# 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 to improve SSIM scores. This strategy may aid the design of encoders or pre-processing filters for video coding. Finally, this model leads to more direct 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 SSIM means from a physical, mathematical, and signal processing standpoint is not entirely 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which no longer represents a proper measure of correlation (structure) or change in contrast.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:

$$\frac{1 - SSIM}{SSIM} \sim \frac{MSE}{2\sigma_{xy}},$$

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. In Section IV, we will drive conclusions.

## 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]:

$$SSIM(x,y) = \frac{2\mu_x\mu_y + C1}{\mu_x^2 + \mu_y^2 + C1} \cdot \frac{2\sigma_{xy} + C2}{\sigma_x^2 + \sigma_y^2 + C2} \quad (1)$$

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

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

$$\sigma_x^2 = \boldsymbol{E}[(x-\mu_x)^2] = \sum_{k=1}^{N} w_k (x_k - \mu_x)^2, \quad (3)$$

$$\sigma_{xy} = \boldsymbol{E}\left[(x - \mu_x)(y - \mu_y)\right] = \sum_{k=1}^{N} w_k (x_k - \mu_x)(y_k - \mu_y), \quad (4)$$

and $C_1, C_2$ are some small constants.

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

The constants $C_1, C_2$ in the original SSIM design [1] are set to $C_1 = (0.01 \cdot I_{\max})^2$, and $C_2 = (0.03 \cdot I_{\max})^2$, where $I_{max}$ 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 when signals approach 0.

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

$$\overline{SSIM}(x,y) = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} SSIM(x(i,j), y(i,j)). \quad (5)$$

$W$ 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:

$$\xi(x,y) = \frac{2\boldsymbol{E}[xy] + C}{\boldsymbol{E}[x^2] + \boldsymbol{E}[y^2] + C} \quad (6)$$

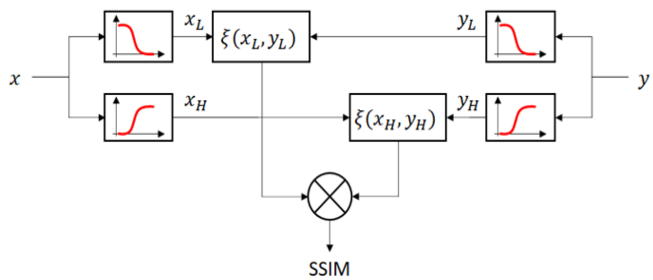


Fig. 1. Computation of SSIM by using formulae (7). SSIM is presented as a product of two identical distance measures $\xi(.,.)$ computed for low-pass and high-pass filtered versions of input signals $x, y$.

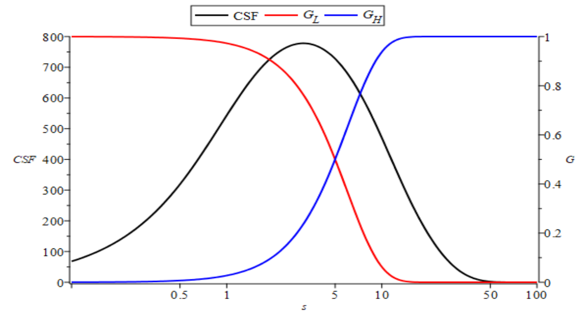


Fig. 3. Superimposed plots of Contrast Sensitivity Function CSF (black) and low- and high-pass filter responses (red and blue) in our model of computation of SSIM. Display Nyquist is assumed to be 40 cpd.

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), this operation is applied to residual signals $x - \mu_x$, and $y - \mu_y$, observed in 11x11 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 $x_L$, $y_L$, then 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

$$SSIM(x, y) \approx \xi(x_L, y_L) \cdot \xi(x_H, y_H) \quad (7)$$

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 $\xi(.,.)$ 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 a 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 the 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 $X_0 = 30°$ and object luminance $L = 200$ cd/m$^2$.

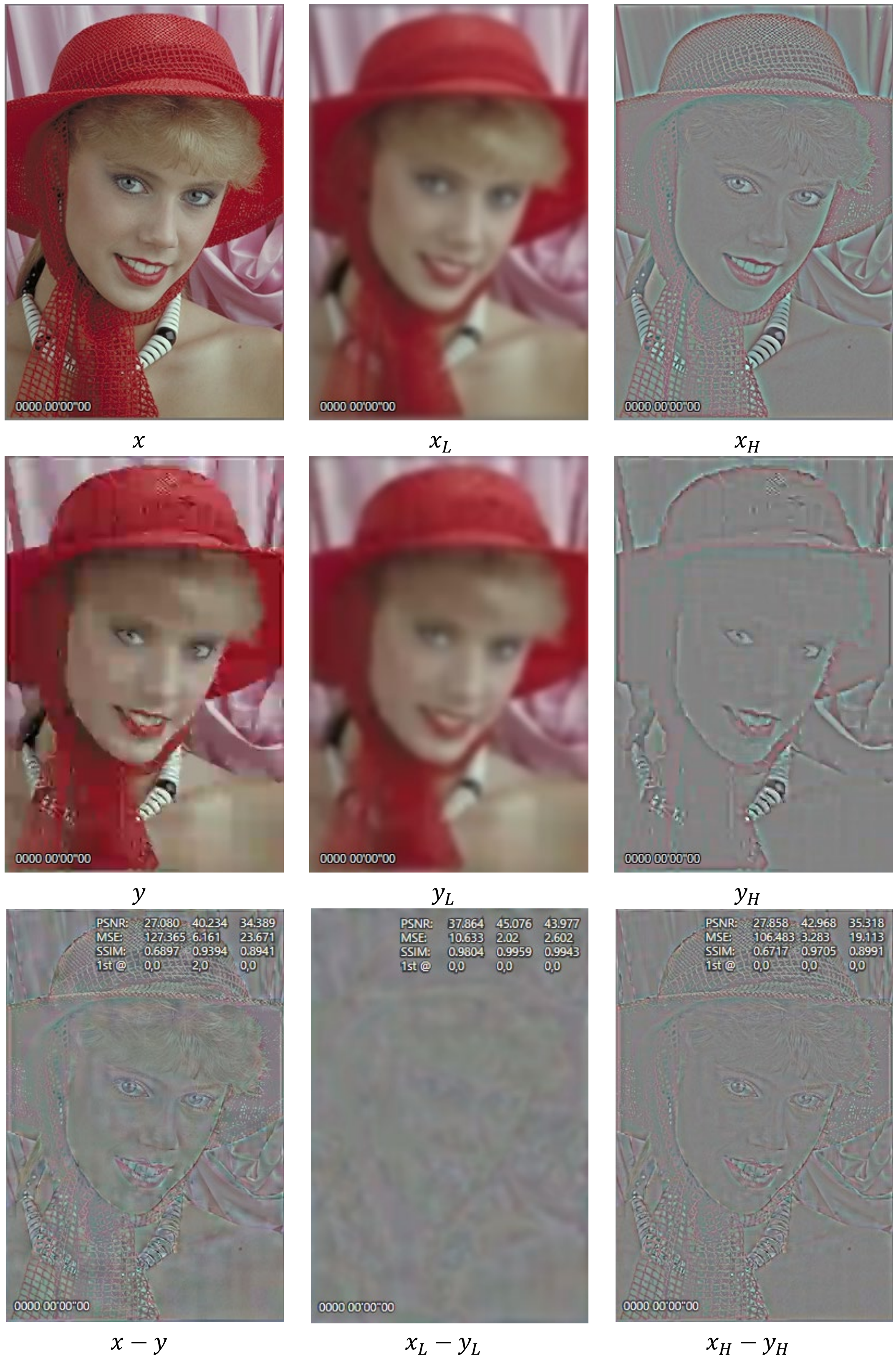


Fig. 2. Visual illustration of steps in the proposed model for computation of SSIM. The top row shows an input image $x$ and its low-pass $x_L$ and high-pass $x_H = x - x_L$ versions. The middle row shows the second input image $y$, and its low-pass $y_L$ and high-pass $y_H = y - y_L$ versions. This image is an encoded and decoded version of $x$, with significant distortions introduced by the codec. The bottom row shows the differences beween 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 decay 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 primarily 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:

$$SSIM(\xi_L, \xi_H) = \xi_L \cdot \xi_H,$$

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

$$\min(\xi_L, \xi_H)^2 \le SSIM(\xi_L, \xi_H) \le \min(\xi_L, \xi_H) \qquad (8)$$

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:

$$\max_{\delta:\, \xi_L=\xi_0+\delta,\, \xi_H=\xi_0-\delta} SSIM(\xi_L, \xi_H) \;\Rightarrow\; \xi_L = \xi_H. \qquad (9)$$

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 enable 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:

$$\xi(x,y) = \frac{2\boldsymbol{E}[xy]}{\boldsymbol{E}[x^2] + \boldsymbol{E}[y^2]} \qquad (10)$$

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 $\boldsymbol{E}[x^2]$ and $\boldsymbol{E}[y^2]$:

$$\rho(x,y) = \frac{\boldsymbol{E}[xy]}{\sqrt{\boldsymbol{E}[x^2]\,\boldsymbol{E}[y^2]}}. \qquad (11)$$

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

$$\xi(x,y) \le \rho(x,y). \qquad (12)$$

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

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

or, equivalently:

$$\frac{1-\xi(x,y)}{\xi(x,y)} = \frac{MSE(x,y)}{2\boldsymbol{E}[xy]} \qquad (13)$$

where $MSE(x,y) = \boldsymbol{E}[(x-y)^2]$ 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 $MSE(x,y)$, since $\boldsymbol{E}[xy]$ is a joint statistic of both signals. To produce a more direct relation, we rewrite (10) as:

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

and then:

$$\frac{1}{1-\xi(x,y)} = SNR(x,y) + SNR(y,x) \qquad (14)$$

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

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

$$SSIM(x,y) \sim \left(1 - \frac{1}{SNR(x_L,y_L)+SNR(y_L,x_L)}\right) \cdot \left(1 - \frac{1}{SNR(x_H,y_H)+SNR(y_H,x_H)}\right) \qquad (15)$$

where $SNR(x_L,y_L)$, $SNR(y_L,x_L)$, $SNR(x_H,y_H)$, $SNR(y_H,x_H)$ 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 in [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 the Kodak dataset [17]. We have converted them to BT.709 YUV format and introduced three different types of distortions: 1) lossy compression artifacts, 2) Gaussian blur filtering, and 3) "salt-and-pepper" type noise.

TABLE I. ACCURACY OF THE PROPOSED METHOD FOR COMPUTING SSIM: H.264 ENCODING/DECODING, 1536X1024 IMAGES..

| File | QP=17 | | QP=22 | | QP=27 | | QP=32 | | QP=37 | | QP=42 | | QP=47 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta |
| k01 | 0.9949 | 0.0003 | 0.9867 | 0.0008 | 0.9697 | 0.0016 | 0.9337 | 0.0028 | 0.8687 | 0.0038 | 0.7766 | 0.0026 | 0.6648 | -0.0026 |
| k02 | 0.9928 | 0.0000 | 0.9784 | 0.0001 | 0.9447 | 0.0001 | 0.8962 | -0.0009 | 0.8318 | -0.0033 | 0.7569 | -0.0069 | 0.6843 | -0.0119 |
| k03 | 0.9879 | 0.0000 | 0.9768 | -0.0001 | 0.9669 | -0.0003 | 0.9506 | -0.0008 | 0.9251 | -0.0016 | 0.8947 | -0.0033 | 0.8617 | -0.0059 |
| k05 | 0.9906 | -0.0001 | 0.9782 | -0.0004 | 0.9603 | -0.0009 | 0.9327 | -0.0018 | 0.8937 | -0.0034 | 0.8453 | -0.0063 | 0.7928 | -0.0108 |
| k06 | 0.9941 | 0.0003 | 0.9869 | 0.0005 | 0.9757 | 0.0007 | 0.9551 | 0.0008 | 0.9168 | 0.0004 | 0.8552 | -0.0015 | 0.7619 | -0.0067 |
| k07 | 0.9930 | 0.0001 | 0.9851 | 0.0003 | 0.9709 | 0.0006 | 0.9427 | 0.0006 | 0.8920 | 0.0000 | 0.8126 | -0.0027 | 0.7108 | -0.0076 |
| k08 | 0.9889 | -0.0002 | 0.9781 | -0.0004 | 0.9707 | -0.0008 | 0.9599 | -0.0016 | 0.9427 | -0.0030 | 0.9173 | -0.0056 | 0.8811 | -0.0087 |
| k11 | 0.9956 | 0.0003 | 0.9853 | 0.0010 | 0.9626 | 0.0022 | 0.9326 | 0.0030 | 0.8876 | 0.0030 | 0.8180 | 0.0019 | 0.7076 | -0.0030 |
| k12 | 0.9893 | 0.0000 | 0.9623 | 0.0001 | 0.9320 | -0.0001 | 0.9180 | -0.0006 | 0.8989 | -0.0016 | 0.8708 | -0.0038 | 0.8376 | -0.0065 |
| k13 | 0.9898 | 0.0000 | 0.9645 | 0.0001 | 0.9328 | -0.0004 | 0.9149 | -0.0014 | 0.8931 | -0.0031 | 0.8649 | -0.0055 | 0.8321 | -0.0084 |
| k14 | 0.9918 | 0.0002 | 0.9799 | 0.0004 | 0.9611 | 0.0006 | 0.9264 | 0.0004 | 0.8701 | -0.0012 | 0.8019 | -0.0044 | 0.7361 | -0.0083 |
| k15 | 0.9887 | 0.0000 | 0.9776 | -0.0002 | 0.9646 | -0.0005 | 0.9443 | -0.0015 | 0.9169 | -0.0029 | 0.8888 | -0.0049 | 0.8673 | -0.0063 |
| k16 | 0.9967 | 0.0001 | 0.9911 | 0.0003 | 0.9753 | 0.0006 | 0.9363 | 0.0011 | 0.8557 | 0.0005 | 0.7283 | -0.0034 | 0.5750 | -0.0123 |
| k20 | 0.9932 | 0.0001 | 0.9826 | 0.0001 | 0.9637 | -0.0001 | 0.9281 | -0.0014 | 0.8704 | -0.0046 | 0.7942 | -0.0102 | 0.7149 | -0.0162 |
| k21 | 0.9909 | -0.0001 | 0.9805 | -0.0003 | 0.9645 | -0.0007 | 0.9393 | -0.0017 | 0.9069 | -0.0033 | 0.8707 | -0.0050 | 0.8339 | -0.0073 |
| k22 | 0.9907 | 0.0000 | 0.9804 | 0.0000 | 0.9677 | -0.0001 | 0.9433 | -0.0003 | 0.8989 | -0.0009 | 0.8316 | -0.0029 | 0.7461 | -0.0078 |
| k23 | 0.9908 | -0.0002 | 0.9687 | -0.0007 | 0.9367 | -0.0018 | 0.9121 | -0.0035 | 0.8840 | -0.0055 | 0.8510 | -0.0079 | 0.8090 | -0.0101 |
| k24 | 0.9945 | 0.0000 | 0.9820 | -0.0001 | 0.9431 | -0.0004 | 0.8858 | -0.0015 | 0.8310 | -0.0038 | 0.7617 | -0.0077 | 0.6729 | -0.0136 |
| k04 | 0.9913 | 0.0001 | 0.9741 | 0.0000 | 0.9555 | 0.0000 | 0.9282 | 0.0002 | 0.8788 | -0.0003 | 0.8172 | -0.0023 | 0.7594 | -0.0041 |
| k09 | 0.9889 | 0.0000 | 0.9739 | -0.0001 | 0.9632 | -0.0001 | 0.9453 | -0.0003 | 0.9128 | -0.0011 | 0.8726 | -0.0033 | 0.8375 | -0.0052 |
| k10 | 0.9912 | 0.0000 | 0.9701 | 0.0001 | 0.9484 | 0.0001 | 0.9269 | 0.0001 | 0.8918 | -0.0003 | 0.8395 | -0.0024 | 0.7723 | -0.0082 |
| k17 | 0.9913 | -0.0001 | 0.9765 | -0.0002 | 0.9562 | -0.0006 | 0.9225 | -0.0017 | 0.8716 | -0.0040 | 0.8097 | -0.0076 | 0.7471 | -0.0108 |
| k18 | 0.9877 | -0.0004 | 0.9705 | -0.0010 | 0.9575 | -0.0017 | 0.9441 | -0.0026 | 0.9286 | -0.0037 | 0.9107 | -0.0046 | 0.8900 | -0.0048 |
| k19 | 0.9928 | 0.0002 | 0.9846 | 0.0003 | 0.9718 | 0.0002 | 0.9471 | -0.0004 | 0.9041 | -0.0024 | 0.8395 | -0.0069 | 0.7560 | -0.0139 |
| **RMS** | 0.9915 | **0.0002** | 0.9781 | **0.0004** | 0.9591 | **0.0009** | 0.9321 | **0.0016** | 0.8909 | **0.0028** | 0.8360 | **0.0052** | 0.7727 | **0.0091** |

TABLE II. ACCURACY OF THE PROPOSED METHOD FOR COMPUTING SSIM: H.264 ENCODING/DECODING, 384X256 IMAGES.

| File | QP=17 | | QP=22 | | QP=27 | | QP=32 | | QP=37 | | QP=42 | | QP=47 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta |
| k01 | 0.99698 | 0.00037 | 0.99194 | 0.00097 | 0.97784 | 0.00241 | 0.94324 | 0.00516 | 0.86771 | 0.00803 | 0.74109 | 0.00880 | 0.57732 | 0.00272 |
| k02 | 0.99501 | 0.00008 | 0.98533 | 0.00023 | 0.95178 | 0.00078 | 0.87002 | 0.00155 | 0.75560 | 0.00090 | 0.64300 | -0.00305 | 0.56365 | -0.00703 |
| k03 | 0.99188 | 0.00019 | 0.98242 | 0.00052 | 0.96545 | 0.00115 | 0.93660 | 0.00200 | 0.89493 | 0.00280 | 0.84024 | 0.00155 | 0.78627 | -0.00283 |
| k05 | 0.99808 | 0.00015 | 0.99444 | 0.00043 | 0.98361 | 0.00114 | 0.95473 | 0.00263 | 0.89125 | 0.00511 | 0.77406 | 0.00763 | 0.61517 | 0.00595 |
| k06 | 0.99536 | 0.00015 | 0.98865 | 0.00034 | 0.96971 | 0.00083 | 0.92473 | 0.00164 | 0.84128 | 0.00247 | 0.71864 | 0.00087 | 0.61210 | 0.00010 |
| k07 | 0.99452 | 0.00031 | 0.99010 | 0.00059 | 0.98075 | 0.00107 | 0.96086 | 0.00157 | 0.92252 | 0.00145 | 0.85035 | -0.00080 | 0.74457 | -0.00555 |
| k08 | 0.99737 | 0.00038 | 0.99269 | 0.00095 | 0.98136 | 0.00176 | 0.95571 | 0.00298 | 0.91088 | 0.00426 | 0.83130 | 0.00527 | 0.69829 | 0.00666 |
| k11 | 0.99482 | 0.00028 | 0.98655 | 0.00067 | 0.96532 | 0.00127 | 0.91783 | 0.00214 | 0.84478 | 0.00142 | 0.76196 | -0.00031 | 0.66883 | -0.00503 |
| k12 | 0.99110 | 0.00020 | 0.97866 | 0.00042 | 0.95682 | 0.00083 | 0.92331 | 0.00111 | 0.89056 | 0.00117 | 0.85733 | 0.00099 | 0.81379 | -0.00187 |
| k13 | 0.99805 | 0.00007 | 0.99442 | 0.00019 | 0.98139 | 0.00053 | 0.93923 | 0.00123 | 0.84021 | 0.00153 | 0.67614 | -0.00186 | 0.49508 | -0.00777 |
| k14 | 0.99661 | 0.00010 | 0.98994 | 0.00028 | 0.97108 | 0.00072 | 0.92420 | 0.00131 | 0.83555 | 0.00043 | 0.71927 | -0.00351 | 0.58296 | -0.01225 |
| k15 | 0.99275 | 0.00017 | 0.98407 | 0.00026 | 0.96894 | 0.00068 | 0.94086 | 0.00142 | 0.90375 | 0.00185 | 0.85833 | 0.00106 | 0.80702 | -0.00049 |
| k16 | 0.99331 | -0.00004 | 0.98418 | -0.00005 | 0.96285 | -0.00011 | 0.91732 | -0.00047 | 0.84069 | -0.00135 | 0.74781 | -0.00335 | 0.65946 | -0.00897 |
| k20 | 0.99266 | 0.00012 | 0.98866 | 0.00020 | 0.97924 | 0.00030 | 0.95427 | 0.00047 | 0.90503 | -0.00029 | 0.85874 | -0.00133 | 0.81152 | -0.00237 |
| k21 | 0.99262 | 0.00007 | 0.98690 | 0.00025 | 0.97717 | 0.00053 | 0.95577 | 0.00103 | 0.91382 | 0.00182 | 0.84091 | 0.00163 | 0.74318 | -0.00203 |
| k22 | 0.99415 | 0.00006 | 0.98579 | 0.00012 | 0.96452 | 0.00026 | 0.91378 | -0.00016 | 0.82362 | -0.00229 | 0.71706 | -0.00778 | 0.62065 | -0.01467 |
| k23 | 0.99122 | -0.00056 | 0.98422 | -0.00092 | 0.97129 | -0.00162 | 0.94555 | -0.00291 | 0.90752 | -0.00497 | 0.86277 | -0.00770 | 0.81030 | -0.00943 |
| k24 | 0.99620 | 0.00024 | 0.99028 | 0.00054 | 0.97486 | 0.00113 | 0.93799 | 0.00191 | 0.85555 | 0.00213 | 0.73588 | -0.00039 | 0.58894 | -0.01315 |
| k04 | 0.99329 | -0.00016 | 0.98424 | -0.00042 | 0.96570 | -0.00098 | 0.92860 | -0.00179 | 0.86661 | -0.00295 | 0.78177 | -0.00573 | 0.69493 | -0.00935 |
| k09 | 0.99097 | 0.00022 | 0.98611 | 0.00041 | 0.97816 | 0.00075 | 0.96182 | 0.00125 | 0.92696 | 0.00132 | 0.86711 | 0.00144 | 0.78502 | -0.00182 |
| k10 | 0.99198 | 0.00007 | 0.98368 | 0.00009 | 0.96899 | 0.00004 | 0.94139 | -0.00020 | 0.89478 | -0.00067 | 0.81985 | -0.00315 | 0.72914 | -0.00769 |
| k17 | 0.99421 | 0.00013 | 0.98443 | 0.00031 | 0.96673 | 0.00036 | 0.93235 | 0.00020 | 0.87392 | -0.00140 | 0.79433 | -0.00467 | 0.69088 | -0.01134 |
| k18 | 0.99589 | 0.00012 | 0.98830 | 0.00032 | 0.97295 | 0.00073 | 0.93719 | 0.00127 | 0.85561 | 0.00123 | 0.70694 | -0.00226 | 0.54630 | -0.01039 |
| k19 | 0.99318 | -0.00006 | 0.98357 | -0.00018 | 0.96557 | -0.00039 | 0.92249 | -0.00078 | 0.84125 | -0.00251 | 0.77152 | -0.00393 | 0.70107 | -0.00336 |
| **RMS** | 0.99426 | **0.00022** | 0.98707 | **0.00048** | 0.97095 | **0.00100** | 0.93520 | **0.00189** | 0.87188 | **0.00285** | 0.78502 | **0.00417** | 0.68746 | **0.00758** |

In all cases, the SSIM values between the original and distorted images have been computed by method in the original paper [1]. Similarly, for all cases we have also computed SSIM values by the proposed method (7). For band separation, we used a circular-symmetric Gaussian filter with $\sigma = 3$. The differences between SSIM values obtained by our method and the reference method [1] are reported in Tables I-IV.

TABLE III. ACCURACY OF THE PROPOSED METHOD FOR COMPUTING SSIM: GAUSSIAN BLUR, 1536X1024 IMAGES.

| File | sigma=0.5 | | sigma=0.7 | | sigma=1 | | sigma=3 | | sigma=5 | | sigma=10 | | sigma=15 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta |
| **k01** | 0.98720 | 0.00048 | 0.94682 | 0.00165 | 0.87318 | 0.00211 | 0.55327 | -0.01726 | 0.47744 | -0.03227 | 0.44126 | -0.02791 | 0.43598 | -0.02557 |
| **k02** | 0.98573 | 0.00008 | 0.94714 | 0.00007 | 0.88935 | -0.00093 | 0.68282 | -0.01827 | 0.64281 | -0.01764 | 0.62244 | -0.01382 | 0.61872 | -0.01157 |
| **k03** | 0.99402 | 0.00002 | 0.97981 | -0.00001 | 0.95905 | -0.00053 | 0.86614 | -0.01306 | 0.83851 | -0.01443 | 0.82062 | -0.01167 | 0.81688 | -0.00989 |
| **k05** | 0.99394 | 0.00013 | 0.97588 | 0.00023 | 0.93818 | -0.00074 | 0.66111 | -0.02633 | 0.55650 | -0.03670 | 0.49014 | -0.02812 | 0.47903 | -0.02371 |
| **k06** | 0.99006 | 0.00021 | 0.95896 | 0.00061 | 0.90340 | 0.00021 | 0.65718 | -0.01601 | 0.60000 | -0.02167 | 0.57249 | -0.01801 | 0.56856 | -0.01613 |
| **k07** | 0.99494 | -0.00008 | 0.98365 | -0.00039 | 0.96785 | -0.00138 | 0.87293 | -0.01663 | 0.80660 | -0.01403 | 0.74749 | -0.00715 | 0.73933 | -0.00321 |
| **k08** | 0.98452 | 0.00052 | 0.93801 | 0.00147 | 0.86230 | 0.00079 | 0.54442 | -0.02261 | 0.45929 | -0.04163 | 0.40624 | -0.03386 | 0.39683 | -0.02919 |
| **k11** | 0.98950 | 0.00015 | 0.95933 | 0.00030 | 0.90901 | -0.00057 | 0.70088 | -0.01919 | 0.64987 | -0.02343 | 0.61996 | -0.01886 | 0.61386 | -0.01632 |
| **k12** | 0.99447 | 0.00001 | 0.98034 | -0.00010 | 0.95877 | -0.00091 | 0.86787 | -0.01407 | 0.84303 | -0.01567 | 0.82811 | -0.01359 | 0.82542 | -0.01226 |
| **k13** | 0.98144 | 0.00029 | 0.92398 | 0.00077 | 0.82716 | -0.00023 | 0.46356 | -0.01772 | 0.38538 | -0.02603 | 0.34533 | -0.01965 | 0.33909 | -0.01612 |
| **k14** | 0.99087 | -0.00001 | 0.96487 | -0.00034 | 0.91966 | -0.00214 | 0.69399 | -0.02678 | 0.62321 | -0.02971 | 0.57784 | -0.02213 | 0.56904 | -0.01755 |
| **k15** | 0.99268 | -0.00004 | 0.97245 | -0.00028 | 0.94195 | -0.00129 | 0.82807 | -0.01410 | 0.79545 | -0.01633 | 0.76964 | -0.01186 | 0.76280 | -0.00892 |
| **k16** | 0.99004 | 0.00012 | 0.96061 | 0.00036 | 0.91245 | -0.00006 | 0.73105 | -0.01430 | 0.69283 | -0.01578 | 0.67404 | -0.01293 | 0.67084 | -0.01125 |
| **k20** | 0.99160 | 0.00004 | 0.97006 | -0.00001 | 0.93853 | -0.00067 | 0.82382 | -0.01119 | 0.79265 | -0.01426 | 0.77086 | -0.01084 | 0.76631 | -0.00903 |
| **k21** | 0.98786 | 0.00009 | 0.95579 | 0.00006 | 0.90823 | -0.00114 | 0.72230 | -0.01587 | 0.67120 | -0.02187 | 0.64053 | -0.01709 | 0.63514 | -0.01424 |
| **k22** | 0.99040 | 0.00002 | 0.96446 | -0.00006 | 0.92320 | -0.00102 | 0.74711 | -0.01904 | 0.69702 | -0.01675 | 0.66932 | -0.01154 | 0.66455 | -0.00843 |
| **k23** | 0.99399 | -0.00016 | 0.98079 | -0.00059 | 0.96404 | -0.00143 | 0.89616 | -0.01090 | 0.87084 | -0.00964 | 0.85306 | -0.00616 | 0.84902 | -0.00367 |
| **k24** | 0.99288 | 0.00000 | 0.97165 | -0.00029 | 0.93133 | -0.00191 | 0.70978 | -0.02570 | 0.64305 | -0.02919 | 0.60436 | -0.02300 | 0.59750 | -0.01931 |
| **k04** | 0.99146 | -0.00007 | 0.96870 | -0.00040 | 0.93370 | -0.00161 | 0.78886 | -0.01777 | 0.74753 | -0.01971 | 0.71973 | -0.01498 | 0.71366 | -0.01167 |
| **k09** | 0.98837 | 0.00007 | 0.96191 | 0.00007 | 0.93092 | -0.00059 | 0.82522 | -0.01349 | 0.78759 | -0.01683 | 0.76389 | -0.01334 | 0.76003 | -0.01129 |
| **k10** | 0.98907 | 0.00002 | 0.96433 | -0.00015 | 0.93486 | -0.00116 | 0.82576 | -0.01746 | 0.78687 | -0.01906 | 0.76038 | -0.01491 | 0.75543 | -0.01235 |
| **k17** | 0.98826 | -0.00022 | 0.96069 | -0.00095 | 0.92650 | -0.00267 | 0.80626 | -0.01901 | 0.76054 | -0.01877 | 0.72108 | -0.01130 | 0.71121 | -0.00627 |
| **k18** | 0.98254 | 0.00003 | 0.93664 | -0.00016 | 0.87364 | -0.00162 | 0.63151 | -0.02020 | 0.55425 | -0.01938 | 0.50747 | -0.01122 | 0.49986 | -0.00686 |
| **k19** | 0.98872 | 0.00011 | 0.95763 | 0.00024 | 0.90834 | -0.00040 | 0.71598 | -0.01373 | 0.66602 | -0.02025 | 0.63962 | -0.01697 | 0.63520 | -0.01502 |
| **RMS** | 0.98978 | **0.00018** | 0.96197 | **0.00058** | 0.91876 | **0.00127** | 0.74227 | **0.01805** | 0.69289 | **0.02260** | 0.66235 | **0.01758** | 0.65678 | **0.01476** |

TABLE IV. ACCURACY OF THE PROPOSED METHOD FOR COMPUTING SSIM: NOISE, 1536X1024 IMAGES.

| File | p=0.00001 | | p=0.0005 | | p=0.001 | | p=0.005 | | p=0.01 | | p=0.05 | | p=0.1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta | SSIM | Delta |
| **k01** | 0.99903 | 0.00001 | 0.99429 | 0.00016 | 0.98857 | 0.00039 | 0.94685 | 0.00179 | 0.90066 | 0.00308 | 0.66301 | 0.00715 | 0.51428 | 0.00692 |
| **k02** | 0.99760 | 0.00001 | 0.98705 | 0.00002 | 0.97412 | 0.00003 | 0.87685 | -0.00020 | 0.77454 | -0.00089 | 0.35328 | -0.00377 | 0.19618 | -0.00228 |
| **k03** | 0.99837 | 0.00001 | 0.99119 | 0.00001 | 0.98273 | 0.00000 | 0.91882 | -0.00026 | 0.85090 | -0.00082 | 0.54850 | -0.00307 | 0.40787 | -0.00254 |
| **k05** | 0.99842 | 0.00006 | 0.99118 | 0.00025 | 0.98250 | 0.00049 | 0.91927 | 0.00181 | 0.85094 | 0.00298 | 0.53878 | 0.00301 | 0.38195 | 0.00123 |
| **k06** | 0.99826 | 0.00001 | 0.99119 | 0.00007 | 0.98259 | 0.00016 | 0.91792 | 0.00045 | 0.84912 | 0.00064 | 0.55184 | 0.00166 | 0.40942 | 0.00264 |
| **k07** | 0.99865 | -0.00002 | 0.99349 | -0.00013 | 0.98663 | -0.00016 | 0.93714 | -0.00113 | 0.88169 | -0.00216 | 0.60637 | -0.00724 | 0.45282 | -0.00830 |
| **k08** | 0.99896 | 0.00002 | 0.99328 | 0.00019 | 0.98633 | 0.00038 | 0.93630 | 0.00179 | 0.88357 | 0.00291 | 0.62646 | 0.00496 | 0.47785 | 0.00392 |
| **k11** | 0.99867 | 0.00002 | 0.99314 | 0.00007 | 0.98642 | 0.00012 | 0.93822 | 0.00043 | 0.88717 | 0.00061 | 0.66258 | 0.00104 | 0.54117 | 0.00104 |
| **k12** | 0.99751 | 0.00000 | 0.98667 | -0.00001 | 0.97341 | 0.00000 | 0.87508 | -0.00028 | 0.77109 | -0.00092 | 0.34640 | -0.00354 | 0.19323 | -0.00192 |
| **k13** | 0.99898 | 0.00002 | 0.99486 | 0.00006 | 0.98993 | 0.00014 | 0.95029 | 0.00065 | 0.90515 | 0.00102 | 0.66921 | 0.00215 | 0.51826 | 0.00206 |
| **k14** | 0.99835 | -0.00001 | 0.99190 | 0.00001 | 0.98361 | 0.00005 | 0.92031 | 0.00020 | 0.85224 | 0.00008 | 0.54490 | -0.00125 | 0.39510 | -0.00142 |
| **k15** | 0.99692 | -0.00003 | 0.98302 | -0.00014 | 0.96643 | -0.00027 | 0.84714 | -0.00163 | 0.72546 | -0.00347 | 0.29058 | -0.00677 | 0.16568 | -0.00320 |
| **k16** | 0.99860 | 0.00000 | 0.99306 | 0.00003 | 0.98600 | 0.00005 | 0.93347 | 0.00011 | 0.87764 | 0.00007 | 0.61034 | -0.00076 | 0.46907 | -0.00058 |
| **k20** | 0.99688 | -0.00003 | 0.98238 | -0.00016 | 0.96519 | -0.00031 | 0.84397 | -0.00179 | 0.72177 | -0.00360 | 0.30715 | -0.00476 | 0.19120 | -0.00147 |
| **k21** | 0.99884 | 0.00001 | 0.99381 | 0.00003 | 0.98732 | 0.00010 | 0.94062 | 0.00049 | 0.88761 | 0.00066 | 0.61701 | -0.00006 | 0.45622 | -0.00111 |
| **k22** | 0.99861 | 0.00000 | 0.99279 | 0.00001 | 0.98596 | 0.00000 | 0.93262 | -0.00023 | 0.87467 | -0.00061 | 0.58993 | -0.00272 | 0.43961 | -0.00260 |
| **k23** | 0.99810 | -0.00001 | 0.98975 | -0.00013 | 0.98002 | -0.00027 | 0.90565 | -0.00137 | 0.82478 | -0.00265 | 0.47568 | -0.00734 | 0.32576 | -0.00627 |
| **k24** | 0.99842 | 0.00003 | 0.99199 | 0.00007 | 0.98442 | 0.00012 | 0.92877 | 0.00050 | 0.86804 | 0.00065 | 0.58666 | 0.00007 | 0.43818 | -0.00065 |
| **k04** | 0.99837 | 0.00000 | 0.99075 | -0.00003 | 0.98146 | -0.00005 | 0.91511 | -0.00056 | 0.84378 | -0.00128 | 0.52395 | -0.00479 | 0.37573 | -0.00494 |
| **k09** | 0.99852 | 0.00001 | 0.99229 | 0.00004 | 0.98434 | 0.00007 | 0.92621 | 0.00013 | 0.86374 | 0.00007 | 0.56167 | -0.00171 | 0.40668 | -0.00218 |
| **k10** | 0.99875 | 0.00001 | 0.99389 | -0.00001 | 0.98791 | 0.00003 | 0.94274 | 0.00005 | 0.89303 | -0.00014 | 0.64885 | -0.00223 | 0.51264 | -0.00324 |
| **k17** | 0.99745 | -0.00003 | 0.98754 | -0.00018 | 0.97569 | -0.00036 | 0.88805 | -0.00184 | 0.79812 | -0.00363 | 0.44390 | -0.00773 | 0.30673 | -0.00594 |
| **k18** | 0.99816 | -0.00001 | 0.99028 | -0.00002 | 0.97994 | -0.00009 | 0.90481 | -0.00068 | 0.82380 | -0.00151 | 0.46808 | -0.00462 | 0.30914 | -0.00354 |
| **k19** | 0.99885 | 0.00002 | 0.99384 | 0.00013 | 0.98743 | 0.00026 | 0.94086 | 0.00109 | 0.88889 | 0.00172 | 0.62521 | 0.00182 | 0.47511 | 0.00021 |
| **RMS** | 0.99830 | **0.00002** | 0.99099 | **0.00011** | 0.98206 | **0.00022** | 0.91659 | **0.00103** | 0.84732 | **0.00193** | 0.54743 | **0.00423** | 0.40514 | **0.00360** |

Tables I and II report the accuracy of our method in the cases of lossy compression artifacts. Such artifacts have been introduced by using the H.264 video encoder [18], operating in Main Profile, "slow" encoding preset, and using fixed-QP rate control mode. QP values of 17, 22, 27, 32, 37, 42, and 47 have been used to control the degree of artifacts introduced. The columns "SSIM" list the reference SSIM values [1], and the column "Delta" lists the differences between the reference

SSIM and the one calculated by our method. Y-channel frame-average SSIM values are used as the basis for comparison. The last rows list RMS average values across all files. Table I reports results for 1536x1024 resolution compressed images, and Table II reports results for 384x256 resolution images. It can be observed that the differences between SSIM values computed by using our model and reference are very small, despite the broad range of variation of SSIM values, distortion, resolutions, and content types.

Table III reports the accuracy of our method in cases of blur artifacts. Such artifacts have been introduced by applying a circular-symmetric Gaussian blur filter, with variance parameter $\sigma$ set to values 0.5, 0.7, 1, 3, 5, 10, and 15. As can be observed, our method shows reasonably accurate performance in all these tests. The most significant deviations are observed in $\sigma = 3..5$ range, but vanish with smaller or larger $\sigma$.

Finally, Table IV reports the accuracy of our method in the cases of "salt and pepper" noise. This noise was introduced by negating a random subset of pixel values in the image. A binary random noise generator with pixel-level flip probabilities of p = 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, and 0.1 have been utilized. The same method was applied to all files. As can be observed, our method achieves high accuracy in computing SSIM values in all these tests.

## IV. Conclusions

In this paper, we have proposed an alternative model of SSIM computation, utilizing subband decomposition and identical distance measures in each subband. We have shown that this model performs very close to the original under various visual content types, resolutions, and distortions introduced. Specifically, we have studied its performance with a broad range of lossy compression artifacts, Gaussian blurring, and "salt-and-pepper"-type noise.

We have also explained the benefits of the proposed model from practical and methodological perspectives. Such benefits include a simple and intuitive explanation of why SSIM works better than full-band metrics, its connection to CSF and related phenomena of vision, SSIM-bosting pre-filtering and encoder optimization techniques, and mathematical connections between SSIM, MSE, and SNR metrics.

## References


[1] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, "Image quality assessment: from error visibility to structural similarity". IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.

[2] Z. Wang and A. C. Bovik, "A universal image quality index," IEEE Signal Processing Letters, vol. 9, no 3, pp. 81–84, 2002.

[3] Z. Wang, E. P. Simoncelli and A. C. Bovik, "Multiscale structural similarity for image quality assessment," 37th Asilomar Conference on Signals, Systems & Computers, 2003, vol.2, pp. 1398-1402.

[4] K. Zeng and Z. Wang, "3D-SSIM for video quality assessment," 19th IEEE International Conference on Image Processing, 2012, pp. 621-624.

[5] AK Moorthy and A. C. Bovik. "Efficient motion weighted spatio-temporal video SSIM index," Human Vision and Electronic Imaging XV, vol. 7527, pp. 440-448. SPIE, 2010.

[6] Y. Wang, T. Jiang, S. Ma, and W. Gao. "Spatio-temporal SSIM index for video quality assessment," 2012 Visual Communications and Image Processing, pp. 1-6. IEEE, 2012.

[7] R. Dosselmann and X. D. Yang, "Existing and emerging image quality metrics," Proceedings of the Canadian Conference on Electrical and Computer Engineering, pp.1906-1913, 2006.

[8] A.Horé, D. Ziou, "Image quality metrics: PSNR vs. SSIM," ICPR 2010.

[9] D. Brunet, E.R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," IEEE Transactions on Image Processing, 21(4), 2011, pp.1488-1499.

[10] T. Richter and K. J. Kim, "A MS-SSIM Optimal JPEG 2000 Encoder," 2009 Data Compression Conference, 2009, pp. 401-410.

[11] F. N. Rahayu, U. Reiter, T. Ebrahimi, A. Perkis, and P. Svensson, "SS-SSIM and MS-SSIM for digital cinema applications," Proc. SPIE 7240, Human Vision and Electronic Imaging XIV, 72400P (10 February 2009)

[12] S. Wang, A. Rehman, Z. Wang, S. Ma and W. Gao, "SSIM-MotivatedRate-Distortion Optimization for Video Coding," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 4, pp. 516-529, April 2012

[13] W. Wu and X. Zhang, "Code performance improvement scheme for X264 based on SSIM," 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, 2012, pp. 396-400.

[14] Z. Wang, K. Zeng, A. Rehman, H. Yeganeh, and S. Wang, "Objective video presentation QoE predictor for smart adaptive video streaming," Proc. SPIE 9599, Applications of Digital Image Processing XXXVIII, 95990Y, 22 September 2015.

[15] Y. Reznik, K. Lillevold, A. Jagannath, J. Greer, and J. Corley, "Optimal design of encoding profiles for ABR streaming," Proc. Packet Video Workshop, Amsterdam, The Netherlands, June 12, 2018.

[16] Y. Reznik, "Average Performance of Adaptive Streaming," Proc. Data Compression Conference (DCC'21), Snowbird, UT, March 2021.

[17] Eastman Kodak Company, Kodak lossless true color image suite (PhotoCD PCD0992).

[18] x264 encoder project, https://www.videolan.org/developers/x264.html.

[19] P. G. J. Barten, "Formula for the contrast sensitivity of the human eye," Proc. SPIE 5294, Image Quality and System Performance, 18 December 2003.

[20] Rec. ITU-R BT.500-13, "Methodology for the subjective assessment of the quality of television pictures," ITU-R, 2014.