Structural Similarity Metrics for Quality Image Fusion Assessment: Algorithms

Silvina Pistonesi\textsuperscript{1,2}, Jorge Martinez\textsuperscript{1}, Silvia María Ojeda\textsuperscript{3}, Ronny Vallejos\textsuperscript{4}

\textsuperscript{1} Departamento de Matemática, Universidad Nacional del Sur, Argentina (lpistone@criba.edu.ar)
\textsuperscript{2} Facultad Regional Bahía Blanca, Universidad Tecnológica Nacional, Argentina
\textsuperscript{3} Departamento de Matemática, Universidad Nacional del Sur, Argentina (martinez@uns.edu.ar)
\textsuperscript{4} Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, Argentina
\textsuperscript{3} Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, Argentina (ojeda@mate.uncor.edu)
\textsuperscript{4} Departamento de Matemática, Universidad Técnica Federico Santa María, Chile (ronny.vallejos@usm.cl)

Abstract

The wide use of image fusion techniques in different fields such as medical diagnostics, digital camera vision, military and surveillance applications, among others, has motivated the development of various image quality fusion metrics, in order to evaluate them. In this paper, we study and implement the algorithms of non-reference image structural similarity based metrics for fusion assessment: Piella’s metric, CVEJIC’s metric, Yang’s metric, and Codispersion Fusion Quality metric. We conduct the comparative experiment of the selected image fusion metrics over four multiresolution image fusion algorithms, performed on different pairs of images used in different applications.

Source Code

The reviewed source code for this article and documentation for these algorithms are available from the web page of this article\textsuperscript{1}. We used a MATLAB code in the implementation of algorithms.

Keywords: image fusion, image quality metrics, structural similarity, non-reference quality measures

\textsuperscript{1}http://www.ipol.im/
1 Introduction

Image fusion is the process of combining information available from two or more images of a scene into a single composite image which is more informative and suitable for both visual perception and computer processing. The goal of image fusion is to reduce uncertainty and minimize redundancy in the output, while maximizing the relevant information particular to an application of interest.

Quality assessment of different image fusion schemes is traditionally carried out by subjective evaluations. Even though this method [10] is reliable, it is expensive and too slow for real world applications. Therefore, several objective image quality measures that are consistent with human visual perception, have been proposed to predict image fusion quality automatically. They were classified into four groups according to their characteristics: information theory based metrics, image feature based metrics, human perception inspired fusion metrics, and image structural similarity based metrics [4]. In the context of measures based on image structural similarity, Piella’s metric [7], Cvejic’s metric [1] and Yang’s metric [13], and Codispersion Fusion Quality metric [8] were developed. These fusion performance measures are based on the Universal Image Quality index (Q) [11]. They do not require a reference image, namely, they do not need the construction of some kind of ideal fused image for using it as a reference for comparing with the experimental fused results, in order to evaluate the performance of different fusion algorithms [1, 7]. Also, these objective quality assessment measures are based on a sliding window approach, taking into account local measures to estimate how well the relevant information from the input images is present in the fused images, without introducing distortions.

The objective of this work is to provide reproducible implementations of the above mentioned measures, along with a quantitative evaluation of their performance.

Table 1 summarizes the details of the adopted notation in the pseudo-codes. The rest of the paper is organized as follows: in Section 2 we introduce the image structural similarity quality indexes and give an overview of the structural similarity-based metrics for image fusion. Along with this Section, the pseudo-codes which implement the described algorithms are presented. Finally, Section 3 contains experimental results obtained by the implemented algorithms, including examples to compare their performance.

2 Image Quality Measures

This section provides a description of the image structural similarity measures and then presents the structural similarity based metrics for image fusion. Additionally, the pseudo codes, which implements the described measures are exposed.

In what follows, we let \( x = \{x_{i,j} | 1 \leq i \leq M, 1 \leq j \leq N \} \) and \( y = \{y_{i,j} | 1 \leq i \leq M, 1 \leq j \leq N \} \), with \( M, N \in \mathbb{N} \), the original and test image signals, respectively.

2.1 Image Structural Similarity Measures

Structural information is one of the important components that affect the visual quality of digital images. In this context, the structural information are those attributes that represent the structure of objects in the scene, independent from the average luminance and contrast. The image structural similarity measures presented below are based on the assumption that the human visual system is highly adapted to extract structural information from the field of vision.
### Table 1: Summary of the notation used in pseudo-codes.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x, y$</td>
<td>Matrix of size $M \times N$, whose elements are the pixel values in 8-bit grayscale.</td>
</tr>
<tr>
<td>$f$</td>
<td>Matrix of size $M \times N$, whose elements are the pixel values in 8-bit grayscale corresponding to the fused image.</td>
</tr>
<tr>
<td>window</td>
<td>Local window for statistics (by default is a Gaussian kernel of size $11 \times 11$ and standard deviation 1.5).</td>
</tr>
<tr>
<td>$m_x$</td>
<td>Matrix whose elements are the local means computed in each sliding window $w$ of the $x$ image.</td>
</tr>
<tr>
<td>$s_x^2$</td>
<td>Matrix whose elements are the local variances computed in each sliding window $w$ of the $x$ image.</td>
</tr>
<tr>
<td>$s_{xy}$</td>
<td>Matrix whose elements are the local covariances computed in each sliding window $w$ between $x$ and $y$ images.</td>
</tr>
<tr>
<td>$q_{\text{map}}$</td>
<td>Matrix whose elements are the local $Q$ index values computed in each sliding window $w$ between two images.</td>
</tr>
<tr>
<td>ssim_map</td>
<td>Matrix whose elements are the local SSIM index values computed in each sliding window $w$ between two images.</td>
</tr>
<tr>
<td>$x_s, x_{s+h}$</td>
<td>Matrix whose elements are the pixel values of the set $X' = {s \in x : s + h \in x}$ and of the set ${s' = s + h : s \in X'}$, respectively, in a specific direction $h$.</td>
</tr>
<tr>
<td>$c_{q_{\text{map}}}$</td>
<td>Matrix whose elements are the local $CQ$ index values computed in each sliding window $w$ between two images, in a specific direction $h$.</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Matrix whose elements are the values of the local weights $\lambda$ computed in each sliding window $w$ of the source images.</td>
</tr>
<tr>
<td>$c$</td>
<td>Matrix whose elements are the values of the local weights $c$ computed in each sliding window $w$ of the source images.</td>
</tr>
<tr>
<td>$C$</td>
<td>Matrix whose elements are the values of the numerator of the local weight $c$ computed in each sliding window $w$ of the source images.</td>
</tr>
<tr>
<td>$x', y'$ and $f'$</td>
<td>Matrix whose elements are the Euclidean norm of the horizontal and vertical gradient images.</td>
</tr>
<tr>
<td>$\text{sim}$</td>
<td>Matrix whose elements are the values of the local weighting factor $\text{sim}$, computed in each sliding window $w$ of the source images and the fused image.</td>
</tr>
<tr>
<td>$c_{q_{\text{maps}}}$</td>
<td>List where each element is a $c_{q_{\text{map}}}$ in a specific direction $h$.</td>
</tr>
<tr>
<td>$c_{q_{\text{map}}_\text{max}}$</td>
<td>Matrix whose elements are the maximum local $CQ$ index values according the different directions $h$, computed in each sliding window $w$ between two images.</td>
</tr>
<tr>
<td>$c_{q_{\text{max}}_{\text{map}}}$</td>
<td>Matrix whose elements are the local $CQ_{\text{max}}$ index values, computed in each sliding window $w$ between two images.</td>
</tr>
<tr>
<td>$q_{S_{\text{map}}}, q_{W_{\text{map}}}, q_{E_{1_\text{map}}}, q_{C_{\text{map}}}, q_{Y_{\text{map}}}, c_{Q_{\text{M_{\text{map}}}}}$</td>
<td>Matrix whose elements are the local $Q_S, Q_W, Q_{E_1}, Q_C, Q_Y$ and $CQ_M$ index values, computed in each sliding window $w$ of size $w_{\text{size}}$ respectively (quality maps of the fused image).</td>
</tr>
</tbody>
</table>
2.1.1 Universal Image Quality Index (Q)

The Universal Image Quality index \( Q \) was introduced by Wang and Bovik [11] and is defined as follows:

\[
Q(x, y) = \frac{4S_{xy}\bar{x}\bar{y}}{(S_x^2 + S_y^2)(\bar{x}^2 + \bar{y}^2)},
\]

where \( \bar{x} \) and \( \bar{y} \) are the sample average values of images \( x \) and \( y \), \( S_x \), \( S_y \) and \( S_{xy} \) are the sample deviations and the sample covariance, respectively.

This index can be decomposed in the following three different factors:

\[
Q(x, y) = s(x, y) \cdot l(x, y) \cdot c(x, y) = \frac{S_{xy}}{S_x S_y} \cdot \frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2} \cdot \frac{2S_x S_y}{S_x^2 + S_y^2}.
\]

The first factor in (2) measures the degree of linear correlation between \( x \) and \( y \); the second measures the similarity between the luminance of \( x \) and \( y \), and the third factor measures the similarity related to the contrast between the images.

Considering image signals are generally non-stationary and image distortions may be space-variant, it is more appropriate to measure the image quality index \( Q \) over local regions and then combine the different results into a single measure. Therefore, Wang and Bovik [11] propose to use a sliding window approach: starting from the top-left corner of the two images \( x \), \( y \), a sliding window of a fixed size block by block over the entire image until bottom-right corner is reached. Finally, the overall image quality index \( Q \) is determined by averaging all local quality indexes \( Q(x, y|w) \) computed in all sliding windows \( w \in W \)

\[
Q = \sum_{w \in W} \frac{Q(x, y|w)}{|W|},
\]

with \( W \) the family of all windows and \( |W| \) is the cardinality of \( W \).

**Pseudo-code of Q Index** A pseudo-code for the implementation of the Q index algorithm is presented as follows:

A Matlab implementation of the Q index is available online at [https://ece.uwaterloo.ca/~z70wang/research/quality_index/demo.html](https://ece.uwaterloo.ca/~z70wang/research/quality_index/demo.html).

2.1.2 Structural Similarity Index (SSIM)

In order to measure the structural distortions of two images \( x \) and \( y \), Wang et al. proposed a structural similarity (SSIM) index [12], defined as follows:

\[
SSIM(x, y) = [s(x, y)]^\gamma \cdot [l(x, y)]^\alpha \cdot [c(x, y)]^\beta = \left( \frac{S_{xy} + C_3}{S_x S_y + C_3} \right)^\gamma \cdot \left( \frac{2\bar{x}\bar{y} + C_1}{\bar{x}^2 + \bar{y}^2 + C_1} \right)^\alpha \cdot \left( \frac{2S_x S_y + C_2}{S_x^2 + S_y^2 + C_2} \right)^\beta.
\]

The parameters \( \alpha \), \( \beta \) and \( \gamma \) adjust the relative importance of the three components. The constants \( C_1 \), \( C_2 \) and \( C_3 \) are included to avoid instability when denominators are very close to zero. In order to simplify the expression (4), Wang et al. set \( \alpha = \beta = \gamma = 1 \) and \( C_3 = C_2/2 \). This results in a specific form of the SSIM index:

\[
SSIM(x, y) = \frac{(2\bar{x}\bar{y} + C_1)(2S_{xy} + C_2)}{(\bar{x}^2 + \bar{y}^2 + C_1)(S_x^2 + S_y^2 + C_2)}.
\]
Algorithm 1: Pseudo-code of the Q similarity quality index algorithm.

**input**: original and test image signals \( x \) and \( y \), window size \( w_{\text{size}} \).

**output**: \( q \) index value between two images and \( q_{\text{map}} \), quality map of the test image.

- \( m_x \leftarrow \text{mean}(x, w_{\text{size}}) \) Computes the local mean.
- \( m_y \leftarrow \text{mean}(y, w_{\text{size}}) \)
- \( s_x^2 \leftarrow \text{variance}(x, w_{\text{size}}) \) Computes the local variance.
- \( s_y^2 \leftarrow \text{variance}(y, w_{\text{size}}) \)
- \( s_{xy} \leftarrow \text{covariance}(x, y, w_{\text{size}}) \) Computes the local covariance.
- \( \text{den}_c \leftarrow s_x^2 + s_y^2 \) Computes the contrast factor denominator (2).
- \( \text{den}_l \leftarrow m_x^2 + m_y^2 \) Computes the luminance factor denominator (2).
- \( q_{\text{map}} \leftarrow 1 \) Initialized with ones.

forall the sliding window \( w, w \in W \) do

if \( \text{den}_c(w) = 0 \) and \( \text{den}_l(w) \neq 0 \) then

\[
q_{\text{map}}(w) \leftarrow l(x, y|w)
\]

else

\[
q_{\text{map}}(w) \leftarrow s(x, y|w)) \cdot l(x, y|w) \cdot c(x, y|w)
\]

end

\( q \leftarrow \frac{1}{|W|} \sum_{w \in W} q_{\text{map}}(w) \) Compute the \( Q \) index (3).

In [12], the authors above mentioned considered \( C_1 = (k_1L)^2 \) and \( C_2 = (k_2L)^2 \), where \( L \) is the dynamic range of the pixel values (255 for 8-bit grayscale images), and \( k_1 \) and \( k_2 \) are small constants. The SSIM measure uses the parameter setting: \( k_1 = 0.01 \) and \( k_2 = 0.03 \). These values are somewhat arbitrary, but Wang et al. found that in their experiments on benchmark databases, the performance of the SSIM index algorithm was fairly insensitive to variations of the values.

The \( Q \) index, defined in (1), corresponds to a special case of SSIM index, if \( C_1 = C_2 = 0 \).

As stated above, it is suitable for image quality assessment to apply the SSIM index locally rather than globally. To evaluate the overall image quality, they used a mean SSIM (MSSIM) index, computed by averaging the local indexes \( \text{SSIM}(x, y|w) \) in \( w \in W \),

\[
\text{MSSIM} = \sum_{w \in W} \frac{\text{SSIM}(x, y|w)}{|W|}. \tag{6}
\]

With the aim that the MSSIM measure exhibits a better consistency with the qualitative visual appearance, Wang et al. used for default a 11 × 11 circular symmetric Gaussian weighting function \( w = \{w_i|i = 1, \ldots, M \cdot N\} \), with standard deviation 1.5 samples, normalized to unit sum \( \left( \sum_{i=1}^{M \cdot N} w_i \right) \).

**Pseudo-code of SSIM Index** A pseudo-code for the implementation of the SSIM index algorithm is shown in Algorithm 2:

An online Matlab implementation of SSIM index is available at [https://ece.uwaterloo.ca/~z70wang/research/ssim/](https://ece.uwaterloo.ca/~z70wang/research/ssim/).

### 2.1.3 Codispersion Quality Index (CQ)

Ojeda et al. [5] suggested a modification of \( Q \) index based on the codispersion coefficient. This similarity measure, labeled \( CQ \) index, captures different levels of spatial similarity between two images by considering different directions, \( h \), in the two-dimensional space. This is not commonly assessed by other measures of similarity between images. The \( CQ \) index, in a specific direction \( h \), is
Algorithm 2: Pseudo-code of the SSIM similarity quality index algorithm.

**input**: original and test image signals \(x\) and \(y\), constants \(k = [k_1 \ k_2]\), local window for statistics \(window\), dynamic range of the images \(L\) and window size \(w\_size\).

**output**: \(mssim\) index value between two images and \(ssim\_map\), quality map of the test image.

\[
\begin{align*}
C_1 &\leftarrow (k_1 \cdot L)^2 \\
C_2 &\leftarrow (k_2 \cdot L)^2 \\
x &\leftarrow \text{convolution}(x, \text{window}) \\
y &\leftarrow \text{convolution}(y, \text{window}) \\
w\_size &\leftarrow \text{size of the window}. \\
m_x &\leftarrow \text{mean}(x, w\_size) \quad \text{Computes the local mean.} \\
m_y &\leftarrow \text{mean}(y, w\_size) \\
s_x^2 &\leftarrow \text{variance}(x, w\_size) \quad \text{Computes the local variance.} \\
s_y^2 &\leftarrow \text{variance}(y, w\_size) \\
s_{xy} &\leftarrow \text{covariance}(x, y, w\_size) \quad \text{Computes the local covariance.} \\
den_l &\leftarrow m_x^2 + m_y^2 + C_1 \quad \text{Computes the luminance factor denominator (4).} \\
den_c &\leftarrow s_x^2 + s_y^2 + C_2 \quad \text{Computes the contrast factor denominator (4).} \\
\text{ssim\_map} &\leftarrow 1 \quad \text{Initialized with ones.} \\
\text{forall the sliding window } w, w \in W \text{ do} \\
\text{if } C_1 > 0 \text{ and } C_2 > 0 \text{ then} \\
\quad \text{ssim\_map}(w) &\leftarrow s(x, y|w) \cdot l(x, y|w) \cdot c(x, y|w) \quad \text{Computes the SSIM index at } w \text{ window (5).} \\
\text{else} \\
\quad \text{if } \text{den}_l(w) \cdot \text{den}_c(w) > 0 \text{ then} \\
\quad \quad \text{ssim\_map}(w) &\leftarrow s(x, y|w) \cdot l(x, y|w) \cdot c(x, y|w) \quad \text{Computes the SSIM index at } w \text{ window (5).} \\
\quad \quad \text{if } \text{den}_l(w) \neq 0 \text{ and } \text{den}_c(w) = 0 \text{ then} \\
\quad \quad \quad \text{ssim\_map}(w) &\leftarrow l(x, y|w) \quad \text{Computes the luminance factor at } w \text{ window (4).} \\
\text{mssim} &\leftarrow \frac{1}{|W|} \sum_{w \in W} \text{ssim\_map}(w) \quad \text{Computes the MSSIM index (6).}
\end{align*}
\]
defined by the following equation:

\[
CQ(x, y, h) = \hat{\rho}(h) \cdot l(x, y) \cdot c(x, y) = \frac{\sum_{s, s+h \in D} a_s b_s}{\sqrt{\hat{V}_x(h) \hat{V}_y(h)}} \cdot \frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2} \cdot \frac{2S_xS_y}{S_x^2 + S_y^2},
\]

where \( \hat{\rho}(h) \) is the sample codispersion coefficient in the direction \( h \), with \( s = (i, j) \), \( h = (h_1, h_2) \), \( D \subset \mathbb{Z}^d \), \( D \) finite set, \( a_s = x_{s+h} - x_s \), \( b_s = y_{s+h} - y_s \), \( \hat{V}_x(h) = \sum_{s, s+h \in D} a_s^2 \), and \( \hat{V}_y(h) = \sum_{s, s+h \in D} b_s^2 \).

It is obvious that \( |\hat{\rho}(h)| \leq 1 \).

The \( CQ \) value can be calculated locally and averaged to get the overall index:

\[
CQ = \frac{\sum_{w \in W} CQ(x, y, h|w)}{|W|}.
\]

**Pseudo-code of CQ Index** A pseudo-code for the implementation of the previously described CQ index algorithm, is presented in Algorithm 3.

Figure 1: (a) An image \( x \) of 8 × 8 pixels size where the arrows indicate the direction \( h \) considered to compute the \( CQ \) index. (b) The \( x_s \) (subimage of \( x \)) composed of the pixels of the set \( X' = \{ s \in x : s + h \in x \} \) (in red) and the \( x_{s+h} \) (subimage of \( x \)) composed of the pixels of the set \( \{ s' = s + h \in x : s \in X' \} \) (in blue).

### 2.2 Image Fusion Metrics

This subsection provides an overview of the metrics based on structural similarity for image fusion and its pseudo-code\(^2\).

#### 2.2.1 Piella’s Metrics (\( Q_S \), \( Q_W \) and \( Q_E \))

Three fusion quality metrics based on \( Q \) index were proposed by Piella and Heijmans [7]. In these measures, \( Q(x, f|w) \) and \( Q(y, f|w) \) are calculated between the source images (\( x \) and \( y \)) and the fused image (\( f \)) in a sliding window \( w \).

\(^2\)We denoted element-wise product of matrices with \( \cdot^* \), the division with \( \cdot/ \) and the power with \( \cdot^\circ \).
Algorithm 3: Pseudo-code of the CQ similarity quality index algorithm.

**input**: original and test image signals \(x\) and \(y\), direction \(h\), window size \(w_{size}\).

**output**: \(cq\) index value between two images and \(cq\_map\), quality map of the test image in a specific direction \(h\).

\[
\begin{align*}
    m_x & \leftarrow \text{mean}(x, w_{size}) \quad \text{Computes the local mean.} \\
    m_y & \leftarrow \text{mean}(y, w_{size}) \\
    s^2_x & \leftarrow \text{variance}(x, w_{size}) \quad \text{Computes the local variance.} \\
    s^2_y & \leftarrow \text{variance}(y, w_{size}) \\
    x_s & \leftarrow \text{subimage of } x \text{ composed of the pixels of the set } X' = \{ s \in x : s + h \in x \} \quad \text{See Figure 1.} \\
    x_{s+h} & \leftarrow \text{subimage of } x \text{ composed of the pixels } \{ s' = s + h : s \in X' \} \\
    y_s & \leftarrow \text{subimage of } y \text{ composed of the pixels of the set } Y' = \{ s \in y : s + h \in y \} \\
    y_{s+h} & \leftarrow \text{subimage of } y \text{ composed of the pixels } \{ s' = s + h : s \in Y' \} \\
    a_s & \leftarrow x_{s+h} - x_s \\
    b_s & \leftarrow y_{s+h} - y_s \\
    \hat{V}_x & \leftarrow \sum_{a_s^2} \\
    \hat{V}_y & \leftarrow \sum_{b_s^2} \\
    \text{den}_\hat{\rho} & \leftarrow \sqrt{\hat{V}_x \cdot \hat{V}_y} \quad \text{Computes the } \hat{\rho} \text{ factor denominator (7).} \\
    \text{den}_l & \leftarrow m_x^2 + m_y^2 \quad \text{Computes the luminance factor denominator (7).} \\
    \text{den}_c & \leftarrow s_x^2 + s_y^2 \quad \text{Computes the contrast factor denominator (7).} \\
    \text{den} & \leftarrow \text{den}_\hat{\rho} \cdot \text{den}_l \cdot \text{den}_c \quad \text{Computes the CQ index denominator (7).} \\
    cq\_map & \leftarrow 1 \quad \text{Initialized with ones.}
\end{align*}
\]

**for all the sliding window** \(w, w \in W\) **do**

**if** \(\text{den}(w) \neq 0\) **then**

\[
    cq\_map(w) \leftarrow \hat{\rho}(w) \cdot l(x, y|w) \cdot c(x, y|w) \quad \text{Computes the CQ index at } w \text{ window (7).}
\]

**else**

**if** \(\text{den}_\hat{\rho}(w) \neq 0 \text{ and } \text{den}_l(w) \neq 0 \text{ and } \text{den}_c(w) = 0\) **then**

\[
    cq\_map(w) \leftarrow \hat{\rho}(w) \cdot l(x, y|w)
\]

**if** \(\text{den}_\hat{\rho}(w) \neq 0 \text{ and } \text{den}_l(w) = 0 \text{ and } \text{den}_c(w) \neq 0\) **then**

\[
    cq\_map(w) \leftarrow \hat{\rho}(w) \cdot c(x, y|w)
\]

**if** \(\text{den}_\hat{\rho}(w) \neq 0 \text{ and } \text{den}_l(w) = 0 \text{ and } \text{den}_c(w) = 0\) **then**

\[
    cq\_map(w) \leftarrow \hat{\rho}(w)
\]

**if** \(\text{den}_\hat{\rho}(w) = 0 \text{ and } \text{den}_l(w) \neq 0 \text{ and } \text{den}_c(w) \neq 0\) **then**

\[
    cq\_map(w) \leftarrow l(x, y|w) \cdot c(x, y|w)
\]

**if** \(\text{den}_\hat{\rho}(w) = 0 \text{ and } \text{den}_l(w) \neq 0 \text{ and } \text{den}_c(w) = 0\) **then**

\[
    cq\_map(w) \leftarrow l(x, y|w)
\]

**if** \(\text{den}_\hat{\rho}(w) = 0 \text{ and } \text{den}_l(w) = 0 \text{ and } \text{den}_c(w) \neq 0\) **then**

\[
    cq\_map(w) \leftarrow c(x, y|w)
\]

\[
    cq \leftarrow \frac{1}{|W|} \sum_{w \in W} cq\_map(w) \quad \text{Computes CQ index (8).}
\]
The first measure is the following:

\[ Q_S(x, y, f) = \frac{1}{|W|} \sum_{w \in W} \left[ \lambda(w) Q(x, f|w) + (1 - \lambda(w)) Q(y, f|w) \right], \tag{9} \]

where the weight \( \lambda(w) \) is defined as

\[ \lambda(w) = \frac{s(x|w)}{s(x|w) + s(y|w)}. \tag{10} \]

The \( s(x|w) \) and \( s(y|w) \) are the local saliencies of the two input images \( x \) and \( y \) within the window \( w \), respectively. They reflect the local relevance of the source image within the window \( w \), and it may depend on contrast, sharpness, or entropy. In Piella’s implementation, \( s(\cdot|w) \) is the variance of image within window \( w \).

**Pseudo-code of \( Q_S \) metric** A pseudo-code for the implementation of the \( Q_S \) metric algorithm is shown in Algorithm 5. The first pseudo-code corresponds to the implementation of the local weight \( \lambda(w) \), indicating the relative importance of source images.

**Algorithm 4:** Pseudo-code of the local weight \( \lambda \) algorithm.

- **input**: source images \( x \) and \( y \), window size \( w_{\text{size}} \).
- **output**: weight \( \lambda \), a map whose elements are the values of \( \lambda \), computed in each window \( w \) of the input images.

\[
\begin{align*}
s_x^2 &\leftarrow \text{variance}(x, w_{\text{size}}) \\
s_y^2 &\leftarrow \text{variance}(y, w_{\text{size}}) \\
\text{forall the sliding window } w, w \in W &\text{ do} \\
\quad &\text{den}(w) \leftarrow s_x^2(w) + s_y^2(w) \\
\quad &\text{if } \text{den}(w) = 0 \text{ then} \\
\quad &\quad \lambda(w) \leftarrow 0 \\
\quad &\text{else} \\
\quad &\quad \lambda(w) \leftarrow \frac{s_x^2(w)}{s_x^2(w) + s_y^2(w)}
\end{align*}
\]

**Computes the variance.**

**Computes the weight (10).**

**Algorithm 5:** Pseudo-code of the \( Q_S \) fusion quality metric algorithm.

- **input**: source images \( x \) and \( y \), fused image \( f \), window size \( w_{\text{size}} \).
- **output**: \( q_S \), the fusion quality metric value and \( q_S\_map \), the quality assessment map of the fusion image.

\[
\begin{align*}
q_{\text{map}_{xf}} &\leftarrow Q(x, f|w_{\text{size}}) \quad \triangleright \text{Algorithm 1.} \\
q_{\text{map}_{yf}} &\leftarrow Q(y, f|w_{\text{size}}) \quad \triangleright \text{Algorithm 1.} \\
\lambda &\leftarrow \text{weight}\_\lambda(x, y, w_{\text{size}}) \quad \triangleright \text{Algorithm 4.} \\
q_{\text{map}} &\leftarrow \lambda \cdot q_{\text{map}_{xf}} + (1 - \lambda) \cdot q_{\text{map}_{yf}}(w) \\
q_S &\leftarrow \frac{1}{|W|} \sum_{w \in W} q_{\text{map}}(w)
\end{align*}
\]

**Computes the \( Q_S \) metric (9).**

The second, **weighted fusion quality index**, gives more weight to those windows where the saliency of the input images is higher. These correspond to areas which are likely to be perceptually important parts of the underlying scene. It is defined as:

\[ Q_W(x, y, f) = \sum_{w \in W} c(w) \left[ \lambda(w) Q(x, f|w) + (1 - \lambda(w)) Q(y, f|w) \right], \tag{11} \]
where $\lambda(w)$ is calculated as in (10) and the coefficient $c(w)$ is computed as

$$c(w) = \frac{C(w)}{\sum_{w' \in W} C(w')}$$

with

$$C(w) = \max \{s(x|w), s(y|w)\}.$$ 

In Piella’s implementation, $C(w)$ is defined as in (13). She also suggested another way to calculate it, as $C(w) = s(x|w) + s(y|w)$.

**Remark.** If the weight $c$ denominator in (12) is zero, we suggested to set $c = 1/|W|$, so in this case the $Q_W = Q_S$.

**Pseudo-code of $Q_W$ metric** A pseudo-code for the implementation of the $Q_W$ metric algorithm, described above, is presented in Algorithm 7. First, the pseudo-code of the implementation of the coefficient $c(w)$, is given

**Algorithm 6:** Pseudo-code of weight $c$ algorithm.

<table>
<thead>
<tr>
<th>input</th>
<th>source images $x$ and $y$, window size $w_{size}$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>$c$ coefficient, a map whose elements are the values of $c$, computed in each window $w$ of the input images.</td>
</tr>
<tr>
<td>$s^2_x \leftarrow \text{variance}(x, w_{size})$</td>
<td>Computes the local variance.</td>
</tr>
<tr>
<td>$s^2_y \leftarrow \text{variance}(y, w_{size})$</td>
<td></td>
</tr>
<tr>
<td>$c \leftarrow 0$</td>
<td>Initialized with zeros.</td>
</tr>
<tr>
<td>forall the sliding window $w$, $w \in W$ do</td>
<td></td>
</tr>
<tr>
<td>$C(w) \leftarrow \max {s^2_x(w), s^2_y(w)}$</td>
<td>Computes $C$ at $w$ window (13).</td>
</tr>
<tr>
<td>$\text{den} \cdot c \leftarrow \sum_{w \in W} C(w)$</td>
<td></td>
</tr>
<tr>
<td>if $\text{den} \cdot c \neq 0$ then</td>
<td></td>
</tr>
<tr>
<td>$c \leftarrow C./\sum_{w \in W} C(w)$</td>
<td>Computes the $c$ coefficient (12).</td>
</tr>
<tr>
<td>else</td>
<td></td>
</tr>
<tr>
<td>$c \leftarrow 1/</td>
<td>W</td>
</tr>
</tbody>
</table>

**Algorithm 7:** Pseudo-code of the $Q_W$ fusion quality metric algorithm.

<table>
<thead>
<tr>
<th>input</th>
<th>source images $x$, $y$, and fused image $f$, window size $w_{size}$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>$q_W$, the fusion quality metric value and $q_W$-map, the quality assessment map of the fused image.</td>
</tr>
<tr>
<td>$q_{\text{map}}<em>{xf} \leftarrow Q(x, f, w</em>{size})$</td>
<td>$\triangleright$ Algorithm 1.</td>
</tr>
<tr>
<td>$q_{\text{map}}<em>{yf} \leftarrow Q(y, f, w</em>{size})$</td>
<td>$\triangleright$ Algorithm 1.</td>
</tr>
<tr>
<td>$\lambda \leftarrow \text{weight} \cdot \lambda(x, y, w_{size})$</td>
<td>$\triangleright$ Algorithm 4.</td>
</tr>
<tr>
<td>$c \leftarrow \text{weight} \cdot c(x, y, w_{size})$</td>
<td>$\triangleright$ Algorithm 6.</td>
</tr>
<tr>
<td>$q_{W\text{-map}} \leftarrow c^\ast (\lambda \cdot q_{\text{map}}<em>{xf} + (1 - \lambda) \cdot q</em>{\text{map}}_{yf})$</td>
<td>Computes the $Q_W$ metric (11).</td>
</tr>
<tr>
<td>$q_W \leftarrow \sum_{w \in W} q_{W\text{-map}}$</td>
<td></td>
</tr>
</tbody>
</table>

Finally, they introduced the *edge-dependent fusion quality index* [7]. This measure takes into account human visual system characteristics, such as sensitivity to edges information. It is obtained as follows:

$$Q_{E_1}(x, y, f) = Q_W(x, y, f) \cdot Q_W(x', y', f')^\alpha.$$  (14)
Another variant of this metric was suggested by Piella in [6], defined as follows:

\[ Q_{E_1}(x, y, f) = Q_W(x, y, f)^{1-\alpha} \cdot Q_W(x', y', f')^\alpha. \]  \hspace{1cm} (15)

In (14) and (15), \(Q_W(x', y', f')\) is the \(Q_W\) calculated with the “edge images” \(x', y'\) and \(f'\) (e.g., the Euclidean norm of the horizontal and vertical gradient images), and \(\alpha\) is a parameter that weights the edge contribution information, \(\alpha \in [0, 1]\). The closer \(\alpha\) is to one, the more important is the edge image.

**Pseudo-code of \(Q_{E_1}\) metric** A pseudo-code for the implementation of the \(Q_{E_1}\) metric algorithm, described above, is presented in Algorithm 8.

**Algorithm 8:** Pseudo-code of the \(Q_{E_1}\) fusion quality metric algorithm.

| input | source images \(x, y\) and fused image \(f\), window size \(w_{\text{size}}\) and \(\alpha\) value. |
| output: \(q_{E_1}\), the fusion quality metric value and \(q_{E_1}\) map, the quality assessment map of the fused image. |

\[ x' \leftarrow \text{norm\_gradient}(x) \]  
\[ y' \leftarrow \text{norm\_gradient}(y) \]  
\[ f' \leftarrow \text{norm\_gradient}(f) \]  
\[ [q_{w_{xyf}}, q_{w_{map_{xyf}}}] \leftarrow Q_W(x, y, f, w_{\text{size}}) \]  
\[ [q_{w_{x'y'f'}}, q_{w_{map_{x'y'f'}}}] \leftarrow Q_W(x', y', f', w_{\text{size}}) \]  
\[ q_{E_1}\text{\_map} \leftarrow q_{w_{map_{xyf}}} \cdot (q_{w_{map_{x'y'f'}}})^\alpha \]  
\[ q_{E_1} \leftarrow q_{w_{xyf}} \cdot q_{w_{x'y'f'}}^\alpha \]

The pseudo-code of the implementation of the \(Q_{E_2}\) algorithm is analogous to that of \(Q_{E_1}\) metric.

### 2.2.2 Cvejic’s Metric (\(Q_C\))

Cvejic et al. defined a fusion performance measure [1] that takes into account the local similarity between the source image and the fused image within the same spatial position, as follows:

\[ Q_C = \frac{1}{|W|} \sum_{w \in W} \text{sim}(x, y, f|w) \cdot Q(x, f|w) + (1 - \text{sim}(x, y, f|w)) \cdot Q(y, f|w), \]  \hspace{1cm} (16)

with

\[ \text{sim}(x, y, f|w) = \begin{cases} 
0, & \text{if } \frac{S_{xf}}{S_{xf} + S_{yf}} < 0, \\
\frac{S_{xf}}{S_{xf} + S_{yf}}, & \text{if } 0 \leq \frac{S_{xf} + S_{yf}}{S_{xf}} \leq 1, \\
1, & \text{if } \frac{S_{xf}}{S_{xf} + S_{yf}} > 1.
\]  \hspace{1cm} (17)

This weighting factor depends on the similarity in spatial domain between the source and fused image. The higher the similarity between the source image and fused image, the larger the corresponding weighting factor.

**Pseudo-code of \(Q_C\) metric** A pseudo-code for the implementation of the \(Q_C\) metric algorithm, is shown in Algorithm 10. At first, the pseudo-code of the implementation of the weighting factor sim, is presented
Algorithm 9: Pseudo-code of the weight sim algorithm.

\textbf{input} : source images $x$, $y$ and fused image $f$, window size $w_{\text{size}}$.

\textbf{output}: $\text{sim}$ weighting factor, a map of similarity in spatial domain between the input and the fused image.

\begin{align*}
s_{xf} & \leftarrow \text{covariance}(x, f, w_{\text{size}}) \\
s_{yf} & \leftarrow \text{covariance}(y, f, w_{\text{size}}) \\
\text{sim} & \leftarrow 1
\end{align*}

forall the sliding window $w, w \in W$ do

if $s_{xf}(w) + s_{yf}(w) = 0$ then

\begin{align*}
\text{sim}(w) & \leftarrow 0
\end{align*}

else

\begin{align*}
\text{sim}(w) & \leftarrow s_{xf}(w) / (s_{xf}(w) + s_{yf}(w)) \\
\text{if} \quad \text{sim}(w) > 1 \quad \text{then}
\begin{align*}
\text{sim}(w) & \leftarrow 1 \\
\text{if} \quad \text{sim}(w) < 0 \quad \text{then}
\begin{align*}
\text{sim}(w) & \leftarrow 0
\end{align*}
end
end

end

Algorithm 10: Pseudo-code of the $Q_{C}$ fusion quality metric algorithm.

\textbf{input} : source images $x$, $y$ and fused image $f$, window size $w_{\text{size}}$.

\textbf{output}: $q_{C}$, the fusion quality metric value and the $q_{C, \text{map}}$, the quality assessment map of the fused image.

\begin{align*}
q_{\text{map}_{xf}} & \leftarrow Q(x, f, w_{\text{size}}) \quad \triangleright \text{Algorithm 1.} \\
q_{\text{map}_{yf}} & \leftarrow Q(y, f, w_{\text{size}}) \quad \triangleright \text{Algorithm 1.} \\
\text{sim} & \leftarrow \text{weight}_{\text{sim}}(x, y, f, w_{\text{size}}) \quad \triangleright \text{Algorithm 9.} \\
q_{C, \text{map}} & \leftarrow \text{sim} \cdot q_{\text{map}_{xf}} + (1 - \text{sim}) \cdot q_{\text{map}_{yf}} \\
q_{C} & \leftarrow \frac{1}{|W|} \sum_{w \in W} q_{C, \text{map}}(w) \quad \triangleright \text{Computes the } Q_{C} \text{ metric (16).}
\end{align*}
2.2.3 Yang’s Metric ($Q_Y$)

Yang et al. proposed another way to use SSIM for fusion assessment [13]. In this metric, the local structural similarity between source images $SSIM(x, y|w)$ is used as a match measure. For those windows whose $SSIM(x, y|w)$ is equal to or larger than a given threshold (to distinguish redundant information), the weighted average of $SSIM(x, f|w)$ and $SSIM(y, f|w)$ is taken, otherwise, the larger of the two is taken.

$$Q_Y(x, y, f|w) = \begin{cases} 
\lambda(w)SSIM(x, f|w) + (1 - \lambda(w))SSIM(y, f|w), \\
\text{if } SSIM(x, y|w) \geq 0.75, \\
\max\{SSIM(x, f|w), SSIM(y, f|w)\}, \\
\text{if } SSIM(x, y|w) < 0.75.
\end{cases}$$  \hspace{1cm} (18)

The local weight $\lambda(w)$ is defined in (10).

The fusion performance metric, $Q_Y$, is calculated averaging all values of $Q_Y(x, y, f|w)$ over the whole image,

$$Q_Y(x, y, f) = \frac{1}{|W|} \sum_{w \in W} Q_Y(x, y, f|w).$$  \hspace{1cm} (19)

**Pseudo-code of $Q_Y$ metric**  A pseudo-code for the implementation of the $Q_Y$ fusion metric algorithm, is given in Algorithm 10.

**Algorithm 10:** Pseudo-code of the $Q_Y$ fusion quality metric algorithm.

```
input : source images $x, y$ and fused image $f$, constants $k = [k_1, k_2]$, local window for statistics $window$ and dynamic range of the images $L$.
output: $q_Y$ fusion quality metric value and the $q_Y$-map, the quality assessment map the fused image.

ssim_map_xf ← SSIM(x, f, k, window, L) \hspace{1cm} ▶ Algorithm 2.

ssim_map_yf ← SSIM(y, f, k, window, L) \hspace{1cm} ▶ Algorithm 2.

ssim_map_xy ← SSIM(x, y, k, window, L) \hspace{1cm} ▶ Algorithm 2.

$\lambda \leftarrow$ weight $\lambda(x, y, w\_size)$ \hspace{1cm} ▶ Algorithm 4.

condition ← ssim_map_xy

forall the sliding window $w, w \in W$ do

    if condition $(w) \geq 0.75$ then
        $q_Y\_map(w) \leftarrow \lambda(w) \cdot ssim\_map_xf(w) + (1 - \lambda(w)) \cdot ssim\_map_yf(w)$
    else
        $q_Y\_map(w) \leftarrow \max\{ssim\_map_xf(w), ssim\_map_yf(w)\}$

$q_Y \leftarrow \frac{1}{|W|} \sum_{w \in W} q_Y\_map(w)$ Computes the $Q_Y$ metric at the window $w$ (18).

Computes the $Q_Y$ metric (19).
```

In order to obtain a image fusion quality measure consistent with human visual evaluations, Yang et al. set the $SSIM$ constants $k_1 = k_2 = 7.8 \times 10^{-9}$.

2.2.4 Codispersion Fusion Quality Metric ($CQ_M$)

Following the structure of Piella’s metric (11), Pistonesi et al. [8] introduced an objective measure for image fusion. This fusion measure, labeled $CQ_M$, is based on a modification of the $CQ$ index, called $CQ_{max}$ index,
\[ CQ_M(x, y, f) = \sum_{w \in W} c(w) [\lambda(w) CQ_{\text{max}}(x, f|w) + (1 - \lambda(w)) CQ_{\text{max}}(y, f|w)], \]  
\[ (20) \]

with
\[ CQ_{\text{max}}(x, y|w) = \max_{\{h \in H : p(h) \geq p_0\}} CQ(x, y, h|w), \]  
\[ (21) \]

where \( p(h) \) is the proportion of the pixels in the image corresponding to the direction \( h \) in the window \( w \), \( p_0 \) is the threshold, denoting the minimum proportion of pixels that is necessary to capture spatial information in different directions and \( H \) the family of all directions in a window \( w \) (see Figure 8 in Appendix A.1).

In the same sense that the structural similarity indexes were defined in Section 2, the maximum codispersion quality index, \( CQ_{\text{max}} \), is determined by averaging all \( CQ(x, y, h|w) \) local maximum quality indexes for all the windows \( w \in W \),
\[ CQ_{\text{max}} = \frac{\sum_{w \in W} CQ_{\text{max}}(x, y|w)}{|W|}. \]  
\[ (22) \]

In each evaluated window \( w \), the \( CQ_{\text{max}}(x, y|w) \) index, is defined as the maximum value of \( CQ(x, y, h|w) \). This implies that \( CQ_{\text{max}} \) seeks the direction \( h \in H \), that maximizes the \( CQ \) in the window \( w \). Note that this direction may not be unique. Its dynamic range is \([-1, 1]\).

The \( CQ_M \) fusion metric calculates the maximum information transferred from source images into the fused image, by means of the maximum codispersion index.

**Pseudo-code of the pixels proportion \( p \), \( CQ_{\text{max}} \) index and \( CQ_M \) metric**

Firstly, a pseudo-code for the implementation of the local proportion \( p \) of pixels corresponding to the direction \( h \), is presented (Algorithm 12). Then, Algorithm 13, gives a pseudo-code for the implementation of the \( CQ_{\text{max}} \) index. Finally, Algorithm 14 shows the pseudo-code for the implementation of the \( CQ_M \) metric.

**Algorithm 12:** Pseudo-code of the \( p \) proportion algorithm.

```
input : direction \( h = (h_1, h_2) \), window size \( w_{\text{size}} = (w_1, w_2) \).
output: \( p \) proportion of the pixels corresponding to the specific direction \( h \) in the window \( w \).

\[ h_1 \leftarrow \text{abs}(h_1) \quad \text{Computes the absolute value of the coordinates of the direction } h. \]

\[ h_2 \leftarrow \text{abs}(h_2) \]

if \((h_1 > \frac{w_1}{2}) \text{ or } (h_2 > \frac{w_2}{2})\) then

\[ p \leftarrow (w_1 - h_1) \cdot (w_2 - h_2) \cdot 2 \]

else

\[ p \leftarrow w_1 \cdot w_2 - 2 \cdot h_1 \cdot h_2 \]

\[ p \leftarrow \frac{p}{w_1 \cdot w_2} \quad \text{Computes the proportion of the pixels corresponding to the direction } h. \]
```

Note that all these fusion performance measures have a dynamic range of \([-1, 1]\). The closer the value to 1, the higher the quality of the composite image.

3 Examples

In this section, we present examples of the implemented fusion metrics applied to three different pairs of images used in distinct applications: infra-red (IR) and visual, computed tomography (CT)
**Algorithm 13:** Pseudo-code of $CQ_{\text{max}}$ quality index algorithm.

**input:** original and test image signals $x$ and $y$, window size $w_{\text{size}}$ and threshold $p_0$.

**output:** $cq$ index value between two images and $cq_{\text{map}}$, quality map of the test image.

$k \leftarrow 0$

for all the $h \in H$ do

    if proportion $(h, w_{\text{size}}) \geq p_0$ then

        Computes the proportion of the pixels corresponding to the direction $h$ in the window $w$.

        $k \leftarrow k + 1$

        $cq_{\text{maps}}(k) \leftarrow CQ(x, y, h, w_{\text{size}})$  \textit{▷ Algorithm 3.}

$cq_{\text{map\_max}} \leftarrow \text{max\_similarity}(cq_{\text{maps}})$  \text{Computes for each window the maximum value of all $cq_{\text{maps}}$.}

$cq_{\text{max}} \leftarrow \frac{1}{|W|} \sum_{w \in W} cq_{\text{map\_max}}(w)$  \text{Computes the $CQ_{\text{max}}$ index (22).}

**Algorithm 14:** Pseudo-code of $CQ_M$ fusion quality metric algorithm.

**input:** source images $x$, $y$ and fused image $f$, window size $w_{\text{size}}$ and threshold $p_0$.

**output:** $cq_M$, the fusion quality metric value and $cq_{M\_map}$, the quality assessment map of the fused image.

$\lambda \leftarrow \text{weight}_\lambda(x, y, w_{\text{size}})$  \textit{▷ Algorithm 4.}

$c \leftarrow \text{weight}_c(x, y, w_{\text{size}})$  \textit{▷ Algorithm 6.}

cq$\_max$$_{\text{map}}_x \leftarrow CQ_{\text{max}}(x, f, w_{\text{size}})$  \textit{▷ Algorithm 13.}

cq$\_max$$_{\text{map}}_y \leftarrow CQ_{\text{max}}(y, f, w_{\text{size}})$  \textit{▷ Algorithm 13.}

cq$M\_map \leftarrow c \cdot [\lambda \cdot cq_{\text{max}}_{\text{map}}_x + (1 - \lambda) \cdot cq_{\text{max}}_{\text{map}}_y]$  \text{Computes the $CQ_M$ metric (20).}

cq$M \leftarrow \sum_{w \in W} cq_{M\_map}(w)$
and magnetic resonance imaging (MRI) and multifocus images. In all of them, were evaluated the following image fusion algorithms: Laplacian Pyramid (LP) [14], Ratio Pyramid (RP) [14], Discrete Wavelet Transform (DWT) [3], and Shift Invariant DWT (SIDWT) [9], the performances of which were subjectively tested and accepted in the literature. For simulation of these methods, the “Image Fusion Toolbox”, provided by Rockinger³, is used.

**Parameter Setting:** For the four image fusion methods, the approximation coefficients of the two input images averaged and the larger absolute values of the high subbands are selected. In the first example, we performed a 3-level decomposition and in the second and third, a 4-level decomposition was used. In order to ensure fairness in comparison, in the algorithms of the fusion quality metrics we used the same setting that appears in [7],[1],[13] and [8]. For Piella’s, Cvejic’s and Codispersion Fusion Quality metrics we used the same window size, 8 × 8 pixels. Also, for the two versions of the edge-dependent fusion quality $Q_{E_1}$ and $Q_{E_2}$ measures, we took as contribution parameter of the edge images: $\alpha = 1$ and $\alpha = 0.5$, respectively. For the proportion parameter of Codispersion Fusion Quality metric, we set $p_0 = 0.75$ and for Yang’s metric, the constants $C_1 = C_2 = 2 \times 10^{-16}$, the dynamic range of the images, $L = 255$ and the $w$ window size used was 7 × 7 pixels.

**First Example:** The “TNO UN Camp” database used as source images consists of 32 sets of infrared (IR) and visual images (V) (see Figure 2).

![Figure 2](image1.png)

Figure 2: An image of the “TNO UN Camp” database: (a) IR image (b) V image, and (c)-(f) fused image obtained by LP, RP, DWT and SIDWT methods.

**Second Example:** For the following example, a magnetic resonance imaging (MRI) and computed tomography (CT) image are used (see Figure 3). A visual inspection of the fusion performance metrics are represented in Figure 4. These maps allow to visualize the local information about the spatially varying quality of the fused image. According to these maps, brightness indicate the magnitude of the local fusion metric.

**Third Example:** In this example, a pair of multi-focus images are used (see Figure 5). The quality assessment maps are shown in Figure 6.

³http://www.metapix.de/toolbox.htm/
### Table 2: Objective evaluations of different image fusion metrics for the fused images in “TNO UN Camp” database (Mean ± Standard Deviation).

<table>
<thead>
<tr>
<th>Metrics</th>
<th>LP</th>
<th>RP</th>
<th>DWT</th>
<th>SIDWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_S$</td>
<td>0.6730 ± 0.0072</td>
<td>0.5925 ± 0.0108</td>
<td>0.6374 ± 0.0084</td>
<td>0.6851 ± 0.0076</td>
</tr>
<tr>
<td>$Q_W$</td>
<td>0.7576 ± 0.0062</td>
<td>0.6289 ± 0.0047</td>
<td>0.7103 ± 0.0053</td>
<td>0.7511 ± 0.0050</td>
</tr>
<tr>
<td>$Q_{E_1}$</td>
<td>0.5338 ± 0.0130</td>
<td>0.3213 ± 0.0064</td>
<td>0.4716 ± 0.0104</td>
<td>0.5366 ± 0.0105</td>
</tr>
<tr>
<td>$Q_{E_2}$</td>
<td>0.7305 ± 0.0090</td>
<td>0.5668 ± 0.0056</td>
<td>0.6867 ± 0.0076</td>
<td>0.7325 ± 0.0072</td>
</tr>
<tr>
<td>$Q_C$</td>
<td>0.7172 ± 0.0039</td>
<td>0.6488 ± 0.0061</td>
<td>0.6810 ± 0.0050</td>
<td>0.7228 ± 0.0044</td>
</tr>
<tr>
<td>$Q_Y$</td>
<td>0.7806 ± 0.0048</td>
<td>0.7044 ± 0.0071</td>
<td>0.7452 ± 0.0060</td>
<td>0.7877 ± 0.0052</td>
</tr>
<tr>
<td>$CQ_M$</td>
<td>0.8332 ± 0.0033</td>
<td>0.7354 ± 0.0072</td>
<td>0.8108 ± 0.0034</td>
<td>0.8417 ± 0.0033</td>
</tr>
</tbody>
</table>

### Table 3: The correlation matrix of image fusion metrics for the fused images in “TNO UN Camp” database. The best Kendall $\tau$ correlation coefficient values are highlighted in bold typeface.

<table>
<thead>
<tr>
<th>Kendall Correlation</th>
<th>$Q_S$</th>
<th>$Q_W$</th>
<th>$Q_{E_1}$</th>
<th>$Q_{E_2}$</th>
<th>$Q_C$</th>
<th>$Q_Y$</th>
<th>$CQ_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_S$</td>
<td>1</td>
<td>0.594</td>
<td>0.649</td>
<td>0.649</td>
<td><strong>0.937</strong></td>
<td><strong>0.954</strong></td>
<td>0.786</td>
</tr>
<tr>
<td>$Q_W$</td>
<td>1</td>
<td><strong>0.847</strong></td>
<td><strong>0.847</strong></td>
<td>0.605</td>
<td>0.594</td>
<td>0.706</td>
<td></td>
</tr>
<tr>
<td>$Q_{E_1}$</td>
<td>1</td>
<td>1</td>
<td>0.658</td>
<td>0.642</td>
<td>0.781</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q_{E_2}$</td>
<td>1</td>
<td>1</td>
<td>0.658</td>
<td>0.642</td>
<td>0.781</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q_C$</td>
<td>1</td>
<td><strong>0.969</strong></td>
<td>0.771</td>
<td>1</td>
<td>0.770</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q_Y$</td>
<td>1</td>
<td>0.770</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$CQ_M$</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 4: Objective evaluations of different image fusion metrics for a fused medical image.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>LP</th>
<th>RP</th>
<th>DWT</th>
<th>SIDWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_S$</td>
<td>0.6168</td>
<td>0.6020</td>
<td>0.6090</td>
<td>0.6506</td>
</tr>
<tr>
<td>$Q_W$</td>
<td>0.8089</td>
<td>0.6319</td>
<td>0.7314</td>
<td>0.7780</td>
</tr>
<tr>
<td>$Q_{E_1}$</td>
<td>0.6565</td>
<td>0.3217</td>
<td>0.5598</td>
<td>0.6429</td>
</tr>
<tr>
<td>$Q_{E_2}$</td>
<td>0.8102</td>
<td>0.5672</td>
<td>0.7482</td>
<td>0.8018</td>
</tr>
<tr>
<td>$Q_C$</td>
<td>0.6247</td>
<td>0.6053</td>
<td>0.6190</td>
<td>0.6587</td>
</tr>
<tr>
<td>$Q_Y$</td>
<td>0.6874</td>
<td>0.6182</td>
<td>0.6368</td>
<td>0.6692</td>
</tr>
<tr>
<td>$CQ_M$</td>
<td>0.8391</td>
<td>0.6903</td>
<td>0.7718</td>
<td>0.8169</td>
</tr>
</tbody>
</table>
Figure 3: Medical images. (a) a CT source image (b) a MRI source image, and (c)-(f) the fused image obtained by different image fusion methods: LP, RP, DWT and SIDWT, respectively.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LP</td>
</tr>
<tr>
<td>$Q_S$</td>
<td>0.8220</td>
</tr>
<tr>
<td>$Q_W$</td>
<td>0.9272</td>
</tr>
<tr>
<td>$Q_{E_1}$</td>
<td>0.8188</td>
</tr>
<tr>
<td>$Q_{E_2}$</td>
<td>0.9048</td>
</tr>
<tr>
<td>$Q_C$</td>
<td>0.8284</td>
</tr>
<tr>
<td>$Q_Y$</td>
<td>0.8816</td>
</tr>
<tr>
<td>$CQ_M$</td>
<td>0.9451</td>
</tr>
</tbody>
</table>

Table 5: Objective evaluations of different image fusion metrics for a fused clock image.
3.1 Results Analysis

In the three different applications, the fusion metrics present similar behaviour, assigning the highest values to the LP and SIDWT methods, followed by DWT and the worst values correspond to the RP method (see Tables 2, 4 and 5). They have a coherent performance with the perceptual evaluations. Resulting images in Figures 2, 3 and 5 clearly certify this.

As seen in the first example, the $CQ_M$ has the lowest standard deviation for LP, SIDWT and DWT fusion methods (see Table 2). This fact demonstrates the good stability of the $CQ_M$ metric for these specific fusion methods [4].

The evaluation results of “TNO UN Camp” images, in Table 3, reveal that the four largest values of Kendall $\tau$ rank correlation coefficient [2] correspond to the correlation of $Q_Y$ and $Q_C$, $Q_Y$ and $Q_S$, $Q_C$ and $Q_S$ and $Q_W$ with $Q_{E_1}$ and $Q_{E_2}$, respectively. These outcomes are consistent with those obtained by Lui et al. [4]. Also, $CQ_M$ shows reasonable agreement with all of them.

The corresponding fusion evaluation results given in Tables 4 and 5 show that the LP method has higher values of $Q_{E_1}$ and $Q_{E_2}$ (metrics that take into account edge information) than the SIDWT method. This is due to its high-pass characteristic and to its known ability to preserve edges and reduce the ringing artifacts around them [7]. There is a stronger separation between the good results (LP, SIDWT and DWT) and the bad results (RP).

From the examples we can appreciate that the fusion metrics provide a relative assessment on the quality of the fused image, depending on the goals and operational requirements of the application of interest. For a general discussion about the fusion metrics, see Lui et. al [4].
Figure 5: Clock images. (a) a focus on right source clock image (b) a focus on left source clock image, and (c)-(f) the fused image obtained by: LP, RP, DWT and SIDWT methods, respectively.
**A Appendix**

**A.1 Implementation Details**

In this subsection, we present some details about directions \( h \) within a window \( w \), considered in \( CQ \) and \( CQ_{max} \) implementation algorithms.

We assume that a window \( w \) is a two-dimensional rectangular grid consisting of \( m \times n \) pixels, \( m, n \in \mathbb{N} \), evenly spaced. The pixel site is denoted by \( s = (i, j), i = 1, \ldots, m; j = 1, \ldots, n \). For our implementation, the window can be characterized by a matrix structure, where the row is the first pixel coordinate and the column, the second (see Figure 7).

Let \( H \) be the set of all directions of a window \( w \) of an image. In order to facilitate algorithmic implementation, we considered \( H \) as the union of two disjoint sets \( H_1 (H_1^-) \) and \( H_2 (H_2^-) \),

\[
H_1 = \{ h = (h_1, h_2) \in H | 0 \leq h_1 \leq (m-1), 1 \leq h_2 \leq (n-1), h_1, h_2 \in \mathbb{Z} \},
\]

\[
(H_1^- = \{ h = (h_1, h_2) \in H | -(m-1) \leq h_1 \leq 0, -(n-1) \leq h_2 \leq -1, h_1, h_2 \in \mathbb{Z} \})
\]

and

\[
H_2 = \{ h = (h_1, h_2) \in H | 1 \leq h_1 \leq (m-1), -(n-1) \leq h_2 \leq 0, h_1, h_2 \in \mathbb{Z} \},
\]

\[
(H_2^- = \{ h = (h_1, h_2) \in H | -(m-1) \leq h_1 \leq -1, 0 \leq h_2 \leq (n-1), h_1, h_2 \in \mathbb{Z} \})
\]

Figure 8 illustrates the \( H_1 \) and \( H_2 \) sets considered in a \( 8 \times 8 \) window size.
Figure 7: Matrix structure of a window $w$ (size $8 \times 8$ pixels) for algorithm implementation.

Figure 8: Directions $h$ of a window $w$ for algorithm implementation. (a) $H_1$ directions set (b) $H_2$ directions set. Proportion of the pixels in the window corresponding to the direction $h$. Darkness indicates more pixels proportion.
An example of four different directions $h$ is represented in Figure 8.

Figure 9: Example of four different directions $h$. First column, a horizontal direction (—), a diagonal direction (/), another diagonal direction (\) and a vertical direction (|). Second column, $x_s$, coloured in red and $x_{s+h}$, in blue. Third column, the pixels proportion corresponding to these directions $h$: 1, 0.97, 0.75 and 0.5, respectively. Darkness indicates more pixels proportion. Fourth column, the $h = (0, 2)$, $h = (1, -1)$, $h = (2, 4)$ and $h = (6, 0)$ directions, respectively.

It can be proved that if $h \in H_1$, $\exists h' \in H_1^-$ such that $\hat{\rho}(h) = \hat{\rho}(h')$ (for example, let $h = (1, 1) \in H_1$ and $h' = (-1, -1) \in H_1^-$). Similarly, if $h \in H_2$, $\exists h' \in H_2^-$ such that $\hat{\rho}(h) = \hat{\rho}(h')$ (for instance, let $h = (1, -1) \in H_2$ and $h' = (-1, 1) \in H_2^-$). Figure 10 demonstrates these results visually. For these reasons, we take into account the sets $H_1$ and $H_2$. 

23
Figure 10: Example of equivalent directions $h$. (a) $h = (1, 1) \in H_1$ and $h' = (-1, -1) \in H_1^-$ (b) $h = (1, -1) \in H_2$ and $h' = (-1, -1) \in H_2^-$. 

Acknowledgments

S. Pistonesi and J. Martinez would like to thank SGCyT-UNS for financial assistance, S. M. Ojeda supported by SeCyT-UNC, and R. Vallejos by AC3E under grant No FB-0008. Part of this article was written while S. Pistonesi and J. Martinez were visiting the Centre de Mathématiques et de Leurs Applications (CMLA) of the École Normale Supérieure de Cachan, Cachan, France. S. Pistonesi and J. Martinez are also grateful to Miguel Colom, Enric Meinhardt-Llopis and Jean-Michel Morel for their hospitality and kindness. The authors would like to thank the IPOL editors and anonymous reviewers for their valuable insights and suggestions that helped to improve and clarify the manuscript.

Image Credits

All images by the authors except:

from the TNO UN Camp database

provided by Rockinger

References


