Multi-focus image fusion using Content Adaptive Blurring

Muhammad Shahid Farid[a,b], Arif Mahmood[*,a], Somaya Ali Al-Maadeed[a]

[a] Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar
[b] College of Information Technology, University of the Punjab, Lahore, Pakistan

ARTICLE INFO

ABSTRACT

Multi-focus image fusion has emerged as an important research area in information fusion. It aims at increasing the depth-of-field by extracting focused regions from multiple partially focused images, and merging them together to produce a composite image in which all objects are in focus. In this paper, a novel multi-focus image fusion algorithm is presented in which the task of detecting the focused regions is achieved using a Content Adaptive Blurring (CAB) algorithm. The proposed algorithm induces non-uniform blur in a multi-focus image depending on its underlying content. In particular, it analyzes the local image quality in a neighborhood and determines if the blur should be induced or not without losing image quality. In CAB, pixels belonging to the blur regions receive little or no blur at all, whereas the focused regions receive significant blur. Absolute difference of the original image and the CAB-blurred image yields initial segmentation map, which is further refined using morphological operators and graph-cut techniques to improve the segmentation accuracy. Quantitative and qualitative evaluations and comparisons with current state-of-the-art on two publicly available datasets demonstrate the strength of the proposed algorithm.

1. Introduction

Most imaging systems have a limited depth-of-field. Scene content within a limited distance from the imaging plan remains in focus, while objects closer or further than that appear as blurred or out-of-focus in the image. For a given optical system, the depth-of-field is influenced by the focal length of the lens, distance from the object the camera is focused on, and the aperture or f-number [1]. In remotely sensed images, atmospheric scatter and spatial and temporal sensor integration are the additional causes of blurring [2]. Multiple images of the same scene can be captured with supplementary focused regions, however such images are not suitable for human perception and for many computer vision applications such as object detection, image analysis, remote sensing applications [3–8]. To this end, multi-focus image fusion is an effective solution which automatically detects the focused regions in multi-focus images and integrates them to produce a composite image in which all objects of interest are in focus.

Numerous multi-focus image fusion techniques have been proposed in the past years [9–13]. Based on their domain, these approaches can be categorized into two classes: transform domain and spatial domain algorithms [3]. Transform domain image fusion algorithms have been very popular in the past years due their more intuitive approach towards the problem. Most transform based multi-focus image fusion

algorithms work in three steps: first, the source images are converted into transform domain. Second, according to some fusion rule these transform coefficients are fused to obtain composite coefficients, and finally these composite coefficients are converted back to the spatial domain to obtain the fused image. Various transforms have been explored for image fusion, however multi-scale transform based image fusion algorithms have received more popularity. Examples of such transforms include: Various pyramid based decompositions, e.g., Laplacian, gradient, and morphological pyramids [14–16], discrete wavelet transform (DWT) [9,17–21], dual-tree complex wavelet transform (DTCWT) [22–24], curvelet transform [25–27], contourlet transform [28–30], discrete cosine transform [31–34], and sparse representation [35–39].

Recently, spatial domain image fusion algorithms have received significant research efforts resulting in several image fusion algorithms that operate directly on the source images without converting them into some other representation. These algorithms apply a fusion rule on the source images to obtain an all-in-focus image. These methods can be categorized into two groups: pixel level and block or region level algorithms. In pixel level image fusion algorithms [40–44,67], the corresponding pixels of the source images are averaged to obtain the composite image. These algorithms are simple and computationally fast, however they suffer from various fusion artifacts such as ghosting

and blurring.

Region based multi-focus image fusion algorithms [45–48] divide the source images into blocks or regions and compute some focus measurement such as variance, image spatial frequency or gradient for each region. Based on this measure, the focused regions are detected and integrated to obtain the fused image. The focus measure can be a binary map representing the blur and the focus regions, or it may be a weight map with continuous values between 0 and 1 representing the degree to which the region or block belongs to the blur or the focused region.

In this paper, we propose a novel multi-focus image fusion algorithm that detects the focused regions in the source images using Content Adaptive Blurring (CAB) and integrates them to obtain the fused image. In the CAB, a multi-focus image is iteratively blurred in such a manner that only the focused regions get blurred whereas the de-focus regions receive a little or no blur at all. It introduces spatially varying amount of blur in the image depending upon the underlying quality of the image. The local image quality degradation due to blurring is measured using a local correlation coefficient (ρ) between the blurred and the original images. If blurring degrades the quality of a local image region more than an allowable limit, that region is not blurred and exempted from further blurring. Thus, the de-focused regions are preserved while the focused regions get blurred.

By taking absolute difference of a CAB-blurred image and the original image, we can find the regions receiving a particular amount of blur. Based on this information, we separate the focused and the de-focused regions in the source images. This segmentation is further refined using graph cut techniques and morphological operators. Finally, the focused regions are fused to obtain an all-in-focus image. The performance of the proposed multi-focus image fusion algorithm is evaluated on two publicly available datasets, and the results are compared with fifteen state-of-the-art methods. Quantitative and qualitative evaluations reveal the excellent performance of the proposed algorithm.

The rest of this paper is organized as follows. An overview of the proposed approach is given in Section 2. In Section 3, Content Adaptive Blurring (CAB) algorithm is presented. The theoretical relationships between the blurred and the original images are derived and used to formulate a strategy to automatically control blurring. Section 4 describes the proposed multi-focus image fusion algorithm based on the CAB algorithm. Experimental results, quantitative and qualitative evaluations and comparison to the current state-of-the-art are presented in Section 5 and finally the paper is concluded in Section 6.

2. Overview of the proposed image fusion algorithm

The central step in any multi-focus image fusion algorithm is the estimation of decision map for the source multi-focus images. The decision map defines the focused and the de-focused regions in a multi-focus image. To achieve this task in the proposed multi-focus image fusion algorithm, we propose a novel Content Adaptive Blurring (CAB) algorithm. The CAB is based on a simple observation. If an image is blurred with a uniform blur kernel, image quality in focused regions degrades significantly more than the blurred regions. It is because, if we repeatedly apply a blur filter on a good quality image region the quality degradation will increase as number of iterations increases. We measure the local image quality degradation by using local correlation coefficient between the blurred and the original images. Based on an allowable degradation in similarity, the CAB induces controlled blur in an image. Already blurred image regions are preserved while the focused image regions get blurred.

When a multi-focus image is repeatedly blurred, the quality degradation in the focused regions occurs significantly more than that of the de-focused regions. To elaborate this fact, we conducted an experiment using 10 natural multi-focus images. We repeatedly blurred each image 20 times. In each image, a focused and a de-focused region is selected to analyze quality after each iteration. In Fig. 1, four blurred

versions of a multi-focus image are shown. In the bottom row, a focused and a de-focused region is magnified to show the impact of blur. It can be noted from the magnified regions that degradation in the focused region is significantly more than that of the de-focused regions. Visually, the quality of the de-focused region in the blurred images looks almost unchanged.

To observe the quality degradation quantitatively, correlation coefficient ρ between blurred images the corresponding regions in the original images is computed as shown in Fig. 2. The green plots show the quality degradation in the focused regions and the blue plots show the same for de-focused regions. These graphs reaffirm our observation that the focused regions receive significant amount of blur in each iteration, whereas the de-focused regions receive a little or no blur at all. In case of focused regions, the drop in quality is significant in the earlier iterations which reduces in the later blur iterations. In the last few iterations, the difference of ρ between consecutive iterations is relatively small because the focused regions become homogeneous due to excessive blurring. But overall quality degradation increases after each iteration. On the other hand, a very small drop in the quality of the de-focused regions is noted in the first two iterations and after that the plot is almost horizontal representing no further degradation.

The absolute difference between the blurred and the original image shows the amount of blur induced at each pixel. We use this information to separate the focused and the de-focused regions in a multi-focus image. This initial segmentation is improved using morphological operators and a graph cut technique to obtain a decision map, which is used to fuse the multi-focus source images into a single all-in-focus image. Fig. 3 shows the schematic diagram of the proposed fusion algorithm.

3. Content Adaptive Blurring (CAB) algorithm

The CAB is a block-based blurring algorithm that induces blur at an image pixel only if the local quality at that pixel is not degraded below a given threshold. Spatial domain blurring is often obtained by convolving an averaging filter with the image. Let I_o be the original multi-focus image, a blurred image I_b is obtained by computing each pixel in I_b by a weighted average of pixels in a small neighborhood in I_o. In the rest of the paper, this neighborhood is referred as *blur support* of that pixel. Let $I_b(x, y)$ represents the pixel intensity at location (x, y). Then

$$I_b(x, y) = \sum_{i=-d}^{d} \sum_{j=-d}^{d} w_{i,j} I_o(x + i, y + j), \tag{1}$$

where w is the averaging filter of size $(2d + 1) \times (2d + 1)$. The filter w is symmetric around the central pixel (x, y) and is normalized to 1 to preserve the overall image energy, i.e.,

$$\sum_{i=-d}^{d} \sum_{j=-d}^{d} w_{i,j} = 1 \tag{2}$$

The proposed CAB algorithm does not require a specific blur filter; blurring may be performed by using any blur filter. In this paper we use Gaussian averaging filter for blurring:

$$w_{i,j} = \frac{1}{\alpha_g} \exp\left(-\frac{i^2 + j^2}{2\sigma_g^2}\right), \tag{3}$$

where α_g is a normalization term and σ_g is Gaussian variance which controls the weight distribution and the filter size.

Definition 3.1. Block blur support: Let $r_b\{x, y\}$ be a block of size $m \times n$ pixels in the blurred image I_b centered at (x, y). The blur support of this block is the set of all pixels in the original image I_o which were used to compute at least one pixel in r_b. Thus, for $r_b\{x, y\}$ the blur support is $(m + 2d) \times (n + 2d)$ pixels in I_o.

A pixel in block $r_b(x, y)$ at offset (i, j) from center (x, y), denoted as
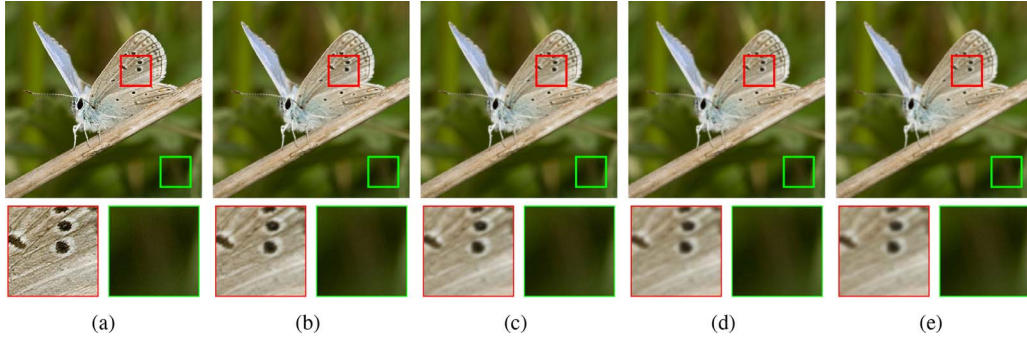
Fig. 1. (a) Top row: Original multi-focus image B_0, (b)–(e) 5th, 10th, 15th and 20th blurred images. Bottom row shows the magnified focused region and de-focused regions in red and green squares, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
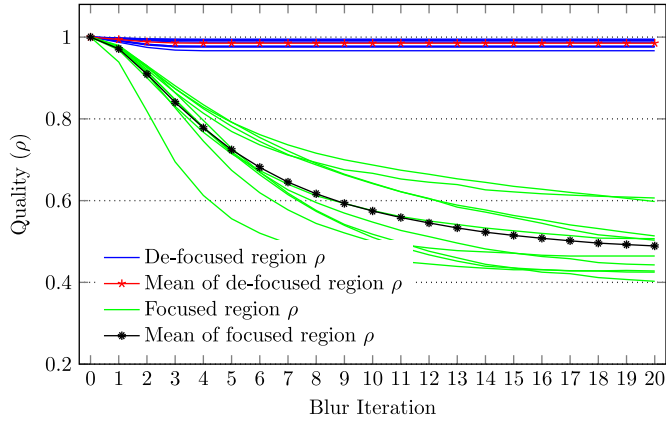

Fig. 2. Quality degradation of focused and de-focused regions with repetitive blur. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$r_b(x + i, y + j)$, is computed from the corresponding block $r_o\{x + i, y + j\}$ of size $(2d + 1) \times (2d + 1)$ in the original image. The block $r_o\{x + i, y + j\}$ of size $(2d + 1) \times (2d + 1)$ is called the blur support of pixel $r_b(x, y)$. The blur support of block $r_b\{x, y\}$ is the union of $(2d + 1) \times (2d + 1)$ overlapped blocks in I_o, each represented by $r_o\{x + i, y + j\}$ in the original image. Thus for block $r_b\{x, y\}$, the blur support is $(m + 2d) \times (n + 2d)$ pixels in I_o.

3.1. Controlling image quality degradation due to blur

To estimate the degradation in the local image quality due to blurring, we propose to use the correlation coefficient measure (ρ). We evaluated a few 2D image quality metrics to find the one that performs the best with the proposed algorithm. Image quality metrics (IQM) like SSIM [49], FSIM [50], UQI [51], etc, aim at evaluating the perceptual quality of the image. In the proposed algorithm, the intensity difference between the original and blurred image is important to obtain accurate decision maps. We found that this difference is better captured by statistical difference-based quality metrics, such as, MSE, PSNR and correlation coefficient (ρ) rather than the perceptual IQMs. PSNR measures the quality degradation relative to the peak intensity, and its results are unbounded and varies from image to image. During iterative blurring in the proposed approach, we wish to exempt the locations with quality less than a predefined threshold. Defining this threshold in PSNR is difficult. The correlation coefficient on the other hand is bounded between [−1,1] and is more appropriate for the proposed algorithm. The mean squared error (MSE) can also be used in the proposed algorithm however normalized MSE and ρ are linearly related to each other, therefore using MSE will not drastically change the results.

Correlation coefficient between blocks r_b and r_o is $\rho(r_b, r_o)$. To keep the notation simple, we represent it as $\rho_{b,o}$ and is computed as:

$$\rho_{b,o} = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{(r_b(x + i, y + j) - \mu_b)(r_o(x + i, y + j) - \mu_o)}{\sigma_b \sigma_o}, \tag{4}$$
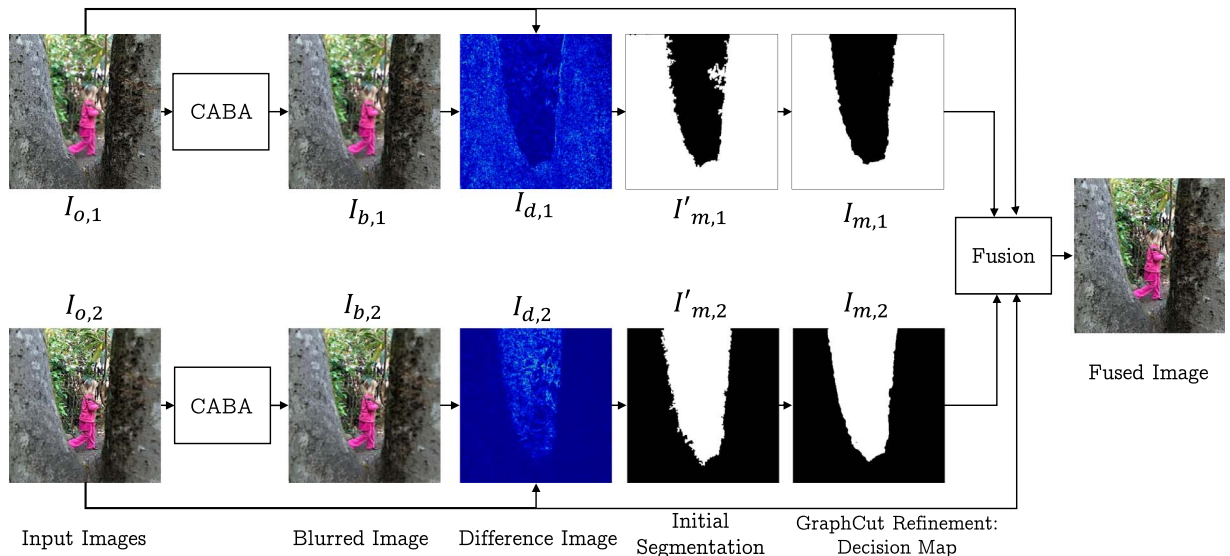
where σ_b and σ_o is:

Fig. 3. Schematic diagram of the proposed multi-focus image fusion algorithm.

$$\sigma_b = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (r_b(x+i, y+j) - \mu_b)^2},$$

$$\sigma_o = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (r_o(x+i, y+j) - \mu_o)^2},$$

and μ_b, μ_o are the means of blocks r_b and r_o respectively.

The Cross-correlation ($\psi_{b, o}$) between the blocks r_b and r_o is

$$\psi_{b,o}(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} r_b(x+i, y+j) r_o(x+i, y+j). \tag{5}$$

Given the cross correlation $\psi_{b,o}(x+i, y+j)$ between the blocks r_o and r_b, the cross correlation for the next blur iteration can be computed from $\psi_{b,o}(x+i, y+j)$ and the blur kernel w. It is denoted as $\widehat{\psi}_{b,o}(x, y)$. That is, $\widehat{\psi}_{b,o}(x, y)$ is the weighted sum of $\psi_{b, o}$ over its blur support:

$$\widehat{\psi}_{b,o}(x, y) = \sum_{i=-d}^{d} \sum_{j=-d}^{d} w_{i,j} \psi_{b,o}(x+i, y+j), \tag{6}$$

Similar to (6), given the correlation coefficient $\rho_{b, o}$ between the blocks r_o and r_b, the correlation coefficient for the next blur iteration, denoted as $\widehat{\rho}_{b,o}$, can be computed from $\rho_{b, o}$. The following result describes this relationship.

Result 3.1. Given the correlation coefficient $\rho_{b, o}$ between the blocks r_o and r_b, the correlation coefficient for the next blur iteration $\widehat{\rho}_{b,o}$ is a variance-ratio weighted sum of the correlation coefficients over the blur support of I_b:

$$\widehat{\rho}_{b,o}(x, y) = \sum_{i,j} \frac{w_{i,j} \sigma_o(x+i, y+j)}{\sigma_b(x, y)} \rho_{b,o}(x+i, y+j) \tag{7}$$

Proof. By definition

$$\widehat{\rho}_{b,o} = \frac{\widehat{\psi}_{b,o} - mn\mu_o\mu_b}{\sigma_o\sigma_b} \tag{8}$$

Substituting values of μ_b (the mean of r_b) and $\widehat{\psi}_{b,o}$ (6):

$$\widehat{\rho}_{b,o} = \frac{\sum_{i,j} w_{i,j} \psi_{b,o}(x+i, y+j) - mn\mu_o \sum_{i,j} w_{i,j}\mu_o(x+i, y+j)}{\sigma_o\sigma_b}$$

$$= \sum_{i,j} \frac{w_{i,j}\sigma_o(x+i, y+j)}{\sigma_b(x, y)} \left(\frac{\psi_{b,o}(x+i, y+j) - mn\mu_o\mu_o(x+i, y+j)}{\sigma_o\sigma_o(x+i, y+j)} \right)$$

$$= \sum_{i,j} \frac{w_{i,j}\sigma_o(x+i, y+j)}{\sigma_b(x, y)} \rho_{b,o}(x+i, y+j)$$

$\square$

If a given image is repetitively blurred, the blur support will increase with the increasing number of iterations. Therefore, the decrease in $\rho_{b, o}$ in (7) will be larger if a focused region is blurred compared to a de-focused region. We use this as an indicator of focused detection in multi-focus images.

3.2. Image quality degradation rate over repeated blurring

If an image is repeatedly convolved with an averaging mask, the area included in the averaging process increases in each iteration. The uncorrelated pixels included in the averaging process cause quality degradation. As the fraction of uncorrelated pixels increases, the rate of quality degradation will also increase. Once the fraction of uncorrelated pixels has reached a saturation level, in the later iterations the degradation rate may decrease. Nevertheless, the overall image quality keeps on decreasing with each blur iteration. In the following Lemma we show that the equivalent size of filter increases in each iteration resulting in inclusion of more content in the averaging process.

Lemma 3.1. *Convolving an image n times with a Gaussian filter with standard deviation σ is equivalent to convolving the same image with an equivalent Gaussian filter with standard deviation $\sqrt{n}\sigma$.*

Proof. We prove it for 1D filter, which can then be extended to higher dimensions without loss of generality. Let $f(x)$ be a 1D Gaussian filter with standard deviation σ and zero mean and $F(u)$ be the Fourier transform of $f(x)$:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \qquad F(u) = e^{-\frac{u^2\sigma^2}{2}} \tag{9}$$

Let I be the original image and I_c be the n times convolved image $I_c = f*(f*(...(f*I)))$. By the convolution theorem, convolution in time domain is equivalent to multiplication in the frequency domain. Let $\widehat{I}$ and $\widehat{I}_c$ be the Fourier transforms of the original and the convolved images Therefore, equivalent filter is given by:

$$F_e(u) = e^{-\frac{u^2\sigma^2}{2}} \cdot e^{-\frac{u^2\sigma^2}{2}} ... e^{-\frac{u^2\sigma^2}{2}} = e^{-\frac{u^2}{2}[n\sigma^2]}$$

The equivalent Gaussian filter in the time domain is given by

$$f_e(x) = \frac{1}{(\sqrt{2n\pi}\sigma)} e^{-\frac{x^2}{2n\sigma^2}} \tag{10}$$

The variance of this resultant filter is $n\sigma^2$, and thus the standard deviation is $\sqrt{n}\sigma$. $\square$

If we convolve an image for 1, 2, 3, ..., n iterations, the standard deviation of the corresponding equivalent Gaussian filter given by (10) will be $\sigma, \sqrt{2}\sigma, \sqrt{3}\sigma, \sqrt{4}\sigma, ..., \sqrt{n}\sigma$ which is an increasing sequence. The size s of the corresponding filter is $s = 2\lceil 2\sqrt{n}\sigma \rceil + 1$ and the size of 2D filter is $s \times s = (2\lceil 2\sqrt{n}\sigma \rceil + 1)^2$, which shows that $s \times s$ will increase as n increases.

To empirically analyze the variation of rate of quality degradation with repeated blurring, we conducted an experiment on a set of 100 natural images. We iteratively blurred each image with a 3×3 Gaussian kernel having $\sigma = 0.4$. We performed 10 blur iterations on the original image B_0 to obtain $B_1, B_2, B_3, ..., B_{10}$ blurred images. To observe the quality degradation by selecting a block of size 15×15 from the center of each blurred image and computing ρ with the corresponding block in the original image B_0. In Fig. 4(a), iteration 0 represents the original image quality ($\rho = 1$) before applying the blur kernel. It can be observed that the quality of the selected region is decreasing with each blur iteration. The average quality after 10 blur iterations is around 0.92.

Fig. 4(b) shows the rate of quality degradation:

$$\Delta\rho_n = (\rho_n - \rho_{n+1})/\rho_n \times 100$$

over consecutive blur iterations. The average plot shows that $\Delta\rho_n$ increases up to $n = 7$ blur iteration and after that it starts decreasing. It is because in each blur iteration the size of blur kernel increases, which incorporates pixel positions farther from the central location. The pixel positions close to the central location have high auto-correlation while the pixel positions away from the central location are uncorrelated. As more and more uncorrelated locations are included in the blur process, the quality degradation increases. It happens up to 7th iteration in this case, then due to significant number of uncorrelated location already included in the blur process, inclusion of further uncorrelated locations results in reduced rate of quality degradation.

3.3. The CAB algorithm

The proposed Content Adaptive Blurring (CAB) method is shown in Algorithm 1. Given a multi-focus image, the CAB blurs the image using (1) and computes block based correlation coefficient between the original image and the blurred image using (8). Locations having correlation less than a threshold λ have observed significant loss of quality. Image content at these locations is restored back from the original image and exempted from further blurring in the next iteration.

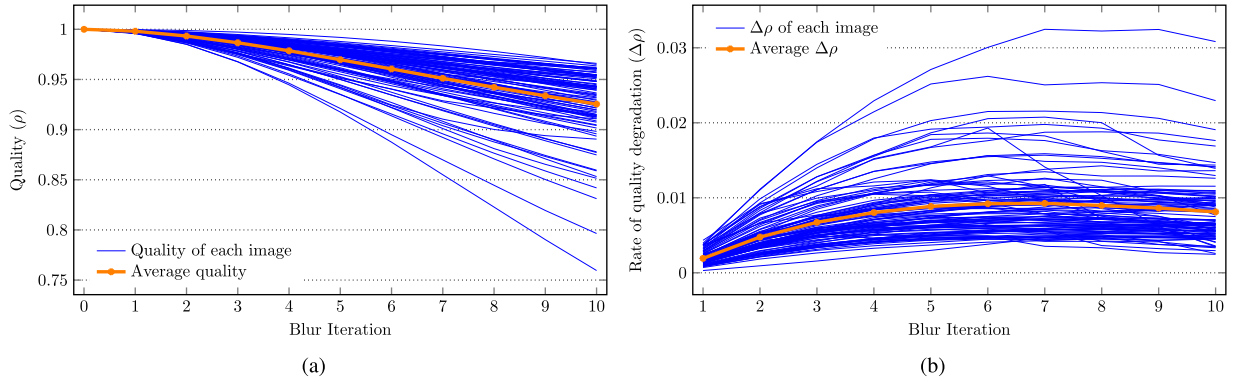Input to the algorithm is a multi-focus image I_o, block size (m, n),

Fig. 4. (a) Quality degradation with each blur iteration, (b) average rate of quality degradation ($\Delta\rho$) against each blur iteration.

quality parameter λ, and blur filter w of size $d \times d$. For the input image, a standard deviation matrix Σ_o is computed using block sum matrix S_o and block sum of squared matrix Q_o. Both of these matrices are computed using an efficient running sum algorithm RUNSUM (Algorithm 2) which requires only four summation operations to find sum of a block of arbitrary size. For an input image of size $p \times q$ pixels, the complexity of RUNSUM algorithm is $O(pq)$.

Note that, the block sum can also be computed using Integral image approach proposed by Viola and Jones [52]. However, our algorithm avoids the cost of computing the integral images and directly computes the sum of a fixed block of size $m \times n$ pixels.

The input image is blurred in iterations. In each iteration, blur filter w is convolved with the image obtained from the last iteration I_{k-1}. Standard deviation matrix Σ_b is computed using block sum S_b and squared block sum Q_b matrices. Also, a product matrix P is computed as point-wise multiplication of input image I_o and the blurred image I_k. Applying running sum algorithm on P we compute cross-correlation matrix $\psi_{o,\,b}$. Then using all these matrices, at each point local correlation coefficient between the blurred image I_k and the input image I_o is computed in matrix R_{ob}. All pixel locations where $R_{ob} < \lambda$ are the locations which should not have been blurred. At these locations blurring has caused loss of quality more than allowed. For these locations pixels from the last blurred image are copied to the output matrix I_b. Values copied to I_b in early iterations are preserved and are not overwritten in the later iterations. When $R_{ob} < \lambda$ becomes true for all pixels of I_k or the specified number of iterations are completed, algorithm stops because further blurring is not required.

4. Proposed multi-focus image fusion algorithm

The proposed multi-focus image fusion approach is outlined in Algorithm 3. It has three steps; in the first step, we detect the focused region in multi-focus source images using CAB as described in the previous section. In the second step, we refine the initial segmentation using graph cut and morphological operations. In the final step, a pixel based fusion is performed to obtain the fused image. These steps are described in detail in the following sections.

4.1. Detecting the focused region

We use the CAB algorithm to compute the decision maps and use them to obtain the fused image from a given set of multi-focus images. Let $I_{o,\,1}$ and $I_{o,\,2}$ be the two source images. We use CAB algorithm to obtain the non-uniformly blurred images $I_{b,\,1}$ and $I_{b,\,2}$ respectively and compute their absolute difference with corresponding source images,

$$I_{d,1} = |I_{o,1} - I_{b,1}|,$$
$$I_{d,2} = |I_{o,2} - I_{b,2}|.$$

The difference maps are thresholded to obtain the focused region

maps $I_{m,\,1}$ and $I_{m,\,2}$ which we refer as decision maps. In particular, the pixels with difference values greater than τ are marked as focused and the rest as de-focused pixels. It is possible that some focused pixels are marked as de-focused or vice versa. We use connected component analysis [54] to eliminate the stranded false positives and flood fill algorithm [55] to reduce the false negatives. Fig. 5 shows an example of detecting the focused region. Fig. 5(a) shows a sample multi-focus image from Lytro multi-focus image fusion database [36,53]. Fig. 5(b) shows the result of applying CAB algorithm. It can be noted from the resultant image, that CAB has induced non-uniform blur in the image. The focused regions have been blurred while the de-focused regions received little or no blur at all. This can be observed in the difference image shown in Fig. 5(c). We see significant pixel intensity variation in the focused image regions, whereas the de-focused regions are mostly dark blue representing very low change in the pixel intensities in these regions. Fig. 5(d) shows the result of applying thresholding on Fig. 5(c).

From experiments, we observed that sometimes the segmentation results are not very accurate at the boundary between the focused and de-focused image regions (compare the detected region in Fig. 5(d) to the original multi-focused image in Fig. 5(a)). To tackle this problem, we use a graph cut technique described in the following section.

4.2. Refining the segmentation using graph cut

Graph cut techniques have emerged as a powerful solutions to many optimization problems in computer vision including the object segmentation [56–58]. In common graph-cut based methods the image pixels are divided into at least two and ideally three classes. We use the latter strategy and divide the image pixels into three classes namely, *definite focus F, definite de-focus B* and *unknown or uncertain U*. The first two classes *F* and *B* are considered as fixed whereas pixels belonging to *U* class are to be classified. This process is known as *partitioning* and is based on the statistical characteristics of pixels in *U* class. Graph cut techniques solve this membership problem through Gaussian mixture models. GMMs for both fixed classes ($F, \ B$) are estimated and the problem of deciding the *U* class pixels is then solved though graph based optimization [79].

The image is represented as an undirected weighted graph where the pixels form the vertices and edges link a vertex with its neighborhood. The weight of an edge represents the similarity between its end vertices. Two terminal nodes *S* and *T* are added in the graph representing the *F* and *B* classes respectively, and they are linked to every other node in the graph. Now, there are two types of links that a vertex has: N-links and T-links. N-links are the links a vertex has with its neighborhood vertices whereas the T-links are the links of a vertex with the terminal nodes (*S* and *T*). The weight of N links w_N between vertices *u* and *v* is computed as in [57]:

$$w_N(u, v) = \frac{\alpha}{dist(u, v)} \exp(-\beta \|c_u - c_v\|^2) \tag{11}$$

Require: Image I_o, Block size m, n, Quality parameter λ, Filter w, Maximum blur iterations $maxItr$
Ensure: Nonuniform blurred image: I_b

1:   $S_o = \text{RUNSUM}(I_o, m, n)$
2:   $Q_o = \text{RUNSUM}(I_o * I_o, m, n)$
3:   $\Sigma_o = \sqrt{Q_o - \frac{1}{mn} S_o * S_o}$
4:   $k = 1, I_k = I_o, I_b = 0$ {k is iteration counter}
5:   **while** $k \leq maxItr$ **do**
6:     $I_k = I_{k-1} \otimes w$ {apply blur filter on current image}
7:     $P = I_k * I_o$ {$*, \div$ are point-wise operators}
8:     $\psi_{ob} = \text{RUNSUM}(P, m, n)$
9:     $S_b = \text{RUNSUM}(I_k, m, n)$
10:    $Q_b = \text{RUNSUM}(I_k * I_k, m, n)$
11:    $\Sigma_b = \sqrt{Q_b - \frac{1}{mn} S_b * S_b}$
12:    $R_{ob} = (\psi_{ob} - \frac{1}{mn} S_b * S_o) \div (\Sigma_b * \Sigma_o)$
13:    **for all** (x, y) **do**
14:      **if** $R_{ob}(x, y) < \lambda$ **and** $I_b(x, y) == 0$ **then**
15:        $I_b(x, y) = I_{k-1}(x, y)$
16:      **end if**
17:    **end for**
18:    $k = k + 1$
19:   **end while**
20:   **for all** (x, y) **do**
21:    **if** $I_b(x, y) == 0$ **then**
22:      $I_b(x, y) = I_{k-2}(x, y)$
23:    **end if**
24:   **end for**

Algorithm 1. Content Adaptive Blurring (CAB) algorithm.

Require: Image I, Block size m, n, Image size h, w
Ensure: S: Block-wise sum of Image I

1:   $R = 0, S = 0$ {Matrix of size $h \times w$ initialized with zero}
2:   **for all** $i < h$ **do**
3:     **for all** $j < w$ **do**
4:       **if** $j >= n$ **then**
5:         $R(i, j) = R(i, j - 1) + I(i, j) - I(i, j - m)$
6:       **else if** $j > 0$ **and** $j < n$ **then**
7:         $R(i, j) = R(i, j - 1) + I(i, j)$
8:       **else if** $j = 0$ **then**
9:         $R(i, j) = I(i, j)$
10:      **end if**
11:     **end for**
12:   **end for**
13:   **for all** $i < h$ **do**
14:     **for all** $j < w$ **do**
15:       **if** $i >= m$ **then**
16:         $S(i, j) = S(i - 1, j) + R(i, j) - R(i - m, j)$
17:       **else if** $i > 0$ **and** $i < m$ **then**
18:         $S(i, j) = R(i, j) + S(i - 1, j)$
19:       **else if** $i = 0$ **then**
20:         $S(i, j) = R(i, j)$
21:      **end if**
22:     **end for**
23:   **end for**

Algorithm 2. RUNSUM.

Require: Multi-focus source images: $I_{o,1}$, $I_{o,2}$, Block size m, n, Quality parameter: λ, Filter w, Threshold τ

Ensure: Fused image: I_f

1: $I_{b,1} = \text{CAB}(I_{o,1}, m, n, \lambda, w)$

2: $I_{b,2} = \text{CAB}(I_{o,2}, m, n, \lambda, w)$

3: $I_{d,1} = |I_{o,1} - I_{b,1}|$

4: $I_{d,2} = |I_{o,2} - I_{b,2}|$

5: $I'_{m,1} = \text{Threshold}(I_{d,1}, \tau)$ {1 if value $> \tau$; 0 otherwise}

6: $I'_{m,2} = \text{Threshold}(I_{d,2}, \tau)$ {1 if value $> \tau$; 0 otherwise}

7: $I''_{m,1} = \text{ConnectedComp}(I'_{m,1})$ {Pick largest component}

8: $I''_{m,2} = \text{ConnectedComp}(I'_{m,2})$ {Pick largest component}

9: $t_{m,1} = \text{TrimapGen}(I''_{m,1})$ {apply erosion then dilation}

10: $t_{m,2} = \text{TrimapGen}(I''_{m,2})$ {apply erosion then dilation}

11: $I_{m,1} = \text{GrabCut}(I_{o,1}, t_{m,1})$

12: $I_{m,2} = \text{GrabCut}(I_{o,2}, t_{m,2})$

13: **for all** (x, y) **do**

14: 　　$\kappa = I_{m,1}(x,y) + I_{m,2}(x,y)$

15: 　　**if** $\kappa > 0$ **then**

16: 　　　　$I_f(x,y) = \frac{1}{\kappa}[I_{o,1}(x,y) \cdot I_{m,1}(x,y) + I_{o,2}(x,y) \cdot I_{m,2}(x,y)]$

17: 　　**else**

18: 　　　　$I_f(x,y) = \frac{1}{2}[I_{o,1}(x,y) + I_{o,2}(x,y)]$

19: 　　**end if**

20: **end for**

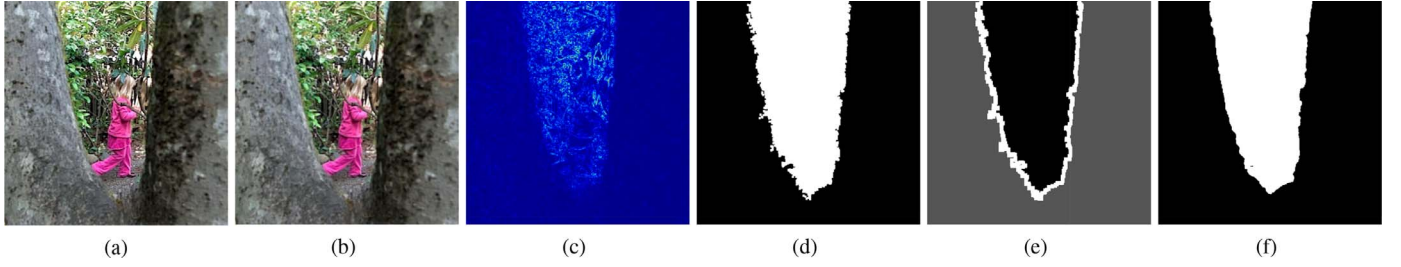**Algorithm 3.** Proposed image fusion algorithm.

Fig. 5. Detecting the focused region in a multi-focus image using CAB. (a) A sample image with background in focused from Lytro dataset [53], (b) blurred image of (a) using CAB, (c) difference of (a) and (b), (d) detected focused region (shown in white), (e) trimap for graph cut (white area shows the uncertain region), (f) refined focused region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $dist(u, v)$ is the Euclidean distance between $u, v; \alpha, \beta$ are constants and c_u represents the color value of u. The weight of a T-link is computed as follows:

$$w_T(u, v) = \begin{cases} 0, & \text{if } u = S \text{ and } v \in B \\ L, & \text{if } u = S \text{ and } v \in F \\ 0, & \text{if } u = T \text{ and } v \in F \\ L, & \text{if } u = T \text{ and } v \in B \\ p(v|\mathcal{G}_F) & \text{if } u = S \text{ and } v \in U \\ p(v|\mathcal{G}_B) & \text{if } u = T \text{ and } v \in U \end{cases}$$

where p is computed using

$$p(v|\mathcal{G}_F) = \sum_{i=1}^{K} \pi_i g_F(v|\mu_i, \Sigma_i) \tag{12}$$

L is a large enough constant that respects the fixed classes F and B, usually a number greater than all the weights in the graph. This initialization respects the fixed classes F and B as they must remain unchanged during the partitioning process. The T-links weights of U class pixels v with S terminal are computed as likelihood p between vertex v and F class GMMs (g_F). The weight of T-link between vertices of U and B is analogously computed using corresponding GMMs (g_B). After weight computation, the fine segmentation of U is achieved by iterative energy minimization through max-flow min-cut algorithm [59]. Fig. 6 shows a typical graph-cut scenario [79].

4.3. Trimap generation

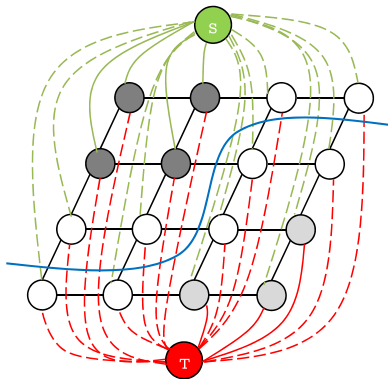To apply graph-cut approach to our problem, we need to define a

Fig. 6. Graph-Cut segmentation scenario. Light gray vertices are background pixels (B), dark gray are foreground pixels (F) and white nodes form the U class. Black edges between the vertices are the N-links. Red and green color edges (both solid and dotted) are T-links. Solid red edges are T-links between the vertex T and the background pixels B and solid green edges are T-links between the vertex S and the foreground pixels F. The solid edges do not take part in partitioning process. The dotted red and green edges are the T-links between {S,T} and U class. The blue edge across the graph represents a possible 'Cut'. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

trimap representing the corresponding classes: *F, B* and *U*. To this end, we take the focused region map obtained in the previous section as F class. Since, the false negatives usually lie in and around the object boundary we use morphological operator *dilation* to estimate the 'Unknown' region U. The unknown region U is then computed as:

$$U = F \setminus F_{dilated}, \tag{13}$$

where $\setminus$ shows the set minus operation and region B is computed by negating the dilated mask:

$$B = \text{NOT}(F_{dilated}) \tag{14}$$

The trimap is then refined via graph cut method [59]. Fig. 5(e) shows the trimap obtained from Fig. 5(d) by using the above described strategy. Fig. 5(f) shows the final focused region mask, improvement achieved by graph-cut based refinement can be noted around the boundary regions. These focused region masks are used as decision maps for fusion process.

4.4. Computing the fused image

The decision maps $I_{m, 1}, I_{m, 2}$ obtained in the previous Section are used to integrate the source images $I_{o, 1}, I_{o, 2}$ to obtain the composite image I_f according to the following fusion rule:

$$I_f(x, y) = \begin{cases} \dfrac{I_{o,1}(x, y) \cdot I_{m,1}(x, y) + I_{o,2}(x, y) \cdot I_{m,2}(x, y)}{\kappa} & \text{if } \kappa > 0 \\ \dfrac{I_{o,1}(x, y) + I_{o,2}(x, y)}{2} & \text{otherwise} \end{cases} \tag{15}$$

where $\kappa = I_{m,1}(x, y) + I_{m,2}(x, y)$. In our fusion rule (15), the pixels which are marked as focused regions or de-focused regions in both decision maps are averaged.

4.5. Complexity analysis of the proposed algorithm

Computation of correlation coefficient matrix R_{ob} has complexity of the order of $O(pq)$, where $\{I_o, I_b\} \in R^{p \times q}$. The complexity of convolution is d^2pq. The overall complexity of Algorithm 1 is $O(kd^2pq)$ where k is the number of iterations. The complexity of connected components analysis is $O(pq)$ [60,61] and that of trimap generation is $O(pq)$. The complexity of GrabCut algorithm is $O((pq)^2)$. Hence, the overall worst-case complexity of the proposed image fusion algorithm is $O((pq)^2)$.

5. Experiments and results

In this section, we present the experimental settings of the proposed algorithm, introduce the datasets used to test the performance, qualitative and quantitative evaluations, and comparisons with the current state-of-the-art multi-focus image fusion algorithms.

5.1. Parameter settings

For the proposed multi-focus image fusion approach, a 1D Gaussian

blur filter $w = \{0.05, 0.20, 0.50, 0.20, 0.05\}$ is used along rows and columns for blurring. Quality parameter $\lambda = 0.95$, block size $m \times n = 15 \times 15$, and threshold $\tau = 8$ are fixed in all experiments. The parameter λ controls the amount of blur that the CAB algorithm may induce and it may be considered a quality control parameter. The difference map of the original and the CAB-blurred images is computed and the threshold τ is used to categorize the pixels as focused or defocused. The values of parameter λ and τ are selected empirically. We tested several images and analyzed the quality degradation in the focused and the de-focused regions with increasing blur iterations. With each blur iteration, the change in ρ in the de-focused regions is very little, whereas a significant drop observed in the focused regions. This means the difference values in the de-focused region are less sensitive to λ. However, varying λ mainly affects the difference values in the focused region. From experiments, we noted that the difference values in the de-focused region are usually very small, not greater than 8 when λ was varied between 0.93 to 0.97. Therefore, in experimental evaluations we used $\tau = 8$ and $\lambda = 0.95$.

5.2. Datasets used for performance evaluation

The performance of the proposed algorithm is evaluated on two multi-focus image datasets including Grayscale and Lytro [36,53]. The Grayscale dataset consists of 14 widely used multi-focus image pairs. Lytro multi-focus dataset consists of 20 multi-focused color image pairs. The size of all source images is 520×520 pixels. Figs. 7 and 8 show few pairs of multi-focus images from the Grayscale and Lytro datasets. In each image pair, the left image is near focused and the right image is far focused.

5.3. Compared methods

To show the effectiveness of the proposed image fusion algorithm, we performed an extensive experimentation and comparison with state-of-the-art image fusion algorithms on both datasets. In particular, we chose 15 representative image fusion algorithms including the recent state-of-the-art to compare the performance of the proposed algorithm. We briefly introduce these algorithms.

Wavelet-based Statistical Sharpness Measure (WSSM) [63] exploits the spreading of the wavelet coefficients to estimate image blur. Discrete Cosine Harmonic Wavelet Transform (DCHWT) [62] uses lifting-based wavelets to reduce the computational complexity of the conventional convolution-based wavelets. Image Fusion using luminance Gradients IFGD [64] is based on blending the gradients of the luminance components of the input multi-focus images and then Haar wavelet-based image reconstruction technique is used to obtain the fused image. In PCA [21], wavelets and Principal Component Analysis (PCA) are used to obtain fused image.

The DCTV [34] method uses variance calculated in Discrete Cosine Transform (DCT) domain. A closely related technique based on Laplacian Pyramids is DCTLP [33]. Orientation information based Pulse Coupled Neural Networks (PCNN) [68,69] is a recent image fusion approach. Independent Component Analysis (ICA) and Topographic Independent Component Analysis bases are used in ICA image fusion algorithm [41].

MSTSR [35] algorithm combines multi-scale transforms with sparse representation to fuse multi-focus images. The spatial domain image fusion algorithm SRCF [36] also uses sparse representation with a learned dictionary. Guided Image Filtering (GIF) [66] uses energy of Laplacian to detect the sharp changes in the intensity of the images to obtain the decision map for image fusion. In CSR [65], each source image is decomposed into a base layer and a detail layer using convolutional sparse representation (CSR).

Cross Bilateral Filter (CBF) [40] obtains the fused image using the fusing weights computed by cross bilateral filtering of source images. Image Fusion using Matting (IFM) algorithm [67] uses morphological filtering and image matting techniques to detect the focused region in multi-focus images, which are combined together to obtain the fused image. Multi-scale Weighted Gradient-based Fusion (MWGF) algorithm [44] proposed an image structure saliency measure to detect the sharp regions at different image scales.

5.4. Qualitative evaluation

The qualitative evaluation of the multi-focus image fusion algorithms is done by comparing the visual quality of the fused images. Therefore, we present the visual comparison of the results achieved by our algorithm and the previous methods. From Lytro dataset, 'Fence' and 'Girl' image sets are chosen and from Grayscale dataset 'Plane' and 'Bottle' source images are selected for visual comparison. Each image set has different level of details and defocused blur. For clearer visual inspection, in each experiment, we magnify selected areas from the foreground focused and background focused regions in the fused images.

Fig. 9 shows the results of proposed algorithm and other best performing image fusion approaches on 'Fence' image set from Lytro dataset. The DCT based approaches, DCHWT [62], DCTLP [33] show ringing artifact around the strong edges, see the magnified fence regions in this example. The DCTV [34] algorithm also distorts the strong edges. The fence edges in the results of CSR [65] and WSSM [63] are also distorted at few places. The fused image produced by PCA [21] is not sharp. The results of IFM [67], CBF [40], GIF [66], and IFGD [64] algorithms suffer from blur artifact around the fence edges. It can be noted that the results obtained by our proposed algorithm are free from such artifacts.

In the second experiment, detailed results of 'Girl' image set are shown in Fig. 10. Two partial regions, background focused (building) and foreground focused (girl's shoulder) are magnified to show the visual details. Similar to the previous experiment, in the results of DCHWT [62] and DCTLP [33] algorithms spurious responses around the magnified building regions can be noted. The girl's shoulder as shown in green color magnified square in the results of IFM [67], PCA, CBF [40], GIF [66], and IFGD [64] algorithms suffer from blur artifact. In the results of IFGD [64], PCA, and DCTV [34] approaches, the sharp edge of the roof and the background buildings in the magnified red square is blurred. The results of WSSM [63] and CSR [65] near the girl's shoulder are better than the other methods. The fusion results of our method are comparable at some locations while significantly better at most of the locations than the compared methods.

On 'Bottle' multi-focus image set (Fig. 11) the results of WSSM [63] and PCA suffer from blur artifact. The ringing artifact around the small bottle in the results of DCTLP [33] and DCHWT [62] can be noticed. The shadow of the cogwheel in the fused images produced by IFM [67] and CBF [40] demonstrate blur artifact and are not in focus. The contour of the cogwheel in the results of DCTV [34] is distorted at few places. The background in the fused image obtained by IFGD [64] is

Fig. 7. Sample multi-focus image pairs from Grayscale dataset.

Fig. 8. Sample multi-focus image pairs from Lytro dataset.

significantly distorted. The fused image obtained by our algorithm demonstrates best visual quality compared with other competing methods.

Fig. 16 presents visual comparison of image fusion algorithms on 'Plane' image set from Grayscale dataset. The contours of the near plane are distorted in the fusion images produced by WSSM [63], ICA [41], and DCTV [34] methods as shown in the red square magnified regions in the respective images. These contours in the results of DCTLP [33], PCA, IFGD [64] and CBF [40] are blurred. The background in the fusion results of CSR [65] is quite blurred and lost the fine details particularly around the near plane region. Compared to these results, the fused image obtained by our method is free of such artifacts and exhibits better visual quality.

Figs. 12 and 13 present two more visual comparison from Lytro dataset. Figs. 14 and 15 show fusion results of various algorithms on two more image sets from Grayscale dataset. Similar observations can be made in these results as well. The visual comparison of our proposed algorithm with other image fusion methods reveals that the proposed algorithm can be used to obtain better quality fused images. To further verify the effectiveness of the proposed fusion method, objective evaluation is also performed and the results are described in the following section.

5.5. Objective evaluation

Objective Quality Assessment (QA) of fused images is a difficult task since in general the ground truth of multi-focus images cannot be acquired. The source images are usually used to evaluate the quality of the fused image.

5.5.1. Image fusion quality metrics

Numerous image fusion QA algorithms have been introduced which exploit various image characteristics such as image contrast, structure similarity and mutual information to assess the quality of the fused images. We choose five well-accepted image fusion QA metrics: Petrovic's metric QAB/F [70], Normalized Mutual Information (NMI) [71], Feature Mutual Information (FMI) [72], Yang's metric (Q_Y) [73], and Nonlinear Correlation Information Entropy (Q_{NCIE}) [74] to compare the quality of the proposed algorithm with current state-of-the-art methods.

Petrovic's metric (QAB/F) [70] is a gradient based fusion metric. It measures the amount of gradient information transferred from the source images to the fused image. QAB/F is calculated as:

$$QAB/F = \frac{\sum_{x=1}^{W} \sum_{y=1}^{H} Q^{AF}(x, y) w^A(x, y) + Q^{BF}(x, y) w^B(x, y)}{\sum_{x=1}^{W} \sum_{y=1}^{H} (w^A(x, y) + w^B(x, y))} \tag{16}$$

where W and H are the width and the height of the image. $QAF(x, y) = Q_g^{AF}(x, y) Q_\alpha^{AF}(x, y)$, where $Q_g^{AF}(x, y)$ and $Q_\alpha^{AF}(x, y)$ are the edge and orientation information at pixel (x, y). The weights w^A and w^B control the relative importance of Q^{AF} and Q^{BF} respectively.

Normalized Mutual Information (NMI) measures the mutual dependance between two random variables [71]. In particular, in image fusion it estimates the information shared between the source images and the fused image. The instability of the metric was overcome in [75]. NMI metric is calculated as:

(a) DCHWT[62] (b) PCA[21] (c) ICA[41] (d) WSSM[63] (e) IFGD[64] (f) DCTLP[33]

(g) CSR[65] (h) DCTV[34] (i) GIF[66] (j) CBF[40] (k) IFM[67] (l) Ours

Fig. 9. Visual comparison of various image fusion methods on 'Fence' source images of Lytro dataset.
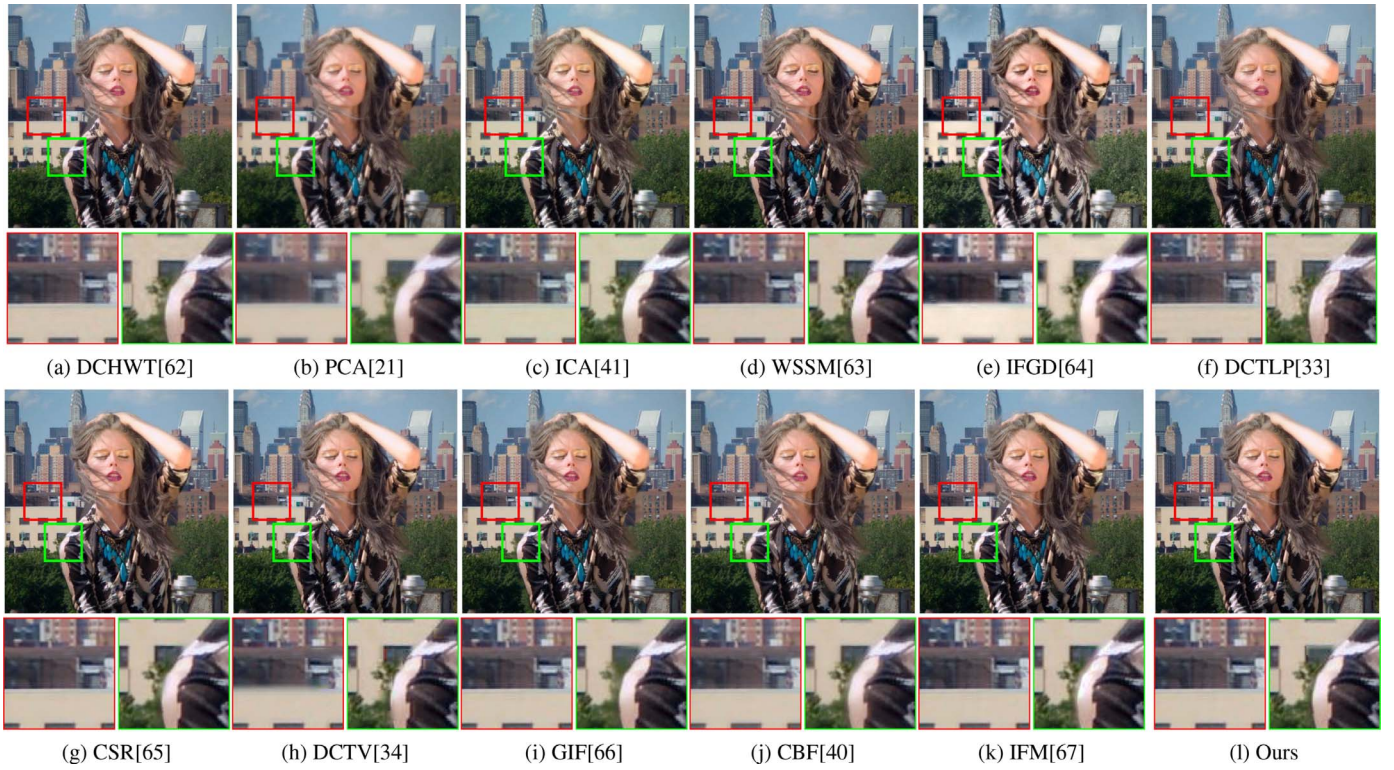
Fig. 10. Visual comparison of various image fusion methods on 'Model Girl' source images of Lytro dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$NMI = \frac{MI(A, F)}{H(A) + H(F)} + \frac{MI(B, F)}{H(B) + H(F)}, \quad (17)$$

where A and B are the source images, F is the fused image. $MI(A, F)$ and $MI(B, F)$ represent the mutual information between images A, F and B, F. $H(\cdot)$ represents the entropy of the image.

Feature Mutual Information (FMI) [72,76] exploits visual image features and utilizes the mutual information to estimate the quality of the fused images. Given the multi-focus source images A and B, and the fused image F, FMI computes the feature image using gradient operator. The Fast-FMI [76] uses the corresponding windows of the source and the fused images to compute the regional FMIs by sliding a window over the entire image. The mutual information between A and F is
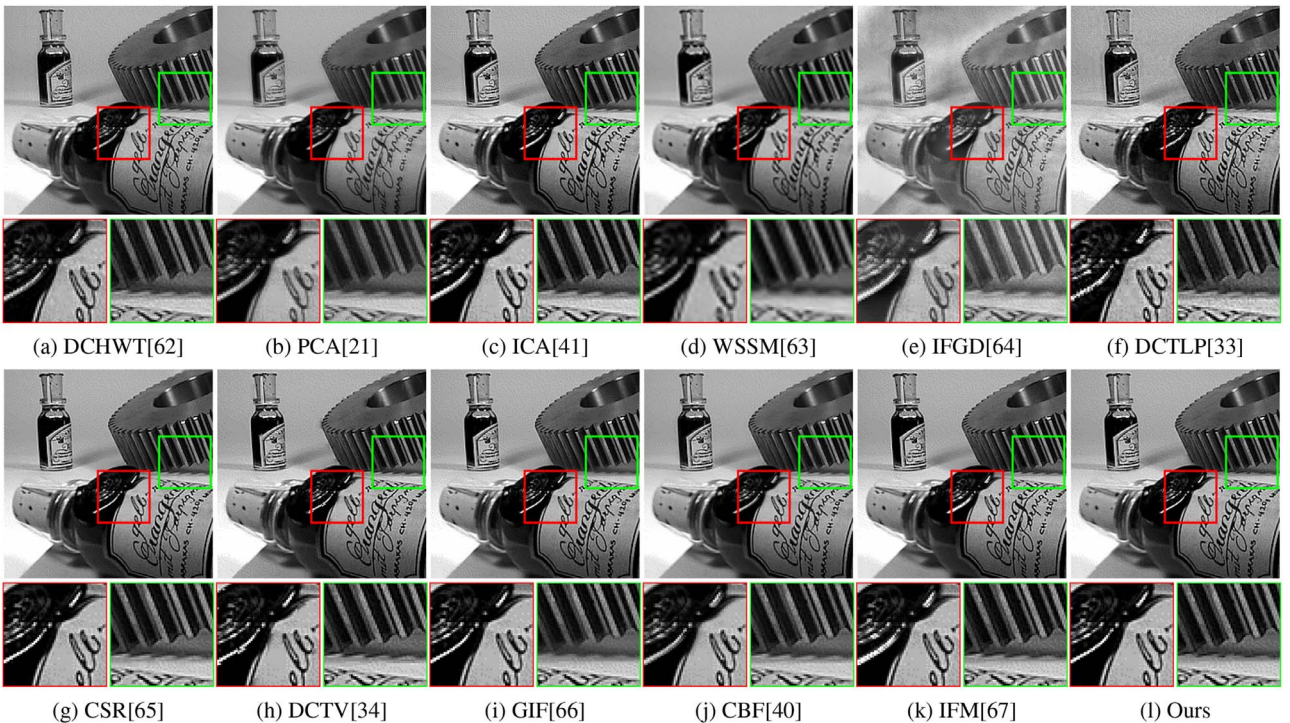

Fig. 11. Visual comparison of various image fusion methods on 'Bottle' source images of Grayscale dataset.
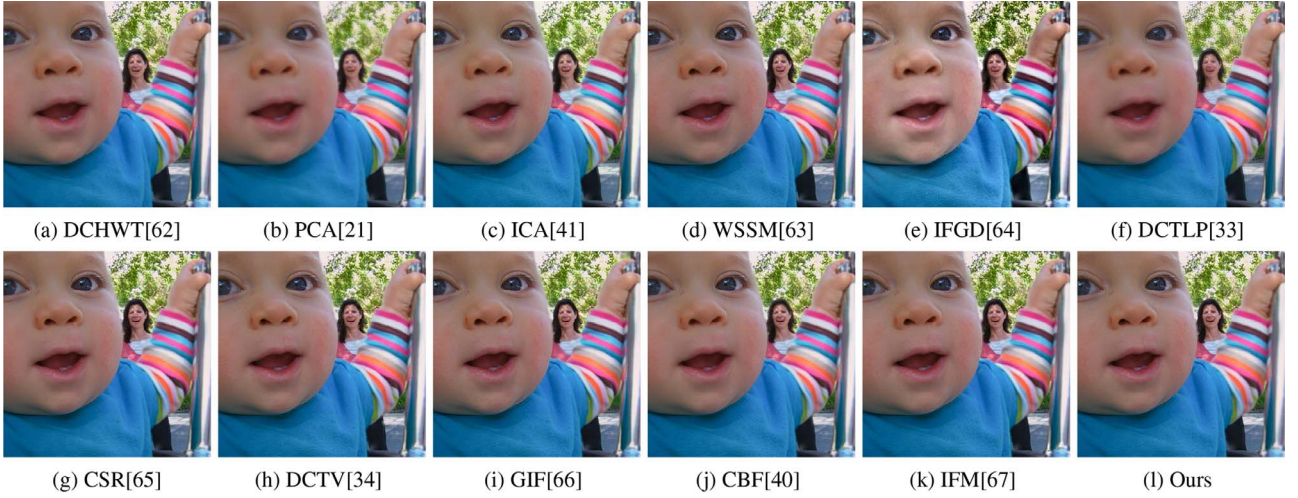
Fig. 12. Visual comparison of various image fusion methods on 'Baby' source images of Lytro dataset.

computed as average local information:

$$I(A, F) = \frac{1}{n} \sum_{i=1}^{n} \frac{I_i(A, F)}{H_i(A) + H_i(F)}, \tag{18}$$

The $I(A, F)$ and $I(B, F)$ are averaged to compute the *FMI* metric

$$FMI = \frac{1}{2}[I(A, F) + I(B, F)]. \tag{19}$$

Yang's metric (Q_Y) [73] is a structural similarity (SSIM) [49] based fusion quality metric. It measures the amount of structural information retained in the fused image F from the source images A and B.

$$Q_Y = \begin{cases} \lambda(w) \ \mathrm{SSIM}(A, F|w) + (1 \\ \qquad - \lambda(w)) \ \mathrm{SSIM}(B, F)|w), \\ \qquad\qquad\qquad\qquad\qquad \text{if } \mathrm{SSIM}(A, B|w) \geq 0.75 \\ \max(\mathrm{SSIM}(A, F|w), \mathrm{SSIM}(B, F|w)), \\ \qquad\qquad\qquad\qquad\qquad \text{if } \mathrm{SSIM}(A, B|w) < 0.75 \end{cases}$$

where w is a local window and $\lambda(w)$ is

$$\lambda(w) = \frac{s(A|w)}{s(A|w) + s(B|w)},$$

where s is a local measure of image saliency.

Nonlinear Correlation Information Entropy (Q_{NCIE}) [74,77] is an information theory-based quality metric. It uses the nonlinear

correlation coefficient (NCC) between the source images A, B and the fused image F and constructs a nonlinear correlation matrix R:

$$R = \begin{bmatrix} 1 & NCC_{AB} & NCC_{AF} \\ NCC_{BA} & 1 & NCC_{BF} \\ NCC_{FA} & NCC_{FB} & 1 \end{bmatrix}$$

Q_{NCIE} is then computed as:

$$Q_{NCIE} = 1 + \sum_{i=1}^{3} \frac{\lambda_i}{3} \log_{256} \frac{\lambda_i}{3} \tag{20}$$

where λ_i are the eigenvalues of the matrix R.

5.5.2. Objective evaluation results and comparisons

We use QAB/F, *NMI*, *FMI*, Q_Y, and Q_{NCIE} image fusion quality metrics to objectively compare the quality of the fused images produced by the proposed and the existing algorithms. The default parameters reported in the respective papers are used. The metrics are computed for all multi-focus image pairs in each dataset and the results are averaged. These metrics are computed using the image fusion QA toolbox [78].

The average scores achieved by the proposed and the compared fusion methods on Lytro dataset are reported in Table 1. The results of the proposed algorithm are compared with 15 current state-of-the-art algorithms. All algorithms were executed with default parameters reported in the respective publications and using the codes made

Fig. 13. Visual comparison of various image fusion methods on 'Cookie' source images of Lytro dataset.
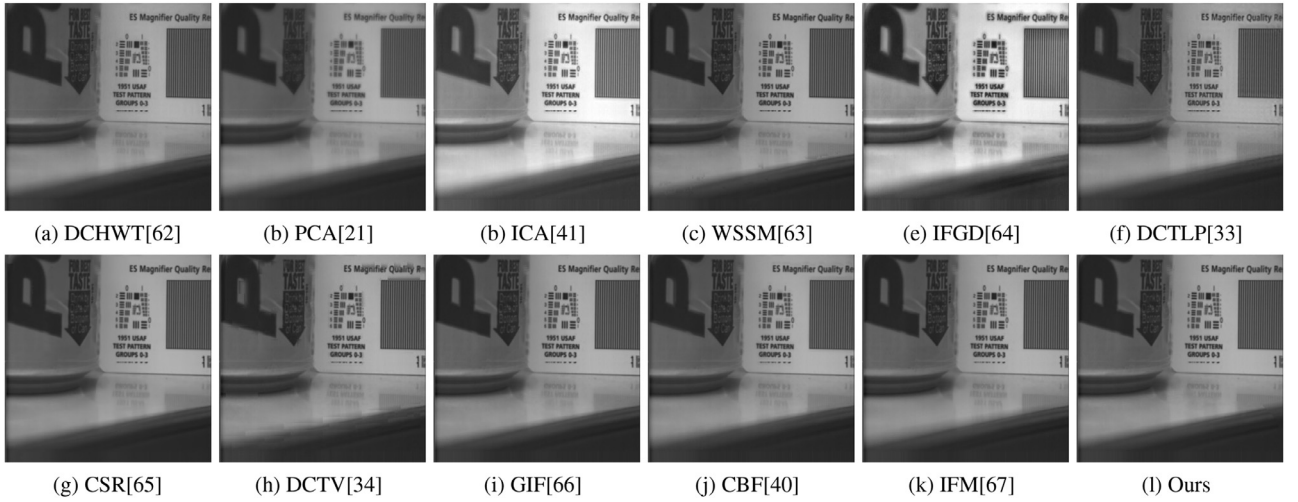
(a) DCHWT[62]	(b) PCA[21]	(b) ICA[41]	(c) WSSM[63]	(e) IFGD[64]	(f) DCTLP[33]

(g) CSR[65]	(h) DCTV[34]	(i) GIF[66]	(j) CBF[40]	(k) IFM[67]	(l) Ours

Fig. 14. Visual comparison of various image fusion methods on 'Pepsi' source images of Grayscale dataset.

(a) DCHWT[62]	(b) PCA[21]	(c) ICA[41]	(d) WSSM[63]	(e) IFGD[64]	(f) DCTLP[33]

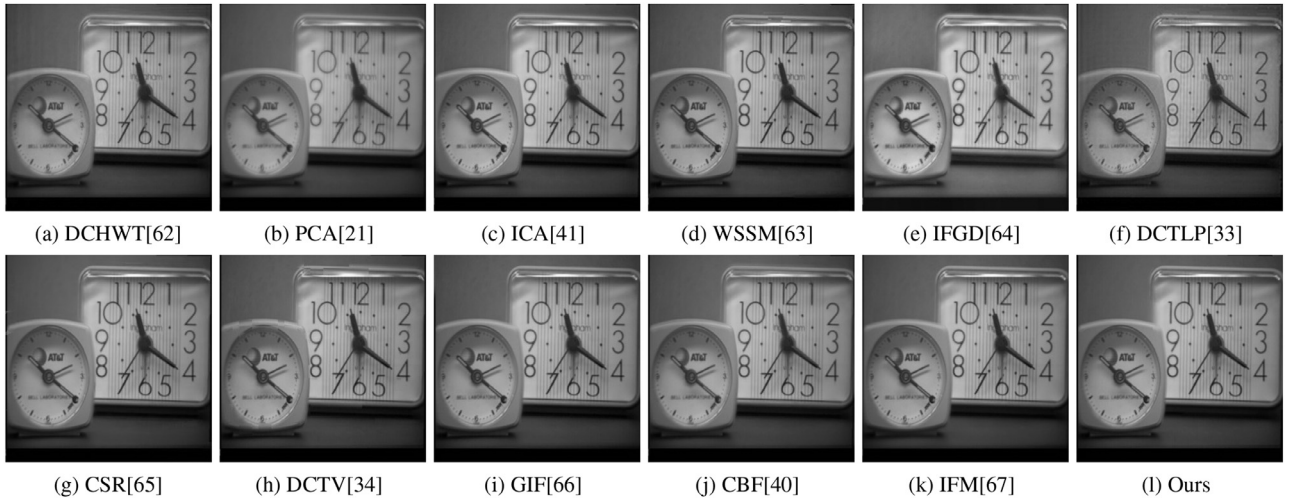| (g) CSR[65]	(h) DCTV[34]	(i) GIF[66]	(j) CBF[40]	(k) IFM[67]	(l) Ours

Fig. 15. Visual comparison of various image fusion methods on 'Clock' source images of Grayscale dataset.

available by the original authors. The results presented in Tab. 1 show that on Lytro dataset the proposed algorithm outperformed all competing image fusion methods in four quality metrics $Q^{AB/F}$, *NMI, FMI* and Q_Y achieving 0.7645, 1.2097, 0.6626 and 0.9895 average scores, respectively. In Q_{NCIE} metric, the PCNN fusion method performs the best with average score 0.8482; our method achieved the second highest average score 0.8474. The performance of PCNN in the other four quality metrics is poor compared to the proposed method.

The objective evaluation results on Grayscale dataset are presented in Table 2. Similar to the previous experiments, the performance of the proposed fusion method on Grayscale dataset is excellent in all quality metrics. It performed the best compared to the competing algorithms. In particular, it achieved average score of 0.7391, 1.2485, 0.6840, 0.9872, and 0.8458 in $Q^{AB/F}$, *NMI, FMI, Q_Y*, and Q_{NCIE} metrics, respectively. Thus, the subjective and objective evaluations show that the performance of the proposed algorithm is excellent for increasing depth-of-view.

5.6. Comparison of computational efficiency

In Table 3 we compare the average execution time of the proposed algorithm and the existing methods on both Grayscale and Lytro datasets. The source codes of the compared methods are obtained from the original authors. All algorithms are implemented in Matlab and executed on the same machine with 3.4 GHz Intel Core i7 processor and 8GB RAM. The reported execution time includes the algorithm execution time and the source images read time, while the write time of resultant fused images is not included.

Table 3 shows that the sparse representation based methods, including CSR, MSTSR, and SRCF are the most time consuming fusion algorithms due to the high computational cost involved in sparse coding. The CSR takes 270 s on average to process a pair of multi-focus images. The MSTSR and SRCF require around 87 and 36 s respectively to fuse a multi-focus image pair. The WSSM is also among the high computational cost fusion methods. In WSSM, wavelet decomposition on each source image is performed and detailed subbands are computed using locally adaptive Laplacian mixture models. The high computational cost of WSSM is mainly due to the use of Laplacian mixture models in which EM algorithm is used to iteratively estimate the model parameters.

The MWGF, DCHWT, IFM, and ICA are in the middle and their computational time is between 4 to 8 s. The rest of the methods require less than 2 s to fuse two images. The PCA, GIF, DCTLP, and IFGD take less than one second to process a pair of images. The average processing time of our proposed method is 1.87 s. The CAB algorithm in the proposed method is very fast due to Algorithm 2. The main cost in the proposed method is incurred by the post-processing steps which involve morphological operations and graph based segmentation. An efficient
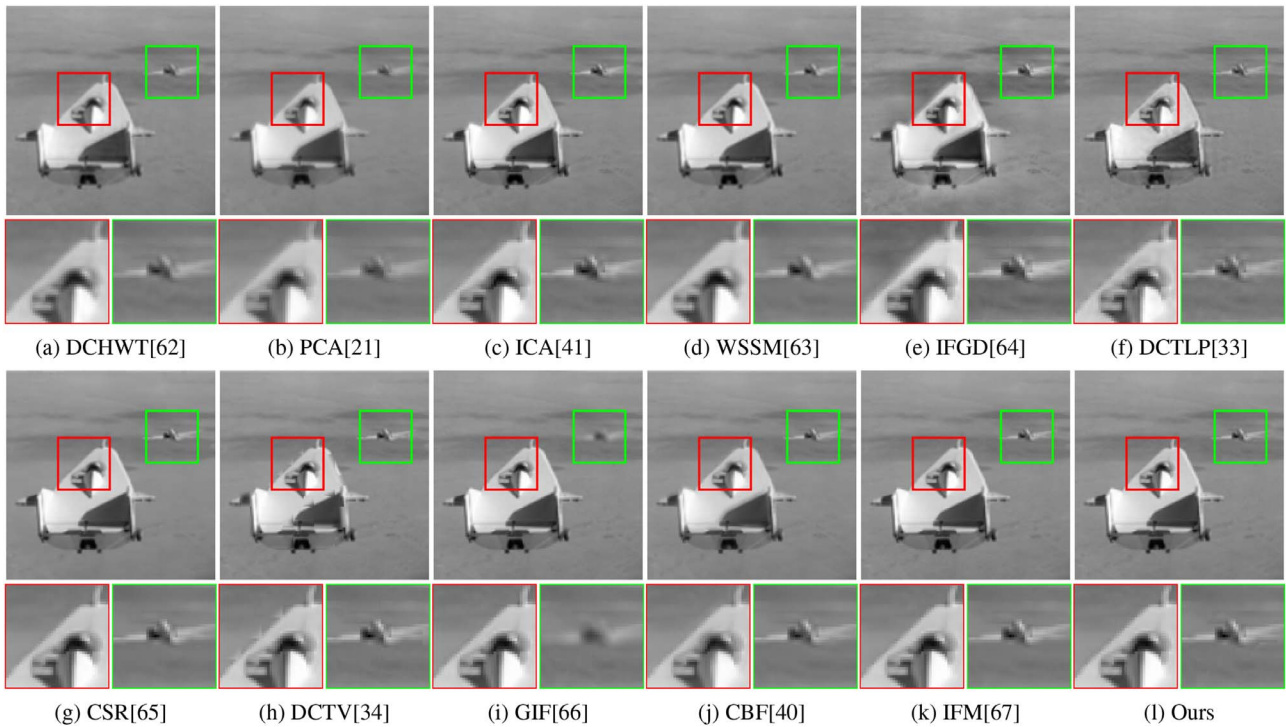
Fig. 16. Visual comparison of various image fusion methods on 'Plane' source images of Grayscale dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Objective comparison of the proposed algorithm with other image fusion approaches on Lytro dataset. The proposed and the best results are shown in bold.

Metric	Fusion Method															
	DCHWT	WSSM	PCNN	DCTLP	MSTSR	CBF	GIF	DCTV	SRCF	IFGD	ICA	PCA	IFM	MWGF	CSR	Ours
$Q^{AB/F}$	0.7124	0.7296	0.7036	0.6526	0.7543	0.7528	0.7608	0.7530	0.7628	0.7174	0.7445	0.5992	0.7534	0.7555	0.7422	**0.7645**
NMI	0.8971	0.9623	1.2068	0.8296	0.9995	1.0184	1.1853	1.1860	1.1930	1.0456	0.9374	0.8939	1.1420	1.1479	1.0135	**1.2097**
FMI	0.5481	0.5732	0.6354	0.5018	0.6081	0.6072	0.6612	0.6333	0.6623	0.5387	0.5834	0.5707	0.6543	0.6527	0.5575	**0.6626**
Y_c	0.9280	0.9594	0.9690	0.8821	0.9675	0.9680	0.9889	0.9657	0.9892	0.8554	0.9555	0.8483	0.9861	0.9884	0.9402	**0.9895**
Q_{NCIE}	0.8275	0.8301	**0.8482**	0.8235	0.8323	0.8349	0.8468	0.8428	0.8465	0.8136	0.8286	0.8259	0.8440	0.8427	0.8327	0.8474

implementation of the proposed method may significantly reduce its computational time. Further qualitative evaluations and comparisons can be found on the paper web site http://www.di.unito.it/~farid/Research/CAB.html, which could not be presented here to limit the length of the paper.

6. Conclusion

In multi-focus image fusion, sharp regions are identified in a set of images of the same scene with different focus levels. These regions are merged together to obtain a single composite image in which all objects are in focus, thus increasing the depth-of-field. In this paper, a novel multi-focus image fusion algorithm based on Content Adaptive Blurring

(CAB) is proposed. The CAB iteratively blurs the local regions based on the underlying contents. Each pixel may receive different amount of blur which is then used to segment the focused and defocused regions. Morphological operators are used to remove the false positives and include the false negatives in the initial estimated map. We employ a graph cut technique to further refine the region maps. A pixel level fusion rule is then applied to obtain the fused image. Qualitative and quantitative comparisons with 15 current state-of-the-art methods on two publicly available datasets using five quality metrics demonstrate excellent performance of the proposed algorithm.

Table 2
Objective comparison of the proposed fusion algorithm with other image fusion approaches on Grayscale dataset. The proposed and the best results are shown in bold.

Metric	Fusion Method														
	DCHWT	WSSM	PCNN	DCTLP	MSTSR	CBF	GIF	DCTV	IFGD	ICA	PCA	IFM	MWGF	CSR	Ours
$Q^{AB/F}$	0.5954	0.5518	0.6802	0.4534	0.7208	0.7185	0.7375	0.7226	0.6716	0.6981	0.6119	0.7380	0.6983	0.7082	**0.7391**
NMI	0.8316	0.8352	1.1684	0.5978	0.9891	1.0071	1.2135	1.1879	1.1198	0.9010	0.9585	1.1968	1.1082	1.0031	**1.2485**
FMI	0.4504	0.4445	0.6538	0.4741	0.6092	0.5951	0.6812	0.6510	0.5313	0.5788	0.5953	0.6683	0.6363	0.5260	**0.6840**
Y_c	0.7845	0.7668	0.9568	0.8211	0.9347	0.9233	0.9866	0.9391	0.8026	0.9068	0.8587	0.9756	0.9396	0.8825	**0.9872**
Q_{NCIE}	0.8238	0.8246	0.8433	0.8234	0.8315	0.8316	0.8439	0.8401	0.8142	0.8274	0.8280	0.8421	0.8379	0.8309	**0.8458**

Table 3
Average execution time (s) comparison of different fusion methods.

Method	DCHWT	WSSM	PCNN	DCTLP	MSTSR	CBF	GIF	DCTV	SRCF	IFGD	ICA	PCA	IFM	MWGF	CSR	Ours
Time	6.3556	135.4343	1.4485	0.2978	86.9946	72.1860	0.2450	1.7112	36.2301	0.8364	8.5740	0.0345	6.5809	4.0684	270.1112	1.8691

References

[1] L. DAndrès, J. Salvador, A. Kochale, S. Süsstrunk, Non-parametric blur map regression for depth of field extension, IEEE Trans. Image Process. 25 (4) (2016) 1660–1673.
[2] J.B. Campbell, R.H. Wynne, Introduction to Remote Sensing, Guilford Press, 2011.
[3] A.A. Goshtasby, S. Nikolov, Guest editorial: image fusion: advances in the state of the art, Inf. Fusion 8 (2) (2007) 114–118.
[4] Z. Wang, Y. Ma, Medical image fusion using m-PCNN, Inf. Fusion 9 (2) (2008) 176–185.
[5] F. Nencini, A. Garzelli, S. Baronti, L. Alparone, Remote sensing image fusion using the curvelet transform, Inf. Fusion 8 (2) (2007) 143–156.
[6] G. Qu, D. Zhang, P. Yan, Medical image fusion by wavelet transform modulus maxima, Opt. Express 9 (4) (2001) 184–190.
[7] G. Simone, A. Farina, F. Morabito, S. Serpico, L. Bruzzone, Image fusion techniques for remote sensing applications, Inf. Fusion 3 (1) (2002) 3–15.
[8] S. Li, X. Kang, Fast multi-exposure image fusion with median filter and recursive filter, IEEE Trans. Consum. Electron. 58 (2) (2012) 626–632.
[9] K. Amolins, Y. Zhang, P. Dare, Wavelet based image fusion techniques an introduction, review and comparison, ISPRS-J. Photogramm. Remote Sens 62 (4) (2007) 249–263.
[10] H. Ghassemian, A review of remote sensing image fusion methods, Inf. Fusion 32 (Part A) (2016) 75–89.
[11] Z. Wang, D. Ziou, C. Armenakis, D. Li, Q. Li, A comparative analysis of image fusion methods, IEEE Trans. Geosci. Remote Sens. 43 (6) (2005) 1391–1402.
[12] R.S. Blum, Z. Liu, Multi-sensor Image Fusion and its Applications, CRC press, 2005.
[13] S. Li, X. Kang, L. Fang, J. Hu, H. Yin, Pixel-level image fusion: a survey of the state of the art, Inf. Fusion 33 (2017) 100–112.
[14] P. Burt, E. Adelson, The laplacian pyramid as a compact image code, IEEE Trans. Comput. 31 (4) (1983) 532–540.
[15] V.S. Petrovic, C.S. Xydeas, Gradient-based multiresolution image fusion, IEEE Trans. Image Process. 13 (2) (2004) 228–237.
[16] Z. Liu, K. Tsukada, K. Hanasaki, Y. Ho, Y. Dai, Image fusion by using steerable pyramid, Pattern Recognit. Lett. 22 (9) (2001) 929–939.
[17] H. Li, B. Manjunath, S. Mitra, Multisensor image fusion using the wavelet transform, Graph. Model Image Process. 57 (3) (1995) 235–245.
[18] J. Tian, L. Chen, Adaptive multi-focus image fusion using a wavelet-based statistical sharpness measure, Signal Process. 92 (9) (2012) 2137–2146.
[19] R. Redondo, F. Šroubek, S. Fischer, G. Cristóbal, Multifocus image fusion using the log-gabor transform and a multisize windows technique, Inf. Fusion 10 (2) (2009) 163–171.
[20] G. Pajares, J.M. de la Cruz, A wavelet-based image fusion tutorial, Pattern Recognit. 37 (9) (2004) 1855–1872.
[21] V. Naidu, J. Raol, Pixel-level image fusion using wavelets and principal component analysis, Def. Sci. J. 58 (3) (2008) 338.
[22] J.J. Lewis, R.J. OCallaghan, S.G. Nikolov, D.R. Bull, N. Canagarajah, Pixel- and region-based image fusion with complex wavelets, Inf. Fusion 8 (2) (2007) 119–130.
[23] Y. Yang, S. Tong, S. Huang, P. Lin, Dual-tree complex wavelet transform and image block residual-based multi-focus image fusion in visual sensor networks, Sensors 14 (12) (2014) 22408–22430.
[24] M. Yin, P. Duan, W. Liu, X. Liang, A novel infrared and visible image fusion algorithm based on shift-invariant dual-tree complex shearlet transform and sparse representation, Neurocomputing 226 (2017) 182–191.
[25] L. Tessens, A. Ledda, A. Pizurica, W. Philips, Extending the depth of field in microscopy through curvelet-based frequency-adaptive image fusion, IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 1 (2007), pp. I-861–I-864.
[26] S. Li, B. Yang, Multifocus image fusion by combining curvelet and wavelet transform, Pattern Recognit. Lett. 29 (9) (2008) 1295–1301.
[27] E. Candes, L. Demanet, D. Donoho, L. Ying, Fast discrete curvelet transforms, Multiscale Model. Simul. 5 (3) (2006) 861–899.
[28] Q. Zhang, B.l. Guo, Multifocus image fusion using the nonsubsampled contourlet transform, Signal Process. 89 (7) (2009) 1334–1346.
[29] L. Yang, B. Guo, W. Ni, Multimodality medical image fusion based on multiscale geometric analysis of contourlet transform, Neurocomputing 72 (1–3) (2008) 203–211.
[30] M. Qiguang, W. Baoshu, A novel image fusion method using contourlet transform,

Int. Conf. Commun., Circuits and Systems, 1 (2006), pp. 548–552.
[31] Y.A.V. Phamila, R. Amutha, Discrete cosine transform based fusion of multi-focus images for visual sensor networks, Signal Process. 95 (2014) 161–170.
[32] L. Cao, L. Jin, H. Tao, G. Li, Z. Zhuang, Y. Zhang, Multi-focus image fusion based on spatial frequency in discrete cosine transform domain, IEEE Signal Process. Lett. 22 (2) (2015) 220–224.
[33] V. Naidu, B. Elias, A novel image fusion technique using dct based laplacian pyramid, Int. J. Inventive Eng. Sci. (IJIES) (2013) 2319–9598.
[34] M.B.A. Haghighat, A. Aghagolzadeh, H. Seyedarabi, Multi-focus image fusion for visual sensor networks in DCT domain, Comput. Electr. Eng. 37 (5) (2011) 789–797.
[35] Y. Liu, S. Liu, Z. Wang, A general framework for image fusion based on multi-scale transform and sparse representation, Inf. Fusion 24 (2015) 147–164.
[36] M. Nejati, S. Samavi, S. Shirani, Multi-focus image fusion using dictionary-based sparse representation, Inf. Fusion 25 (2015) 72–84.
[37] Y. Liu, S. Liu, Z. Wang, A general framework for image fusion based on multi-scale transform and sparse representation, Inf. Fusion 24 (2015) 147–164.
[38] B. Yang, S. Li, Multifocus image fusion and restoration with sparse representation, IEEE Trans. Instrum. Meas. 59 (4) (2010) 884–892.
[39] L. Chen, J. Li, C.L.P. Chen, Regional multifocus image fusion using sparse representation, Opt. Express 21 (4) (2013) 5182–5197.
[40] B.K.S. Kumar, Image fusion based on pixel significance using cross bilateral filter, Signal Image Video Process. 9 (5) (2015) 1193–1204.
[41] N. Mitianoudis, T. Stathaki, Pixel-based and region-based image fusion schemes using ICA bases, Inf. Fusion 8 (2) (2007) 131–142. Special Issue on Image Fusion: Advances in the State of the Art.
[42] S. Li, X. Kang, J. Hu, Image fusion with guided filtering, IEEE Trans. Image Process. 22 (7) (2013) 2864–2875.
[43] Y. Liu, S. Liu, Z. Wang, Multi-focus image fusion with dense SIFT, Inf. Fusion 23 (2015) 139–155.
[44] Z. Zhou, S. Li, B. Wang, Multi-scale weighted gradient-based fusion for multi-focus images, Inf. Fusion 20 (2014) 60–72.
[45] I. De, B. Chanda, Multi-focus image fusion using a morphology-based focus measure in a quad-tree structure, Inf. Fusion 14 (2) (2013) 136–146.
[46] M. Li, W. Cai, Z. Tan, A region-based multi-sensor image fusion scheme using pulse-coupled neural network, Pattern Recognit. Lett. 27 (16) (2006) 1948–1956.
[47] S. Li, B. Yang, Multifocus image fusion using region segmentation and spatial frequency, Image Vis. Comput. 26 (7) (2008) 971–979.
[48] N. Cvejic, D. Bull, N. Canagarajah, Region-based multimodal image fusion using ica bases, IEEE Sensors J. 7 (5) (2007) 743–751.
[49] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[50] L. Zhang, L. Zhang, X. Mou, D. Zhang, Fsim: a feature similarity index for image quality assessment, IEEE Trans. Image Process. 20 (8) (2011) 2378–2386.
[51] Z. Wang, A.C. Bovik, A universal image quality index, IEEE Signal Process. Lett. 9 (3) (2002) 81–84.
[52] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[53] Lytro Multi-focus Dataset, http://mansournejati.ece.iut.ac.ir/content/lytro-multi-focus-dataset, (Acc. Oct. 31, 2016).
[54] R.C. Gonzalez, R.E. Woods, et al., Digital Image Processing, Prentice hall, Upper Saddle River, NJ, 2002.
[55] P. Soille, Morphological Image Analysis: Principles and Applications, 2nd edition, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
[56] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (11) (2001) 1222–1239.
[57] Y. Boykov, G. Funka-Lea, Graph cuts and efficient n-d image segmentation, Int. J. Comput. Vis. 70 (2) (2006) 109–131.
[58] B. Peng, L. Zhang, D. Zhang, A survey of graph theoretical approaches to image segmentation, Pattern Recognit. 46 (3) (2013) 1020–1038.
[59] C. Rother, V. Kolmogorov, A. Blake, "grabcut": interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (3) (2004) 309–314.
[60] K. Suzuki, I. Horiba, N. Sugie, Linear-time connected-component labeling based on sequential local operations, Comput. Vis. Image Understanding 89 (1) (2003) 1–23.
[61] F. Chang, C.J. Chen, C.J. Lu, A linear-time component-labeling algorithm using contour tracing technique, Comput. Vis. Image Understanding 93 (2) (2004) 206–220.
[62] B.K.S. Kumar, Multifocus and multispectral image fusion based on pixel significance using discrete cosine harmonic wavelet transform, Signal Image Video Process. 7 (6) (2013) 1125–1143.
[63] J. Tian, L. Chen, Adaptive multi-focus image fusion using a wavelet-based statistical sharpness measure, Signal Process. 92 (9) (2012) 2137–2146.
[64] S. Paul, I.S. Sevcenco, P. Agathoklis, Multi-exposure and multi-focus image fusion in gradient domain, J. Circuit Sys. Comp. (2016) 1650123.
[65] Y. Liu, X. Chen, R.K. Ward, Z.J. Wang, Image fusion with convolutional sparse representation, IEEE Signal Process. Lett. 23 (12) (2016) 1882–1886.

[66] K. Zhan, J. Teng, Q. Li, J. Shi, A novel explicit multi-focus image fusion method, J. Inf. Hiding Multimedia Signal Process. 6 (3) (2015) 600–612.

[67] S. Li, X. Kang, J. Hu, B. Yang, Image matting for fusion of multi-focus images in dynamic scenes, Inf. Fusion 14 (2) (2013) 147–162.

[68] X. Qu, C. Hu, J. Yan, Image fusion algorithm based on orientation information motivated pulse coupled neural networks, 7th World Congress on Intelligent Control and Automation, (2008), pp. 2437–2441.

[69] X.B. Qu, J.W. Yan, H.Z. Xiao, Z.Q. Zhu, Image fusion algorithm based on spatial frequency-motivated pulse coupled neural networks in nonsubsampled contourlet transform domain, Acta Autom. Sin. 34 (12) (2008) 1508–1514.

[70] C. Xydeas, V. Petrovic, Objective image fusion performance measure, Electron. Lett. 36 (4) (2000) 308–309.

[71] G. Qu, D. Zhang, P. Yan, Information measure for performance of image fusion, Electron. Lett. 38 (7) (2002) 1.

[72] M.B.A. Haghighat, A. Aghagolzadeh, H. Seyedarabi, A non-reference image fusion metric based on mutual information of image features, Comput. Electr. Eng. 37 (5) (2011) 744–756.

[73] C. Yang, J.Q. Zhang, X.R. Wang, X. Liu, A novel similarity based quality metric for image fusion, Inf. Fusion 9 (2) (2008) 156–160.

[74] Q. Wang, Y. Shen, J.Q. Zhang, A nonlinear correlation measure for multivariable data set, Physica D 200 (3–4) (2005) 287–295.

[75] M. Hossny, S. Nahavandi, D. Creighton, Comments on 'information measure for performance of image fusion, Electron. Lett. 44 (18) (2008) 1066–1067.

[76] M. Haghighat, M.A. Razian, Fast-fmi: Non-reference image fusion metric, IEEE 8th Int. Conf. App. Inf. Commun. Technol. (AICT), (2014), pp. 1–3.

[77] Q. Wang, Y. Shen, J. Jin, Performance Evaluation of Image Fusion Techniques, Image Fusion: Algorithms and Applications, Elsevier, 2008, pp. 469–492. Ch. 19.

[78] Z. Liu, E. Blasch, Z. Xue, J. Zhao, R. Laganiere, W. Wu, Objective assessment of multiresolution image fusion algorithms for context enhancement in night vision: a comparative study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (1) (2012) 94–109.

[79] M.S. Farid, M. Lucenteforte, M. Grangetto, DOST: a distributed object segmentation tool, Multimedia Tools Appl. (2017), http://dx.doi.org/10.1007/s11042-017-5546-4.