**Image Enhancement of Magnetic Resonance Imaging under Clustering Environment**

Dr.V. Murugan

Assistant Professor & Head

Department of Computer Science

Government Arts and Science College

Kadayanallur, Tenkasi District, India

smv.murugan@gmail.com

**ABSTRACT**

The goal of image denoising is to take the noise out of an image. Instead of the original pixel values, noise in an image can be seen as pixels with variable intensity values. The primary cause of noise may arise during the acquisition or transmission of images. Thus, practically it is not possible to escape from noise. The amount of noise in an image is influenced by the image's quality. Therefore, in order to enhance the quality of an image, it is important to eliminate or reduce such noise. This chapter focuses on salt and pepper and gaussian noise and is based on client-server architecture. The server divides the input image into four parts, and each part is shared by the client systems. Client systems use denoising algorithms on the partitioned input and send the results to the server. To create a single denoised image, the server combines all of the denoised images. With this effort, a large amount of time is saved, and the image is denoised quickly.

**Keywords-**noise; filter; cluster; master-slave environment

# INTRODUCTION

As it extends to multiple disciplines of image processing, including image classification, image registration, and image segmentation, among others, image denoising is a difficult task. The amount of noise in an image is inversely correlated with its quality. Some of the main sources of noise are poor lighting, unfavourable sensor temperatures, signal interference during image transmission, etc. As a result, a number of techniques have been put out in the past for lowering or eliminating the noise in an image.[1].

 A denoising algorithm's primary goal is to remove noise from an image while maintaining the image's edges. To remove the noise from an image, the appropriate denoising algorithm is first used when the noise is found in the image.

 The entire image is degraded by black and white dots if the image is affected to salt and pepper noise. The name "Salt and Pepper" is pronounced with reference to the similarity of the noise. Due to sudden changes in the signal or during image acquisition, this kind of noise may be visible in an image. Other names for the salt and pepper noise are impulse noise, spike noise, and random noise [2]. Each pixel in the noisy image is the sum of the original pixel value and the dispersed Gaussian noise value. Gaussian noise is distributed throughout the image. Spatial and transform domain filtering are two distinct strategies used in the process of image denoising. [3].

Spatial filtering is the simple concept for denoising and this is divided into linear and non-linear filters. Linear filters introduce smoothness into an image and it may miss the important detail of an image. Some of the linear filters are Wiener and mean filters. Non-linear filters eliminate the noise from an image. The best example for non-linear filter is median filter [4].

Median filter follows the concept of windowing. This filter eliminates the noise by calculating the median of the window and the center pixel of the image is modified with it.

A filtering technique is proposed in [5] for eliminating the Salt and Pepper noise in binary images is proposed. This technique works on the basis of computation over multi-diagonal binary matrix of noisy binary image and the thresholding operation is carried out.

In [6], an adaptive multi-column stacked sparse denoising auto-encoder is presented. Multiple stacked sparse denoising auto-encoders is combined by computing column weights and a network is trained to predict the optimal weights. The need of noise type detection is eliminated in this system and hence it is robust.

[7] presents a region classification method based on neural networks. An image's regions are divided into homogeneous and texture areas. The statistical inputs are used to train a neural network. Shearlets and wavelets, respectively, are used to denoise the two groups of regions, namely homogeneous and texture regions.

In [8], an adaptive mean filtering with a fuzzy foundation is proposed. According to this process, the membership value of each pixel is used to determine if a pixel is repaired or uncorrupted. The mean of the uncorrupted pixels is used to replace the corrupted pixel's value.

Motivated by the above mentioned works, it is planned to reduce the time taken to denoise an image and thus the client server architecture is employed. This system is effective in noise detection and removal.

# PARALLEL COMPUTING AND DISTRIBUTED SYSTEMS

**A. Parallel Computing**

A different approach to solving issues that need for managing or processing massive amounts of information in a reasonable length of time is parallel [9]. The ability to generate several tasks that collaborate to solve problems is a feature of parallel processing programmes. The primary concept is to break the problem down into manageable parts and solve them simultaneously to divide the time. The architecture is chosen based on the needs of the application and the available funds. There are three basic ways that parallelism can be used in image processing applications:

1) Data Parallel

2) Task Parallel

3) Pipeline Parallel

**1. Data Parallel:**

The data is partitioned and distributed across the processing units in the data parallel technique [9]. Effective data decomposition and result composition is the key challenge. Load balance is the major concern that must be taken into account for effective parallel execution. The distribution of image data among computing units should be done so that there is less unwanted communication among them and that each unit receives approximately the same load.

The data parallelism to image data can be applied using one of three basic ways:

i) Pixel parallel

ii) Row or column parallel

iii) Block parallel.

Row/Column parallel or block parallel is now used by the most of parallel image processing applications.

1. **Task Parallel:**

In accordance with the task parallel approach [9], low level operations and image processing instructions are divided into tasks and each work is sent to a particular computing unit. There are various different operations that go into an image processing application. Effective data segmentation and result composition are the primary task parallel approach challenge.

1. **Pipeline Parallel:**

Images can be pipeline processed if an image processing application has to process numerous images. Images will go through many stages of processing in a pipeline simultaneously. For a parallel programme to operate correctly and effectively, it needs to include a few features. These characteristics consist of the following:

**B. Distributed System:**

There are two main architectures of distributed systems

1) The master slave

2) Peer to peer.

These are discussed below:

**Master Slave:**

The master slave architecture approach uses the “Distribute Compute and Gather” philosophy for parallel image processing. The master processing unit divides and distributes the image data to the slave processing units in this architecture style. The allocated task is completed in parallel by all slave processing units. The image is then collected and put back together by the master processing unit.

**Peer to Peer:**

Each participating component in a peer-to-peer architecture has the same capabilities, and either component can start a communication. Without need central coordination, the participating entities make some of their resources readily available to other networked involved nodes.

# PROPOSED ARCHITECTURE

Client-server architecture is used in Image Denoising Client Server Architecture (IDCSA) to divide the burden among the resources available and to efficiently utilize all of the resources. The server divides the input image into four parts, and each portion is shared by the client systems. Client systems use denoising algorithms on the partitioned input and send the results to the server. To create a single denoised image, the server combines all of the denoised images. With this effort, a significant amount of time is saved, and the image is denoised quickly.

The system is depicted in Fig.1. The system is depicted more clearly in Fig.2.The advantages of using this method are:

* We can use different denoising methods in each client system.
* Processing time is reduced.

The disadvantages of using this method are:

* If any of the clients is crashed, the total system fails.
* The server may get overloaded if more images are processed

**Figure 1:Overview of IDCSA**

**Figure2: Image splitting and merging by IDCSA**

# EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

Images of various image sizes will be taken in order to analyse and compare performance. The performance of IDCSA will be evaluated using various image quality metrics as well as other parallel computing parameters like speedup, efficiency, serial time, parallel time, reaction time, and resource consumption.

1. **Speed up :**

It is the ratio of sequential execution time to parallel execution time, where sequential execution time is the whole amount of time it takes for each task to be computed individually and parallel execution time is the length of the scheduling on a small number of processors.

1. **Efficiency:**

A parallel program's performance is a measure of processing utilization. It is given as

 EFF=Sp/Np (1)(1)

Where,

Sp=Speedup,

Np=Number of processors.

1. **Overheads:**

The extra time required to complete the computations can be used to measure the overhead of the parallel programme.

 Overheads = parallel time – (Serial time/No. of processors) (2) (2)

1. **Fork Time:**

It is described as the interval of time when data is divided among the available processors.

1. **Join Time:**

It is defined as the point in time when the outcome of the number of processors is known.

This chapter identifies the outcomes of the parallel IDCSA implementation. Along with the evaluation of the performance obtained by parallel computation, this also deals with the environment utilised in the experiments, the test images, and the results.

Table1 shows the performance comparison of client server system and single system. The values in the table are given in milliseconds.

**Table 1 Performance analysis of single and client server system**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **System type** | **Image Name** | **Image size** | **Fork time** | **Processing time** | **Join time** | **Total Processing Time** | **Efficiency** | **Speed up** | **Overhead** |
| Single system | MRI Image | 512$×$512 | 5 | 8 | 3 | 16 | 86 | 71 | 63 |
| Client server system | MRI Image | 512$×$512 | 3 | 6 | 2 | 11 | 92 | 85 | 48 |

In the above Table 1, it is observed that the proposed architecture reduces the total processing time to 11ms for image of size 512 x 512.The overhead is also reduced by converting to client server system. Figure 3 shows the performance comparison chart of single system and client server system.

**Figure 3: Barchart showing Comparison of single system and client server system**

The system is also tested for various image sizes. Table 2 shows the analysis of various image sizes in single system.

**Table 2 Processing in single system on various image sizes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Image Name** | **Image Size** | **Fork Time** | **Processing Time** | **Join Time** | **Total Processing Time** | **Efficiency** | **Speed** | **Overhead** |
| Image1 | 512$×$512 | 5 | 8 | 3 | 16 | 86 | 71 | 63 |
| Image2 | 256$×$256 | 4.5 | 7.7 | 2.8 | 15 | 88 | 73 | 60 |
| Image3 | 128$×$128 | 4.1 | 7.3 | 2.5 | 13.9 | 89 | 76 | 58 |
| Image4 | 64$×$64 | 3.8 | 6.9 | 2.0 | 12.7 | 90 | 79 | 55 |
| Image5 | 32$×$32 | 3.5 | 6.5 | 1.8 | 11.8 | 91 | 81 | 53 |

From Table 2, it is studied that if the image is reduced, the processing time is also reduced. Even if the size of the image is reduced to half, the processing time is not reduced to half. The efficiency, speed and overhead is also reduced when image size is reduced.

Table 3 shows the Processing time in client-server system on various image sizes. The same images are chosen for single and client-server system.

**Table 3 Processing in client-server system on various image sizes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Image Name** | **Image Size** | **Fork Time** | **Processing Time** | **Join Time** | **Total Processing Time** | **Efficiency** | **Speed** | **Overhead** |
| Image1 | 512$×$512 | 3 | 6 | 2 | 11 | 92 | 85 | 48 |
| Image2 | 256$×$256 | 2.8 | 5.6 | 1.9 | 10.3 | 93 | 86 | 46 |
| Image3 | 128$×$128 | 2.6 | 5.4 | 1.7 | 9.7 | 94 | 88 | 44 |
| Image4 | 64$×$64 | 2.4 | 5.2 | 1.5 | 9.3 | 94.5 | 89 | 42 |
| Image5 | 32$×$32 | 2.2 | 5.0 | 1.3 | 8.7 | 96 | 92 | 40 |

In client server system also, the processing time is very much reduced if the image size is reduced.

**V. CONCLUSION**

 A novel system called Image Denoising based on Client-Server Architecture (IDCSA) is suggested in this work. This framework is based on the "Divide and conquer" strategy. To take use of the parallel processing capabilities of today's multi-core processors, a parallel implementation of the algorithm is developed using Matlab threads, and the speed, efficiency, and parallel time are all acceptable. We also given importance to other visual properties, and the outcomes were assessed. When IDCSA's performance is compared to that of numerous other current systems, IDCSA performs better. The execution time of this system is appreciable.

**References:**

1. RohitVerma, Dr.Jahid Ali, "A Comparative Study of Various Types of Image Noise and Efficient Noise Removal Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, 3(10), pp. 617-622, 2013.
2. Charles Boncelet, “Image Noise Models”. Alan C.Bovik. Handbook of Image and Video Processing, second Edition,2005.
3. Reza Ahmadi., JavadKangaraniFarahani, FarbodSotudeh, AshkanZhaleh, SaeidGarshasbi, “Survey of Image Denoising Techniques”, Life Science Journal, 10(1), pp.753-755,2013.
4. Windyga, S. P, “Fast Impulsive Noise Removal”, IEEE transactions on image processing, 10(1), pp.173-178, 2001.
5. GholamrezaAnbarjafari, HasanDemirel, and Ahmet E. Gokus, "A Novel Multi-diagonal Matrix Filter for Binary Image Denoising", Journal of Advanced Electrical and Computer Engineering, 1, pp.14-21, 2014.
6. Forest Agostinelli Michael R. Anderson Honglak Lee, "Adaptive Multi-Column Deep Neural Networks with Application to Robust Image Denoising", Advances in Neural Information Processing Systems, pp.1493- 1501, 2013.
7. Preety D. Swami, Alok Jain, and Dhirendra K. Swami, "Region Classification Based Image Denoising Using ShearletAnd Wavelet Transforms', International Journal of Computer Science and Information Technology, 6(1), pp.241-248, 2014.
8. Punyaban Patel, Bibekananda Jena, BanshidharMajhi, C.R.Tripathy, "Fuzzy Based Adaptive Mean Filtering Technique for Removal of Impulse Noise from Images", International Journal of Computer Vision and Signal Processing, 1(1), pp.15-21, 2012.
9. Preetikaur ,“Implementation of Image Processing Algorithms on The Parallel Platform Using Matlab”, International Journal of Computer Science & Engineering Technology (IJCSET), ISSN : 2229-3345 Vol.4 No.06, Page: 696-706, Jun 2013.