SELF-TUNING ALGORITHM FOR NOISE SUPPRESSION IN COLOUR IMAGES

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

S.Sai Ujwala (317126512114)

P.Suhasini(317126512102)

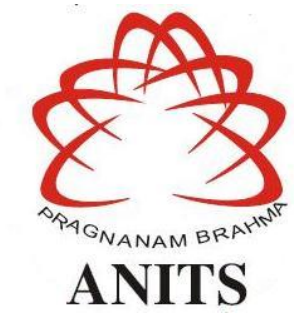
M.Asha Sumitra(317126512092)

P.Kavya (318126512L23)

Under the guidance of

Dr.J Bhaskar Rao

Assistant Professor,Department of ECE



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)
Sangivalasa, bheemili mandal, visakhapatnam dist.(A.P)

2020-2021

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)

Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

ANITS

CERTIFICATE

This is to certify that the project report entitled **"SELF TUNING ALGORITHM FOR NOISE SUPRESSION IN COLOUR IMAGES"** submitted by **S Sai Ujwala (317126512114), P Suhasini (317126512102), Asha Sumitra (317126512092), P Kavya (318126512123)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide
Dr. J Bhaskar Rao M.E,Ph.D
Assistant Professor
Department of E.C.E
ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of the Department
Dr.V.Rajyalakshmi M.E,Ph.D,MIEEE,MIE,MIETE
Professor & HOD
Department of E.C.E
ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa - 531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Dr.J.Bhaskar Rao** Assistant Professor, Department of Electronics and Communication Engineering, ANITS, for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr.V.Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

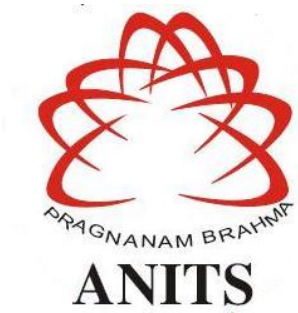
We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS
S Sai Ujwala (317126512114),
P Suhasini (317126512102),
Asha Sumitra (317126512092),
P Kavya (318126512l23)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)
Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)



CERTIFICATE

This is to certify that the project report entitled **"SELF TUNING ALGORITHM FOR NOISE SUPRESSION IN COLOUR IMAGES"** submitted by **S Sai Ujwala (317126512114), P Suhasini (317126512102), Asha Sumitra (317126512092), P Kavya (318126512l23)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide
Dr. J Bhaskar Rao M.E,Ph.D
Assistant Professor
Department of E.C.E
ANITS

Head of the Department
Dr.V.Rajyalakshmi M.E,Ph.D,MIEEE,MIE,MIETE
Professor & HOD
Department of E.C.E
ANITS

ABSTRACT

Digital images are often corrupted with noise during acquisition and transmission, degrading performance in later tasks such as: image recognition and medical diagnosis. Many denoising algorithms have been proposed to improve the accuracy of these tasks when corrupted images must be used. However, most of these methods are carefully designed only for a certain type of noise or require assumptions about the statistical properties of the corrupting noise. Classical image denoising algorithms based on single noisy images and generic image databases will soon reach their performance limits. Here, a self-tuning version of the newly introduced Fast Adaptive Switching Trimmed Arithmetic Mean Filter, which is a very efficient technique for impulsive noise suppression, is elaborated. Most of the methods presented in the rich literature have numerous parameters, whose proper settings are crucial for efficient noise suppression. Although researchers often provide recommended values for their algorithms' parameters, the actual choice remains in the hands of the user. Our goal is to free the operator from parameter selection dilemma and to propose an algorithm which includes required expert knowledge within itself. The only obligatory inputs of the proposed algorithm (from the user perspective) are the image itself and the size of the operating window

CONTENTS

LIST OF SYMBOLS

- X—input image
- $x_{u,v}$—input image pixel at coordinates (u, v)
- $\hat{X}$ —output image
- $\hat{x}_{u,v}$—output image pixel at (u, v)
- O—original image
- $o_{u,v}$ —original image pixel at (u, v)
- M—original noise map
- $m_{u,v}$—pixel corruption state at (u, v)
- $\hat{M}$ —final noise map estimation
- $\hat{m}_{u,v}$—pixel contamination classification at (u, v)
- W—operating window size
- x_i —i^{th} pixel located in W
- w—size of W
- n—count of pixels present in W
- $d(x_i , x_j)$—distance measured between any two pixels from the window size W
- $c_{u,v}$—cumulative total of least distances that indicate raw impulsiveness of $x_{u,v}$
- Wc—window bearing the raw impulsiveness values that are calculated
- c_{min}—least accumulated distance present in W_c
- $s_{u,v}$—true impulsiveness measure of pixel at (u, v)
- k—number of iteration
- t—threshold value
- t_k—threshold value that is obtains after k^{th} iteration

LIST OF FIGURES

LIST OF TABLES

Table no **Title** **Page no**

LIST OF ABBREVATIONS

PSNR	Peak signal-to-noise
MSE	Mean-square error
SSIM	Structural Similarity Index
VMF	Vector Median Filter
FASTAMF	Fast Adaptive Switching Trimmed Arithmetic Mean Filter
RGB	Red Blue Green

CHAPTER-1.

INTRODUCTION

Noise reduction belongs to the most important image processing operations. The image restoration and enhancement methods are mainly relevant due to the miniaturization of high-resolution, low-cost image sensors, which frequently operate in poor lighting conditions. Quite often, colour images are corrupted by various types of noise, introduced by imperfections in sensors which influence the image formation process, signal instabilities, aging of the storage material, flawed memory locations, transmission errors in noisy channels and electromagnetic interferences. The quality of colour images is severely decreased by impulsive noise distortions, and their removal is one of the most frequently performed low-level processing tasks.

In the project, a modified switching filters is used for the impulsive noise removal in colour images. This method uses the concept of cumulated distances between the processed pixel and its neighbours for noise detection before application of filtering process to modify the pixel value. In filtering process, the sum of distances to only the most similar pixels of the neighbourhood serves as a measure of impulsiveness, was elaborated. The trimmed measure is dependent on the image local structure, an adaptive mechanism was also incorporated. Manual experimental choice of the main filter parameter is avoided by introduction of the Self-tuning mechanism. Performance parameters like Peak Signal to Noise Ratio (PSNR) and Featured Structural Index Measures (FSIM) are used for evaluating the efficiency ug the method.

CHAPTER-2.

IMAGE PROCESSING

2.1INTRODUCTION

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows. In a (8-bit) greyscale image each picture element has an assigned intensity that ranges from 0 to 255. A grey scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey.

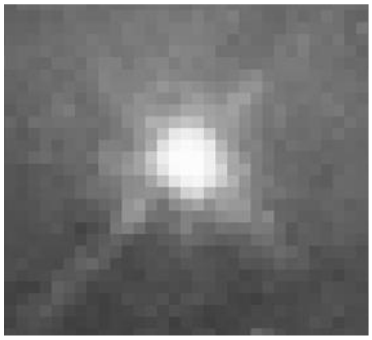


Fig. 2.1: An image — an array or a matrix of pixels arranged in columns and rows.

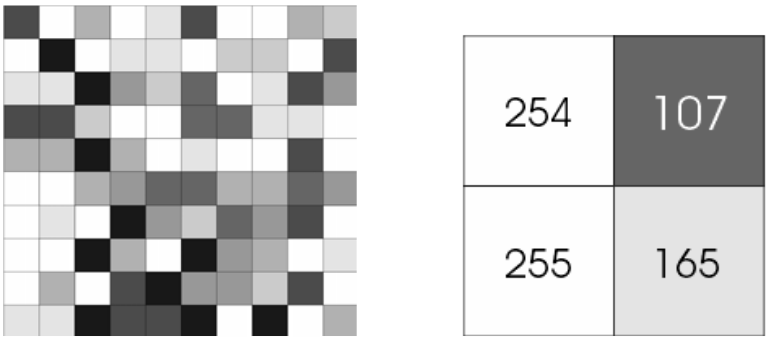


Fig.2.2:Image showing each pixel having a value from 0 (black) to 255 (white)

Some grayscale images have more grayscales, for instance 16 bit = 65536 grayscales. In principle three grayscale images can be combined to form an image with 281,474,976,710,656 grayscales.

There are two general groups of ‘images’: vector graphics (or line art) and bitmaps (pixel-based or ‘images’). Some of the most common file formats are:

GIF — an 8-bit (256 colour), non-destructively compressed bitmap format. Mostly used for web. Has several sub-standards one of which is the animated GIF.

JPEG — a very efficient (i.e. much information per byte) destructively compressed 24 bit (16 million colours) bitmap format. Widely used, especially for web and Internet (bandwidth-limited).

TIFF — the standard 24 bit publication bitmap format. Compresses non-destructively with, for instance, Lempel-Ziv-Welch (LZW) compression.

PS — Postscript, a standard vector format. Has numerous sub-standards and can be difficult to transport across platforms and operating systems.

PSD – a dedicated Photoshop format that keeps all the information in an image including all the layers.

Pictures are the most common and convenient means of conveying or transmitting information. A picture is worth a thousand words. Pictures concisely convey information about positions, sizes and inter relationships between objects. They portray spatial information that we can recognize as objects. Human beings are good at deriving information from such images, because of our innate visual and mental abilities. About 75% of the information received by human is in pictorial form. An image is digitized to convert it to a form which can be stored in a computer's memory or on some form of storage media such as a hard disk or CD-ROM. This digitization procedure can be done by a scanner, or by a video camera connected to a frame grabber board in a computer.

Once the image has been digitized, it can be operated upon by various image processing operations.

Image processing operations can be roughly divided into three major categories, Image Compression, Image Enhancement and Restoration, and Measurement Extraction. It involves reducing the amount of memory needed to store a digital image. Image defects which could be caused by the digitization process or by faults in the imaging set-up (for example, bad lighting) can be corrected using Image Enhancement techniques. Once the image is in good condition, the Measurement Extraction operations can be used to obtain useful information from the image. Some examples of Image Enhancement and Measurement Extraction are given below. The examples shown all operate on 256 grey-scale images. This means that each pixel in the image is stored as a number between 0 to 255, where 0 represents a black pixel, 255 represents a white pixel and values in-between represent shades of grey. These operations can be extended to operate on colour images. The examples below represent only a few of the many techniques available for operating on images. Details about the inner workings of the operations have not been given, but some references to books containing this information are given at the end for the interested reader.

2.2IMAGES AND PICTURES

As we mentioned in the preface, human beings are predominantly visual creatures: we rely heavily on our vision to make sense of the world around us. We not only look at things to identify and classify them, but we can scan for differences, and obtain an overall rough feeling for a scene with a quick glance. Humans have evolved very precise visual skills: we can identify a face in an instant; we can differentiate colors; we can process a large amount of visual information very quickly.

However, the world is in constant motion: stare at something for long enough and it will change in some way. Even a large solid structure, like a building or a mountain, will change its appearance depending on the time of day (day or night); amount of sunlight

(clear or cloudy), or various shadows falling upon it. We are concerned with single images: snapshots, if you like, of a visual scene. Although image processing can deal with changing scenes, we shall not discuss it in any detail in this text. For our purposes, an image is a single picture which represents something. It may be a picture of a person, of people or animals, or of an outdoor scene, or a microphotograph of an electronic component, or the result of medical imaging. Even if the picture is not immediately recognizable, it will not be just a random blur.

Image processing involves changing the nature of an image in order to either

1. Improve its pictorial information for human interpretation,
2. Render it more suitable for autonomous machine perception.

We shall be concerned with digital image processing, which involves using a computer to change the nature of a digital image. It is necessary to realize that these two aspects represent two separate but equally important aspects of image processing. A procedure which satisfies condition, a procedure which makes an image look better may be the very worst procedure for satisfying condition. Humans like their images to be sharp, clear and detailed; machines prefer their images to be simple and uncluttered.

2.3IMAGES AND DIGITAL IMAGES

Suppose we take an image, a photo, say. For the moment, lets make things easy and suppose the photo is black and white (that is, lots of shades of grey), so no colour. We may consider this image as being a two dimensional function, where the function values give the brightness of the image at any given point. We may assume that in such an image brightness values can be any real numbers in the range (black) (white).

A digital image from a photo in that the values are all discrete. Usually they take on only integer values. The brightness values also ranging from 0 (black) to 255 (white). A digital image can be considered as a large array of discrete dots, each of which has a brightness associated with it. These dots are called picture elements, or more simply

pixels. The pixels surrounding a given pixel constitute its neighborhood. A neighborhood can be characterized by its shape in the same way as a matrix: we can speak of a neighborhood,. Except in very special circumstances, neighborhoods have odd numbers of rows and columns; this ensures that the current pixel is in the centre of the neighborhood.

2.4 IMAGE PROCESSING FUNDAMENTALS:

2.4.1.Pixel:

In order for any digital computer processing to be carried out on an image, it must first be stored within the computer in a suitable form that can be manipulated by a computer program. The most practical way of doing this is to divide the image up into a collection of discrete (and usually small) cells, which are known as *pixels*. Most commonly, the image is divided up into a rectangular grid of pixels, so that each pixel is itself a small rectangle. Once this has been done, each pixel is given a pixel value that represents the color of that pixel. It is assumed that the whole pixel is the same color, and so any color variation that did exist within the area of the pixel before the image was discretized is lost. However, if the area of each pixel is very small, then the discrete nature of the image is often not visible to the human eye.

Other pixel shapes and formations can be used, most notably the hexagonal grid, in which each pixel is a small hexagon. This has some advantages in image processing, including the fact that pixel connectivity is less ambiguously defined than with a square grid, but hexagonal grids are not widely used. Part of the reason is that many image capture systems (*e.g.* most CCD cameras and scanners) intrinsically discretize the captured image into a rectangular grid in the first instance.

2.4.2.Pixel Connectivity:

The notation of pixel connectivity describes a relation between two or more pixels. For two pixels to be connected they have to fulfill certain conditions on the pixel brightness and spatial adjacency.

First, in order for two pixels to be considered connected, their pixel values must both be from the same set of values V. For a grayscale image, V might be any range of graylevels, e.g. V={22,23,...40}, for a binary image we simple have V={1}.

To formulate the adjacency criterion for connectivity, we first introduce the notation of neighborhood. For a pixel p with the coordinates (x,y) the set of pixels given by:

$$N_4(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\}$$

is called its 4-neighbors. Its 8-neighbors are defined as

$$N_8(p) = N_4 \cup \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\}$$

From this we can infer the definition for 4- and 8-connectivity:

Two pixels p and q, both having values from a set V are 4-connected if q is from the set $N_4(p)$ and 8-connected if q is from $N_8(p)$.

General connectivity can either be based on 4- or 8-connectivity; for the following discussion we use 4-connectivity.

A pixel p is connected to a pixel q if p is 4-connected to q or if p is 4-connected to a third pixel which itself is connected to q. Or, in other words, two pixels q and p are connected if there is a path from p and q on which each pixel is 4-connected to the next one.

A set of pixels in an image which are all connected to each other is called a connected component. Finding all connected components in an image and marking each of them with a distinctive label is called connected component labeling.

An example of a binary image with two connected components which are based on 4-connectivity can be seen in Figure 1. If the connectivity were based on 8-neighbors, the two connected components would merge into one.

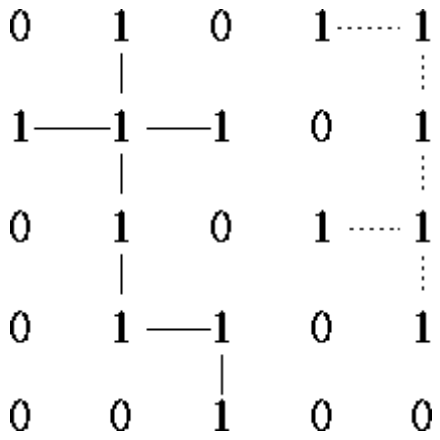


Fig.2.3:Two connected components based on 4-connectivity.

2.4.3.Pixel Values:

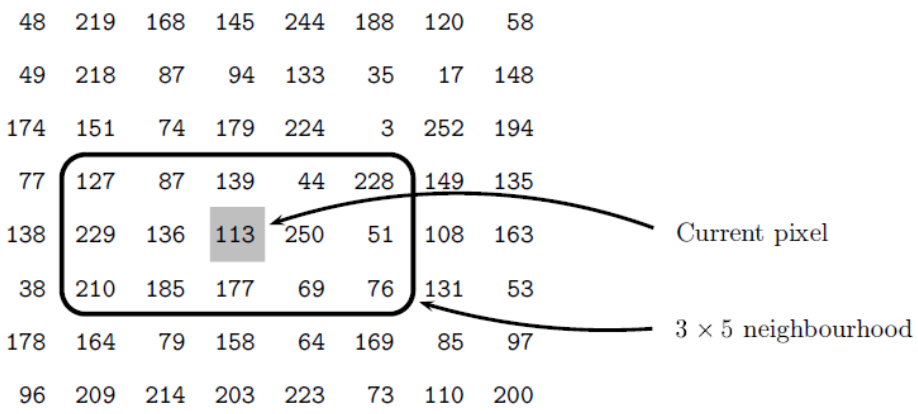
Each of the pixels that represents an image stored inside a computer has a *pixel value* which describes how bright that pixel is, and/or what color it should be. In the simplest case of binary images, the pixel value is a 1-bit number indicating either foreground or background. For a gray scale images, the pixel value is a single number that represents the brightness of the pixel. The most common *pixel format* is the *byte image*, where this number is stored as an 8-bit integer giving a range of possible values from 0 to 255. Typically zero is taken to be black, and 255 is taken to be white. Values in between make up the different shades of gray.

To represent colour images, separate red, green and blue components must be specified for each pixel (assuming an RGB colour space), and so the pixel `value' is actually a vector of three numbers. Often the three different components are stored as three separate `grayscale' images known as *color planes* (one for each of red, green and blue), which have to be recombined when displaying or processing. Multispectral Images can contain even more than three components for each pixel, and by extension these are stored in the same kind of way, as a vector pixel value, or as separate color planes.

The actual grayscale or color component intensities for each pixel may not actually be stored explicitly. Often, all that is stored for each pixel is an index into a colour map in which the actual intensity or colors can be looked up.

Although simple 8-bit integers or vectors of 8-bit integers are the most common sorts of pixel values used, some image formats support different types of value, for instance 32-bit signed integers or floating point values. Such values are extremely useful in image processing as they allow processing to be carried out on the image where the resulting pixel values are not necessarily 8-bit integers. If this approach is used then it is usually necessary to set up a colormap which relates particular ranges of pixel values to particular displayed colors.

2.4.4.Pixels, with a neighborhood:



2.5.RGB

The **RGB color model** is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. RGB uses additive color mixing and is the basic color model used in television or any other medium that projects color with light. It is the basic color model used in computers and for web graphics, but it cannot be used for print production.

The secondary colors of RGB – cyan, magenta, and yellow – are formed by mixing two of the primary colors (**red, green** or **blue**) and excluding the third color. Red and green combine to make yellow, green and blue to make cyan, and blue and red form magenta. The combination of red, green, and blue in full intensity makes white.[figure1.4]

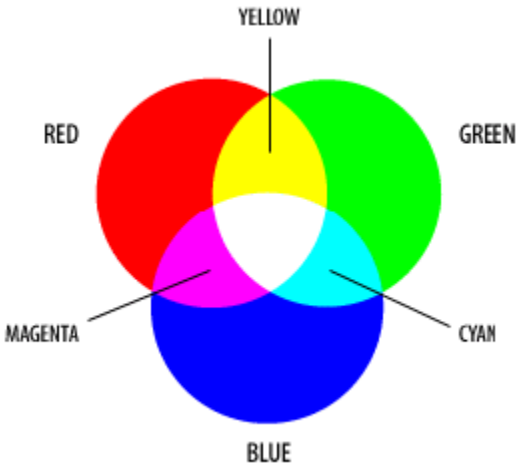


Fig.2.4: The additive model of RGB.

To see how different RGB components combine together, here is a selected repertoire of colors and their respective relative intensities for each of the red, green, and blue components:

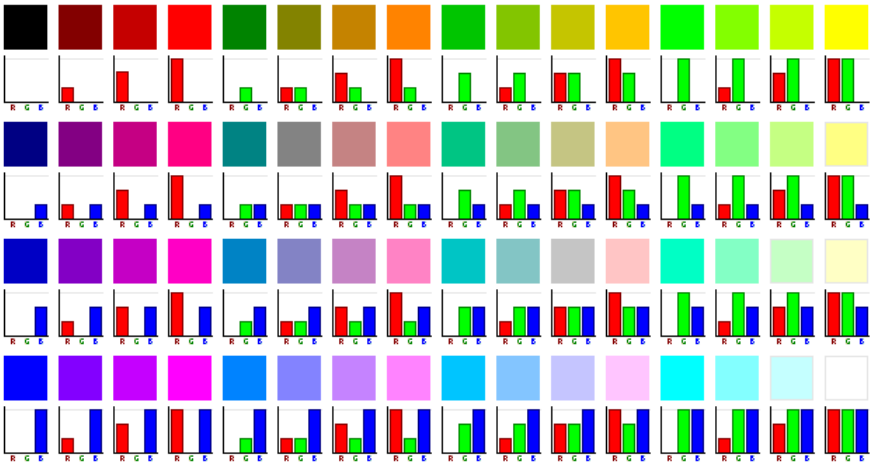


Fig.2.5 Repertoire of colors and Relative intensities for red green and blue components

2.6.APPLICATIONS:

Image processing has an enormous range of applications; almost every area of science and technology can make use of image processing methods. Here is a short list just to give some indication of the range of image processing applications.

1. Medicine
    - Inspection and interpretation of images obtained from X-rays, MRI or CAT scans,
    - Analysis of cell images, of chromosome karyotypes.
2. Agriculture
    - Satellite/aerial views of land, for example to determine how much land is being used for different purposes, or to investigate the suitability of different regions for different crops,
    - inspection of fruit and vegetables distinguishing good and fresh produce from old.
3. Industry
    - Automatic inspection of items on a production line,
    - inspection of paper samples.
4. Law enforcement
    - Fingerprint analysis,
    - sharpening or de-blurring of speed-camera images.

2.7.ASPECTS OF IMAGE PROCESSING:

It is convenient to subdivide different image processing algorithms into broad subclasses. There are different algorithms for different tasks and problems, and often we would like to distinguish the nature of the task at hand.

- **Image enhancement**: This refers to processing an image so that the result is more suitable for a particular application.

Example include:

- sharpening or de-blurring an out of focus image,
- highlighting edges,

- improving image contrast, or brightening an image,
- removing noise.

- **Image restoration**. This may be considered as reversing the damage done to an image by a known cause, for example:
- removing of blur caused by linear motion,
- removal of optical distortions,
- removing periodic interference.

- **Image segmentation**. This involves subdividing an image into constituent parts, or isolating certain aspects of an image:
- circles, or particular shapes in an image,
- In an aerial photograph, identifying cars, trees, buildings, or roads.

These classes are not disjoint; a given algorithm may be used for both image enhancement or for image restoration. However, we should be able to decide what it is that we are trying to do with our image: simply make it look better (enhancement), or removing damage (restoration).

2.8.STAGES OF IMAGE PROCESSING:

We will look in some detail at a particular real-world task, and see how the above classes may be used to describe the various stages in performing this task. The job is to obtain, by an automatic process, the postcodes from envelopes. Here is how this may be accomplished:

- **Acquiring the image**: First we need to produce a digital image from a paper envelope. This can be done using either a CCD camera, or a scanner.
- **Preprocessing:** This is the step taken before the major image processing task. The problem here is to perform some basic tasks in order to render the resulting image more suitable for the job to follow. In this case it may involve enhancing the contrast, removing noise, or identifying regions likely to contain the postcode.

- **Segmentation**: Here is where we actually get the postcode; in other words we extract from the image that part of it which contains just the postcode.
- **Representation and description** These terms refer to extracting the particular features which allow us to differentiate between objects. Here we will be looking for curves, holes and corners which allow us to distinguish the different digits which constitute a postcode.
- **Recognition and interpretation**: This means assigning labels to objects based on their descriptors (from the previous step), and assigning meanings to those labels. So we identify particular digits, and we interpret a string of four digits at the end of the address as the postcode.

CHAPTER-3.

ORIGINAL FASTAMF ALGORITHM

3.1.INTRODUCTION

One of the most popular methods based on reduced ordering, used in many filtering designs, is the Vector Median Filter (VMF). The VMF output is the pixel from W for which the sum of cumulated distances to other samples is minimized. It is always one of the pixels of the filtering window, which is profitable as the filter does not introduce any new colors to the processed image. However, when all pixels of W are affected, for example by additional Gaussian noise, the output is also noisy. Numerous solutions devoted to the elimination of this undesired behavior were introduced, resulting in significantly better filtering performance. To increase the VMF efficiency, weights are assigned to the distances between pixels, which privilege the central pixel of the filtering window, thus diminishing the number of unnecessarily altered pixels. The efficiency of the techniques utilizing various vector ordering schemes is limited due to a common feature— every pixel of the image is processed, regardless whether it is contaminated or not. This results in the inevitable distortion of uncorrupted pixels and degradation of image quality. Therefore, a natural improvement has been made introducing more efficient switching filters, which aim at the restoration of only the polluted pixels, leaving the uncorrupted ones unaltered.

In the majority of the switching techniques, there is a need to determine the dissimilarity between the color pixels. The most intuitive and popular approach is to compute the Euclidean distance in the RGB color space; however, there are many other measures of vector dissimilarity applied in various filtering frameworks. Further improvement resulting in better robustness to the occurrence of outliers was achieved by calculation of only a few smallest distances between a pixel and other samples belonging to the same processing window. Such modification, in which the trimmed cumulative

distance is utilized as a measure of pixel corruption, also facilitates the preservation of the original image edges and tiny details. The decision-making step, differentiating between the distorted and uncorrupted pixels, seems to be more important than the choice of the algorithm used for the replacement of pixels classified as noise. The reason is simple more precise impulse detection process results in less unwanted original pixels alteration.

There are numerous noisy pixel detection schemes proposed in the literature, and among switching filters, several groups of filtering designs can be enumerated. The Sigma Vector Median Filter (SVMF) and Adaptive Vector Median Filter (AVMF) can be regarded as popular representatives of techniques based on reduced ordering statistics. An efficient family of filters based on the peer group framework was proposed in. The idea of this switching strategy can be found in various works. Also significant improvement has been made introducing the Fast Peer Group Filter (FPGF). It was also an inspiration of the recently proposed Fast Averaging Peer Group Filter (FAPGF), which delivers a very good performance for highly contaminated images. Another group of switching filters dedicated to the suppression of the impulsive noise in color images is based on the elements of the quaternion theory. The color pixels, which are generally represented by three channels in the RGB color space, are expressed as quaternions without the real component. In this way, the similarity between pixels is defined in the quaternion form and is used as an alternative for the Euclidean distance, commonly used in the popular filtering designs. The methods based on fuzzy set theory were also elaborated for the impulsive noise removal. These algorithms proved to be very flexible and offer a powerful performance not only for single image applications, but also for the enhancement of video sequences. The filters proposed in this paper belong to the family of switching techniques. The impulse detection step is based on the reduced ordering and computation of trimmed cumulative Euclidean distances. Both Arithmetic Mean Filter (AMF) and VMF will be considered as the filter providing the estimate of the corrupted pixels, to enable a comparison of these two competitive solutions.

3.2.ADAPTIVE SWITCHING ALGORITHM DESIGN:

Most of the filtering techniques determine their output for the pixel located at position (u, v) using n samples belonging to a sliding, operating window W with $x_{u,v}$ at its center. In order to simplify further analysis, the pixels belonging to W will be denoted as x1...; xn, and x1 will be the central pixel of W as shown in Fig.3.1

$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$
$\boldsymbol{x}_5$	$\boldsymbol{x}_{u,v}$ $\boldsymbol{x}_1$	$\boldsymbol{x}_6$
$\boldsymbol{x}_7$	$\boldsymbol{x}_8$	$\boldsymbol{x}_9$

Fig-3.1:Notation of pixels in the filtering window

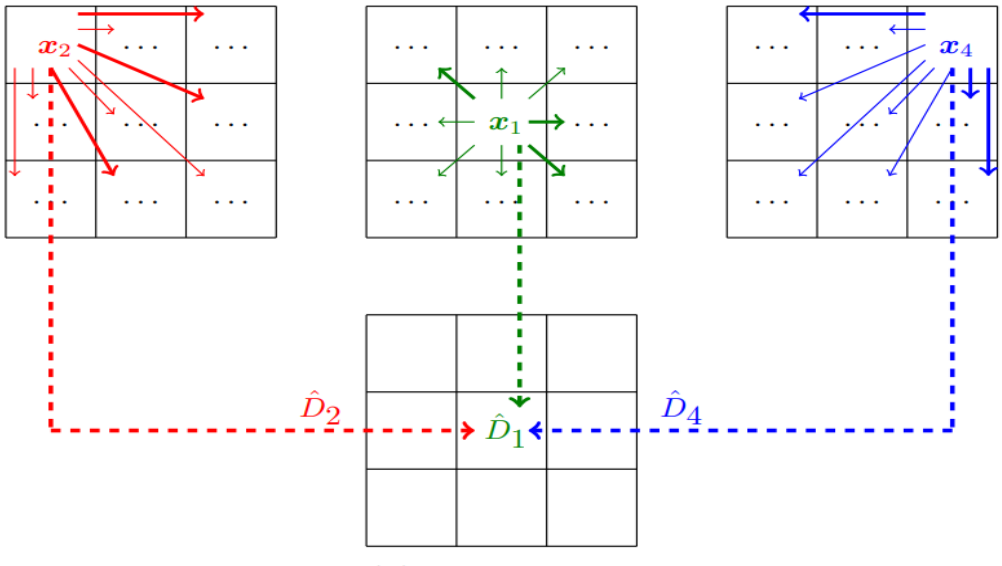


Fig-3.2:AST scheme

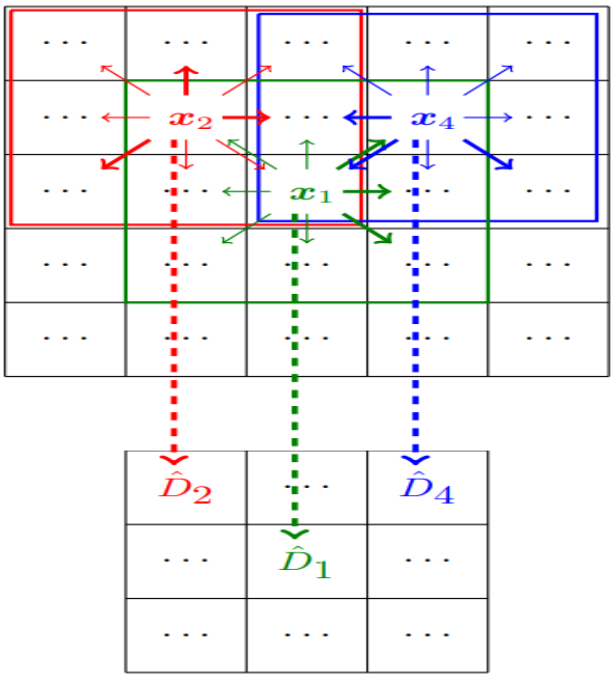


Fig-3.3:FAST scheme

In the AST scheme, the trimmed sums have to be computed for every pixel in W in order to determine the minimum one. Despite the computational efficiency of this solution, which will be shown later, it requires a relatively large number of distance computations for every pixel in the processing window and additionally the minimum value has to be determined.Therefore, a simplified and faster approach has also been taken under consideration. The fast AST (FAST) scheme can be performed in two steps. In the first step, for every image pixel, the computation of the trimmed sum of distances to its neighbors is performed. In the second step, the minimum trimmed sum is computed from all the values assigned to the pixels belonging to W and the decision concerning the central pixel corruption is performed according. The AST and FAST schemes are summarized in Fig.3.2 and Fig.3.3.

In the AST scheme the trimmed sum of distances have to be computed for each pixel of the filtering window W and then the minimum value is calculated. The FAST scheme requires only the calculation of the trimmed cumulative distances for each central pixel of

W (each image pixel), and the minimum value of is taken from the values previously assigned to the pixels of W. Therefore, the FAST scheme is n times faster than AST, as only one trimmed distance measure has to be calculated for each pixel, instead of n values required in the AST scheme.

3.3.FLOW DIAGRAM:

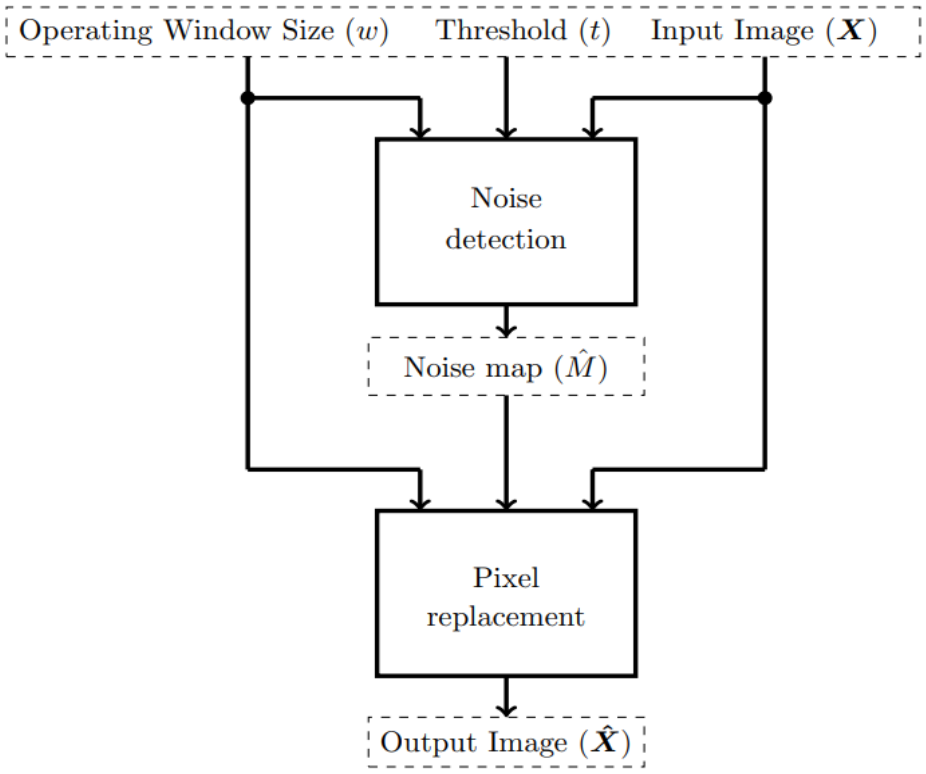


Fig-3.4.Flow diagram of FASTAMF

The FASTAMF algorithm is composed of two main processing phases (Fig. 1): 1. noise detection—the noise map ($\hat{M}$) is estimated upon input image (X), using reduced ordering scheme and two parameters provided by the user: operating window size (w) and threshold (t). 2. pixel replacement—the output image ($\hat{X}$) is obtained using input image , and noise map ($\hat{M}$) provided by noise detection phase. Only pixels classified as noisy are

processed by AMF with operating window size w. The filter operates on every pixel of input image X located at coordinates (u, v), denoted as xu,v, using operational window W containing n = w2 samples. Pixels in W are denoted x1 …, xn, and x1 = xu,v is the center pixel of W.

3.4COMPUTATIONAL COMPLEXITY:

Although the noise suppression efficiency, expressed by quality measures, seems to be the most obvious criterion for algorithm selection, the computational complexity is very often equally important. Therefore, in this Section, a straightforward analysis of computational complexity of the Fast Adaptive Switching Trimmed filter with AMF output— FASTAMF is presented. This filter was chosen as it belongs to the fastest available filtering designs and because of its very satisfying denoising efficiency. It will be compared with the state-of-the-art fast techniques: FPGF, Fast Averaging Peer Group Filter (FAPGF), Fast Fuzzy Noise Reduction Filter (FFNRF), Fast Modified VMF and Vector Median Filter (VMF) which can serve as a reference filter. The analysis of the computational burden will be performed for impulse detection (decision-making step) and output computation step separately.

We assume that color image is encoded with L channels, and the operating window W used by the filter consists of n pixels. The elementary mathematical operations used by an algorithm will be labeled as follows: Addition—ADD, Multiplication—MULT, Division—DIV, Exponentiation— EXP, Extractions of root—SQRT, Comparison—COMP. A detailed analysis of the computational load with commentary is performed for FASTAMF algorithm only. The complexity of the competitive algorithms is summarized in Table 7. The impulse detection step of the FASTAMF requires: – Computation of Euclidean distances. Each distance requires: Calculation of the m smallest distances: COMP; Sum of the m smallest distances: ADD, – One subtraction (ADDS), division (DIV and comparison (COMP). As during the noisy pixel replacement, the algorithm requires the map of noise array M, consisting of values: 1 for uncorrupted pixels, when condition is satisfied and 0 for pixels found to be corrupted, the output computation step

of the AMF requires n x L x MULT, n x L xADDS to acquire a sum of uncorrupted pixels channel values and n xADD, 1xDIV to obtain the final color pixel estimate.

3.5.SUMMARY:

The evaluation of the performance of the described filter family provided in the previous Sections confirmed its high efficiency. The proposed filters are competitive against known fast filtering techniques intended for impulsive noise removal. Especially useful is the Fast Adaptive Switching Trimmed filter with AMF output—FASTAMF, which restores efficiently the corrupted pixels even for strong noise contamination. Its performance is comparable with the recently proposed Fast Averaging Peer Group Filter (FAPGF).

The beneficial feature of the FASTAMF is its low computational complexity, which makes the filter interesting for the real-time color image denoising. The proposed concept of trimmed sum of ordered distances is a very efficient way of determining whether a pixel is corrupted or not. Also the AMF output computed using only pixels recognized as uncorrupted, proved to be a very efficient and computationally inexpensive solution. Additionally, the adaptation mechanism implemented in the AST and FAST decision-making schemes substantially improves the performance of the filters when the image is contaminated by noise of low and medium intensity. For higher noise intensity levels, this mechanism fails to detect the outliers, due to a small number of the uncorrupted samples in the filtering window. The performed experiments confirmed the low computational complexity of the proposed filtering technique and its attractiveness for real-time image processing.

CHAPTER-4.

SELF-TUNING ALGORITHM

4.1 INTRODUCTION

Rapid development of miniaturized high-resolution, low cost image sensors, dedicated to operate in various lighting conditions, makes image enhancement and noise suppression to be very important operations of digital image processing. There are various types of noise which affect acquisition and processing of digital color images. The disturbances may be introduced by:

- Electric signal instabilities
- Physical imperfections in sensors
- Corrupted memory locations
- Transmission errors
- Aging of the storage material
- Natural or artificial electromagnetic interferences.

Therefore, noise suppression is one of the most frequently performed low-level image processing tasks. There are plentiful different techniques tailored for suppression of distinct type of noise, but most of them are vulnerable to occurrence of a impulsive noise, which introduces significant deviations of color image channel values. Therefore, the suppression of the impulsive noise is a critical step of image preprocessing.

Impulsive noise removal techniques are contextual processing schemes which estimate the channels of the processed pixel using information obtained from its neighborhood, represented by a sliding operational window. Many of them are based on a vector-ordering scheme, and use cumulative distances between samples in a window as dissimilarity estimates. Those accumulated distances are then sorted and constitute the basis for further processing in various filtering algorithms.

One of the most basic filtering techniques, utilizing this ordering scheme, is the vector median filter (VMF). The output of VMF is the pixel from operational window for which the sum of distances to other samples from the window is minimized. Although this filter does not introduce any new colors to the processed image, there is no guarantee that the output pixel is itself noise-free, and thus, numerous solutions were developed to solve this problem and improve filtering performance. The main reason, that the efficiency of

vector-ordering schemes is limited, lies in processing of every image pixel, regardless whether it is corrupted or not. Unnecessary processing of noise-free pixels results in inevitable degradation of the image quality. To address this issue, a significant improvement has been made by introduction of more sophisticated switching filters, which focus on the restoration of corrupted pixels only. The switching techniques use various approaches to determine if the processed pixel is corrupted or not. Then, only those classified as noisy are further processed by the output estimation algorithm. This way, not only the quality of output of restored image is preserved, but also a significant reduction of the computational cost is often achieved.

There are numerous techniques of noisy pixel detection to be found in the literature.

Those schemes can be categorized by the following families:

• Schemes based on reduced vector ordering
• Techniques using peer group concept
• Filters utilizing quaterions
• Methods based on fuzzy set theory

The Fast Adaptive Switching Trimmed Arithmetic Mean Filter (FASTAMF), concerned here, has been proposed recently and is a very efficient technique from both noise suppression efficiency and computational cost point of view. The main practical drawback of the algorithm (which is common among alternatives) is the necessity of manual parameter adjustment to image noise contamination severity, to achieve its optimal performance. Therefore, the main goal of the research presented here is the introduction of a self-tuning mechanism, so that manual experimental choice of the main filter parameter (threshold) will no longer be required.

4.1.2 IMPULSIVE NOISE MODELS

Four different noise models are considered. In all of those models, the main parameter is the noise density expressed by the percentage of corrupted pixels in the processed image:

• Channel Together Random Impulse (CTRI)—if a pixel is noisy, all of its RGB channels are corrupted.
• Channel Independent Random Impulse (CIRI)—if a pixel is contaminated, the alteration of every channel is independent.
• Channel Correlated Random Impulse (CIRI)—if a pixel is contaminated, then the corruption of channels is correlated with fixed correlation coefficient.
• Custom Probability Random Impulse (CPRI)—if a pixel is contaminated, there is a fixed set of probabilities that single RGB channels are corrupted or that all channels are corrupted together.

The model does not take into account the corruption of two channels at once. In all above models, contaminated pixel channel is represented by a random value taken from full encoding range: ⟨0, 255⟩ (for 8-bit RGB image coding)

4.2.PERFORMANCE MEASURES:

The performance of the algorithm is being measured by using the following criteria:

$$MSE = \frac{1}{m\,n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

Peak signal-to-noise (PSNR) is that the ratio between the utmost possible power of a picture and therefore the power of corrupting noise that affects the standard of its representation. The mean-square error (MSE) and therefore the peak signal-to-noise (PSNR) are wont to compare compression quality. The MSE represents the cumulative squared error between the compressed and therefore the original image, whereas PSNR represents a measure of the height error. The lower the worth of MSE, the lower the error. The Structural Similarity Index (SSIM) is a perceptual concept that compute image quality degradation, caused by image processing techniques like data compression or data transmission. It is a reference metric that needs two images from the same source (reference image and processed image). Correlation is the process of operating a filter (kernel) over the image and calculating the sum of products at each location. In other way, the first and second values indicates the zero displacement of the filter and unit of displacement.

4.3 SELF TUNING

As it has been shown , there are three inputs for the algorithm: processed image (X), threshold (t), and operational window size (w). As long as w is intuitive parameter to adjust, the proper choice of t may be a difficult one. It was shown that optimal choice of t is dependent on impulsive noise density, which is mostly unknown in real-case scenarios, so the operator is forced to experimental search of adequate value of t. To free the user

from manual adjusting of this parameter, a self-tuning modification is introduced. The main concept of this improvement is to use the estimated noise map $\hat{M}$ (obtained during noise detection phase) to compute the estimated noise density. Combining noise density with proper tuning characteristics like provided in enables to adjust the t value, which can be used to obtain more accurate noise map.

4.3.1 ALGORITHM

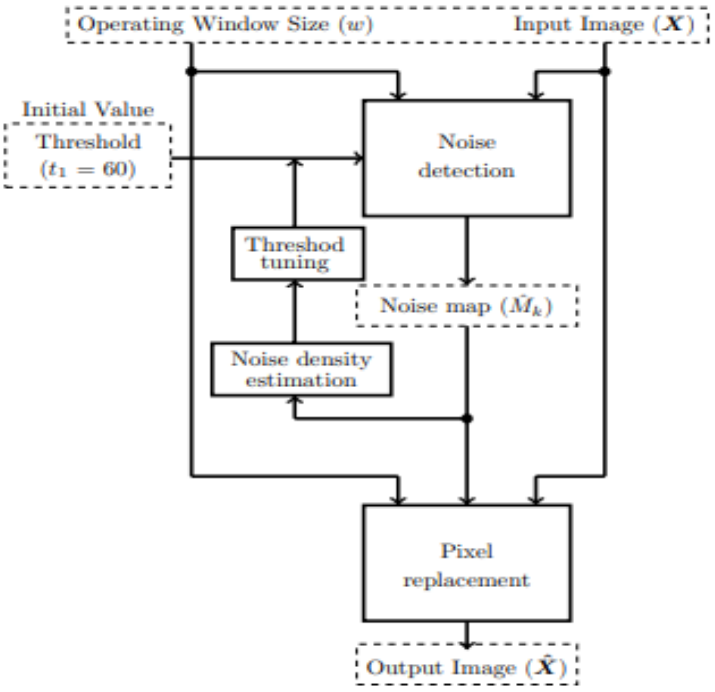


Fig-4.1.Flow diagram of self tuning FASTAMF

Based on the aforementioned idea, the self-tuning modification is introduced. Before execution of the algorithm, the input t is set to initial value t1 = 60 (recommended value for t in FASTAMF using chebyshev distance, obtained experimentally). Then, the noise detection is performed until corrected impulsiveness measure is obtained for every pixel in the input image X.

Next, the recursive procedure of automatic t adjustment is performed by the following steps:

(a) The estimated map of noise for the current iteration $\hat{M}_k$ is obtained by using t_k.

(b) The estimated noise density for the current iteration is evaluated by n_k / θ, where n_k is the number of pixels designated as corrupted (in kth iteration) and θ is the number of pixels in image X.

(c) t_{k+1} value is interpolated (simple linear interpolation between two closest values) using tuning tables.

Steps (a)–(c) are repeated in a loop until desired convergence $|tk+1 - tk| < 1$ is achieved.

Final estimated map of noise, denoted as $\hat{M}$, is then taken as an input to the pixel replacement phase. It is important that only the final step of the entire noise detection phase has to be recursively repeated, so the increase in computational cost is not significant.

Although it might be tempting to design a similar solution for the adaptive tuning of operation window size w, it is pointless to do so due to the following reasons:

• It is very intuitive to choose w value, and using windows larger than 3 × 3 is reasonable for high noise intensities only (noise density> 50%).

• Window size w has a critical impact on computational cost of the algorithm, so its automatic on-the-fly tuning certainly makes its execution time extremely unpredictable.

• Alteration of the w during algorithm's execution requires repetition of the entire noise detection phase, which is very costly form computational point of view. Therefore, such tuning algorithm would be inapplicable for real-time image processing tasks.

• Preliminary tests (omitted in the paper) revealed that w has stronger impact on noise detection phase performance than on pixel replacement phase. Therefore, partial solution, assuming the use of altered on-the-fly w for pixel replacement phase only, resulted in lack of restored image quality improvement.

4.4.FEASIBILITY STUDY:

The feasibility study is carried out to test whether the proposed system is worth being implemented or not.

The feasibility carried out mainly in three sections namely.

• Economic Feasibility-This procedure determines the benefits and savings that are expected from the proposed system. This is also known as cost benefit analysis.

• Technical Feasibility-This study is regarding the system hardware and software. It also tells us about to what extent the proposed system can be implemented using the existing system without increasing the implementation cost.

• Behavioral Feasibility- The proposed system can generate reports with day to day information immediately at the user's request, instead of getting a report which doesn't contain much detail.

4.5.SUMMARY

The overall performance of the self-tuning algorithm is satisfactory by considering the value of SSIM. The SSIM value gives us the similarity between input and output image .If value of SSIM is unity ,then the output is said to be perfect ,our measured value is 0.89 which is nearly equal to 1,stating the efficient performance of the algorithm. The new self-tuning method achieves slightly better or a minimum of not worse overall performance than the original algorithm. Also the computational cost of the self-tuning isn't significantly higher, since it works after the foremost computationally expensive a neighborhood of the algorithm. It would be even more advantageous for processing of video sequences distorted by noise with time-dependent parameters.

CHAPTER-5.

MATLAB

5.1.INTRODUCTION:

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and FORTRAN.

Matlab is a data analysis and visualization tool which has been designed with powerful support for matrices and matrix operations. As well as this, Matlab has excellent graphics capabilities, and its own powerful programming language. One of the reasons that Matlab has become such an important tool is through the use of sets of Matlab programs designed to support a particular task. These sets of programs are called toolboxes, and the particular toolbox of interest to us is the image processing toolbox. Rather than give a description of all of Matlab's capabilities, we shall restrict ourselves to just those aspects concerned with handling of images. We shall introduce functions, commands and techniques as required. A Matlab function is a keyword which accepts various parameters, and produces some sort of output: for example a matrix, a string, a graph. Examples of such functions are sin, imread, imclose. There are many functions in Matlab, and as we shall see, it is very easy (and sometimes necessary) to write our own.

Matlab's standard data type is the matrix_all data are considered to be matrices of some sort. Images, of course, are matrices whose elements are the grey values (or possibly the RGB values) of its pixels. Single values are considered by Matlab to be

matrices, while a string is merely a matrix of characters; being the string's length. In this chapter we will look at the more generic Matlab commands, and discuss images in further chapters.

When you start up Matlab, you have a blank window called the Command Window_ in which you enter commands. Given the vast number of Matlab's functions, and the different parameters they can take, a command line style interface is in fact much more efficient than a complex sequence of pull-down menus.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement financial modeling and analysis. Add-on toolboxes (collections of special-purpose MATLAB functions) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

When working with images in Matlab, there are many things to keep in mind such as loading an image, using the right format, saving the data as different data types, how to display an image, conversion between different image formats.

Image Processing Toolbox provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can perform image enhancement, image deblurring, feature detection, noise reduction, image segmentation, spatial transformations, and image registration. Many functions in the toolbox are multithreaded to take advantage of multicore and multiprocessor computers.

MATLAB and images:

- The help in MATLAB is very good, use it!
- An image in MATLAB is treated as a matrix
- Every pixel is a matrix element

- All the operators in MATLAB defined on
  matrices can be used on images: +, -, *, /, ^, sqrt, sin, cos etc.
- MATLAB can import/export several image formats

– BMP (Microsoft Windows Bitmap)
– GIF (Graphics Interchange Files)
– HDF (Hierarchical Data Format)
– JPEG (Joint Photographic Experts Group)
– PCX (Paintbrush)
– PNG (Portable Network Graphics)
– TIFF (Tagged Image File Format)
– XWD (X Window Dump)
– MATLAB can also load raw-data or other types of image data

- Data types in MATLAB

– Double (64-bit double-precision floating point)
– Single (32-bit single-precision floating point)
– Int32 (32-bit signed integer)
– Int16 (16-bit signed integer)
– Int8 (8-bit signed integer)
– Uint32 (32-bit unsigned integer)
– Uint16 (16-bit unsigned integer)
– Uint8 (8-bit unsigned integer)

Images in MATLAB

Binary images : {0,1}

• Intensity images : [0,1] or uint8, double etc.
• RGB images : m-by-n-by-3
• Indexed images : m-by-3 color map
• Multidimensional images m-by-n-by-p (p is the number of layers)

5.2.IMAGE TYPES IN MATLAB

Outside Matlab images may be of three types i.e. black & white, grey scale and colored. In Matlab, however, there are four types of images. Black & White images are called binary images, containing 1 for white and 0 for black. Grey scale images are called intensity images, containing numbers in the range of 0 to 255 or 0 to 1. Colored images may be represented as RGB Image or Indexed Image.

In RGB Images there exist three indexed images. First image contains all the red portion of the image, second green and third contains the blue portion. So for a 640 x 480 sized image the matrix will be 640 x 480 x 3. An alternate method of colored image representation is Indexed Image. It actually exist of two matrices namely image matrix and map matrix. Each color in the image is given an index number and in image matrix each color is represented as an index number. Map matrix contains the database of which index number belongs to which color.

5.3.IMAGE TYPE CONVERSION

- RGB Image to Intensity Image (rgb2gray)
- RGB Image to Indexed Image (rgb2ind)
- RGB Image to Binary Image (im2bw)
- Indexed Image to RGB Image (ind2rgb)
- Indexed Image to Intensity Image (ind2gray)
- Indexed Image to Binary Image (im2bw)
- Intensity Image to Indexed Image (gray2ind)
- Intensity Image to Binary Image (im2bw)
- Intensity Image to RGB Image (gray2ind, ind2rgb)

5.4.KEY FEATURES

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving

- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces

•Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, FORTRAN, Java, COM, and Microsoft Excel

5.5.TYPICAL USES OF MATLAB:

-Math and computation.

-Algorithm development

-Data acquisition

-Modeling, simulation, and prototyping

-Data analysis, exploration, and visualization

-Scientific and engineering graphics

-Application development, including graphical user interface building

CHAPTER-6.

RESULTS

6.1 ORIGINAL ALGORITHM OUTPUTS

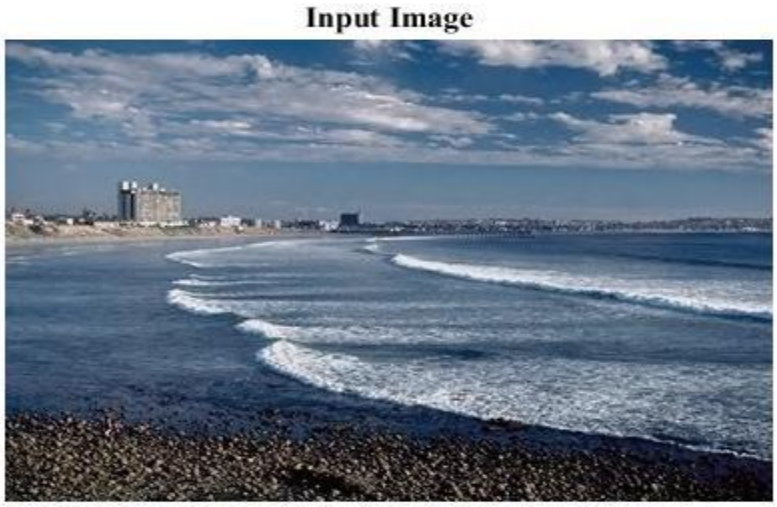


Fig-6.1: Input image

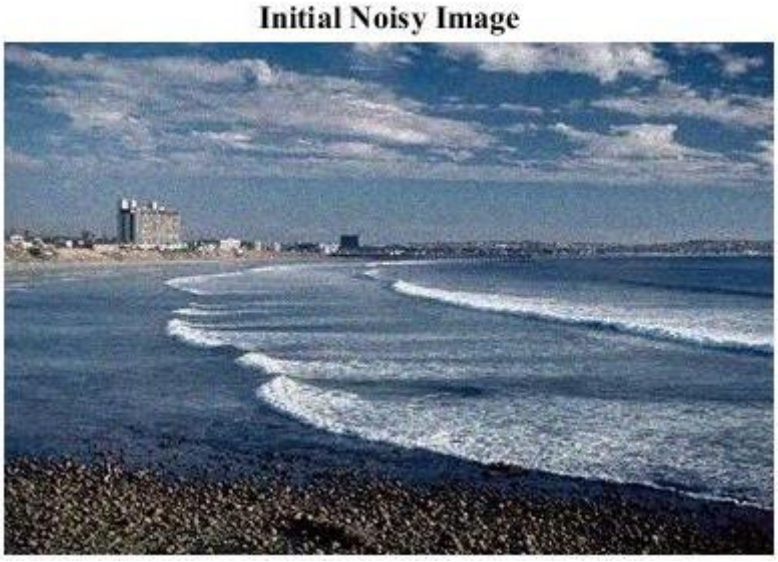


Fig-6.2: Noisy Image (Noise added to the original input image)

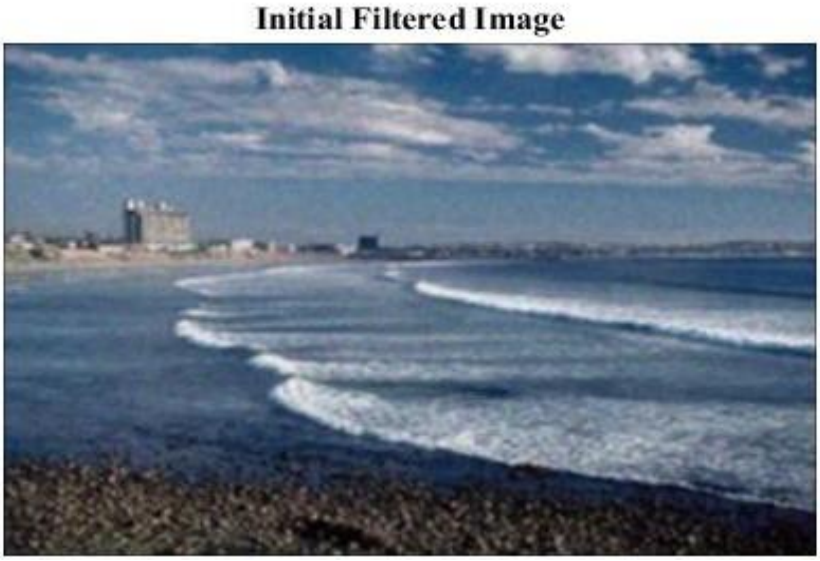


Fig-6.3: output of original algorithm

6.2 SELF TUNING ALGORITHM OUTPUTS

Fig-6.4: Individual Red, Green and Blue channels of the input image

Fig-6.5: Individual Red, Green and Blue channels after the addition of noise

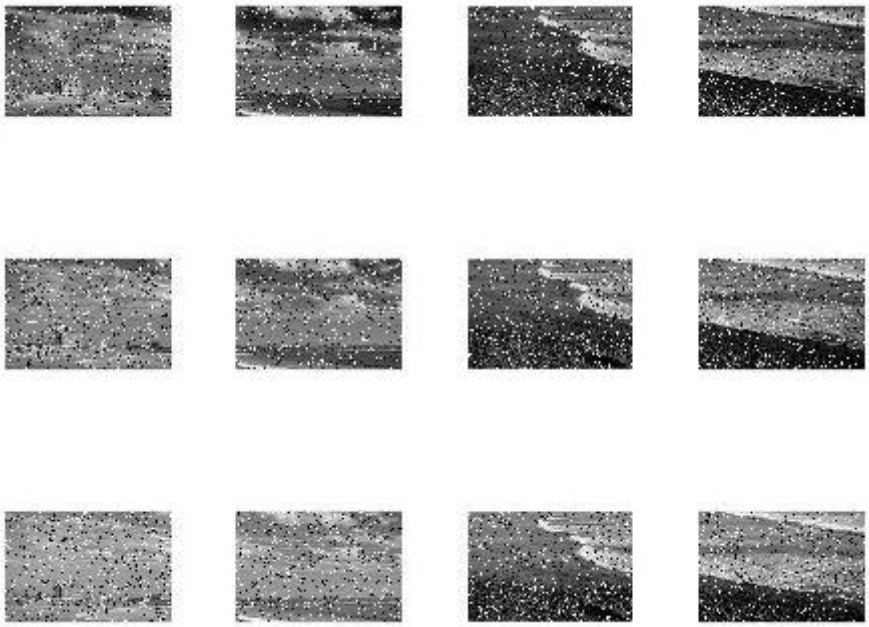


Fig-6.6: Visual representation of the patches

Fig-6.7: Individual Red, Green and Blue channels after the denoising process.

Fig-6.8: Final output image (Concatenated image)

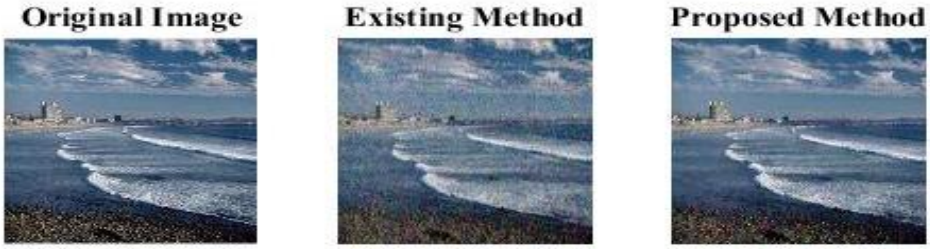


Fig-6.9 Output images of FASTAMF and self tuning FASTAMF

Table-2.1: Noise performance measurement using parameters MSE, PSNR, Correlation and SSIM for different noise densities.

n(%)	MSE	PSNR	correlation	SSIM
1	27.94	33.66	0.9487	0.898
5	27.264	33.77	0.9496	0.8928
6	27.389	33.755	0.9511	0.89837
7	27.621	33.778	0.9508	0.89834
8	27.907	33.6735	0.9527	0.89685
9	27.8225	33.6868	0.948	0.89532
10	27.5806	33.7248	0.9501	0.89528
15	27.2044	33.7844	0.94946	0.89753
20	27.4031	33.7528	0.9496	0.89699
25	27.4813	33.7404	0.9490	0.89647
30	27.6764	33.7097	0.9498	0.8974
35	27.4676	33.7426	0.9513	0.8967
40	27.6096	33.7202	0.9514	0.89846

CONCLUSION

The achieved denoising results are very satisfactory, since the reduction of the number of FASTAMF parameters and a better overall performance have been the main goal of this research. The new self-tuning FASTAMF, achieves slightly better or at least not worse overall performance than the original algorithm, yet it has no parameters which require experimental adjustment. Also the computational cost of the self-tuning is not significantly higher, since it works after the most computationally expensive part of the algorithm (estimation of the pixel impulsiveness).The major virtue of the self-tuning FASTAMF is its adaptability to noise density. As it can be useful for filtering of images contaminated by impulsive noise of unknown density, it might be even more advantageous for processing of video sequences distorted by noise with time-dependent parameters. The initial value of t (for self-tuning mechanism) can be carried out from frame to frame implementations, decreasing the number of potential iterations, required to achieve required convergence. The application of proposed filtering scheme to the video enhancement will be the subject of future work.

REFERENCES

1. Plataniotis, K., Venetsanopoulos, A.: Color Image Processing and Applications. Springer, New York (2000)

2. Lukac, R., Smolka, B., Martin, K., Plataniotis, K., Venetsanopou los, A.: Vector fltering for color imaging. IEEE Signal Process. Magn. 22(1), 74–86 (2005a)

3. Boncelet, C.G.: Image noise models. In: Bovik, A.C. (eds), Handbook of Image and Video Processing, Communications, Networking and Multimedia, Academic Press, Cambridge, pp. 397–410 (2005)

4. Zheng, J., Valavanis, K.P., Gauch, J.M.: Noise removal from color images. J. Intell. Robot. Syst. 7(1), 257–285 (1993)

5. Faraji, H., MacLean, W.J.: CCD noise removal in digital images. IEEE Trans. Image Process. 15(9), 2676–2685 (2006)

6. Liu, C., Szeliski, R., Kang, S., Zitnick, C., Freeman, W.: Automatic estimation and removal of noise from a single image. IEEE Trans. Pattern Anal. Mach. Intell. 30(2), 299–314 (2008)

7. Huang, Y., Ng, M., Wen, Y.: Fast image restoration methods for impulse and Gaussian noise removal. IEEE Signal Proc. Lett. 16(6), 457–460 (2009)

8. Lien, C., Huang, C., Chen, P., Lin, Y.: An efficient denoising architecture for removal of impulse noise in images. IEEE Trans. Comput. 62(4), 631–643 (2013)

9. Yang, S.M., Tai, S.C.: A design framework for hybrid approaches of image noise estimation and its application to noise reduction. J. Vis. Commun. Image Rep. 23(5), 812–826 (2012)

10. Astola, J., Haavisto, P., Neuvo, Y.: Vector median filters. Proc. IEEE 78(4), 678–689 (1990)

11. Celebi, M.E.: Distance measures for reduced ordering-based vector filters. IET Image Process. 3(5), 249–260 (2009). ISSN 1751-9659
12. Celebi, M., Kingravi, H., Lukac, R., Celiker, F.: Cost-effective implementation of order-statistics based vector filters using minimax approximations. J. Opt. Soc. Am. A 26(6), 1518–1524 (2009)
13. Lukac, R., Smolka, B., Plataniotis, K., Venetsanopoulos, A.: Entropy vector median filter. Lect. Notes Comput. Sci. 2652, 1117–1125 (2003)
14. Smolka, B., Malik, K.: Reduced ordering technique of impulsive noise removal in color images. Lect. Notes Comput. Sci. 7786, 296–310 (2013)
15. Vertan, C., Malciu, M., Buzuloiu, V., Popescu, V.: Median filtering techniques for vector valued signals. In: Proceedings of ICIP, volume I, Lausanne, pp. 977–980 (1996)
16. Viero, T., Oistamo, K., Neuvo, Y.: Three-dimensional medianrelated filters for color image sequence filtering. IEEE Trans. Circ. Syst. Video Technol. 4(2), 129–142 (1994)
17. Ponomaryov, V., Gallegos-Funes, F., Rosales-Silva, A.: Realtime color image processing using order statistics filters. J. Math. Image. Vis. 23(3), 315–319 (2005)
18. Masoomzadeh-Fard, A., Venetsanopoulos, A.N.: An efficient vector ranking filter for colour image restoration. Can. Conf. Electr. Comput. Eng. 2, 1025–1028 (1993)
19. Lukac, R.: Adaptive vector median filtering. Pattern Recognition. Lett. 24(12), 1889–1899 (2003)
20. Morillas, S., Gregori, V.: Robustifying vector median filter. Sensors 11(8), 8115 (2011)
Automation and Robotics (MMAR), pp. 855–860 (2017)
21.Malinski, L., Smolka, B.: Training image set (2018a). https:// www.kaggle.com/lmalinski/training-image-set. Accessed 01 Feb 2019
22.Malinski, L., Smolka, B.: Validation image set (2018b). https:// www.kaggle.com/lmalinski/validation-image-set. Accessed 01 Feb 2019 61. Lukac, R.: Adaptive color image filtering based on centerweighted vector directional filters. Multidim. Syst. Signal Process. 15(2), 169–196 (2004)

23. Siegel, S., Castellan, N.: Nonparametric statistics for the behavioral sciences, 2nd edn. McGraw-Hill Inc, New York (1988)

PAPER DETAILS

Paper is submitted to ICTACT Journal entitled as “SELF-TUNING ALGORITHM FOR NOISE SUPPRESSION IN COLOUR IMAGES”.