FUSION OF OUT OF FOCUS IMAGES USING PRINCIPAL COMPONENT ANALYSIS AND SPATIAL FREQUENCY

V.P.S. Naidu* and J.R. Raol*

Abstract
Details of principal component analysis and spatial frequency are presented. Two image fusion architectures are developed to fuse multi focused images and their performance is compared. In first architecture source images to be fused are considered as whole in the fusion process. In second architecture the source images to be fused are divided into blocks and then used in the fusion process. Overall SF shows slightly better performance. Block based image fusion scheme (second architecture) shows superior performance. This architecture is very simple and can be used in real time applications.

Keywords: PCA, Spatial Frequency, Image Fusion, Fusion Performance Index

Introduction
Single imaging sensor cannot provide complete information about a situation. Multi imaging sensor fusion would provide better or enhanced information about the situation. Off late multi sensor fusion has emerged as an innovative and promising research area. Sensor fusion could take place at signal level, pixel level, feature level and symbol level [1]. Multi sensor image fusion (MSIF) is a technique for merging the registered multi sensor images to enhance the image information. The fused image has improved contrast and it could be easy for the user to detect, recognize and identify the targets and increase users situational awareness [2]. The fusion of images is of vast significance in numerous applications viz. microscopic imaging, medical imaging, remote sensing, robotics and computer vision. Some common requirements would be imposed on the fusion results: (1) fused image should preserve all relevant information contained in the source images, (2) fusion process would not introduce any artifacts or inconsistencies which would amuse the human observer or following processing stages and (3) irrelevant features and noise should be suppressed in the fused image to a maximum extent [3]. When image fusion is done at pixel level the source images are combined without any pre-processing. The pixel level fusion (also called image level fusion) algorithms vary from simple image averaging to very complex algorithms. The simplest MSIF is to take the average of the grey level source images pixel by pixel. This technique would produce several undesired

Nomenclature
\[ \lambda = \text{eigenvalue} \]
\[ CF = \text{column frequency} \]
\[ h_i = \text{normalized histogram} \]
\[ I_1 \text{ and } I_2 = \text{source images to be fused} \]
\[ I_f = \text{fused image} \]
\[ I_r = \text{reference or ground truth image} \]
\[ I_{1k} \text{ and } I_{2k} = k^{th} \text{ block source images} \]
\[ L = \text{number of gray levels} \]
\[ NSF_i = i^{th} \text{ principal component} \]
\[ NPC_i = i^{th} \text{ principal component} \]
\[ RF = \text{row frequency} \]
\[ SF = \text{spatial frequency} \]
\[ th = \text{threshold} \]
\[ V = \text{orthogonal projection matrix} \]
\[ X = \text{random vector} \]
\[ DCT = \text{Discrete Cosine Transform} \]
\[ FFT = \text{Fast Fourier Transform} \]
\[ MSIF = \text{Multi Sensor Image Fusion} \]
\[ PCA = \text{Principal Component Analysis} \]
\[ PFE = \text{Percentage Fit Error} \]
\[ PSNR = \text{Peak Signal to Noise Ratio} \]
\[ RMSE = \text{Root Mean Square Error} \]
\[ SD = \text{Standard Deviation} \]
\[ SF = \text{Spatial Frequency} \]

* Scientist, Multi Sensor Data Fusion Laboratory, Flight Mechanics and Control Division, National Aerospace Laboratories, Post Box No. 1779, Bangalore-560 017, India, Email : vpsnaidu@gmail.com

Manuscript received on 06 Nov 2007; Paper reviewed, revised and accepted on 13 May 2008
effects and reduced feature contrast. To overcome this problem, multi-scale transforms, such as wavelets, Laplacian pyramids, morphological pyramid and gradient pyramid have been proposed. Multi-resolution wavelet transforms could provide good localization in both spatial and frequency domains. Discrete wavelet transform would provide directional information in decomposition levels and contain unique information at different resolutions [4, 5].

In some situations the objects in the scene would be at different distances from the imaging sensor. Inexpensive sensor would not focus everywhere. If one object is in focus then another will be out of focus. Fusing these images there would be little out of focus in the fused image [6]. In this paper, an efficient image level fusion algorithms based on principal component analysis and spatial frequency are proposed. Two different fusion architectures are evaluated. In former architecture source images to be fused are considered as whole in the fusion process. There would be some discrepancy in the fused image that local variations (level of focus) in the source images are not considered. In second architecture the source images are decomposed into small blocks and these blocks are used in the image fusion process. In this the discrepancy would be reduced since the local variations are considered in the fusion process. In Ref. 6, block size and threshold are user defined parameters utilized in the second architecture. It could be very hard to choose the threshold to get optimal fusion results. In this paper, a modified algorithm that computes normalized spatial frequencies is introduced in the fusion process. Since the spatial frequencies of source images are normalized, the user can choose the threshold in between 0 to 0.5. Similar methodology is adopted for principal component analysis based image fusion algorithm. Since choosing of block size and threshold is complex, a simple solution is provided to get the optimal fusion results at the cost of execution time. The performances of these algorithms are evaluated with performance evaluation metrics.

One of the important prerequisites to be able to apply fusion techniques to source images is the image registration i.e., the information in the source images needed to be adequately aligned and registered prior to fusion of the images. In this paper, it is assumed that the source images are already registered.

**Fusion Algorithms**

The details of principal component analysis and spatial frequency computations are described in this section.

**Principal Component Analysis**

Principal component analysis (PCA) involves a mathematical procedure that transforms a number of correlated variables into a number of uncorrelated variables called principal components. PCA computes a compact and optimal description of the data set. The first principal component accounts for as much of the variance in the data as possible and each succeeding component accounts for as much of the remaining variance as possible. First principal component is taken to be along the direction with the maximum variance. The second principal component is constrained to lie in the subspace perpendicular of the first. Within this subspace, this component points the direction of maximum variance. The third principal component is taken in the maximum variance direction in the subspace perpendicular to the first two and so on. PCA is also called as Karhnen-Loeve transform or the Hotelling transform. PCA does not have a fixed set of basis vectors like FFT, DCT and wavelet etc. and its basis vectors depend on the data set. PCA is used extensively in image compression and image classification.

Let \(X\) be a d-dimensional random vector and assume it to have zero empirical mean. Orthonormal projection matrix \(V\) would be such that \(Y = V^T X\) with the following constraint. The covariance of \(Y\), i.e. \(\text{cov}(Y)\) is a diagonal and inverse of \(V\) is equivalent to its transpose. Using matrix algebra [7],

\[
\]

Multiplying both sides of (Eq.1) by \(V\), one would get

\[
V \text{cov}(Y) = V V^T \text{cov}(X)
\]

One could write \(V = [V_1, V_2, \ldots, V_d]\) and \(\text{cov}(Y)\) in the diagonal from as :
\[
\begin{bmatrix}
\lambda_1 & 0 & \ldots & 0 & 0 \\
0 & \lambda_2 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & \lambda_{d-1} & 0 \\
0 & 0 & \ldots & 0 & \lambda_d
\end{bmatrix}
\]  
(3)

By substituting (Eq.3) into (Eq.2):

\[
\begin{bmatrix}
\lambda_1 V_1, \lambda_2 V_2, \ldots, \lambda_d V_d
\end{bmatrix} = \begin{bmatrix}
cov(X) V_1, cov(X) V_2, \ldots, cov(X) V_d
\end{bmatrix}
\]  
(4)

This could be rewritten as:

\[\lambda_i V_i = cov(X) V_i\]  
(5)

where \(i = 1, 2, \ldots, d\)

\(V_i\) is an eigenvector of \(cov(X)\)

### PCA Algorithm

Let the source images (images to be fused) be arranged in two column vectors. The steps followed to project this data into two dimensional subspaces are:

- Organize the data into column vectors. The resulting matrix \(Z\) has a dimension of \(n \times 2\).

- Compute the empirical mean along each column. The empirical mean vector \(M\) has a dimension of \(2 \times 1\).

- Subtract the empirical mean vector \(M\) from each column of the data matrix \(Z\). The resulting matrix \(X\) is of dimension \(n \times 2\).

- Find the covariance matrix \(C\) of \(S\) i.e. \(C = X^T X\).

- Compute the eigenvectors \(V\) and eigenvalue \(D\) of \(C\) and sort them by decreasing eigenvalue. Both \(V\) and \(D\) are of dimension \(2 \times 2\).

- Consider the first column of \(V\) which corresponds to larger eigenvalue to compute the principal components \(NPC_1\) and \(NPC_2\) as:

\[NPC_1 = V(1) \sum V\] and \[NPC_2 = V(2) \sum V\]  
(6)

### Image Fusion by PCA

The information flow diagram of PCA based weighted average image fusion algorithm (first architecture) is shown in Fig.1a. The source images (images to be fused) \(I_1\) and \(I_2\) are arranged in two column vectors and their empirical means are subtracted. The resulting vector has a dimension of \(n \times 2\), where \(n\) is length of the each image vector. Eigenvector and eigenvalues for this resulting vector are computed and the eigenvectors corresponding to the larger eigenvalue are obtained. The principal components \(NPC_1\) and \(NPC_2\) (i.e. \(NPC_1 + NPC_2 = 1\)) using eq.6 are computed from the obtained eigenvector. The fused image is obtained by:

\[I_f = NPC_1 I_1 + NPC_2 I_2\]  
(7)

The information flow diagram of PCA based block image fusion algorithm (second architecture) is shown in Fig.1b. The input images are decomposed into blocks (\(I_{1k}\) and \(I_{2k}\)) of size \(m \times n\). Where \(I_{1k}\) and \(I_{2k}\) denotes the \(k\)th blocks of \(I_1\) and \(I_2\) respectively. Principal components for each block using Eq.6 are computed. Let the principal components corresponding \(k\)th blocks be \(NPC_{1k}\) and \(NPC_{2k}\) (i.e. \(NPC_{1k} + NPC_{2k}\)). The fusion of \(k\)th block of the fused image is:

![Fig.1a Information flow diagram of PCA based weighted image fusion algorithm](image1a.png)

![Fig.1b Information flow diagram of PCA based block image fusion algorithm](image1b.png)
\[
I_{f_k} = \begin{cases} 
I_{1k} & \text{NPC}_{1k} > \text{NPC}_{2k} + th \\
I_{2k} & \text{NPC}_{1k} < \text{NPC}_{2k} - th \\
\frac{I_{1k} + I_{2k}}{2} & \text{otherwise}
\end{cases}
\] (8)

where \( th \) : user defined threshold

\[
\frac{I_{1k} + I_{2k}}{2} : \text{gray level averaging of corresponding pixels}
\]

**Spatial Frequency**

Spatial frequency measures the overall information level in an image \([6,8]\). The spatial frequency for a given image \( I \) of dimension \( M \times N \) is defined as follows:

Row Frequency :

\[
RF = \sqrt{\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=1}^{N-1} [I(i,j) - I(i,j-1)]^2} \tag{9}
\]

Column Frequency :

\[
CF = \sqrt{\frac{1}{MN} \sum_{j=0}^{N-1} \sum_{i=1}^{M-1} [I(i,j) - I(i-1,j)]^2} \tag{10}
\]

Spatial Frequency :

\[
SF = \sqrt{RF^2 + CF^2} \tag{11}
\]

where

\( M \) = number of rows; \( N \) = number of columns

\((i, j) = \text{pixel index} ; I = \text{given image} \\
I(i, j) = \text{gray value at pixel} (i, j)\)

**Image Fusion by SF**

The information flow diagram of SF based weighted image fusion algorithm (first architecture) is shown in Fig.2a. Denote the \( SF_1 \) and \( SF_2 \) spatial frequencies of input images \( I_1 \) and \( I_2 \) respectively. The computed spatial frequencies are then normalized as:

\[
NSF_1 = \frac{SF_1}{SF_1 + SF_2} \quad \text{and} \quad NSF_2 = \frac{SF_2}{SF_1 + SF_2} \tag{12}
\]

The fused image is obtained by

\[
I_f = NSF_1 I_1 + NSF_2 I_2 \tag{13}
\]

The information flow diagram of SF based block image fusion algorithm (second architecture) is shown in Fig.2b. The input images are decomposed into blocks \((I_{1k} \text{ and } I_{2k})\). Normalized spatial frequencies for each block using Eq.12 are computed. Denote the normalized spatial frequencies of \( I_{1k} \) and \( I_{2k} \) by \( NSF_{1k} \) and \( NSF_{2k} \) respectively. The fusion of the \( k^{th} \) block of the fused image is:

\[
I_{f_k} = \begin{cases} 
I_{1k} & NSF_{1k} > NSF_{2k} + th \\
I_{2k} & NSF_{1k} < NSF_{2k} - th \\
\frac{I_{1k} + I_{2k}}{2} & \text{otherwise}
\end{cases} \tag{14}
\]

**Majority Filter**

In block image fusion algorithm, majority filter is used to avoid the artifacts in fused image caused by the fusion rules. If the center block comes from \( I_1 \) and the surroundings blocks are from \( I_2 \) then the centre block will be replaced by the block from \( I_2 \) and vice versa \([7]\).
The majority filter with the order of 3 x 3 is used in this study.

**Example:** Majority filter working principal is demonstrated here. Denote $a$ and $b$ as the block images coming from $I_1$ and $I_2$ respectively. A block window in the thematic map is shown in left side. One can see that the blocks $(a)$ from $I_1$ is six times and blocks $(b)$ from $I_2$ is three times. The majority filter replaces the centre block with the block coming from $I_1$, since the majority of neighboring blocks are coming from the $I_1$.

---

**Performance Evaluation**

**With Reference Image**

When the reference image is available, the performance of image fusion algorithms can be evaluated using the following metrics.

- **Root Mean Square Error (RMSE)** [9]

  This metric is computed as the root mean square error of the corresponding pixels in the reference image $I_r$ and the fused image $I_f$. This metric will be nearly zero when the reference and fused images are similar. This will increase when the dissimilarity increases.

  $$RMSE = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (I_r(i,j) - I_f(i,j-1))^2}$$

- **Percentage Fit Error (PFE)** [9]

  This metric is computed as the norm of the difference between the corresponding pixels of reference and fused image to the norm of the reference image. This will be zero when both reference and fused images are exactly alike and it will be increased when the fused image is deviated from the reference image.

  $$PFE = \frac{\text{norm}(I_r - I_f)}{\text{norm}(I_r)} \times 100$$

- **Peak Signal to Noise Ratio (PSNR)** [10]

  Its value will be high when the fused and reference images are similar. Higher value implies better fusion. The peak signal to noise ratio is computed as:

  $$PSNR = 20 \log_{10} \left( \frac{L^2}{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (I_r(i,j) - I_f(i,j-1))^2} \right)$$

  where $L$ is the number of gray levels in the image.

**Without Reference Image**

When the reference image is not available, the performance of image fusion algorithms can be evaluated using the following metrics.

- **Standard Deviation** [11]

  It is known that standard deviation is composed of the signal and noise parts. This metric would be more efficient in the absence of noise. It measures the contrast in the fused image. An image with high contrast would have a high standard deviation.

  $$\sigma = \sqrt{\sum_{i=0}^{L} (i - \bar{i})^2 h_f(i)} , \quad \bar{i} = \sum_{i=0}^{L} i h_f(i)$$

  where $h_f(i)$ is the normalized histogram of the fused image $I_f(x, y)$ and $L$ number of frequency bins in the histogram.

- **Spatial Frequency** [8]

  This frequency in spatial domain indicates the overall activity level in the fused image. It is computed using the Eq.11.

  The fused image with higher SF has to be chosen, since SF shows the overall information content in the image.

**Results and Discussion**

The ground truth image $I_t$ is shown in Fig.3a. The source images $I_1$ and $I_2$ to be fused are shown in Fig.3b. The source images have been created by blurring the some portions of the reference image with a Gaussian mask.
using diameter of 12 pixels. The fused & error images by PCA and SF with the procedure given in first architecture are shown in Fig.4a and 4b respectively. Performance evaluation metrics are shown in Table-1. From these results it is observed that image fusion by SF is marginally better.

Figure 5a shows an 100 x 100 image block and Fig.5b to d show the degraded versions after blurring with a disk of radius 5, 9 and 21 pixels respectively. Fig.5a is taken as one of the source images \( I_1 \) and the blurred image is taken as another source image \( I_2 \). The computed principal components and normalized spatial frequencies are shown in Table-2. Fig.6 shows the principal components and normalized spatial frequencies with respect to increase in the amount of blur. It is observed that second principal component and spatial frequency are diminished as the images get more blurred. The rate of change of principal components and normalized spatial frequencies are shown in Fig.7. It is observed that the rate of change of SF is high. Hence, SF would be better indicator of the degradation.

The performance metrics for different thresholds and block sizes are shown in Tables-3 and 4. It is observed that threshold greater than 0.1 in case of PCA and 0.15 in case of SF is better. The performance metrics for different thresholds and block sizes are shown in Tables-3 and 4. It is observed that threshold greater than 0.1 in case of PCA and 0.15 in case of SF is better.
of SF show the degraded performance. The fusion algorithm becomes gray level averaging of corresponding pixels when the chosen threshold is too high (see Eq. 8 and 14). And block sizes $4 \times 4$, $8 \times 8$ and $32 \times 32$ show the degraded performance in both PCA and SF. The fused and error images by PCA and SF are shown in Fig. 8a and 8b respectively with block size $64 \times 64$ and $\theta = 0.025$. Table-5 shows the performance evaluation metrics. It is observed that SF shows slightly better performance. Fig. 9a and Fig. 9b show the fused and error images by PCA and SF respectively with block size $4 \times 4$ and $\theta = 0.2$. And Table-6 shows the performance evaluation metrics. It is observed that in this situation also SF showed slightly better performance.

SF shows slightly better performance. Block based image fusion scheme (second architecture) shows enhanced performance. It could be due to the consideration of local variations presented in the source images. Selection of block size and threshold are very difficult in practice. One way to obtain optimal fusion image is, compute the performance of the fusion for different combination of block sizes and thresholds and then select the fused image corresponding to maximum value from the performance metrics.

![Fig. 6 Normalized spatial frequencies and principal components for varying radius and standard deviation of 10](image)

![Fig. 7 Rate of change of NSF/NPC with magnitude blur](image)

<table>
<thead>
<tr>
<th>th</th>
<th>4x4</th>
<th>8x8</th>
<th>16x16</th>
<th>32x32</th>
<th>64x64</th>
<th>128x128</th>
<th>256x256</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8.0595</td>
<td>6.5148</td>
<td>3.8312</td>
<td>0.0880</td>
<td>0.7468</td>
<td>0.8170</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.025</td>
<td>8.0261</td>
<td>6.5426</td>
<td>3.8312</td>
<td>0.0882</td>
<td>0.7468</td>
<td>0.8170</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.05</td>
<td>8.0046</td>
<td>6.5426</td>
<td>3.8312</td>
<td>0.0882</td>
<td>0.7468</td>
<td>0.8170</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.075</td>
<td>7.9869</td>
<td>6.5014</td>
<td>3.8311</td>
<td>0.0702</td>
<td>0.7468</td>
<td>7.7641</td>
<td>7.5779</td>
</tr>
<tr>
<td>0.1</td>
<td>7.9500</td>
<td>6.5014</td>
<td>3.8311</td>
<td>0.0702</td>
<td>0.7530</td>
<td>8.8116</td>
<td>7.5779</td>
</tr>
<tr>
<td>0.125</td>
<td>7.9454</td>
<td>6.5012</td>
<td>3.8311</td>
<td>0.0702</td>
<td>0.7539</td>
<td>8.8116</td>
<td>7.5779</td>
</tr>
<tr>
<td>0.15</td>
<td>7.9717</td>
<td>6.5208</td>
<td>3.8311</td>
<td>0.0702</td>
<td>0.7695</td>
<td>9.2799</td>
<td>7.5779</td>
</tr>
<tr>
<td>0.175</td>
<td>7.9111</td>
<td>6.4796</td>
<td>3.8311</td>
<td>0.0848</td>
<td>0.7695</td>
<td>9.2799</td>
<td>9.4173</td>
</tr>
<tr>
<td>0.2</td>
<td>7.9420</td>
<td>6.4796</td>
<td>3.8311</td>
<td>0.0932</td>
<td>1.3909</td>
<td>9.9585</td>
<td>9.4173</td>
</tr>
<tr>
<td>0.225</td>
<td>8.0528</td>
<td>6.5012</td>
<td>3.8311</td>
<td>0.0932</td>
<td>2.5048</td>
<td>10.5712</td>
<td>9.4173</td>
</tr>
<tr>
<td>0.25</td>
<td>8.0637</td>
<td>6.5454</td>
<td>3.8311</td>
<td>0.1074</td>
<td>2.5068</td>
<td>9.9022</td>
<td>9.4173</td>
</tr>
<tr>
<td>0.275</td>
<td>8.0652</td>
<td>6.5080</td>
<td>3.8311</td>
<td>0.1153</td>
<td>2.5034</td>
<td>9.9022</td>
<td>9.4173</td>
</tr>
<tr>
<td>0.3</td>
<td>8.0737</td>
<td>6.5274</td>
<td>4.8848</td>
<td>0.1402</td>
<td>2.5127</td>
<td>9.9022</td>
<td>9.4173</td>
</tr>
</tbody>
</table>
## Table-3b: SD of fused image by PCA for different thresholds and block sizes

<table>
<thead>
<tr>
<th>th</th>
<th>4x4</th>
<th>8x8</th>
<th>16x16</th>
<th>32x32</th>
<th>64x64</th>
<th>128x128</th>
<th>256x256</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>49.1966</td>
<td>49.6046</td>
<td>49.9234</td>
<td>50.1269</td>
<td>50.1223</td>
<td>50.1231</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.025</td>
<td>49.2081</td>
<td>49.6216</td>
<td>49.9234</td>
<td>50.1269</td>
<td>50.1223</td>
<td>50.1231</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.05</td>
<td>49.2135</td>
<td>49.6216</td>
<td>49.9234</td>
<td>50.1269</td>
<td>50.1223</td>
<td>50.1231</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.075</td>
<td>49.2137</td>
<td>49.6237</td>
<td>49.9235</td>
<td>50.1268</td>
<td>50.1223</td>
<td>48.3723</td>
<td>47.2823</td>
</tr>
<tr>
<td>0.1</td>
<td>49.2343</td>
<td>49.6237</td>
<td>49.9235</td>
<td>50.1268</td>
<td>50.1216</td>
<td>47.4560</td>
<td>47.2823</td>
</tr>
<tr>
<td>0.125</td>
<td>49.2401</td>
<td>49.6238</td>
<td>49.9235</td>
<td>50.1268</td>
<td>50.1214</td>
<td>47.4560</td>
<td>47.2823</td>
</tr>
<tr>
<td>0.15</td>
<td>49.2367</td>
<td>49.6235</td>
<td>49.9235</td>
<td>50.1268</td>
<td>50.1210</td>
<td>47.0240</td>
<td>47.2823</td>
</tr>
<tr>
<td>0.175</td>
<td>49.2784</td>
<td>49.6412</td>
<td>49.9235</td>
<td>50.1266</td>
<td>50.1210</td>
<td>47.0240</td>
<td>47.2823</td>
</tr>
<tr>
<td>0.2</td>
<td>49.2811</td>
<td>49.6412</td>
<td>49.9234</td>
<td>50.1266</td>
<td>50.0752</td>
<td>46.4345</td>
<td>45.8803</td>
</tr>
<tr>
<td>0.225</td>
<td>49.2512</td>
<td>49.6238</td>
<td>49.9234</td>
<td>50.1266</td>
<td>49.8585</td>
<td>45.8920</td>
<td>45.8803</td>
</tr>
<tr>
<td>0.25</td>
<td>49.2524</td>
<td>49.6213</td>
<td>49.9234</td>
<td>50.1267</td>
<td>49.8574</td>
<td>46.1012</td>
<td>45.8803</td>
</tr>
<tr>
<td>0.275</td>
<td>49.2497</td>
<td>49.6220</td>
<td>49.9234</td>
<td>50.1267</td>
<td>49.8576</td>
<td>16.1012</td>
<td>45.8803</td>
</tr>
<tr>
<td>0.3</td>
<td>49.2652</td>
<td>49.6218</td>
<td>49.7869</td>
<td>50.1265</td>
<td>49.8567</td>
<td>46.1012</td>
<td>45.8803</td>
</tr>
</tbody>
</table>

## Table-4a: RMSE of fused image by SFA for different thresholds and block sizes

<table>
<thead>
<tr>
<th>th</th>
<th>4x4</th>
<th>8x8</th>
<th>16x16</th>
<th>32x32</th>
<th>64x64</th>
<th>128x128</th>
<th>256x256</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6.9607</td>
<td>3.9930</td>
<td>2.0716</td>
<td>0.1861</td>
<td>0.2168</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.025</td>
<td>6.7950</td>
<td>3.7123</td>
<td>2.0716</td>
<td>0.1861</td>
<td>0.2168</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.05</td>
<td>6.7549</td>
<td>3.7123</td>
<td>2.0716</td>
<td>0.1861</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.075</td>
<td>6.6645</td>
<td>3.7866</td>
<td>2.0716</td>
<td>0.1861</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.1</td>
<td>6.6217</td>
<td>3.7202</td>
<td>2.0716</td>
<td>0.1861</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.125</td>
<td>6.5846</td>
<td>3.5754</td>
<td>2.0717</td>
<td>0.1862</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.15</td>
<td>6.4883</td>
<td>3.6633</td>
<td>2.0717</td>
<td>0.1862</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.175</td>
<td>6.4092</td>
<td>3.5619</td>
<td>2.0570</td>
<td>0.1862</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.2</td>
<td>6.3023</td>
<td>3.5618</td>
<td>2.0570</td>
<td>0.1862</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.225</td>
<td>5.9935</td>
<td>3.5868</td>
<td>2.0570</td>
<td>0.1862</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.25</td>
<td>5.9588</td>
<td>3.5869</td>
<td>2.0570</td>
<td>0.1862</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.275</td>
<td>5.9126</td>
<td>3.5873</td>
<td>2.0570</td>
<td>0.1862</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
<tr>
<td>0.3</td>
<td>5.8955</td>
<td>3.5873</td>
<td>2.0570</td>
<td>0.1862</td>
<td>0.4002</td>
<td>0.0214</td>
<td>0.0214</td>
</tr>
</tbody>
</table>
Table 4b: SD of fused image by SFA for different thresholds and block sizes

<table>
<thead>
<tr>
<th>th</th>
<th>4x4</th>
<th>8x8</th>
<th>16x16</th>
<th>32x32</th>
<th>64x64</th>
<th>128x128</th>
<th>256x256</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>49.6070</td>
<td>49.9562</td>
<td>50.0699</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.025</td>
<td>49.6653</td>
<td>49.9744</td>
<td>50.0698</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.05</td>
<td>49.6709</td>
<td>49.9744</td>
<td>50.0698</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.075</td>
<td>49.7048</td>
<td>49.9712</td>
<td>50.0698</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.1</td>
<td>49.7127</td>
<td>49.9714</td>
<td>50.0698</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.125</td>
<td>49.7371</td>
<td>49.9786</td>
<td>50.0698</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.15</td>
<td>49.7499</td>
<td>49.9736</td>
<td>50.0699</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.175</td>
<td>49.7714</td>
<td>49.9810</td>
<td>50.0706</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.2</td>
<td>49.8036</td>
<td>49.9811</td>
<td>50.0706</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.225</td>
<td>49.8809</td>
<td>49.9792</td>
<td>50.0706</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.25</td>
<td>49.8991</td>
<td>49.9792</td>
<td>50.0706</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.275</td>
<td>49.8958</td>
<td>49.9790</td>
<td>50.0706</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
<tr>
<td>0.3</td>
<td>49.8886</td>
<td>49.9790</td>
<td>50.0706</td>
<td>50.1265</td>
<td>50.1262</td>
<td>50.1270</td>
<td>50.1270</td>
</tr>
</tbody>
</table>

Fig. 8a: Fused and error images by PCA (th = 0.025 and block size 64 x 64)

Fig. 9a: Fused and error images by PCA (th = 0.02 and block size 4 x 4)

Fig. 8b: Fused and error images by SFA (th = 0.025 and block size 64 x 64)

Fig. 9b: Fused and error images by SF (th = 0.02 and block size 4 x 4)
PCA and SF based image fusion algorithms are developed to fuse multi focused images and their performance is compared. It is concluded that SF showed slightly better performance. Block based image fusion scheme showed enhanced performance. This architecture is very simple and can be used in real time applications.

### Conclusion

PCA and SF based image fusion algorithms are developed to fuse multi focused images and their performance is compared. It is concluded that SF showed slightly better performance. Block based image fusion scheme showed enhanced performance. This architecture is very simple and can be used in real time applications.

### References


### Table-5 : Performance evaluation metrics with block size of 64x64 and \( th = 0.025 \)

<table>
<thead>
<tr>
<th></th>
<th>RMSE</th>
<th>PFE</th>
<th>PSNR</th>
<th>SD</th>
<th>SF</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>0.1669</td>
<td>0.073</td>
<td>55.9395</td>
<td>57.0859</td>
<td>18.8963</td>
</tr>
<tr>
<td>SF</td>
<td>0.161</td>
<td>0.0704</td>
<td>56.0964</td>
<td>57.086</td>
<td>18.8962</td>
</tr>
</tbody>
</table>

### Table-6 : Performance evaluation metrics with block size of 4x4 and \( th = 0.2 \)

<table>
<thead>
<tr>
<th></th>
<th>RMSE</th>
<th>PFE</th>
<th>PSNR</th>
<th>SD</th>
<th>SF</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>4.8068</td>
<td>2.102</td>
<td>41.3462</td>
<td>56.7722</td>
<td>18.7141</td>
</tr>
<tr>
<td>SF</td>
<td>4.0151</td>
<td>1.7558</td>
<td>42.1279</td>
<td>56.8962</td>
<td>18.8518</td>
</tr>
</tbody>
</table>