Proximal Dehaze-Net: A Prior Learning-Based Deep Network for Single Image Dehazing (Supplementary Material)

Dong Yang and Jian Sun

Abstract

This is a supplementary material for ECCV 2018 paper "Proximal Dehaze-Net: A Prior Learning-Based Deep Network for Single Image Dehazing". In this material, we first supplement some necessary deductions of the updating formulas for Q, T and U. We also include the computations of gradients of final loss with respect to the variables. Second, we show the computation graph of GIF-Block and show that it is a differential part of our end-to-end trainable system.

1 Preliminaries

The haze imaging model used in our submission is

$$I(\mathrm{x}) = J(\mathrm{x})T(\mathrm{x}) + A(1 - T(\mathrm{x})), \tag{1}$$

where $I \in \mathbb{R}^{M\times N\times 3}$ is the haze degraded image, $J \in \mathbb{R}^{M\times N\times 3}$ is the latent haze-free image, $T \in \mathbb{R}^{M\times N}$ is the media transmission and $A \in \mathbb{R}^3$ is a global atmospheric light. We assume a known A ,then for each color channel $c \in \{r, g, b\}$, divide by A^c on both size of Eqn. (1), we get

$$\frac{I^c(\mathrm{x})}{A^c} = \frac{J^c(\mathrm{x})}{A^c}T(\mathrm{x}) + (1 - T(\mathrm{x})). \tag{2}$$

For simplicity, let $P^c = \frac{I^c}{A^c}$, $Q^c = \frac{J^c}{A^c}$ and we can rewrite Eqn. (2) in a concise form as

$$P^c = Q^c \circ T + 1 - T, \tag{3}$$

where $\circ$ is the element-wise multiplication for matrices. In such a way, P denotes the scaled hazy image and Q denotes the scaled haze-free image.

The dark channel [1] of a color image P is defined as

$$P^{dk}(\mathrm{x}) = \min_{c\in\{r,g,b\}} \left(\min_{\mathrm{y}\in\Omega(\mathrm{x})} P^c(\mathrm{y}) \right). \tag{4}$$

Assume that T is constant within a local patch, i.e. $T(\mathrm{y}) = T(\mathrm{x}),\ \mathrm{y} \in \Omega(\mathrm{x})$. Compute dark channel on both size of Eqn. (3) and we get

$$\begin{aligned} P^{dk}(\mathrm{x}) &= \min_{c\in\{r,g,b\}} \left(\min_{\mathrm{y}\in\Omega(\mathrm{x})} Q^c(\mathrm{y})T(\mathrm{y}) + (1 - T(\mathrm{y})) \right) \\ &= \min_{c\in\{r,g,b\}} \left(\min_{\mathrm{y}\in\Omega(\mathrm{x})} Q^c(\mathrm{y}) \right) T(\mathrm{x}) + (1 - T(\mathrm{x})) \\ &= Q^{dk}(\mathrm{x})T(\mathrm{x}) + (1 - T(\mathrm{x})). \end{aligned} \tag{5}$$

where P^{dk} and Q^{dk} are dark channels of P and Q respectively.

By forcing Eqns. (3) and (5) as loss terms in both image space and dark channel space, and adding regularizations on T and Q^{dk}, we can build an energy function

$$\begin{aligned} E(Q,T) = &\frac{\alpha}{2}\sum_{c}\|Q^c \circ T + 1 - T - P^c\|_F^2 \\ &+ \frac{\beta}{2}\|Q^{dk} \circ T + 1 - T - P^{dk}\|_F^2 + f(T) + g(Q^{dk}), \end{aligned} \tag{6}$$

in which $\alpha > 0$ and $\beta > 0$ are coefficients of data terms.

Solving for Q, T by minimizing Eq (6) is a ill-posed problems as there are multiple solutions. Unlike other image inverse problems, such as denoising, super-resolution and non-blind deblurring, single image dehazing behaves as a blind inverse problem and is apparently more difficult to be settled.

Using half-quadratic splitting algorithm to solve the above energy function, we introduce an auxiliary variable U to substitute Q^{dk} and we get the augmented energy function

$$\begin{aligned} E(Q,T,U) = &\frac{\alpha}{2}\sum_{c}\|Q^c \circ T + 1 - T - P^c\|_F^2 + \frac{\beta}{2}\|U \circ T + 1 - T - P^{dk}\|_F^2 \\ &+ \frac{\gamma}{2}\|U - Q^{dk}\|_F^2 + f(T) + g(U), \end{aligned} \tag{7}$$

where $\gamma > 0$ is a penalty parameter. When $\gamma \to \infty$, the solution to minimize Eqn. (7) converges to that of minimizing Eqn. (6).

The minimization of Eqn. (7) is achieved by alternatively solving three optimization sub-problems under HQS algorithm in an iterative manner. Let $Q_0 = P$, T_0 be a matrix of 1s in size of $M \times N$ and U_0 be a matrix of zeros in size of $M \times N$. For the n-th iteration, given U_{n-1}, T_{n-1} and Q_{n-1}

$$\begin{aligned} U_n &= \arg\min_{U} \frac{\beta}{2}\|T_{n-1} \circ U - (P^{dk} + T_{n-1} - 1)\|_F^2 + \frac{\gamma}{2}\|U - Q_{n-1}^{dk}\|_F^2 + g(U), \\ T_n &= \arg\min_{T} \frac{\alpha}{2}\sum_{c}\|Q_{n-1}^c \circ T - (P^c + T - 1)\|_F^2 + \frac{\beta}{2}\|U_n \circ T - (P^{dk} + T - 1)\|_F^2 + f(T), \\ Q_n &= \arg\min_{Q} \frac{\alpha}{2}\sum_{c}\|T_n \circ Q^c - (P^c + T_n - 1)\|_F^2 + \frac{\gamma}{2}\|Q^{dk} - U_n\|_F^2. \end{aligned} \tag{8}$$

2 Deductions of Updating Formulas

In this section, we give a detailed description on how optimization problems (8) are solved. But first, we shall introduce the proximal operator [3]

$$\text{prox}_{\lambda R}(V) = \arg\min_{Z} \frac{1}{2}\|Z - V\|_F^2 + \lambda R(Z). \tag{9}$$

In our submission, we further assume that regularization $R(Z)$ is separable on Z, i.e.

$$R(Z) = \sum_{\text{x}} R(Z(\text{x})).$$

Strict as it is, the assumption is true for many common regularizations, such as L-1 or L-2 norm, anisotropic total variation, etc.

The sub-problem of updating U_n and T_n in problem (8) can be interpreted in a general form

$$Z^* = \arg\min_{Z} \sum_{k=1}^{K} \frac{\mu_k}{2}\|A_k \circ Z - B_k\|_F^2 + R(Z), \tag{10}$$

where $\mu_k > 0$ is a positive coefficient, A_k, B_k, Z are matrices in size of $M \times N$ and $R(Z)$ is some regularization

separable on Z. Then Eqn. (10) can be solved as

$$\begin{aligned} Z^* &= \arg\min_Z \sum_{\mathrm{x}} \sum_{k=1}^{K} \frac{\mu_k}{2} A_k^2(\mathrm{x}) Z^2(\mathrm{x}) - \mu_k A_k(\mathrm{x}) B_k(\mathrm{x}) Z(\mathrm{x}) + R(Z(\mathrm{x})) \\ &= \arg\min_Z \sum_{\mathrm{x}} \frac{1}{2} Z^2(\mathrm{x}) - \frac{\sum_k \mu_k A_k(\mathrm{x}) B_k(\mathrm{x})}{\sum_k \mu_k A_k^2(\mathrm{x})} Z(\mathrm{x}) + \frac{R(Z(\mathrm{x}))}{\sum_k \mu_k A_k^2(\mathrm{x})} \\ &= \arg\min_Z \sum_{\mathrm{x}} \frac{1}{2} \Big[Z(\mathrm{x}) - \frac{\sum_k \mu_k A_k(\mathrm{x}) B_k(\mathrm{x})}{\sum_k \mu_k A_k^2(\mathrm{x})} \Big]^2 + \frac{R(Z(\mathrm{x}))}{\sum_k \mu_k A_k^2(\mathrm{x})} \\ &= \arg\min_Z \frac{1}{2} \Big\| Z - \frac{\sum_k \mu_k A_k \circ B_k}{\sum_k \mu_k A_k \circ A_k} \Big\|_F^2 + \frac{R(Z)}{\sum_k \mu_k A_k \circ A_k} \\ &= \mathrm{prox}_{\frac{1}{z} R}(\hat{Z}), \end{aligned} \tag{11}$$

where we denote $z = \sum_k \mu_k A_k \circ A_k$ and $\hat{Z} = \frac{1}{z} \sum_k \mu_k A_k \circ B_k$. Note that the divisions involved are element-wise performed on matrices.

Now assume that regularizations $f(T)$ and $g(U)$ are separable on T and U respectively. Using Eqn. (11), the updating of U and T in Eqn. (8) can be easily achieved.

Update U.

For the n-th iteration, given Q_{n-1} and T_{n-1}, minimizing Eqn. (7) with respect to U is equivalent to

$$U^* = \arg\min_U \frac{\beta}{2} \|T_{n-1} \circ U - (P^{dk} + T_{n-1} - 1)\|_F^2 + \frac{\gamma}{2} \|U - Q_{n-1}^{dk}\|_F^2 + g(U). \tag{12}$$

Using Eqn. (11), let $\mu_1 = \beta$, $\mu_2 = \gamma$ and substitute $A_1 = T_{n-1}$, $B_1 = P^{dk} + T_{n-1} - 1$, $A_2 = \mathbf{1}$ ($\mathbf{1}$ means a matrix with 1s in the same size as U), $B_2 = Q_{n-1}^{dk}$, then we get

$$\begin{aligned} b_n &= \beta T_{n-1} \circ T_{n-1} + \gamma, \\ \hat{U}_n &= \frac{1}{b_n} \big[\beta T_{n-1} \circ (P^{dk} + T_{n-1} - 1) + \gamma Q_{n-1}^{dk} \big], \\ U_n &= \mathrm{prox}_{\frac{1}{b_n} g}(\hat{U}_n). \end{aligned} \tag{13}$$

Update T.

For the n-th iteration, given Q_{n-1} and U_n, minimizing Eqn. (7) with respect to T is equivalent to

$$T^* = \arg\min_T \frac{\alpha}{2} \sum_c \|(Q_{n-1}^c - 1) \circ T - (P^c - 1)\|_F^2 + \frac{\beta}{2} \|(U_n - 1) \circ T - (P^{dk} - 1)\|_F^2 + f(T), \tag{14}$$

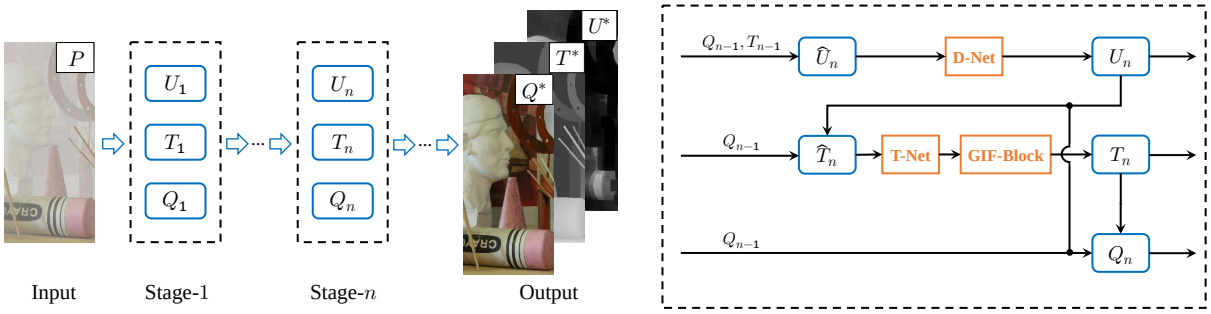
where $c \in \{r, g, b\}$ is a color channel. Using Eqn. (11), let $\mu_1 = \mu_2 = \mu_3 = \alpha$, $\mu_4 = \beta$ and substitute $A_1 = Q_{n-1}^r - 1$, $A_2 = Q_{n-1}^g - 1$, $A_3 = Q_{n-1}^b - 1$, $A_4 = U_n - 1$, $B_1 = P^r - 1$, $B_2 = P^g - 1$, $B_3 = P^b - 1$, $B_4 = P^{dk} - 1$, then we get

$$\begin{aligned} c_n &= \sum_c \alpha (Q_{n-1}^c - 1) \circ (Q_{n-1}^c - 1) + \beta (U_n - 1) \circ (U_n - 1), \\ \hat{T}_n &= \frac{1}{c_n} \big[\sum_c \alpha (Q_{n-1}^c - 1) \circ (P^c - 1) + \beta (U_n - 1) \circ (P^{dk} - 1) \big], \\ T_n &= \mathrm{prox}_{\frac{1}{c_n} f}(\hat{T}_n). \end{aligned} \tag{15}$$

Update Q.

For the n-th iteration, given U_n and T_n, minimize Eqn. (7) with respect to Q is equivalent to

$$Q^* = \arg\min_Q \frac{\alpha}{2} \sum_c \|Q^c \circ T_n + 1 - T_n - P^c\|_F^2 + \frac{\gamma}{2} \|Q^{dk} - U_n\|_F^2. \tag{16}$$



(a) Multi-stage network for image dehazing

(b) Network structure for the n–th stage

Figure 1: Structure of the proximal dehaze-net

To solve the above optimization problem, we first let $\mathcal{T}_n$ be a 3-D matrix of $M \times N \times 3$, with each color channel as T_n. We denote $\overrightarrow{Q}$ as the vectorized form of matrix Q. The operation to compute dark channel of an image is to extract the minimum of a local patch, which can be regarded as a linear transform D operating on the image vector such that $D\overrightarrow{Q} = \overrightarrow{Q^{dk}}$. The operator D satisfies that $D^\top D$ is diagonal. Then Eqn. (16) can be written as

$$\overrightarrow{Q}^* = \arg\min_{\overrightarrow{Q}} \frac{\alpha}{2}\|\overrightarrow{Q} \circ \overrightarrow{\mathcal{T}_n} + 1 - \overrightarrow{\mathcal{T}_n} - \overrightarrow{P}\|_2^2 + \frac{\gamma}{2}\|D\overrightarrow{Q} - \overrightarrow{U_n}\|_2^2. \tag{17}$$

Taking the gradient with respect to $\overrightarrow{Q}$ and let it be zero, then we get

$$\left[\alpha\overrightarrow{\mathcal{T}_n} \circ \overrightarrow{\mathcal{T}_n} + \gamma\mathrm{diag}(D^\top D)\right] \circ \overrightarrow{Q} - \left[\alpha\overrightarrow{\mathcal{T}_n} \circ (\overrightarrow{P} + \overrightarrow{\mathcal{T}_n} - 1) + \gamma D^\top \overrightarrow{U_n}\right] = 0.$$

We find that $\overrightarrow{\mathcal{T}_n} \circ \overrightarrow{\mathcal{T}_n}$, $\overrightarrow{\mathcal{T}_n} \circ (\overrightarrow{P} + \overrightarrow{\mathcal{T}_n} - 1)$ and $D^\top \overrightarrow{U_n}$ are diagonal matrices, so we get the optimal $\overrightarrow{Q}_n$ as

$$\overrightarrow{Q}_n = \frac{\alpha\overrightarrow{\mathcal{T}_n} \circ (\overrightarrow{P} + \overrightarrow{\mathcal{T}_n} - 1) + \gamma D^\top \overrightarrow{U_n}}{\alpha\overrightarrow{\mathcal{T}_n} \circ \overrightarrow{\mathcal{T}_n} + \gamma\mathrm{diag}(D^\top D)}. \tag{18}$$

We then reshape $\overrightarrow{Q}_n$ into size of $M \times N \times 3$ to get Q_n.

3 Computing Gradients for Back Propagation

3.1 Gradient computation of the whole network

In our submission, we extend $f(\cdot)$ and $g(\cdot)$ to be general regularizations that do not require to be separable and learn their corresponding proximal operators using deep convolutional neural networks. We thus unfold the above iterative algorithm into a deep neural network, dubbed as *proximal dehaze-net*, as shown in Fig. 1.

Our proximal dehaze-net is a multi-stage network, with each stage implementing an iteration of the HQS algorithm. For each stage of the proximal dehaze-net,

$$\begin{aligned} U_n &= \mathrm{prox}_{\frac{1}{b_n}g}(\hat{U}_n) = \text{D-Net}(\hat{U}_n, P; \Theta_{g,n}), \\ T_n &= \mathrm{prox}_{\frac{1}{c_n}f}(\hat{T}_n) = \text{GIF-Block}(\text{T-Net}(\hat{T}_n, P; \Theta_{f,n})), \end{aligned}$$

in which, we concatenate $\hat{U}_n$ and $\hat{T}_n$ with hazy image P for better incorporating the image structure information. D-Net and T-Net are CNNs and $\Theta_{g,n}$, $\Theta_{f,n}$ are network parameters for D-Net and T-Net respectively. GIF-Block following T-Net implements guided image filtering which is a part of our end-to-end trainable system as will be discussed later.

The gradients for training the network is obtained via back propagation. As shown in Algorithm 1, during back propagation, we first compute the gradients of final loss ℓ w.r.t. U_N, T_N and Q_N, denoted as $\partial\ell/\partial U_N$,

Algorithm 1 Back propagation of proximal dehaze-net

Input: Hazy image P, ground truth image Q, transmission map T and dark channel U.

Output: Gradients of final loss ℓ w.r.t. network parameters $\partial\ell/\partial\Theta_{g,n}$ and $\partial\ell/\partial\Theta_{f,n}$, $n = 1, 2, \ldots, N$.

Perform network forward propagation following Fig. 1 and get Q_N, T_N and U_N.
Compute $\partial\ell/\partial Q_N$, $\partial\ell/\partial T_N$ and $\partial\ell/\partial U_N$ from loss function.
for $n = N : -1 : 1$ **do**
  Given $\partial\ell/\partial Q_n$, compute $[\partial\ell/\partial T_n]_{Q_n}$ and $[\partial\ell/\partial U_n]_{Q_n}$ using Eqns. (24) (25).
  if $n = N$ **then**
    Let $\partial\ell/\partial T_n = \partial\ell/\partial T_N + [\partial\ell/\partial T_n]_{Q_n}$.
  else
    Let $\partial\ell/\partial T_n = [\partial\ell/\partial T_n]_{\hat{U}_{n+1}} + [\partial\ell/\partial T_n]_{Q_n}$.
  end if
  Given $\partial\ell/\partial T_n$, compute $\partial\ell/\partial\hat{T}_n$ and $\partial\ell/\partial\Theta_{f,n}$ following standard CNN back propagation algorithm.
  Given $\partial\ell/\partial\hat{T}_n$, compute $[\partial\ell/\partial U_n]_{\hat{T}_n}$ and $[\partial\ell/\partial Q_{n-1}]_{\hat{T}_n}$ using Eqns. (21) (22).
  if $n = N$ **then**
    Let $\partial\ell/\partial U_n = [\partial\ell/\partial U_n]_{\hat{T}_n} + [\partial\ell/\partial U_n]_{Q_n} + \partial\ell/\partial U_N$.
  else
    Let $\partial\ell/\partial U_n = [\partial\ell/\partial U_n]_{\hat{T}_n} + [\partial\ell/\partial U_n]_{Q_n}$.
  end if
  Given $\partial\ell/\partial U_n$, compute $\partial\ell/\partial\hat{U}_n$ and $\partial\ell/\partial\Theta_{g,n}$ following standard CNN back propagation algorithm.
  Given $\partial\ell/\partial\hat{U}_n$, compute $[\partial\ell/\partial T_{n-1}]_{\hat{U}_n}$ and $[\partial\ell/\partial Q_{n-1}]_{\hat{U}_n}$ using Eqns (19) (20).
  Let $\partial\ell/\partial Q_{n-1} = [\partial\ell/\partial Q_{n-1}]_{\hat{T}_n} + [\partial\ell/\partial Q_{n-1}]_{\hat{U}_n}$.
end for
return $\partial\ell/\partial\Theta_{g,n}$ and $\partial\ell/\partial\Theta_{f,n}$, $n = 1, 2, \ldots, N$.

$\partial\ell/\partial T_N$ and $\partial\ell/\partial Q_N$ respectively. To train our proximal dehaze-net using gradient method, we need to compute the following necessary gradients

$$\begin{aligned}
&\text{given } \frac{\partial\ell}{\partial Q_n}, \text{ compute } \left[\frac{\partial\ell}{\partial U_n}\right]_{Q_n} \text{ and } \left[\frac{\partial\ell}{\partial T_n}\right]_{Q_n};\\
&\text{given } \frac{\partial\ell}{\partial\hat{T}_n}, \text{ compute } \left[\frac{\partial\ell}{\partial Q_{n-1}}\right]_{\hat{T}_n} \text{ and } \left[\frac{\partial\ell}{\partial U_n}\right]_{\hat{T}_n};\\
&\text{given } \frac{\partial\ell}{\partial\hat{U}_n}, \text{ compute } \left[\frac{\partial\ell}{\partial Q_{n-1}}\right]_{\hat{U}_n} \text{ and } \left[\frac{\partial\ell}{\partial T_{n-1}}\right]_{\hat{U}_n},
\end{aligned}$$

where we denote $\left[\dfrac{\partial\ell}{\partial X}\right]_Y = \dfrac{\partial}{\partial X}\left(\dfrac{\partial\ell}{\partial Y}\right)$.

Gradients involved in updating U_n.

When computing the gradient of final loss ℓ w.r.t. T_{n-1} involved in Eqn. (13), all the operations are element-wise. Thus for each pixel $\mathrm{x} \in \Omega(\mathrm{x})$, we have

$$\hat{U}_n(\mathrm{x}) = \frac{\beta T_{n-1}(\mathrm{x})(P^{dk}(\mathrm{x}) + T_{n-1}(\mathrm{x}) - 1) + \gamma Q^{dk}_{n-1}(\mathrm{x})}{\beta T^2_{n-1}(\mathrm{x}) + \gamma}.$$

So the partial gradient of $\hat{U}_n(\mathrm{x})$ w.r.t. $T_{n-1}(\mathrm{x})$ is

$$\frac{\partial\hat{U}_n(\mathrm{x})}{\partial T_{n-1}(\mathrm{x})} = \frac{\beta(P^{dk}(\mathrm{x}) + 2T_{n-1}(\mathrm{x}) - 1) - 2\beta T_{n-1}(\mathrm{x})\hat{U}_n(\mathrm{x})}{\beta T^2_{n-1}(\mathrm{x}) + \gamma}.$$

And the gradient of final loss ℓ w.r.t. T_{n-1} in Eqn. (13) can be achieved using chain rule

$$\left[\frac{\partial \ell}{\partial T_{n-1}}\right]_{\hat{U}_n} = \frac{\partial \ell}{\partial \hat{U}_n} \circ \frac{\beta(P^{dk} + 2T_{n-1} - 1) - 2\beta T_{n-1} \circ \hat{U}_n}{\beta T_{n-1} \circ T_{n-1} + \gamma}. \tag{19}$$

Next is to compute the gradient of ℓ w.r.t. Q_{n-1}, to do which we have to rewrite Eqn. (13) in vector form and replace $\overrightarrow{Q_{n-1}^{dk}}$ with $D\overrightarrow{Q_{n-1}}$

$$\overrightarrow{\hat{U}_n} = \frac{\beta\overrightarrow{T_{n-1}} \circ (\overrightarrow{P^{dk}} + \overrightarrow{T_{n-1}} - 1) + \gamma D\overrightarrow{Q_{n-1}}}{\beta\overrightarrow{T_{n-1}} \circ \overrightarrow{T_{n-1}} + \gamma}.$$

Using chain rule, we get the gradient of final loss w.r.t. $\overrightarrow{Q_{n-1}}$

$$\left[\frac{\partial \ell}{\partial \overrightarrow{Q_{n-1}}}\right]_{\overrightarrow{\hat{U}_n}} = \frac{\partial \overrightarrow{\hat{U}_n}^\top}{\partial \overrightarrow{Q_{n-1}}} \frac{\partial \ell}{\partial \overrightarrow{\hat{U}_n}} = D^\top \left(\frac{\partial \ell}{\partial \overrightarrow{\hat{U}_{n-1}}} \circ \frac{\gamma}{(\beta\overrightarrow{T_{n-1}} \circ \overrightarrow{T_{n-1}} + \gamma)} \right). \tag{20}$$

We then reshape $\left[\frac{\partial \ell}{\partial \overrightarrow{Q_{n-1}}}\right]_{\overrightarrow{\hat{U}_n}}$ to matrix size to get $\left[\frac{\partial \ell}{\partial Q_{n-1}}\right]_{\hat{U}_n}$.

Gradients involved in updating T_n.

All operations included in the updating of $\hat{T}$ are element-wise, so the gradients of final loss ℓ w.r.t. Q_{n-1} and U_n involved in Eqn. (15) are directly computed using chain rule in the same way as Eqn. (19).

$$\left[\frac{\partial \ell}{\partial Q_{n-1}^c}\right]_{\hat{T}_n} = \frac{\alpha(P^c - 1) - 2\alpha(Q_{n-1}^c - 1) \circ \hat{T}_n}{\sum_c \alpha(Q_{n-1}^c - 1) \circ (Q_{n-1}^c - 1) + \beta(U_n - 1) \circ (U_n - 1)} \circ \frac{\partial \ell}{\partial \hat{T}_n}, \tag{21}$$

$$\left[\frac{\partial \ell}{\partial U_n}\right]_{\hat{T}_n} = \frac{\beta(P^{dk} - 1) - 2\beta(U_n - 1) \circ \hat{T}_n}{\sum_c \alpha(Q_{n-1}^c - 1) \circ (Q_{n-1}^c - 1) + \beta(U_n - 1) \circ (U_n - 1)} \circ \frac{\partial \ell}{\partial \hat{T}_n}. \tag{22}$$

Gradients involved in updating Q_n.

According to Eqn. (18), the gradient of final loss ℓ with respect to $\overrightarrow{\mathcal{T}_n}$ is

$$\left[\frac{\partial \ell}{\partial \overrightarrow{\mathcal{T}_n}}\right]_{\overrightarrow{Q_n}} = \frac{\alpha(\overrightarrow{P} + 2\overrightarrow{\mathcal{T}_n} - 1) - 2\alpha\overrightarrow{\mathcal{T}_n} \circ \overrightarrow{Q_n}}{\alpha\overrightarrow{\mathcal{T}_n} \circ \overrightarrow{\mathcal{T}_n} + \gamma \mathrm{diag}(D^\top D)} \circ \frac{\partial \ell}{\partial \overrightarrow{Q_n}}. \tag{23}$$

Note that $\mathcal{T}_n$ is concatenated by T_n, so

$$\left[\frac{\partial \ell}{\partial T_n}\right]_{Q_n} = \sum_{c \in \{r,g,b\}} \left[\frac{\partial \ell}{\partial \mathcal{T}_n^c}\right]_{Q_n}. \tag{24}$$

Finally, the gradient of ℓ w.r.t. $\overrightarrow{U_n}$ can be computed as

$$\left[\frac{\partial \ell}{\partial \overrightarrow{U_n}}\right]_{\overrightarrow{Q_n}} = D\left(\frac{\gamma}{\alpha\overrightarrow{\mathcal{T}_n} \circ \overrightarrow{\mathcal{T}_n} + \gamma \mathrm{diag}(D^\top D)} \circ \frac{\partial \ell}{\partial \overrightarrow{Q_n}} \right). \tag{25}$$

Remark 1. For an image P, the linear operator D is to compute its dark channel P^{dk} such that $\overrightarrow{P^{dk}} = D\overrightarrow{P}$. The transposed operator $D^\top$ is to place pixels in the dark channel back to their origin place in the image. In implementation, we realize D and $D^\top$ operator using CUDA programs.

Remark 2. When computing Q_n, we need to compute dark channel Q_n^{dk} which is not causal, so we approximately compute Q_n^{dk} from Q_{n-1}. In addition, the divisions appeared in the algorithm are element-wise.

Remark 3. For D-Net and T-Net in stage n, the network inputs are respectively $\hat{U}_n$ and $\hat{T}_n$ concatenated with the scaled input image P, in order to incorporate image structural information for better updating the dark channel and transmission map.

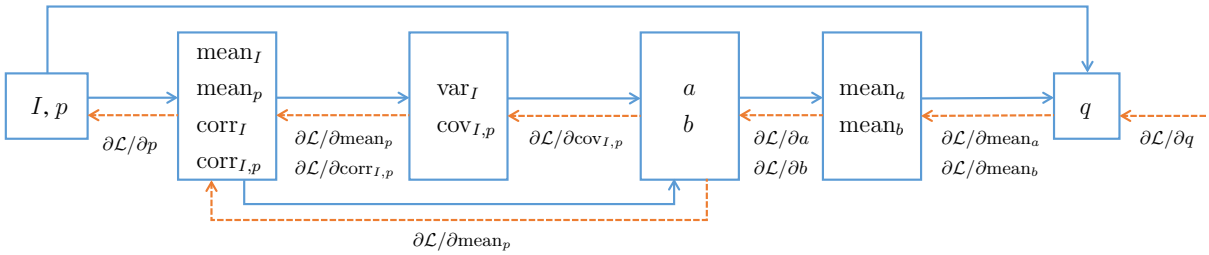


Figure 2: The computation flow graph of GIF-Block.

Algorithm 2 Forward of GIF-Block

Input: filtering input p, guidance image I, mean filter f, regularization ε

Output: filtered image q

$\text{mean}_I = \text{conv}(I, f)$
$\text{mean}_p = \text{conv}(p, f)$
$\text{corr}_I = \text{conv}(I \circ I, f)$
$\text{corr}_{I,p} = \text{conv}(I \circ p, f)$
$\text{var}_I = \text{corr}_I - \text{mean}_I \circ \text{mean}_I$
$\text{cov}_{I,p} = \text{corr}_{I,p} - \text{mean}_I \circ \text{mean}_p$
$a = \text{cov}_{I,p} \oslash (\text{var}_I + \varepsilon)$ // $\oslash$: *element-wise division*
$b = \text{mean}_p - a \circ \text{mean}_I$
$\text{mean}_a = \text{conv}(a, f)$
$\text{mean}_b = \text{conv}(b, f)$
$q = \text{mean}_a \circ I + \text{mean}_b$
return filtered image q.

3.2 Computation graph of GIF-Block

We next show the computation graph of the GIF-Block of our proximal dehaze-net. GIF-Block follows the standard computing process in [2]. For better illustration, we draw a simple data flow graph of guided image filtering, as shown in Fig. 2, in which, blue solid lines represent the forward flow and orange dash lines represent the backward flow. GIF-Block takes image p as input and I as guide image and outputs the filtered image q. In our case, p is the transmission map estimated by T-Net, I is the original hazy image and q is the refined transmission map. In implementation, we realize the filtering operations using convolutions, thus the forward pass of GIF-Block can be formulated as Algorithm 2.

While the forward of GIF-Block is clear, the backward pass is also simple. Since we do not learn filter f, what we need is to compute the gradient of loss $\mathcal{L}$ with respect to input p. Following the principle of back-propagation, we need to compute the intermediate gradients step by step, along the orange dash lines in Fig. 2. We show the back-propagation of GIF-Block in Algorithm 3. In the following, we denote the derivative of convolution by dconv, i.e.

$$\text{dconv}(x, f) = \frac{\partial \text{conv}(x, f)}{\partial x}. \tag{26}$$

According to chain rule and following Fig. 2, given $\partial\mathcal{L}/\partial q$, we have

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \text{mean}_a} &= \frac{\partial \mathcal{L}}{\partial q} \circ I, \\ \frac{\partial \mathcal{L}}{\partial \text{mean}_b} &= \frac{\partial \mathcal{L}}{\partial q}. \end{aligned} \tag{27}$$

Algorithm 3 Backward of GIF-Block

Input: gradient w.r.t. output $\partial\mathcal{L}/\partial q$

Output: gradient w.r.t. input $\partial\mathcal{L}/\partial p$

1: Given $\partial\mathcal{L}/\partial q$, compute $\partial\mathcal{L}/\partial\mathrm{mean}_a$ and $\partial\mathcal{L}/\partial\mathrm{mean}_b$
2: Given $\partial\mathcal{L}/\partial\mathrm{mean}_a$ and $\partial\mathcal{L}/\partial\mathrm{mean}_b$, compute $\partial\mathcal{L}/\partial a$ and $\partial\mathcal{L}/\partial b$
3: Given $\partial\mathcal{L}/\partial a$ and $\partial\mathcal{L}/\partial b$, compute $\partial\mathcal{L}/\partial\mathrm{cov}_{I,p}$ and $[\partial\mathcal{L}/\mathrm{mean}_p]_1$
4: Given $\partial\mathcal{L}/\partial\mathrm{cov}_{I,p}$, compute $[\partial\mathcal{L}/\partial\mathrm{mean}_p]_2$ and $\partial\mathcal{L}/\partial\mathrm{corr}_{I,p}$
5: $\partial\mathcal{L}/\partial\mathrm{mean}_p = [\partial\mathcal{L}/\mathrm{mean}_p]_1 + [\partial\mathcal{L}/\partial\mathrm{mean}_p]_2$
6: Given $\partial\mathcal{L}/\partial\mathrm{mean}_p$ and $\partial\mathcal{L}/\partial\mathrm{corr}_{I,p}$, compute $\partial\mathcal{L}/\partial p$
7: **return** gradient $\partial\mathcal{L}/\partial p$.

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial b} &= \frac{\partial\mathcal{L}}{\partial\mathrm{mean}_b}\circ\mathrm{dconv}(b,f),\\ \frac{\partial\mathcal{L}}{\partial a} &= \frac{\partial\mathcal{L}}{\partial\mathrm{mean}_a}\circ\mathrm{dconv}(a,f) - \frac{\partial\mathcal{L}}{\partial b}\circ\mathrm{mean}_I.\end{aligned} \tag{28}$$

$$\begin{aligned}\left[\frac{\partial\mathcal{L}}{\partial\mathrm{mean}_p}\right]_1 &= \frac{\partial\mathcal{L}}{\partial b},\\ \frac{\partial\mathcal{L}}{\partial\mathrm{cov}_{I,p}} &= \frac{\partial\mathcal{L}}{\partial a}\oslash(\mathrm{var}_\mathrm{I}+\varepsilon).\end{aligned} \tag{29}$$

$$\begin{aligned}\left[\frac{\partial\mathcal{L}}{\partial\mathrm{mean}_p}\right]_2 &= -\frac{\partial\mathcal{L}}{\partial\mathrm{cov}_{I,p}}\circ\mathrm{mean}_I,\\ \frac{\partial\mathcal{L}}{\partial\mathrm{corr}_{I,p}} &= \frac{\partial\mathcal{L}}{\partial\mathrm{cov}_{I,p}}.\end{aligned} \tag{30}$$

$$\frac{\partial\mathcal{L}}{\partial\mathrm{mean}_p} = \left[\frac{\partial\mathcal{L}}{\partial\mathrm{mean}_p}\right]_1 + \left[\frac{\partial\mathcal{L}}{\partial\mathrm{mean}_p}\right]_2. \tag{31}$$

$$\frac{\partial\mathcal{L}}{\partial p} = \frac{\partial\mathcal{L}}{\partial\mathrm{mean}_p}\circ\mathrm{dconv}(p,f) + \frac{\partial\mathcal{L}}{\partial\mathrm{corr}_{I,p}}\circ\mathrm{dconv}(I\circ p,f)\circ I. \tag{32}$$

References

[1] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[2] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis & machine intelligence*, (6):1397–1409, 2013.

[3] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.