Chapter 4

A semantic edge-aware parameter efficient image filtering technique

4.1 Introduction

To incorporate semantic information for edge-map generation, the technique presented in Chapter 2 and the Approach I of Chapter 3, both considered a fixed size window which is manually defined by trial and error method. Although for the images of having regular texture it is possible to define a suitable window but it is impossible for varying scale irregular textures. Thus, for the images of having varying scale irregular textures, both the techniques may have failed to incorporate sufficient semantic information to generate the semantic edge-map. In order to mitigate the drawbacks of both the techniques in this chapter, we proposed another parameter efficient filtering technique by exploiting the properties of these two techniques.

To incorporate semantic information for better discriminating the structural and textural edges, in this chapter, we introduce a novel approach. Our method generates the edge-map of the input image by exploiting semantic information in two phases. In the first phase, a semantic gradient image (SGI) is generated by analyzing the local distribution of input image and morphological gradient image. In the second phase, with the help of the generated SGI, the size of the windows are defined to exploit JS Divergence for generating the semantic edge-map. Once the edge-map is obtained, the edge-aware adaptive recursive median filter proposed in Chapter 2 is utilized to produce the filter image. Although the proposed filtering technique required to define multiple windows, the size and shape of most of these windows are either automatically defined or kept fixed, irrespective of the consider input images. There exists only a pair of windows that needed to be fixed manually by choosing one of the four options. Our technique provides satisfactory results for a diverse set of input images with minimal fine-tuning of its parameters, which is an important benefit of the proposed technique in contrast to the current state-of-the-art techniques. This chapter's key contributions are:

- Proposes a novel technique to incorporate semantic information for edge-map generation.
- Proposes an parameter efficient edge-aware texture filtering technique.

The rest of the Chapter is structured as follows: The proposed technique is described in Section 4.2. Section 4.3 analyses and analyzes the experimental findings. In Section 4.4, a conclusion is determined.

4.2 Proposed filtering technique [112]

To reduce the effects from the impulse noises and textural oscillations we used the equation 3.2 defined in Chapter 3 for generating the preprocessed image J from the input image I. Once the preprocessed image J is generated, our developed technique follows two steps to produce the filtered image. In the first step, it proposes a novel method that generates a semantic-aware edge-map of the input image. Then, in the second step, the adaptive recursive median filter proposed in Chapter 2 is used to produce the filter image.

4.2.1 Generation of semantic-aware edge-map

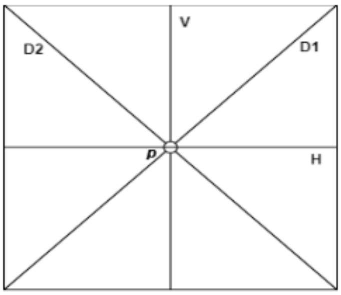


Figure 4-1: A fixed-size square window and the four lines H, V, D_1, and D_2 passing through the center pixel p

We proposed a novel method that generates semantic-aware edge-map of the input image. Our method considers semantic information in two phases. In the first phase, for each pixel p, it computes mean gradient and average local skewness to generate the semantic gradient image (SGI). To this end, a set $W = \{w_1, w_2, \ldots, w_n\}$ containing increasing size windows are considered. For each window $w_i \in W$, four lines $H, V, D1$, and $D2$ passing through the center pixel p in four directions (as shown in Figure 4-1) are taken. To compute the mean gradient MD_W^p of the pixel p in the pre-processed image J, each line $L \in \{H, V, D1, D2\}$ inside the windows is partitioned into two equal halves L_1 and L_2 at p and then calculate their average intensity difference as follows:

$$MD_W^p = \frac{1}{|L| \times |W|} \times \sum_W \sum_L |mean(J_{L_1}) - mean(J_{L_2})| \tag{4.1}$$

where $mean(J_{L_1})$ and $mean(J_{L_2})$ represents the average intensity of the pixels on the line segments L_1 and L_2, respectively. To compute the mean gradient of a pixel equation 4.1 incorporates semantic information by analyzing the local spatial distribution of its neighbor pixels in different directions. The high value of the mean gradient MD_W^p indicates that the pixel p is more probable to be a structural edge pixel, and the low value indicates that it is more probable to be a non-edge pixel.

To compute the average local skewness of the pixel p, our technique generates the morphological gradient image J_{mg} as follows [111]:

$$J_{mg} = (\delta_{SE}(J) - \varepsilon_{SE}(J)) \tag{4.2}$$

Considering the pixel $p \in J_{mg}$ as center pixel of each window $w_i \in W$, a local histogram h_{w_i} is constructed by taking into account only the pixels inside the window w_i lie on the four lines $H, V, D1$, and $D2$. Then the average local skewness Sk_W^p of the pixel p is computed as:

$$Sk_W^p = \frac{1}{|W|} \sum_{w_i \in W} \frac{max(h_{w_i}) - median(h_{w_i})}{max(h_{w_i}) - min(h_{w_i})}. \tag{4.3}$$

To compute the average local skewness of a pixel equation 4.3 incorporates semantic information by analyzing the distribution of its neighbor pixels in morphological gradient image J_{mg} and always provides positive value. High value of Sk_W^p indicates that the pixel p is more probable to be a structural edge pixel and low value indicates that it more probable to be a non-edge pixel. After compute the average gradient MD_W^p and the average local skewness Sk_W^p, the proposed semantic gradient image (SGI) associated to the input image I is generated by combining these two features as follows:

$$SGI(p) = MD_W^p \times Sk_W^p \tag{4.4}$$

MD_W^p and Sk_W^p considers semantic information of a pixel p by analyzing the local distribution of pre-processed image J and morphological gradient image J_{mg}, respectively. Both provided higher values for structural edge pixels and lower values for textural or non-edge pixels. Thus, equation 4.4 combines two different semantic information of the pixels for generating the semantic gradient image (SGI). Hence, the proposed SGI image is capable of better differentiating the structural and textural edge pixels of the input image. Figures 4-2 (b) and (g) show the semantic gradient images generated by the proposed technique from the input images shown in Figures 4-2 (a) and (f). From these figures one can see that the higher intensity pixels of SGI are associated to the structural edge and the lower intensity pixels are associated to either non-edge or textural edge. Note that for the input image of irregular and varying scale textures, there is a possibility of having some pixels in SGI with high intensity values associated to textural edge pixels. In such situation even SGI provides little information to discriminate structural and textural edge pixels.

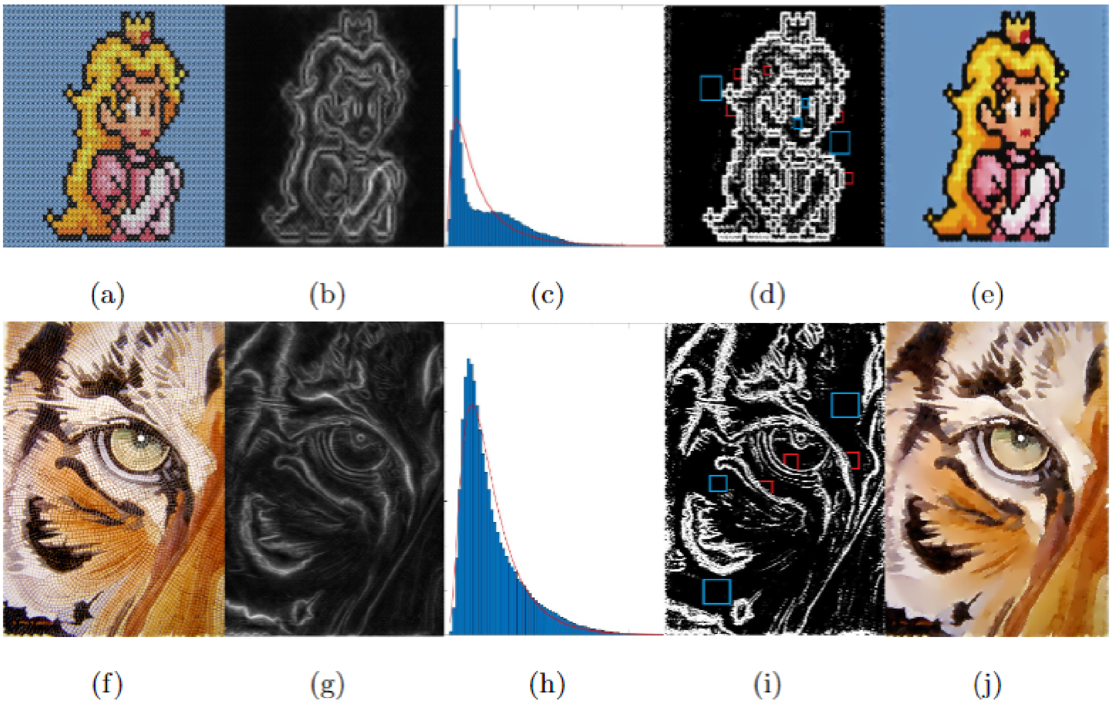


Figure 4-2: (a)(f) Original images and their corresponding generated (b)(g) semantic gradient images (SGIs), (c)(h) histograms of SGIs, (d)(i) edge-maps, and (e)(j) filter images.

In order to discriminate the structural edge and non-edge pixels of the input image more accurately, we proposed a novel method that exploits JS divergence to consider semantic information of the pixels by defining appropriate window. Since, the number of structural edge pixels in an image are much smaller than the number of non-edge pixels. The histogram of the SGI is likely to follow a positive-skewed distribution as shown in Figures 4-2(c) and (h). Thus, the histogram of SGI can be approximated with lognormal distribution by defining its two parameters μ and σ as follows:

$$\begin{cases} f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\frac{-(\ln(x)-\mu)^2}{2\sigma^2}} \\ \mu = \ln\left(\frac{\nu}{\sqrt{1+\frac{\gamma^2}{\nu^2}}}\right) \\ \sigma = \sqrt{\ln\left(1+\frac{\gamma^2}{\nu^2}\right)} \end{cases} \qquad \text{where } x > 0, \sigma > 0 \tag{4.5}$$

The red curves in Figures 4-2(c) and (h) are the lognormal distributions that approximated the histograms of SGIs by estimating μ and σ using the maximum likelihood estimator. The pixels in SGI associated to the left tail and the right tail of the approximated histogram are more certain to be non-edge pixels and structural edge pixels, respectively. The certainty level of the pixels between left and right tails are less. In our work to consider semantic information of a pixel, the size of the window is defined based on its certainty level. Pixels which are certain to be non-edge and structural edge are used in large and small window, respectively and the moderate size windows are used for uncertain pixels. The certainty level of the pixels are defined with help of the approximated histogram. After estimating μ and σ, the histogram of SGI is divided into four intervals (0, $e^{\mu-\sigma}$), $[e^{\mu-\sigma}, e^{\mu})$, $[e^{\mu}, e^{\mu+\sigma})$ and $[e^{\mu+\sigma}, +\infty)$. The intensity values of the pixels in SGI within the first interval (0, $e^{\mu-\sigma}$) and the fourth interval $[e^{\mu+\sigma}, +\infty)$ are more certain to be non-edge pixels and structural edge pixels, respectively. So, the window w_{01}, w_{12}, w_{23}, and w_{34} (such that the size of $w_{01} > w_{12} > w_{23} > w_{34}$) is used to consider semantic information of the pixels associated to the intervals (0, $e^{\mu-\sigma}$), $[e^{\mu-\sigma}, e^{\mu})$, $[e^{\mu}, e^{\mu+\sigma})$ and $[e^{\mu+\sigma}, +\infty)$, respectively. Once the window for each pixel of the input image is defined, the proposed technique exploits JS divergence to consider semantic information for determining whether it belongs to a structural edge or not.

In probability theory, JS divergence is a useful tool for measuring the similarity between two probability distributions. In this work JS divergence is exploited same way as presented in Subsection 3.3.2.3. Lower the value of $JSD(P, Q)$ indicates P and Q have similar distributions. For each pixel p of the pre-processed image J, its corresponding window $w \in \{w_{01}, w_{12}, w_{23}, w_{34}\}$ is used to incorporate semantic information. To this end, considering p as center pixel of w, four lines, namely horizontal H, vertical V, two diagonal $D1$ and $D2$ passes through p as shown in Figure 4-1 are taken. The probability density function of each line H, V, $D1$, and $D2$ is defined as follows:

$$\begin{cases} P_L(q) = \frac{|J_q - min(mean(J_{L_1}), mean(J_{L_2}))|}{\sum |J_q - min(mean(J_{L_1}), mean(J_{L_2}))|}, \\ \qquad q \in L_1 \cup L_2, \quad L \in \{H, V, D1, D2\} \end{cases} \tag{4.6}$$

where J_q represents the intensity value of the pixel $q \in L$. The probability density function of at least one of the four lines (i.e., P_H, P_V, P_{D1} or P_{D2}) will be similar to a discrete step distribution if the pixel p is in structural edge. Otherwise, the probability density functions of all four lines will be more akin to discrete uniform distributions. To incorporate such semantic information two reference functions, a uniform function U and a step function T are taken. Let Q_L and R_L denote the probability density functions for the step function T and the uniform function U, respectively. If the pixel p is a structural edge pixel then for at least one line $L \in \{H, V, D1, D2\}$, the distribution of P_L will be similar to the step distribution Q_L i.e., the value of $JSD(P_L, Q_L)$ will be low. If the pixel p is not a structural edge pixel then for all the four lines the distribution of P_L will be similar to the uniform distribution R_L i.e., the value of $JSD(P_L, R_L)$ will be low. In our work such semantic information is exploited to discriminate structural edge pixels and pixels of the input image that are non-edge and generate the binary edge-map E_b as follows:

$$E_b(p) = \begin{cases} 1, \; if \;\; min_L\{JSD(P_L, Q_L)\} \\ \qquad \leq \frac{1}{|L|}\sum_L\{JSD(P_L, R_L)\} \\ 0, \; otherwise \end{cases} \tag{4.7}$$

Figures 4-2 (d) and (i) show the semantic-aware edge-maps generated by the proposed technique.

4.2.2 Edge-aware adaptive median filter

Once the semantic edge-map of the input image is obtained, the filtered image is produced by applying the recursive edge-aware adaptive median filter developed in Chapter 2. Figure 4-2 shows the semantic edge-maps and the filtered images generated the proposed technique. The pseudo-code for this technique is provided in Algorithm 4.

Algorithm 4 Proposed parameter efficient image filtering technique

```
Input: Image $I$
Output: Filtered Image $I_f$
Generate pre-processed image $J$ using equation (3.1)
for each pixel $p \in J$
  for each pre-processed image band $J_c$
    for each window $w \in W$
      Compute feature $SGI(p)$ by
      using equation (4.4) .
    end for
  end for
end for
SGI is divided into four regions by $(0, e^{\mu-\sigma})$, $[e^{\mu-\sigma}, e^{\mu})$, $[e^{\mu}, e^{\mu+\sigma})$ and
$[e^{\mu+\sigma}, +\infty)$, and a set of window sizes $F_1\|F_2\|F_3\|F_4$ are determined for different region.
Generate edge-map $E_b$ by applying equation 4.7.
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.
```

4.2.3 Parameters of the proposed technique

Although the proposed technique considers several windows but most of them are of fixed-size. In pre-processing step, irrespective of input image, a fixed-size 3×3 square window is used as SE to generate the pre-processed image J. To generate SGI, the proposed technique computes the average gradient and the average local skewness of each pixel by considering semantic information. For the image of having varying scale irregular textures, it is not possible to capture sufficient semantic information by using a single window. To incorporate appropriate semantic information, a set of windows $W = \{w_1, w_2, \dots, w_n\}$ of increasing sizes are considered. The present work suggested to consider only four windows w_1, w_2, w_3 and w_4 to capture semantic information of variety of textures starting from smaller to moderate to larger and the size of these windows are defined by taking 3^{rd}, 5^{th}, 7^{th}, and 9^{th} numbers from the Fibonacci series $[1, 1, 2, 3, 5, 8, 13, 21, 34, \dots]$. In our work the window size is calculated using the formula $2x + 1$, where x is a number taken from Fibonacci series. So, irrespective of the considered input image, four windows of size $5 \times 5, 11 \times 11, 27 \times 27$, and 69×69 are used to generate the SGI.

After generating the SGI, to incorporate semantic information for each input image pixel for determining whether it belongs to a structural edge or not, our technique considers one of the four windows w_{01}, w_{12}, w_{23}, or w_{34}. In this research irrespective of the considered input image, the window size w_{01} and w_{34} is fixed to as ($w_{01} = 69 \times 69$) and ($w_{34} = 5 \times 5$), respectively. Moreover, depending upon the textures present in the input image, the size of the other two windows w_{12} and w_{23} are manually fixed from four discrete options F_1, F_2, F_3 and F_4. Where, $F_1 = [w_{12} = 43 \times 43, \ \ w_{23} = 27 \times 27]$, $F_2 = [w_{12} = 27 \times 27, \ \ w_{23} = 17 \times 17]$, $F_3 = [w_{12} = 17 \times 17, \ \ w_{23} = 11 \times 11]$, and $F_4 = [w_{12} = 11 \times 11, \ \ w_{23} = 7 \times 7]$ is defined by considering four consecutive pairs of Fibonacci numbers $[21, 13], [13, 8], [8, 5]$ and $[5, 3]$, respectively. Note that the size of windows w_{12} and w_{23} are the only parameter of the proposed technique that needs to be fixed manually by choosing one of the four options F_1, F_2, F_3 or F_4. Irrespective of the input images, the other parameters are either fixed or defined automatically. Conversely, the majority of existing state-of-the-art methods require multiple parameters, and their results are highly sensitive to their parameter values which need to be fine tuned manually within a wide interval. Figure 4-3 shows the filtered images produced by the proposed technique considering the options F_1, F_2, F_3, and F_4 for a simple cartoon image. From this figure one can see that the options F_3 and F_4 are suitable for preserving smaller details, whereas the options F_1 and F_2 are suitable for smoothing.

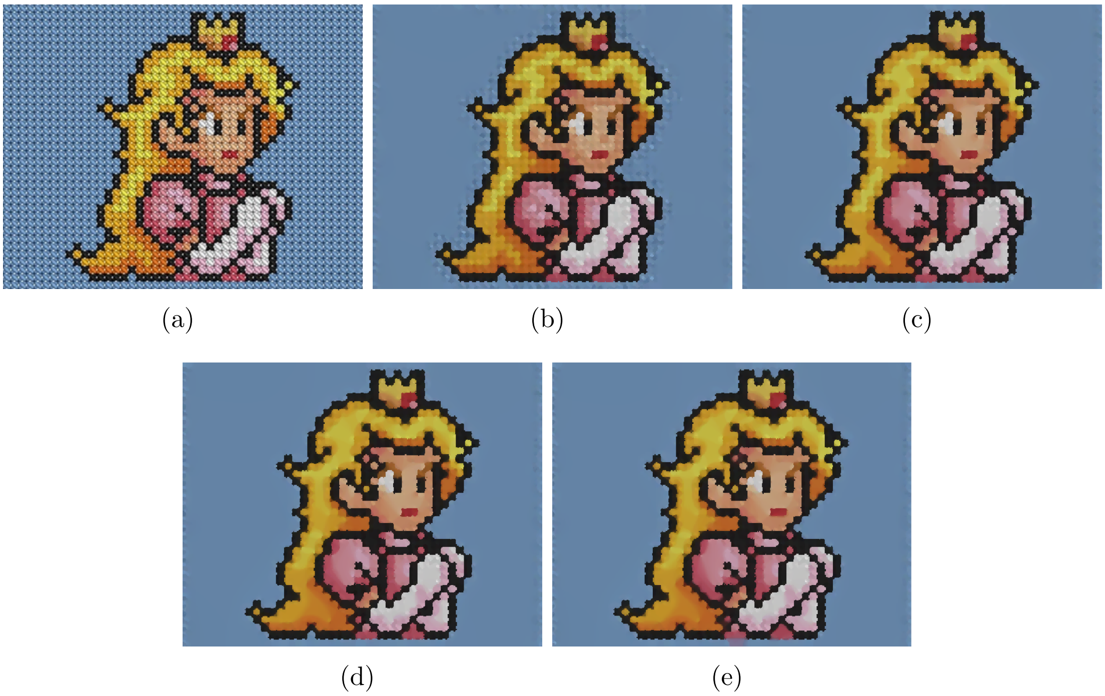
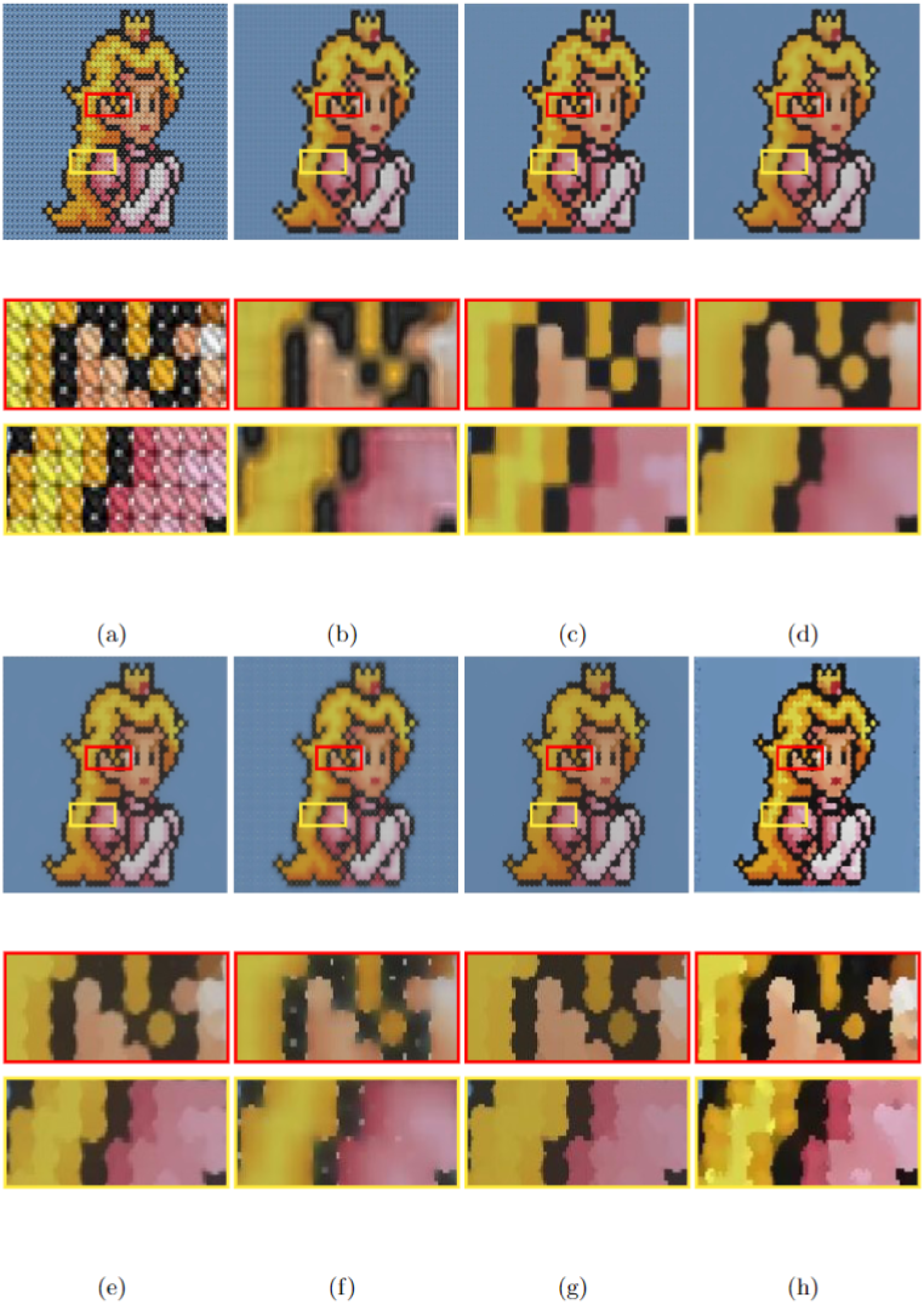

(a) (b) (c) (d) (e)

Figure 4-3: (a) Original image and the filtered images produced by our developed technique by using the options (b) F_4, (c) F_3, (d) F_2, and (e) F_1.

4.3 Experimental results and analysis

To assess the robustness of the proposed technique, the experiment includes both qualitative and quantitative comparisons with various state-of-the-art techniques.



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

(e) (f) (g) (h)

Figure 4-4: Filtering results of a cross-stitch cartoon image having regular texture: (a) the original image and the filtered images produced by (b) RegCov ($k = 15, ps = 6, \sigma = 0.2$) [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], and (h) Proposed (F_4) techniques.

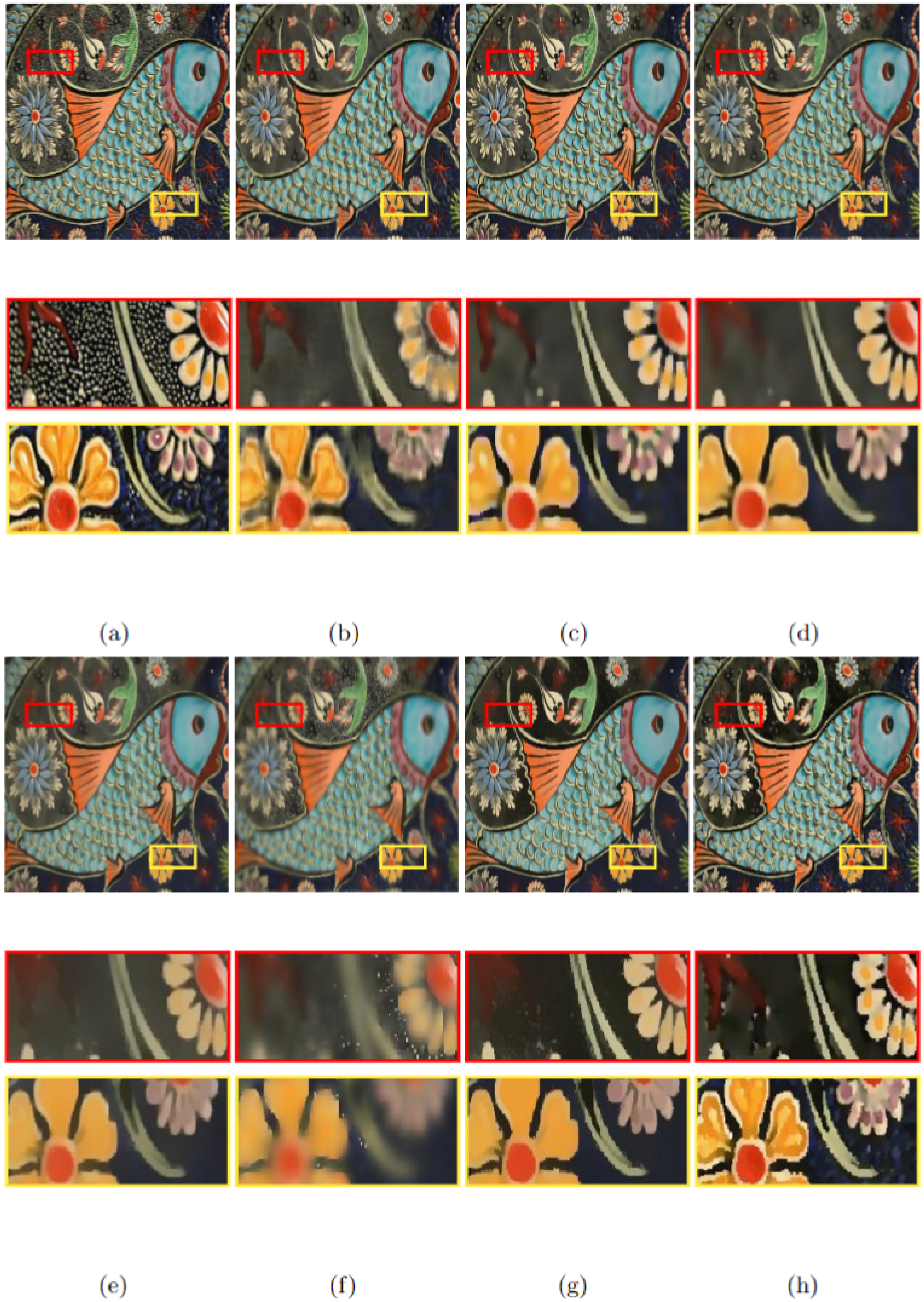


Figure 4-5: Filtering results of a mosaic image having complex textures and structures: (a) the original image and the filtered images produced by (b) RegCov ($k = 15, ps = 6, \sigma = 0.2$) [72] (c) BTF, $k = 9, n_{itr} = 7$ [32], (d) SATF $ss = 3, sr = 0.1, st = 0.1, n_{itr} = 7, div = 30$ [69], (e) 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], and (h) Proposed (F_3) techniques.

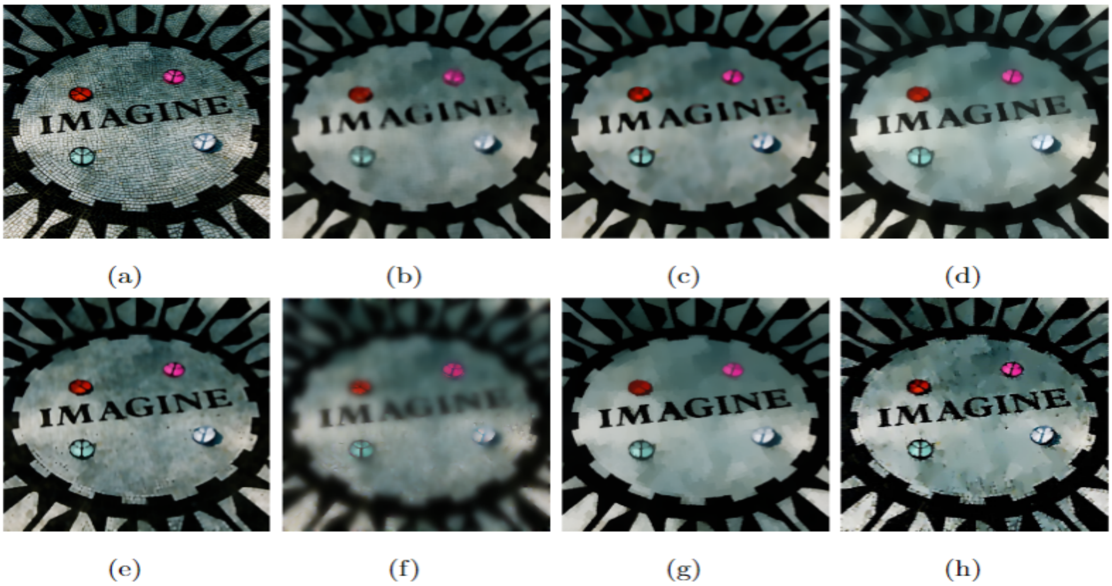


Figure 4-6: Filtering results of a floor image having irregular texture: (a) the original image and the filtered images generated by (b) Reg-Cov ($k = 15, ps = 6, \sigma = 0.2$) [72] (c) BTF, ($k = 9, n_{itr} = 7$) [32], (d) SATF ($ss = 3, sr = 0.1, st = 0.1, n_{itr} = 7, div = 30$) [69], (e) SATV, ($\lambda = 2.5$) [125], (f) RILS ($\rho_{smooth} = 3, \rho_{sharp} = 5$) [91], (g) GISF, ($\lambda = 50, \gamma = 20/255, n_{itr} = 15$) [92], and (h) Proposed (F_3) techniques.

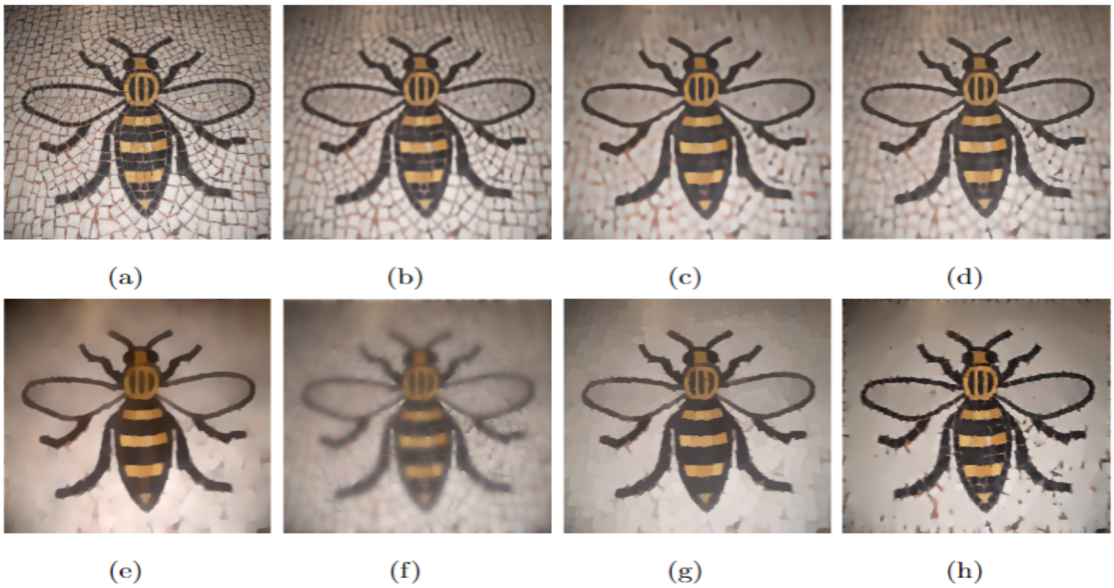
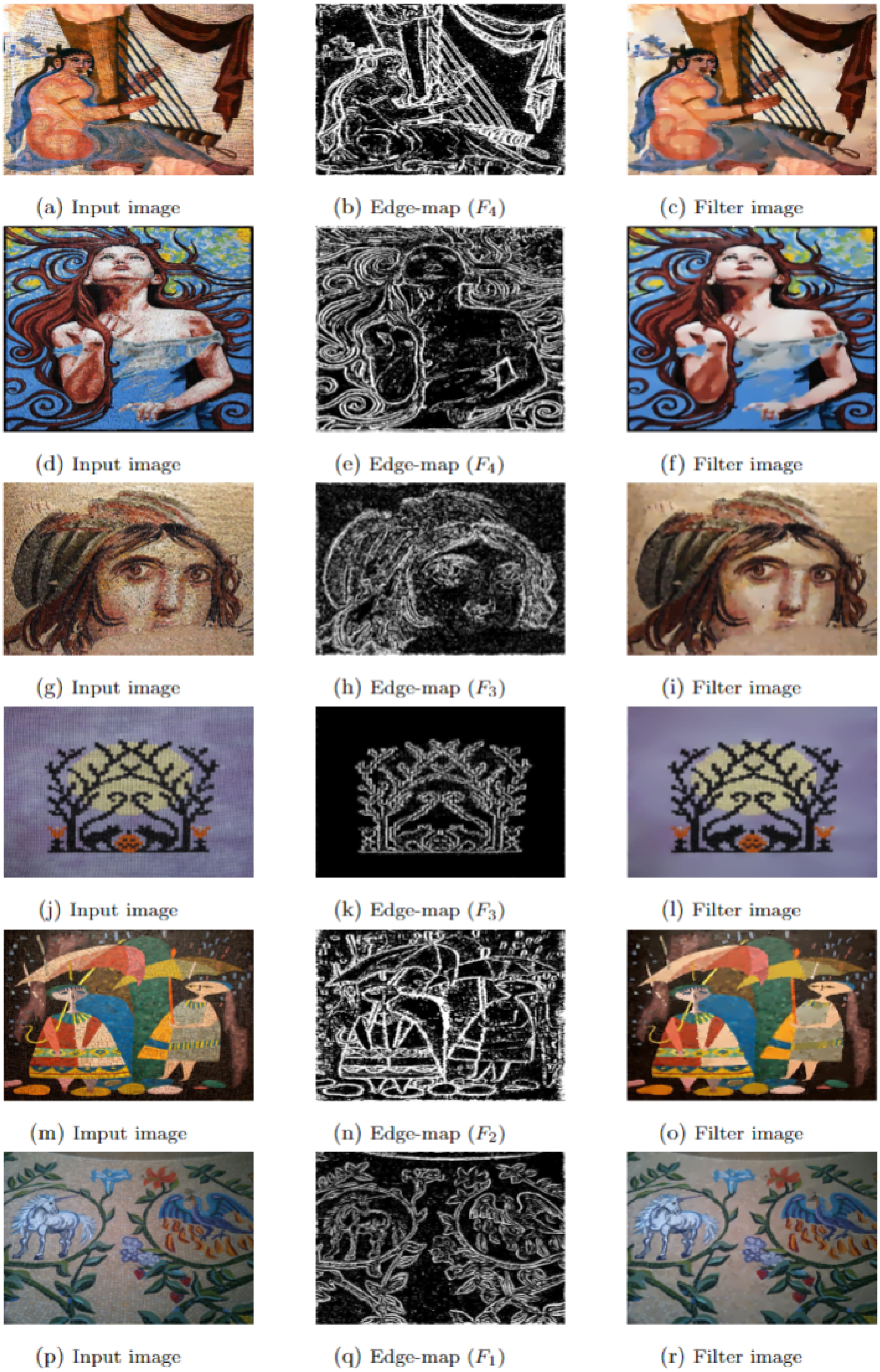


Figure 4-7: Filtering results of a bee image having irregular varying scale textures: (a) the original image and the filtered images generated by (b) Reg-Cov ($k = 15, ps = 6, \sigma = 0.2$) [72] (c) BTF, ($k = 9, n_{itr} = 7$) [32], (d) SATF ($ss = 3, sr = 0.1, st = 0.1, n_{itr} = 7, div = 30$) [69], (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], and (h) Proposed (F_2) techniques.



(a) Input image (b) Edge-map (F_4) (c) Filter image

(d) Input image (e) Edge-map (F_4) (f) Filter image

(g) Input image (h) Edge-map (F_3) (i) Filter image

(j) Input image (k) Edge-map (F_3) (l) Filter image

(m) Imput image (n) Edge-map (F_2) (o) Filter image

(p) Input image (q) Edge-map (F_1) (r) Filter image

Figure 4-8: (a)(d)(g)(j)(m)(p) Original images with various textures and structures and their corresponding (b)(e)(h)(k)(n)(q) edge-maps and (c)(f)(i)(l)(o)(r) filtered images generated by the proposed technique.

4.3.1 Qualitative comparison

In this experiment, the quality of the filtered images produced by our developed technique is compared with those generated by six well-known state-of-the-art structure-preserving filtering techniques: Reg-Cov [72], BTF [32], SATF [69], SATV [125], RILS [91], and GISF [92]. In qualitative comparison, the filter images produced by the different techniques are visually analyzed to find out how well the structures and textures present on the input images are discriminated for removing minute details of textures while retaining the key structures.

For varieties of input images, the filtered images obtained by the different techniques are shown in Figures 4-4, 4-5, 4-6, 4-7 and 4-8. Examining these figures reveals that the proposed technique excels in both texture smoothing and structure preservation. For example, the input image shown in Figure 4-4(a) is a simple cartoon image with regular textures and smaller structural details of different colors. The filtered images produced by the different techniques and some zoom portions of these filtered images are presented in Figure 4-4. From these figures, it can be seen that our developed technique better preserves the smaller color details while smoothing out the textural details. Figure 4-5(a) shows another input image with random textures and complex structural details. The corresponding filtered images and some zoomed portions of these filtered images are illustrated in Figure 4-5. Again, from these images, it is evident that, in contrast to the existing state-of-the-art methods, the proposed technique better preserves the smaller structures while smoothing out the textural details. Figures 4-6 and 4-7 show the filtering results for another two input images with varying scale textures. By visually analyzing these results, it can be observed that our developed technique is able to preserve the structures and smooth out the textures much better than most of the state-of-the-art techniques. Note that all the literature techniques considered here have multiple parameters, and their filtering results are heavily determined by these parameters. The measurements of these parameters are varied from image to image and needed to be fixed manually within a wide range of continuous interval. To ensure a fair comparison, this experiment uses parameter values from the respective papers where the best results are reported, whenever feasible. Otherwise, the optimal values for these parameters are set manually through trial and error. Whereas, as presented in Section 4.2.3 the size of windows w_{12} and w_{23} are the only critical parameters of the proposed technique that needed to be fixed manually by choosing one of the four discrete options of F_1, F_2, F_3 or F_4. This appears to be a significant advantage of our developed technique over existing state-of-the-art methods. Our proposed technique is tested on a substantial number of images with a variety of regular and irregular textural patterns and always gets satisfactory results. Figure 4-8 show few more filtering results from our developed technique for different types of input images. From these results, one can observe that our technique accomplishes numerous contradictory aims, such as locating and removing texture, maintaining structural boundaries, safeguarding subtle features like corners, and avoiding over-sharpening and/or over-blurring artifacts.

4.3.2 Quantitative comparison

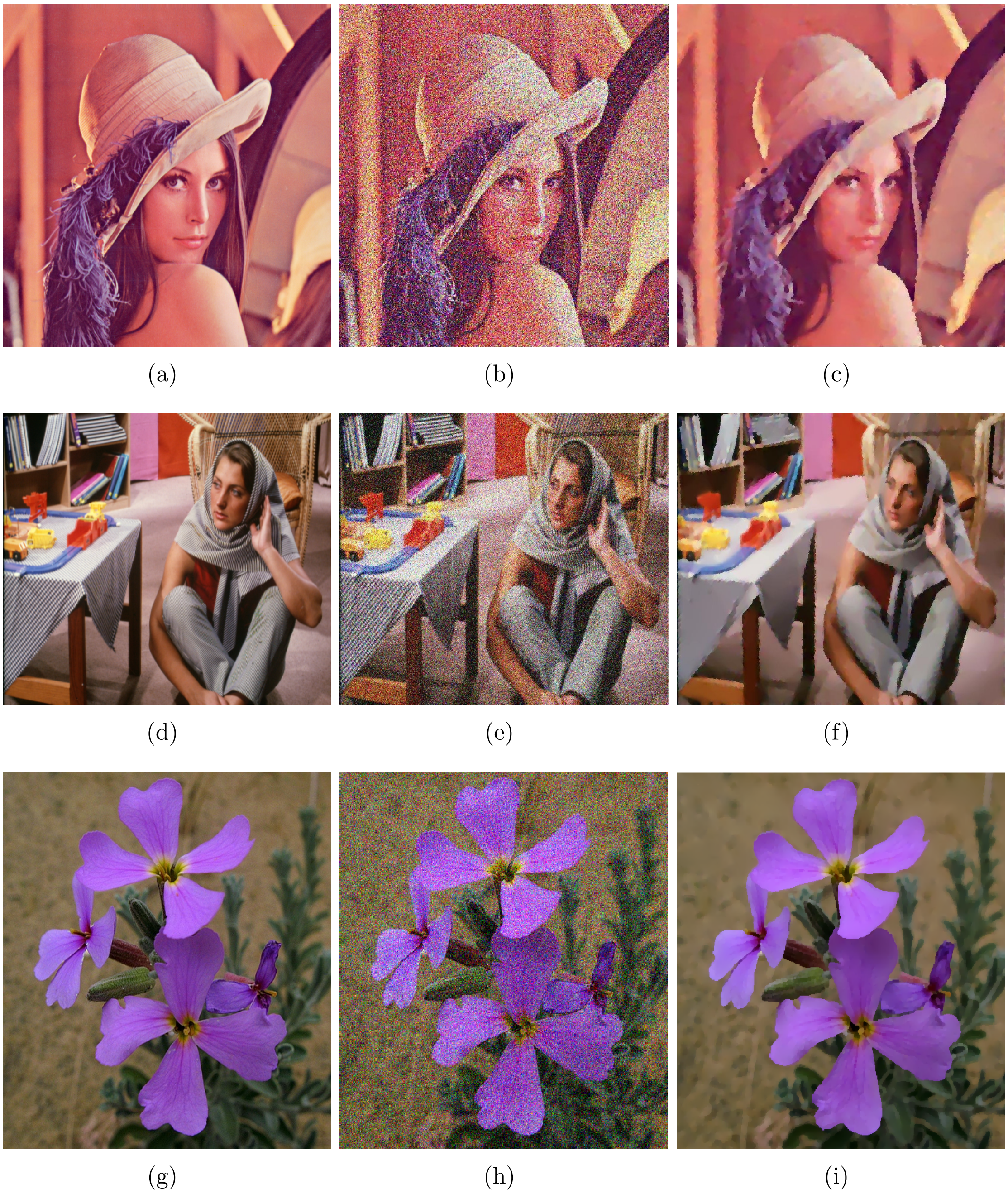
In this experiment, a quantitative comparison between the filtered images produced by our developed and state-of-the-art techniques are carried out. Here we use three subjective reference IQA metrics: SSIM [136], MSSIM, and MI [131] and one subjective no-reference metric, PIQE [132] for quantitative comparisons, where both SSIM and MSSIM range from [0, 1], MI ranges from [0-log($m \times n$)], where $m \times n$ is the image size, and PIQE ranges from [0, 100]. For the first three indices SSIM, MSSIM and MI, a higher value indicates better performance, whereas a lower value for PIQE indicates better. Table 4.1 reports the scores of different IQA measures computed from the filter images generated by different techniques. From these results, it is evident that the proposed technique consistently yields better IQA scores compared to the existing state-of-the-art techniques.

Table 4.1: The SSIM, MSSIM, and PIQE for the filtered images produced by different techniques. Bold type face indicates the best results.

Images	***Metrics***	***RegRov* [72]**	***BTF* [32]**	***SATF* [69]**	***SATV* [125]**	***RILS* [91]**	***GISF* [92]**	***Proposed***
Cartoon- Fig. 4-4 PIQE=44.5055	SSIM	0.5227	0.5772	0.6006	0.6091	0.5017	0.5424	**0.6236**
	MSSIM	0.5161	0.5401	0.5613	0.5704	0.4276	0.4348	**0.5939**
	MI	2.9846	2.8573	2.9095	2.5602	2.9771	1.8120	**2.9857**
	PIQE	66.5025	87.1371	87.1248	82.4621	100	85.5485	**53.8407**
Fish- Fig. 4-5 PIQE=31.6049	SSIM	0.4968	0.5527	0.6353	0.6759	0.4945	0.4544	**0.6924**
	MSSIM	0.6909	0.7431	0.8171	0.8284	0.6063	0.5238	**0.8522**
	MI	1.5275	1.6297	1.7915	1.8397	1.4633	1.6064	**1.9401**
	PIQE	69.88529	87.2677	84.6146	73.6599	83.8448	80.7166	**64.4851**
Imagine- Fig. 4-6 PIQE=37.2748	SSIM	0.36	0.38	0.43	0.31	0.26	0.44	**0.4583**
	MSSIM	0.58	0.61	**0.68**	0.62	0.54	0.62	0.6551
	MI	2.30	2.27	2.08	2.00	1.91	2.31	**2.33**
	PIQE	86.78	88.85	87.05	84.74	88.19	86.26	**82.6128**
Bee- Fig. 4-7 PIQE=16.2815	SSIM	0.49	0.44	0.57	0.51	0.47	0.53	**0.5588**
	MSSIM	0.58	0.74	0.70	0.51	0.50	0.53	**0.76**
	MI	2.03	2.07	2.50	2.45	2.37	1.89	**2.5107**
	PIQE	86.74	88.86	89.96	86.47	100	87.91	**72.2437**

4.3.3 Applications

As in previous chapter, also in this chapter our developed technique is validated across three different applications: image denoising, image enhancement, and tone mapping.



(a) (b) (c)

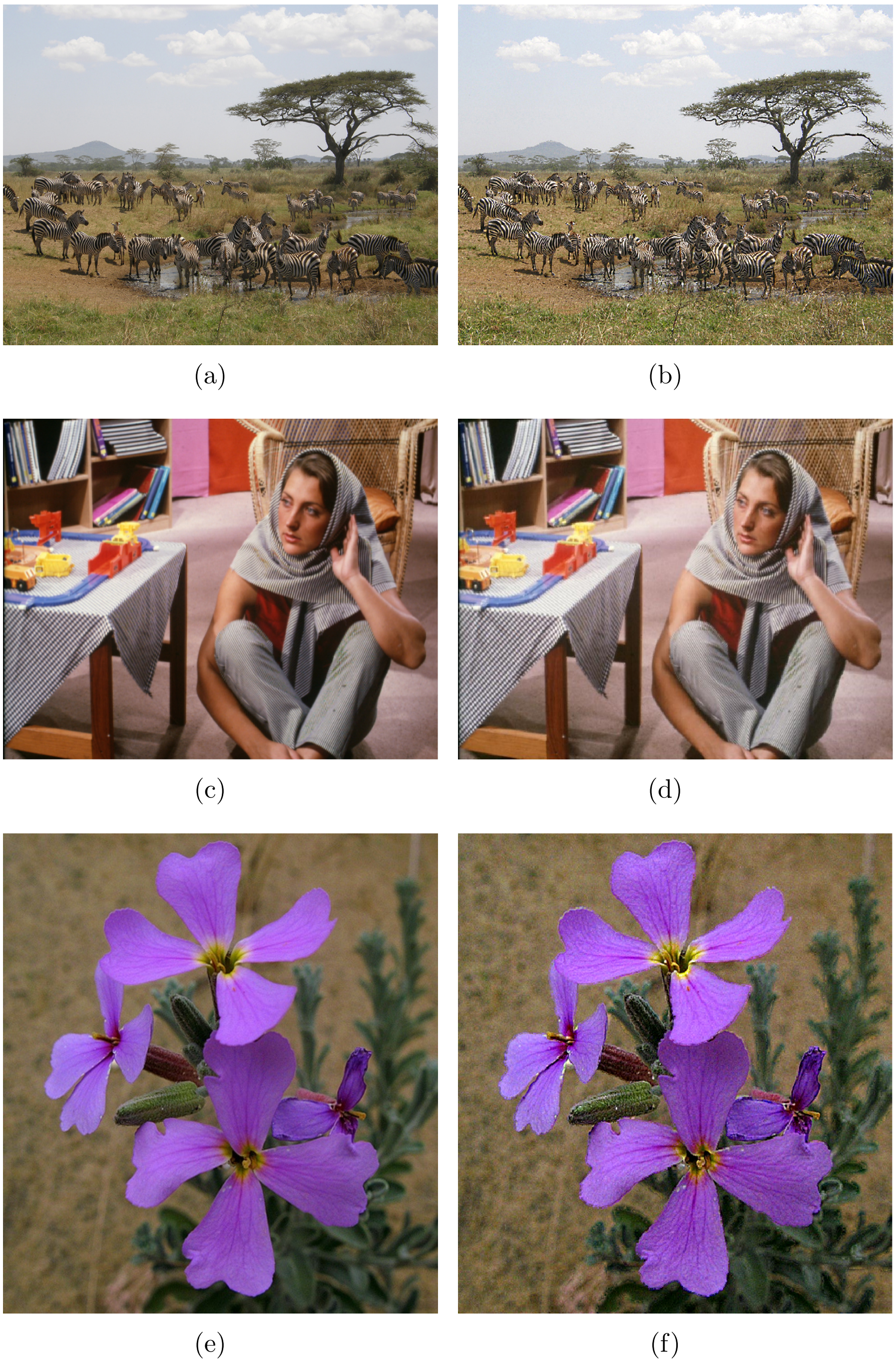
(d) (e) (f)

(g) (h) (i)

Figure 4-9: (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.

4.3.3.1 Image denoising

In Figure 4-9, the presented results display both the noisy and the corresponding denoised images achieved through the proposed filtering technique. The filtered image clearly demonstrates its ability to preserve structures effectively while successfully eliminating noise.



(a) (b)

(c) (d)

(e) (f)

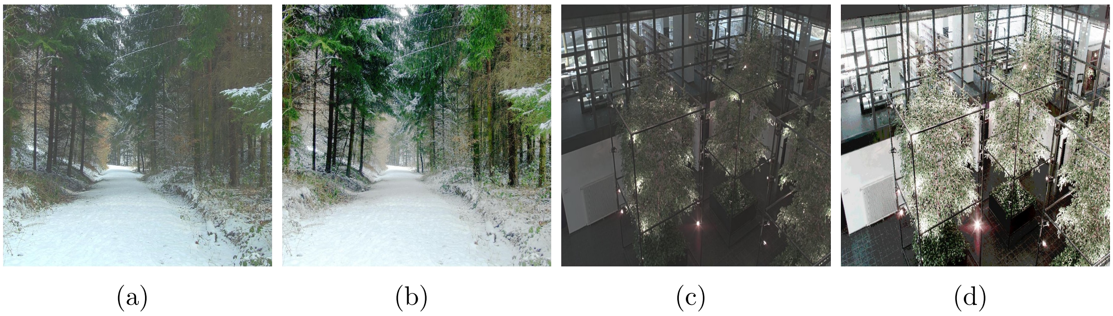
Figure 4-10: (a)(c)(e) Original images and their corresponding (b)(d)(f) enhanced images obtained by the proposed technique.

4.3.3.2 Detail enhancement

In Fig. 4-9, the presented images showcase both the original images and their corresponding enhanced versions achieved through the proposed filtering method. The enhancement process begins by decomposing the input image into two components - one being a smoothed image and the other capturing textural details. The final enhanced image is produced by incorporating these textural details into the original image. The enhanced images prominently highlight the textural details, showcasing the performance of our developed filtering technique.

4.3.3.3 Tone mapping

In Figure 4-10(a)(c), two LDR image (RGB) tone-mapped from two HDR images using Farbman's technique [47] is presented. Figure 4-10(b)(d) illustrates the corresponding tone-mapped RGB version generated with the proposed filtering technique. Notably, the proposed filtering technique was seamlessly integrated in place of bilateral filtering (BF) within Durand's method [49] with $\gamma = 0.5$. This replacement demonstrates the impact of our developed technique in the context of tone mapping.



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

Figure 4-11: (a)(c) RGB tone mapped images by Farbman techniques [47] and (b)(d) RGB tone mapped images by proposed filtering, respectively, from two different HDR images.

4.3.4 Computational performance analysis

The computational complexity of our developed technique is dependent on the algorithm used to generate the semantic-aware edge-map and the recursive median filter. For an input image of size $(M \times N)$, the computational complexity of our algorithm that generates the semantic-aware edge-map is $\Theta(M \times N \times S_1 \log S_1)$, where S_1 represents the size of the largest window in W used to incorporate semantic information. Where skewness calculation for window S_1 takes $\Theta(S_1 \log S_1)$. The computational complexity of the adaptive median filter used in the proposed technique is $\Theta(M \times N \times S_2 \log S_2)$, where S_2 represents the size of the largest window among the adaptive windows L_w and w_s^a automatically define for filtering non-edge and edge pixels, respectively. The complexity of median calculation is $\Theta(S_2 \log S_2)$. Therefore, the overall computational complexity associated with our approach is $\Theta(M \times N \times (S_1 \log S_1 + S_2 \log S_2))$.

4.3.5 Limitations

The method begins with computing a morphological semantic gradient on input images using a series of increasing window sizes selected from alternate Fibonacci numbers (e.g., 2, 5, 13, 34), corresponding to square windows of sizes 5×5, 11×11, 27×27, and 69×69. This approach reduces sensitivity to parameter selection by capturing gradients for objects of varying scales, from small to large, making it effective for images with diverse object sizes and scales.

Subsequently, a Jensen-Shannon Divergence (JSD)-based gradient identification is performed in four directions. The window size for this operation is adaptively determined based on the gradient values computed in the initial phase, ensuring that the first layer of operation guides optimal window selection for the second phase. The accuracy of the final edge map depends on the sensitivity of the second phase to the initial semantic gradient computation.

As a two-layer technique, this method effectively handles multiscale edge detection but incurs higher runtime complexity due to its iterative and adaptive processes of two phases.

4.4 Conclusions

The most challenging task in structure preserving filtering is to discriminate important structures from textures, especially when irregular textural patterns with different scales are present. In many cases, spectral and spatial variations of the input image is not sufficient to discriminate structures and textures. In such cases, semantic information may provide additional useful insight. In this regard, the existing methods are very much parameter sensitive. Here, we introduced a novel approach for leveraging the semantic information of the image to enhance the discrimination of structural and textural edges. Our technique first generates a semantic-aware edge-map of the input image by exploiting semantic information. Then an edge-aware adaptive recursive median filter is utilized to generate the final filtered image. Although the developed filtering technique requires multiple windows, the size of most of these windows are either fixed or defined automatically, irrespective of the considered input image. There exists only a pair of windows that need to be fixed manually by choosing one of the four discrete options. The proposed technique provides satisfactory results for a wide selection of input images with minimal fine-tuning of its parameters, which is an important benefit in comparison to the current state-of-the-art techniques. Furthermore, in contrast to the majority of existing techniques, our technique accomplishes numerous contradictory aims, such as locating and removing texture, maintaining structural boundaries, safeguarding subtle features like corners, and avoiding over-sharpening and/or over-blurring artifacts. Furthermore, along with different computer graphics applications, the proposed technique also shows its robustness to incorporate spatial information for image classification.

List of publications from this chapter

Journals

1. Pradhan K, Patra S. A semantic edge-aware parameter efficient image filtering technique. Computers & Graphics. 2024 Nov 1;124:104068. ***(IF-2.5)***