Adaptive Filters on Android Enabled Devices

Manoj Kumar M
Asst. Professor, Dept. of CSE,
Jyothy Institute of Technology,
VTU, Bangalore, India

Vadiraja A
Asst. Professor, Dept. of ISE,
Jyothy Institute of Technology,
VTU, Bangalore, India

***Abstract:* Android is a mobile operating system which is very popular because of its user friendliness and openness. Android provides a complete set of software for mobile devices: an operating system, middle ware and key mobile applications. Image filtering is useful for many applications such as removing noise, smoothing, sharpening, edge detection and etc.. A filter is defined by a kernel, which is a small array applied to each pixel and its neighbors within an image. Image filtering is a branch of Digital Image Processing (DIP),[1] where the aim is to reduce noise while maintaining the quality of the signal data. Adaptive filters offers advantages over 'fixed' filters. A fixed filter reduces image quality. Adaptive filters are used to overcome from the problem of image non-stationary. Recursive filter have many advantages when they applied to an adaptive filters to image.[1]**

Keywords: Android, Adaptive Filters, Contra Harmonic Mean Filter, Quality Metrics, image processing

I. INTRODUCTION

Recently, a lot of progress has been made in the direction of designing, building and implementation of different applications for Android mobiles like weather forecasting, Business applications, Games, Global positioning system, Image processing applications etc. Image processing on mobile phones is an exciting field with many challenges due to limited hardware and connectivity. The estimation of images is an essential problem in two important areas of image processing: enhancement and data compression. Two approaches have been used in dealing with these non stationary image signals. The first approach is to design an adaptive method of filtering which takes into account the image non stationary and varies their parameters according to these image changes[6]. The second method turns the problem around and transforms the image so that it possesses near stationary statistics before feeding it to a non adaptive filtering process. Adaptive filters offering advantages over 'fixed' filters have been used but suffer some reduction of image quality. Linear filtering does not take into account the local features of the image because it causes for example blurring of the edges. An improvement is changing the filter parameters according to the local statistics, Change the shape and the size of the neighborhood. Suppress the filtering if there are features that we want to preserve in the neighborhood. Adaptive filtering techniques must be implemented to promote accurate solutions and a timely convergence to that solution. Adaptive filters are used to overcome from the problem of image non-stationary.

A. Adaptive Filters

Adaptive filtering techniques must be implemented to promote accurate solutions and a timely convergence to that solution[9]. The basic structure of Adaptive Image Filters involves loading the input image that is to be processed, we applying different adaptive filter method to process image and Quality Parameters are calculated which is shown below, a systematic flow diagram of Adaptive Image Filters is shown in Fig 1.

B. Adaptive Filters on Android platform

The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. Android applications are developed using the Java language.

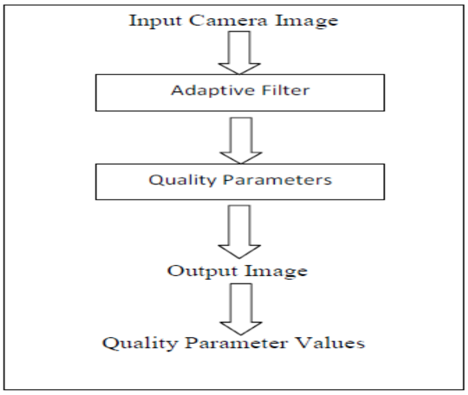

Figure 1: Flowchart of Adaptive image filtering system

Figure 1 depicts a flowchart of an adaptive image filtering system, the input for the system is an image, the input image is processed with adaptive filters, and quality parameters are applied to get quality parameter values.

II. NOISE MODEL

Any undesired information that contaminates an image. Noise models is a random variable with a probability density function (PDF) that describes its shape and distribution The actual distribution of noise in a specific image is the histogram of the noise. Noise can be modeled with Gaussian (“normal”), uniform, salt-and-pepper (“impulse”), or Rayleigh distribution[12].

A. Salt and Pepper Noise

Salt and pepper noise can also called impulse noise, shot noise or spike noise typically caused by malfunctioning pixel element in camera sensors, faulty memory locations, or timing errors in digitization process. It represents itself as randomly occurring white and black pixels. This type of noise involves usage of median filter and mean filter.

$$H_{salt\&paper} = \begin{bmatrix} A & \text{for g = a ("pepper")} \\ B & \text{for g = b ("salt")} \end{bmatrix}$$

There are only 2 possible values, a and b, and the probability of each is typically less than 0.2 – with numbers greater than this the noise will swamp out the image.

B. Gaussian Noise

This is independent at each pixel and independent of signal intensity caused by SyQuest noise. SyQuest noise is an electronic noise generated by thermal agitation of charge carriers inside electrical conductors occur from electronic noise in image acquisition system. Most problematic with poor lighting conditions or vary high temperatures.

$$H_g = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(g-m)^2/2\sigma^2}$$

g = gray level
m = mean (average)
σ = standard deviation
σ^2 = variance

C. Uniform Noise

This is caused by quantizing the pixel of the sensed image to a number of discrete levels. It can be used to generate any other type of noise distribution, and is often used to degrade images for the evaluation of image restoration also since provides the most unbiased or neutral noise model.

$$H_{\text{Uniform}} = \begin{cases} \frac{1}{b-a} & \text{For a<=g<=b} \\ 0 & \text{Otherwise} \end{cases}$$

Where
Mean = (a + b)/2
Variance = $(a – b)^2/12$

III. ADAPTIVE FILTER ALGORITHM

The Adaptive Filter methods implemented in Android platform are described in detail with their respective algorithms as follows,

A. Adaptive-neighborhood noise subtraction (ANNS)

Used for removing additive signal-independent noise. Estimates the noise value in the seed pixel with an adaptive neighborhood and subtracts it to obtain an estimate of the original.

$$\tilde{f}(m,n) = \mu_g(m,n) + \left[1 - \sqrt{\frac{\sigma_\eta^2(m,n)}{\sigma_f^2(m,n) + \sigma_\eta^2(m,n)}}\right]\left[g(m,n) - \mu_g(m,n)\right]$$

Algorithm:
Input: Noise image, size MxN, g(m,n).
Output: Filtered image, size MxN, F(m,n).

Step1: consider the noise image g(m,n) as input image.
Step2: calculate the mean (μ_n(m,n)) and variance (σ_n^2(m,n)) of the input image g(m,n).
Step3: calculate local mean (μ_g(m,n)) and local variance (σ_g^2(m,n)) from rectangular window around the pixel and slide the widow pixel by pixel.
Step4: apply the formula,

B. Alpha Trimmed Mean Filter

In this algorithm we are going to calculate the alpha mean for entire Image. The entire image is replaced with alpha mean calculated. This algorithm works on different noise images such as Gaussian noise, speckle noise, salt and pepper noise etc.

$$\text{f (m,n)} = \frac{1}{(N-2P)} * \sum_{i=P}^{N-P} A(i)$$

Algorithm:
Input: Noise image, size MxN, g(m,n)
Output: Filtered image, size MxN, F(m,n)

Step1: Consider the noise image g(m,n).
Step 2: Select the pixel range, the range will be (P,N-P) where P is the clipping factor.
Step 3: Calculate the alpha mean,
Step 4: Replace the value of each pixel by the alpha mean calculated.

C. Order Statistic Filter

One of the most important families of nonlinear image filters is based on order statistic. The widely used median filter is the best known filter of this family. The adaptation of order statistics filters is a very important task. It is well known that image characteristics (e.g. local statistics) change from one image region to the other. Noise characteristics usually vary with time.

Algorithm:
Input: Noise image, size MxN, g(m,n)
Output: Filtered image, size MxN, F(m,n)
Step1: Take noise Image g(m,n)
Step 2: Select the neighborhood pixel around the pixel.
Step 3: Apply the formula,

$$\text{F (m, n)} = \sum_{i=0}^{n-1} \text{Wi Xi}$$

Wi is the weight and Xi is the grey level at each pixel.

D. Contra Harmonic Mean Filter

Contra harmonic mean filter: This filter computes the contra harmonic mean of the pixels intensity values. The contra harmonic filter reduces to the mean filter for Q = 0, and to the harmonic mean filter for Q = -1.

$$A(out) = \sum_{(i,j)\in M} A(x+i, y+j)^{(P+1)} \Big/ \sum_{(i,j)\in M} A(x+i, y+j)^{P}$$

Algorithm:

Input: A(i,j), noise image, size MxN.
Output: A(out), Filtered image, size MxN.

Step 1: Consider the noise image A(i,j).
Step 2: Apply sliding window to the image and then apply the formula,
Step 3: Replace the value of each pixel by the Contra Harmonic mean calculated.
Step 4: Apply the quality parameters.

IV. QUALITY METRICS

Image Processing Filters are used to reduce the noise or speckles in an image. These are mainly used to suppress either high frequency components in the image i.e. smoothing the image or low frequency components i.e. enhancing or detecting edges in an image. The metrics used for image filtering make use of full reference quality metrics and no-reference quality metrics. The metrics used for image filtering make use of full reference quality metrics and no reference quality metrics.

V. DESIGN ANALYSIS

Design Analysis is the process or art of defining the architecture, components, modules, interfaces, and data for a design to satisfy specified requirements. The modules of the system are as follows. Loading of Image: This is the first phase in an image processing system. This phase deals with loading the acquired image by the camera which is the input to the image processing system. The images are pushed to sdcard or obb so that it can be processed further. Selecting the particular noise image: This is the second phase in the system, where particular noise image from the sdcard or obb is selected to process. The noise may be Gaussian noise, uniform noise or Salt and Pepper noise. Applying Adaptive Filter Method: In this method the Adaptive Filter Methods are applied to the degraded image, actually all seven methods are applied on the three different noisy images in order to remove the noise[12]. Calculating Quality metrics: In order to analyze which algorithm works better on which type of noise we need to calculate the Quality Parameters value.

A. Sequence diagram for Adaptive Filters on Android platform

Sequence diagram is also called as timing diagram. It shows series of operations that takes place between user and application. Sequence diagram shows live process, vertical lines, and horizontal lines shows the message carried between the user and application.

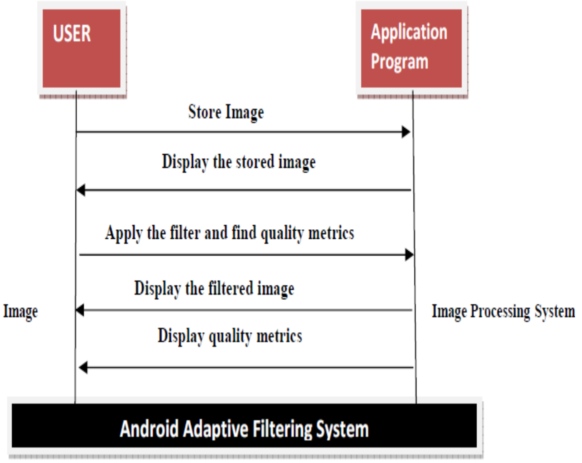

Figure 2 Adaptive image filtering system

Figure 2 shows the interaction between the user and Adaptive image filtering system to get quality metrics.

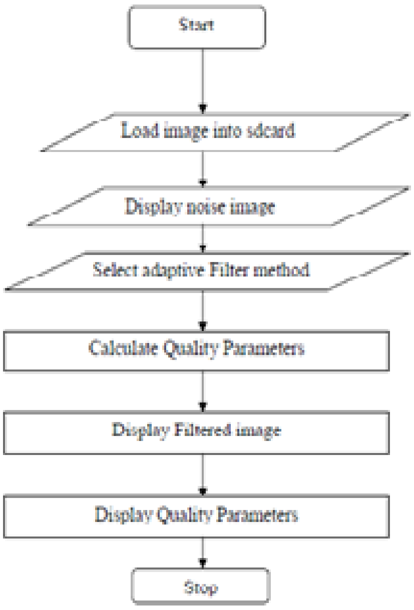
Start
Load image into sdcard
Display noise image
Select adaptive Filter method
Calculate Quality Parameters
Display Filtered image
Display Quality Parameters
Stop

Figure 3 Flowchart of adaptive image filter

Figure 3 shows detailed flowchart of an adaptive image filtering system, the input for the system is an image, select the adaptive filter method, the method calculates quality parameters, filtered image and quality parameters are displayed.

VI. RESULTS AND ANALYSIS

Quality Parameters

(i) Gaussian Noise: Filtered Images of Adaptive Filters

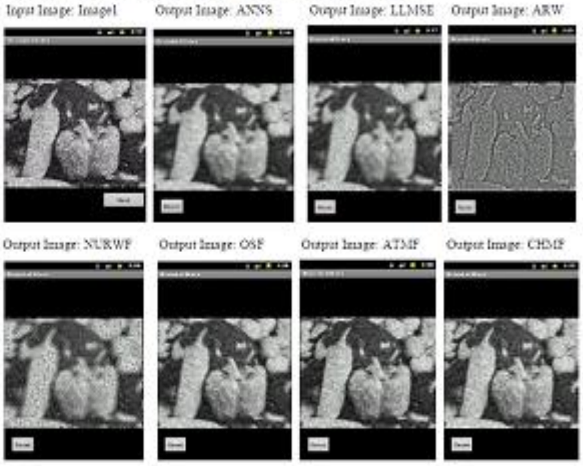

Figure 4 Gaussian noise images

(ii) Uniform Noise: Filtered Images of Adaptive Filters

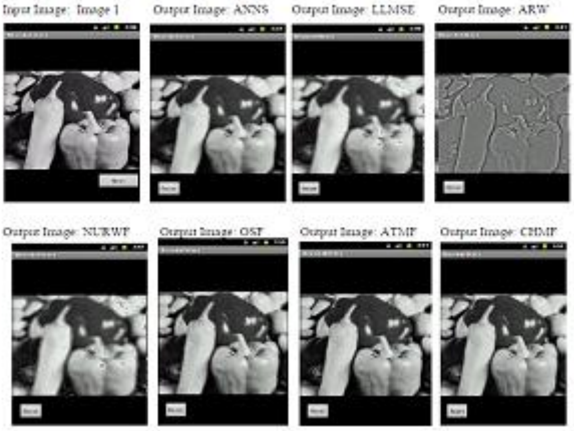

Figure 5 Uniform noise images

(iii) Salt and Pepper Noise: Filtered Images of Adaptive Filters

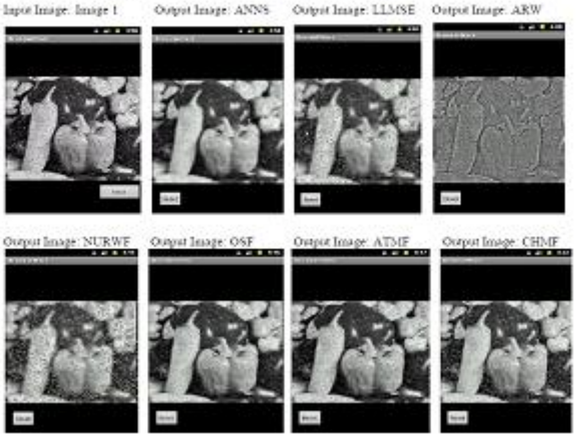

Figure 6 Salt and Pepper noise images

Image 1: Gaussian Noise	MSE	PSNR	MAE
Adaptive Neighborhood Noise Subtraction	1187.221	92.333	25.052
The Local LMMSE Filter	2728.306	72.998	40.799
The Adaptive Rectangular Window Filter Window	3640.164	66.540	51.942
Noise Updating Repeated Wiener Filter	4014.735	64.106	49.530
Order Statistic Filter	1369.416	88.866	28.482
Alpha Trimmed Mean Filter	1335.716	89.439	26.807
Contra Harmonic Mean Filter	**726.693**	**103.453**	**20.406**

Table1: Quality Parameters for adaptive Filter algorithm

Image 1: Uniform Noise	MSE	PSNR	MAE
Adaptive Neighborhood Noise Subtraction	578.677	108.884	14.366
The Local LMMSE Filter	1095.677	93.999	23.856
The Adaptive Rectangular Window Filter	5021.037	59.137	62.363
Noise Updating Repeated Wiener Filter	1638.147	84.741	27.836
Order Statistic Filter	458.789	114.039	14.184
Alpha Trimmed Mean Filter	732.650	103.264	19.541
Contra Harmonic Mean Filter	**279.715**	**125.403**	**10.831**

Table2: Quality Parameters for adaptive Filter algorithm

Image 1: Salt and Pepper Noise	MSE	PSNR	MAE
Adaptive Neighborhood Noise Subtraction	1289.864	90.429	23.152
The Local LMMSE Filter	3054.142	70.401	37.850
The Adaptive Rectangular Window Filter	4425.979	62.041	58.521
Noise Updating Repeated Wiener Filter	4662.988	60.660	46.198
Order Statistic Filter	1674.169	84.240	27.460
Alpha Trimmed Mean Filter	1496.987	86.815	27.367
Contra Harmonic Mean Filter	**992.209**	**96.283**	**21.254**

Table3: Quality Parameters for Adaptive Filter algorithm

Figure 4, Figure 5 and Figure 6 depicts examples of Gaussian, Uniform and Salt pepper filtering techniques respectively. Table 1, Table 2 and Table 3 shows quality parameters obtained by Gaussian, Uniform and Salt pepper techniques respectively.

CONCLUSION AND FUTURE WORK

Adaptive filter is the one of the best method to remove the noise present in the image. Adaptive filters works excellently on any kind of noise finally resulting in filtered output image from the noise image. In this project we used three different noises such as “Gaussian noise”, “Uniform noise” and “Salt and Pepper Noise” and applied different Adaptive filters in Android platform. From the result obtained we finally conclude that, for “Gaussian noise”, and “Uniform noise”, “Contra Harmonic Mean Filter” filters gives better results based on the quality metrics values. More samples can be considered and can be tested with android adaptive filtering system.

REFERENCES

[1]. Pitas and Anastasios N. evetsanopolos, “Order Statistics in Digital Image Processing”, Ioannis Processing of the IEEE, vol. 80, no. 12, December 1992.

[2]. Ji-Nan Lin and Rolf Unbehauen, ”2-D Adaptive CPWQ Filter for Image Enhancement”, *IEEE*, University Erlangen-Niirnberg 1993.

[3]. K. N. Plataniotis A. N. Venetsanopoulos, “Adaptive Filters for Multichannel Signal Processing”, IEEE, School of Computer Science, 1998.

[4]. Mohiy M. Hddhoud David W. Thomas, “The Two-Dimensional Adaptive LMS [TDLMS] Algorithm”, IEEE Transaction on Circuits and systems vol. 35, no. *5,* May 1988.

[5]. Michael Ropert and Danielle Pele, “Synthesis of Adaptive Weighted Order Statistic Filters with Gradient Algorithms and Application to Image processing”, IEEE, 1994

[6]. Stephen J. Viscera, Annear S. Dawood, and John A. Williams, “FPGA Based Real-time Adaptive Filtering for Space applications”, IEEE, 2002.

[7]. Kevin Erler and Ed Jernigan, “Adaptive Image Restoration Using Recursive Image Filters” IEEE Transactions on Signal Processing, vol. 42, no. *1,* July 1994.

[8]. C Smith, “The Two-Dimensional LMS algorithm and its applications to sub-band Filtering” D R Campbell University of Paisley, UK, July 1995.

[9]. Xiaoyin Xu, Eric L. Miller, Ongbin Chen, and Mansoor Sarhadi, “Adaptive Two-Pass Rank Order Filter to Remove Impulse Noise in Highly Corrupted Images”, IEEE, February 2004.

[10]. Edmond Nezry, Armand Lopes, Francis Yakam-Simen “Prior Scene Knowledge for the Bayesian Restoration of Mono- and Multi-Channel SAR Images”, IEEE, 1997.

[11]. M. Gabrani, c. Kotropodos, I. Pitas, Cellular LMS L-Filters for noise suppression in still Images and image Sequences”, 1994 IEEE.

[12]. Mahdi Shaneh, and Arash Golibagh Mahyari, “Image Enhancement using α-Trimmed Mean ε-Filters”, World Academy of Science, Engineering and Technology 59, 2011.

[13]. Mohammed Charif- Chefchaouni and Dan Schonfeld, “Morphological Representation of Order-Statistics Filters”, , IEEE Transactions on Image Processing vol 4. no 6, 1995