Appendix –A

The calculation of the MSSIM value is based on the following MATLAB 7.0 code applied through the MATLAB 7.0[67]

function [mssim, ssim_map] = ssim_index(img1, img2, K, window, L)

% Input: (1) img1: the first image being compared
%        (2) img2: the second image being compared
%        (3) K: constants in the SSIM index formula default value: K = [0.01 0.03]
%        (4) window: local window for statistics default window is Gaussian given by
%        window = fspecial('gaussian', 11, 1.5);
%        (5) L: dynamic range of the images. default: L = 255
%
% Output: (1) mssim: the mean SSIM index value between 2 images.
%        If one of the images being compared is regarded as
%        perfect quality, then mssim can be considered as the
%        quality measure of the other image.
%        If img1 = img2, then mssim = 1.
%        (2) ssim_map: the SSIM index map of the test image. The map
%        has a smaller size than the input images. The actual size:
%        size(img1) - size(window) + 1.
%
% Default Usage:
% Given 2 test images img1 and img2, whose dynamic range is 0-255
%
% [mssim ssim_map] = ssim_index(img1, img2);
%
% Advanced Usage:
% User defined parameters. For example
%
% K = [0.05 0.05];
% window = ones(8);
% L = 100;
% [mssim ssim_map] = ssim_index(img1, img2, K, window, L);
%
%See the results:
%
% mssim                          %Gives the mssim value
% imshow(max(0, ssim_map).^4)    %Shows the SSIM index map
%
%==============================================================================

if (nargin < 2 | nargin > 5)
    ssim_index = -Inf;
    ssim_map = -Inf;
    return;
end
if (size(img1) ~= size(img2))
    ssim_index = -Inf;
    ssim_map = -Inf;
    return;
end

[M N] = size(img1);

if (nargin == 2)
    if ((M < 11) | (N < 11))
        ssim_index = -Inf;
        ssim_map = -Inf;
        return
    end
    window = fspecial('gaussian', 11, 1.5);
    K(1) = 0.01;
    K(2) = 0.03;
    L = 255;
end
if (nargin == 3)
  if ((M < 11) | (N < 11))
    ssim_index = -Inf;
    ssim_map = -Inf;
    return
  end
  window = fspecial('gaussian', 11, 1.5);
  L = 255;
  if (length(K) == 2)
    if (K(1) < 0 | K(2) < 0)
      ssim_index = -Inf;
      ssim_map = -Inf;
      return;
    end
  else
    ssim_index = -Inf;
    ssim_map = -Inf;
    return;
  end
end

if (nargin == 4)
  [H W] = size(window);
  if ((H*W) < 4 | (H > M) | (W > N))
    ssim_index = -Inf;
    ssim_map = -Inf;
    return
  end
  L = 255;
  if (length(K) == 2)
    if (K(1) < 0 | K(2) < 0)
      ssim_index = -Inf;
      return;
    end
  end
end
ssim_map = -Inf;
return;
end
else
ssim_index = -Inf;
ssim_map = -Inf;
return;
end
end
if (nargin == 5)
[H W] = size(window);
if ((H*W) < 4 | (H > M) | (W > N))
ssim_index = -Inf;
ssim_map = -Inf;
return
end
if (length(K) == 2)
if (K(1) < 0 | K(2) < 0)
ssim_index = -Inf;
ssim_map = -Inf;
return;
end
else
ssim_index = -Inf;
ssim_map = -Inf;
return;
end
end
C1 = (K(1)*L)^2;
C2 = (K(2)*L)^2;
window = window/sum(sum(window));
img1 = double(img1);
img2 = double(img2);

mu1 = filter2(window, img1, 'valid');
mu2 = filter2(window, img2, 'valid');
mu1_sq = mu1.*mu1;
mu2_sq = mu2.*mu2;
mu1_mu2 = mu1.*mu2;
sigma1_sq = filter2(window, img1.*img1, 'valid') - mu1_sq;
sigma2_sq = filter2(window, img2.*img2, 'valid') - mu2_sq;
sigma12 = filter2(window, img1.*img2, 'valid') - mu1_mu2;

if (C1 > 0 & C2 > 0)
    ssim_map = ((2*mu1_mu2 + C1).*(2*sigma12 + C2))./((mu1_sq + mu2_sq + C1).*sigma1_sq + sigma2_sq + C2));
else
    numerator1 = 2*mu1_mu2 + C1;
    numerator2 = 2*sigma12 + C2;
    denominator1 = mu1_sq + mu2_sq + C1;
    denominator2 = sigma1_sq + sigma2_sq + C2;
    ssim_map = ones(size(mu1));
    index = (denominator1.*denominator2 > 0);
    ssim_map(index) = (numerator1(index).*numerator2(index))./(denominator1(index).*denominator2(index));
    index = (denominator1 ~= 0) & (denominator2 == 0);
    ssim_map(index) = numerator1(index)./denominator1(index);
end

mssim = mean2(ssim_map);

return
Appendix –B

1) Standard Figure chart on all different Parameters consider in the experiments

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Parameters</th>
<th>Highest Value</th>
<th>Lowest Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MSSIM</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>Standard Deviation</td>
<td>0.0</td>
<td>200.0</td>
</tr>
<tr>
<td>3</td>
<td>SC</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>NK</td>
<td>0.0</td>
<td>2.0</td>
</tr>
<tr>
<td>5</td>
<td>PSNR</td>
<td>0.0</td>
<td>100</td>
</tr>
</tbody>
</table>

3) Commands used in the experiments

1) Histogram  To pass the intensity to the image
2) IMABSDIFF  To find difference between images
3) IMREAD()   To Read the image
4) IMRESIZE   To resize the unstructured image
5) IMVIEW     To View the image
6) RGB2gray   To Convert color to gray image
Appendix –C

I) Paper Published in ITA Pune 2009

“IMPROVE THE QUALITY OF NOISY FINGERPRINT IMAGE THROUGH CONTRAST IMAGE ENHANCEMENT METHOD”

Abstract

A technology for recognizing fingerprints for high security purposes is proving as regards as reliable but efficient recognition is depending on the quality of input fingerprint images. Fingerprint images are varies in quality of the fingerprint impressions, a quality image is nothing but the pure and clear impressions of the finger, because of the certain reasons it may varies in quality such as cut of impressions or un identical impressions called as noisy images. The automatic fingerprint identification system (AFIS) is a system used for identification and the performance of an automatic fingerprint identification system (AFIS) is totally based on the quality of input fingerprint images, it is essential to introduce a fingerprint enhancement module in the AFIS system. In this paper, we introduce a one of the special domain fingerprint enhancement method that is contrast image enhancement which decomposes the input fingerprint image into a set of filtered images. From the filtered images, the orientation field is estimated and a quality mask which distinguishes the recoverable and unrecoverable corrupted regions in the input image is generated. Using the estimated orientation field, the input fingerprint image is adaptively enhanced in the recoverable regions.

Through the Experimental results it seen that enhancement Methods improves the performance of the fingerprint Images makes it more robust with respect to the quality of input.

References:

II) Paper published in international conference ICSCI Hydrabad 2007

III) Paper Published in international journal IJCSNS Journal South Coria in Sept Issue 2007(Vol. 7 No. 9 pp. 225-230)

“NOISY FINGERPRINT IMAGE ENHANCEMENT TECHNIQUE FOR IMAGE ANALYSIS: A SPECIAL DOMAIN APPROACH”

Abstract

Fingerprint images vary in quality. In order to ensure that the performance of an automatic fingerprint identification system (AFIS) will be robust with respect to the quality of input fingerprint images, it is essential to incorporate a fingerprint enhancement module in the AFIS system. In this paper, we introduce a special domain fingerprint enhancement methods which decomposes the input fingerprint image into a set of filtered images. From the filtered images, the orientation field is estimated and a quality mask which distinguishes the recoverable and unrecoverable corrupted regions in the input image is generated. Using the estimated orientation field, the input fingerprint image is adaptively enhanced in the recoverable regions.

A technology for recognizing fingerprints for security purposes is proving as regards as reliable but efficient recognition is depending on the quality of input fingerprint image. Recognition of the fingerprint becomes a complex computer problem while dealing with noisy and low quality images. In this Paper work we are focusing the special domain biometric System of noisy and low quality images, which will be beneficial for recognition system. Experimental results show that our enhancement Methods improves the performance of the fingerprint Images makes it more robust with respect to the quality of input.

References: