ISSN: 2320-5407 Int. J. Adv. Res. 10(04), 1058-1067

Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/14646
DOI URL: http://dx.doi.org/10.21474/IJAR01/14646

RESEARCH ARTICLE

REDUCING ATMOSPHERIC TURBULENCE IN IMAGES CAPTURED THROUGH CCTV IN WAR FIELD AREA

Isha Padhy
Department of Computer Science and Engineering Chaitanya Bharathi Institute of Technology.

Manuscript Info

Manuscript History
Received: 28 February 2022
Final Accepted: 31 March 2022
Published: April 2022

Key words:-
Bregman Iterations, Turbulence Extraction, OpenCV, Warfield Images

Abstract

Atmospheric turbulence caused by chaotic changes in air pressure and velocity will deter Charge Coupled Devices from getting a useful image which limits the capabilities of a surveillance system. The turbulent nature of the atmosphere poses a great problem in many situations. One such area is the Military. With rising conflicts around the world, security through surveillance and reconnaissance has increased. In such a situation, videos or images captured through long range observation systems in sensitive areas across the world need to be clear and without errors. Atmospheric turbulence is mainly caused by Refractive Index variation. This variation can be across space as well as time. Though refractive index of the atmosphere is a major factor, there are other factors like – temperature, wind, pressure and humidity. The major goal of this paper is to reconstruct/repair the video/image obtained from the surveillance systems where the video/images are affected by turbulence. In this regard, we have focussed on the following aspects they are(a) Correcting the pixels and (b) Differentiating between static and dynamic objects.To achieve these objectives various solutions have been proposed based on both- mathematical and registration techniques. The registration techniques need some time during which many trial images/videos are taken in-order to construct a proper image. However, mathematical techniques involve performing complex operations in-order to achieve the correct value of the pixel. Our paper considersthe minimization of total variation in order to reduce the turbulence. It addresses the problem of turbulence in both, normal videos and thermal imaging.

Introduction:-

Image processing is one of the most potential branches in science today. Images capture the situation in the past. The problem of deciphering an image which has been affected with noise is very prevalent in different applications all across the world. With the need to tackle the problem, several mathematic theorems have been invented in the past as well as in the present. These techniques range from simple to complex implementation of different algorithms in order to enhance the image to an extent where a user can obtain useful information from it.

Atmospheric turbulence is one of the causes for image degradation. With varying weather and climatic conditions all across the world, the problem only expands to many applications to which work on images captured in open

Corresponding Author:- Isha padhy
Address**:-** Department of Computer Science and Engineering Chaitanya Bharathi Institute of Technology.

environment of far places. It hinders users from taking decisions in different applications. One such problem is seen in military.

The military of a country functions in places where it would be very difficult for people to live in given the weather and climatic conditions. For instance, the western border of India has a climatic condition ranging from high temperatures in Rajasthan to below 0° temperatures in Kashmir. Given, this disparity in weather conditions, the observation or surveillance system fall prey to the extremities. They fail to give users useful information which helps them to secure the borders and protect the country. This problem stemming from lack of proper images, is to be solved and is the main objective of this paper.

Problem Definition:-

Given images/videos which have been affected by atmospheric turbulence, enhance the images/videos in order to help the user extract useful information by looking at it is the definition of the problem which is being tackled by this paper The enhancement of the images is the crux of the problem. This step has been solved in different ways previously. However, they do not restrict this paper to propose a better solution.

The proposed system will overcome the problem of reference image creation. This paper will propose a mathematical solution without a necessity to create a reference image. Moreover, the system will treat frames of video in isolation i.e., an image won't depend on its previous image or the next image. Mathematical technique of total variation minimization will be used. This will reduce the noise there by enhancing the image.

The existing systems consist of reference image creation techniques where multiple images are used to create a reference image. This image will then be used in the subsequent process of image enhancement. Different methods like elastic image registration technique and turbulence mitigation through turbulence extraction use a reference image. Mathematical techniques were also used previously. These solutions use Principal Component analysis, centroids and other mathematical techniques to enhance the image. However, these techniques suffer with many problems. In techniques which use reference image creation, the system cannot be deployed immediately. This is because, the creation of reference image takes time. Moreover, the number of frames which are required to create a reference image cannot be determined. Extremely less number of frames to create a reference image is not suitable for the systems as it will have considerable number of errors. Extremely high number of frames to create a reference image won't be viable as at some point, the contribution by additional images for reference image will be negligible. Mathematical techniques suffer with similar problems like in capability to handle large input stream of images. Some solutions require a set of images to create one single image which will lead to loss of frames in between.

The proposed system will overcome the problem of reference image creation. This paper will propose a mathematical solution without a necessity to create a reference image. Moreover, the system will treat frames of video in isolation i.e. an image won't depend on its previous image or the next image. Mathematical technique of total variation minimization will be used. This will reduce the noise there by enhancing the image.

The remaining of the paper is divided into four parts, the section-2 describes the background study, the

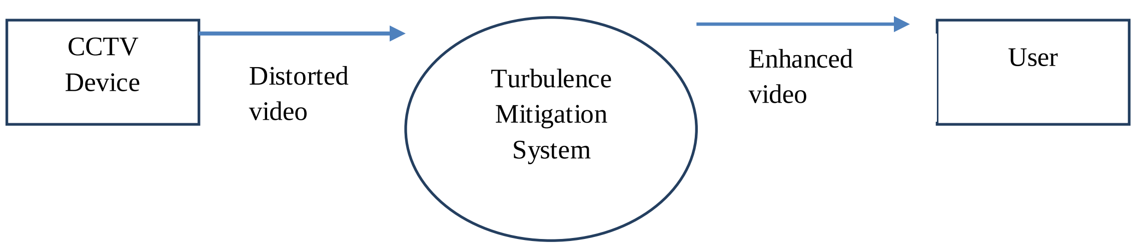

Fig1:- Dataflow diagram level-0.

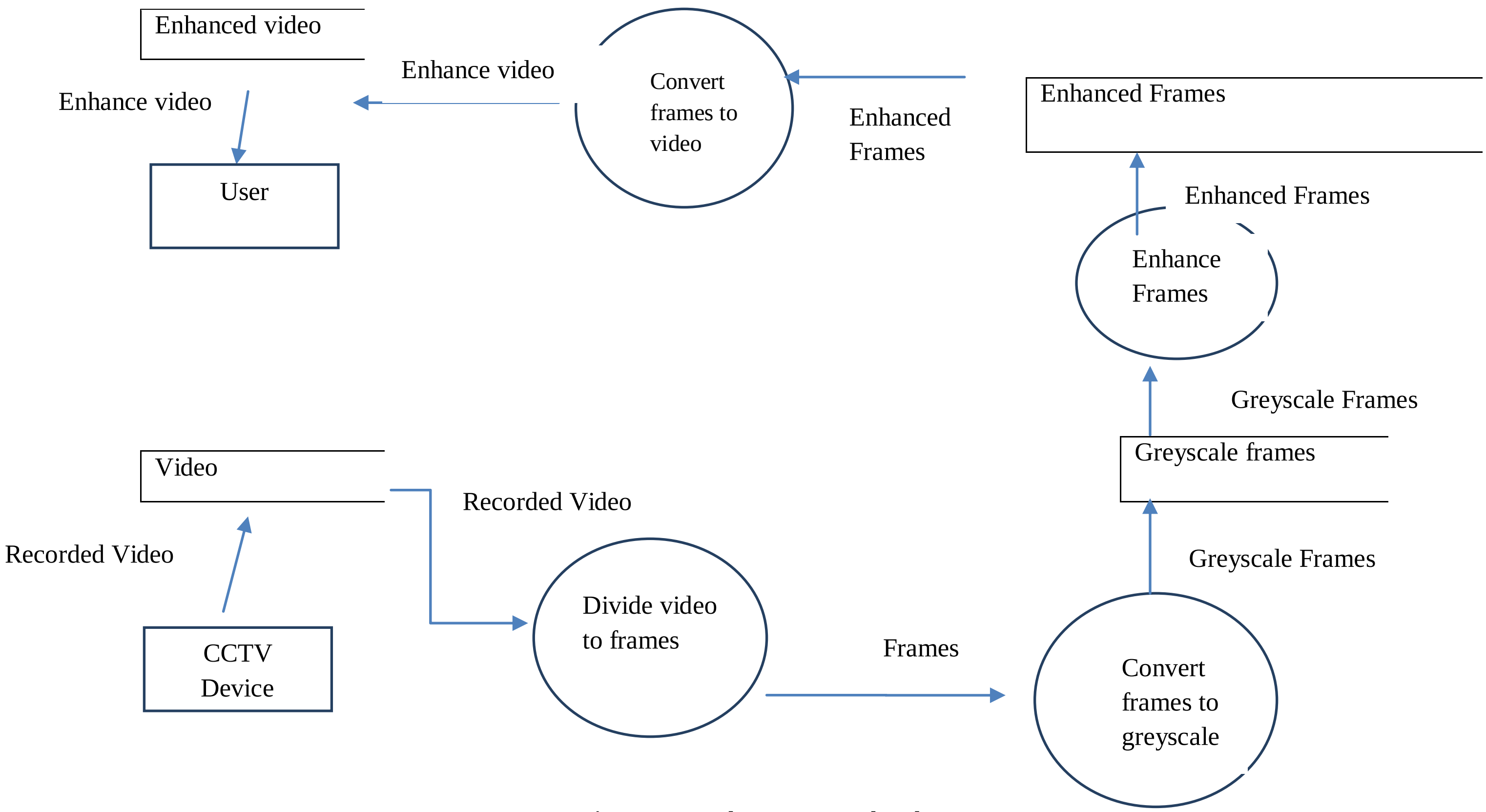

Fig 2:- Data Flow Diagram level-1.

section-3 describes the implementation and section-4 describes the conclusion and future work.

The data flow diagram of the system proposed is shown in the Figure 1 and Figure 2. Figure 1 shows a level-0 DFD. It has a CCD/TI Device and a User as its entities accessing the system which is represented in the centre as a process. Figure 2 shows a level-1 DFD. The data flows from the CCD/TI Device and is stored in a database Video. It is then accessed to divide into the frames which are then converted into grayscale. These grayscale frames are stored in a database to be accessed by a frame enhancing process to remove the turbulence. The enhanced frames are then stored to be combined later into an enhanced video. This enhanced video is accessed by the user.

Background study:-

Image enhancement can be done in different ways. To restore a scene distorted by atmospheric turbulence an algorithm which detects or differentiates static parts of the image and areas of real motion of objects in it[1]. One of the ways in which images can be enhanced is by using Non-linear elastic image registration technique. This is necessary for correcting the geometry of the image. This is achieved by first generating a geometrically undistorted reference (still image) of the area of our focus. This is iteratively done to achieve accuracy in the image. Then an Elastic registration model is created. The next step will involve correcting the turbulent image with the model created. The correction or compensation will allow restoration of pixels to its correct location i.e. to warp them back to their actual positions if turbulence were absent. This is iteratively done again for more accuracy by considering the previously compensated set of images. This step is especially used to restore the static portions of the image to their correct geometry by correcting the pixel position. However, the actual correction of moving objects occurs in the final stages. Accurate image can only be obtained if the system can differentiate between the distortion due to turbulence of atmosphere and the distortion due to a real moving object. This can be achieved by realizing the fact that the mathematical computations while compensating the pixels will result in different values for turbulent distortions and moving object distortions. This difference will be considered to restore the moving object in the image.

In this algorithm, an element wise rank filtering of each pixel is used to obtain a reference image for elastic registration. The rank smoothing has been used here because, when a light ray is passing through a turbulent atmosphere it gets deflected. However, the deflection is only up to a certain radius. Moreover, the deflection will be in such a way that the mean will be zero. That is, the center of the area of the light rays which are deflected will be the same in case these light rays have travelled in atmosphere which has zero turbulence. Having created a reference image, an elastic registration model will be used. The elastic registration model will map the reference image to the degraded image by considering neighbourhood of each pixel. These neighborhoods are registered to the reference image. All the pixels are mapped to non-turbulent locations. Once the mapping to the actual locations is done, translation vectors are obtained.

$$F(X,Y,T)= F'(x+dy,y+dy)$$

Where F(X,Y,T) is turbulent image frame and F'(x,y) are target reference image.

With the help of translation vectors obtained the pixels are restored to their original locations. That is, the pixels are returned to the positions which would have been the case when there was no turbulence. Figure 1shows the effect of the algorithm on a turbulence affected image. In fig 3. the left image shows degraded one and the right image shows the enhanced image.

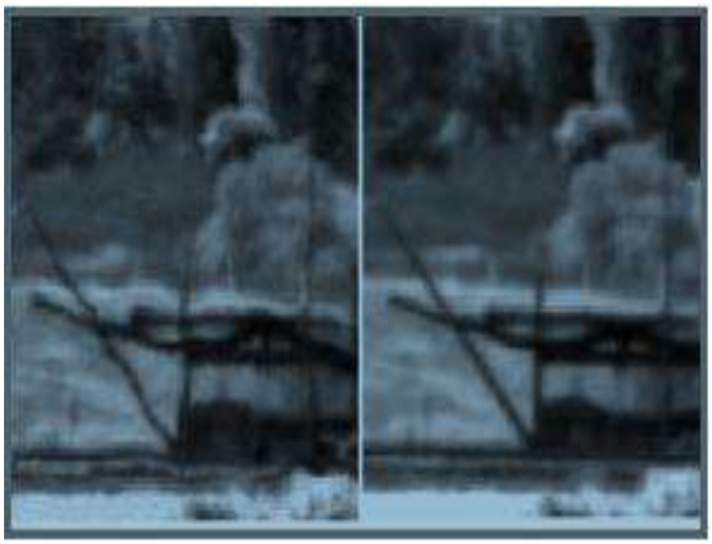

Fig 3:- Left image is a capture of degraded image, The right image is an enhanced image.

Another method which can be used for a turbulence mitigation correction through Principal Component Analysis [2]. In this method a number of images are taken to create one good image. These images are represented in terms of a column vector. Each pixel is an attribute. If there are K pixels in an image, then K parameters are considered in the column vector. These pixel values are normalized in such a way that they are in the interval [0,1]. If M images are used to construct a single proper image, it means M observations have been made over K parameters. It is assumed that the degradation is caused due to a low pass filter. A low pass filter is a one which reduces the frequency. Blurring is a smoothing process. When an image is blurred, the variance decreases i.e. it implies that blurring is a low pass filter which degraded the image. The solution proposes that by deburring the original image can be obtained.

The convolution shown below depicts how the turbulent image is obtained.

$$I_j= h*I$$

Where Ij is a degraded image, I is the original proper image and h is a low pass filter.

The solution proposes to construct a matrix such that the columns of the matrix are the column vectors of the M images subtracted by the temporal average of all images (say u). Let the matrix be A. The variance-covariance matrix of this obtained matrix is constructed, and the principle components are calculated by computing Eigen vectors to the corresponding Eigen value. The eigen vector obtained for the largest Eigen value is considered to be the first principle component. Let v_1 be the principle component with the largest variance i.e. the first principle component. This vector is multiplied with the vector A to produce the proper image J. A and v_1 are normalized in order to avoid different norms.

$$J=(A)(v_1) + u$$

The above is mathematical representation of the description.

Another method to remove the turbulence is through turbulence extraction [3].

Unlike traditional methods where turbulence is removed directly, this method removes turbulence by extracting the turbulent components and enhancing them. To do this, a sequence of operations need to be performed. Firstly, geometric distortion should be removed from the given input as a pre-processing step. Geometric distortion is nothing but the absence of linear correlation between pixels of subsequent frames. To remove spatially and temporally varying distortions (geometric) we apply a B-spline based non – rigid registration method. Then to separate the scene object and varying turbulence, we utilize a matrix decomposition method. This decomposition results in two components – a low rank background scene component and a sparse turbulent component. The low – rank background scene component is free from turbulence and the sparse turbulence component contains all the noise in the input image. Though the background scene component is free from turbulence, it has a blurred effect. Hence, the fine detail and coarse structure should be restored in the background scene component and noise should be removed from sparse turbulent component. To restore details and structure, a deconvolution method is applied on the low – rank background scene image. This removes the blur in this component. To remove noise from the turbulent component, turbulent patches are first identified. These patches are places where a sharp turbulence is observed. These patches are then extracted from the turbulent component. Now, these patches are enhanced to remove noise in them. Thus, we obtain a turbulence free component. Finally, output image or video is generated by fusing the deburred background image and enhanced detail layer together. Experiments have shown that this approach removes turbulence effectively. Thus, turbulence free images and videos are generated using extraction method.

The fourth method is the centroid method to enhance the image [4]. A common turbulence – degraded video is represented by local deformation as follows:

$$II(x) = I(x + u_i(x))$$

Where I is the base image without turbulence and u_1, u_2, u_3… is a sequence of random vector fields. The vector fields u_i are assumed to be smooth and have a point wise distribution k(u) of zero mean and finite variance. Now, to remove turbulence, we have to find I from the above equation. To solve this, there are two approaches.

First, we take the average of all input images and then de – convolve the average image obtained. In doing so, the input sequence converges to the image I. Second method is to invert one of the images in the input sequence I_i to obtain I. Although these two methods exist, there are limitations to these methods. In the first method, one must know exactly the complete distribution k of the vector fields ui and have large enough set of independent images I_i. In the second case, only a single image I_1 is needed but the field u_1 must be known exactly. In either case, it is impossible to know these values. Hence, these values must be approximated.

Thus, the centroid method is introduced. This method is intended to at least find ui from the input sequence Ii. In this method, the deformation between images is found through direction of optical flow between a pair of images. Optical flow allows to interpolate two images without interpolating their color values. Therefore, instead of computing the linear average image, we transform the image by the average of optical flows. This construction corresponds to the Karcher – Frechet mean on a metric space of images. Thus, the centroid method computes the Karcher – Frechet mean of the inputsequence in order to correct turbulence. To determine the optical flow, any of the optical flow algorithms may be used. Thus centroid method corrects turbulence. One main advantage of this method is that it uses optical flow between pair of images rather that the average of images. This is also effective to get better results using small number of inputs.

Methodology:-

System Design:-

The proposed solution will enhance the image by reducing the total variation. The geometrical distortions will be reduced in this solution. Total variation regularization is method to reduce noise in the images [5]. Images with noise usually have high total variation. That is the signals with more detail will have high total variation. The solution proposes to reduce the total variation. The regularization problem can be solved using BregmenIterations [6] are used .Bregmen Iterations is a mathematical technique which is used to solve optimization problems.

In this case it is to minimize the total variation in the image.
Let observed image be denoted by f.
Let u be the actual image to be captured.
Let D be a geometric deformation on image.

$$f = (u)(D)+noise$$

The above equation is the mathematical representation of the image obtained. The regularization is on the term u (denoted by J(u)), which denotes that u should have the least total variation possible. The problem will be solved by initially guessing the value of u. In this paper, the initial guess of u will be the average of a particular number of frames obtained. Such an average calculated is blurry. The estimated u will then be used to estimate D from the above equation. Let the estimate of D be A. The Bregmen Iterations are then started.

The equations are given by:

$$u^k=\min J(u)+(G/2)||Au-f^k||^2$$
$$f^k=f^k+f-Au^k$$

Now after the first iteration of this equation, D is estimated again (say A). And the whole process is repeated several times. The number of iterations is given by Nb which are known as Bergman iteration count. A regularization parameter G is also used. The superscript for f and u denotes the version of the u and k obtained. Finally, an image u is obtained which is the proper image.

Proposed algorithm:-
1. A video is given as an input to the algorithm.
2. The video is divided into the frames
3. Each frame is converted to greyscale do (4)
4. For Nb iterations do

a. 4.1) Estimate k(x) from equation (a) . Say A.
b. 4.2) $u^k=\min J(u)+(G/2)||Au-f^k||^2$
c. 4.3) $f^k=f^k+f-Au^k$
d. 4.4) u is result

5. Combine all the frames to form the video.

Implementation of proposed solution:-
Given the following tasks:
1. Split video to frames.
2. Convert each frame to greyscale.
3. Enhance each frame by Total Variation minimization
4. Combine enhanced frame to form the video.

Different libraries and tools have been used to develop each module. All the modules have been integrated and tested successfully. However, to integrate all the modules shell and python scripts have been used. To execute shell commands in python os.system() method has been used. . High level tools/commands can be executed using this utility. In the context of this paper, os.system() has been used to invoke the ffmpeg tool.

The task of splitting the frames and converting them to greyscale is carried out by the methods in OpenCV library implemented in python. The different methods used are:
1. VideoCapture which is used to capture the video from either a live source or take a video from a location in the system
2. Read() is used to extract each frame from the video
3. cvtColor() is used to convert the Blue Green Red format of an image to a greyscale one.

It must be noted that the reason for the conversion of RGB image to Greyscale is necessary to maintain the uniformity. RGB makes an image vary in 3 dimensions. With the image reduced to only 1 dimension i.e. each pixel having only one value, the burden of considering all the 3 aspects i.e. Red, Blue and Green can be avoided. It improves the performance by avoiding unnecessary computations. With this the flow of the program arrives at the point of repairing the frames i.e. reducing the turbulence in the video. The entire process however has a single point entry.

The system has a frontend graphical user interface which makes it very simple for a user to utilize the system as shown in Figure 4. The flask python library has been used to host the system. As soon as the program is executed to begin the system, a flask server runs by giving a web application feature to it.

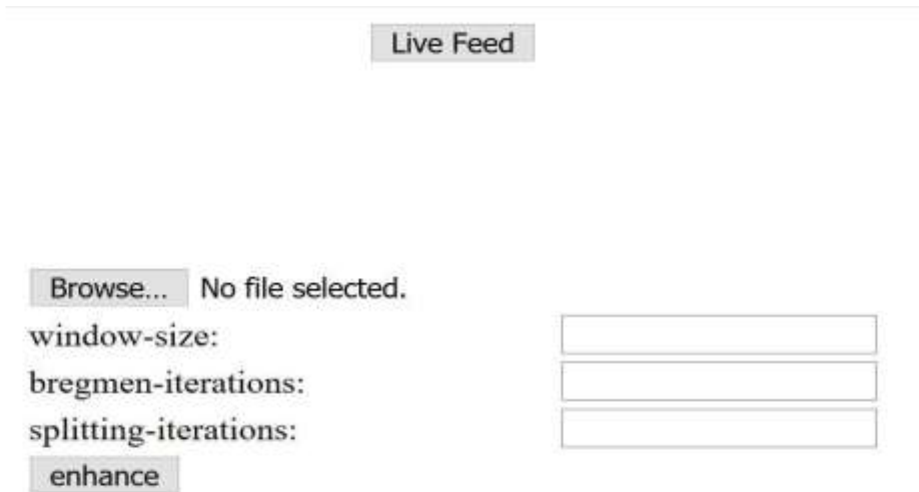

Fig 4:-

The server runs to
1. Access the turbulent effected video
2. Initiate the flow of program
3. Enter parameters like:-
4. Number of Bregmen iterations
5. Number of operator splitting iterations to solve Bregmen iterations.
6. Window size parameter

The flask server acts as a single point entry to the entire system which has a linear flow. All the 4 tasks mentioned at the beginning of this subsection happen in a sequential fashion once the parameters are entered into the flask server.

The window size parameter specifies the number of frames taken each time an enhanced frame is to be generated. For example, to construct frame 1 frames 1-6 are taken. Similarly, for 2nd frame to be generated, frames 2 to 7 are taken. Windows are chosen for Bregmen iteration to work is to consider an initial estimate of the image for which an average is considered. The window-size, Bregmen iteration count and operator splitting iterations to solve Bregmen equations when increased, affects the performance since the number of computations increase.

While the opencv library is used for splitting the video into frames and converting to greyscale it doesn’t have any role in enhancing the quality of the frames. However, in order to perform the Bregman iterations for enhancing the images, Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman technique is used.[7] The source code for mathematical implementation of Bergman iteration has an open-source license. This image enhancement algorithm runs through all the frames which have been generated from the input video i.e. the video which has been affected by turbulence. All the enhanced images are then stored to be later converted into the video. In order to do this the ffmpeg tool is used.

The ffmpeg tool is given with many arguments such as:-
1. Frame Rate- The frequency of frames or the number of frames per second
2. Frame size- gives the width and length of the frame
3. Source- this gives the location of the input file
4. Vcodec- to specify the encoder to be used.
5. CRF- Constant Rate Factor is the default quality settings for encoders. It is used in making decisions to balance the trade-offs between space and quality of the video or image.
6. Pixel format- which specifies the format of the pixel like youv420p

The ffmpeg tool is also used to compare the results side by side. The filter hstack is used to generate a video file which has both the input video (affected with turbulence) and the output video (enhanced video) side by side for the sake of comparison.

The entry point to the program is main.py. This file upon executed, begins the flask server. The flask server starts running at 127.0.0.1 as programmed in the file. Once the server runs, it renders the html file which is the one in Figure 4. Once the values are entered and the video file is browsed, the submit button is clicked to begin the process. The flow transfers to frames.py. The frames.py file implements the OpenCV library. It reads the entire video and divides them into frames. The video supplied here for example is sample.mp4. Each frame is in turn converted from RGB to Greyscale. They are stored in the directory “data”. These sequences of frames are named in a uniform format so that the accessing of such files can be done by using regular expression. The folder looks as given in the figure 5.

Fig 5:- Format of the generated frames.

Results And Discussions:-

Different test cases have been used to evaluate the system. These test cases range from those affected severely by turbulence to the ones affected mildly. The working of this system has showed that the solution proposed works even for thermal imaging. The working of this system however cannot be appreciated when viewed frame by frame for the test cases provided. This is because a frame constitutes around 1/24th of a second. Recognizing that the frame is affected by turbulence and appreciating the reduction in the turbulence after running the system is very difficult. Some of the test cases are provided.

Fig 6:- Test case with result.

Fig 6:- The left frame image is affected by turbulence. The right side frame is the enhanced one. Careful observation into these pictures show the differences.

With the increase in the resolution of the video the time taken to enhance the frame increases. This is because each frame when enhanced requires each of the pixel value to be repositioned or recalculated. Hence, the time taken will increase. On the other hand, the length of the video also affects the time. As the length of video increases the time taken to enhance it also increases.

The number of frames per second is also a very important factor to consider. More the number of frames per second, more the smoothness in the video. Good videos usually have higher frame rates i.e. frames per second. However, increase in the frame rate increases the number of frames which in turn leads to more number of computations. In a standard video stream, each second has 24 frames in it. However, many situations require even more number of frames per second to avoid missing even the smallest details. Hence, though higher number of frames per second is desirable, there is a trade-off with the performance of system which takes less time for a smaller frame rate.

The number of frames required to guess the initial image which is iteratively enhanced also influences the performance. It is to be noted that this system has been tested with 6 frames in a window to calculate the object. The increase in the number of frames taken to calculate the average increases the time.

Finally, the number of Bregman iterations clearly influences the performance of the system. The higher the number of Bregman iterations performed, the better is the quality of the video. This is because the total variation is minimized at each iteration which is the reason behind reducing the turbulence. However, to perform each iteration is an additional cost on the time taken to run. The operator splitting technique is used to solve the Bergman iterations. Each Bregman iteration is solved using operator splitting method. They can be viewed as two loops one inside the other. Considering both the issues, the timing complexity is non-linear.

The space complexity is directly dependent only on the length and the number of frames per second. Increase in any one of the parameters, length or frame rate, increases the number of images to be stored both- after and before the enhancement.

Another feature which can be implemented is the live feed. The live feed is the ability of the system to capture the situation live and send it to the enhancement part of the system. The system must be implemented in such a way that whenever the capturing starts, immediately the frames obtained must be sent for enhancement. The capturing of the situation and the enhancement of the frames must take place simultaneously. However, this feature requires high computational power.

Conclusion And Future Work:-

The system clearly reduces the turbulence in the videos. It not only works for normal videos but also for thermal imaging. It is completely integrated and automated. A flask server with a frontend acts as an entry point to the system. The tasks of dividing the video, converting the frames to greyscale, enhancing the frames, combining them into video and comparing it with the input are done in a sequential manner eventually. However, many aspects can still be focused on.

One of the main issues being faced in this paper is that the output is obtained as a greyscale video even if the input has RGB components in it. The conversion of this greyscale video into RGB is challenging. To get back the colours, we need to create a map for the values of the intensity of colours at each pixel. However, such a task requires huge amount of space requirements- almost 4 times the space which is required now. Given the application of this paper for surveillance, significantly large videos are used every day. Storing and mapping RGB components for each frame in such a large video is very difficult and unviable. Moreover, the frames have been modified. Mapping back the colours directly is not a good idea. Another issue is the detection of fast-moving objects. As the speed of the object in a video increases the rate of change in the frames per second also increases. The amount of unique data in each frame increases. In such situations the average of the frames might result in the data loss. This might include the structure of the body being distorted.

In future work, we plan to implement object detection using neural network techniques so that the objects can be clearly considered. One of the major difficulty that we faced is storing huge amount of data which has to be worked on by considering any one of the features of the images.

References:-

1. S.Gepsthein , A.Shtainman , B.Fishbain , L.P.Yaroslavsky, “Restoration of atmospheric turbulent video conaining real motion using rank filtering and elastic image registration” 2004.
2. Tristan Dagobert , YohannTendero , Stephane Landeau , “Study of the principal component analysis method for the correction of images degraded by turbulence” 2018.
3. Renjie He , Zhiyong Wang , Yangyu Fan , David Fengg, “Atmospheric turbulence mitigation based on turbulence extraction” 2016.
4. Enric Meinhardt-Llopis , Mario Micheli, “Implementation of centroid method for the correction of turbulence” , 2014.
5. Leonid I.Rudin , Stanley Osher , Emad Fatemi, “Nonlinear total variation based noise removal algorithms”, 2002.
6. Stanley Osher , Martin Burger , Donald Goldfarb , Jinjun Li , Wotao Yin, “An iterative regularization method for total variation-based image restoration” 2006.
7. Pascal Grtreur, “Rudin–Osher–Fatemi Total Variation Denoising using SplitBregman”, 2012.
8. Nieuwenhuizen, R., Dijk, J. & Schutte, K. Dynamic turbulence mitigation for long-range imaging in the presence of large moving objects. J Image Video Proc. 2019, 2 , https://doi.org/10.1186/s13640-018-0380-9.