Scientific Journal of Impact Factor (SJIF): 5.71
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 03, March -2018

MOTION DEBLURR

[1]D.SONY, [2] M.KRISHNA

[1]*Assistant Professor, Geethanjali College of Engineering & Technology*
[2]*Geethanjali College of Engineering & Technology*

ABSTRACT- *In this project we deblurr an a image i.e improved the quality of the image that has been blurred due to the motion. Here due to motion blurr the original image gets smoothed as the intensity of a single pixel gets distributed among its neibhouring pixels as result the image seems to be unclear. DEBLURRING refers to improving and sharpening of the image quality. The occurance of blurr may be due to the motion of either the object or camera. As the blurred image appears to be smoothened (as a result of averaging)in order to remove the occurred blurr we need to use an high pass filter which sharpens the image by attenuating the low frequencies of the Fourier transform relatively unchanged.*
Debluring algorithm has been implemented by using Weiner filter. The blurred image is deconvolved using the Weiner filter to get back the original from the blurred image.
Most of the times many events occur which we cannot get them back to happen such events we try store them as memories. Events here can be anything like first birthday celebration of a child, when any one gets an award. The easy way to have such unforgettable events as memories is to store in form of photographs.
Some of those events occur in cases like where there is no proper lightening sufficient brightness and even it occurs a case when the object is moving if such an event as to be stored as photograph, the photos taken in such conditions will not appear to be clear such photos are to be processed using digital image processing techniques like contrast improvement, brightness correction.
In this project we are trying to improve the quality of the image which has been taken when either the object or camera is in motion, in this case the image appears to be blurred, since that this blur is occurring due to the motion of the camera or object is known as MOTION BLURRING and we are removing this blur using DEBLURRING technique.

INTRODUCTION TO BLURRING:

When we scan, shoot, print, or develop a photo a little bit blurr is added at each stage and nearly all photographs can be sharpened with a high pass filtering. the real question is not ”can the photo be sharpened?” but the real question is “how much can they sharpen some photos can only sharpened at a tiny bit, while others can be sharpened quick significantly.

In different ways images get blurred:
1) Camera out of focus: A single pixel expands out to a circle of pixels. The blurr is an equal amount in all directions
2) Motion blurr: The camera or object moves while the picture is taken. The blurr is only in one direction
.**3) The digitizing process for a digital camera:** Continuous gradiations of color are transformed into points on a regular sampling grid Detail finer than the sampling frequency get average into a single pixel producing a softening effect

a)Developing a film: When a film is developed a negative is projected onto a photographic paper. The equipment can again be out of focus .This is more of an issue for old photographs which were manually developed, but even modern machines add a little amount of blurr.

b) **Scanning an image:** This can add quite a substantial amount of blurr especially if a cheap or odd scanner is used. Lots of scanners interpolate pixels to give higher ppi than what they can actually scan .

c) Printing the image: Inkjets and color laser printers used either patterns ,magazines and newspapers ,make “half tone” plates where the image is made into millions of tiny dots .It is often beneficial to slightly over-sharpen an image if it is to be printed on one of these media.

Deblurring technique is used to sharpen the blurred image where the causes of occurance of blurr may be one of them those are mentioned above.
Here we tried to improve the quality of the image which is being motion blurred by using deconvolution filters like

→ Regularized Filter
→ Lucy Richard Filter

→ **Weiner Filter**
→ **Gaussian Filter**
Depending upon the application of the filter we are using any filter for deblurring the motion blurred image.

2. REQUIREMENT ANALYSIS

In this project we deblurr an image i.e to improve the quality of the image that has been motion blurred .Deblurring is nothing but improving the quality of the image that has been degraded due to motion blurr which nothing but restoration.

What is image restoration?
Image restoration deals with methods to improve the quality of the blurred images it especially deals with the recovery of the information that was lost to the human eye during some degradation .For better understanding we consider the following degradation model

$$G(u,v) = H(u,v)*F(u,v)+N(u,v)$$

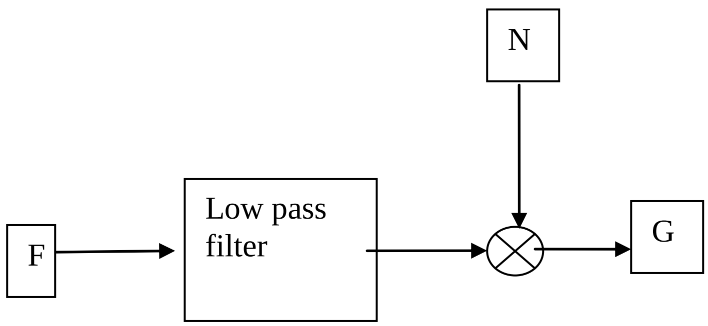


Figure 1. Image Restoration

Where G and F are Fourier transforms of the degraded image g and the input image f respectively .H is called the degradation function ,and N is a noise term modeled as an additive value.

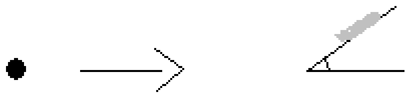
REQIREMENTS FOR RESTORATION:

The successful restoration of blurred image requires accurate estimation of PSF parameters .in our project, we deal with images, which are blurred by the relative motion between the imaging system and the original scene. Thus, given a motion blurred and noisy image, the task is to identify the PSF parameters and apply the restoration filter to get an approximation to the original scene.

Parameter estimation is based on the observation that image characteristics along the direction of motion are different than the characteristics in other directions .the PSF of motion blur is characterized by two parameters namely blur direction and the blur length.

Blur parameters:
Blur Length: it is the number of pixels by which the image is degraded .it is the number of pixel positions by which the pixel is shifted from its original position.

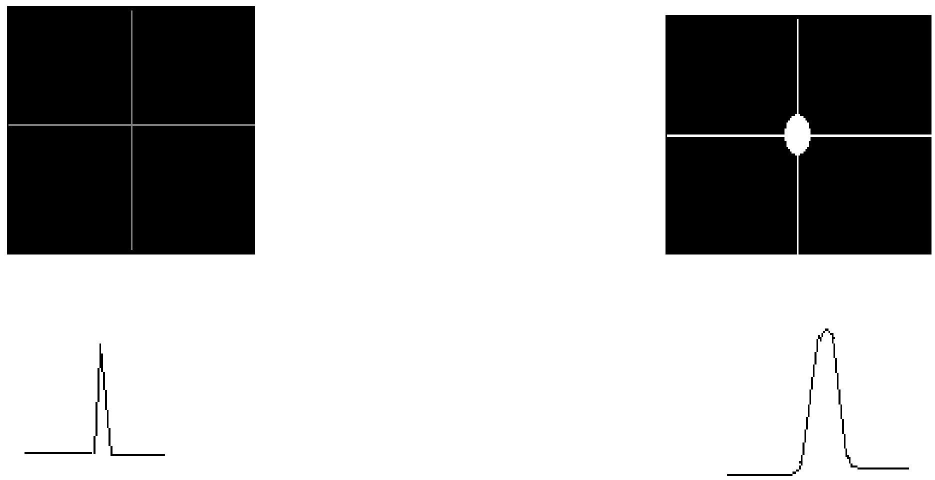
Blur Angle: it is the angle at which the image is degraded.Along the direction of the blur the image appears to be smoothened i.e, the intensity information of all the pixels along that direction remains almost the same.



3. DESIGN:

In this project we initially take an image which is considered as original image through out the project and we are blurring this image by using convolution techniques using PSF

PSF is the point spread function the information (intensity) at a pixel is spread over several pixels



Original **Degraded**

Figure 2.Pixel & PSF

Now that the image is blurred the necessity of deblurring occurs which involves deconvolution operation with the blurred image using the any of the filters mentioned earlier. here in this we are using Weiner filter since the blur is not caused due to lack of focus.

Weiner Filter:
The most important technique for removal of blur in images due to linear motion or unfocussed optics is the Wiener filter. From a signal processing standpoint, blurring due to linear motion in a photograph is the result of poor sampling. Each pixel in a digital representation of the photograph should represent the intensity of a single stationary point in front of the camera. Unfortunately, if the shutter speed is too slow and the camera is in motion, a given pixel will be an amalgram of intensities from points along the line of the camera's motion. This is a two-dimensional analogy to

$$G(u,v)=F(u,v).H(u,v)$$

Where F is the fourier transform of an "ideal" version of a given image, and H is the blurring function. In this case H is a sinc function: if three pixels in a line contain info from the same point on an image, the digital image will seem to have been convolved with a three-point boxcar in the time domain. Ideally one could reverse-engineer a Fest, or F estimate, if G and H are known. This technique is inverse filtering

Now that the image is blurred the necessity of deblurring occurs which involves deconvolution operation with the blurred image using the same filter as that was used in blurring.

BLOCK DIAGRAM:

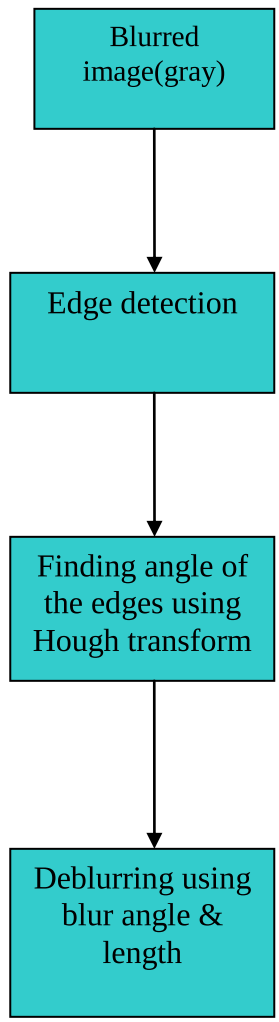


Figure 3. Block diagram of Proposed technique

4. IMPLEMENTATION

Here we are reading any image of jpg, png or tif by using the command “imread”.

Imread: The imread function supports four general syntaxes, described below. The imread function also supports several other format-specific syntaxes. See Special Case Syntax for information about these syntaxes.

Syntax: A = imread (filename, fmt)

After reading the image we are blurring this image by using the convolution operation over the original image with an averaging filter such as point spread function in turn this psf is created using fspecial function. Blurring is defined by two parameters namely:

→Blurr angle and

→Blurr length

Blurr angle: It specifies the direction along which the blurr has occurred.

Blurr length: It specifies the length to which the blurr is being spread.

Fspecial: h = fspecial (type) creates a two-dimensional filter h of the specified type. fspecial returns h as a correlation kernel, which is the appropriate form to use with imfilter. Type is a string having one of these values.
For motion blurring we use the following syntax of fspecial

Syntax: h = fspecial ('motion', len, theta) returns a filter to approximate, once convolved with an image, the linear motion of a camera by len pixels, with an angle of theta degrees in a counterclockwise direction. The filter becomes a vector for horizontal and vertical motions. The default len is 9 and the default theta is 0, which corresponds to a horizontal motion of nine pixels.

Here h=psf i.e., an averaging filter which on convolution with an original image gives the blurred output. This convolution is performed by using the Imfilter function.

Imfilter: B = imfilter (A, h, option) filters the multidimensional array A with the multidimensional filter h. The array A can be a no sparse numeric array of any class and dimension. The result B has the same size and class as A.

Where option is the operation we are performing i.e., convolution.

Now B is the blurred image. Now the original thing comes into picture that is deblurring. Here we use deconvolution technique using the Weiner filter as the blurr has occurred due to linear motion and not due to lack of focus. Here to deconvolute the image we need the blurr parameters to be determined.The blurr angle which is one among the parameters to be determined for deblurring process.

In the blurred image we can observe that along the blur direction the intensity information at a pixel is distributed among its neighbours and the image appears to be smoothened in that direction where as in the other direction the intensity from pixel to pixek changes depending on the image ,i.e, the image does not appears to be blurred in other direction here we show the edges of the original and the blurred image.

Figure 4: Edge of original image

Figure 5 : Edge of blurred image

By observing the above figures a and b it is clear that the edges of the blurred image are oriented in the direction of blur hence the orientation of edge can be used to find out the blur angle.

Edges can be found only if the image is in gray. To convert the rgb image to gray we are using the command rgb2gray.Edges can be found by using sobel operator (edge detecting mask).The orientation of these edges in order to find blur angle can be found by using Hough transform(since edges are nothing but the straight lines).

INTRODUCTION TO HOUGH TRANSFORM:

The **Hough transform** (HT), named after **Paul Hough** who patented the method in 1962, is a powerful global method for detecting edges. It transforms between the Cartesian space and a parameter space in which a Straight line (or other boundary formulation) can be defined.Let's consider the case where we have straight lines in an image. We first note that for every point (x,y) in that image, all the straight lines passing through that point satisfy Equation 1 for varying values of line slope and intercept (a,b) see Figure 1.

$\mathbf{Y_i=mx_i+c}$……………….> Eq (4.1)

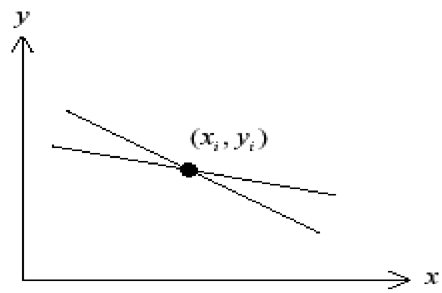


Figure 6. Lines through a point in Cartesian coordinates

Now if we reverse our variables and look instead at the values of (m, c) as a function of the image point coordinates (,) *i i x y* , then Equation 1 becomes:

$c = y_i - mx_i$……………………………………..*Eq (2)*

Equation 2 describes a straight line on a graph of c against m as shown in Figure 2.

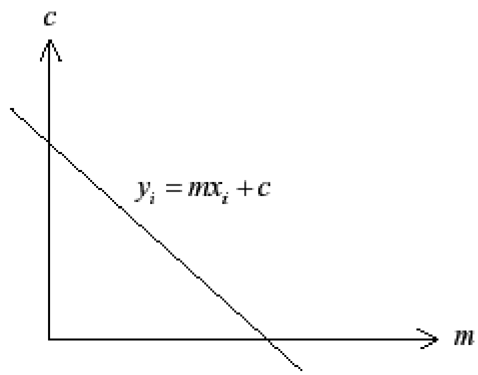


Figure 7. The (m,c) domain.

At this point, it is easy to see that each different line through the point (xi, yi) corresponds to one of the points on the line in the (m,c) space. Now, consider two pixels P1 and P2, which lie on the same line in the (x, y) space. For each pixel, we can represent all the possible lines through it by a single line in the (m, c) space. Thus a line in the (x, y) space that passes through both pixels must lie on the intersection of the two lines in the (m, c) space, which represent the two pixels. This means that all pixels which lie on the same line in the (*x*, *y*) space are represented by lines which all pass through a single point in the (*m*, *c*) space, see Figure 3 and Figure 4.

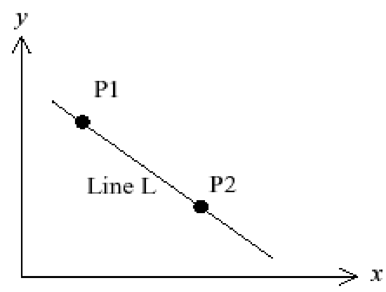


Figure 8. Points on the same line

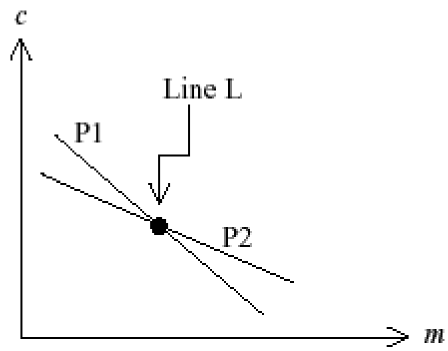


Figure 9. Mapping of P1 & P2 from Cartesian space to (m,c) space.

Following the discussion above, we now can describe an algorithm for detecting lines in images. The steps are as follows:
1. Find all the edge points in the image using any suitable edge detection scheme.
2. Quantize the (*m*, *c*) space into a two-dimensional matrix H with appropriate Quantization levels.
3. Initialize the matrix H to zero.
4. Each element of H matrix, $H(mi, c_i)$, which is found to correspond to an edge point is incremented by 1. The result is a histogram or a vote matrix showing the frequency of edge points corresponding to certain (*m*,*c*) values (i.e. points lying on a common line).
5. The histogram H is thresholded where only the large valued elements are taken. These elements correspond to lines in the original image.

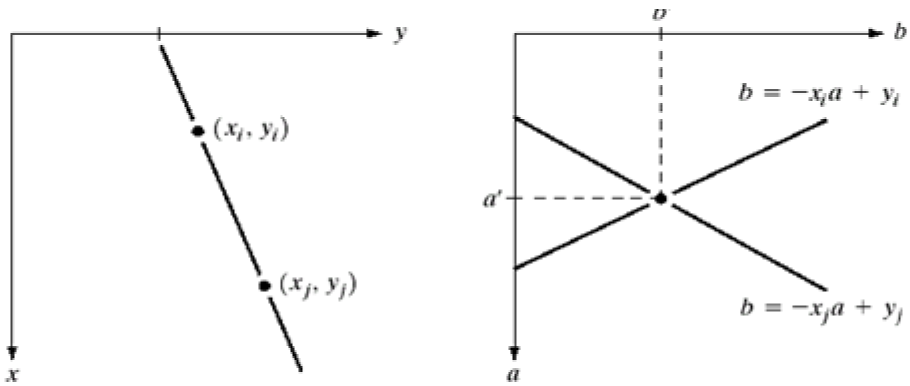
Advantages and Disadvantages:

The advantage of the Hough transform is that the pixels lying on one line need not all be contiguous. This can be very useful when trying to detect lines with short breaks in them due to noise, or when objects are partially occluded.
As for the disadvantages of the Hough transform, one is that it can give misleading results when objects happen to be aligned by chance. This clearly shows another disadvantage which is that the detected lines are infinite lines described by their (*m*,*c*) values, rather than finite lines with defined end points.

LINE DETECTION USING HOUGH TRANSFORM

Consider a point (x_i, y_i) and the general equation of straight line in slope-intercept form $y_i = ax_i + b$. infinitely many lines pass through (x_i, y_i), but they all satisfy equation yi=axi+b for varying values of a and b`. However. Writing this equation as $b = -x_i a + y_i$ and considering the ab plane (also called parameter space) yields the equation of a single line for a fixed pair (x_i, y_i). Further more, a second point (x_j, y_j) also has a line in parameter space associated with it, and this line intersects the line

associated with (xi, yi) at (a', b') where a' is the slope and b' is the intercept of the line containing both (xi, yi) and (x_j, y_j) in the XY plane. In fact all the points contained on this line have lines in parameter space that intersect at (a', b')
Figure illustrates these concepts.



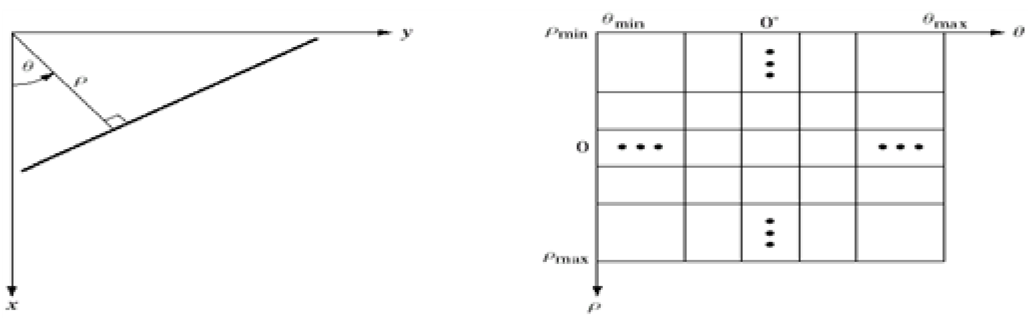


Figure 10. Normal representation of line and subdivision of (ρ,θ) plane into cell

The computational attractiveness of the Hough transform arises from subdivision fo the parameter space into so called accumulator cells, as illustrated in fig. 7.16 where (a_{max} , a_{min}) and (b_{max}, b_{min}) are the expected ranges of slope and intercept values. The cell at coordinates (i, j) with accumulator value A (i, j), corresponds to the square associated with parameter space coordinates of (a_i, b_j). Initially this cells are set to zero. Then for every point (x_k, y_k) in the image plane, we let the parameter a equal each of the allowed subdivision values on the a axis and solve for the corresponding b using the equation b= $-x_k a+y_k$, the resulting b's are rounded off to the nearest allowed value in the b axis. If a choice of a_p, results in solution b_q, we let A (p,q) =A(p,q)+1. at the end of this procedure, a value of M in A(i, j) corresponds to M points in the XY plane lying in the line y=a_ix+b_i. The accuracy of co linearity of these points is determined by the number of subdivision in the ab plane.

Note that subdividing the a axis into K increments gives, for every point (x_k, yk), K values of b corresponding to the K possible values of a. with n image points, this methods involves n_K competition. Thus the procedure discussed is linear in n, and the product n_K does not approach the number of computation discussed at the beginning of this section unless K approaches or exceeds n.

A problem with using the equation y=ax+b to represent a line is that both the slope and intercept approach infinity as the line approaches the vertical, one way around this difficulty is to use the normal representation of a line:

xcos θ + ysin θ =ρ………..eq(5.1)

the meaning of the parameters used in Eq(5.1).By using the procedure as mentioned above we construct the accumulator cells for parameter space. The use of this representation in constructing a table of accumulators is identical to the method discussed for the slope-intercept representation. Instead of straight lines. However, the loci are sinusoidal curves in the pw plane. As before. M collinear points lying on a line xcos θ_j +ysin $\theta_j = \rho_i$ yields M sinusoidal curves the that intersect at (ρ_i, θ_j) In the parameter space. Incrementing θ and solving for the corresponding ρ gives M entries in accumulator A (i, j) associated with the cell determined by (ρ_i, θ_j) . The range of angle θ is $\pm 90^0$ measured with respect to the x axis, fig.7.17 (a) a horizontal line has $\theta=0^0$, with ρ being equal to the positive x intercept, similarly a vertical line has $0=90^0$, with p being equal to the positive y intercept, or $\theta=-90^0$, with ρ being equal to the positive u intercept, or θ is $\pm 90^0$, with ρ being equal to the negative y intercept.

ALGORITHM FOR HOUGH TRANSFROM:

- Take the highlighted pixel from the edge of the blurred image.
- Find all the lines passing through the pixel that has been selected
- Find rho and theta for all the lines and increment the corresponding cell in Hough plane

Thus the angle which has been found using the Hough transform is used for deconvolution along with blurr length. Thus the output of deconvolution operation gives the desired output i.e., the DEBLURRED IMAGE.

5.RESULT:

BLURRED IMAGE

DEBLURRED IMAGE

ORIGINAL IMAGE

6. CONCLUSION:

Here we have developed a technique to deblurr the motion blurred images. By using the weiner filter in our project we have tried to remove blurr to the maximum extent and tried to produce more qualitative images.The deblurring is possible only if we know the blur length and blur angle with which the image has been blurred.In this project we have found the blur angle to deblur .Here there is also the need to know the blur length in prior with which the image has been blurred.

Limitation:

We tried to find blur length by using below methods

- By finding the standard deviations of the blurred image making using of masks by incrementing the size of the mask and comparing the result.
- Selecting the three or four extreme points of the edges of the blurred image and finding its average but were not able to get the accurate resulting determining the blur length.

Future scope:
To find the blur length by finding the FFT of the blurred image which gives the parallel lines with spacing between them may be proportional to the blur length by which the original image has been blurred.

7. REFERENCES

- Digital image processing using MATLAB--Gonzalez
- Digital image processing –Gonzalez
- Motion Deblurring in the Wild---Mehdi Noroozi, Paramanand Chandramouli, Paolo Favaro(Submitted on 5 Jan 2017 (v1), last revised 29 Aug 2017 (this version, v2))
- IEEE paper on Motion Blur Parameter Identification from a Linearly blurred image Tanaka,M.;Yoneji,K.;Okutomi,M.Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers

