-SSIM$_c$ algorithm. As the example listed above, for the input sequence with lots of images, the MEF-SSIM$_c$ has a property to create vivid color in the fused images which is not achieved by other compared MEF algorithms.

However, there are also some drawbacks in the proposed solution for future improvements. First of all, the nosie module is built upon the MEF-SSIM$_c$ algorithm, which is one of the state of the art algorithms in the field. If the basic MEF algorithm can be improved, the performance of the combined algorithm can have a better result as well. On the other hand, from Table 1, we can notice that, when noice level improves, the score drops, this means the denoising algorithm used in the solution have some space for improvement, the can be either using a direct color image denoising algorithm which fully explore the correlation between color channels, or using a better grayscale image denoising algorithm.

References

[1] Peter J Burt. The pyramid as a structure for efficient computation. In *Multiresolution image processing and analysis*, pages 6–35. Springer, 1984.

[2] A Ardeshir Goshtasby. Fusion of multi-exposure images. *Image and Vision Computing*, 23(6):611–618, 2005.

[3] Kede Ma, Zhengfang Duanmu, Hojatollah Yeganeh, and Zhou Wang. Multi-exposure image fusion by optimizing a structural similarity index. *IEEE Transactions on Computational Imaging*, 4(1):60–72, 2017.

[4] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[5] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. In *Computer graphics forum*, volume 28, pages 161–171. Wiley Online Library, 2009.

[6] Shanmuganathan Raman and Subhasis Chaudhuri. Bilateral filter based compositing for variable exposure photography. In *Eurographics (short papers)*, pages 1–4, 2009.

[7] Jianbing Shen, Ying Zhao, Shuicheng Yan, Xuelong Li, et al. Exposure fusion using boosting laplacian pyramid. *IEEE Trans. Cybernetics*, 44(9):1579–1590, 2014.

[8] Kede Ma and Zhou Wang. Multi-exposure image fusion: A patch-wise approach. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1717–1721. IEEE, 2015.

[9] Mingli Song, Dacheng Tao, Chun Chen, Jiajun Bu, Jiebo Luo, and Chengqi Zhang. Probabilistic exposure fusion. *IEEE Transactions on Image Processing*, 21(1):341–357, 2011.

[10] Wei Zhang and Wai-Kuen Cham. Gradient-directed multiexposure composition. *IEEE Transactions on Image Processing*, 21(4):2318–2323, 2011.

[11] Zhengguo Li, Jinghong Zheng, Zijian Zhu, and Shiqian Wu. Selectively detail-enhanced fusion of differently exposed images with moving objects. *IEEE Transactions on Image Processing*, 23(10):4372–4382, 2014.

[12] Xiameng Qin, Jianbing Shen, Xiaoyang Mao, Xuelong Li, and Yunde Jia. Robust match fusion using optimization. *IEEE transactions on cybernetics*, 45(8):1549–1560, 2014.

[13] Neil DB Bruce. Expoblend: Information preserving exposure blending based on normalized log-domain entropy. *Computers & Graphics*, 39:12–23, 2014.