Image Dehazing by Joint Estimation of Transmittance and Airlight using Bi-Directional Consistency Loss Minimized FCN

Ranjan Mondal, Sanchayan Santra, Bhabatosh Chanda

Electronics and Communication Sciences Unit
Indian Statistical Institute, Kolkata, India

Presented at: CVPR Workshop: NTIRE 2018, Salt Lake City, Utah

Image Dehazing

Figure: Hazy image and its dehazed version

Imaging Equation

Image formation in haze is modeled as follows [1]

$$I(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))A \qquad (1)$$

Common Assumptions,

- For RGB image $I(\mathbf{x})$, $J(\mathbf{x})$ and A are 3×1 vectors and $t(\mathbf{x})$ is a scalar, assuming it to be constant across color channels.
- The environmental illumination (A) is constant in the whole scene.

Imaging Equation

Assumption,

- For RGB image $I(\mathbf{x})$, $J(\mathbf{x})$ and A are 3x1 vectors and $t(\mathbf{x})$ is a scalar, assuming it to be constant across color channels.
- The environmental illumination (A) is constant in the whole scene.

This constant environmental assumption is true only when the sky is overcast[1].

Relaxed Imaging Equation

$$I(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))A, \tag{2}$$

is changed to

$$I(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))A(\mathbf{x}). \tag{3}$$

Where, A is changed to $A(\mathbf{x})$ to account for the space-variant illumination within an image.

Relaxed Imaging Equation

We don't use relaxed equation in its original form. Instead we use the following,

$$I(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))A(\mathbf{x}), \tag{4}$$

$$= J(\mathbf{x})t(\mathbf{x}) + K(\mathbf{x}). \tag{5}$$

Threfore, given an image we try to estimate $t(\mathbf{x})$ and $K(\mathbf{x})$ using a CNN.

Challenges

- Interdependence of $t(\mathbf{x})$ and $K(\mathbf{x})$.
- The NTIRE Dehazing dataset contains only Hazy image and its corresponding clear image.
- The input image can be of various sizes.

Challenges

- Interdependence of $t(\mathbf{x})$ and $K(\mathbf{x})$.
- The NTIRE Dehazing dataset contains only Hazy image and its corresponding clear image.
- The input image can be of various sizes.

Proposed way,

- Joint estimation of $t(\mathbf{x})$ and $K(\mathbf{x})$.

CNN Architecture

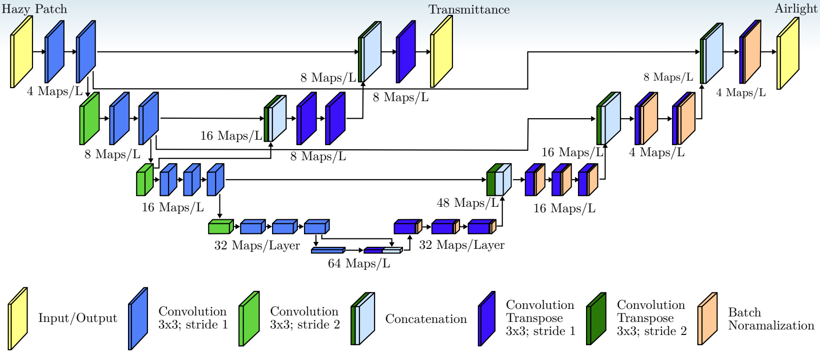


Figure: Architecture of our proposed CNN

Input: $I(\mathbf{x})$ (M$\times$N$\times$3) and Output: $t(\mathbf{x})$ (M$\times$N)and $K(\mathbf{x})$ (M$\times$N$\times$3).

CNN Architecture

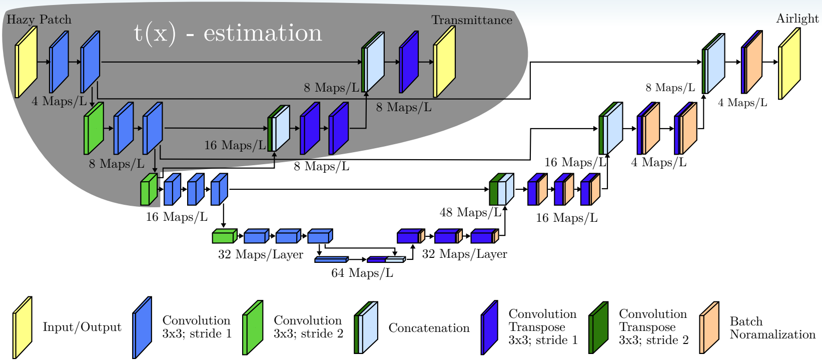


Figure: Architecture of our proposed CNN: $t(\mathbf{x})$ estimation path

CNN Architecture

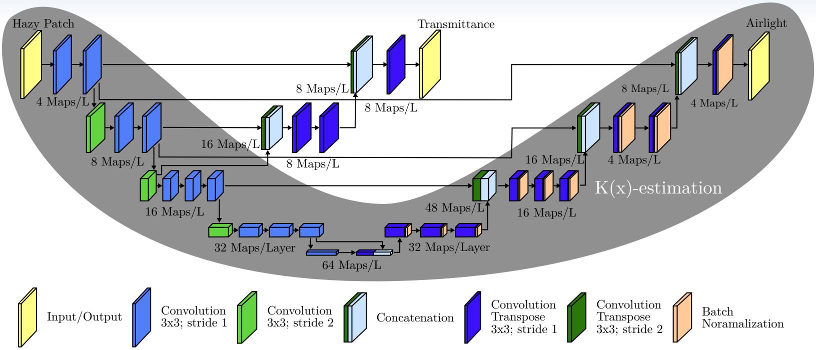


Figure: Architecture of our proposed CNN: $K(\mathbf{x})$ estimation path

Challanges

- Interdependence of $t(\mathbf{x})$ and $K(\mathbf{x})$.
- The NTIRE Dehazing dataset contains only Hazy image and its corresponding clear image.
- The input image can be of various sizes.

Proposed way,

- Bi-directional Consistency Loss

Bi-directional Consistency Loss

$$L = \frac{1}{N}\sum_{\mathbf{x}}\big(L_1(\mathbf{x}) + L_2(\mathbf{x})\big) \tag{6}$$

$$\text{Forward Loss: } L_1(\mathbf{x}) = |I(\mathbf{x}) - J(\mathbf{x})t'(\mathbf{x}) - K'(\mathbf{x})| \tag{7}$$

$$\text{Backward Loss: } L_2(\mathbf{x}) = \left| J(\mathbf{x}) - \frac{I(\mathbf{x}) - K'(\mathbf{x})}{\max\{t'(\mathbf{x}), \epsilon\}} \right|. \tag{8}$$

- Works with only input image ($I(\mathbf{x})$) and ground truth clean image ($J(\mathbf{x})$).
- This also avoids the case when a small error in $t(\mathbf{x})$ deviates the dehazed output quite a bit.
- This also ensures the network converges to the correct solution only.

Challanges

- Interdependence of $t(\mathbf{x})$ and $K(\mathbf{x})$.
- The NTIRE Dehazing dataset contains only Hazy image and its corresponding clear image.
- The input image can be of various sizes.

Proposed way,

- Multi-level approach.

Multi-level Training

The network needs to be trained so that it handles scale variation. So, training data generation we do the following.

- Extract overlapping patches from the both clear and corresponding hazy images.
- In the first level, we take patches of size $P \times P$, where $P = \min\{H, W\}$ for a $H \times W$ image.
- In the second level, we extract patches of size $\frac{P}{2} \times \frac{P}{2}$.
- In the third level patch size becomes $\frac{P}{4} \times \frac{P}{4}$.
- This halving process is repeated until the patch size falls below 128×128.

All the extracted patches are resized to 128×128 before they are used for training.

Multi-level estimation

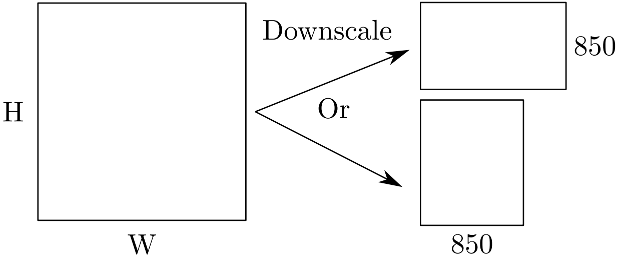
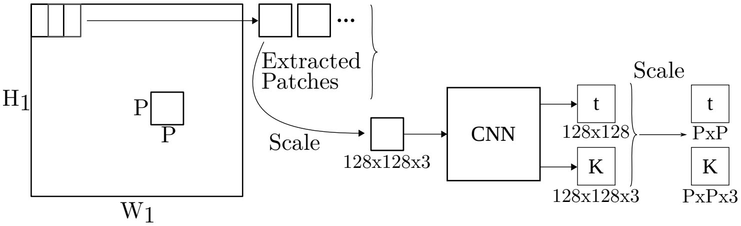


Figure: Step1: Down-scaling

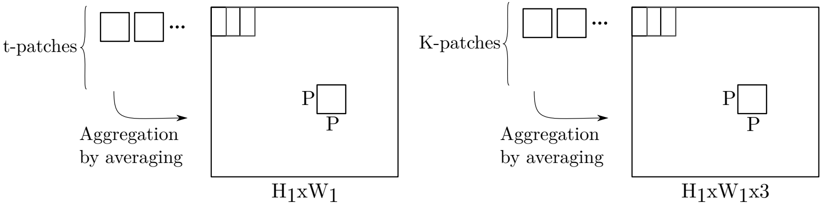
Multi-level estimation of $t(\mathbf{x})$ and $K(\mathbf{x})$



This is done for patch sizes $\mathrm{P}\times\mathrm{P} = 256 \times 256$, 384×384 and 512×512.

Aggregation of patches

We have patches with three different sizes (256×256, 384×384 and 512×512). We aggregate similar sized patches.



So, we will get three t and three K maps of same size by using three different sized patches.

Aggregation of $t(\mathbf{x})$ and $K(\mathbf{x})$

We have obtained three t and three K maps. We need to aggregate them to form single transmittance and airlight map in the following way,

$$t(\mathbf{x}) = \frac{\sum_{i=1}^{l} w_i^{(t)} t_i(\mathbf{x})}{\sum_{i=1}^{l} w_i^{(t)}}, \tag{9}$$

$$K(\mathbf{x}) = \frac{\sum_{i=1}^{l} w_i^{(K)} K_i(\mathbf{x})}{\sum_{i=1}^{l} w_i^{(K)}}. \tag{10}$$

Here $t_i(\mathbf{x})$ and $K_i(\mathbf{x})$ denote the estimates we have obtained at level i and l denotes the number of levels we operate on. Here, we have taken all the weights to be 1.

Dehazed Output with the aggregated $t(\mathbf{x})$ and $K(\mathbf{x})$

We have used Guided Filter[2] to refine (smooth) them.

After Guided Filter

Results



(a) Hazy Image (b) Berman *et al.*[3] (c) Cai *et al.*[4]



(d) Ren *et al.*[6] (e) Ours (f) Ground Truth

Figure: Comparison on image *39* of NTIRE Hazy validation dataset

Results

	Image	Berman *et al.*[3]	Cai *et al.*[4]	Ren *et al.*[6]	Ours
Indoor	26	12.42/0.65/20.15	10.17/0.69/24.64	11.02/0.72/22.36	**15.71**/**0.78**/**13.86**
	27	14.8/0.66/18.03	14.51/0.67/17.74	17.61/0.77/12.31	**21.94**/**0.77**/**8.25**
	28	13.3/0.62/19.24	13.39/0.72/17.7	13.11/0.72/17.06	**16.15**/**0.73**/**13.71**
	29	14.67/0.67/15.73	11.91/0.55/20.78	17.6/0.84/11.43	**21.88**/**0.83**/**9.33**
	30	13.93/0.61/19.09	15.53/0.71/15.16	16.79/0.73/14.21	**20.66**/**0.73**/**12.19**
Outdoor	36	16.92/0.58/14.43	16.59/0.64/13.17	19.46/0.68/11.84	**23.23**/**0.68**/**7.6**
	37	14.99/0.52/15.14	15.76/0.57/15.36	17.73/0.6/13.27	**21.4**/**0.63**/**8.53**
	38	15.55/0.64/16.92	13.25/0.6/21.85	16.21/0.66/19.02	**22.4**/**0.69**/**8.52**
	39	17.65/0.62/16.43	12.78/0.57/20.71	15.75/0.61/16.74	**19.95**/**0.64**/**10.84**
	40	17.04/0.61/15.06	16.53/0.67/11.62	18.67/0.7/11.96	**22.2**/**0.71**/**7.85**
	Average	15.13/0.62/17.02	14.04/0.64/17.87	16.39/0.7/15.02	**20.55**/**0.72**/**10.07**

Table: Quantitative comparison of PSNR, SSIM, CIEDE2000 (lower better) values on NTIRE hazy dataset

Results

Image	Berman *et al.*[3]	Cai *et al.*[4]	Li *et al.*[5]	Ren *et al.*[6]	Ours
church	**15.69**/0.88/**16.91**	14.64/0.82/20.45	9.44/0.61/34.64	14.18/0.85/20.26	14.47/**0.89**/24.4
couch	17.28/0.86/14.18	16.71/0.82/14.34	16.79/0.82/17.33	18.02/**0.87**/**12.92**	**19.54**/0.84/12.94
dolls	15.71/0.8/15.74	16.26/0.81/12.43	**17.24**/**0.82**/**10.88**	16.95/0.83/12.38	14.91/0.81/13.51
flower1	12.15/0.71/20.99	19.81/0.94/16.72	12.21/0.79/29.42	9.08/0.42/24.65	**21.35**/**0.94**/**14.72**
flower2	11.86/0.67/21.17	19.44/0.91/15.37	13.13/0.78/25.27	10.82/0.59/22.45	**22.75**/**0.94**/**11.39**
lawn1	14.78/0.83/**17.93**	13.8/0.81/23.01	11.33/0.67/31.74	14.38/0.8/21.0	**16.17**/**0.86**/20.22
lawn2	**15.32**/0.85/**17.81**	13.61/0.81/22.47	10.98/0.66/31.7	13.3/0.76/22.27	14.91/**0.86**/20.92
mansion	17.34/0.87/15.84	17.39/0.84/17.42	14.23/0.69/24.01	17.7/0.87/17.53	**21.89**/**0.92**/**13.65**
moebius	14.59/0.83/22.4	**19.18**/**0.94**/16.38	13.21/0.76/27.61	16.38/0.89/19.86	18.22/0.89/**15.29**
raindeer	16.6/0.8/15.28	17.87/0.84/13.73	16.54/0.79/18.5	16.83/0.8/15.49	**22.66**/**0.89**/**10.71**
road1	**16.33**/0.87/19.06	13.73/0.79/22.2	11.75/0.65/29.32	14.13/0.82/22.22	16.17/**0.89**/18.42
road2	**18.23**/**0.89**/**16.83**	13.22/0.77/23.43	11.95/0.61/30.96	16.45/0.86/20.18	15.89/0.9/20.79
Average	15.49/0.82/17.84	16.31/0.84/18.16	13.23/0.72/25.95	14.85/0.78/19.27	**18.24**/**0.89**/**16.41**

Table: Quantitative comparison of PSNR, SSIM, CIEDE2000 (lower better) values on Fattal dataset

Failure Cases

(a) Input

(b) Berman *et al.*[3]

(c) Li *et al.*[5]

(d) Our

(e) Transmittance

(f) Airlight

Figure: Failure on *canon7* image

References I

S. G. Narasimhan and S. K. Nayar.
Vision and the Atmosphere.
International Journal of Computer Vision, 48(3):233–254, July 2002.

K. He, J. Sun, and X. Tang.
Guided Image Filtering.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(6):1397–1409, June 2013.

D. Berman, T. Treibitz, and S. Avidan.
Non-local Image Dehazing.
In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1674–1682, June 2016.

References II

B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao.
DehazeNet: An End-to-End System for Single Image Haze Removal.
IEEE Transactions on Image Processing, 25(11):5187–5198, Nov. 2016.

B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng.
AOD-Net: All-In-One Dehazing Network.
In *The IEEE International Conference on Computer Vision (ICCV)*, pages 4770–4778, Oct 2017.

W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang.
Single Image Dehazing via Multi-scale Convolutional Neural Networks.
In *European Conference on Computer Vision*, pages 154–169. Springer, 2016.

Thank You