

A PAPER ON AUTOMATIC FABRICS FAULT PROCESSING USING IMAGE PROCESSING TECHNIQUE IN MATLAB

R.Thilepa

Department of EEE

Adhiyamaan Educational & Research Institute, Hosur-Tamil nadu 635 109

Email:thilepasudhakar@yahoo.co.in

M.Thanikachalam

Department of Civil Engineering,

Velammal Engineering College, Chennai-600 066

Email:thanikachalam84@yahoo.com

ABSTRACT

The main objective of this paper is to elaborate how defective fabric parts can be processed using Matlab with image processing techniques. In developing countries like India especially in Tamilnadu, Tirupur the Knitwear capital of the country in three decades yields a major income for the country. The city also employs either directly or indirectly more than 3 lakhs of people and earns almost an income of 12, 000 crores per annum for the country in past three decades [2]. To upgrade this process the fabrics when processed in textiles the fault present on the fabrics can be identified using Matlab with Image processing techniques. This image processing technique is done using Matlab 7.3 and for the taken image, Noise Filtering, Histogram and Thresholding techniques are applied for the image and the output is obtained in this paper. This research thus implements a textile defect detector with system vision methodology in image processing.

Keywords: Image processing, Matlab 7.3, Gray image, Histogram, Thresholding.

INTRODUCTION

The city Tirupur in Tamilnadu consists of at present 3000 sewing units, 450 knitting units and 100s of dyeing units at present for fabrics processing. These units involve both manual and automation for all processes. The annual income for the past year 2008 stands at Rs.8000 crore. This city is also named as Manchester of South India [12]. Since the city yields a major income on textiles and fabrics, it is given more importance to this field here. In this paper a fabric faulty part is taken for analysis from textiles. For this process we have used Matlab 7.3 in image processing software. Here we can analyze all faults present on fabrics such as hole, scratch, dirt spot, fly, crack point, color bleeding etc... automatically. Several authors have considered defect detection on textile materials. Kang et al. [16], [17] analyzed fabric samples from the images obtained from transmission and reflection of light to determine its interlacing pattern. In addition if the faults are to be identified manually the time consumption is more comparatively less and the percentage of fault identification is deduced to a lower rate. Also it has been observed that the price of textile fabric is reduced by 45% to 65% due to defects[13] . Machine vision automated

inspection system for textile defects has been in the research industry for longtime [14], [15]. Recognition of patterns independent of position, size, brightness and orientation in the visual field has been the goal of much recent work. Hence the efficiency is also reduced in this process. To overcome all these drawbacks this automation process can be implemented.

Indian textile and clothing exports to US is lowest in Nov 2009:

Month	From world(\$ bn)	From India (\$ mn)
January	6.9	433
February	6	411
March	5.9	450
April	5.8	403
May	6	379
June	6.6	351
July	7.7	378
August	7.5	364
September	7.7	379
October	7.6	355
November	6.2	325

The above table [11] shows the details of income to India by the textile field. Hence if this automaton process of fault identification process using Matlab is implemented we can deduce the fault in a major way.

Growth Rate of Manufacturing and Textile sector in India:

Year	Growth rate of manufacturing (%)	Growth rate of Textiles (%)
2003-04	7.4	-1.1
2004-05	9.1	9.3
2005-06	9.1	8.2
2006-07	12.5	10.9
2007-08	9.0	5.8
2008-09	3.2	-1.2
2009-10(April-November)	7.7	5.8

The above tables [11] are the reports produced by FICCI (Fiber Consumption of Indian Society).There by automation using Matlab in image processing we can increase the manufacturing %ge in an enormous way efficiently. In our paper a defective fabric image is taken and noise filtering is done at the initial stage. Then the output filtered image is converted into gray scale image. Now the gray scale image is processed for histogram process and finally Thresholding is done using Matlab 7.3 software.

An image may be defined as a two –dimensional function, $f(x, y)$, where x and y are spatial (plane) coordinates, and the amplitude of f at any pair coordinates (x, y) is called the intensity or gray level of the image at that point. [1].When x , y , and the amplitude values of f are

all finite, discrete quantities; we call the image as a digital image. The Image Processing technique is a collection of functions that extend the capability of the MATLAB in numeric computing environment.

The toolbox supports a wide range of image processing operations, including: open image file, add noise of a given type (e.g. salt & pepper, Gaussian, Speckles...) to intensity image, 2-D median filtering and adaptive filtering, Image analysis and enhancement, Color Image decomposition into RGB Channels, Image histogram, Image segmentation, image Multithresholding,, image movie , signal plotting and etc... , Many of the toolbox functions are MATLAB M-files, Model files and a series of MATLAB statements that implement specialized image processing algorithms. Then Histogram is done for the image and finally thresholding is done for the same image using this image processing toolbox.

METHODOLOGY OF THE PAPER:

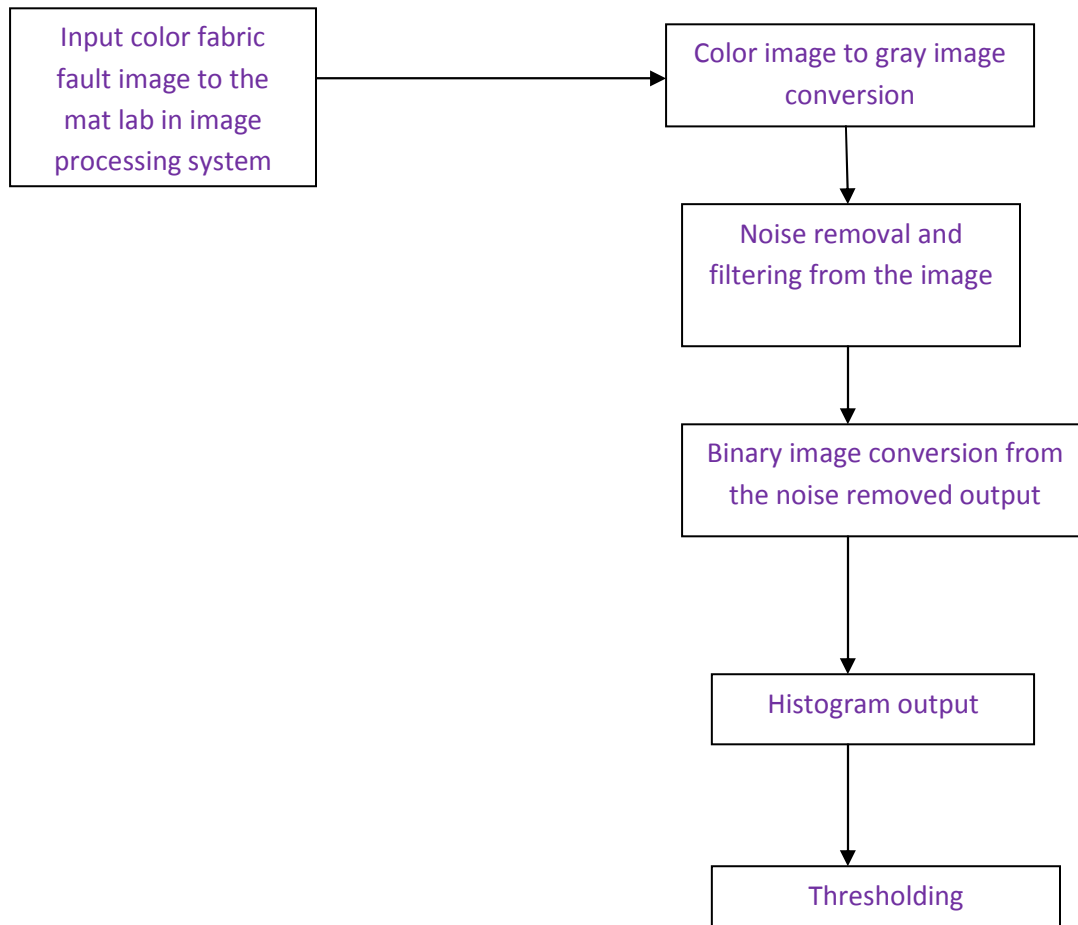


Fig 1: Methodology of the paper

IMAGE PROCESS IMPLEMENTATION USING MATLAB:

- Make a new directory in which you Work for this course [3].
- Now download the image ‘text.tif’ from

http://www.syseng.anu.edu.au/~luke/cvcourse_files/images/text.tiff

- Save this image in a new directory.
- Open Mat lab from the Start Programs menu.
- Change to a new directory (use cd).
- Types edit to open a new script file, and we can start work.

We can get help on any Matlab function by typing `help <function>` or to get help Specifically on the Image Processing Toolbox type `help images`, typing just `help` will show you all the available help topics.

Load the image into `im_rgb` with
`im_rgb = imread('text.tiff');`

Convert to type double
`im_rgb = double(im_rgb);`

Convert to grey-scale
`im_grey = (im_rgb(:,:,1)+im_rgb(:,:,2)+im_rgb(:,:,3))/3;`

Or you can use `rgb2gray` which gives a slightly different result since it converts to a Luminance rather than an intensity image.

Convert from [0,255] to [0,1]

`im_grey = im_grey/255;`

to view an image use either `imshow(im)` or `imagesc(im)`, note the difference [4]. We can use `imagesc` and then set the axis for an image with `axis image;`

Axis labelling can be turned on and off by

`axis on;`
`axis off;`

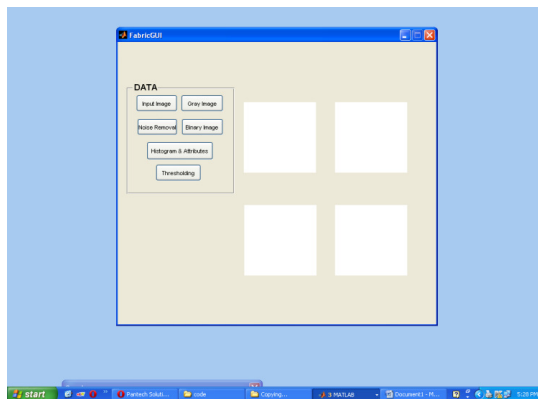


Fig 1: Matlab Window ready for taking input Fabric color Image

The defective fabric part is now processed using Matlab by the following steps. In the first step it is converted into gray image as shown in the above figure then noise removal is done. This data is used to obtain the Histogram output after thresholding is done. This is the procedure followed for fault fabric processing using Matlab in this paper. The below program is executed in Matlab 2008 and the defective fabric image is obtained. Similarly the Histogram image is obtained from the image. The below is the Matlab program which is used for fabric fault analyzing and for obtaining histogram output.

```
function varargout = FabricGUI(varargin)
% FABRICGUI M-file for FabricGUI.fig
%   FABRICGUI, by itself, creates a new FABRICGUI or raises the existing
%   singleton*.
%
%   H = FABRICGUI returns the handle to a new FABRICGUI or the handle to
%   the existing singleton*.
%
%   FABRICGUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in FABRICGUI.M with the given input arguments.
%
%   FABRICGUI('Property','Value',...) creates a new FABRICGUI or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before FabricGUI_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to FabricGUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help FabricGUI

% Last Modified by GUIDE v2.5 21-Aug-2010 17:11:49
% Begin initialization code –
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @FabricGUI_OpeningFcn, ...
                  'gui_OutputFcn', @FabricGUI_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```

end
% End initialization code
% --- Executes just before FabricGUI is made visible.
function FabricGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to FabricGUI (see VARARGIN)
% Choose default command line output for FabricGUI
handles.output = hObject;
inputimage= ones(256,256);
axes(handles.axes1);
imshow(inputimage);
axes(handles.axes2);
imshow(inputimage);
axes(handles.axes3);
imshow(inputimage);
axes(handles.axes5);
imshow(inputimage);

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FabricGUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = FabricGUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on button press in Defected_Image.
function Defected_Image_Callback(hObject, eventdata, handles)
% hObject    handle to Defected_Image (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename, pathname] = uigetfile('*.bmp;*.jpg;*.pgm;*.png', 'Pick an image'); %%uigetfile is a
function to get images from
                %%%from database

if isequal(filename,0) || isequal(pathname,0)

    warndlg('Image is not selected'); %%%to show warning dialog use warndlg

else

```

```
a=imread(filename); %%%read an input image
inputimage = imresize(a,[256 256]); %%%resize that image in to 256 rows and 256 columns

axes(handles.axes1); %%% axes which is shown in GUI file ,,in that showing image
imshow(inputimage); %%%imshow function to show image

end
```

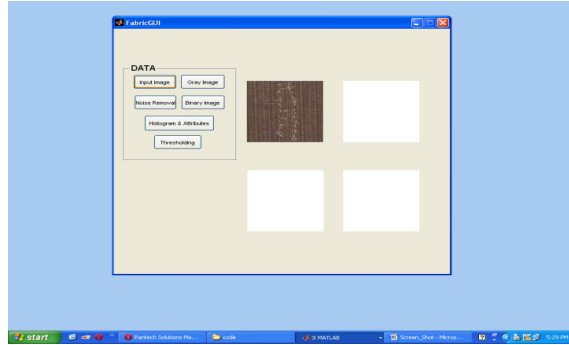


Fig 2: Color fabric Image is applied to Matlab

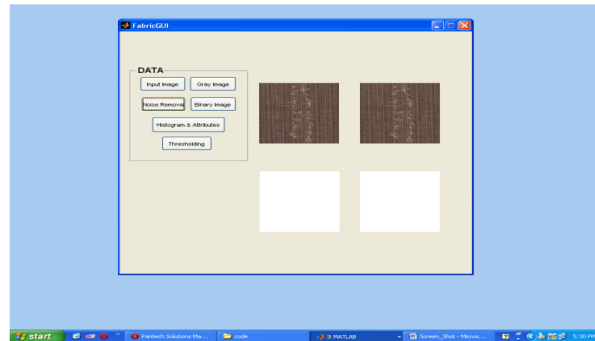


Fig 3: Color Image is ready for Conversion to Gray Image

A grayscale image usually requires that each pixel be stored as a value between 0 - 255 (byte), where the value represents the shade of gray of the pixel [5]. The number of gray levels typically is an integer power of 2 ($L=2^K$). The image formats are .Gif, .tif, .Jpeg, and .bmp. In this paper we used color images (RGB images) and separated into their components (Red, Green, and Blue).

COLOR IMAGE TO GRAY SCALE CONVERSION:

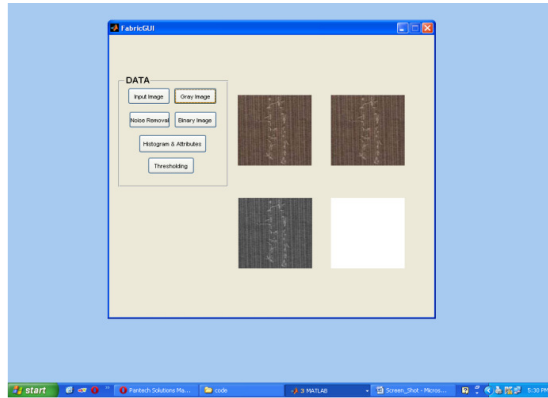


Fig 4: Fabric color Image is converted into Gray scale Image

Explore noise reduction in images using linear and nonlinear filtering techniques is applied. WIENER2 Performs 2-D adaptive noise-removal filtering [6]. WIENER2 low pass filters an intensity image that has been degraded by constant power additive noise. WIENER2 (Adaptive filtering) uses a pixel-wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel.

GRAY SCALE TO BINARY IMAGE WITH NOISE FILTERING:

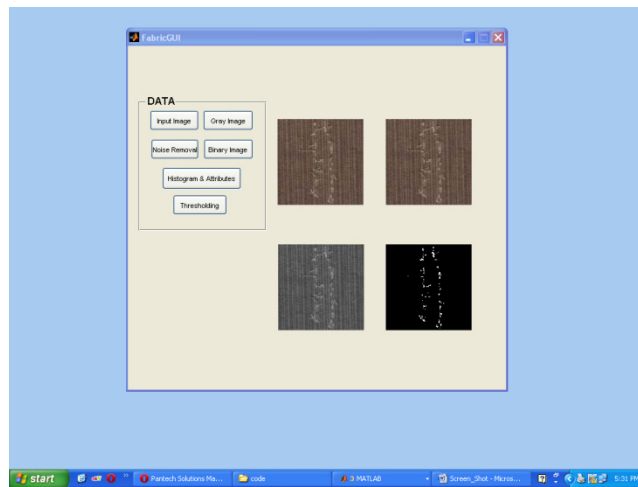


Fig 5: Gray Image Conversion with Noise removal to obtain Histogram

The below program gives the details of noise filtering in the images from the gray image. The adaptive filtering method is used to filter the noise present in the image.


```
>> im = imread('saturn.tif');  
>> mean_filter = ones(9,9); % change the size as you like...  
>> mean_filter = mean_filter/numel(mean_filter);  
>> filtered_im = imfilter(im,mean_filter);  
>> figure(1);clf;colormap gray  
>> subplot(121)  
>> imshow(im)  
>> title('Original image')  
>> subplot(122)  
>> imshow(filtered_im)  
>> title('Smoothed image')
```

THRESHOLDING:

In thresholding, the color-image or gray-scale image is reduced to a binary image. Thresholding is a process of converting a grayscale input image to a bi-level image by using an optimal threshold [7]. The purpose of thresholding is to extract those pixels from some image which represent an object (either text or other line image data such as graphs, maps). Though the information is binary the pixels represent a range of intensities. For a thresholding algorithm to be really effective, it should preserve logical and semantic content. There are two types of thresholding algorithms:

- (i) Global thresholding algorithms
- (ii) Local or adaptive thresholding algorithms

In global thresholding, a single threshold for all the image pixels is used. When the pixel values of the components and that of background are fairly consistent in their respective values over the entire image, global thresholding could be used. In adaptive thresholding, different threshold values for different local areas are used. In our paper adaptive thresholding is used.

HISTOGRAM OUTPUT OF THE FAULTY FABRIC:

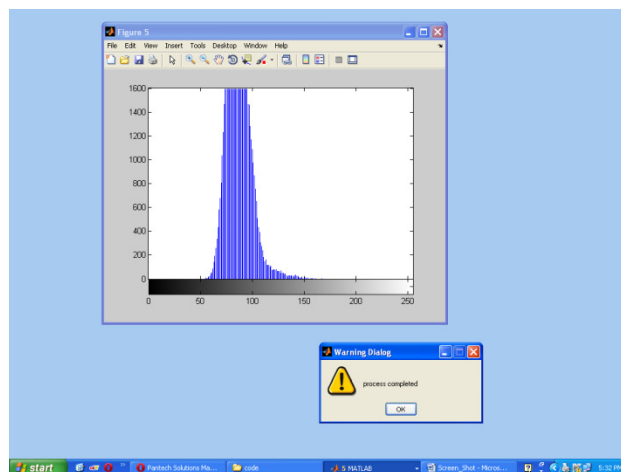


Fig 6: Histogram output for the faulty Fabric part

To perform histogram graph for an image, pixels of each color in the image, and producing a running sum of the count. Then by simply scaling the output, we can perform histogram process. The histogram's x -axis reflects the range of values in Y . The histogram's y -axis shows the number of elements that fall within the groups; therefore, the y -axis ranges from 0 to the greatest number of elements deposited in any bin. The x -range of the leftmost and rightmost bins extends to include the entire data range in the case when the user-specified range does not cover the data range. Histograms bins are created as patch objects and always plotted with a face color that maps to the first color in the current color map (by default, blue) and with black edges[9].

```

        int pixelCount[256];
        //initialize all indexes to zeros
        for(int x= 1; x < imageWidth ; x++)
        { for(int y = 1 ; y < imageHeight ; y++)
            {
                pixelCount[getpixel(x,y)]++;
            }
        }
        int sum = 0;
        for(int i= 0; i < 256 ; i++)
        {
            pixelCount[i]+=sum;
            Sum = pixelCount[i];
        }

        img = imread (imageName);
        imGray = rgb2gray(img);
        myHist = imhist(imGray);
        eqImage = imhisteq(imGray);
        figure, imshow(imGray);
        figure, imshow(eqImage);
    
```

- The first line reads the image file into a 3 dimensional array (x, y, color). about the details (like the width being a multiple of four).
- The second line converts an RGB image into a gray image. This is not necessary if the image is already a gray level image.
- The third line returns the image histogram in the array myHist. This can be used to analyze the image histogram and possibly perform histogram specification.
- The fourth line performs histogram equalization.
- The final two lines display the un-equalized image and the equalized image.

- In order to view the histogram of the image, we will supply two output arguments to the function for histogram equalization. The output arguments are the transformation function and the equalized image respectively. This is given as follows :

```
[eqImage, transfFunc] = imhisteq(imGray);  
figure, plot(transfFunc);  
figure, imshow(eqImage);
```

The second and third lines display the transformation function and the histogram equalized image respectively[10].

FINAL OUTPUT OF THE PROCESS AFTER THRESHOLDING:

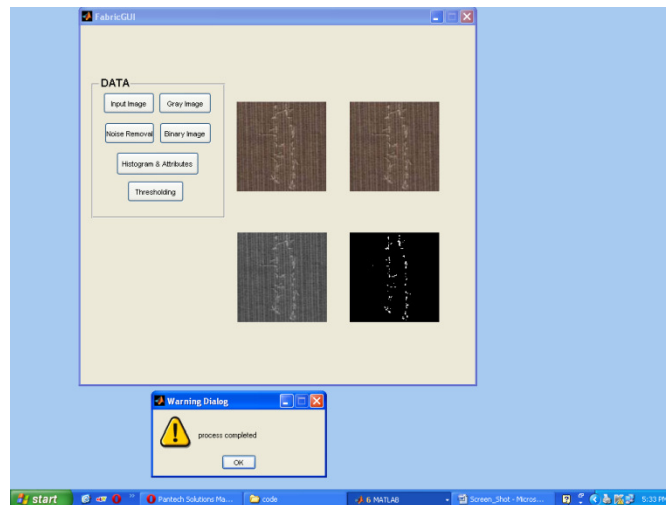


Fig 7: Final output after Histogram and Thresholding

CONCLUSION:

It is easy to identify faults on fabric images and process by using this method. Thus the MATLAB Implementation is done for fault identification such as hole, scratch, fading and other faults on fabrics can be identified and processed. Thus the overall efficiency is 85% by using this process compared to other methods of fault identification. Hence a different approach wherein various thresholding algorithms can be successively applied on the input image can yield better results. The manual textile quality control usually goes over the human eye inspection. Notoriously, human visual inspection is tedious, tiring and fatiguing task, involving observation, attention and experience to detect correctly the fault occurrence. The accuracy of human visual inspection declines with dull jobs and endless routines. Sometimes slow, expensive and erratic inspection is the result. Therefore, the automatic visual inspection protects both: the man and the quality. Here, it has been demonstrated that Textile Defect Recognition System is capable of detecting fabrics' defects with more accuracy and efficiency Thereby applying Matlab 7.3 version to the color faulty fabrics it is processed and finally the Histogram is obtained for the same image

and thresholding is done to obtain the intensity of the image [8]. In future this can be extended to any number of fault identifications on fabrics and can be processed.

FUTURE SCOPE OF THE WORK:

In future this work may be extended such that the output is given to neural network and the Microcontrollers of any type can be utilized and programmed such that it can detect the faulty fabric part. If the microcontroller is connected with motors of any type then it will be operated under normal fabric condition and can stop the motor if there is any fault on fabrics.

REFERENCES:

1. R. C. Gonzalez, R. E. Woods, S. L. Eddins, “*Digital Image Processing using MATLAB*”, ISBN 81-297-0515-X, 2005, pp. 76-104,142-166
2. [http:// en.wikipedia.org/wiki/Tirupur](http://en.wikipedia.org/wiki/Tirupur)
3. Kenneth R. Castelman, Digital image processing, Tsinghua Univ Press, 2003.
4. I.Pitas, Digital Image Processing Algorithm and Applications. *John Wiley & Sons, Inc.2002.*
5. ENGN 4528 Computer Vision, Semester 1, 2003 Lab 1: *Introduction to Image Processing in Matlab & Binary Image Analysis*
6. newsgroups.derkeiler.com > Archive > Comp > comp.soft-sys.matlab > 2007-09.
7. Thresholding (image processing) - Wikipedia, the free encyclopedia.mht
8. Thresholding A Pixel-Level Image Processing Methodology Preprocessing Technique for an OCR System for the Brahmi Script Devi Ancient Asia.mht
9. Histogram plot - MATLAB.mht
10. Color histogram - Wikipedia, the free encyclopedia.mht
11. Textile Views - Textile news, Apparel news, fabric, yarns, Tirupur exporters , Tirupur Ready made garments , apparel news, Tirupur yarn market , CMT cost.mht
12. <http://www.scribd.com/doc/7015798/Tirupur-case-study>
13. Ahmed Ridwanul Islam, Farjana Zebin Eishita, Jesmine Ara Bubby, “Implementation of a RealTime Automated Fabric Defect Detection System” 2007.
14. B. G. Batchelor and P. F. Whelan, “Selected Papers on Industrial Machine Vision . Systems,” SPIE Milestone Series, 1994.
15. T. S. Newman and A. K. Jain, “A Survey of Automated Visual Inspection,” Computer Vision and Image Understanding, vol. 61, 1995, pp. 231–262.
16. Kang T.J. et al. “Automatic Recognition of Fabric Weave Patterns by Digital Image Analysis”, Textile Res. J. 69(2), 77-83 (1999)
17. Kang T.J. et al. “Automatic Structure Analysis and Objective Evaluation of Woven Fabric Using Analysis”, Textile Res. J. 69(2), 77-83 (1999)