Research Method- Hybrid Impulse Noise Removal (HINR) Filter

Earlier, the Order Statics Filters and Rank Order Rank Selection Filters were used to

remove the impulse noise. But, those filters does not able to distinguish between minute details

such as sharp edges and noise, also its performance becomes poor with increasing level of

noise density. To eliminate all these drawbacks, in this paper we propose a Hybrid Impulse

Noise Removal (HINR) filter. This filter is used to identify and handle high density impulse noisy

color images and it covers wide range of samples of an image. It also preserves the fine details

such as edges as well.

The novel HINR filter algorithm comprised of the following steps:

Step 1:

Convert the RGB input image in to grayscale using the relation:

Grayscale=R\*0.21 + G\*0.72 + B\*0.01 (1)

Step 2: **(Modified)**

Let 'n' be the size of the window (typically set to 9). For each 3x3 neighborhood centered at pixel (x, y) in the grayscale image:

a. If f(x, y) < 255, retain all 8 neighboring pixels.

b. If f(x, y) == 255, apply the following conditions:

**Condition-1:** If (f(x-1, y+1) == 255) and if (f(x, y+1) == 255) and if (f(x+1, y+1) == 255) and if (f(x-1, y) == 255) and if (f(x+1, y) == 255) and if (f(x-1, y-1) == 255) and if (f(x, y-1) == 255) and if (f(x+1, y-1) == 255), then compute the sum using the following weighted relation:

Sum = (f(x-1, y+1) + f(x, y+1) + f(x+1, y+1) + f(x-1, y) + f(x+1, y) + f(x-1, y-1) + f(x, y-1) + f(x+1, y-1)) / 8

Replace the center pixel with the computed weighted average.

Condition-2:

Initiate Sum=0

If (f(x-1,y+1) !=255) then

Sum=Sum+f(x-1,y+1)

If (f(x,y+1)!=255) then

Sum=Sum+f(x,y+1)

If (f(x+1,y+1)!=255) then

Sum=Sum+f(x+1,y+1)

If (f(x-1,y)!=255) then

Sum=Sum+f(x-1,y)

If (f(x+1,y)!=255) then

Sum=Sum+f(x+1,y)

If (f(x-1,y-1)!=255) then

Sum=Sum+f(x-1,y-1)

If (f(x,y-1)!=255) then

Sum=Sum+f(x,y-1)

If (f(x+1,y-1)!=255) then

Sum=Sum+f(x+1,y-1) then the average is to be computed and is replaced in the center

pixel.

Step 3: (Modified)

Apply the bilateral filter to the output image obtained from Step 2. The bilateral filter is a powerful edge-preserving filter that considers both spatial and intensity differences when filtering an image. It helps enhance edges while minimizing smoothing.

The bilateral filter operation replaces each pixel value with a weighted average of its neighbouring pixels, where the weights are determined by both spatial proximity and similarity in pixel values.

Step 4:

Subtract the blurred image from the obtained image through step (2). The resultant

image contains only the enhanced edge information.

Step 5:

The image obtained from step (4) is added to the image obtained from step (2) to get

the sharpened image and thus the smoothing effect in the filtered image is eliminated.

1. **Adaptive Window Size:** Instead of using a fixed 3x3 window, consider using an adaptive window size based on the local characteristics of the image. You can use techniques like local variance or gradient magnitude to determine an appropriate window size for each pixel.
2. **Weighted Averaging:** In Step 2, when computing the average for replacing the center pixel, you can consider using weighted averaging based on the distance from the center pixel. Pixels closer to the center should have a higher weight in the average, which can help preserve finer details.
3. **Noise Estimation:** Before applying the HINR filter, consider estimating the noise level in the image. This estimation can be used to adapt the filter's parameters or to decide whether to apply the filter at all. If the noise level is low, you may skip the filter to avoid unnecessary smoothing.
4. **Edge Detection:** Instead of using a simple box blur for sharpening in Step 3, consider applying an edge-preserving filter, such as the bilateral filter or an edge-aware sharpening filter. These filters can enhance edges while reducing smoothing artifacts.
5. **Non-Local Means Filtering:** Consider incorporating a non-local means filter in your algorithm. This filter can help remove noise while preserving image details by considering similarities between patches of pixels rather than just local neighbourhoods.
6. **Parallelization:** To improve computational efficiency, you can parallelize certain parts of the algorithm, especially the pixel-wise operations in Step 2 and the image subtraction/addition in Steps 4 and 5. This can be particularly useful for processing large images.
7. **Boundary Handling:** Ensure that the algorithm handles image boundaries properly to avoid artifacts at the edges of the image.
8. **Performance Metrics:** Implement performance metrics to quantitatively evaluate the filter's performance, such as Peak Signal-to-Noise Ratio (PSNR) or Structural Similarity Index (SSI). This will help you objectively assess the quality of your modified algorithm.
9. **Consider Deep Learning:** Depending on your dataset and the complexity of the noise patterns, you may explore the use of deep learning-based approaches for impulse noise removal. Convolutional Neural Networks (CNNs) can learn complex noise patterns and perform exceptionally well in image denoising tasks.

Choose gaussian noise, impulse noise, exponential noise, blurred image from satellite.

Digital image processing by steither

Digital image processing by 3 authors -Jayram and Esakkirajan

As discussed, these are the Goals for next review:

1.Complete description of dataset for 3 different types with real time images as per the panels suggestion either in medical or in remote sensing areas (minimum 25 images in each category) (target date -3-10-23)

2.Graph- Histogram graph or from the source paper as the reference to define the type of images and keep the source or histogram graph in the same folder (target date - 9th oct)

3.implementation of the image enhancement technique for one type of noise with quantitative metrics such as MSE and PSNR (Target date -23rd Oct)

4.Defining the algorithms for next 2 types (target date- 13th Nov)

5.Submission of PPT by mentioning all the above work (target date-22nd Nov)

6.Review - 2 on 29th Nov