Derivative of Structural Similarity Index*

Hazique Aetesam† and Suman Kumar Maji†

In this tutorial, we study about finding the derivative of structural similarity (SSIM) for it to be used as data fidelity term in optimization problems. Higher values of SSIM denote better perceptual quality of the image. Therefore, unlike mean square error (ℓ_2-norm) or mean absolute deviation (ℓ_1-norm) which are to be minimized for any gradient based technique, SSIM needs to be maximized. In general, for any additive noise η corrupting data along with linear degradation operator H, image formation model can be represented as:

$$y = Hx + \eta \tag{1}$$

For denoising problems (as in the present tutorial), we consider $H = I$. Under variational approach, we can consider the following optimization problem:

$$\hat{x} = \arg\min_x (1 - SSIM(x, y)) + \alpha R(x) \tag{2}$$

Here, x is the reference groundtruth and y is the corrupted image whose SSIM needs to be calculated. Since SSIM is a symmetric measure, SSIM(x, y) = SSIM(y, x). For benchmarking the restoration results of different techniques, y can also be considered as the restored image. SSIM is composed of three different components; namely luminosity, contrast and structural information.

$$\text{SSIM } (x, y) = l(x, y)^{\alpha} \cdot c(x, y)^{\beta} \cdot s(x, y)^{\gamma} \tag{3}$$

Here, the values of α, β and γ as chosen to be 1. Individual terms are calculated as:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{4a}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{4b}$$

$$s(x, y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{4c}$$

Here, $C_3 = C_2/2$ and μ_x and μ_y are means of x and y respectively. σ_x and σ_y are standard deviations and σ_{xy} is the covariance between x and y such that:

$$\mu_x = \frac{1}{N}\sum_i x_i; \quad \sigma_x = \frac{1}{N-1}\sum_i \left[(x_i - \mu_x)^2\right]^{1/2}$$

*This supplementary text is in support of our paper titled "Perceptually Motivated Generative Model for Magnetic Resonance Image Denoising" accepted in *The Journal of Digital Imaging (JDI)*, an official peer-reviewed journal of the Society for Imaging Informatics in Medicine (SIIM), Springer.

†H. Aetesam and S. K. Maji are associated with the Department of Computer Science and Engineering, Indian Institute of Technology-Patna, Bihta, 801106 India e-mail: (hazique.pcs16@iitp.ac.in (Hazique Aetesam).

$$\sigma_{xy} = \left[\frac{1}{N-1} \sum_i (x_i - \mu_x)(y_i - \mu_y) \right]$$

μ_y and σ_y can defined similarly. Combining all the terms together, we obtain:

$$\text{SSIM}\ (x, y) = \underbrace{\left(\frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \right)}_{l(p)} \cdot \underbrace{\left(\frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right)}_{cs(p)} \tag{5}$$

However, it is found that the values of the component parameters ($\mu_x,\ \mu_y,\ \sigma_x,\ \sigma_y,\ \sigma_{xy}$) are spatially variant in different sections of the image and hence should not be calculated collectively over entire image. Therefore, local window SSIM can be calculated as:

$$\text{SSIM}(x(p), y(p)) = l(p) \cdot cs(p) \tag{6}$$

where $x(p)$ is the local window centred at pixel p. For that, we calculate local features using Gaussian filter G_σ with a fixed window size and average those values to obtain the final SSIM value. This SSIM is called mean SSIM (MSSIM).

$$\text{MSSIM}\ (x, y) = \frac{1}{M} \sum_j\ \text{SSIM}\ (x_i, y_j) \tag{7}$$

where M is the total number of patches. Using these values in Eq. (2), we get:

$$\arg\min_x (1 - MSSIM(x, y)) + \alpha R(x) \tag{8}$$

Further, when we consider Gaussian filter for calculating parameters, we obtain:

$$\mu_x(p) = G_\sigma \star P_x \quad \sigma_x = G_\sigma \star P_x^2 - \mu_x^2(p) \quad \sigma_{xy} = G_\sigma \star (P_x \cdot P_y) - \mu_x(p)\mu_y(p)$$

P_x is the local image window centred at pixel p and $\star$ denoted convolution operator. Convolution between two different functions are defined as:

$$g(x) = (f \star w)(x) = \sum_i f(n) w(x - n) \tag{9}$$

Now, using Eq. (6) to find out the derivative with respect to pixel q, we obtain:

$$\frac{\partial\ \text{SSIM}(x(p), y(p)}{\partial x(q)} = cs(p) \frac{\partial}{\partial x(q)} l(p) + l(p) \frac{\partial}{\partial x(q)} cs(p) \tag{10}$$

Using chain rule, we can write Eq. (10) as:

$$\frac{\partial l(p)}{\partial x(q)} = \frac{\partial l(p)}{\partial \mu_x(p)} \cdot \frac{\partial \mu_x(p)}{\partial x(q)} \tag{11a}$$

$$\frac{\partial l(p)}{\partial \mu_x(p)} = \frac{2[\mu_y(\mu_x^2 + \mu_y^2 + C_1) - \mu_x(2\mu_x\mu_y + C_1)]}{(\mu_x^2 + \mu_y^2 + C_1)^2} \tag{11b}$$

$$= 2\left[\frac{\mu_y}{(\mu_x^2 + \mu_y^2 + C_1)} - \mu_x \underbrace{\frac{2\mu_x\mu_y + C_1}{(\mu_x^2 + \mu_y^2 + C_1)}}_{l(p)} \cdot \frac{1}{(\mu_x^2 + \mu_y^2 + C_1)}\right] \tag{11c}$$

$$= 2\left[\frac{\mu_y - \mu_x l(p)}{\mu_x^2 + \mu_y^2 + C_1}\right] \tag{11d}$$

$$\frac{\partial u_x}{\partial x(q)} = G_\sigma(p - q) \tag{11e}$$

Similarly, using chain rule, we can find the derivative of the second expression in Eq. (10).

$$\frac{\partial cs(p)}{\partial x(q)} = \frac{\partial cs(p)}{\mu_x(p)} \cdot \frac{\partial \mu_x(p)}{\partial x(q)} \tag{12a}$$

$$\frac{\partial cs(p)}{\partial \mu_x(p)} = \frac{\partial}{\partial \mu_x(p)}\left[\frac{2(x - \mu_x)(y - \mu_y) + C_2}{(\sigma_x^2 + \sigma_y^2 + C_2)}\right] \tag{12b}$$

$$= \frac{-2(y - \mu_y)}{(\sigma_x^2 + \sigma_y^2 + C_2)} + 2\left[\underbrace{\frac{(2\sigma_{xy} + C_2)}{(\sigma_x^2 + \sigma_y^2 + C_2)}}_{cs(p)} \times \frac{(x - \mu_x)}{\sigma_x^2 + \sigma_y^2 + C2}\right] \tag{12c}$$

$$= \frac{2}{(\sigma_x^2 + \sigma_y^2 + C_2)}\left[cs(p)(x - \mu_x) - (y - \mu_y)\right] \tag{12d}$$

$$\frac{\partial \mu_x}{\partial x(q)} = G_\sigma(p - q) \tag{12e}$$

Total MSSIM measure is given by:

$$\frac{1}{N}\sum_p cs(p) \cdot \frac{\partial l(p)}{\partial x(q)} + l(p) \cdot \frac{\partial cs(p)}{\partial x(q)} \tag{13}$$

$$\frac{1}{N}\sum_p cs(p)\frac{\partial l(p)}{\partial x(q)} = (2G_\sigma(p - q))\left[\frac{\mu_y - \mu_x l(p)}{\mu_x^2 + \mu_y^2 + C_1}\right] \cdot cs(p) \tag{14a}$$

Re-arranging the above expression, we can write:

$$\frac{2}{N}\sum_p cs(p)\left[\frac{\mu_y - \mu_x l(p)}{\mu_x^2 + \mu_y^2 + C_1}\right] G_\sigma(p - q) \tag{14b}$$

From the definition of convolution in Eq. (9), we obtain:

$$\frac{2}{N}\sum_p h(p) \cdot G_\sigma(p - q) \Rightarrow \frac{2}{N}(h \star G_\sigma)(q) \tag{14c}$$

where $h = cs(p) \cdot \dfrac{\mu_y - \mu_x l(p)}{\mu_x^2 + \mu_y^2 + C_1}$ Similarly, bringing the second expression $\dfrac{1}{N}\sum_p l(p) \cdot \dfrac{\partial cs(p)}{\partial x(q)}$ into picture, we obtain:

$$\frac{2}{N}\sum_p l(p)\left[\frac{G_\sigma(p-q)}{\sigma_x^2 + \sigma_y^2 + C_2}\right][y(q) - \mu_y - cs(p)\cdot x(q) + cs(p)\cdot \mu_x] \tag{15}$$

Eq. (15) can be split into four terms as:

$$\frac{2}{N}(Q + R + S + T) \tag{16}$$

$$Q = y(q)\sum_p \underbrace{\frac{l(p)}{\sigma_x^2 + \sigma_y^2 + C_2}}_{f_1(p)} \cdot G_\sigma(p-q) \Rightarrow y(q)\cdot(f_1 \star G_\sigma)(q) \tag{17a}$$

$$R = -\sum_p \underbrace{\frac{l(p)\cdot \mu_y(p)}{\sigma_x^2 + \sigma_y^2 + C_2}}_{f_2(p)} \cdot G_\sigma(p-q) \Rightarrow -(f_2 \star G_\sigma)(q) \tag{17b}$$

$$S = -x(q)\sum_p \underbrace{\frac{l(p)\cdot cs(p)}{\sigma_x^2 + \sigma_y^2 + C_2}}_{f_3(p)} \cdot G_\sigma(p-q) \Rightarrow -x(q)\cdot(f_3 \star G_\sigma)(q) \tag{17c}$$

$$T = \sum_p \underbrace{\frac{l(p)\cdot cs(p)\cdot \mu_x(p)}{\sigma_x^2 + \sigma_y^2 + C_2}}_{f_4(p)} \cdot G_\sigma(p-q) \Rightarrow \cdot(f_4 \star G_\sigma)(q) \tag{17d}$$

Hence, the final expression for obtaining MSSIM is:

$$\frac{\partial \text{ MSSIM } (x,y)}{\partial x(q)} = \frac{2}{N}\Bigg[(h \star G_\sigma) + y(q)\cdot(f_1 \star G_\sigma)(q) - (f_2 \star G_\sigma)(q) \tag{18}$$

$$-x(q)\cdot(f_3 \star G_\sigma)(q) + (f_4 \star G_\sigma)(q)\Bigg] \tag{19}$$

It can be concluded that finding MSSIM derivative using convolutions (Eq. (18)) offer better alternative than using primitive operations (Eq. (11) and Eq. (12) calculated with respect to each pixel q centred at p in the filtering window). Since an efficient technique for finding convolution is Fast Fourier Transform (FFT), we can obtain runtime of $O(N \log N)$. However, derivative using equations (11) and (12) performs the same operations in $O(mN \log N)$ time; where m is the patch size. Difference in the derivative computation performance using these two techniques is exacerbated especially in deep learning based restoration methods where training is already overwhelmed with the huge amount of data. Therefore, we can expect performance benefits by order of m by using the modified strategy.