Chapter 2

Semantic-aware structure preserving median morpho-filtering

2.1 Introduction

Natural images typically contain meaningful or significant (structure), as well as irrelevant or insignificant (texture) visual information. Structure preserving image filtering is a key operation to derive the meaningful (semantic) information from the images, which have a variety of applications in different fields like medical imaging, document analysis, remote sensing image analysis, etc [13, 70, 85]. Recent research focuses on the development of techniques which identify semantically significant objects from regular or irregular textural patterns for filtering. In the literature such techniques that incorporate semantic information for automatic structure-texture separation is termed as semantic-aware filtering [25, 133, 141, 145, 149]. In the current chapter, we have proposed a robust semantic-aware structure preserving filtering technique. Our technique defines a novel method to obtain an edge-aware adaptive window of dynamic shape for filtering each pixel of the image by excluding its neighbour pixels belonging to different textural or structural regions. Fig:2-1 illustrates the importance of the

uses of adaptive dynamic shape windows in comparison to fixed shape windows for filtering. As shown in Fig:2-1(b), the small fixed shape window does not work well to filter out the large-scale textures. On the contrary, large window does not preserve edges properly (blur the edges) as illustrated in Fig:2-1(c) and Fig:2-1(f). Since larger windows are useful to identify large-scale textures and smaller ones are effective for both detection and preservation of smaller structural edges. Defining an appropriate dynamic window for filtering varying size textures or structures is very much important. In the proposed work this issue is addressed by defining edge-aware adaptive dynamic shape windows as depicted in Fig:2-1(g) and Fig:2-1(h). In our technique, first, a novel method is proposed to generate the semantic edge-map by computing the skewness of global and local morphological gradient histograms. Then, using the generated edge-map an edge-aware structure preserving adaptive median morpho-filter is designed. The edge-map generation followed by median morpho-filtering iteratively runs to generate the final filter image. Note that morphological filters generally preserve the larger structural shapes while removing the smaller irregular shapes. In contrast, median filter converges the pixel values into the background pixels with greater counts. Thus, the proposed median morpho-filter merges the narrow texture to its original background as well as preserves the significant structures. The experimental results demonstrate, unlike many current techniques, our filtering method successfully balances multiple conflicting objectives. It identifies and removes texture, preserves structural edges, protects subtle features such as corners, and avoids introducing artifacts and distortions from over-sharpening or over-blurring. The key participation of this work are:

- Proposes a texture-structure decomposition technique by analysing the morphological gradient histogram of the image.
- Proposes a semantic-aware median morpho-filter for texture smoothing while preserving the significant structures.

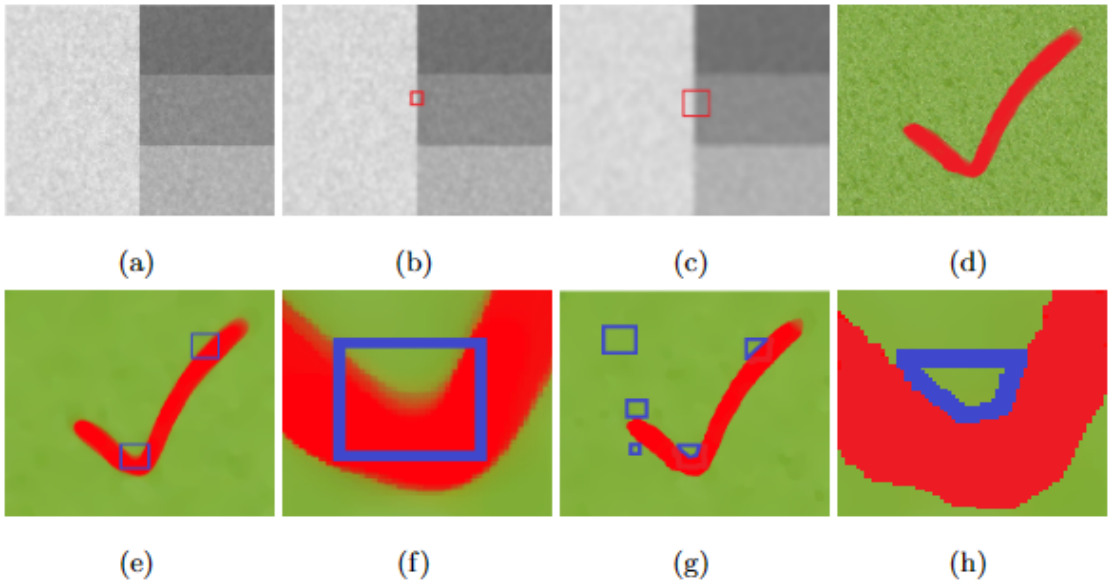

Figure 2-1: (a) Original image with simple structure. (b)(c) The filtered images obtained using smaller and larger static box windows. (d) Original image with complex structure. The filtered images and their zoomed portion obtained using (e)(f) static box window and (g)(h) proposed dynamic window.

The next part of this chapter is presented as details: Section 2.2 introduces briefly to morphological and median filters. The proposed technique is detailed in Section 2.3. Experimental findings are analyzed in Section 2.4, and finally, the conclusion of this work is presented in Section 2.5.

2.2 Morphological and median filters

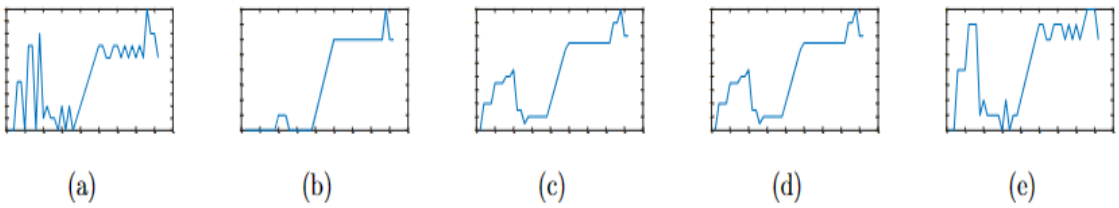

Figure 2-2: (a) Original 1-dimensional signal and the corresponding filtered signal obtained by applying (b) Opening, (c) Closing, (d) (Opening + Closing)/2, and (e) Median operations with a fixed SE of size 3 unit.

Our filtering technique exploits the properties of morphological and median filters [60, 121]. This section presents a concise overview of the two filtering techniques.

2.2.1 Morphological filter

It works on the rank ordering principle. The basic two operations dilation (δ) and erosion (ε) filter the image by replacing the center pixel of the predefined neighbour called structuring element (SE) with the local maximum and minimum value, respectively. The other operations such as opening ($\delta(\varepsilon)$) and closing ($\varepsilon(\delta)$) are designed by applying successive alternative operations of dilation and erosion. The very basic nature of these operations intuitively removes the impulse noises smaller in size than the neighbouring window (i.e SE) without affecting the basic shape of objects. Figure 2-2 displays the filtering outcomes obtained on a 1-dimensional signal by applying different morphological operations with a fixed SE of size 3 unit. In the proposed technique morphological filter is exploited to improve the discrimination between structural edges and textural regions of the image.

2.2.2 Median filter

It's a widely used rank order filter that substitutes each pixel with the median value of the defined neighbouring pixels. As the median value always divides the distribution into equal two parts in the same number of counts, it is naturally not sensitive to the impulse noise/outliers. Unlike the weighted average filters, a median filter always replaces the pixel value with a value that already exists within the window containing odd number of pixels. If the neighbour pixels follow a positively skewed distribution then the median will be less than the mean. On the other hand, for a negatively skewed distribution the median will be greater than the mean. For a symmetric/near-symmetric distribution (i.e., skewness is near to zero) the median will be equal to or closer to the mean value. Figure 2-2 (e) shows the filtering result obtained from Figure 2-2 (a) by applying median operation with a fixed window of size 3 unit. The median filter merges the high

intensity textures with its low intensity background if the local distribution is positively skewed and the low intensity texture with its high intensity foreground if the distribution is negatively skewed. In our proposed method, this intuitive property of the median filter is exploited to develop a robust edge preserving smoothing technique.

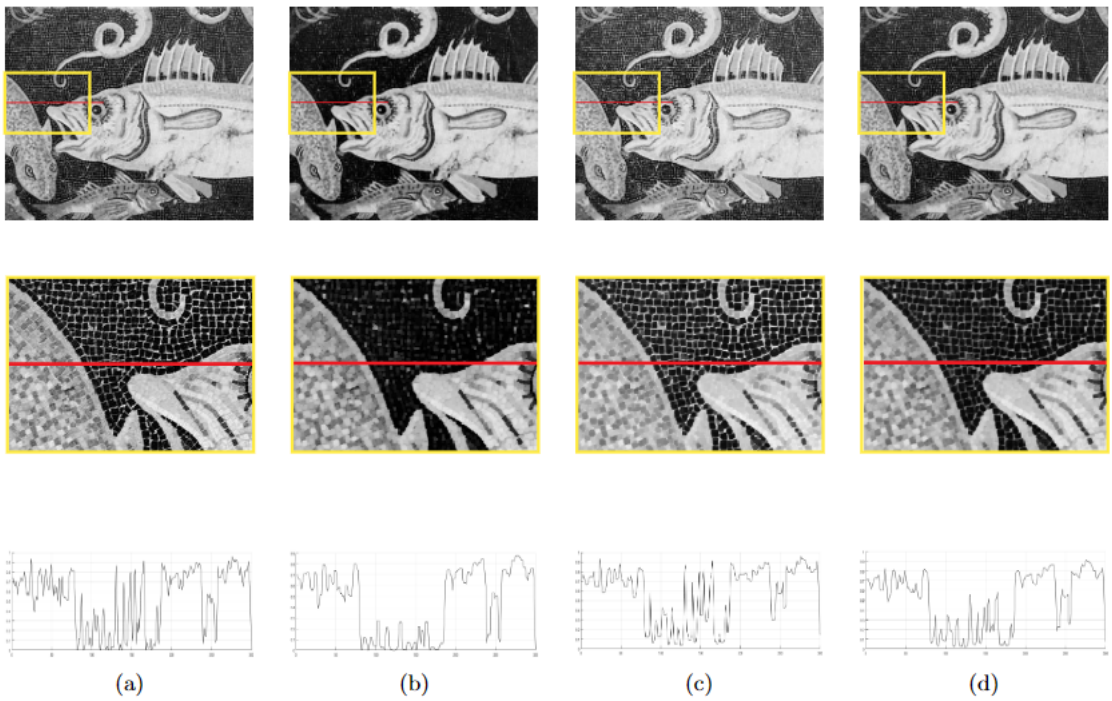

Figure 2-3: (a) Original image, its magnified part enclosed with yellow rectangle and the intensity values of the pixels on the red line. The filtering results obtained by applying (b) Opening, (c) Closing, and (d) (Opening + Closing) / 2.

2.3 Proposed technique [111]

The proposed filtering technique has two main tasks. First, it generates a binary edge-map by analysing the morphological gradient histogram of the image. Then, based on the generated edge-map, a semantic edge-aware median morpho-filtering technique is designed to generate the filtered image. The subsequent subsections provide a detailed explanation of the proposed filtering technique:

2.3.1 Image gradients for edge detection

In image processing, a traditional gradient represents the directional change in intensity or color within an image. Including the morphological gradient, there are several other traditional ways to generate gradients in images:

Sobel gradient: The Sobel operator uses convolution with specific filters to detect edges by emphasizing horizontal and vertical gradients. It's commonly used to highlight edges in both x (horizontal) and y (vertical) directions.

Prewitt gradient: Similar to Sobel but with slightly different kernel values, the Prewitt operator is also used for edge detection. It's simpler than Sobel but less sensitive to noise, making it effective for general gradient detection.

Scharr gradient: The Scharr operator is an improved version of the Sobel operator, especially for detecting edges in high-noise images. It emphasizes edges more strongly by using a different kernel, which can produce sharper gradients.

Laplacian of Gaussian (LoG): The Laplacian of Gaussian first applies Gaussian smoothing to reduce noise, then the Laplacian operator detects the gradient. LoG is beneficial for finding fine details and edges in complex images.

Roberts cross gradient: The Roberts Cross operator detects edges by calculating gradients diagonally (at 45-degree angles). It's particularly useful for simple edge detection in noiseless images.

Canny edge detector: The Canny detector is an advanced edge detection method that involves Gaussian smoothing, finding intensity gradients, and applying non-maximum suppression and edge tracking. While more complex, it's highly effective for identifying strong and weak edges, making it suitable for detecting image boundaries.

Isotropic gradient: This involves computing the gradient magnitude

and direction using isotropic (rotation-invariant) kernels. The gradient vector's magnitude and direction provide detailed information about the intensity changes.

Each of these methods has its own unique characteristics and application suitability, depending on image quality, noise levels, and the specific purpose (e.g., edge detection, feature extraction, or enhancing contrast). In addition to these, all the above-mentioned techniques are particularly applicable to the single band/channel or gray-scale images. For multi band/channel or RGB image these methods can be applied after converting/reducing to singe band/channel or gray scale image. There are techniques that can directly applied to the RGB or multi band/channel images directly for gradient generation.

Marginal Ordering: This approach involves applying morphological operations independently on each color channel (e.g., R, G, B in an RGB image). While straightforward, it can ignore the interactions between channels, potentially resulting in inconsistencies when combining channels back together.

Vectorial Ordering: This strategy considers the multichannel image as a vector field, treating each pixel as a vector of values across channels (e.g., [R,G,B]). Morphological operations are then defined based on the vector magnitude or another vector ordering scheme. This approach better preserves inter-channel relationships, making it particularly relevant for tasks requiring nuanced color interpretation, such as hyperspectral imaging.

In the proposed techniques we have used morphological gradient from the corresponding gray scale image, as because this makes simple both in concept and computationally also. Morphological gradients offer significant advantages over other traditional gradient methods, particularly in their robustness to noise, preservation of shape integrity, and resilience to illumination changes. By leveraging the difference between dilation and erosion, they produce precise and well-defined object boundaries while minimizing false positives in complex scenes.

Their adaptability through customizable structuring elements allows them to excel in diverse applications such as medical imaging, satellite-based remote sensing, industrial inspection, and document analysis. Additionally, their seamless integration with other morphological operations makes them a powerful tool for advanced image analysis tasks.

2.3.2 Generation of gradient image

Our filtering method works on both gray scale as well as colour images. If the input is a colour image, then its corresponding gray scale image is used to generate the morphological gradient image. In many cases, the images may have high gradient textural contents which could be misinterpreted as the structural edges. To optimize the discrimination between textural and structural contents of the image, in this work, we proposed a pre-processing step that performs filtering on the input image by defining a morphological filter combining opening and closing operations. Let I_{gray} represent the grayscale image derived from the input image. The proposed pre-processing step generates a filtered image J from I_{gray} using a (3×3) SE of 8-neighbours as follows:

$$J = (\delta_{SE}(\varepsilon_{SE}(I_{gray}) + \varepsilon_{SE}(\delta_{SE}(I_{gray}))/2 \tag{2.1}$$

Figure 2-2(a) shows a 1-dimensional signal with some high textural oscillations having equal or higher gradients than the structural edges. Figure 2-2(d) shows the filtered signal obtained by applying the morphological operation defined in equation (2.1). From this figure, one can see that the average of opening and closing removes the lower textural oscillations and diminishes the higher textural gradient while keeping the higher structural gradients mostly unchanged. Thus, it helps to create a difference between high gradient textural and structural contents in the gradient domain. To better understand the usefulness of the morphological

operation defined in equation (2.1), Figure 2-3 displays a real image alongside the filtered images obtained through applying different morphological operations. It also plots the pixels intensity values on the red line (passes through both textural and structural regions) for these images. From Figure 2-3(a) it is showing that in the original image, there are some textural edges whose gradients are as high as the structural edges. From Figures 2-3(b) and 2-3(c) one can see that the individual operation of the opening increases the darker textural gradient and closing increases the brighter textural gradient. But the average of both the operations as defined in equation (2.1) diminishes the textural high gradients (or impulse noises) while keeping structural edges with minimum distortion as shown in Figure 2-3(d). After obtaining the filtered image J, the proposed technique generates morphological gradient image J_{mg} as follows [114]:

$$J_{mg} = (\delta_{SE}(J) - \varepsilon_{SE}(J)) \tag{2.2}$$

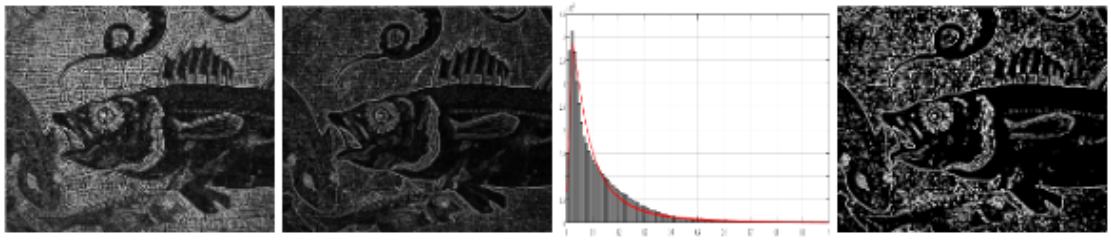

Figure 2-4: (a) Gradient image without applying pre-processing, (b) Gradient image with applying proposed pre-processing and (c) Its histogram with the best fitted log-normal distribution curve by maximum likelihood estimator. (d) The initial edge-map E_b obtained by applying global threshold t_1.

Figure 2-4(a) and Figure 2-4(b) is the gradient image obtained without and with applying the pre-processing step, respectively. From these figures, one can see that the proposed pre-processing step diminishes high gradient textures while keeping the structural edges with minimum distortions.

2.3.3 Generation of edge-map

The gradient image J_{mg} obtained from equation (2.2) is applied to produce the edge-map. For an image, a very less number of pixels are associated with significant structural edges as compared to the non-edge pixels. Thus, in the gradient image J_{mg}, higher gradient pixels associated with the structural edges are much less in number than the lower gradient pixels associated with smooth areas. So, the histogram of the gradient image is likely to follow a positively-skewed distribution with a high peak of lower values and a right-tail of higher values, as illustrated in Figure 2-4(c). Note that the domain of the gradient image J_{mg} is associated with non-negative values. So the positively-skewed histogram of the non-negative gradient can be assumed to follow a log-normal distribution [3, 40]:

$$\begin{cases} f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\frac{-(\ln(x)-\mu)^2}{2\sigma^2}} \\ \mu = \ln\left(\frac{\nu}{\sqrt{1+\frac{\gamma^2}{\nu^2}}}\right) \\ \sigma = \sqrt{\ln\left(1+\frac{\gamma^2}{\nu^2}\right)} \end{cases} \quad \text{where } x > 0, \sigma > 0 \tag{2.3}$$

The probability density function of a log-normal distribution $f(x)$ for a random variable x is a continuous distribution of positive values as define in equation (2.3), such that the logarithm of x i.e., $\ln(x)$ follow a normal distribution $N(\mu, \sigma)$. Where $\mu = \ln(m)$ is the scale parameter represents mean of $\ln(x)$ and m is the median of the x, σ is the shape parameter represents standard deviation of $\ln(x)$. The standard deviation and mean of x are denoted as γ and ν respectively. Since the gradient image J_{mg} may have zero gradient value we have added a very small positive value η as $J_{mg} = J_{mg} + \eta$ to model J_{mg} as log-normal distribution. In order to generate an accurate edge-map, first, a global threshold t_1 from the gradient image J_{mg} histogram is detected by exploiting the properties of the log-normal distribution. The threshold t_1 roughly differentiates the edge pixels and the non-edge ones in the gradient image. For a log-normal variate x,

the corresponding normal distribution $ln(x)$ contain 66.68% of values within the range $(\mu - \sigma, \mu + \sigma)$, which corresponds to the high peak portion $(e^{\mu-\sigma}, e^{\mu+\sigma})$ of the log-normal distribution. The right-tail potion $(>= e^{\mu+\sigma})$ of the corresponding log-normal distribution contains less than 33.32% of total values. In this work, we have taken the global threshold $t_1 = e^{\mu+\sigma}$ to roughly differentiate between the edge (high gradient) and the non-edge (low gradient) pixels. After obtaining the threshold t_1, the initial binary edge-map E_b from the gradient image is generated as follows:

$$E_b(p) = \begin{cases} 1, & \text{if } J_{mg}(p) >= t_1 \\ 0, & \text{if } J_{mg}(p) < t_1 \end{cases} \tag{2.4}$$

The initial edge-map E_b generated by considering the global threshold t_1 may have textural edges along with structural edges. To further reduces the presence of non-structural edges in the edge-map, a novel technique is proposed to refine it. For each edge pixels p in E_b, the technique constructs a local histogram $H(J^w_{mg})$ of the gradient image J_{mg} considering a fixed size window $W = (w \times w)$ centering at pixel p. Then to determine whether the pixel p is part of a structural edge or not, it computes the non-parametric skewness value Sk_p from the histogram $H(J^w_{mg})$ as follows:

$$\begin{cases} Sk_p = \frac{\mu_l - m_l}{\sigma_l} \\ \mu_l = mean\ of\ H(J^w_{mg}) \\ m_l = median\ of\ H(J^w_{mg}) \\ \sigma_l = standard\ deviation\ of\ H(J^w_{mg}) \end{cases} \tag{2.5}$$

In the proposed technique for each edge pixel p in J_{mg}, if the skewness Sk_p is positive, then the pixel remains an edge pixel. In other cases, it converts into a non-edge pixel. So, the proposed technique generates the refine edge-map

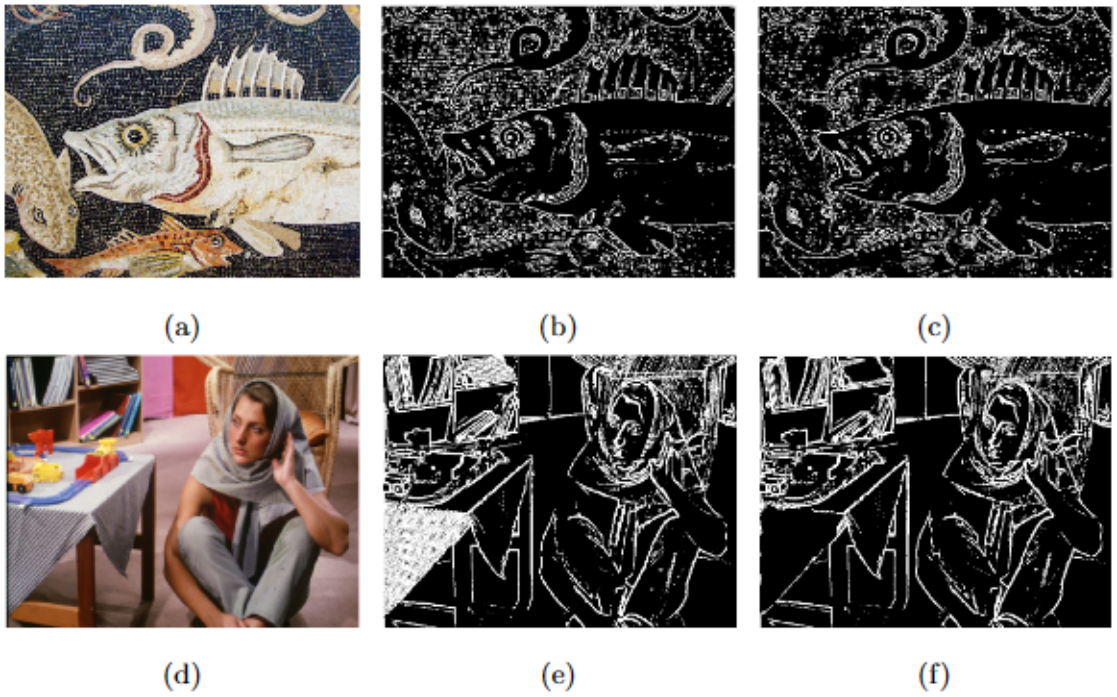

Figure 2-5: (a)(d) Original images, their (b)(e) initial edge-maps E_b, and (c)(f) refined edge-maps E_r produced by the proposed technique.

E_r from the initial edge-map E_b as follows:

$$E_r(p) = \begin{cases} 1, & \text{if } E_b(p) = 1 \; and \; Sk_p > 0 \\ 0, & \text{Otherwise} \end{cases} \tag{2.6}$$

Table 2.1: The skewness values obtained from local histograms are associated with the different pixels of the gradient image.

Test Pixels (p)	***Skewness (Sk_p)***	***Status***
$p1$	1.0715	edge
$p2$	-0.0083	non-edge
$p3$	0.7069	edge
$p4$	-0.0467	non-edge
$p5$	-0.0733	non-edge

Figure 2-5 show the initial edge-map E_b and the corresponding refined edge-map E_r produced by the proposed technique for two different images. For a deeper insight how the local skewness value of the gradient image helped to find whether a pixel is an edge pixel or not, five edge pixels p_1, p_2, p_3, p_4, and p_5 from

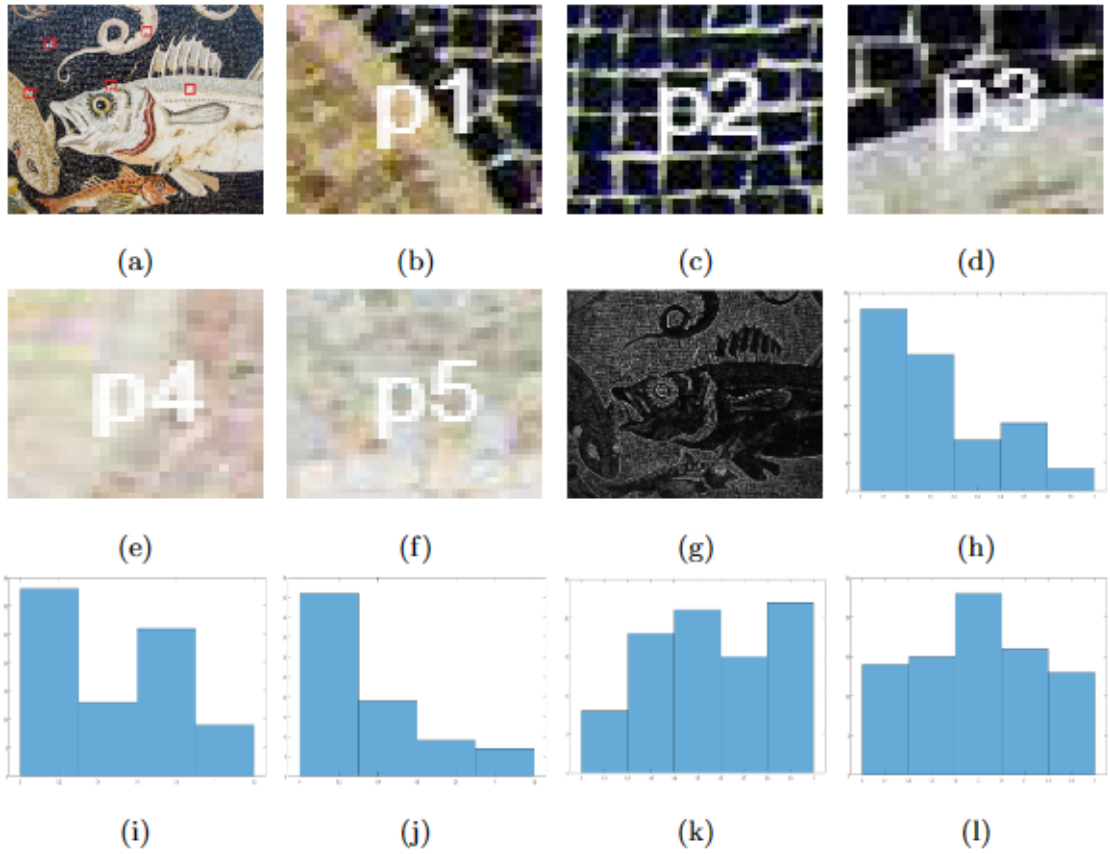

Figure 2-6: (a) Original image I and (b) the corresponding gradient image J_{mg}. (c)(d)(e)(f)(g) The zoomed version of the red windows of center pixels p_1, p_2, p_3, p_4 and p_5 are shown in the original image, and (h)(i)(j)(k)(l) their corresponding local histograms obtained from the gradient image.

different portions of the gradient image J_{mg} obtained from the Figure 2-6(a) are chosen. Considering these pixels as center pixels, the corresponding portions in the original image cover by the local windows (red boxes) are shown in Figure 2-6(a). The zoomed version of these windows are depicted in Figures 2-6(c), 2-6(d), 2-6(e), 2-6(f), and 2-6(g). From these figures, one can see that pixels p_1 and p_3 originally belong to the structural-edge regions, whereas pixels p_2, p_4 and p_5 originally belong to the textural or smoother regions. Figure 2-6(h), Figure 2-6(i), Figure 2-6(j), Figure 2-6(k), and Figure 2-6(l) shows the local histogram (considering window W) of the gradient image J_{mg} associated to the center pixel p_1, p_2, p_3, p_4 and p_5 respectively. From these figures, it is evident that the gradient distribution of the local histograms corresponding to the pixels p_1 and p_3 which are on the structural-edge regions are positively skewed (i.e., their local

histograms from J_{mg} contain few locally higher gradient pixels those are part of the significant edges). Whereas, the distribution of the local histograms corresponding to the pixels p_2, p_4 and p_5 which belong to the textural or smoother regions are not positively skewed (i.e., their local histograms in J_{mg} contain more locally higher gradient pixels or similar gradient pixels those are part of some textural or smoother regions). Table 2.1 reports the skewness values computed for these five local histograms. Form the table one can see that the pixels having positive skewness values are on the structural-edges, and pixels of negative skewness are on the textural/smoother regions. Note that for the presence of high gradient noises, some pixels in the textural region may possess positive skewness values. In such cases, our method may wrongly identify these pixels as structural-edge pixels. To mitigate this, in the proposed work the edge-map for the next iteration is updated based on the filtered image generated in the current iteration.

2.3.4 Median morpho-filtering

To produce the final filtered image, we propose a novel semantic-aware structure preserving filtering technique by exploiting the generated edge-map E_r. Let I be the original image will be filtered and E_r be the corresponding edge-map generated by the proposed technique. By exploiting the edge-map E_r, our technique follows different approaches to determine the dynamic window of the median filter for filtering the edge pixels and non-edge ones. For filtering the pixels belonging to textural regions (i.e., non-edge pixels), our technique determines the dynamic size square window by looking into the surrounding edge pixels present in E_r. For each non-edge pixel, the algorithm starts with a small square window L_w and then increases its size uniformly until a sufficiently large size square window is obtained or an edge pixel is found. Thus, for filtering non-edge pixels our algorithm determine dynamic size square windows by looking into the edge pixel on the edge-map. The blue boxes in Figure 2-7 depict the windows obtained by

the proposed algorithm for some non-edge pixels. From this figure one can see that the larger size windows are formed for filtering the pixels of textural areas which are far from the structural-edges and smaller size windows are formed for filtering the pixels near to the structural-edges. Thus, the size of the windows defined by the proposed algorithm for filtering non-edge pixels is adaptive in nature. On the other hand, conversely, for filtering the pixels that are part of structural-edges (i.e., edge pixels), our technique determines the shape of the window of the median filter dynamically by identifying the appropriate object in which the edge pixel belongs. Note that edge pixels always belong to either one of the two objects that are responsible for forming the edge. By computing the intensity value of the edge pixel within a fixed-size window, our algorithm tried to find out the appropriate object to which the considered edge pixel belongs and then reshape the window for filtering. The red curves in Figure 2-7 formed the dynamic windows by considering the appropriate edge pixels for filtering some structural-edge pixels. The window sizes defined by the developed algorithm for filtering edge pixels is also adaptive in nature.

In greater detail, let p be an edge pixel and its intensity value in the original image I is I_p. Our algorithm first defines a fixed-size square window W_s by using p as a center pixel. Considering the window W_s, let I_{w_s} and $E_r^{w_s}$ be the sub-image obtained from the input image I and the edge-map image E_r, respectively. Let $I_{w_s}^e$ represent the intensity values of the edge pixels in I_{w_s}. The middle intensity value R_{mid} of the edge pixels within W_s will be $R_{mid} = (max(I_{w_s}^e) + min(I_{w_s}^e))/2$. The pixel intensity I_p is larger (or smaller) than R_{mid} implying that the edge pixel p belongs in the higher(or lower) intensity object. Thus, for filtering the pixel p, if I_p is larger than R_{mid} then the pixels in I_{w_s} having higher intensity values than R_{mid} are considered to define the window W_s^u of the median filter. Otherwise, the pixels in I_{w_s} having lower intensity values than R_{mid} are considered to define the window W_s^l of the median filter. In the proposed technique for filtering edge pixels the dynamic size window W_s^a of the median filter within W_s is defined as follows:

$$\begin{cases} R_{mid} = (max(I^e_{w_s}) + min(I^e_{w_s}))/2 \\ W^l_s = \ pixels\ in\ W_s\ for\ which\ (I_{w_s} <= R_{mid}) \\ W^u_s = \ pixels\ in\ W_s\ for\ which\ (I_{w_s} > R_{mid}) \\ \text{if } (I_p <= R_{mid}),\ then\ W^a_s = W^l_s \\ \text{if } (I_p > R_{mid}),\ then\ W^a_s = W^u_s \end{cases} \tag{2.7}$$

Figure 2-7: Adaptive windows for the median filter are designed using the generated edge-map (E_r).

From the above equations, one can see that the dynamic window W^u_s is formed when the center pixel's (I_p) intensity value is higher than R_{mid}, otherwise, it is formed the dynamic window W^l_s for filtering the edge pixel p. The proposed algorithm splits the fixed size window W_s to form an edge adaptive dynamic shape window along the approximate edge alignment. The red curves in Figure 2-7 connected with appropriate edge pixels show the dynamic windows defined by the proposed algorithm for filtering edge pixels. From this figure, one can

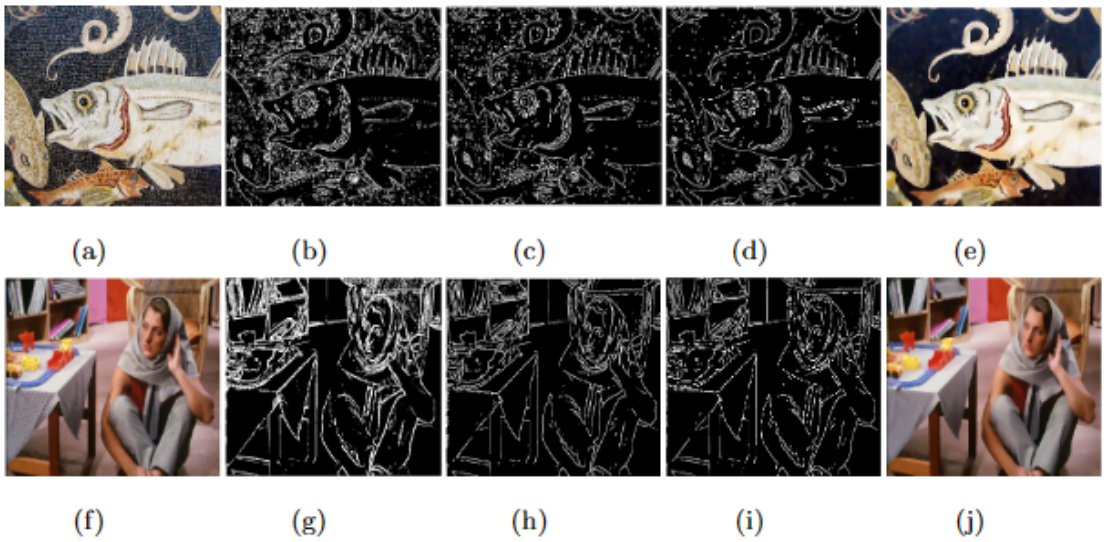

Figure 2-8: (a)(f) Original images and the generated edge-maps after (b)(g) iteration 1, (c)(h) iteration 2, (d)(i) iteration 3. (e)(j) Filtered images produced by the proposed technique after 3^{rd} iteration.

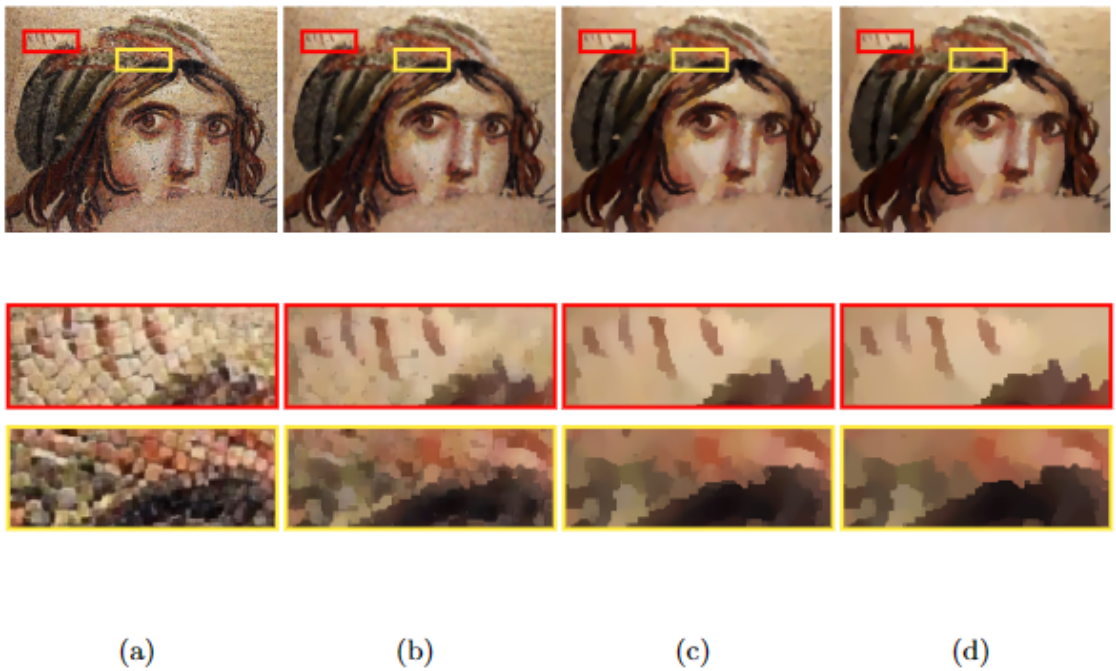

Figure 2-9: (a) Original image and some magnified portions of it. Filtered images obtained by the proposed technique after (b) iteration 1, (c) iteration 3, and (d) iteration 5.

Algorithm 1 Proposed semantic-aware structure preserving filtering technique

Input: Image I
Output: Filtered Image I_f
  Obtain grayscale image I_{gray} from the input image I
  Generate pre-processed image- $J = (\delta_{SE}(\varepsilon_{SE}(I_{gray}) + \varepsilon_{SE}(\delta_{SE}(I_{gray}))/2$
  Generate gradient image $J_{mg} = \delta_{SE}(J) - \varepsilon_{SE}(J)$
for iteration = 1 to n_{itr}
  for all pixel $p \in J_{mg}$
    if $(J_{mg}(p) >= e^{(\mu+\sigma)})$
      $E_b(p) = 1$
    else
      $E_b(p) = 0$
    end if
  end for
  for all pixels $p \in E_b$
    if $(E_b(p) = 1 \ \& \ Sk_p > 0.0)$
      $E_r(p) = 1$
    else-if $(E_b(p) = 1 \ \& \ Sk_p <= 0.0)$
      $E_r(p) = 0$
    end if
  end for
  for image band $c = 1$ to C
    for all pixels $p \in I_c$
      if $(E_r(p) = 1 \ \& \ I_c(p) <= R_{mid})$
        Define window $W_s^a = W_s^l$ using (2.7)
      else-if $(E_b(p) = 1 \ \& \ I_c(p) > R_{mid})$
        Define window $W_s^a = W_s^u$ using (2.7)
      else-if $(E_b(p) = 0)$
        Define window W_s^a by starting from L_w
        to a maximal size square window that does
        not contain any edge content.
      end if
      Replace $I_c(p)$ by the median value from I_c
      within the window W_s^a.
    end for
    $I_f(c) = (\delta_{SE}(\varepsilon_{SE}(I_c) + \varepsilon_{SE}(\delta_{SE}(I_c))/2$
   end for
  $I = I_f$
end for

see that windows at the edge are formed according to the edge linearity. Since there is a possibility that the window division for edge pixels may not always be as accurate as the actual edge alignment. In such cases, a small amount of edge distortion may occur after applying the proposed adaptive median filter. In order to refine these distortions, the morphological operation defined in equation (2.1)

with a small static window is applied. The adaptive median filtering followed by morphological filtering alternatively suppresses/refines the artifacts/distortions of each other. That is, the edge dilation of median filtering regains its sharpness by the morphological counterpart and the impulse edge shape distortion of morphological filtering is smoothed by median filtering. In this work, we refer to this technique as adaptive median morpho-filtering.

All the steps of the proposed technique need to be repeated multiple times to generate the desired filtered image. Let I_f be the filtered image obtained from the input image I. Then the output image I_f is considered as the input image ($I = I_f$) for the next iteration. Figure 2-8 shows the edge-maps produced by the proposed technique at different iterations and the filtered image obtained after 3^{rd} iteration. Figure 2-9 shows the filter images produced by the proposed technique at different iterations. From these figures, it is evident that the developed technique successfully preserves the structure of the objects. The developed technique is outlined in Algorithm 1.

2.4 Experimental results and analysis

To evaluate the effectiveness of the proposed filtering technique, it is compared with six widely recognized structure-preserving smoothing techniques considered as state-of-the-art, such as bilateral texture filtering(BTF) [32], Relativity of Gaussian(RoG) [22], scale-aware texture filtering(SATF) [69], structure adaptive total variation(SATV) [125], fast adaptive bilateral filtering(FABF) [55], and real-time iterative least square(RILS) [91]. To achieve an equitable comparison, the parameters of all these state-of-the-art methods are meticulously chosen. Whenever possible, the parameter values are adopted from the respective papers that reported the best results. Otherwise, we manually adjust the parameter values to achieve the best possible results using a trial-

and-error approach. The images analyzed in this study are accessible publicly at http://www.cse.cuhk.edu.hk/~leojia/projects/texturesep/index.html and http://cg.postech.ac.kr/publications. Detailed comparisons are outlined in the subsequent subsections.

2.4.1 Qualitative comparison

Qualitative comparisons between the developed and the state-of-the-art methods are performed 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. Figure 2-10 shows the filtering results obtained by the developed as well as the six state-of-the-art methods for the part of the Roman marine life mosaic image. By looking into the zoomed portions of the filter images produced by the different methods, it is evident that our method is outperforming the existing methods in both terms: i) texture smoothing and ii) structure preserving. From the figure, it is observed that BTF [32], SATF [69], FABF [55] are capable of preserving the significant structures but producing poor smoothing results for the varying scale textures. Whereas, the RoG [22] filtering technique is good in texture smoothing but fails to preserve the smaller structures, and the RILS [91] produces blur structural objects. On the contrary, our developed method is equally effective in smoothing varying scales texture as well as preserving the significant structures of different sizes. Unlike the other methods, instead of blurring, the proposed method sharpens the filtered image by preserving the structural edges. Moreover, it is capable to preserve the small structural details like the iris and eyelids of the fishes more prominently than the others. In the experiment performed on a large number of images containing different types of textural patterns the proposed technique always produced satisfactory results.

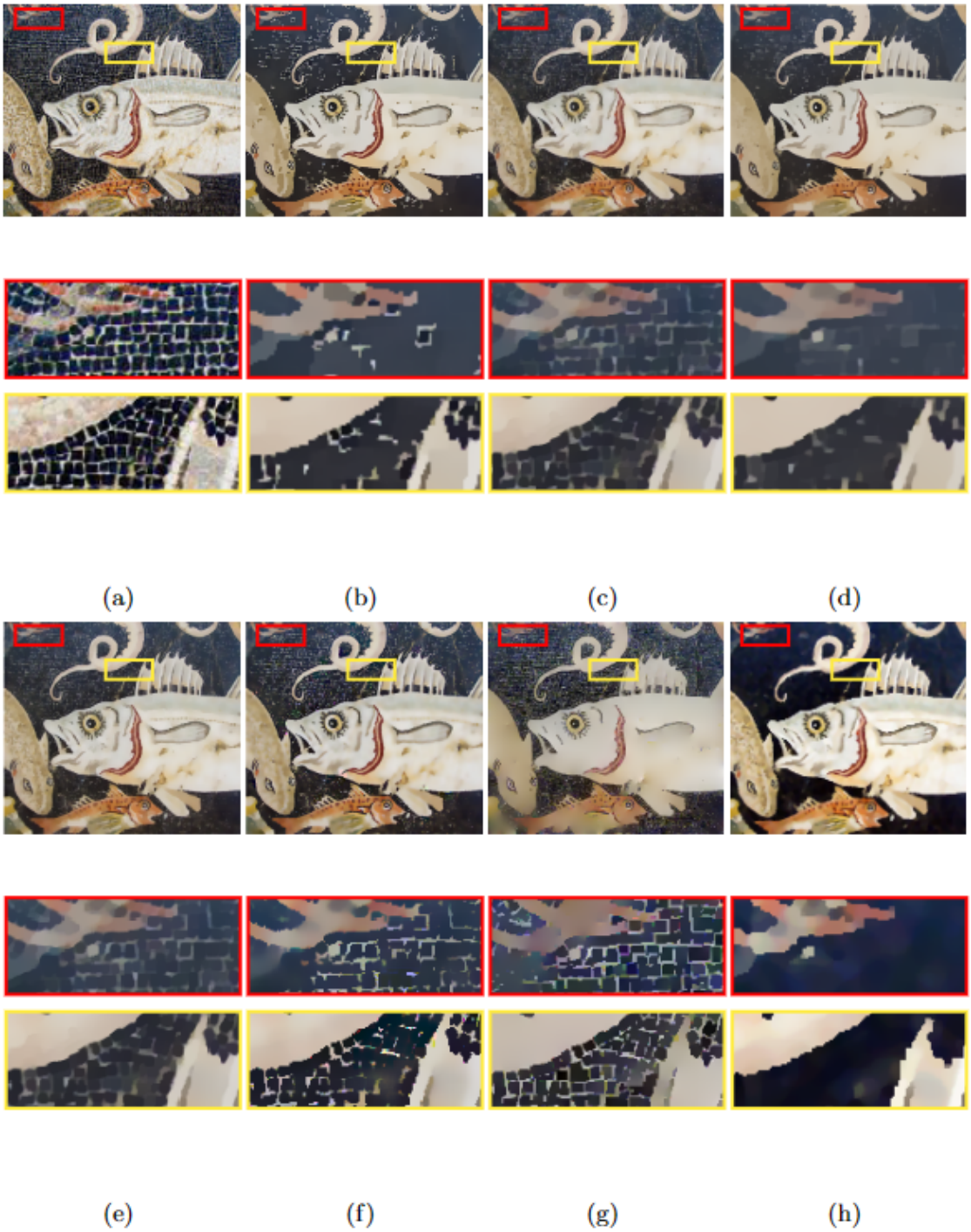

Figure 2-10: Comparative results for part of the Roman marine life mosaic image in Pompeii (a) Original image and some zoomed portions of it. Filtered images generated by applying (b) BTF, $k = 7, n_{itr} = 5$ [32], (c) SATF $ss = 4, sr = 0.05, st = 0.1, n_{itr} = 7, div = 30$ [69], (d) RoG, $\lambda = 0.01, \sigma_1 = 2, \sigma_2 = 4, K = 4, dec = 2.0$ [22], (e) SATV, $\lambda = 1.25, n_{itr} = 19$ [125], (f) FABF $\rho_{smooth} = 5, \rho_{sharp} = 5$ [55], (g) RILS, $\lambda = 0.35, \gamma = 50/255, n_{itr} = 25$ [91], and (h) Proposed, $W = 31 \times 31, n_{itr} = 4$

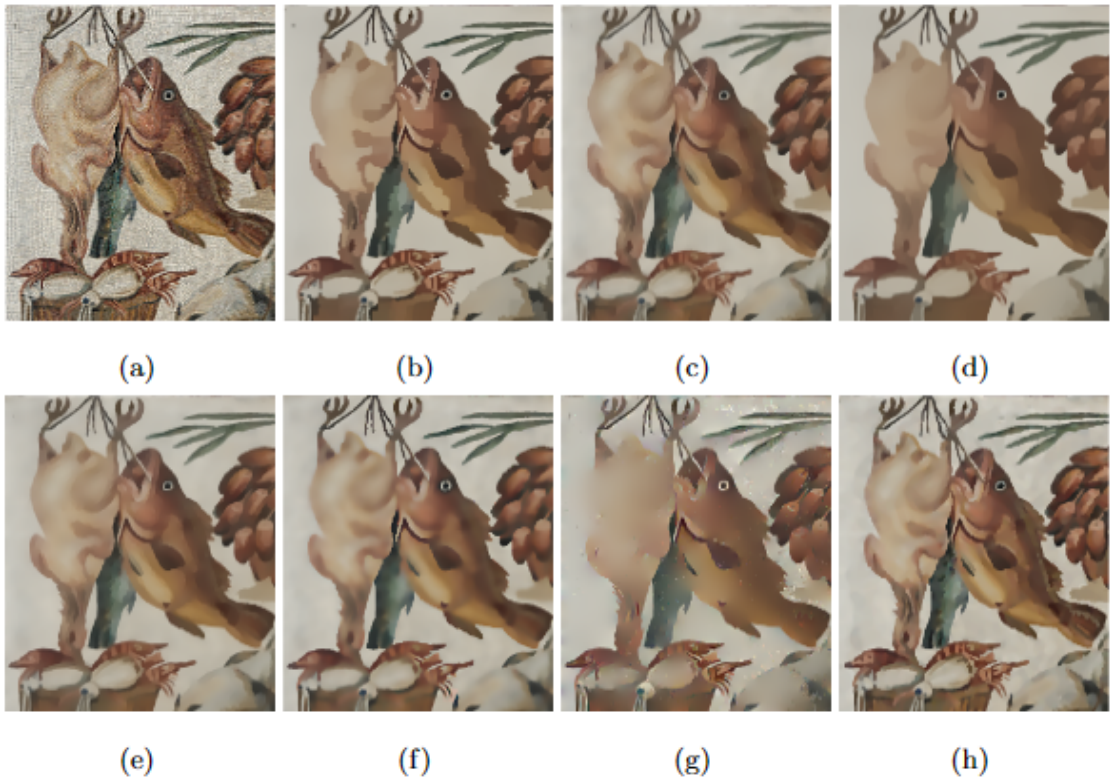

Figure 2-11: Comparative results for part of a roman still life mosaic image: (a) Original image and the filtered images produced by applying (b) BTF, $k = 5, n_{itr} = 3$ [32], (c) SATF $ss = 3, sr = 0.1, st = 0.1, n_{itr} = 7, div = 30$ [69], (d) RoG, $\lambda = 0.01, \sigma_1 = 1, \sigma_2 = 2, K = 2, dec = 2.0$ [22], (e) SATV, $\lambda = 2.5$ [125], (f) FABF $\rho_{smooth} = 2, \rho_{sharp} = 4$ [55], (g) RILS, $\lambda = 0.5, \gamma = 25/255, n_{itr} = 15$ [91], and (h) Proposed, $W = 17 \times 17, n_{itr} = 3$

The filtering results obtained from a few of such images are shown in Figures 2-11, 2-12, and 2-13. From these figures, one can see that for the input images of smaller textural patterns e.g. part of the Roman still life mosaic image in Figure 2-11(a) as well as the images of larger texels e.g. the brick wall graffiti image in Figure 2-12(a), the proposed method always provides better or as good as the best result produced by the existing state-of-the-art methods. Looking into the filtering results in Figure 2-11 and Figure 2-12, one can see that our method outperforms most of the existing techniques. Figure 2-13 displays the filtering results of the developed technique on a few more popular images containing rich textural variation.

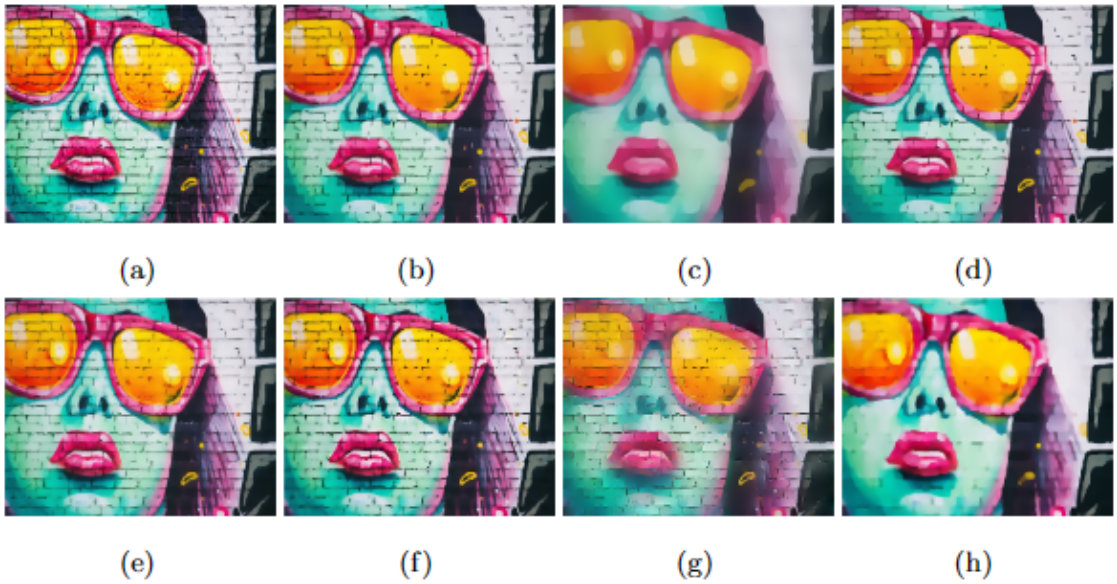

Figure 2-12: Comparative results for a brick wall graffiti image: (a) Original image and the filtered images produced by applying (b) BTF, $k = 9, n_{itr} = 7$ [32], (c) SATF $ss = 7, sr = 0.05, st = 0.1, n_{itr} = 7, div = 30$ [69], (d) RoG, $\lambda = 0.01, \sigma_1 = 2, \sigma_2 = 4, K = 4, dec = 2.0$ [22], (e) SATV, $\lambda = 0.75$ [125], (f) FABF $\rho_{smooth} = 5, \rho_{sharp} = 5$ [55], (g) RILS, $\lambda = 0.75, \gamma = 50/255, n_{itr} = 25$ [91], and (h) Proposed, $W = 7 \times 7, n_{itr} = 7$

Figure 2-13: (a)(b)(c)(d) Original images, and (e)(f)(g)(h) the filtered images obtained by the proposed technique.

2.4.2 Quantitative comparison

From recent studies, it was found that the classical non-subjective Image Quality Assessment (IQA) metrics like Peak Signal to Noise Ratio (PSNR), Signal to Noise Ratio (SNR) are not good enough for assessing the quality of the filtered images, particularly when they are generated by structure-preserving or texture filtering techniques [140]. The subjective reference IQA metrics such as Structural Similarity Index (SSIM) [136], Multi-scale SSIM (MSSIM), Mutual Information (MI) [131] and the subjective no-reference metrics like Perception based Image Quality Evaluator (PIQE) [132] and Naturalness Image Quality Evaluator (NIQE) are more closer to human perception. For additional information on these IQA metrics reader may refer to [131, 132, 136, 140]. To evaluate the effectiveness of the developed technique with respect to both texture smoothing and structure preservation, a quantitative comparison is performed with the help of three subjective references IQA metrics SSIM, MSSIM, MI, and a subjective no-reference metric PIQE, where SSIM and MSSIM have a range of [0, 1], while MI ranges from [0, $\log(m \times n)$], where $m \times n$ denotes the image size, and PIQE ranges from [0, 100]. Higher values for SSIM, MSSIM, and MI indicate better performance, whereas a lower value for PIQE signifies better quality. Table 2.2 reports the values provided by those IQA metrics for different filtered images. This table demonstrates that for the natural texture images, the developed technique always provides superior results contrast to all six state-of-the-art methods considered for comparison. For the Lena image with artificial texture and noise, our method yields competitive results to the best literature method. Thus, by analyzing both the quantitative and qualitative results leads us to conclude that the proposed technique surpasses existing state-of-the-art methods in terms of both texture smoothing and structure preservation.

Table 2.2: The values of the IQA metrics SSIM, MSSIM, MI and PIQE obtained for the filter images generated by the different techniques. The bold font indicates the best values of these IQA metrics.

Images	*Metrics*	*BTF [32]*	*SATF [69]*	*RoG [22]*	*SATV [125]*	*FABF [55]*	*RILS [91]*	*Proposed [111]*
Marine life PIQE=20.18	SSIM	0.43	0.41	0.42	0.42	0.44	0.42	**0.48**
	MSSIM	0.58	0.55	0.50	0.51	0.47	0.58	**0.63**
	MI	1.74	1.72	1.65	1.69	1.67	1.73	**1.80**
	PIQE	89.10	89.56	88.64	82.90	90.03	89.34	**79.37**
Still life PIQE=34.63	SSIM	0.56	0.53	0.56	0.55	0.50	0.57	**0.59**
	MSSIM	0.76	0.69	0.70	0.70	0.61	0.74	**0.77**
	MI	2.30	2.27	2.30	2.28	2.26	2.31	**2.35**
	PIQE	86.86	88.85	86.64	75.28	85.30	86.82	**75.10**
Graffiti PIQE=27.43	SSIM	0.75	0.72	0.78	0.72	0.74	0.71	**0.79**
	MSSIM	0.68	0.57	0.71	0.56	0.58	0.62	**0.74**
	MI	2.03	1.92	2.11	1.90	1.88	1.89	**2.27**
	PIQE	86.74	92.19	87.70	87.23	87.39	83.05	**78.88**

2.4.3 Parameters setting and analysis

To incorporate information from neighboring pixels, the proposed technique considers a few windows. The window W used to form the local histogram of the gradient image for computing the local skewness is one such parameter of the proposed technique. The optimum size of this window is dependent on the scale and size of the texels (single textural pattern) present in the input image. In the experiment, the size of W is set manually by looking at the input image. The square windows L_w and W_s of the proposed technique are the spatial parameters supplied as input to the median filter for filtering textural and structural-edge pixels, respectively. Since our technique increases the initial size of L_w in an adaptive manner, the size of the square window L_w should be small (can be taken from 3×3 to 11×11 depending on the size of the small structure to be preserved) and is not a critical parameter to set. In the experiment, the initial size of L_w is set to 3×3 for all the considered images. As for filtering structural-edge pixels, the window W_s is divided into two parts along with the edge alignment. The size of W_s should cover sufficient pixels to form two groups. If we fix the size of W_s

very large, then there is a possibility of overlapping the textural and structural edges. Experimentally we found that the size of W_s between 5×5 to 11×11 provides satisfactory results for a wide variety of images. Keep in mind that the size of the square window W_s is not a critical parameter to set as our technique dynamically reshapes it before filtering. Taking its size smaller preserves more but smoothing needs more iterations. In the experiment the size of W_s is set to 7×7 for all the considered images. For morphological filtering, the proposed technique uses a small SE of size 3×3. Table 2.3 illustrated the list of parameters of the developed technique and their suggested values. From this table, one can see that the proposed technique has only a critical parameter W. Figure 2-14 shows filter images and their corresponding values of different IQA metrics of the developed technique for different window sizes W.

Table 2.3: List of parameters of the proposed technique and their suggested values.

Parameters	***Used for***	***Suggested range***	***Taken in experiment***
W	Local gradients skewness calculation	5×5 to 41×41	set manually
L_w	Filtering textural pixels	3×3 to 11×11	3×3
W_s	Filtering edge pixels	5×5 to 11×11	7×7
SE	Morphological operations	3×3 or 5×5	3×3
n_{itr}	No of iterations	1 to 10	set manually

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

Figure 2-14: (a) Original image of a mosaic floor (PIQE=38.97) and some zoomed portions of it. Filtered images obtained by the proposed technique after iteration 3 for (b)$W = 7 \times 7$ (SSIM=0.75, MSSIM=0.76, MI=2.27, PIQE=79.92), (c)$W = 15 \times 15$ (SSIM=0.75, MSSIM=0.77, MI=2.28, PIQE=78.34), and (d)$W = 23 \times 23$ (SSIM=0.76, MSSIM=0.77, MI=2.28, PIQE=78.50).

2.4.4 Applications

Structure-preserving filtering techniques find widespread applications in the realm of image processing and analysis. They prove valuable as both pre-processing and post-processing tools for tasks such as image denoising, enhancement, tone mapping, abstraction, classification, segmentation, and more.

Below are brief descriptions of some specific applications where the proposed filtering technique demonstrates its efficacy.

2.4.4.1 Image denoising

Image denoising involves the removal of unwanted or random variations in pixel values, known as noise, from digital images. Noise often accompanies image ac-

Figure 2-15: (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.

quisition, transmission, or processing, leading to a degradation in image quality. The objective of denoising is to redevelop the original, clean image by reducing or removing the impact of noise. 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.

2.4.4.2 Detail enhancement

(a) (b)

(c) (d)

(e) (f)

Figure 2-16: (a)(c)(e) Original images and their corresponding (b)(d)(f) enhanced images obtained by the proposed technique.

Image enhancement encompasses the modification of an image to enhance its visual quality, rendering it more suitable for human perception or specific computer vision tasks. These enhancement techniques aim to accentuate particular features, amplify contrast, reduce noise, and overall elevate the image's quality. 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 layers - one being a smoothed image and the other capturing textural details. The final enhanced image is produced by appending these textural details to the original image. The enhanced images prominently highlight the textural details, showcasing the impact of the developed filtering technique.

2.4.4.3 Tone mapping

Tone mapping is a technique employed in computer graphics and image processing to transform high-dynamic-range (HDR) images into a format suitable for display on devices with lower dynamic ranges (LDR), such as computer monitors, TVs, or printed media. HDR images capture a broader range of luminance values compared to standard images, and tone mapping assists in rendering these images on regular displays while retaining as much visual information as possible. 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 the developed technique in the context of tone mapping.

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

Figure 2-17: (a)(c) Tone mapped images generated by Farbman's method [47] and (b)(d) tone mapped images generated by Durand's method [49] integrating proposed filtering technique.

2.4.4.4 Edge detection

Edge detection is a fundamental step in image processing, impacting applications such as object recognition, segmentation, and feature extraction. Traditional edge detection methods like the Sobel, Prewitt, and Laplacian operators often face challenges in accurately detecting edges in noisy or complex images. The Canny Edge Detector is one of the popular techniques that use sobel operator with Gaussian smoothing. Early edge detection methods like Sobel and Prewitt operators relied on calculating directional (horizontal or vertical) intensity gradients using simple 3x3 convolution kernels. These methods were computationally efficient and easy to implement but suffered from noise sensitivity, especially in high-frequency areas. Laplacian and Laplacian of Gaussian (LoG) operators improved edge localization by highlighting regions of rapid intensity change; however, they were also noise-prone and often required additional smoothing to avoid spurious edges. The Canny Edge Detector, with its multi-step process (including Gaussian smoothing, gradient calculation, and edge tracking), set a new standard by improving edge continuity and reducing noise, though at a higher computational cost.

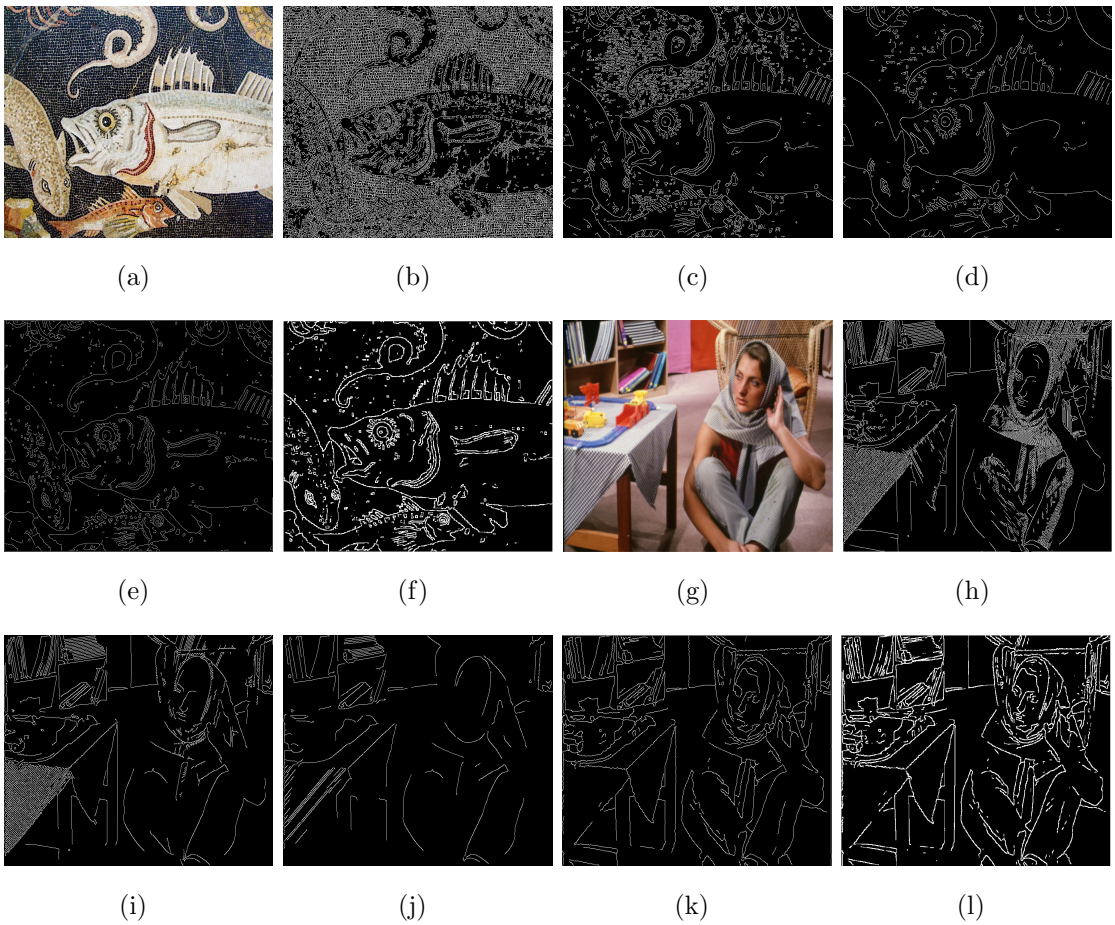

Figure 2-18: (a)(g) Original images and (b)(h) Edge detected from the original images, (c)(i) Edge detected after 5×5 Gaussian smoothing, (d)(j) Edge detected after 9×9 Gaussian smoothing, (e)(k) Edge detected after smoothing by the proposed technique using Canny operator, and (f)(l) Edge detected by the proposed technique with proposed morphological gradient-based approach.

In this work, we propose a morphological gradient-based approach to address edge detection challenges. The morphological gradient offers advantages over traditional directional gradients, as it detects intensity changes in all directions. However, applying the morphological gradient directly for edge detection can result in sensitivity to noise and high frequencies, similar to the existing methods. To mitigate this, our approach explores both the local and global distribution of the morphological gradient, which reduces the noise and high frequency sensitivity while effectively identifying semantic edges. In the Figure 2-18 presents a comparative analysis: the edge detection results from the Canny detector applied to two original images, followed by results after Gaussian smoothing with masks of sizes

5×5 and 9×9, respectively. We then show the edges detected by the canny edge detector from the filtered images of the proposed technique, along with the results from our morphological gradient-based approach. These comparisons highlight the effectiveness of our technique in precisely detecting semantic edges.

2.4.5 Computational performance analysis

Table 2.4: The execution times (in seconds) of the developed method in contrast to the state-of-the-art methods.

Images	*BTF [32]*	*SATF [69]*	*RoG [22]*	*SATV [125]*	*FABF [55]*	*RILS [91]*	*Proposed*
Marine life (675 × 900)	56.04	77.95	9.96	5.57	7.47	3.21	56.67
Still life (600 × 450)	11.05	53.46	5.49	3.94	5.05	2.98	19.01
Graffiti (640 × 960)	46.42	101.58	10.08	7.50	7.76	3.51	88.20

The proposed technique has two important steps. First, it generates the edge-map, then based on the generated edge-map, adaptive median morpho-filtering produces the filtered image. The edge-map generation followed by median morpho-filtering is iteratively executed to obtain the final filtered image. The edge-map generation step has the complexity of $\mathrm{O}(M \times N \times W)$, where $M \times N$ is the size of the image and W is the window for local gradient skewness calculation. The adaptive median morpho-filtering has the complexity of $\mathrm{O}(M \times N \times W_s^a)$, where W_s^a is the adaptive window for filtering edge and non-edge pixels. So, the computational complexity of the developed algorithm is $\mathrm{O}(M \times N \times W^{'})$, where $W^{'} = max\{W, W_s^a\}$.

The proposed technique and the available Matlab codes of the 6 state-of-the-art techniques used for comparison are executed on a 3.70 GHz Xeon(R)-8 Core(s) processor with 128 GB RAM. Table 2.4 shows the execution time (in seconds) of different techniques. From this table, it is evident that although the developed technique takes lesser time than the existing SATF technique and comparable time with BTF technique, it takes significantly higher time than the RoG,

SATV, FABF, and ILS techniques. Note that our current Matlab implementation is worked on a CPU environment and is not heavily optimized. It takes some time for skewness, adaptive windows, and median computation, which can be significantly enhanced through parallel GPU implementation [60] and by employing more advanced sampling [64] and sorting algorithm (like counting sort [118]).

2.5 Conclusions

The main challenge for developing a robust structure preserving filtering technique lies in the fact of handling varying scale textural patterns. In order to better handle varying scale textural patterns, in this chapter we have proposed a semantic-aware structure preserving filtering technique. Our technique defines a novel method to obtain an edge-aware adaptive window of dynamic shape for filtering each pixel by excluding its neighbour pixels belonging to different textural or structural regions. To this end, at first, a novel approach is proposed to generate the edge-map by analysing the skewness of global and local histograms of the morphological gradient. Then, using the generated edge-map a semantic-aware structure preserving median morpho-filtering is designed by combining the output of median and morphological filters. The key advancements of this work are: i) proposes a texture structure decomposition technique analysing the global and local morphological gradient distribution, and ii) proposes a combined median morpho-filtering for better texture smoothing while preserving the significant structures. Our filtering method, unlike most existing techniques, can simultaneously achieve several conflicting goals, including identifying and removing texture, preserving structural edges, safeguarding subtle features such as corners, and preventing both over-sharpening and over-blurring artifacts or distortions.

Although the developed technique successfully works on a broad spectrum of images but our present implementation is not of real-time. Implementation of the proposed technique in real-time or near real-time and apply this technique to

different image/video processing applications like semantic segmentation, object detection and classification etc would be an interesting extension of this work.

List of publications from this chapter

Journals

1. Pradhan K., Patra S., "Semantic-aware structure-preserving median morpho-filtering". The Visual Computer, 2023, March 5:1-7, https://doi.org/10.1007/s00371-023-02796-z. ***(IF-3.0)***