Enhanced Morphological Component Analysis Based Image Fusion

Hemani and Puneet Sharma

1 M. Tech. Student, Department of Computer Sc. & Engineering, SPGOI Rohtak, Haryana, India
hmalhotra111@gmail.com

2 Asstt. Professor, Department of Computer Sc. & Engineering, SPGOI Rohtak, Haryana, India

Abstract

MCA is a novel image decomposition method based on sparse representation. The basic idea is separating various morphological characteristics included in the image on the basis of image morphological composition differences. The research modifies the MCA technique. In the existing morphological component analysis technique there are some constraints present that introduce the noise. This noise degrades the resultant image. The proposed algorithm uses the filters to remove the noise. The filters used at the different level. The two input image is separated in two parts i.e. textual part and the smooth part. Then theses parts are filtered and combined to get the resultant image.

The proposed algorithm with the disk filter outperforms the other filters performance.

Keywords: Image Fusion, MCA, Disk Filter, Fusion Method.

1. Introduction

Image fusion is the combination of two or more different images to form a new image by using a certain algorithm. Image Fusion is a process of combining the relevant information from a set of images of the same scene into a single image and the resultant fused image will be more informative and complete than any of the input images. Input images could be multi sensor, multimodal, multifocus or multi temporal. There are some important requirements for the image fusion process [1]:

- The fused image should preserve all relevant information from the input images.
- The image fusion should not introduce artifacts which can lead to a wrong diagnosis.

In the Image fusion process it combines relevant information of two images and then generates the output into a single relevant image. The resulting image will be more informative than any of the input image.

2. Classification of Fusion Methods

A. Pixel-Level Fusion Methods

Most of the methods have been developed for the fusion of static imagery, so temporal aspects arising in the fusion of image sequences, e.g. temporal stability and consistency, are usually not addressed [2]

B. Feature Level Methods

Feature level methods are the next stage of processing where image fusion may take place [3]. Fusion at the feature level requires extraction of features from the input images. Features can be pixel intensities or edge and texture features. The Various kinds of features are considered depending on the nature of images and the application of the fused image. The features involve the extraction of feature primitives like edges, regions, shape, size, length or image segments, and features with similar intensity in the images to be fused from different types of images of the same geographic area. These features are then combined with the similar features present in the other input images through a pre-determined selection process to form the final fused image. The feature level fusion should be easy. However, feature level fusion is difficult to achieve when the feature sets are derived from different algorithms and data sources.
C. Decision-Level Fusion

Decision-level fusion consists of merging information at a higher level of abstraction, combines the results from multiple algorithms to yield a final fused decision. Input images are processed individually for information extraction. The obtained information is then combined applying decision rules to reinforce common interpretation [4]. Decision fusion or classifier combination can be interpreted as making a decision by combining the outputs of different classifiers for a test image. In our case, instead of different type of classifiers, we combined outputs of nearest neighbor classifiers trained by different blocks that correspond to different regions on a face image. For 16x16 blocks, we have 16 different block positions and a separate nearest neighbor classifier is trained by using the features extracted over the training data for that block. From a given test image, 16 feature vectors each corresponding to a different block are extracted. For each test image, local feature vector is given as an input to the corresponding classifier and the outputs of the classifiers are then combined to make an ultimate decision for the test image [5]. Unlike fixed combination methods, trainable combiners use the outputs of the classifier, class posterior probabilities, as a feature set. From the class posterior probabilities of several classifiers each corresponding to a block, a new classifier is trained to provide an ultimate decision by combining the posteriors. To train a combiner, training dataset is divided into two parts as train and validation data. Individual classifiers are trained using the training data part [5]. Then, the class posterior probabilities for each block are calculated on the validation data. For each image, these posterior probabilities are concatenated into a long vector \( \left[ p(C_1|x_1), p(C_2|x_1), ..., p(C_N|x_B), p(C_N|x_B), p(C_N|x_B) \right]^T \)

Which is then used to train the combiner.

3. MCA

MCA is a novel image (signal) decomposition method based on sparse representation. The basic idea is separating various morphological characteristics included in the image (signal) on the basis of image (signal) morphological composition differences (which can be sparsely represented by different dictionaries). The theory of MCA can be applied in image compression, reconstruction, noise suppression and feature extraction, and it’s extremely useful in the respect of image segmentation and image imprinting. For a given n-sample image X, we assume that it consists of a sum of K different signals, which are corresponding to K different morphologies \( \{x_i\}_{i=1,2,...,K} \) respectively. Each \( x_i \) is called a morphological component, and the linear superposition of the K morphological components will form image X, that is \( X = \sum_{i=1}^{K} x_i \). Possibly contaminated with noise \( Y = \sum_{i=1}^{K} x_i + \epsilon \). The MCA framework aims at recovering the components \( \{x_i\}_{i=1,2,...,K} \) from their observed linear combination. MCA assumes that each morphological component \( x_i \) can be sparsely represented in an associated basis \( \Phi_i \).

\[ x_i = \Phi_i a_i, i = 1, 2, ..., K \]

Where \( a_i \) is a sparse coefficient vector (sparse means that only a few coefficients are large and most are negligible, this way, the key message of signals and images can be expressed by the least non-zero coefficients possible). The independent orthogonal bases \( \Phi_i (i = 1, 2, ..., K) \) amalgamated build the over-complete dictionary \( \Phi = [\Phi_1, ..., \Phi_K] \) for each i, the representation of \( x_i \) in \( \Phi_i \) is sparse and not, or at least not as sparse in other \( \Phi_j (i \neq j) \). Thus, the dictionary \( \Phi_i \) plays an important part in discrimination between content types, preferring the component \( x_i \) over other parts. This is a key observation for the success of the MCA-based separation algorithm. Taking advantage of this feature of dictionary \( \Phi \), we can achieve the purpose that decomposing the images (signals) morphological component [6].

4. Proposed Technique

In the existing morphological component analysis technique there are some constraints present that introduce the noise. This noise degrades the resultant image. This degraded resultant image must be enhanced. The proposed algorithm uses the filters to remove the noise. The filters used at the different level. The two input image is separated in two parts i.e. textual part and the smooth part. Then theses parts are filtered and combined to get the resultant image. The process can be explained by following algorithm

**Proposed Algorithm**

1. Input Image 1 say I1
2. Input image 2 say I2
3. Convert image to gray scale
   - I1=RGB2gray(I1)
   - I2=RGB2gray(I2)
4. Separate textual and smooth part of images
TI1=textual(I1)
SI1=smooth(I1)
TI2=textual(I2)
SI2=smooth(I2)
5. Filter The smooth part
NSI1= Filter(SI1)
NSI2= Filter(SI2)
6. Filter The textual part
NTI1= Filter(TI1)
NTI2= Filter(TI2)
7. Fuse textual and smooth parts
NTR=NTI1+NTI2
NSR=NSI1+NSI2
8. Resultant Image=[NTR NSR]

The performance of the above algorithm can be compared with the existing algorithm. The performance may vary from filter to filter.

Table 1: PSNR and MSE Values on Different Images Using Existing MCA Algorithm

<table>
<thead>
<tr>
<th>Image Name</th>
<th>PSNR</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st image name &quot;seventeen.jpg&quot;</td>
<td>24.5151</td>
<td>3.4128</td>
</tr>
<tr>
<td>2nd image name &quot;seven.jpg&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st image name &quot;A.jpg&quot;</td>
<td>24.4342</td>
<td>0.7924</td>
</tr>
<tr>
<td>2nd image name &quot;seven.jpg&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st image name &quot;B.jpg&quot;</td>
<td>26.6685</td>
<td>2.9398</td>
</tr>
<tr>
<td>2nd image name &quot;seven.jpg&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st image name &quot;common.jpg&quot;</td>
<td>21.7068</td>
<td>5.9192</td>
</tr>
<tr>
<td>2nd image name &quot;sevenA.jpg&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st image name &quot;C.jpg&quot;</td>
<td>29.0032</td>
<td>2.2690</td>
</tr>
<tr>
<td>2nd image name &quot;D.jpg&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: PSNR Values on Different Images Using Existing MCA Algorithm

Figure 2: MSE Values on Different Images Using Existing MCA Algorithm
Table 2: PSNR And MSE Values of Modified MCA Using Disk Filter

<table>
<thead>
<tr>
<th>Image Name</th>
<th>Filter Name</th>
<th>PSNR</th>
<th>MSE</th>
</tr>
</thead>
</table>
| 1st image A
| disk, 5     | 25.0526 | 0.4815|
| 2nd image A
| disk, 5     | 25.0526 | 0.4815|
| 1st image B
| disk, 7     | 25.8516 | 0.2198|
| 2nd image B
| disk, 7     | 25.8516 | 0.2198|
| 1st image A
| disk, 10    | 26.9644 | 0.2419|
| 2nd image A
| disk, 10    | 26.9644 | 0.2419|
| 1st image B
| disk, 11    | 27.5183 | 0.3985|
| 2nd image B
| disk, 11    | 27.5183 | 0.3985|
| 1st image A
| disk, 15    | 30.8593 | 0.9076|
| 2nd image A
| disk, 15    | 30.8593 | 0.9076|

Figure 3: PSNR Values of Modified MCA Using Disk Filter

Figure 4: MSE Values of Modified MCA Using Disk Filter

Figure 5: Maximum PSNR Value On Existing, Unsharp Filter, Disk Filter, Gaussian Filter
5. Conclusion

This modifies the MCA technique. In the existing morphological component analysis technique there are some constraints present that introduce the noise. This noise degrades the resultant image. The proposed algorithm uses the filters to remove the noise. The filters used at the different level. The two input image is separated in two parts i.e. textual part and the smooth part. Then these parts are filtered and combined to get the resulting image. The dissertation compares the performance of the existing algorithm with the proposed algorithm. The analysis parameters are PSNR and the MSE. The PSNR of the proposed is better than the existing and the MSE isles. This proves the better performance of the proposed algorithm over the existing algorithm. Then the proposed algorithm is compared for the different types of filters. Firstly the comparison is done on the different value of same filter for the various filters. Then the best solutions are compared with each other. The performance of the DISK filter is found to better than the other existing filters. The proposed algorithm with the disk filter outperforms the other filters.

References