Another Look at SSIM Image Quality Metric

Yuriy Reznik
Brightcove Inc.
Boston, USA
yreznik@brightcove.com

Abstract—We review the design of the SSIM quality metric and offer an alternative model of SSIM computation, utilizing subband decomposition and identical distance measures in each subband. We show that this model performs very close to the original and offers many advantages from a methodological standpoint. It immediately brings several possible explanations of why SSIM is effective. It also suggests a simple strategy for band noise allocation optimizing SSIM scores. This strategy may aid the design of encoders or pre-processing filters for video coding. Finally, this model leads to more straightforward mathematical connections between SSIM, MSE, and SNR metrics, improving previously known results.

Keywords—Image quality, PSNR, SSIM, SNR

I. INTRODUCTION

The Structural Similarity (SSIM) metric [1,2] has been around for nearly two decades and has become one of the most established and frequently used metrics for image and video quality analysis. Many research papers have followed, extending SSIM in multiscale, 3D, and temporal dimensions [3-6], analyzing its mathematical properties [7-9], and discussing its applications for improving the design of encoding algorithms [10-13] and streaming systems [14-16].

Yet, despite all the progress and broad acceptance in practice, what specifically SSIM means from a physical, mathematical, and signal processing standpoint is not completely understood. The original papers [1,2] explain SSIM as a generalized mean of 3-types of distortion criteria: changes in luminance, contrast, and structure. However, in the final formula in [1,2], some factors cancel out, producing a somewhat mysterious fraction

$$\frac{2\sigma_{xy}}{\sigma_x^2 + \sigma_y^2}$$

which no longer represents a proper measure of correlation (structure) or change in contrast.

Adding to the mystery, several suggestions have been made over the years that there must be some simple relation between SSIM and other distortion metrics, such as PSNR or MSE [7,8]. For instance, in 2010, A. Horé and D. Ziou [8] suggested that:

$$1 - \frac{\text{MSE}}{\text{SSIM}} \sim \frac{2\sigma_{xy}}{\sigma_x^2 + \sigma_y^2}$$

where MSE is a mean square error and $\sigma_{xy}$ is a covariance between image patches $x$ and $y$. However, since $\sigma_{xy}$ depends on both images, this formula does not reduce SSIM (as a distance metric between $x$ and $y$) to MSE!

In this paper, we will look at SSIM again and offer an alternative model for its computation, leading to a more straightforward interpretation of the SSIM metric and its connection to MSE, SNR, and some other metrics. In Section II, we will bring definitions, introduce our main results, and discuss their consequences. In Section III, we will perform experimental validation of our proposed model for computing SSIM.

II. MAIN RESULTS AND THEIR CONSEQUENCES

A. SSIM definition

Recall that at a patch level, the SSIM between images $x$ and $y$ is defined as follows [1]:

$$\text{SSIM}(x, y) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \cdot \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

(1)

where $\mu_x, \mu_y, \sigma_x, \sigma_y, \sigma_{xy}$ represent patch-level statistics:

$$\mu_x = E[x] = \sum_{k=1}^{N} w_k x_k,$$

$$\sigma_x^2 = E[(x - \mu_x)^2] = \sum_{k=1}^{N} w_k (x_k - \mu_x)^2,$$

$$\sigma_{xy} = E[(x - \mu_x)(y - \mu_y)] = \sum_{k=1}^{N} w_k (x_k - \mu_x)(y_k - \mu_y),$$

(2-4)

and $C_1, C_2$ are some small constants.

In all these formulae, $E[.]$ denotes expectation operators with some density $w$ superimposed over the patch. The original SSIM implementation [1] uses 11x11-pixels patches and circular-symmetric Gaussian density with a standard deviation $\sigma = 1.5$. The weights $w_k$ are normalized: $\sum w_i = 1$.

The constants $C_1, C_2$ in the original SSIM design [1] are set to $C_1 = (0.01 \cdot I_{\text{max}})^2$, and $C_2 = (0.03 \cdot I_{\text{max}})^2$, where $I_{\text{max}}$ is the maximum pixel value (e.g., 255 for 8-bit pixels). As explained in [1], the purpose of these constants is to avoid numerical instabilities in cases when signals approach 0.

Considering the whole images, the mean SSIM is computed as the average of patch-level SSIMs at each pixel location $(i,j)$:

$$\overline{\text{SSIM}}(x, y) = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} \text{SSIM}(x(i,j), y(i,j)).$$

(5)

$W$ and $H$ denote image width and height, respectively.

B. Proposed Alternative Form

By looking at formula (1), we first notice that SSIM is essentially a product of two nearly identical functions:

$$\xi(x, y) = \frac{2E[xy] + C}{E[x^2] + E[y^2] + C}$$

(6)
applied to different signals and at different scales. In the first term in the SSIM expression (1), this operation is applied to DC values \( \mu_x, \mu_y \), treated as scalars \( N = 1 \). In the second term in (1), it is applied to residual signals \( x - \mu_x \), and \( y - \mu_y \), observed in \( 11 \times 11 \) patches \( (N = 11^2) \).

However, if we examine the derivation of DC values \( \mu_x, \mu_y \) in (1,2), we notice that they can also be understood as pixel values taken from some low-pass-filtered versions of input images \( x_L, y_L \). Similarly, we also realize that the residual signals in each path \( x - \mu_x \), and \( y - \mu_y \) must be similar to signals taken from high-pass versions of the same images: \( x_H = x - x_L \), \( y_H = y - y_L \). And finally, we also notice that since DC values are coming from low-pass filtered images \( x_L, y_L \), the relative distance between them as scalars \( \xi(\mu_x, \mu_y) \) or computed over surrounding patches \( \xi(x_L, y_L) \) must be very small. The low-pass removes local variations making the patch-average results almost identical.

In other words, by combining all these observations, we can conjecture that patch-level SSIM can be computed in a fully symmetric manner, as follows

\[
SSIM(x, y) \approx \xi(x_L, y_L) \cdot \xi(x_H, y_H)
\]

where \( x_L, y_L \) are the patches in low-pass filtered images, \( x_H = x - x_L \), \( y_H = y - y_L \) are the patches in high-pass filtered images, and where \( \xi(\ldots) \) are identical distance functions (6) computed over patches in low-pass and high-pass images.

Figure 1 shows the flow-diagram explaining computations according to (7). In Section III, we present an experimental study, indicating that this process yields very similar results to the original SSIM formula. Our experiments further show that to split \( x, y \) into low-frequency and high-frequency components, it is sufficient to apply Gaussian filter with pixel-level standard deviation \( \sigma = 3 \). This value is 2x larger than the standard deviation used in windows for computing patch-level distances \( \xi(x_L, y_L), \xi(x_H, y_H) \).

C. Illustration of operation

One of the immediate benefits of the proposed model (7) is a simple signal processing interpretation of how SSIM works, and why it is better than full-band metrics, such as MSE or PSNR. We illustrate this in Figure 2.

As input \( x \) in this example, we use image k04 from the Kodak data set [17]. The low-pass (\( x_L \)) and high-pass (\( x_H \)) versions of this image are shown in the top row in Figure 2. The input \( y \) in this example is a reconstruction of the same image after it was compressed with H.264 encoder [18] with QP=47. This image is highly distorted relative to the original. The low-pass \( (y_L) \) and high-pass \( (y_H) \) versions of this image are shown in the middle row. Finally, the last row shows the differences between these images.

Looking at the last row, we immediately notice that the direct difference between input images \( x - y \) looks much "busier" compared to the difference images in low- and high-frequency domains. We see that magnitudes of errors in each subband domain become lower (particularly in the low-band), and their impacts become more obvious visually and conceptually. The low-pass filtering captures image changes in overall shapes, while high-pass shows differences in fine-grain details – contours, textures, etc.

In other words, we can see that the principal difference between SSIM and PSNR, MSE, and other simple metrics is that SSIM analyzes images in two subbands. Subband processing allows the separation of errors such that their impacts can be more accurately measured and incorporated in the final score.

D. Connection to CSF

Let us next look at the split between bands used in our model for computing SSIM. The pixel-level standard deviation of the Gaussian low-pass filter used to produce \( x_L \) and \( y_L \) is \( \sigma = 3 \). Consequently, the cut-off frequency of this filter is \( f_c = \frac{F_s}{2\pi \sigma} \approx \frac{F_s}{18.85} \), where \( F_s \) is the sampling rate. By assuming that display Nyquist frequency approximately matches the human visual acuity limit (which is typically the case for ITU-R BT-500 [20] visual tests), this implies that \( F_s \) must be about 80 cpd (cycles per degree), and therefore \( f_c \approx 4.244 \) cpd.

As shown in Figure 3, this frequency approximately matches the peak of the Contrast Sensitivity Function (CSF) of human vision. This figure uses the Barten CSF model [19] and frequency responses of Gaussian low-pass and high-pass filters employed in our model for computing SSIM. Barten CSF model rendered for an observation angle \( X_0 = 30^\circ \) and object luminance \( L = 200 \text{ cd/m}^2 \).
Fig. 2. Visual illustration of steps in the proposed model for computation of SSIM. The top row shows an input image $x$ and its low-pass $x_L$ and high-pass $x_H = x - x_L$ versions. The middle row shows the second input image $y$, and its low-pass $y_L$ and high-pass $y_H = y - y_L$ versions. This image is an encoded and decoded version of $x$, with significant distortions introduced by the codec. The bottom row shows the differences between the original images, as well as their filtered versions. It can be observed that full difference is much “busier” and less descriptive than low- and high-pass differences. Low-pass emphasizes changes in overall shapes, while high-pass emphasizes changes in textures, countours, and other fine features.
This observation brings another argument suggesting why SSIM may be effective. It does a proper signal separation considering the involved mechanisms of human vision. Recall that the decays in contrast sensitivity in low- and high-frequency bands are caused by significantly different phenomena. In the high-band, sensitivity decay is mainly caused by the eye optical MTF [22]. In the low-band, it is mainly the result of lateral inhibition [22,23]. The latter is a non-linear effect. The processing of low-band and high-band signals in the visual cortex is also different, as they turn into different sets of spatial frequency channels [23].

Even though in their motivations in [1], the authors of SSIM intended to design a metric that is not explicitly driven by the CSF and related phenomena of human vision, the use of patches and particular filters in the computation of SSIM, as we have shown above, leads back to this connection!

E. Impacts of differences in low- and high-frequency bands

By denoting each factor in SSIM expression (7) as \( \xi_L = \xi(x_L,y_L) \) and \( \xi_H = \xi(x_H,y_H) \), we can rewrite it as:

\[ \text{SSIM}(\xi_L,\xi_H) = \xi_L \cdot \xi_H \]

By using inequality between geometric and \(-\infty\) means, and also noting that \( \xi_L,\xi_H \leq 1 \), we can show that:

\[ \min(\xi_L,\xi_H) \leq \text{SSIM}(\xi_L,\xi_H) \leq \min(\xi_L,\xi_H) \]  

(8)

This inequality means that it is the smallest between subband differences \( \xi_L \) and \( \xi_H \) that has a limiting impact on the overall SSIM score!

Consequently, this also means, is that if one tries to balance codec-introduced errors to maximize the overall SSIM value, then the best such balance would be achieved when the magnitudes of such errors in both bands are the same:

\[ \max_{\delta: \xi_L=\xi_0+\delta, \xi_H=\xi_0-\delta} \text{SSIM}(\xi_L,\xi_H) \Rightarrow \xi_L = \xi_H \]  

(9)

This observation offers a simple principle for encoder optimizations improving SSIM scores. This idea may also lead to the design of a pre-processing filter that removes some of the low-frequency features and thus allows the encoder to encode the remaining high-frequency content with fewer errors. With proper tuning, such a filter may allow existing encoders to achieve improved SSIM scores.

F. The relation between SSIM and other objective metrics

Finally, let us now take a closer look at quantities \( \xi(x,y) \) in formula (7). For conceptual simplicity, we will discard constant C and will look instead at pure ratios:

\[ \xi(x,y) = \frac{2E[xy]}{E[x^2] + E[y^2]} \]  

(10)

We first note that ratio \( \xi(x,y) \) is not a proper measure of correlation. To measure correlation, one has to apply normalization by the geometric mean of \( E[x^2] \) and \( E[y^2] \):

\[ \rho(x,y) = \frac{E[xy]}{\sqrt{E[x^2]E[y^2]}} \]  

(11)

But these quantities are related. By inequality between geometric and arithmetic means, we can see that:

\[ \xi(x,y) \leq \rho(x,y) \]  

(12)

Next, by looking at the reciprocal of (9), we observe that:

\[ \frac{1}{\xi(x,y)} = \frac{E[x^2] + E[(x+(y-x))^2]}{2E[xy]} = 1 + \frac{E[(y-x)^2]}{2E[xy]} \]

or, equivalently:

\[ \frac{1 - \xi(x,y)}{\xi(x,y)} = \frac{\text{MSE}(x,y)}{2\text{E}[xy]} \]  

(13)

where \( \text{MSE}(x,y) = \text{E}[(x-y)^2] \) is the Mean Square Error.

The obtained formula (11) is similar to the result of A. Horé and D. Ziou [8], with an essential distinction that (13) is defined for subband signals. In their derivations, Horé and Ziou have assumed that DC differences are negligible (or equivalently, that \( \xi_L = 1 \)), but this is not always the case in practice. Equation (13) establishes a more general and accurate relation.

However, as we already noted, formula (13) does not reduce \( \xi(x,y) \) to \( \text{MSE}(x,y) \), since \( \text{E}[xy] \) is a joint statistic of both signals. To produce a more direct relation, we rewrite (10) as:

\[ \xi(x,y) = 1 - \frac{\text{E}[(y-x)^2]}{\text{E}[x^2] + \text{E}[y^2]} \]

and then:

\[ \frac{1}{1 - \xi(x,y)} = \frac{\text{SNR}(x,y) + \text{SNR}(y,x)}{\text{SNR}(x,y)} \]  

(14)

where \( \text{SNR}(x,y) = \frac{\text{E}[x^2]}{\text{E}[(x-y)^2]} \) is the Signal to Noise Ratio.

Combining (14) and (7), we produce:

\[ \text{SSIM}(x,y) \sim \left( 1 - \frac{1}{\text{SNR}(x_L,y_L)+\text{SNR}(y_L,x_L)} \right) \cdot \left( 1 - \frac{1}{\text{SNR}(x_H,y_H)+\text{SNR}(y_H,x_H)} \right) \]  

(15)

where \( \text{SNR}(x_L,y_L), \text{SNR}(y_L,x_L), \text{SNR}(x_H,y_H), \text{SNR}(y_H,x_H) \) are signal-to-noise ratios computed for patches in low-pass and high-pass-filtered images respectively.

The formula (15) shows that SSIM can be computed using a combination of SNR metrics between low- and high-pass-filtered images. These derivations confirm that connections between SSIM and other objective metrics exist, but they are not as simple as suggested earlier [7,8].

III. EXPERIMENTAL VALIDATION

In this section, we describe tests performed to validate the accuracy of the proposed method for the computation of SSIM.

For our experiments, we used 24 standard images from Kodak dataset [17]. We have converted them to BT.709 YUV files and then produced encoded and decoded versions using
H.264 video encoder [18], operating in Main Profile, "slow" encoding preset, and using fixed-QP rate control mode.

Table 1 shows the results. The columns "SSIM" list reference SSIM values computed as described in [1] for each file and QP setting, and the column "Delta" lists differences between the reference SSIM and the one calculated by our method. For band separation, we used a circular-symmetric Gaussian filter with \( \sigma = 3 \). The last row in Table 1 lists RMS average values across all files.

It can be observed that the differences between SSIM produced by using our model and reference are very small, despite the broad range of variation of SSIM values, distortion, and content types. Table 1 provides results for images at 1536x1024 resolutions. The results for other resolutions are very similar and omitted due to paper size restrictions.

**REFERENCES**


