Chapter 3

Reduced parameter sensitive edge-aware semantic image filtering

3.1 Introduction

Although, the filtering technique presented in the previous chapter provides satisfactory results for a wide varieties of images, it has a critical parameter that needs to be define manually for incorporating appropriate semantic information of the pixels. To establish whether a pixel is an edge or non-edge, the technique used a fixed size square window to compute the skewness of local histogram of the gradient image for incorporating semantic information. So, the semantic edge-map generated by this technique is highly sensitive to the size of this window, which is varied from image to image. Moreover, for varying scale textural images, a fixed size window will not be capable of capturing appropriate semantic information. Thus, the technique may failed to provide good results for such images. Furthermore, to produce the appropriate edge-map for the input image, the technique presented in the previous chapter iterates some steps. The optimal number of iterations is varied from image to image and need to be fixed manually.

In order to mitigate above problems, in this chapter, we exploits Jensen

Shannon Divergence (JSD) to incorporate semantic information for edge-map generation. Here we presented two approaches for generating the semantic edge-map by JSD. In the first approach the JSD is directly used to distinguish structural edge and non-edge pixels. In the second approach by exploiting JSD we proposed some novel features for discriminating structural edge and non-edge pixels. The first approach has one significant parameter that needs to be set manually for different images. On the other hand, the second approach reduced the parameter selection into four discrete options, which make it significantly less parameter dependent. Both the approaches generated the semantic edge-map in single iteration.

The key motivation of this approach is taken from the fact that if a line is passed through the different objects on a perfectly smoothed image as shown in Figure 3-1(a), the intensity distribution of the pixels on the image associated to the different portions of the line either follow step distribution or uniform distribution. The pixels associated to the portions of the line on the boundary regions of two objects follow 'Z' or reverse 'Z' type step distribution as displayed in Figures 3-1(b), and 3-1(d). The portions on a single object follow uniform distribution as displayed in Figure 3-1(c). In the proposed work this fact is incorporated as semantic information for determining whether the pixel is located on the structural edge or not. This chapter makes the following key contributions:

- It exploits JSD to propose novel features for better discriminating structural and textural edge pixels.
- It proposes a less parameter sensitive edge-aware filtering technique.

The next part of this chapter is organized as follows: Section 3.2 details the proposed technique. Section 3.3 discusses and analyzes the experimental findings. Finally, Section 3.4 concludes the chapter.

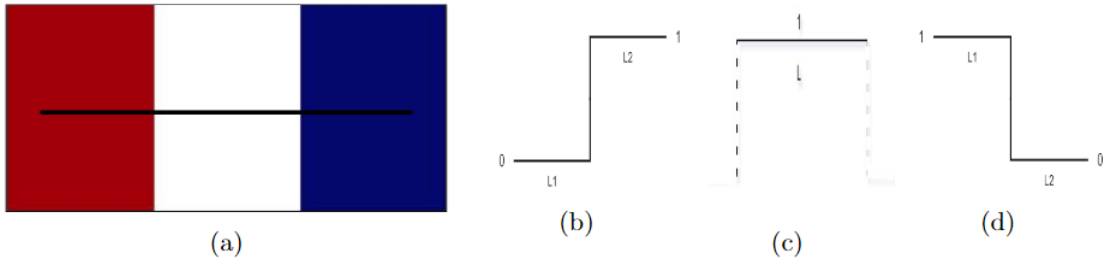


Figure 3-1: (a) A perfectly smooth image and a line passes through different objects on it, (b) (c) (d) intensity distribution of the pixels on the image associated to the different portions of the line.

3.2 Proposed filtering technique

In this section, a novel structure preserving semantic texture filtering technique is presented. The technique proposed two different approaches to generate the semantic edge-map of the input image by exploiting JS divergence. To remove impulse noises and enhance the discrimination between textural and structural edges on the input image I, first, our proposed technique applied morphological opening ($\delta_{SE}(\varepsilon_{SE}(I))$ and closing ($\varepsilon_{SE}(\delta_{SE}(I))$ operations on I by using a 3×3 structuring element (SE) and generate the pre-processed image J as follows:

$$J = (I + \delta_{SE}(\varepsilon_{SE}(I)) + \varepsilon_{SE}(\delta_{SE}(I)))/3 \tag{3.1}$$

Once the preprocessed image J is obtained, next our proposed technique generates semantic edge-map of the input image.

3.2.1 Generation of semantic edge-map

The success of edge-aware filtering techniques is dependent on the identification of appropriate structural edge pixels on the image. To this end, some recent research focuses on the generation of an edge-map that retains only the structural edges of the input image. Identification of appropriate structural edges of an image is a difficult task and still a challenging research topic. In this subsection, we propose

two different approaches to generate semantic edge-map of the original image I by exploiting Jensen Shannon Divergence (JSD). In the first approach, JSD metric is directly used and in the second approach novel discriminating features are extracted by exploiting JSD to generate the semantic edge-map for the input image. The particulars of both the approaches are outlined in the following subsections.

3.2.2 Approach I: Generation of semantic edge-map by direct use of JSD metric [110]

In probability theory, the JSD is widely used to measure the dissimilarity between two probability distributions [86]. The lower the JSD score, the closer the two distributions are to one another. It is an extension of Kullback-Leibler Divergence (D_{KL}) to calculate a symmetrical score between two distributions. Let P and Q be the probability distribution of a discrete random variable, the Jensen Shannon-Divergence $JSD(P||Q)$ between the pair (P, Q) is calculated as follows:

$$\begin{cases} JSD(P||Q) & = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M) \\ where, & \\ M & = \frac{1}{2}(P+Q) \\ D_{KL}(P||M) & = -\sum P\log\frac{M}{P}, \\ D_{KL}(Q||M) & = -\sum Q\log\frac{M}{Q} \end{cases} \tag{3.2}$$

Where $D_{KL}(P||M)$ computes the Kullback-Leibler Divergence between P and M. In the proposed work, to determine whether a pixel is on a structural edge or not, first, a fixed size square window is defined by considering it as a center pixel. Then within the window, several lines in different directions passing through the center pixel are considered (as shown in Figure 3-2). If spatial distributions of all such lines (i.e., the intensity values of the pixels on the line) are similar to a uniform distribution, then the center pixel is considered to be a non-edge pixel. On the

other hand, if the spatial distribution of any of the lines is more similar to a step distribution then it is considered to be a structural edge pixel. In the proposed work two discrete distributions viz; a step distribution and a uniform distribution are used as reference distributions (as shown in Figure 3-3). The JSD scores computed between the line inside the window and the reference distribution help us to determine whether the pixel is part of the structural edge or not.

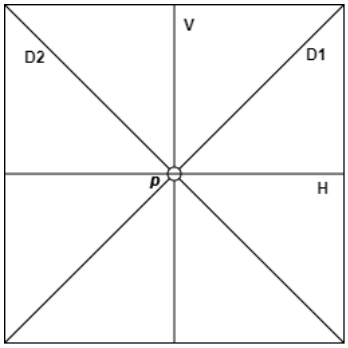


Figure 3-2: Four lines horizontal H, vertical V, diagonal D_1 and D_2 passing through the center pixel p within a square window w.

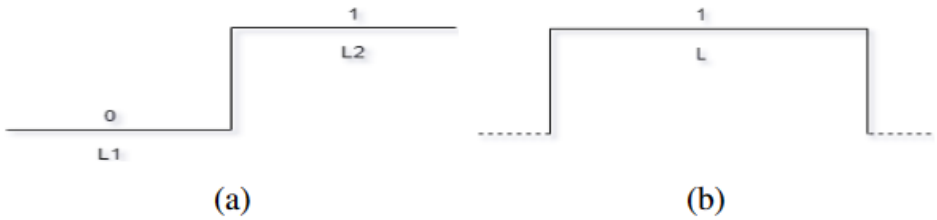


Figure 3-3: Reference (a) step distribution function T_L and (b) uniform distribution function U_L

In greater detail, let J be the preprocessed input image and J_p stands for the intensity value of a pixel p in J. Considering p as the center pixel, a fixed-size square window w is defined. Then inside w four lines viz; horizontal H, vertical V, and two diagonal D_1 and D_2 passing through p are taken (see Figure 3-2). The probability distribution P_H, P_V, P_{D_1} and P_{D_2} associated to the line H, V, D_1, and D_2, respectively is computed as follow:

$$P_L(q) = \frac{|J_q - min(mean(J_{L_1}), mean(J_{L_2}))|}{\sum |J_q - min(mean(J_{L_1}), mean(J_{L_2}))|}, \qquad \forall q \in L_1 \cup L_2 \tag{3.3}$$

Where each line $L = H|V|D_1|D_2$ (i.e., $L = H$ or V or D_1 or D_2) is partitioned into two line segments L_1 and L_2 at center pixel p. From equation (3.3) it is seen that if pixel p is in structural edge, then the probability distribution of at least one line (i.e., P_H, P_V, P_{D_1} or P_{D_2}) is more similar to a discrete step distribution. On the other hand, if p is not in structural edge, then the probability distributions of all four lines are more similar to discrete uniform distributions. In the proposed work for each $P_L, (L = H|V|D_1|D_2)$ two discrete reference functions as shown in Figure 3-3 namely uniform function U_L and step function T_L are defined as follow:

$$\begin{cases} U_L = 1, \ \forall \ pixels \in L \end{cases} \tag{3.4}$$

$$\begin{cases} if \ \ avg(P_{L_1}) > avg(P_{L_2}) \\ T_L = \begin{cases} 1, \ \forall \ pixels \in L_1 \\ 0, \ \forall \ pixels \in L_2 \end{cases} \\ else \\ T_L = \begin{cases} 0, \ \forall \ pixels \in L_1 \\ 1, \ \forall \ pixels \in L_2 \end{cases} \end{cases} \tag{3.5}$$

Where $avg(P_{L_1})$ and $avg(P_{L_2})$ are the average values of P_{L_1} and P_{L_2} respectively. Let for a line $L(i.e., L = H|V|D_1|D_2)$, Q_L and R_L be the probability density function of the step function T_L and the uniform function U_L, respectively. If p is in structural edge then the spatial distribution of at least one line among four lines will be more similar to its step probability density function. Thus, in this work, an edge score S_p^w for the pixel p is computed by taking the minimum JSD between P_L and Q_L as follows:

$$S_p^w = \min_{L=H,V,D_1,D_2} \{JSD(P_L, Q_L)\} \tag{3.6}$$

If p is in structural edge then the edge score S_p^w computed in equation (3.6) will be smaller than the average JSD between P_L and R_L i.e., if $S_p^w <$

$avg_{L\in w}\{JSD(P_L, R_L)\}$ then the pixel p will be in structural edge. Note that one can generate the edge map that contains only the structural edges of the image by applying this criterion for every pixel in J. In that case, the width of structural edges in the edge map will be determined by the considered window size w. To make the width of structural edges invariant to the window w, we also compute the edge score $S_p^{w_5}$ considering a 5×5 window. Then the binary edge map E_b for the image J is generated as follows:

$$E_b(p) = \begin{cases} 1, \ if \ \ S_p^{w_5} < avg_{L\in w_5}\{JSD(P_L, R_L)\} \\ \quad and \ \ S_p^{w} < avg_{L\in w}\{JSD(P_L, R_L)\} \\ 0, \ otherwise \end{cases} \tag{3.7}$$

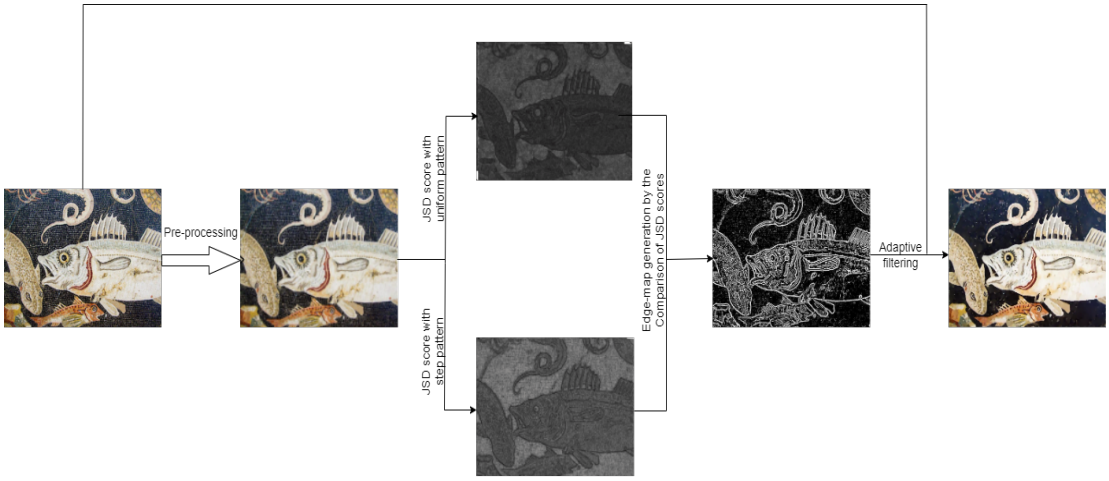


Figure 3-4: Approach I: Overall framework of the developed technique

Figure 3-4 illustrates the overall framework of our developed filtering technique by generating the edge-map using approach I and algorithm 2 shows the precise steps of it.

Algorithm 2 Approach I: Proposed reduced parameter sensitive image filtering technique

Input: Original Image I
Output: Filtered Image I_f
Generate pre-processed image J using equation (3.1)
For a selected window size w
for each pixel $p \in J$
  for each pre-processed image band $J_c \in J$
    Compute the binary edge score $E_b^c(p)$ using equation 3.7
  end for
  $E_b(p) = \mathrm{Max}_c(\mathrm{E}_b^c(p))$
end for
Apply recursive adaptive median filtering using the edge-map E_b to obtain the filtered image I_f as described in Section 2.3.4.

3.2.3 Approach II: Generation of semantic edge-map by extracting discriminating features using JSD

In Approach I, the semantic edge-map generated by the developed technique is heavily determined by the window size w which is defined manually. Note that the optimal size of this window varied from image to image. Moreover, for the images with varying scale textures, a single window will not be capable of providing sufficient semantic information. In order to reduce the parameter dependency as well as to incorporate better semantic information for edge-map generation in the second approach we proposed to extract some novel features by exploiting JSD to discriminate structural edge and non-edge pixels. The overall frame work of our developed filtering technique using Approach II is shown in Figure 3-5.

In this approach, each pixel of the preprocessed image is expressed with a features set. These features are extracted in such a way so that they incorporate semantic information of the pixel. To incorporate appropriate semantic information associated to a pixel, first, considering it as a center pixel, multiple square windows of increasing sizes are taken. Then, within each window, four lines passing through the center pixel in different directions are considered (as shown in Figure 3-2). If the spatial distributions of the pixels on all the four lines are

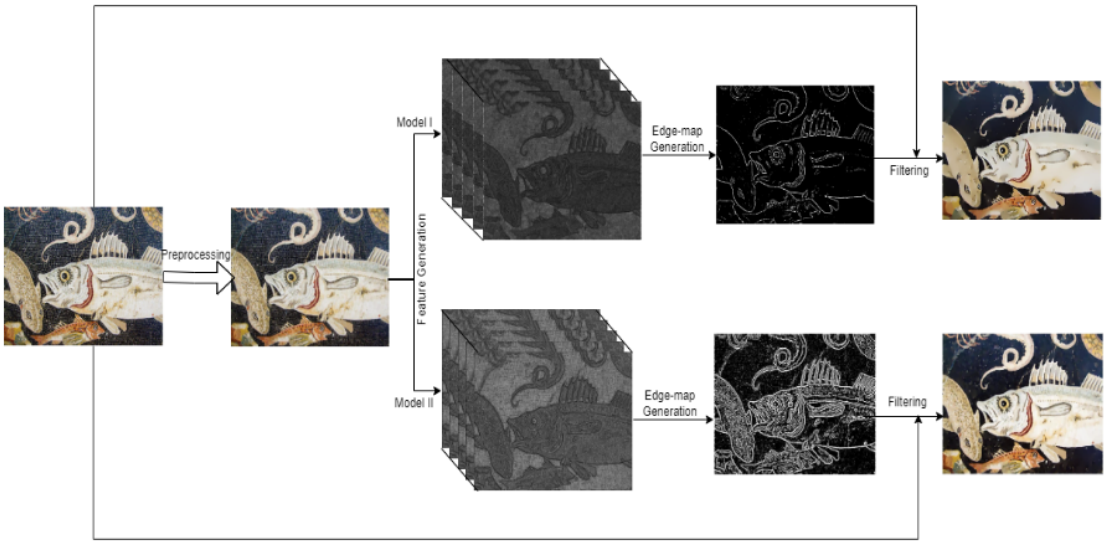


Figure 3-5: Approach II: Overall framework of the developed technique

similar to a uniform distribution, then the center pixel will more probable to be a non-structural edge pixel (i.e., either a non-edge pixel or a textural edge pixel). On the contrary, if the spatial distribution of at least one of these lines is similar to a step distribution then the pixel will more probable to be a structural edge pixel. The JS Divergence between the spatial distributions of the lines passing through the center pixel and the reference distributions are considered as the features of the center pixel. Since features of the pixels are computed by incorporating semantic information of whether they belong to structural edges or not. When the pixels are projected into the feature space, they will form two groups, one for structural edge pixels and the other for non-edge pixels. Thus, a semantic edge-map of is easily obtained by applying a simple clustering technique.

Let the probability density function derived from the uniform function U and the step function T denotes as R_L and Q_L, respectively. If the center pixel p is on structural edge then for at least one $L, (L \in \{H|V|D_1|D_2\})$ (as shown in Figure 3-2) the value of $JSD(P_L, Q_L)$ will be low. Otherwise, for all four lines $JSD(P_L, R_L)$ will be low. To incorporate semantic edge information of pixel p, its features SS_L^p and US_L^p for $L \in \{H|V|D_1|D_2\}$ are computed as follows:

$$SS_L^p = \{JSD(P_L, Q_L)\}, \quad L \in (H, V, D_1, D_2) \tag{3.8}$$

$$US_L^p = \{JSD(P_L, R_L)\}, \quad L \in (H, V, D_1, D_2) \tag{3.9}$$

Moreover, to define gradient feature of the center pixel p, the weighted Gaussian mean of the line segments L_1 and L_2 is computed as follows:

$$Gmean(L_i) = \frac{1}{S} \sum_{q \in L_i} J_q \times e^{-d^2}, \quad i = 1, 2. \tag{3.10}$$

where, the distance metric $d = ||p - q||_1$ and $S = \sum e^{-d^2}$. Then the gradient feature MD_L^p for $L \in \{H|V|D_1|D_2\}$ is defined as:

$$MD_L^p = |(Gmean(J_{L_1}) - Gmean(J_{L_2}))|, \quad L \in (H, V, D_1, D_2) \tag{3.11}$$

Inside window w, considering each of the four lines (i.e., $L = H, L = V, L = D_1$, and $L = D_2$) equations (3.8), (3.9) and (3.10) computes the feature value SS_L^p, US_L^p and MD_L^p, respectively. If p is a structural edge pixel then at least one direction (i.e., for one line) the value of SS_L^p, US_L^p, and MD_L^p will be low, high, and high, respectively. If p is a non-structural edge pixel then all four directions (i.e., for all the four lines) value of US_L^p will be low. To discriminate the structural and non-structural edge pixels into the feature space, in this research we proposed two models by taking different combinations of the features SS_L^p, US_L^p and MD_L^p as follows:

Model I: This model is proposed for texture smoothing. In this model the feature set F_w^1 for the window w is generated by giving more emphasis on US_L^p and MD_L^p than SS_L^p by applying non linear stretching on SS_L^p using the sigmoid function (

$\frac{1}{1+e^{-x}}$) as follows:

$$F_w^1(p) = \begin{cases} (US_L^p + MD_L^p) \times (1 + e^{-(SS_L^p)})\ if\ (SS_L^p <= avg_{L \in w}\{US_L^p\}) \\ 0,\ \ Otherwise \end{cases} \tag{3.12}$$

The four components (features) of F_w^1 that represent a pixel p of the image are generated by considering the lines L as H, V, D_1, and D_2 inside the window w.

Model II : This model is proposed for preserving structural small details. In this model the feature set F_w^2 for the window w is generated by giving more emphasis on SS_L^p than US_L^p and MD_L^p by applying non linear stretching on $(US_L^p + MD_L^p)$ using the sigmoid function ($\frac{1}{1+e^{-x}}$) as follows:

$$F_w^2(p) = \begin{cases} SS_L^p \times (1 + e^{-}(US_L^p + MD_L^p))\ \ if\ \ \ (SS_L^p > avg_{L \in w}\{US_L^p\}) \\ 0,\ \ \ Otherwise \end{cases} \tag{3.13}$$

The four components (features) in F_w^2 that represent a pixel p on the image are generated by considering the lines L as H, V, D_1, and D_2 inside the window w.

In both the models, the feature set generated from a single window will not provide enough semantic information. To capture better semantic information, in this work multiple windows of increasing sizes are taken, and for each window, four features are generated by using Eq. (3.11) or Eq. (3.12). Thus, for a gray scale image considering n number of windows, the developed technique generates $4 \times n$ features and for an image of c bands, it generates $4 \times n \times c$ features. Since features of the pixels are computed by incorporating semantic information of whether they belong to structural edges or not. When the pixels are projected into $R^{4 \times n \times c}$ dimensional feature space, they will be formed two clusters, one for structural edge pixels and the other for non-edge pixels. Thus, applying k-means

clustering with $k = 2$, can easily produce a semantic-aware binary edge-map for the input image. Figure 3-6 displays the edge-maps produced by applying the developed Model I and Model II as well as the popular Sobel operator, Prewitt operator and Canny algorithm. From these figures one can see that the traditional edge detection models are completely failed to discriminate structural and textural edges. Whereas, both the proposed models are showed their robustness to discriminate structural and textural edges. Furthermore, from this figure one can see that Model I is better for texture smoothing, whereas Model II is suitable for preserving small structural details.

3.2.4 Generation of filtered image

Either using Approach I or Approach II once the semantic edge-map is obtained, the recursive edge-aware adaptive median filter developed in Chapter 2 is applied to produce the filtered image. Figure 3-7 shows the semantic edge-maps and the filtered images generated by the approach II of the proposed technique.

3.3 Experimental analysis and results

To validate the proficiency of our developed technique, in this section the filtered images generated by considering the semantic edge-maps produced by Approach I and Approach II are analyzed. The particulars of the experimental findings for each of the Approach are given in the next subsections.

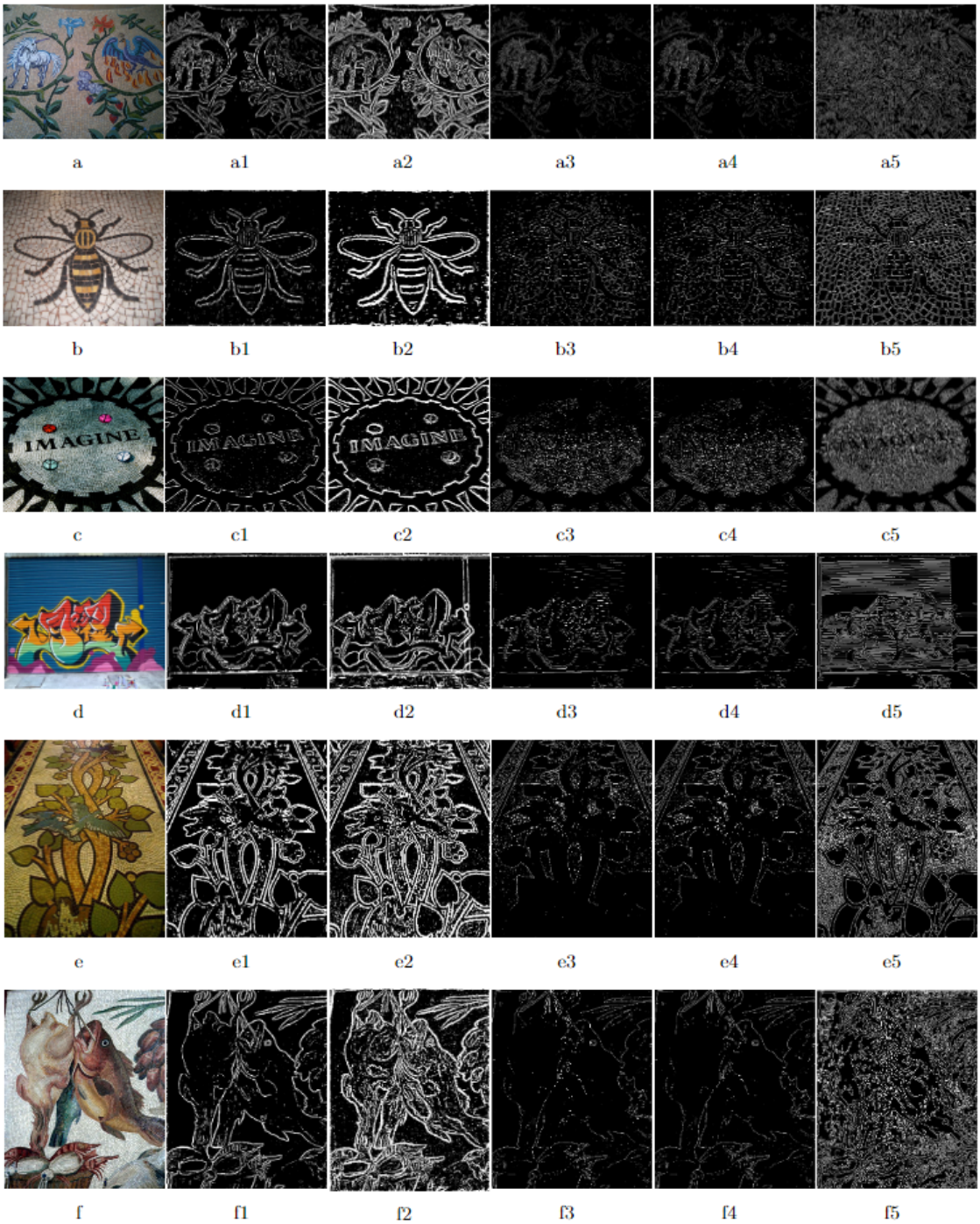


Figure 3-6: Input images (a)(b)(c)(d)(e)(f) and their corresponding edge-maps generated by Model I (a1)(b1)(c1)(d1)(e1)(f1), Model II (a2)(b2)(c2)(d2)(e2)(f2), Sobel operator (a3)(b3)(c3)(d3)(e3)(f3), Prewitt operator (a4)(b4)(c4)(d4)(e4)(f4), and Canny edge detection algorithm (a5)(b5)(c5)(d5)(e5)(f5).

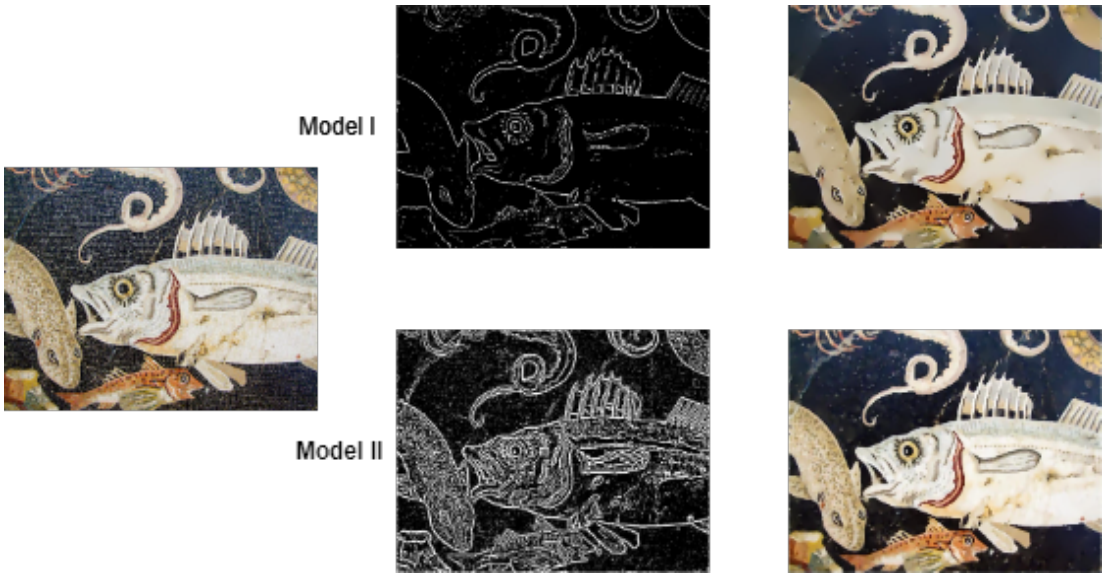


Figure 3-7: Original image of Pompeii Marine Life mosaic, corresponding edge-maps, and filtered images produced by Model I and Model II of the proposed technique

Algorithm 3 Approach II: Proposed reduced parameter sensitive image filtering technique

Input: Original Image I
Output: Filtered Image I_f
Generate pre-processed image J using equation (3.1)
Select W_j, $j = 1|2|3|4$
for each pixel $p \in J$
  for each pre-processed image band $J_c \in J$
    for each window $w \in W_j$
      Compute feature set $F_w^1(p)$ for Model I and $F_w^2(p)$ for Model II by using equation (3.12) and equation (3.13), respectively.
      $F_c^1(p) = \bigsqcup_w F_w^1(p)$ and $F_w^2(p) = \bigsqcup_w F_w^2(p)$
    end for
    $F^1(p) = \bigsqcup_c F_c^1(p)$ and $F^2(p) = \bigsqcup_c F_c^2(p)$
  end for
end for
// $\bigsqcup$ **is vector concatenation operator** //
Project all the pixels into the feature space either using $F^1(p)$ or $F^2(p)$ or both $F^1(p) \bigsqcup F^2(p)$.
Generate edge-map E_b by applying 2-means clustering in the feature space.
Apply recursive adaptive median filtering using the edge-map E_b to obtain the filtered image I_f as illustrated in Section 2.3.4.

3.3.1 Results: Filtered image obtained from the edge-map produced by Approach I

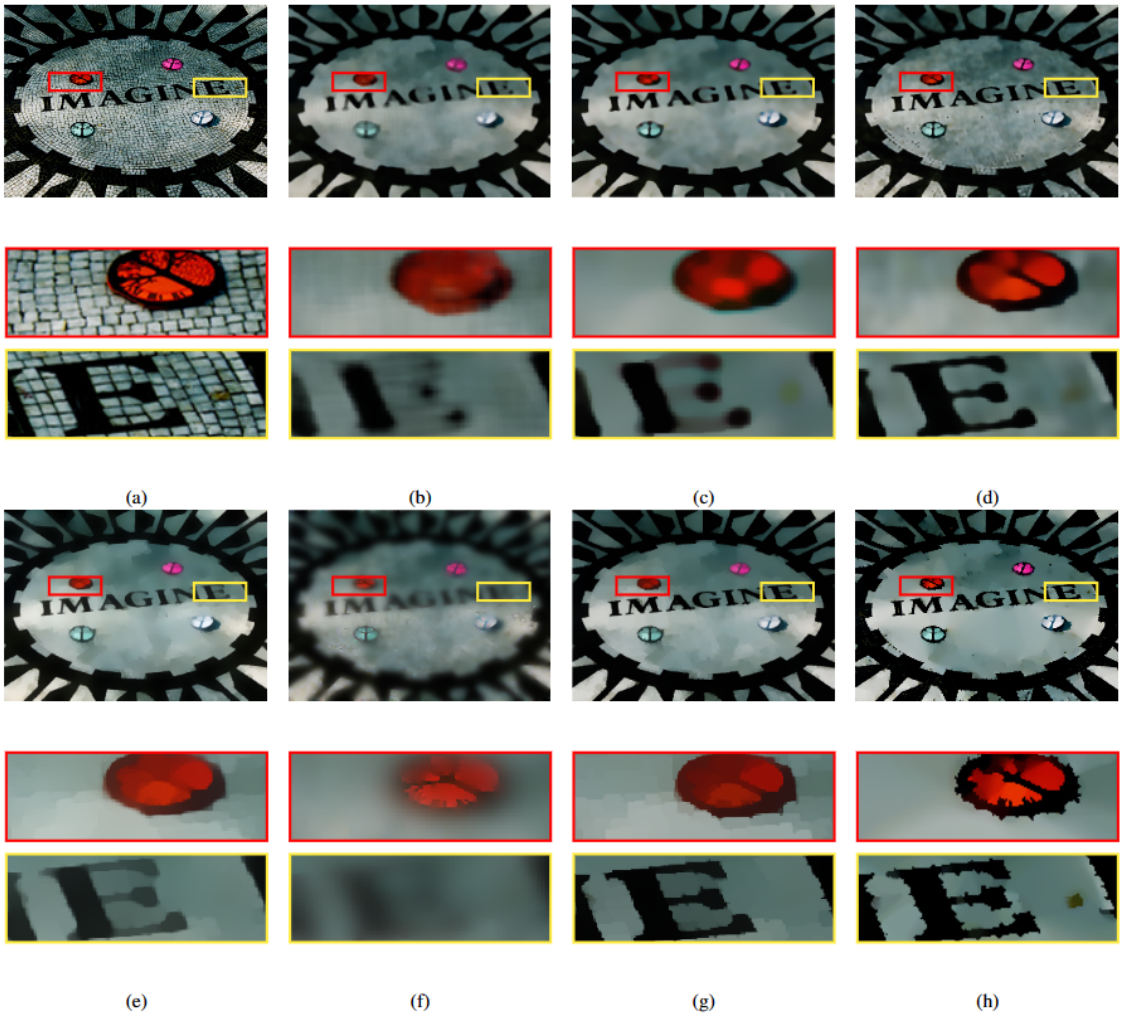


Figure 3-8: Comparative results of a mosaic floor image: (a) Original image and zoomed portions of it. Filtered images with best possible parameters and its respective SSIM Values obtained by (b) Reg-Cov ($k = 15, ps = 6, \sigma = 0.2$) [SSIM = 0.36] [72] (c) BTF ($k = 9, n_{itr} = 7$) [SSIM = 0.37] [32], (d) SATF ($ss = 3, sr = 0.1, st = 0.1, n_{itr} = 7, div = 30$) [SSIM = 0.44] [69], (e) SATV ($\lambda = 1.25$) [SSIM = 0.32] [125], (f) RILS ($\lambda = 1.5, \gamma = 50/255, n_{itr} = 15$) [SSIM = 0.25] [91], (g) GISF ($\alpha = 0.75, \lambda = 1.25, n_{itr} = 10$) [SSIM = 0.45] [92], and (h) Proposed ($w = 11 \times 11, n_{itr} = 5$) [SSIM = 0.51] techniques.

To validate Approach I of the developed technique, the filtered image generated from the edge-map produced by it is compared with the filtered images provided by six structure preserving filtering techniques of state-of-the-art such as Reg-Cov [72], bilateral texture filtering (BTF) [32], scale-aware texture filtering (SATF

) [69], structure adaptive total variation (SATV) [125], real-time iterative least square (RILS) [91], and generalized image smoothing framework (GISF) [92]. The first two (i.e., Reg-Cov [72] and BTF [32]) are textural feature based, next two (i.e., SATF [69] and SATV [125]) are structural edge based and last two (i.e., RILS [91] and GISF [92]) are learning based texture filtering techniques. Comparison with these techniques is carried out qualitatively on the basis of how much they are effective to differentiate texture and structure by smoothing out fine details of textures while preserving the significant structures. Structural Similarity (SSIM) [136] is calculated for a quantitative comparison.

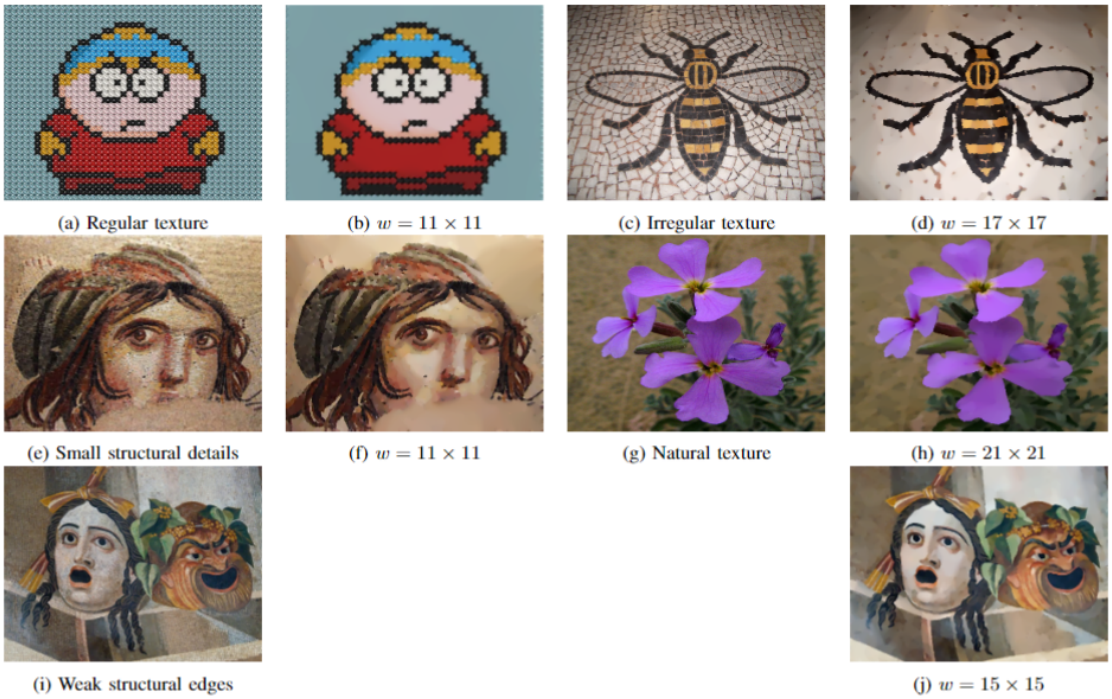


Figure 3-9: (a)(c)(e)(g)(i) Original images with different types of textures and structures, (b)(d)(f)(h)(j) Corresponding filtered images generated by the proposed technique.

Figure 3-8 displays the filtering results attained by the developed as well as the state-of-the-art methods for the floor image. By examining the zoomed-in sections of the filtered images produced by all these methods, it is evident that our method surpasses the other compared methods in both respects: i) texture smoothing and ii) preserving the significant structures. From this figure, it is

apparent that the first two textural feature based methods i.e., Reg-Cov [72] and BTF [32] are performing overall smoothing well but from the zoomed part it is seen that they blurred the edges while preserving the smaller structure. The edge aware techniques, SATF [69] produce poor smoothing results. Whereas SATV [125] blurred the edges of the smaller structure. The RILS [91] and GSF [92] both blurring the smaller structural objects. In contrast, our developed method is equally effective in texture smoothing of varying scales as well as preserving the significant structure of different sizes. In contrast to the other considered methods, instead of blurring, the developed method sharpens the structural edges of the filtered image . It is capable of preserving the small structural details, like alphabets and light sources on the floor image more prominently than the others. The SSIM values are also proclaiming the better preservation of structural similarity by the developed technique. To further evaluate the effectiveness of it, we have applied to a wide variety of images across different categories and sizes of textural patterns and obtained satisfactory results. The filtering results obtained from a few of such images are shown in Figure 3-9. These results show the potentiality of our technique for filtering varieties of images.

Parameter analysis: The semantic edge-map generated by Approach I is heavily determined by the window size w (i.e., the fixed size square window for considering lines in different directions within it) which is defined manually. Note that the optimal size of this window varied from image to image. Moreover, for the images with varying scale textures, a single window will not be capable of providing sufficient semantic information. Thus the success of the technique is significantly dependent on the parameter w and finding its optimal value is difficult task. In Approach II we try to reduce the drawbacks of Approach I by extracting discriminating features using JSD.

3.3.2 Results: Filtered image obtained from the edge-map produced by Approach II

In this section, we first demonstrate our developed technique's robustness with respect to its parameters. Subsequently, we assess the efficiency of the developed technique through qualitative and quantitative outcomes contrasting with multiple state-of-the-art methods.

3.3.2.1 Parameters setting

In this approach for edge-map generation each pixel is expressed with a set of feature and these features are generated by considering multiple windows of increasing size. As the feature values rely on the window size, it is important to consider appropriate windows for generating the features of the pixels. In this work to consider appropriate windows, a subset $\{2, 3, 5, 8, 13, 21, 34\}$ from Fibonacci series is selected. Then by taking four consecutive elements of it, four subsets $W_1 = \{2, 3, 5, 8\}, W_2 = \{3, 5, 8, 13\}, W_3 = \{5, 8, 13, 21\}$ and $W_4 = \{8, 13, 21, 34\}$ are formed. And then looking into the kind of texture present on the input image one of these four subsets is chosen to define the windows for feature generation. For example, if the subset W_2 is chosen, the windows of sizes ($2 \times W_2 + 1$) i.e 7×7, 11×11, 17×17, and 27×27 are used to generate the features of each pixel. In this research, for the input images of having fine-scale textural details W_1 or W_2 and for the image of having large-scale textural details W_3 or W_4 is chosen for feature generation. As a result, in edge-map generation step of the proposed approach only the parameter w, where $w = \{W_1|W_2|W_3|W_4\}$ is needs to be selected manually. The radius of the windows are taken from The Fibonacci numbers as because the series grows exponentially with a non-uniform spacing, but their ratios converges to the golden ratio (approximately 1.618). This work well in generating semantic feature in variety of scales.

In the proposed filtering technique except the parameter w of edge-map generation step, all the other parameters are manually set with fixed values irrespective of the input images. Depending on the kind of input image only the w parameter is required to be set manually among four discrete options. This represents a significant advantage of the our technique over other state-of-the-art methods. It provides satisfactory results for varieties of images with minimal manual fine tune of its parameters. Note that the filtering results of most of the existing state-of-the-art methods are highly relied on one or multiple parameters which are required to fine tune manually within a continuous interval.

3.3.2.2 Qualitative Comparison

To evaluate the performance of our technique, it is contrasted with six state-of-the-art structure-preserving filtering techniques including Reg-Cov [72], BTF [32], SATF [69], SATV [125], RILS [91], and GISF [92]. Figure 3-10 shows the filtered images generated by applying various techniques for a floor image. By looking at the zoomed areas of the image one can observe that contrast to the state-of-the-art techniques the developed technique performed superior in terms of both, structure preserving and texture smoothing. The textural feature based techniques Reg-Cov [72] and BTF [32], and the structural edge-aware techniques SATF [69] and SATV [125], blurred the edges while smoothing smaller structures.

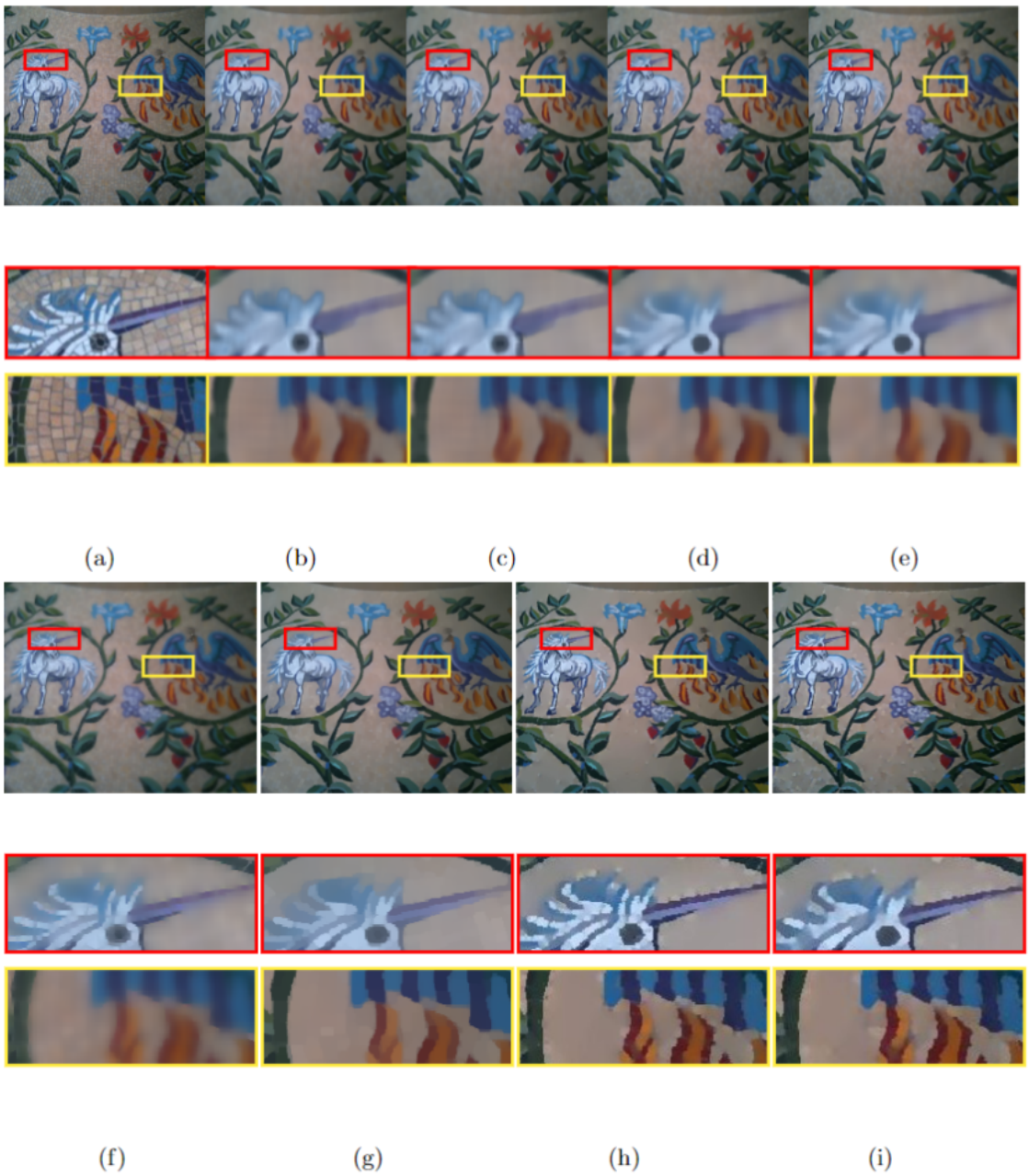


Figure 3-10: Comparative results for Mosaic Floor I image: (a) Input image and some highlighted portions of it. Filtered images produced by applying (b) Reg-Cov ($k = 15, ps = 6, \sigma = 0.2$) [72] (c) BTF, $k = 9, n_{itr} = 7$ [32], (d) SATF $ss = 3, sr = 0.1, st = 0.1, n_{itr} = 7, div = 30$ [69], (e) SATV, $\lambda = 2.5$ [125], (f) RILS $\rho_{smooth} = 3, \rho_{sharp} = 5$ [91], (g) GISF, $\lambda = 50, \gamma = 20/255, n_{itr} = 15$ [92], (h) Proposed Model I, ($w = W_2$), and (i) Proposed Model II, ($w = W_3$) techniques.

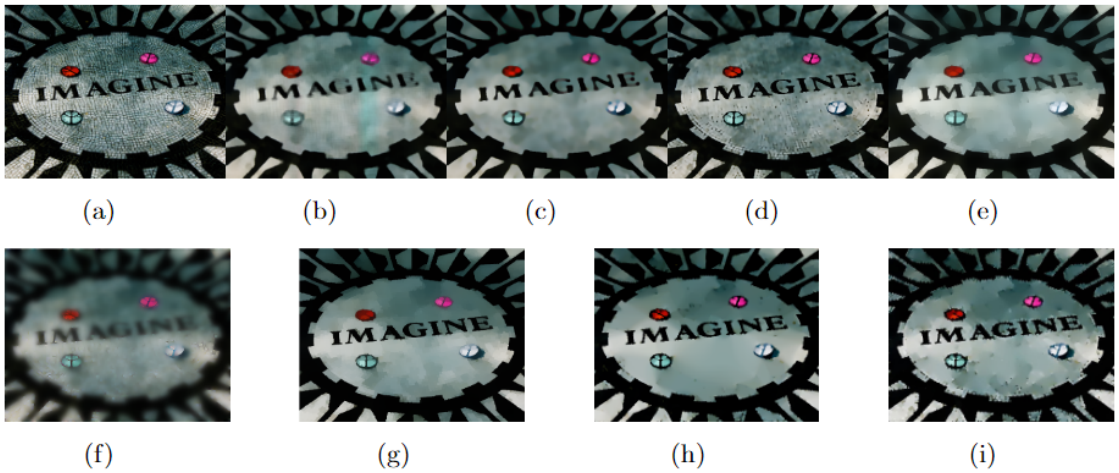


Figure 3-11: Comparative results for Mosaic Floor II image: (a) Input image. Filtered images produced by applying (b) Reg-Cov ($k = 15, ps = 6, \sigma = 0.2$) [72] (c) BTF, ($k = 9, n_{itr} = 7$) [32], (d) SATF ($ss = 3, sr = 0.1, st = 0.1, n_{itr} = 7, div = 30$) [69], (e) SATV, ($\lambda = 2.5$) [125], (f) RILS ($\rho_{smooth} = 3, \rho_{sharp} = 5$) [91], (g) GISF, ($\lambda = 50, \gamma = 20/255, n_{itr} = 15$) [92], (h) Proposed Model I, ($w = W_2$), and (i) Proposed Model II, ($w = W_3$) techniques.

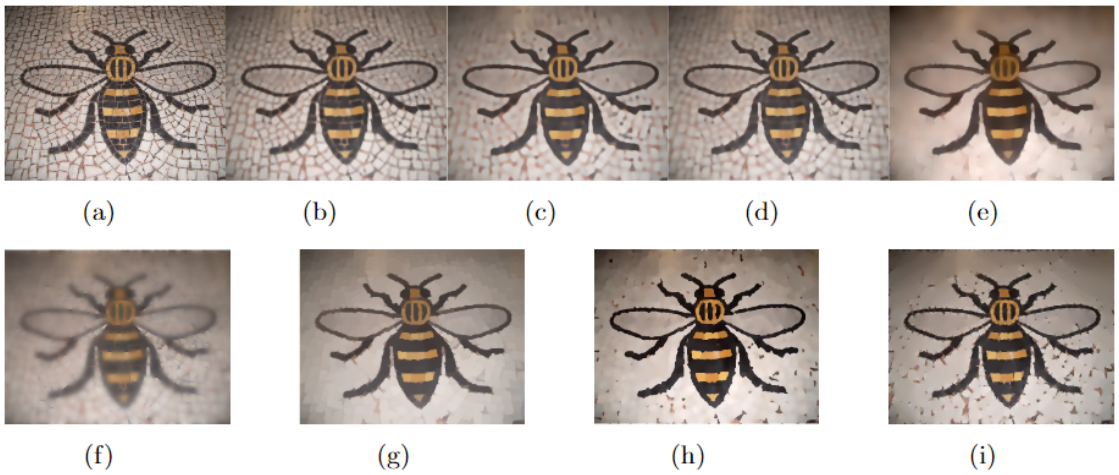
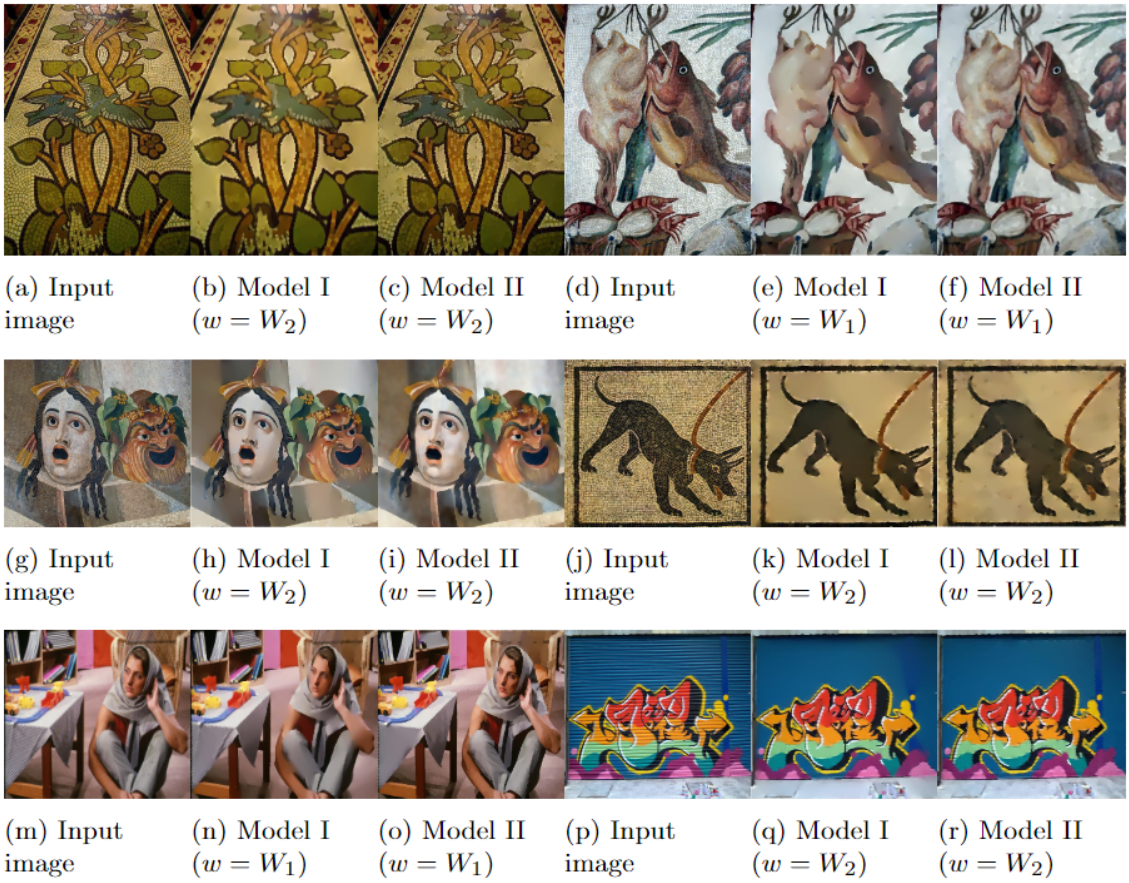


Figure 3-12: Comparative results for Mosaic Floor III image: (a) Input image. Filtered images produced by applying (b) Reg-Cov ($k = 15, ps = 6, \sigma = 0.2$) [72] (c) BTF, ($k = 9, n_{itr} = 7$) [32], (d) SATF ($ss = 3, sr = 0.1, st = 0.1, n_{itr} = 7, div = 30$) [69], (e) SATV, ($\lambda = 2.5$) [125], (f) RILS ($\rho_{smooth} = 3, \rho_{sharp} = 5$) [91], (g) GISF, ($\lambda = 50, \gamma = 20/255, n_{itr} = 15$) [92], (h) Proposed Model I, ($w = W_3$), and (i) Proposed Model II, ($w = W_3$) techniques.



(a) Input image (b) Model I ($w = W_2$) (c) Model II ($w = W_2$) (d) Input image (e) Model I ($w = W_1$) (f) Model II ($w = W_1$)

(g) Input image (h) Model I ($w = W_2$) (i) Model II ($w = W_2$) (j) Input image (k) Model I ($w = W_2$) (l) Model II ($w = W_2$)

(m) Input image (n) Model I ($w = W_1$) (o) Model II ($w = W_1$) (p) Input image (q) Model I ($w = W_2$) (r) Model II ($w = W_2$)

Figure 3-13: (a)(c)(e)(g)(i) Input images with a wide range of diverse textures and structures, and (b)(d)(f)(h)(j) Respective filtered images obtained by applying developed method.

Whereas, Model I and Model II of the proposed approach able to preserve the important structures of various sizes, as well as better smoothing the textures present at varying scales. Moreover, in contrast to the state-of-the-art techniques, both the models of the developed technique produced sharpen filtered images without distorting the structural edges. Small structural details such as alphabets and light sources as shown in Figure 3-11 are better preserved by the developed technique. To establish the robustness of the developed technique, we tested it with a wide varieties of images containing diverse types of regular and irregular textural patterns and always found satisfactory results. Figures 3-10, 3-11, 3-12, and 3-13 show some of the filtered images provided by the developed technique. By looking at these figures, it can be observed the robustness of our technique for smoothing varying scale regular and irregular textures, as well as for preserving significant structural details.

Table 3.1: The scores of the IQA metrics SSIM, MSSIM, MI and PIQE of the filtered images produced by the different techniques. The bold font are indicating the best scores of these IQA metrics.

Images	***Metrics***	***RegRov [72]***	***BTF [32]***	***SATF [69]***	***SATV [125]***	***RILS [91]***	***GISF [92]***	***(Model I, II)***
	SSIM	0.62	0.68	0.68	0.69	0.60	0.72	**0.76,** 0.71
Mosaic Floor I	MSSIM	0.58	0.78	0.79	0.78	0.68	0.79	**0.88,** 0.82
PIQE=16.28	MI	1.74	1.91	1.90	1.97	1.67	1.73	2.16, **2.26**
	PIQE	91.62	89.57	88.70	81.28	100	81.41	**69.14,** 75.10
	SSIM	0.36	0.38	0.43	0.31	0.26	0.44	0.45 **, 0.46**
Mosaic Floor II	MSSIM	0.58	0.61	**0.68**	0.62	0.54	0.62	0.66, **0.68**
PIQE=37.27	MI	2.30	2.27	2.08	2.00	1.91	2.31	2.46, **2.55**
	PIQE	86.78	88.85	87.05	84.74	88.19	86.26	**82.72**, 83.86
	SSIM	0.49	0.44	0.57	0.51	0.47	0.53	**0.64,** 0.59
Mosaic Floor III	MSSIM	0.58	0.74	0.70	0.51	0.50	0.53	**0.78,** 0.70
PIQE=15.18	MI	2.03	2.07	2.50	2.45	2.37	1.89	2.65, **2.75**
	PIQE	86.74	88.86	89.96	86.47	100	87.91	**78.34,** 80.00

3.3.2.3 Quantitative Comparison

To demonstrate the robust performance of the developed technique, in this experiment a quantitative comparison between the filtered images produced by the developed and the state-of-the-art techniques is carried out. Here, for quantitative comparison three subjective IQA metrics like SSIM [136], MSSIM, MI [131] and a subjective no-reference metric PIQE [132] are used, where SSIM and MSSIM both range from [0, 1], and MI ranges from [0, $\log(m \times n)$], where $m \times n$ represents the image size, PIQE ranges from [0, 100]. For all the three SSIM, MSSIM, and MI, higher values indicate better performance, whereas for PIQE, a lower value indicates better quality. The scores of these IQA measures are reported in Table 3.1. From the table it can be observed that both the models of the developed technique yielded superior results contrast to the existing state-of-the art techniques. This once again confirms the robustness of the developed technique..

3.3.2.4 Applications

Structure-preserving filtering techniques find diverse numerous applications in image processing and analysis. These techniques can be employed as either preprocessing or post-processing steps for various tasks, including image denoising, image enhancement, tone mapping, image abstraction, image classification, image segmentation, and more. The following subsections describe some of the key applications of the proposed filtering technique.

Image denoising: In Figure 3-14 it is showing the noisy and corresponding denoised images produced the proposed filtering technique. The filtered images are showing that it preserving structures well while removing noises.

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3-14: (a)(d)(g) Original images, (b)(e)(h) images with Gaussian noise ($\sigma = 0.03$) and their corresponding (c)(f)(i) denoised images obtained by applying proposed filtering technique.

Detail enhancement: Here the images in Figure 3-15 are showing the original images and their corresponding enhanced images using proposed filtering. The filtering technique first decompose the input image into two part- one is smoothed image and other is textural details. By adding the textural details to the origi-

nal image we can get the final enhanced image. The enhanced images are more prominently showing the textural details.

(a) (b)

(c) (d)

(e) (f)

Figure 3-15: (a)(c)(e) Original images and their corresponding (b)(d)(f) enhanced images obtained by the proposed technique.

Tone mapping: Figure 3-16(a)(c) showing LDR image (RGB) tone mapped

from two HDR image by Farbman techniques [47] and Figure. 3-16(b)(d) shows the corresponding tone mapped RGB version produced by using the proposed filtering technique. We have simply replaced the proposed filtering technique in place of bilateral filtering (BF) in Durand's method [49] with $\gamma = 0.5$.

(a) (b) (c) (d)

Figure 3-16: (a)(c) RGB tone mapped images generated by Farbman techniques [47] and (b)(d) RGB tone mapped images generated by integrating proposed filtering technique in [49].

All these above three results on different applications showing that the proposed technique providing satisfactory results. Also by comparing the results for these applications with the previous technique proposed in Chapter 2, it can be seen that for image denoising and image enhancement technique proposed technique in Chapter 2 perform better whereas in tone mapping application proposed technique in this chapter is doing better.

3.3.2.5 Computational analysis

The computational complexity of the developed technique is determined by the semantic-aware edge-map generation and the recursive median filter used to produce the filtered image. For an image of size $M \times N$, if S_1 is the size of the largest window used to extract the features of the pixels, the computational complexity of the generation of semantic-aware edge-map is $\theta(M \times N \times S_1)$. The recursive median filter used by the proposed technique define adaptive window L_w and w_s^a for filtering non-edge and edge pixels, respectively. The complexity of the recursive

median filter is $\theta(M \times N \times S_2)$, where $S_2 = \max\{L_w, w_s^a\}$. Thus the computational complexity of our technique is $\theta(M \times N \times S)$, where $S = \max\{S_1, S_2\}$. table 3.2 is showing the comparative execution time taken by the developed and the state of the art techniques.

Table 3.2: The run time (in seconds) of the state-of-the-art and the technique we developed.

Images	*RegRov [72]*	*BTF [32]*	*SATF [69]*	*SATV [125]*	*RILS [91]*	*GISF [92]*	*Proposed*
Mosaic Floor I (675 × 900)	510.23	177.95	126.34	11.72	6.204	13.49	306.67
Mosaic Floor II (600 × 800)	415.74	153.46	113.65	11.440	6.01	49.99	216.93
Mosaic Floor III (450 × 600)	232.49	101.58	54.38	7.26	3.46	22.44	108.456

3.3.2.6 Selection of optimal model (Model I or Model II)

The proposed techniques have two main different approaches: the first one is the wise manual parameter (windoiw size) selection, and that parameter needs to be selected depending on the texture sizes in the input image. That is for smaller textural pattern removal, window size may be chosen as smaller as 7×7 to 11×11, and for the larger size textural images, it may be taken as 15×15 to 21×21. The second approach is mainly focused on the reduction of this manual parameter selection. This approach has two different models (Model I, Model II) with two different types of feature generation methods. Model I and II have the same parameters set, that is W_1, W_2, W_3, W_4. These parameters $W_1 = \{2, 3, 5, 8\}, W_2 = \{3, 5, 8, 13\}, W_3 = \{5, 8, 13, 21\}$ and $W_4 = \{8, 13, 21, 34\}$ basically represent a range of window sizes, where selecting W_1 or W_2 suitable for the images (like in Figure 3-11) having smaller objects to be preserved and selecting W_3 or W_4 is preferable for the images (like in Figure 3-7) having smaller insignificant objects to be removed. The basic differences between the two models are that Model I is preferable for maximal smoothing, whereas Model II is suitable for preserving most of the objects while smoothing less compared to Model I.

3.4 Conclusions

Structural edge-aware features are more robust than the textural features for filtering images of having varying scale irregular textures. However, the effectiveness of edge-aware filtering techniques dependent on their ability to identify the right structural edges.

In this chapter, we exploits JSD to incorporate semantic information for edge-map generation. We presented two approaches for generating the semantic edge-map. In Approach I, JSD is directly used to distinguish structural edge and non-edge pixels. In Approach II, by exploiting JSD we extracted some novel discriminative features for representing structural edge and non-edge pixels. Then by projecting the pixels into the feature space our approach applied k-means clustering to generate a semantic-aware edge-map. Once the edge-map is obtained either by applying Approach I or Approach II, an edge-aware adaptive recursive median filter is used to generate the filtered image. The first approach has one significant parameter that needs to be set manually for different images. On the other hand, the second approach reduced the parameter selection into four discrete options, which make it significantly less parameter dependent. However, both the approaches generated the semantic edge-map in single iteration.

The performance of the our developed technique is demonstrated by contrasting it with several state-of-the-art methods using a diverse set of images consisting of varying scale regular and irregular textures. The proposed technique can achieve several competing objectives that cannot be accomplished by using a single existing method, such as detecting and smoothing textures, protecting corners and other areas that are easy to overlook, avoiding over-blurring and/or over-sharpening artifacts, and maintaining structural edges.

List of publications from this chapter

Journals

1. Pradhan, K., Patra, S. Reduced parameter sensitive edge-aware semantic image filtering. (***Under review to 'Pattern Analysis and Applications***).

Conference

1. K. Pradhan and S. Patra, "Structure Preserving Semantic Texture Filtering," 2022 IEEE Calcutta Conference (CALCON), Kolkata, India, 2022, pp. 283-287, doi: 10.1109/CALCON56258.2022.10060620.