Single Image Dehazing Using Deep Convolution Neural Networks

Shengdong Zhang[1], Fazhi He[1](✉), and Jian Yao[2]

[1] Wuhan University State Key Lab of Software Engineering, School of computer science

[2] School of Remote Sensing and Information Engineering, Wuhan University

Abstract. Haze removal is urgently desired in multi-media system. In this paper, a deep learning-based method, called dehazingCNN, is proposed to estimate an approximate clear image for a hazy image. Our method attempts to recover a clear image by a learning model, which is different from traditional learning based method. Our method also adopts Deep Convolution Neural Networks (CNN) which takes a hazy image as the input and outputs the corresponding clear image. As shown in experiment results section, the output of the network is high quality except some block artifacts and color distortions. We can remove the color distortion in the approximate clear image via atmospheric scattering model and guided filter effectively. Experimental results on different type of images, such as synthetic and benchmark of hazy images, demonstrate that the proposed method is comparative to and even better than the more complex state-of-the-art methods in term of the dehazing effect.

Keywords: Haze removal, image restoration, Deep Convolution Neural Networks

1 Introduction

Natural images captured in outdoor are often degraded because of the bad weather conditions, such as air particles, fog, haze, smoke, rain or snow, which greatly reduce the visibility and quality of the images. In general, the degraded image presents a low contrast and quality.

Haze removing or defogging is urgently needed in computer vision applications and commercial/computational photography. However defogging or haze removal is a challenging problem due to the fog formulation depends on the unknown depth information of the scene. Obviously, the haze removal problem is ill-posed because only three equations for the pixels in a single input image can be provided but there exists four unknown quantities to be recovered. Therefore, in the past decades, a lot of methods have been proposed by using additional data or multiple images [1,2].

Furthermore, removing haze from single input image is a more difficult case but recently a significant progress has been made for single image dehazing due to its wide applications. Traditional methods [3,4,5,6,7,8,9,10,11] have been applied

to single image dehazing successfully. Because our method is a learning-based method, we focus on comparing with learning-based methods [10,12,13,14].

In [12], Tang et al. investigate the different haze-relevant feature of hazy image, and use best suitable feature combination to estimate the corresponding transmission map for a hazy image. In [10], Zhu et al. propose a learning-based method which considers the transmission as a linear combination of the saturation and brightness of pixels in a hazy image. They use a learning strategy to get the parameters of the model. The most relative works to ours are Cai et al.[13] and [14], which are also a deep learning-based method for estimating of transmission map. In [14], Ren et al. propose a multi-scale deep convolutional neural networks to dehazing. In contrast, our method estimates a haze-free image from hazy image directly. Compared with other learning-based methods, our network is much simpler and generates high quality results. Our main contributions are threefold: First, to the best of our knowledged, our work is the first to remove haze from single image without estimating transmission map. Second, we convert the problem from color domain to detail domain, which makes directly removing haze without estimating transmission map possible. Third, we show our method can achieve the state-of-the-art performance by a lot of experiments.

2 Haze Removal

In this section, we describe our method for removing haze from single image. Our method consists of three essential steps: a preprocessing to get $\mathbf{I}_A(x)$, estimating clear image using Deep Convolution Neural Networks, removing color distortion and block artifacts (see Alg. 1).

Modeling of Haze Images: Haze model [15] is widely used in computer vision, which also explains the formulation of the haze :

$$\mathbf{I}(i,j) = \mathbf{J}(i,j) * t(i,j) + (1 - t(i,j)) * \mathbf{A}, \tag{1}$$

where $\mathbf{I}$ represents the haze image, $\mathbf{J}$ represents the haze free image, $\mathbf{A}$ represents the global atmospheric light, t represents the transmission describing the probability of the light that is not scattered and absorption by air particle or mist and arrives at the camera. To remove haze from the hazy image is equal to solve the $\mathbf{A}$, t from $\mathbf{I}$.

Notation: We estimate A using one of the previous methods [7,16]. We define I_A,$r(x)$ and J_A as follows:

$$\mathbf{I}_A(x) = \mathbf{I}(x) - \mathbf{A} \tag{2}$$

$$r(x) = ||\mathbf{I}_A(x)|| \tag{3}$$

$$\mathbf{J}_A(x) = \mathbf{J}(x) - \mathbf{A} \tag{4}$$

$$\tilde{r}(x) = ||\mathbf{J}_A(x)|| \tag{5}$$

Where $\mathbf{I}_A$ translates the 3D RGB coordinate of hazy image to sphere coordinate, in which the airlight is at the origin, $r(x)$ represents the distance in RGB space

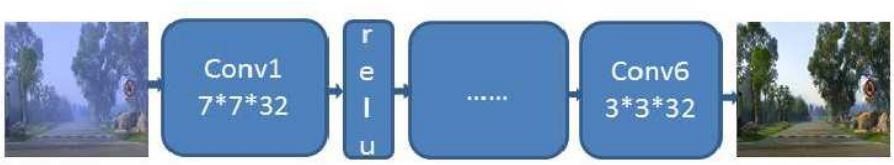

Fig. 1: The CNN architecture used for haze removal experiments.

of each pixel in hazy image to the airlight, $\mathbf{J}_A$ translate a clear image $\mathbf{J}$ to sphere coordinate, $\tilde{r}(x)$ represents the distance in RGB space of each pixel in haze-free image to the airlight.

Formulation: Consider a single hazy image, we first get $\mathbf{I}_A(x)$ by using equation 2, which is the only preprocessing we perform. Our goal is to recover from $\mathbf{I}_A(x)$ the init clear image patch $F(\mathbf{I}_A)$, which is as similar as possible to the truth clear image patch $\mathbf{J}_A$. For the ease of presentation, we still call $\mathbf{I}_A(x)$ a hazy image, although it is $\mathbf{I}(x) - \mathbf{A}$.

In order to estimate clear image exactly, our goal can be expressed as following:

$$\min ||F(\mathbf{I}_A) - \mathbf{J}_A||^2, \tag{6}$$

$F(\mathbf{I}_A)$ represents the output of the network, X represents the truth $\mathbf{J}_A$.

In order to effectively and efficiently solve the dehazing problem, we utilize a preprocessing to input hazy image. This preprocessing converts our problem from image color domain to image details domain. This preprocessing is inspired by classification using DCNN. Our DCNN method can benefit from this preprocessing, which makes our network easy to train.

It is worthy to point out that it is hard to train a network for predicting a haze-free image from hazy image directly. We first attempt to train such a network, and test the output of network, we find that the output can not be regarded as haze-free image. So our method benefits much from learning $\mathbf{J}_A$ from $\mathbf{I}_A$, which makes our method remove haze from hazy image without estimating transmission.

Estimating Initial Clear Image: Based on the fact that weights sharing allows for relatively larger interactive range than other fully connected structures, we choose convolutional neural network (CNN) architecture. Our CNN architecture is very simple, which can be implemented easily. For convenience, we denote Y as $\mathbf{I}_A(x)$. In Fig 1, we show the architecture of our network. Our convolutional network architecture can be expressed as:

$$F^0(Y) = Y, \tag{7}$$

$$F^n(Y) = \max(W^n * F^{(n-1)}(Y) + B^n, 0), n = 1, \ldots, 5 \tag{8}$$

$$F_W(Y) = W^n * F^{(n-1)}(Y) + B^n, n = 6 \tag{9}$$

n represents the layers, which ranges from 1 to 5. Our convolutional network architecture consists of five layers and contains four hidden layers for convolution generation. For the bottom layer with index 0, which is the input layer and expressed by equation 7.

In each intermediate layer, which is expressed by equation 8, represents a convolution process for the nodes in the convolution network regarding its neighbors. By convention, $*$ represents the convolution operation, W^n represents the convolution kernel and B^n is the bias. The top layer with $F_W(Y)$ in equation 9 generates the $\tilde{\mathbf{J}}_A(x)$ from the network.

Once $\tilde{\mathbf{J}}_A(x)$ is predicted, it is added with $\mathbf{A}(x)$, which can generate a high quality image dehazing result. Although our dehazing result is very similar to haze-free image, it contains block artifacts. Next we will introduce how to use the output of the network to get a more visual pleasing result.

Remove Color Distortion and Block Artifacts: Our result from the network contains some color distortion and block artifacts, which is induced by the network and the assumption of constant transmission in a small patch. We apply an regularization operation to remove the block artifacts and color distortion. Because we recover $\mathbf{J}_A(x)$ from $\mathbf{I}(x)_A(x)$, we can use this information and $\mathbf{I}_A(x)$ to recover the init transmission of each pixel using 10.

$$\tilde{t}(x) = r(x)/\tilde{r}(x), \tag{10}$$

then we use guided filter to get a smooth transmission map $t(x)$.

Dehazing: Once the transmission map is estimated, we can recover the haze-free image using Equation 1 :

$$J(x) = \frac{\mathbf{I}(x) - \mathbf{A}}{t(x)} + \mathbf{A} \tag{11}$$

In Fig. 2, we show an example of our method, which is summarized in Algorithm 1. Fig. 2(a) shows the input hazy image. Our haze removal result is shown in Fig. 2(b). Fig. 2(c) shows the distance in RGB space of every pixel in the hazy image to the airlight. Fig. 2(d) shows the estimated distance of clear pixel to the airlight. Since larger values are represented by brighter colors, this indicates that the distance to the airlight is increased. Fig. 2(e) shows the initial transmission calculate using Equation 10. Fig. 2(f) shows the final transmission.

Train Data Preparation and Train DehazeCNN Network: It is hard to collect the vast amount of labelled data for training DehazingCNN [13], so we adopt same strategy as [12,10,13]. Based on the two assumptions [12]: first, the transmission in a local patch is constant. Second, image content has no relation with the transmission. Given a haze-free image patch, we assume that the transmission for this patch is constant with value t, we use the equation 1 to get a hazy image patch. After we get the hazy image patches and the corresponding haze-free image patches, we can get a pair of $\mathbf{I}_A(x)$ and $\mathbf{J}_A(x)$, we use these data to train our network. In this paper, we train a model for each value of A. In this paper, we use patch size 16×16, we implements our CNN architecture using Caffe package [17], which is a popular tool of deep learning. We use stochastic

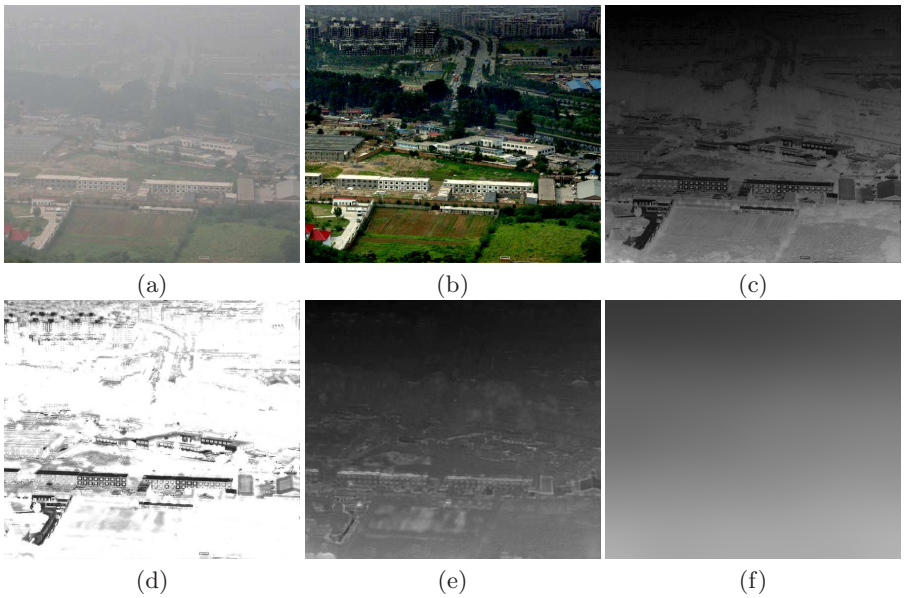

Fig. 2: Intermediate and final results of our method: (a) An input hazy image; (b) The output image; (c) The distance $r(x)$ of every pixel of the hazy image to the airlight; (d) The estimate distance $\tilde{r}(x)$; (e) The init $\tilde{t}(x)$; (f) The final $t(x)$.

gradient descent (SGD) to train DehazingCNN. We provide demo with a model train with $A = [0.78, 0.78, 0.78]$ and $A = [0.5, 0.6, 1]$, which can be found on https://github.com/killsking/DehazingCNN.

It is worth nothing to point out that the commonly used constant assumption on the transmission in a local patch results in block artifacts [7]. Because we use the constant assumption on the transmission in a local patch to prepare the training data, our result also exists block artifacts, we use guided filter [18] to eliminate the block artifacts.

It is also worthy to point out that our work can be extend to a complete end-to-end work for dehazing by using DCNN to remove block artifacts of our result. In Fig. 3, ourJ represents the result of $\tilde{\mathbf{J}}_A(x) + \mathbf{A}$. As we can see from the Fig. 3, ourJ is high quality for first image. But for second hazy image, we find that ourJ contains some block artifacts, so we need to use guided filter or anther DCNN model to eliminate the block artifacts.

3 Experiment Results

In this section, we evaluate our method on a large dataset containing both synthetic and natural images and compare our performance to state-of-the-art methods [7,19,20,13,14]. In this section, we use the $L1err = \frac{1}{N}\sum_{c\in R,G,B}|\mathbf{J}^c - \mathbf{G}^c|$ as metric, where $\mathbf{J}$ resents the dehazing result image and $\mathbf{G}$ resents the ground

Algorithm 1 Our proposed single image dehazing framework.

Input: The hazy image.
Output: The haze free image.
1: Compute the atmospheric light using Meng et al.'s method;
2: $\mathbf{I}_A(x) = \mathbf{I}(x) - \mathbf{A}$;
3: Compute $r(x)$ using Equation 3;
4: Compute $\tilde{\mathbf{J}}_A(x)$ using deep convolutional network;
5: Compute $\tilde{r}(x)$ using Equation 5.
6: Compute transmission:$\tilde{t} = r(x)/\tilde{r}(x)$.
7: Get the final transmission $t(x)$ map using guided filter.
8: Get the finial dehazing result using Equation 11.

truth image. We convert the dehazing result and ground truth image into range of $[0, 1]$. In order to evaluate the dehazing methods, we generate an indoor hazy image dataset. This dataset is based on the indoor RGBD dataset [21], we use $\mathbf{A} = [0.78, 0.78, 0.78]$ and choose three value for β as $0.06, 0.3, 0.54$ to generate hazy image [14].

3.1 Synthetic hazy images

In this subsection, we compare our method with state-of-the-art methods on both indoor and outdoor synthetic hazy images. Although our deep network is trained on synthetic outdoor images, we note that it can be applied for indoor images as well. First, we compare our method with other state-of-the-art methods, and list the overall results. Second, we show some results on hazy images in Fattal's dataset and some images in our dataset.

An outdoor synthetic hazy images dataset was introduced by [19], which is available online. For this dataset, we sum all L1 error as quality standard. As shown in table 1a, we can see that our results are best. In this section, we also compare our method with some state-of-art methods [7,19,20] on some images in dataset. As shown in Fig. 3, OurJ is the result of adding $\mathbf{A}$ to $\mathbf{J}_A(x)$, which is the output of our DCNN.

We compare our method with some state-of-art methods [14,20]. We use our synthetic hazy images to evaluate our performance, the result is shown in table 1b

type	He	Fattal	Berman	Our
outdoor	5.77	4.54	5.96	3.87

(a) Fattal's dataset.

type	Ren	Berman	Our
indoor	1.58e+03	1.05e+03	0.85e+03

(b) Our dataset.

Table 1: Overall comparison. Red color indicates the best results and blue indicates the second.

Fig. 3: Comparison on Outdoor hazy images.

Fig. 4: Comparison on Indoor hazy images. The number in left is SSIM value and the right is L1ERR

Outdoor Hazy Images: In this part, we also compare our method with some state-of-art methods [7,19,20] on every image in dataset. We show the results in Table 2. As we can see from the results, our method can get a very similar results to the ground truths in general and also can get high quality result for particular image. In Fig. 3 we show some results on two hazy images. As we can see our network output is very similar to haze-free image.

Indoor Hazy Images:In this part, we compare our method with Ren et al. [14] and Berman et al. [20]. The structural similarity (SSIM) image quality assessment index [22] is used to evaluate performance of the methods. The higher value of SSIM shows the dehazing result is better. First we compare all image in our dataset, we can get highest SSIM score for 2393 images in 4347 image. Second, we show some results use SSIM and L1Err.

3.2 Quantitative Evaluation on Benchmark Natural Images Dataset

In this subsection, we compare our method with state-of-the-art methods. As previously pointed by [7], the image after dehazing might look dim, since the scene radiance is usually not as bright as the airlight. For display, we perform a global linear contrast stretch on the output, clipping 0.05% of the pixel values

image	He	Fattal	Berman	Our
D1	0.135	0.130	0.308	0.127
D2	0.117	0.107	0.116	0.113
D3	0.112	0.083	0.170	0.087
S10	0.140	0.160	0.127	0.130
S25	0.164	0.291	0.171	0.174
S50	0.226	0.400	0.241	0.226
church	0.117	0.107	0.116	0.113

(a) Church.

image	He	Fattal	Berman	Our
D1	0.150	0.0915	0.299	0.086
D2	0.142	0.084	0.060	0.062
D3	0.152	0.050	0.107	0.074
S10	0.146	0.094	0.072	0.077
S25	0.165	0.156	0.122	0.116
S50	0.213	0.269	0.224	0.186
lawn1	0.142	0.086	0.060	0.062

(b) Lawn1.

image	He	Fattal	Berman	Our
D1	0.92	0.067	0.267	0.072
D2	0.085	0.052	0.100	0.052
D3	0.088	0.036	0.198	0.044
S10	0.080	0.065	0.115	0.060
S25	0.089	0.106	0.146	0.084
S50	0.149	0.159	0.190	0.134
mansion	0.085	0.049	0.100	0.052

(c) Mansion.

image	He	Fattal	Berman	Our
D1	0.121	0.08	0.300	0.106
D2	0.101	0.069	0.080	0.083
D3	0.093	0.055	0.148	0.056
S10	0.115	0.081	0.090	0.093
S25	0.142	0.139	0.125	0.121
S50	0.202	0.243	0.192	0.179
road1	0.101	0.069	0.080	0.083

(d) Road1.

Table 2: Quantitative comparison. Red color indicates the best results and blue indicates the second.

both in the shadows and in the highlights. In Fig. 5 we compare our method with state-of-the-art methods. Some of the results are provided by Fattal [19] , Berman [20] and Cai [13], which are online. We also get some results via the program provided by Ren [14]. As shown in Fig. 5, Ancuti et al.'s method [23] can't remove haze completely, while Meng et al. 's method [16] exits the problem of overenhancement. The result of Luzón-González et al. [24] can't deal with the boundary between segments well, which results in a lot of artifacts. He et al.'s method can yield an excellent result in general but lack some micro-contrast details when compared to [19] and ours. This is obvious in the zoomed-in buildings shown in Cityscape results, where in our result and [19] the windows are more clear than in [7]. We also find that the result of Ren et al. losses some details of tree in Cityscape. In contrast, our method can deal this area well.The results of velly show that our method gets a result of high quality, which can recover more image details and vivid color than other state-of-the-art methods.

4 Conclusions

In this paper, we address the haze removal problem using a deep network which learns effective features to estimate the corresponding haze-free image for a hazy image. Compared to previous learning-based methods which learn a relation between hazy image and transmission map, the proposed learning method tries

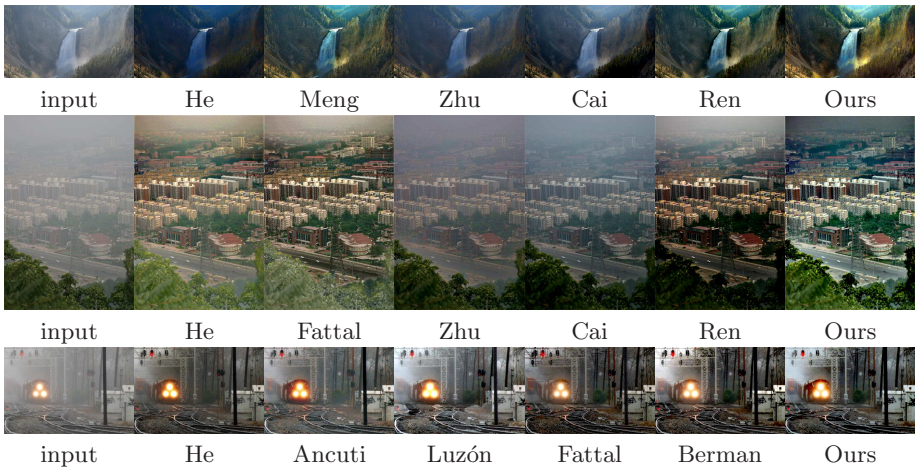

Fig. 5: Comparison on natural images:[Left] Input images. [Right] Our result. Middle columns display results by several methods, since each paper reports results on a different set of images.

to restore the corresponding haze-free image directly from hazy image without estimating of transmission map. Experimental results show our method can get a haze-free image directly from hazy image. As shown in experimental results, the output of CNN network can be regarded as haze-free results. To the best of our knowledge, our work is the first work which learns a map between the hazy image and haze-free image. Our method also maybe fail for some hazy images, which contain a lot of depth jumps. For these hazy images we can remove the block artifacts using guided filter. Experimental results on synthetic outdoor and indoor images and real images show the effectiveness of the proposed algorithm. Our experimental results show our method yields higher quantitative and qualitative performance than existing state-of-the-art methods.

References

1. Schechner, Y.Y., Narasimhan, S.G., Nayar, S.K.: Instant dehazing of images using polarization. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1., IEEE (2001) 327–325
2. Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D.: Deep photo: Model-based photograph enhancement and viewing. ACM Transactions on Graphics (TOG) **27** (2008) 116
3. Tan, R.T.: Visibility in bad weather from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2008) 1–8
4. Tarel, J.P., Hautiere, N.: Fast visibility restoration from a single color or gray level image. In: IEEE International Conference on Computer Vision, IEEE (2009) 2201–2208
5. Fattal, R.: Single image dehazing. ACM Transactions on Graphics (TOG) **27** (2008) 1–9

6. Kratz, L., Nishino, K.: Factorizing scene albedo and depth from a single foggy image. IEEE International Conference on Computer Vision **30** (2009) 1701 – 1708
7. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 2341–2353
8. Nishino, K., Kratz, L., Lombardi, S.: Bayesian defogging. International Journal of Computer Vision **98** (2012) 263–278
9. Gibson, K.B., Nguyen, T.Q.: An analysis of single image defogging methods using a color ellipsoid framework. Eurasip Journal on Image and Video Processing **2013** (2013) 1–14
10. Zhu, Q., Mai, J., Shao, L.: A fast single image haze removal algorithm using color attenuation prior. IEEE Transactions on Image Processing **24** (2015) 3522–3533
11. Li, Z., Zheng, J.: Edge-preserving decomposition-based single image haze removal. IEEE Transactions on Image Processing **24** (2015) 5432–5441
12. Tang, K., Yang, J., Wang, J.: Investigating haze-relevant features in a learning framework for image dehazing. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2014) 2995–3002
13. Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: Dehazenet: An end-to-end system for single image haze removal. arXiv preprint arXiv:1601.07661 (2016)
14. Ren, W., Liu, S., Zhang, H., Pan, J., Cao, X., Yang, M.H.: Single image dehazing via multi-scale convolutional neural networks. In: European Conference on Computer Vision, Springer (2016) 154–169
15. Harald, K.: Theorie der horizontalen Sichtweite: Kontrast und Sichtweite. Volume 12. Keim & Nemnich (1924)
16. Meng, G., Wang, Y., Duan, J., Xiang, S., Pan, C.: Efficient image dehazing with boundary constraint and contextual regularization. In: IEEE International Conference on Computer Vision. (2013) 617–624
17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM International Conference on Multimedia, ACM (2014) 675–678
18. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013) 1397–1409
19. Fattal, R.: Dehazing using color-lines. ACM Transactions on Graphics (TOG) **34** (2014) 13
20. Berman, D., treibitz, T., Avidan, S.: Non-local image dehazing. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 1674–1682
21. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision, Springer (2012) 746–760
22. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13** (2004) 600–612
23. Ancuti, C.O., Ancuti, C., Hermans, C., Bekaert, P.: A fast semi-inverse approach to detect and remove the haze from a single image. In: Asian Conference on Computer Vision, Springer (2011) 501–514
24. Luzón-González, R., Nieves, J.L., Romero, J.: Recovering of weather degraded images based on rgb response ratio constancy. Applied Optics **54** (2015) B222–B231