Learning to estimate transient parameters in computational digital photography for coded image enhancement

A **Project Report** submitted in partial fulfillment of
the requirements for the degree of

Bachelor of Technology

by

Mayur Talole	**B.Tech J-37**
Rohit Somwanshi	**B.Tech J-34**
Suraj Shiras	**B.Tech K-75**

Under the guidance of
Prof. Milind V. Kamble

DEPARTMENT OF ELECTRONICS ENGINEERING
VISHWAKARMA INSTITUTE OF TECHNOLOGY PUNE

2014 - 2015

Bansilal Ramnath Agarwal Charitable Trust's

VISHWAKARMA INSTITUTE OF TECHNOLOGY, PUNE - 37

(An Autonomous Institute Affiliated to SAVITRIBAI PHULE PUNE UNIVERSITY)

CERTIFICATE

This is to certify that the **Project Report** entitled **Learning to estimate transient parameters in computational digital photography for coded image enhancement** has been submitted in the academic year **2014-15** by

Mayur Talole	**B.Tech J-37**
Rohit Somwanshi	**B.Tech J-34**
Suraj Shiras	**B.Tech K-75**

under the supervision of **Prof. Milind V. Kamble** in partial fulfillment of the requirements for the degree of Bachelor of Technology in **Electronics / Electronics and Telecommunication Engineering** as prescribed by SAVITRIBAI PHULE PUNE UNIVERSITY.

Guide/Supervisor
Name: Prof. Milind V. Kamble
Signature:

Head of the Department
Name: Prof. A. M. Chopde
Signature:

External Examiner
Name:
Signature:

Acknowledgments

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of our project work. At the outset, we would like to thank Vishwakarma Institute of Technology, Pune for giving me the opportunity to pursue a project under the guidance of Prof. Milind V. Kamble. I would like to express my sincere gratitude to my advisor Prof. Milind V. Kamble for his patience, encouragement, motivation and support throughout the course of the project.

We would like to thank Prof. A. M. Chopde and Prof. M. Shidore for providing helping hand during the project work. Many thanks will also be directed towards the staff at the Electronics department at for allowing to complete our project successfully.

A special thank goes to Mr. Nikhil Nirhale for providing camera equipment for project purposes without any hesitation. Our team is grateful to Mr. Mayuresh Maradkar for his help in making of camera lab which later proved to be most important part of this project.

We would like to thank Mr. Shantanu Sinha and Mr. Achuta Kadambi at the Camera Culture Group at the MIT Media Lab, USA to share vital information and give expert opinion and suggestions. Without them, my compilation on this topic would not have been possible. It is difficult to put in words my appreciation for all guidance and support provided by so many people over course of project.

Date: __________________

Mayur Talole

Rohit Somwanshi

Suraj Shiras

Contents

List of Tables

List of Figures

Abstract

Capturing of fast moving objects with Digital Camera is a hard problem for almost everyone. To avoid motion blur in captured images, it is essential to have knowledge of camera parameters such as shutter speed, aperture, ISO, exposure etc. This project is among one to enable a layman to capture fast moving object after few trials. Project work involves knowledge of digital image processing, embedded systems, pattern recognition. Our Team has developed a laboratory in order to avoid human interfaces and errors while creating training data samples.

Computational photography combines plentiful computing, digital sensors, modern optics, actuators, and smart lights to escape the limitations of traditional cameras, enables novel imaging applications and simplifies many computer vision tasks. However, a majority of current Computational photography methods involves taking multiple sequential photos by changing scene parameters and fusing the photos to create a richer representation. Our Team has described applications of coding exposure, aperture, illumination and sensing and describe emerging techniques to recover scene parameters from coded photographs.

Team has developed an algorithm for quantitative measurements of image parameters such as blur, contrast and entropy of a captured image. Also the project proposes a hypothesis for a new concept of behavioral entity of camera parameters which will enable user to determine camera constraints automatically and efficiently. Using curve fitting tool, project arrived at inter-depending plots of camera parameters and image parameters. Our Team then developed clusters to classify captured image. Each cluster is having distinct equations. Processing these equations after classification of image would give set of camera parameters to be used for next take.

Chapter 1

Introduction

1.1 An Idea

The main idea here is to construct the system which will enable the user to capture the perfect photos using DSLR camera. The perfection of image is generally examined by considering some image parameters such as resolution, blurness, and contrast of image. In present day, photography is done mainly using auto mode in DSLRs, which causes user to lose lot of important data in image. So now a days people are choosing to take photos in manual mode. It is very difficult for an amateur to take perfect photos in manual mode using any DSLR camera. So our team is working on developing the application which will suggest user at what parameters one should take photos. The system will check every environmental conditions or other constraints which may affect camera behaviour.

1.2 Towards results

Here the application mainly relates the points such as blur and contrast to the sharpness of image captured by camera. Primary function of the system concerns with developing an adaptive algorithm which will be able to quantify the blur content present in the image. This blur is dependent of the camera parameters such as shutter speed, aperture, ISO and exposure. For the ease of complexity the current system is concentrating on keeping two parameters ISO and Exposure constant. The dependence of the shutter speed and blur, shutter speed and contrast, aperture and blur and aperture and contrast in the image is determined by analyzing the database of images captured at xed camera parameters. Initially the relation is expressed using various curve tting models in MATLAB. Each tting is checked for under tted, critically tted or over tted nature of curve based on the smoothing of tted curve. The important problem faced during the detection and calculation of blur in the image was the increase in the darkness of image with respect to decrease in shutter speed and F-stop numbers of aperture. The problem is solved by applying the separate algorithms to each of the image which are separated on the basis of the contrast value. This was done by keeping in mind that generally the images which have the contrast value higher than predened threshold value have been separated into other section based on the fact that those image contents are darker and they need RGB normalization which employs the histogram stretching and histogram equalization fundamentals in all planes separately. Presently this value is predened but later it will be adaptive based on the continuously captured photos by taking the average of those contrasts.

1.3 Formulating new concept

As this application is camera specific so the formulated relations and formulae cannot be applied to use of another camera. Here our team proposed theory of behavioural entity of camera which will be basic entity of each camera possessing some properties. This entity will adapt to determine the behavior model of camera by taking some photos and working along the algorithm. This model will be different for different cameras will be based on automatic processing. This behavioral entity will not only help to our current application but also other work which will reduce time spend on determining the camera conditions.

1.4 End user application

Then the normalised image is again sent to blur determination function in MATLAB. This complete set of data is used for equation generation and tested for some of the images. Initially it is tested across the interrelation curve of camera parameters such as aperture and shutter speed to generate equations. Later it is intended to employ same technique in order to get interrelation between all the camera parameters. The nal project will be the system which will process the photo based on the algorithms and will give the output in terms of the aperture a shutter speed values using which the perfect image will be captured which will have good intensity, minimum blur and good contrast.

Chapter 2

Camera Parameters

2.1 Aperture

Aperture is the opening in the lens. Aperture is measured in f-stops. In optics, an aperture is a hole or an opening through which light travels. More specifically, the aperture of an optical system is the opening that determines the cone angle of a bundle of rays that come to a focus in the image plane. The aperture determines how collimated the admitted rays are, which is of great importance for the appearance at the image plane. If an aperture is narrow, then highly collimated rays are admitted, resulting in a sharp focus at the image plane. If an aperture is wide, then uncollimated rays are admitted, resulting in a sharp focus only for rays with a certain focal length. This means that a wide aperture results in an image that is sharp around what the lens is focusing on. The aperture also determines how many of the incoming rays are actually admitted and thus how much light reaches the image plane (the narrower the aperture, the darker the image for a given exposure time). In the human eye, the pupil is the aperture.

The aperture stop of a photographic lens can be adjusted to control the amount of light reaching the film or image sensor. In combination with variation of shutter speed, the aperture size will regulate the film's or image sensor's degree of exposure to light. Typically, a fast shutter will require a larger aperture to ensure sufficient light exposure, and a slow shutter will require a smaller aperture to avoid excessive exposure.

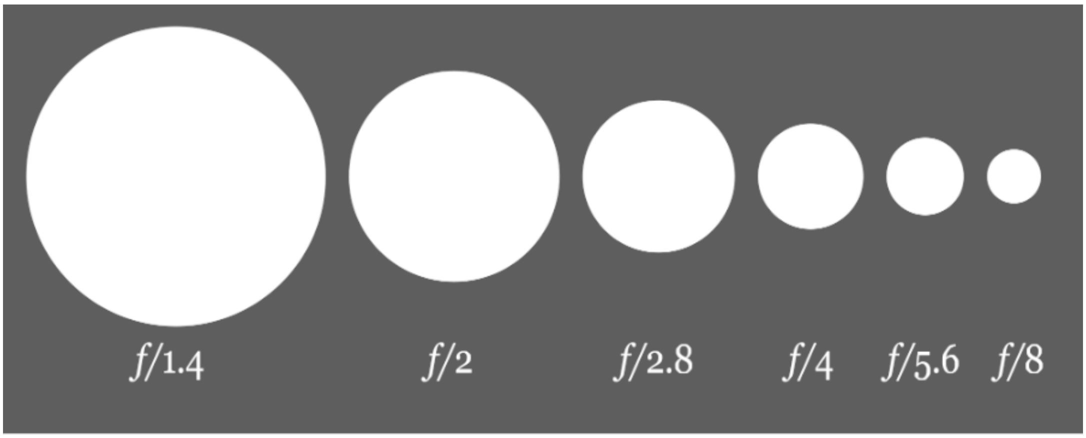


Figure 2.1: Aperture ranges

Figure 2.2: Large and small aperture comparison

2.1.1 Depth of Field

Depth of field (DOF), also called focus range or effective focus range, is the distance between the nearest and farthest objects in a scene that appear acceptably sharp in an image. Although a lens can precisely focus at only one distance at a time, the decrease in sharpness is gradual on each side of the focused distance, so that within the DOF, the unsharpness is imperceptible under normal viewing conditions.

In some cases, it may be desirable to have the entire image sharp, and a large DOF is appropriate. In other cases, a small DOF may be more effective, emphasizing the subject while de-emphasizing the foreground and background.

The depth of field does not abruptly change from sharp to unsharp, but instead occurs as a gradual transition. In fact, everything immediately in front of or behind the focusing distance begins to lose sharpness - even if this is not perceived by our eyes or by the resolution of the camera.

2.2 Shutter Speed

Shutter speed is the amount of time that the shutter is open. In photography, a shutter is a device that allows light to pass for a determined period of time, exposing photographic film or a light-sensitive electronic sensor to light in order to capture a permanent image of a scene. A shutter can also be used to allow pulses of light to pass outwards, as seen in a movie projector or a signal lamp.

Slow shutter speed combined with panning the camera can achieve a motion blur for moving objects. Hence moving pixel introduces blur in image.

Shutter speeds change as shown below:

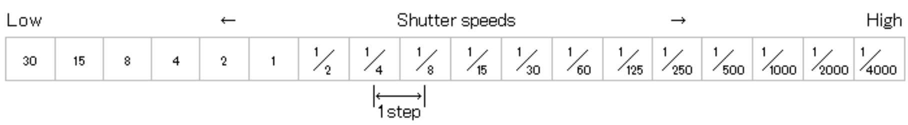


Figure 2.3: Various ranges of shutter speeds

Choosing a shutter speed one step faster than the current shutter speed (by, for example, changing shutter speed from 1/60 s to 1/125 s) is referred to as increasing shutter speed by one step and halves the amount of time the shutter is open. Choosing a shutter speed one step slower than the current shutter speed (for example, by changing shutter speed from 1/125 s to 1/60 s) is referred to as slowing shutter speed by one step and doubles the amount of time the shutter is open.

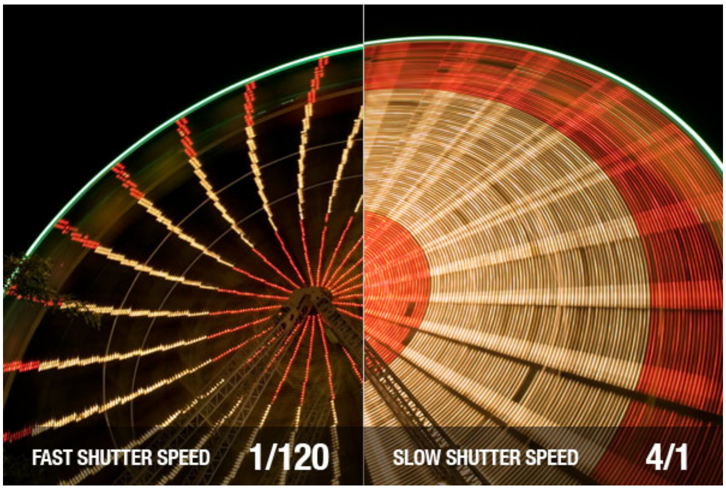


Figure 2.4: comparison of fast and slow shutter speed

2.3 Camera Blur and Motion Blur

If the camera or subject moves while the shutter is open, the picture will be blurred. Blur caused by subject movement is referred to as subject blur or motion blur; blur caused by camera movement (camera shake) is referred to as camera blur. The results in both cases are similar, but whereas blur caused by subject movement is generally regarded as a legitimate way of expressing motion in photographs, blur caused by camera shake is frequently seen as a flaw. While camera blur does not necessarily render a photograph a failure, caution should be observed to avoid unintentional camera blur. The main subject is in both cases blurred, but the results are distinct from blur caused by the subject being out of focus (focus blur).

2.4 ISO

Film speed is the measure of a photographic film's sensitivity to light, determined by sensitometry and measured on various numerical scales, the most recent being the ISOsystem. A closely related ISO system is used to measure the sensitivity of digital imaging systems.

In digital cameras, raising the ISO means a similar decrease in quality, with an increase in what's called "noise." It's the digital equivalent of grain. Very early digital cameras had objectionable levels of noise at ISOs as low as 800. Today most digital SLRs can make good quality images at ISOs up to 1600 and above. However, several variables affect this.

One important factor affecting the amount of digital noise in an image is the size of the pixels used on the sensor. Large pixels result in less noise than small ones. That's why digital SLRs perform much better at high ISOs than compact cameras. The SLRs have larger sensors and larger pixels.

Another factor is the amount and type of noise reduction being applied in the camera. Because all pixels collect some noise, every digital camera runs processing on every image (although with a NEF, or RAW, file that can be changed later) to minimize that noise. Newer cameras use newer technology to reduce that noise, with the result being less noise at similar ISOs than what earlier cameras could achieve.

All of this means photographers are constantly doing a balancing act. They want to keep their ISO low for high quality images (low noise), but also they need a fast enough shutter speed to get a sharp picture.

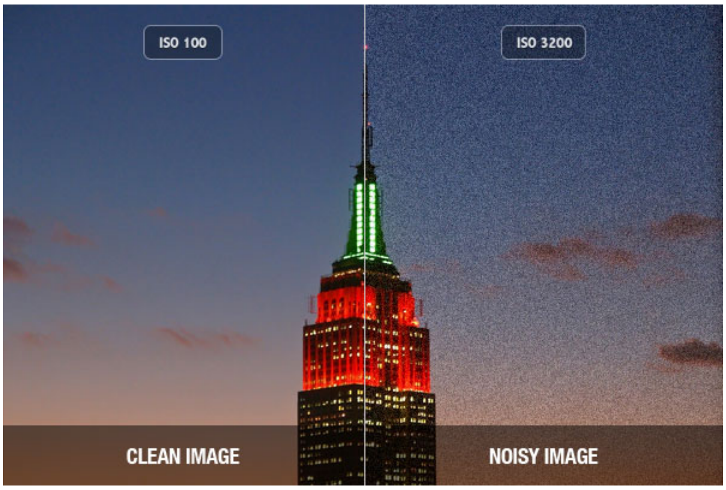


Figure 2.5: Comparison of low and high ISO

How does ISO speed affect the photograph?

If we set the digital camera to a low ISO, for example 100, the resulting photograph will be better quality than one set at 1600. The higher the ISO the grainier the photo will look. Therefore go for a low ISO number whenever possible.

However there are circumstances where a lower quality photograph is better than none at all. For example taking photographs of fish in a dark aquarium would normally be out of range for most point and shoot cameras. Setting camera to a high 1600 ISO made this task possible. Otherwise there would not have been enough light let in to the sensor and the image would have turned out totally black.

We could have chosen a lower ISO camera setting and used the in built flash. With this in mind a higher ISO (say 800) is also recommended if we want to take photographs without using a flash.

2.5 Exposure and Exposure Triangle

Exposure is a critical element that determines what is actually recorded on film or the image sensor. There are three adjustable elements that control the exposure - ISO, Aperture and Shutter Speed.

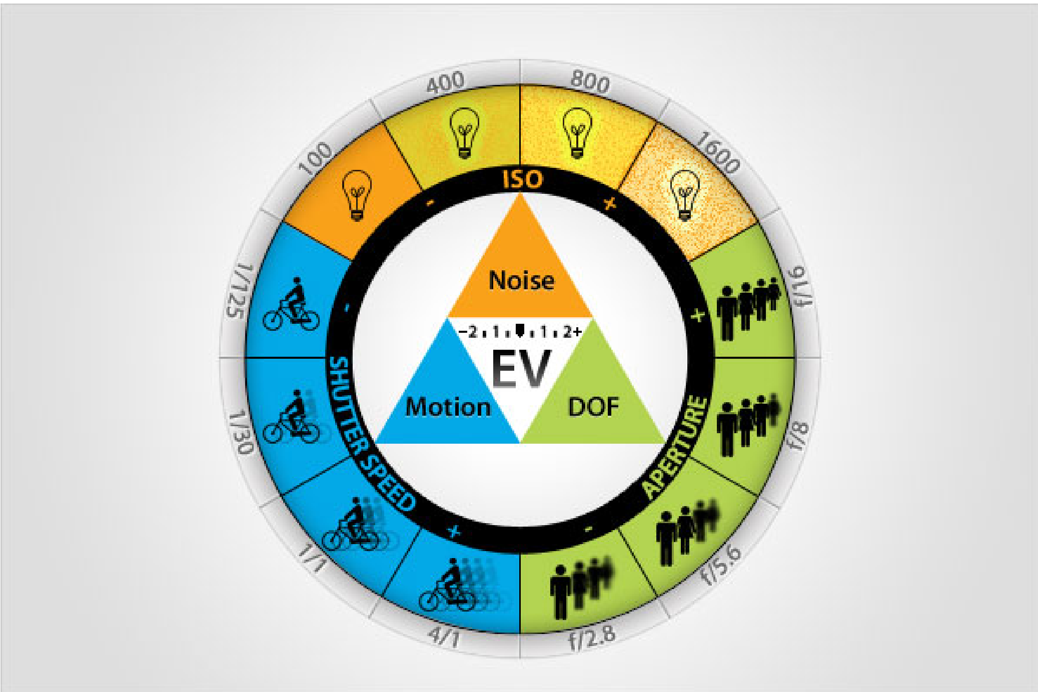


Figure 2.6: Interrelated exposure triangle

When these three elements are combined, they represent a given exposure value (EV) for a given setting. Any change in any one of the three elements will have a measurable and specific impact on how the remaining two elements react to expose the film frame or image sensor and how the image ultimately looks. For example, if you increase the f-stop, you decrease the size of the lens diaphragm thus reducing the amount of light hitting the image sensor, but also increasing the DOF(depth of field) in the final image. Reducing the shutter speed affects how motion is captured, in that this can cause the background or subject to become blurry. However, reducing shutter speed (keeping the shutter open longer) also increases the amount of light hitting the image sensor, so everything is brighter. Increasing the ISO, allows for shooting in lower light situations, but you increase the amount of digital noise inherent in the photo. It is impossible to make an independent change in one of the elements and not obtain an opposite effect in how the other elements affect the image, and ultimately change the EV.

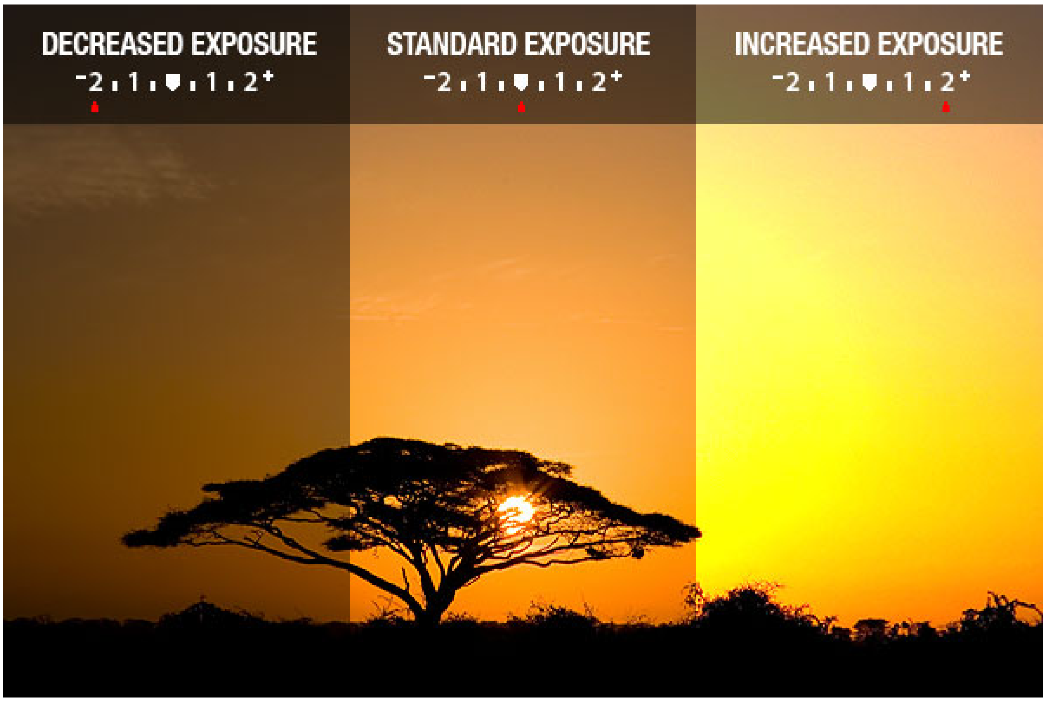


Figure 2.7: Effect of different exposure values to a captured image

Chapter 3

Image Parameters

3.1 Motion Blur

Motion blur is the apparent streaking of rapidly moving objects in a still image or a sequence of images such as a movie or animation. It results when the image being recorded changes during the recording of a single frame, either due to rapid movement or long exposure. When a camera creates an image, that image does not represent a single instant of time. Because of technological constraints or artistic requirements, the image may represent the scene over a period of time. Most often this exposure time is brief enough that the image captured by the camera appears to capture an instantaneous moment, but this is not always so, and a fast moving object or a longer exposure time may result in blurring artifacts which make this apparent. As objects in a scene move, an image of that scene must represent an integration of all positions of those objects, as well as the camera's viewpoint, over the period of exposure determined by the shutter speed. In such an image, any object moving with respect to the camera will look blurred or smeared along the direction of relative motion. This smearing may occur on an object that is moving or on a static background if the camera is moving. In a film or television image, this looks natural because the human eye behaves in much the same way.

Because the effect is caused by the relative motion between the camera, and the objects and scene, motion blur may be avoided by panning the camera to track those moving objects. In this case, even with long exposure times, the objects will appear sharper, and the background more blurred.

Figure 3.1: Motion blur

3.2 Contrast

Contrast is defined as the separation between the darkest and brightest areas of the image. Increase contrast and you increase the separation between dark and bright, making shadows darker and highlights brighter. Decrease contrast and you bring the shadows up and the highlights down to make them closer to one another. Adding contrast usually adds ”pop” and makes an image look more vibrant while decreasing contrast can make an image look duller. Here is an example where we add some contrast.

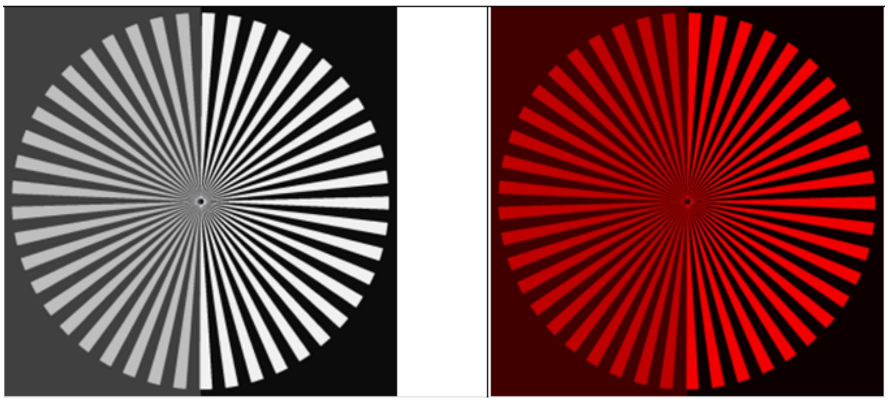


Figure 3.2: Distinction between two contrast values

In figure, we have added contrast to the right half of both images. As we can see, the white/red spokes have gotten brighter while the background has gotten darker. This causes the image to look more defined. By making the highlights brighter, however, we've also increased the brightness of the spokes, causing the image to appear brighter since the spokes are the main focus of the image. On the red image, increasing the brightness of the spokes has also increase saturation. Finally, sharpness has also been increased on both images. Here, we have increased brightness, contrast, saturation, and sharpness simply by adding contrast! Note that not all areas of the image will be affected equally and a lot depends on the content of the image itself. Saturation effects, for example, will be less noticeable in images that don't show bright colors because there is very little saturation to begin with. As an extreme example, take a look at the B/W image above. Since B/W images have zero saturation by definition, changing contrast cannot change saturation in B/W (gray) areas of image.

Chapter 4

Traditional Photography Vs. Computational Photography

Computational photography or computational imaging refers to digital image capture and processing techniques that use digital computation instead of optical processes. Computational photography can improve the capabilities of a camera, or introduce features that were not possible at all with film based photography, or reduce the cost or reduce the size of camera elements. Examples of computational photography include in-camera computation of digital panoramas, high-dynamic-range images, and light field cameras. Light field cameras use novel optical elements to capture three dimensional scene information which can then be used to produce 3D images, enhanced of depth-of-field, and selective de-focusing (or ”post focus”). Enhanced depth-of-field reduces the need for mechanical focusing systems. All of these features use computational imaging techniques.

In traditional film-like digital photography, camera images represent a view of the scene via a 2D array of pixels. Computational Photography attempts to understand and analyze a higher dimensional representation of the scene. Rays are the fundamental primitives. The camera optics encode the scene by bending the rays, the sensor samples the rays over time, and the final ’picture’ is decoded from these encoded samples. The lighting (scene illumination) follows a similar path from the source to the scene via optional spatio-temporal modulators and optics. In addition, the processing may adaptively control the parameters of the optics, sensor and illumination.

4.1 Encoding and Decoding

The encoding and decoding process differentiates Computational Photography from traditional 'film-like digital photography'. With film-like photography, the captured image is a 2D projection of the scene. Due to limited capabilities of the camera, the recorded image is a partial representation of the view. Nevertheless, the captured image is ready for human consumption: what you see is what you almost get in the photo. In Computational Photography, the goal is to achieve a potentially richer representation of the scene during the encoding process. In some cases, Computational Photography reduces to 'Epsilon Photography', where the scene is recorded via multiple images, each captured by epsilon variation of the camera parameters. For example, successive images (or neighboring pixels) may have a different exposure, focus, aperture, view, illumination, or instant of capture. Each setting allows recording of partial information about the scene and the final image is reconstructed from these multiple observations. In other cases, Computational Photography techniques lead to 'Coded Photography' where the recorded photos capture an encoded representation of the world. In some cases, the raw sensed photos may appear distorted or random to a human observer. But the corresponding decoding recovers valuable information about the scene.

1. Generalized Optics
2. Generalized Sensor
3. Processing
4. Generalized Illumination

The first three form the Computational Camera. Like other imaging fields, in addition to these geometry defining elements, Computational Photography deals with other dimensions such as time, wavelength and polarization.

4.2 The Traditional Camera

Over the last century, the evolution of the camera has been truly remarkable. However, through this evolution the basic model underlying the camera has remained essentially the same, namely, the camera obscura (Figure (a)). The traditional camera has a detector and a standard lens which only captures those principal rays that pass through its center of projection, or effective pinhole,

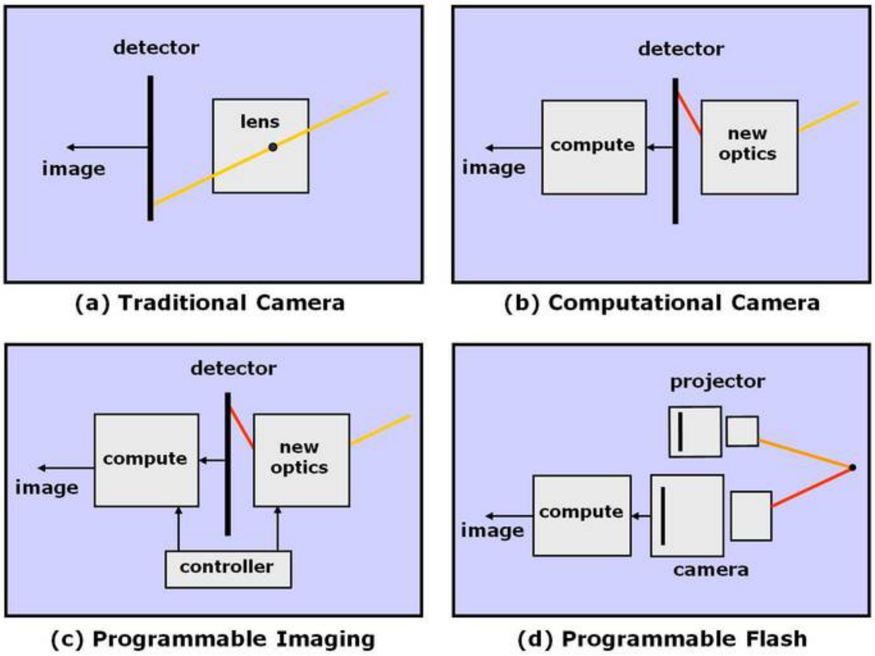


Figure 4.1: Types of digital cameras

to produce the familiar linear perspective image. In other words, the traditional camera performs a very simple and restrictive sampling of the complete set of rays, or the light field, that resides in any real scene

4.3 Computational Cameras

A computational camera (Figure (b)) uses a combination of novel optics and computations to produce the final image. The novel optics is used to map rays in the light field of the scene to pixels on the detector in some unconventional fashion. For instance, the ray shown in Figure (b) has been geometrically redirected by the optics to a different pixel from the one it would have arrived at in the case of a traditional camera. As illustrated by the change in color from yellow to red, the ray could also be photo-metrically altered by the optics. In all cases, the captured image is optically coded and may not be meaningful in its raw form. The computational module has a model of the optics, which it uses to decode the captured image to produce a new type of image that could benefit a vision system. The vision system could either be a human observing the image or a computer vision system that uses the image to interpret the scene it represents.

Chapter 5

Collection of training data samples

5.1 Camera Lab

To get training data samples with minimum human interface our team has developed a laboratory. It is consisting of mechanical components such as d. c. shaft motor, power screw, plastic fan. A cubical box is made up of aluminum links. Accurate dimensions of box are 45 cm * 45 cm * 95 cm.

We then fitted digital camera on a flat surface attached to power screw. To measure distance covered by a camera, we calculated distance covered per second and using timer control a prescribed distance traversing made easier. A hall effect IC (UC1881) is used to set initial point for calculating distances. Hall Effect IC toggles its current state on sensing a magnet. Hence UC1881 is used as a limiting switch.

Our project work includes study of camera and image parameters with change in intensity of light. We have worked on 3 different intensities. To measure intensity we have used 2 LDR. By taking mean of both, we arrived at output ranging from 0-255. LDR is interfaces with Arduino UNO R3 board. An analog to digital conversion is carried out in Arduino to quantify the output levels.

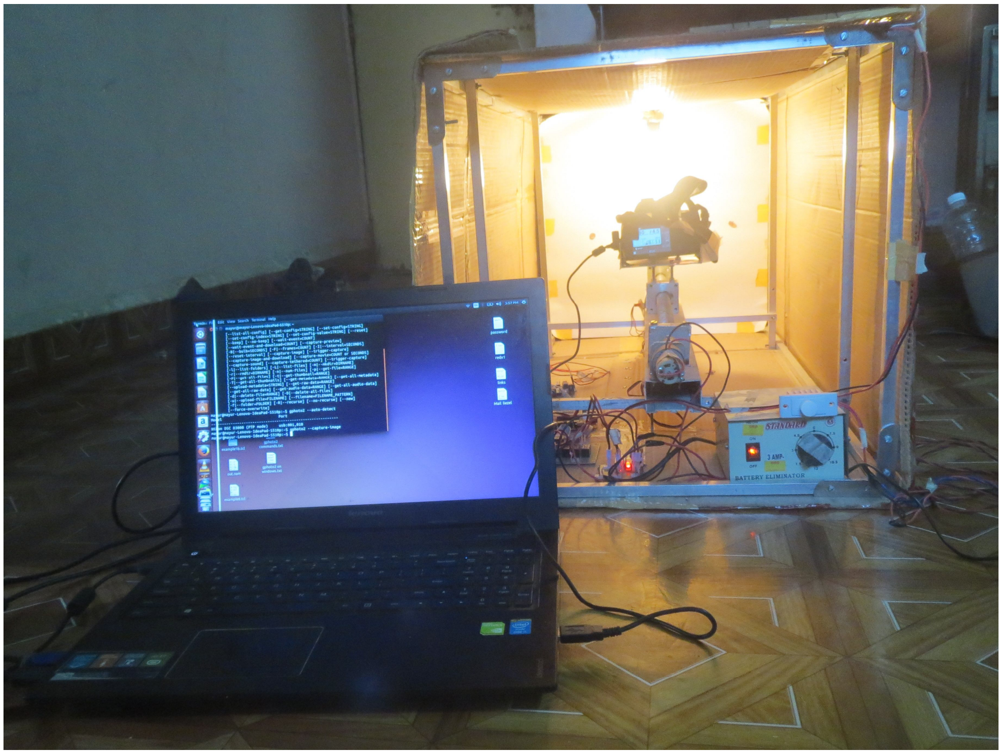


Figure 5.1: Lab front view

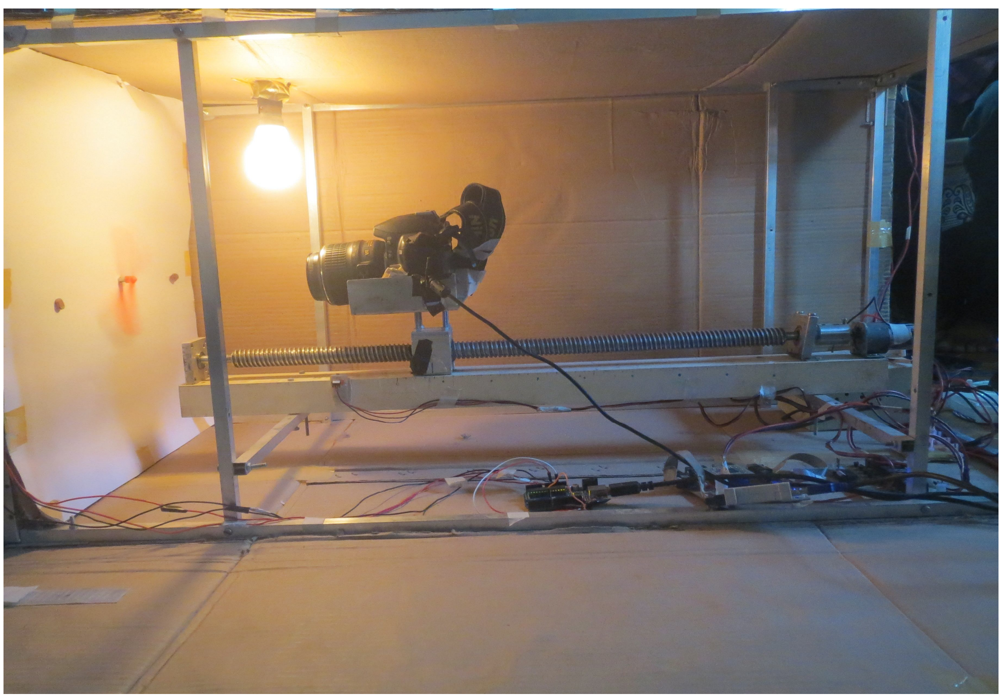


Figure 5.2: Lab side view

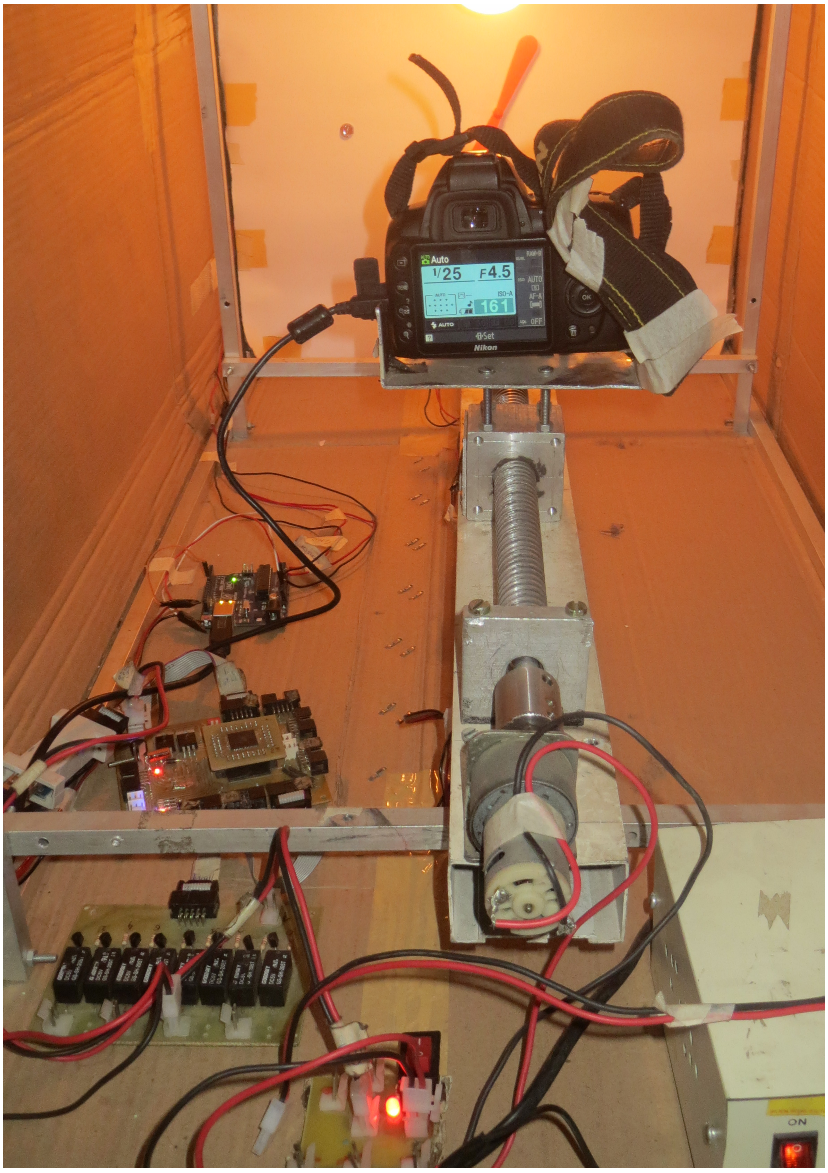


Figure 5.3: Lab with inclusion of components

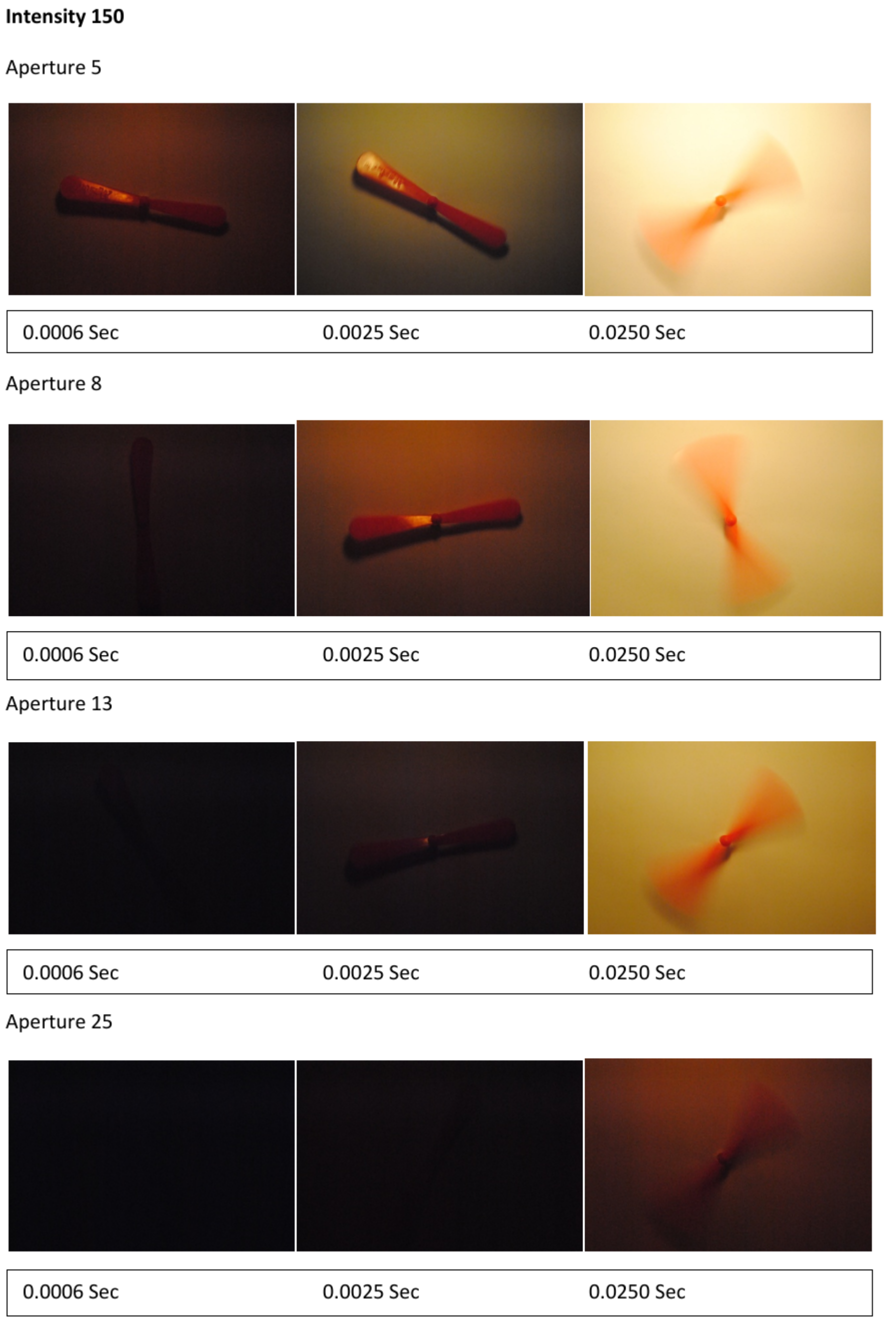


Figure 5.4: Data samples for intensity 150

Intensity 190

Aperture 5

0.0006 Sec	0.0025 Sec	0.0250 Sec

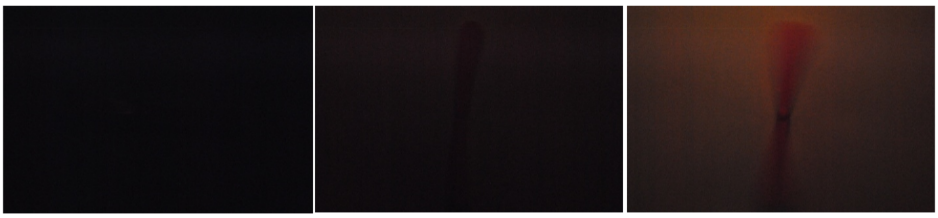
Aperture 8

0.0006 Sec	0.0025 Sec	0.0250 Sec

Aperture 13

0.0006 Sec	0.0025 Sec	0.0250 Sec

Aperture 25



0.0006 Sec	0.0025 Sec	0.0250 Sec

Figure 5.5: Data samples for intensity 190

Intensity 230

Aperture 5

0.0006 Sec	0.0025 Sec	0.0250 Sec

Aperture 8

0.0006 Sec	0.0025 Sec	0.0250 Sec

Aperture 13

0.0006 Sec	0.0025 Sec	0.0250 Sec

Aperture 25

0.0006 Sec	0.0025 Sec	0.0250 Sec

Figure 5.6: Data samples for intensity 230

Chapter 6

Algorithms to determine image parameters

Before attempting to process the image for any useful information, it is critical to ascertain that the image being processed is of good quality. Among several quality attributes including blur, noise, contrast and saturation, image blur deteriorates high frequency contents in the image thereby making the image unworthy for any useful information retrieval. Incorporating image blur detection and contrast measurement in a camera will help discard bad image data at the source itself. This problem is solved by

6.1 Blur content in image

1. Edge Sharpness Analysis Blur Detection in Digital Image

   Common edge sharpness analysis methods use the contrast edge of object in the image for blur analysis. Using the Harr wavelet transform, to detect blur and extend of blurring. Using perceptual-based no-reference objective image sharpness/blurriness metric by integrating the concept of just noticeable blur into a probability summation model. Using standard deviation of the edge gradient magnitude profile and the value of the edge gradient magnitude with weighted average.

2. Depth of Field (D.O.F.) Blur Detection in Digital Image

   Focusing on object detection in Object of Interest (OOI) technique in image by photographer. It even works on low Depth of Field.

3. Blind De-Convolution Blur Detection in Digital Image

   The blurred image is modeled as a convolution between the original image and an unknown point-spread function. The angle of motion blur is estimated using three different approaches. The first employs the cepstrum, the second a Gaussian filter, and the third the Radon transform. To estimate the extent of the motion blur, two different cepstral methods are employed. The accuracy of these methods is evaluated using artificially blurred images with varying degrees of noise added.

4. Wavelet-Based Histogram Blur Detection in Digital Image

   The fundamental principle behind this approach is examining the edge sharpness level to arrive at a decision. Wavelet theory has proved to be an excellent mathematical tool for the analysis of singularities like edges in images and subsequently for edge detection. The Step-structure edges are further classified into Astep-structure and Gstep-structure edges based on whether change of pixel intensity is gradual or abrupt.

5. Modified Blur detection algorithm using SVD

   The Haar algorithm usually results in a number of missed detections and false alarms because the blur decision is heavily dependent on the correct identification of edges. Image data with a smooth trend resulted in false alarms whereas images that were truly blurred but had sudden variation from high detail to low detail resulted in missed detection.

6. Directional Frequency Energy Blur Detection in Digital Image

   It is important to study the characteristics of these motion blurred images before attempting to estimate the PSF.

7. No reference blur estimation One can quantify the blur annoyance on a picture by blurring it and comparing the variations between neighboring pixels before and after the low-pass filtering step. Consequently, the first step consists in the computation of the intensity variations between neighboring pixels of the input image. On this same image, we apply a low-pass filter and compute also the variations between the neighboring pixels. Then, the comparison between these intensity variations allows us to evaluate the blur annoyance. Thus, a high variation between the original and the blurred image means that the original image was sharp whereas a slight variation between the original and image means that the original image was already blurred.

It is known that the sharpness of an image is contained in its gray component. This assumption, which is verified with subjective tests, justify that we estimate the blur annoyance only on the luminance component. The flow chart in Figure describes the steps of the algorithm description and refers to the following equations. Let F be the luminance component of an image or a video frame of size of m n pixels. To estimate the blur annoyance of F the first step consists in blurred it in order to obtain a blurred image B. We choose an horizontal and a vertical strong low-pass filter (1) to model the blur effect and to create BVer and BHor.

We need to analyze the variation of the neighboring pixels after the blurring step. If this variation is high, the initial image or frame was sharp whereas if the variation is slight, the initial image or frame was already blur.

Finally, we have to normalize the result in a defined range from 0 to 1.

We note that the variations between the two differences images DF and DB are always slighter than the values of the initial difference image DF. Then, we select the blur the more annoying among the vertical one and the horizontal one as the final blur value.

To summarize, we obtain a no-reference perceptual blur estimation blurF ranging from 0 to 1 which are respectively the best and the worst quality in term of blur perception.

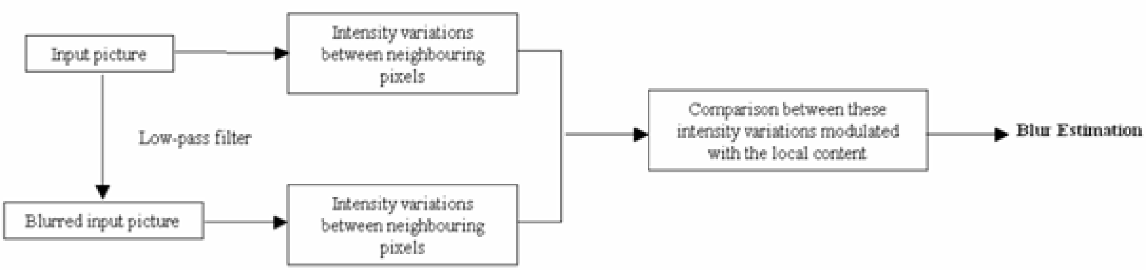


Figure 6.1: Flowchart for blur estimation

It is known that the sharpness of an image is contained in its gray component. This assumption, which is verified with subjective tests, justify that we estimate the blur annoyance only on the luminance component. The flow chart in Figure 5 describes the steps of the algorithm description and refers to the following equations. Let F be the luminance component of an image or a video frame of size of m n pixels. To estimate the blur annoyance of F the first step consists in blurred it in order to obtain a blurred image B.

We choose an horizontal and a vertical strong low-pass filter (1) to model the blur effect and to create BVer and BHor.

Then, in order to study the variations of the neighboring pixels, we compute the absolute difference images D

As we explain in the previous subsection, we need to analyze the variation of the neighboring pixels after the blurring step. If this variation is high, the initial image or frame was sharp whereas if the variation is slight, the initial image or frame was already blur. This variation is evaluated only on the absolute differences which have decreased:

Then, in order to compare the variations from the initial picture, we compute the sum of the coefficients of

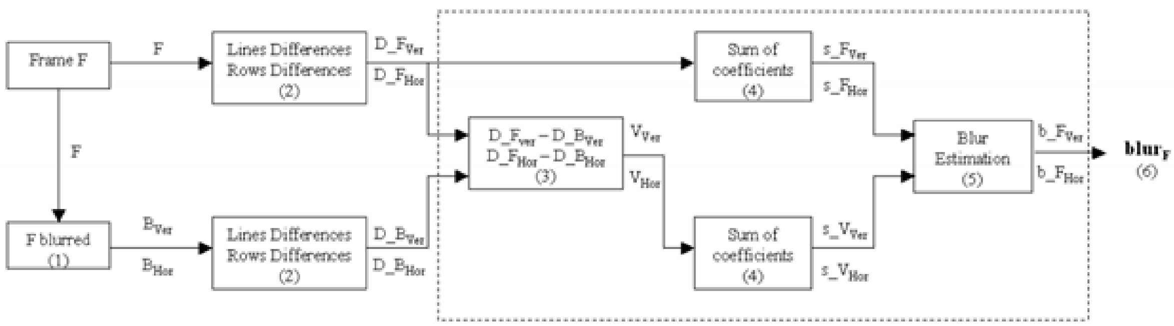


Figure 6.2: Flow chart for no reference blur matrix determination

6.2 Contrast of image

1. Reference image method

   Traditionally, the image quality of digital cameras has been measured using test targets. Test-target measurements are tedious and require a controlled laboratory environment. Algorithm metrics can be divided into three groups: full-reference (FR), reduced-reference (RR) and no reference (NR). FR metrics cannot be applied to the computation of image quality captured by digital cameras because pixel-wise reference images are missing. NR metrics are applicable only when the distortion type is known and the distortion space is low-dimensional. RR metrics provide a tradeoff between NR and FR metrics. An RR metric does not require a pixel-wise reference image; it only requires a set of extracted features. With the aid of RR features, it is possible to avoid problems related to NR metrics.

2. Using Discrete cosine transform

   This algorithm which works in DCT domain separates DCT coefficients into various bands and enhances them using a scaling factor. This technique is based on the measurement of contrast. Contrast is defined as the ratio of low frequency contents to high frequency contents of an image. Algorithm reported works for Gray level image but not for colour images.

6.3 Energy content of Image

Image entropy as used in datasets is calculated with the same formula used by the Galileo Imaging Team:

$$Entropy = -\sum(P_i \log_2 P_i) \tag{6.1}$$

In the above expression, P i is the probability that the difference between 2 adjacent pixels is equal to i, and Log 2 is the base 2 logarithm.

6.4 RGB normalizing of image

1. Histogram stretching

   Contrast is a measure of how much the pixel brightness changes relative to the average brightness. A technique known as histogram stretching can be used to shift the pixel values to fill the entire brightness range, resulting in high contrast. The first step is to find the pixel values that should get mapped to 0 and 100 brightness. Any real-world image has noise, however. To keep the noise from unduly influencing the stretching, an assumption is made: a small percentage of the brightest and darkest pixels are ignored, writing them off to sensor noise. In MATLAB this feature is applied to image using function imadjust. A color space that better represents the human visual system, like L*a*b* or Luv can provide more natural stretching in some cases. In both of these color spaces, the L channel represents the brightness, while the (a*, b*) or (u, v) channels represent the color. The stretching is performed on the L channel after converting the colors.

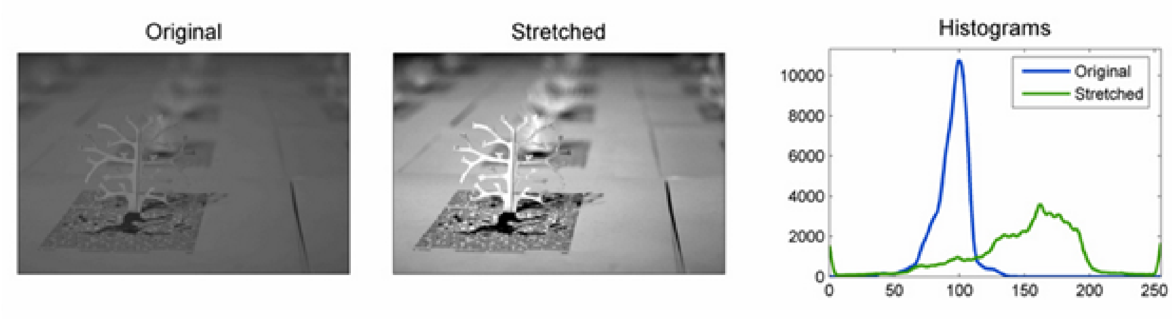


Figure 6.3: Comparison of original image and stretched histogram

2. If you plotted the CDF of some of your image histograms, you may have noticed that the CDF does not form a straight linemeaning that the pixel values are not equally likely to occur (since the CDF is the integral of the PDF). The good news is that most natural images do not have flat CDFs. That said, some industrial applications can benefit from having a flat CDF. The process of flattening the CDF is called histogram equalization.

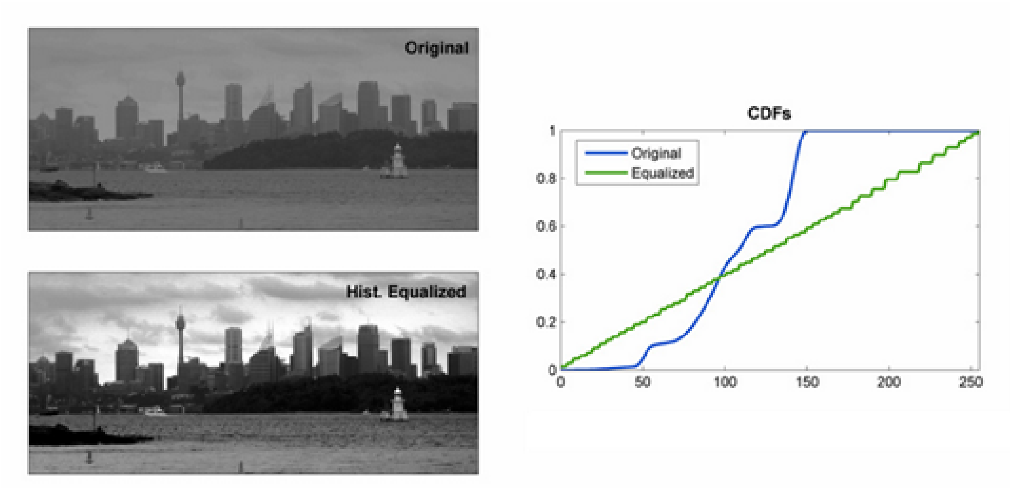


Figure 6.4: Comparison of original and histogram equalized image

Chapter 7

Proposed Algorithm

7.1 The Proposed algorithm for suggestion generation for user

1. The camera will take a photo on predefined camera parameters aperture and shutter speed.
2. Now code will process the image and will check for constraints in order to enter into clusters differed according to brightness present in the environment. The constraints will be predefined as the camera response will be known.
3. The constraints will be applied on the values of image parameters- blur content, contrast measure and energy content of image (entropy).
4. Once the code has entered into the cluster, the instructions of capturing photos at predefined higher possible and lower possible combination of camera parameters will be given to camera.
5. Then the photos taken will be processed to get limits of entropy of image. As the aim of application is to reduce blur content in the image, camera will try to reduce blur by reducing the shutter speed value. But this will eventually make images darker.
6. If the image taken by camera is dark enough that no one will be able to see the object captured by the camera. So entropy limit will help to get good possible image provided that the image will be visible.
7. Also in order to restore the brightness in the image if camera changed parameter to capture the photos then the images may get brighter so that the object whose image is captured is not visible.

8. So creating a constraint on entropy of image so that it will be always in visible range.
9. Then user will be asked to capture the photo at whatever camera parameters he may like.
10. The captured image will be transferred for processing so that it will determine aperture at which the image was captured.
11. And based on the determined aperture an adaptive equation stating the relation between shutter speed and blur, shutter speed and contrast will be generated. And using those equations shutter speed will be determined at which the photo was captured.
12. Then these calculated aperture and shutter speed values will be put into another set of equations which will try to increase contrast of image.
13. Final suggestion will be generated based on the output of above.

7.2 Adaptive behavioral entity of camera parameters

The core of the algorithm proposed above is represented by an entity which consists of behaviors of inter-relating all camera parameters to internal and external factors. Weve hypothesized this concept of making an entity which will be consists of all basic relations and equations, will be almost same or adaptive. As the entity is going to be same for all cameras one will not have to actually formulate the camera parameter effect on the captured image. Adaptive behavioral entity will run its predefined scripts to determine all related image parameters such as blur, contrast, noise and resolution. So it has to be entity consists of both internal and external factors.
We can write,
Entity = $f(int_i ext_i)$
i varies from 0 to n
n=number of camera parameters
int- vector represents all internal parameters
ext- vector represents all external parameters

ext=(bx, cx, Ix)
int=(A, S, ISO)

bx is blur value of captured image

cx is contrast of captured image

Ix is predefined intensity clusters

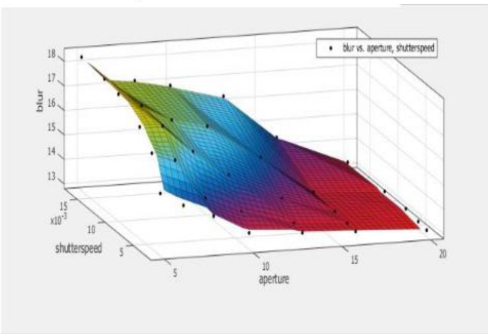
A is aperture of camera

S is shutter speed of camera

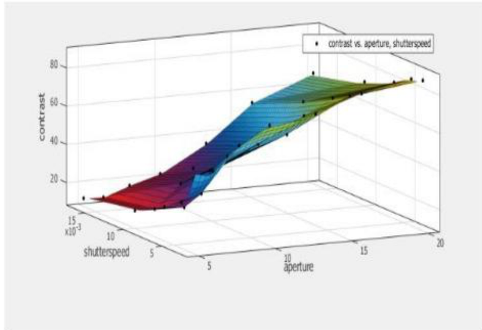
ISO is ISO value of camera

As this behavioral entity is specific to camera Nikon DSC3000 so the features of the entity will be limited will change according to intensity of environment light. This common basis developed for all cameras will help the camera to adapt itself for many operations such as autofocus and many more. All features graphs for camera Nikon DSC3000 are as follows:

For Intensity value= 150

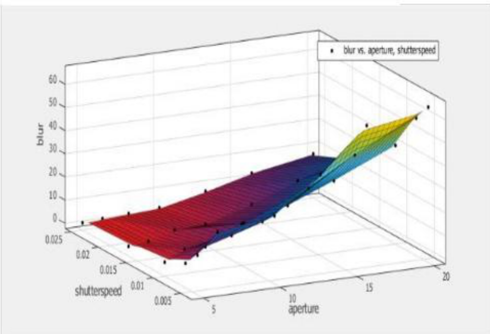


For Blur dependency on aperture and shutter speed

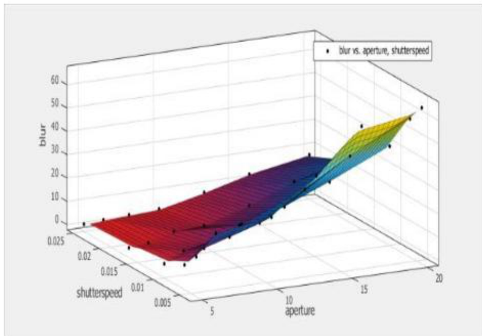


For contrast dependency on aperture and shutter speed

For Intensity value= 190

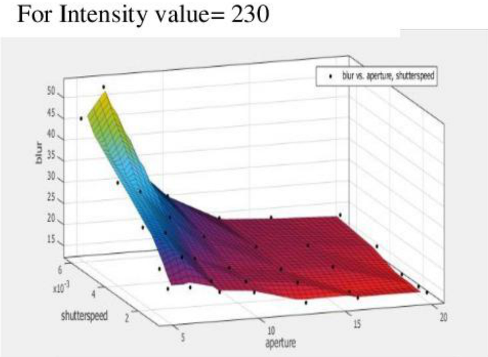


For Blur dependency on aperture and shutter speed

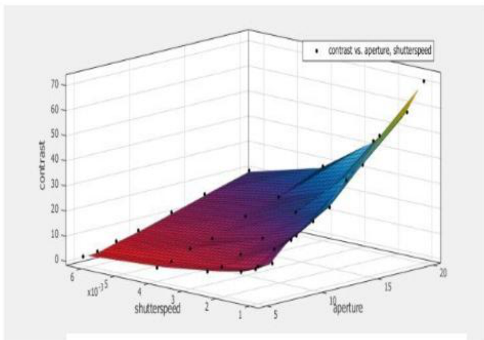


For contrast dependency on aperture and shutter speed

For Intensity value= 230



For Blur dependency on aperture and shutter speed



For Contrast dependency on aperture and shutter speed

Figure 7.1: Feature classification of specific camera at various intensities

Chapter 8

Remote tethering of Digital Camera

Remote tethering of DSLR camera was an important part while taking the image database in camera lab. If we take all photos manually then this may introduce handshakes in image or there may be some orientation change of an object whose picture we are taking as comparison. Also it was not possible to take photos manually inside camera lab. So it was decided to use remote tethering for camera control and capturing photos. Initially team tried to search for readymade softwares such as digicamcontrol, qDSLR dashboard, entangle and many more. But many of them had either compatibility issues or limited features. So decision of making our own tether which will satisfy our requirements using gphoto2 library.

Here is given the procedure to use gphoto library on Ubuntu (linux) to capture photos at defined aperture and shutter speed value. At the outset, install gphoto2 library on Ubuntu using the terminal.

1. Detecting the camera

   $gphoto2 –auto-detect command will detect the camera connected and will show at which port it is connected.(refer fig. 8.1)

2. Capturing the photos

   $ gphoto2 -I 2 -F 2 –capture-image-and-download

   The -I parameter sets the time interval in seconds until the next shot. The -F option sets the number of frames. This command therefore takes two photos every two seconds. The –capture-image-and-download statement completes the action and ensures that gPhoto2 stores the image in the target directory.(refer fig. 8.2)

3. Changing the camera parameters- ISO

   $gphoto2 –get-config /main/imgsettings/iso command will give current value of iso of camera and also the choice available to select. SO the user may select any ISO choice he may like.(refer fig. 8.3)

4. Changing the camera parameters- Aperture

   $gphoto2 –get-config /main/capturesettings/f-number command will give current value of aperture of camera and also the choice available to select. As this parameter is related to capture process of photo, it comes under capture settings. This gives around 16 choices of choosing aperture supported by specific camera.(refer fig. 8.4)

5. Changing the value of aperture

   $gphoto2 –set-config /main/capturesettings/f-number=7 this command will help user to select the aperture of f-number shown at choice number 7.(refer fig. 8.5 and 8.6)

6. Changing the camera parameters- Shutter speed

   $gphoto2 –get-config /main/capturesettings/shutterspeed this command will work same as that for aperture. It will give around 52 choices for choosing shutter speed for specific camera.(refer fig. 8.7)

7. Changing the value of shutter speed

   $gphoto2 –set-config /main/capturesettings/shutterspeed =41 this command will help user to select the shutter speed of camera shown at choice number 41. In snapshots of camera screen u can see that change made that shutter speed of 1/200 changed to 3 seconds.(refer fig. 8.8 and 8.9)

8. Getting image file from camera to pc

   $gphoto2 –list-files this command will give a list of all files contained within storage of camera(refer fig. 8.10)

   $gphoto2 –get-file 919 this will detect the files number 919 stored in camera and will download it to predefined path and following snapshots show that file at number 919 was selected and downloaded in home folder. Using all these procedures the dataset for this project was created.(refer fig. 8.11 and 8.12)

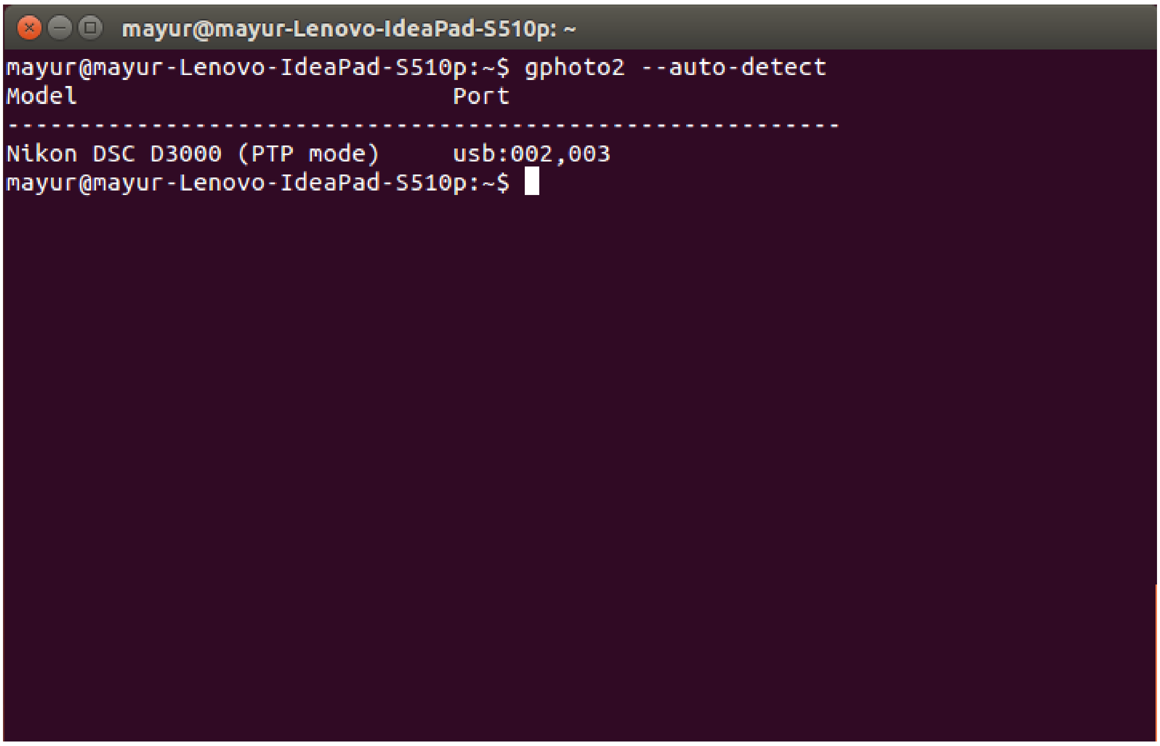


Figure 8.1: Command for detecting the camera

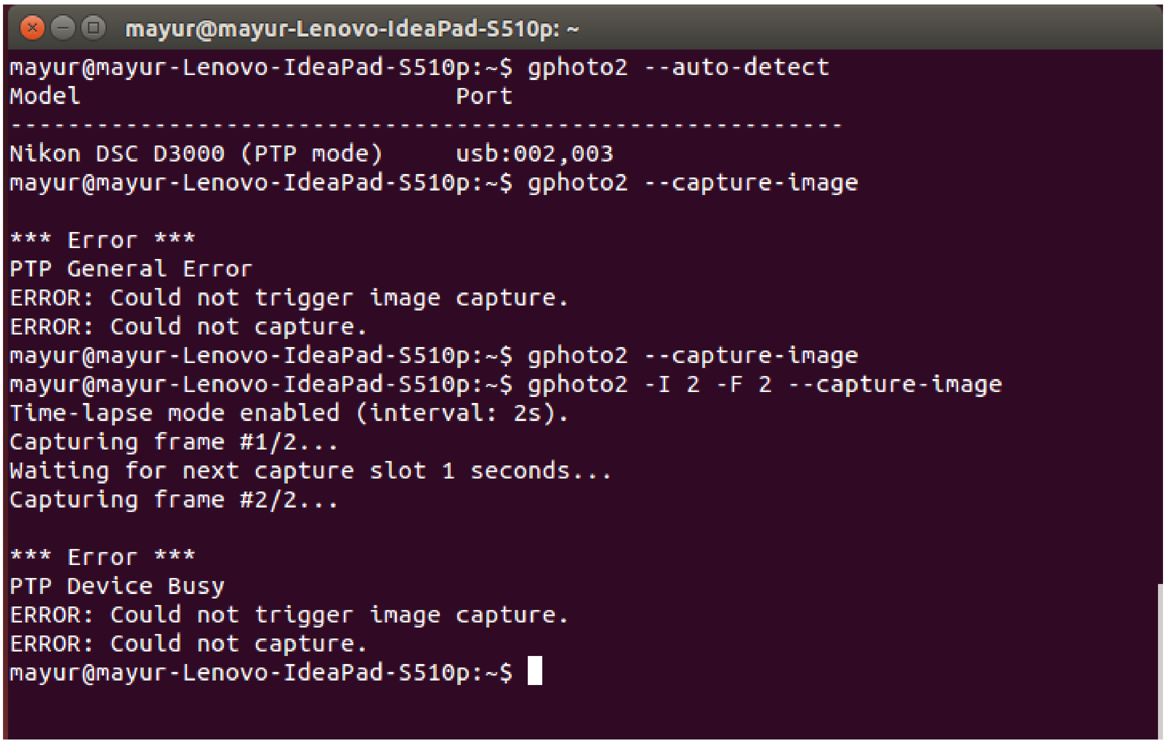


Figure 8.2: Capturing the photo

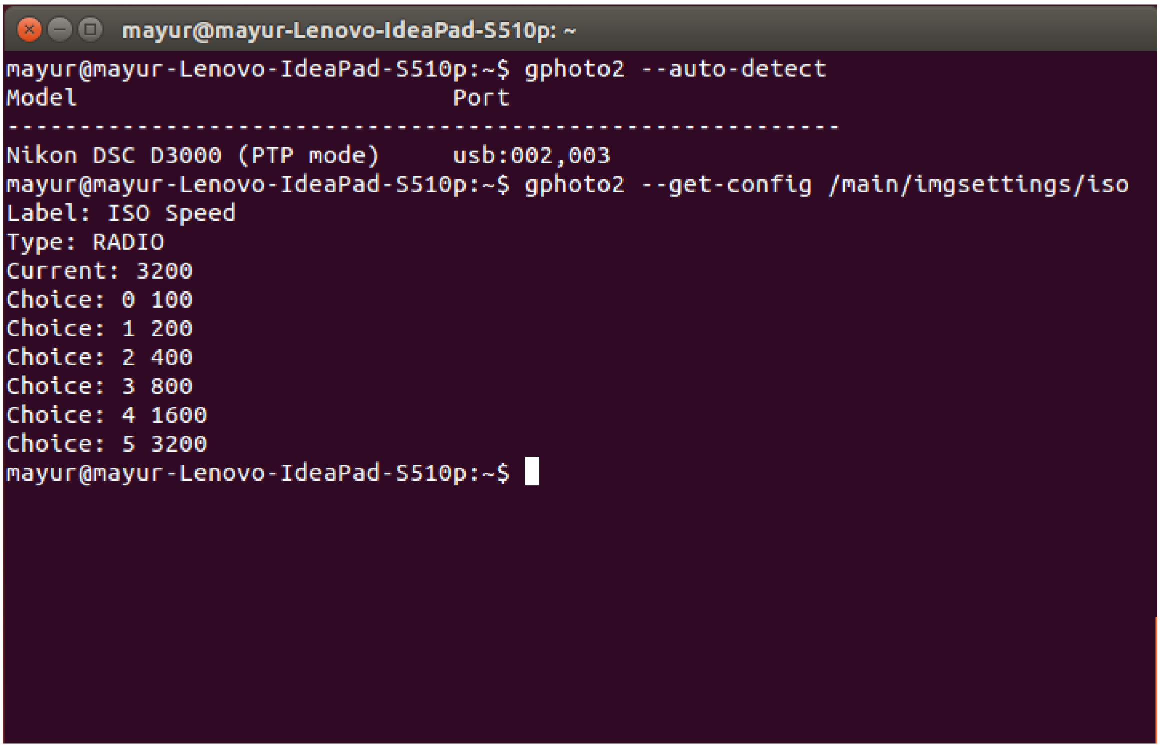


Figure 8.3: Changing the camera parameter - ISO

```
mayur@mayur-Lenovo-IdeaPad-S510p: ~
mayur@mayur-Lenovo-IdeaPad-S510p:~$ gphoto2 --get-config /main/capturesettings/f-n
umber
Label: F-Number
Type: RADIO
Current: f/4
Choice: 0 f/4
Choice: 1 f/4.5
Choice: 2 f/5
Choice: 3 f/5.6
Choice: 4 f/6.3
Choice: 5 f/7.1
Choice: 6 f/8
Choice: 7 f/9
Choice: 8 f/10
Choice: 9 f/11
Choice: 10 f/13
Choice: 11 f/14
Choice: 12 f/16
Choice: 13 f/18
Choice: 14 f/20
Choice: 15 f/22
Choice: 16 f/25
mayur@mayur-Lenovo-IdeaPad-S510p:~$
```

Figure 8.4: Changing the camera parameter - Aperture

```
mayur@mayur-Lenovo-IdeaPad-S510p: ~
mayur@mayur-Lenovo-IdeaPad-S510p:~$ gphoto2 --get-config /main/capturesettings/f-n
umber
Label: F-Number
Type: RADIO
Current: f/4
Choice: 0 f/4
Choice: 1 f/4.5
Choice: 2 f/5
Choice: 3 f/5.6
Choice: 4 f/6.3
Choice: 5 f/7.1
Choice: 6 f/8
Choice: 7 f/9
Choice: 8 f/10
Choice: 9 f/11
Choice: 10 f/13
Choice: 11 f/14
Choice: 12 f/16
Choice: 13 f/18
Choice: 14 f/20
Choice: 15 f/22
Choice: 16 f/25
mayur@mayur-Lenovo-IdeaPad-S510p:~$ gphoto2 --set-config /main/capturesettings/f-n
umber=7
mayur@mayur-Lenovo-IdeaPad-S510p:~$
```

Figure 8.5: Changing the value of Aperture size

Figure 8.6: Result of change in aperture size

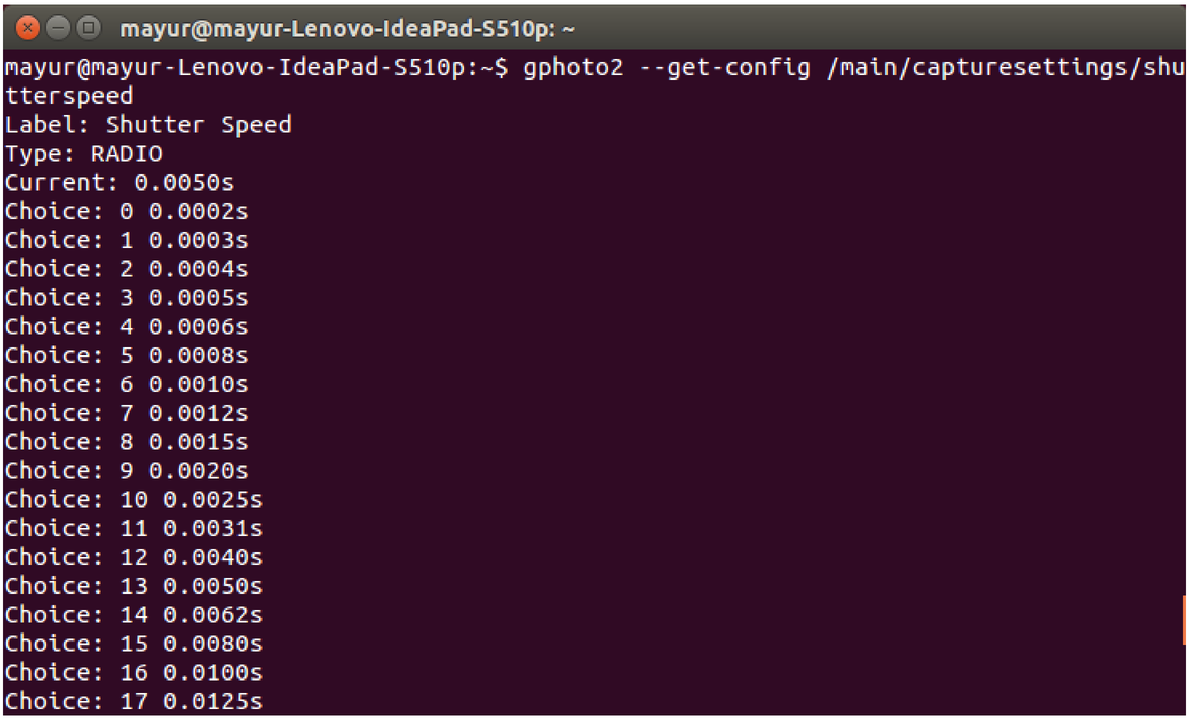


Figure 8.7: Change of the camera parameter - Shutter Speed

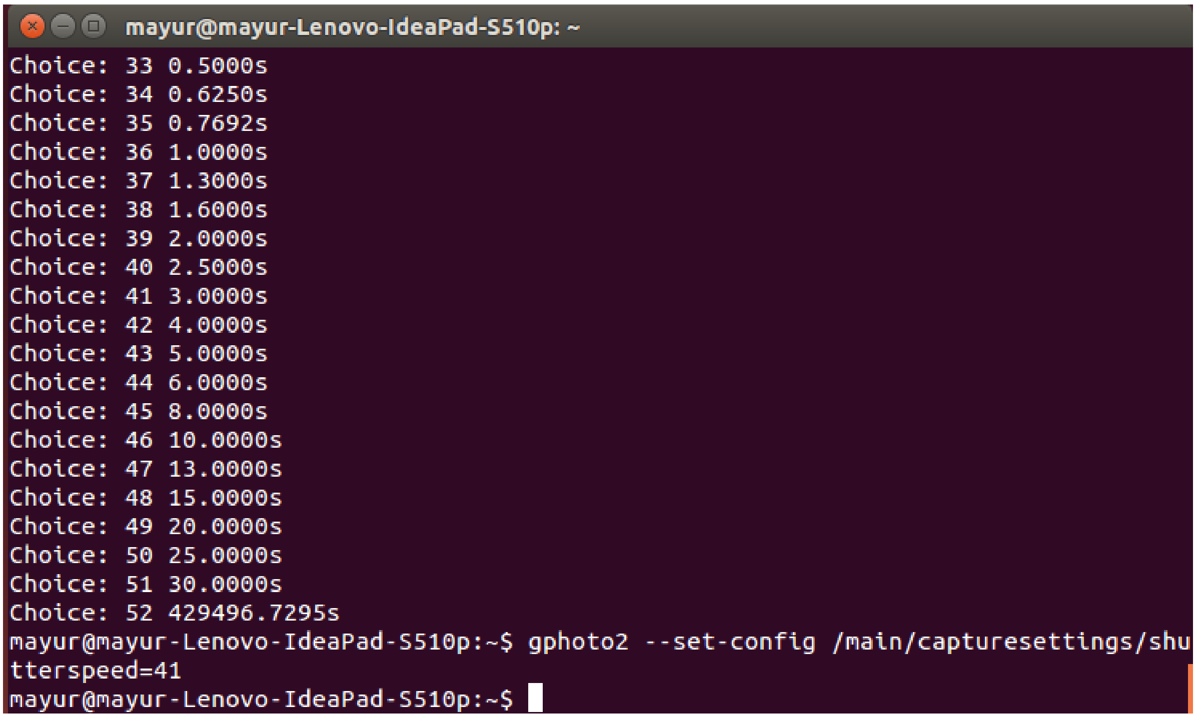


Figure 8.8: Change of the value of shutter speed

Figure 8.9: Result of change in the shutter speed value

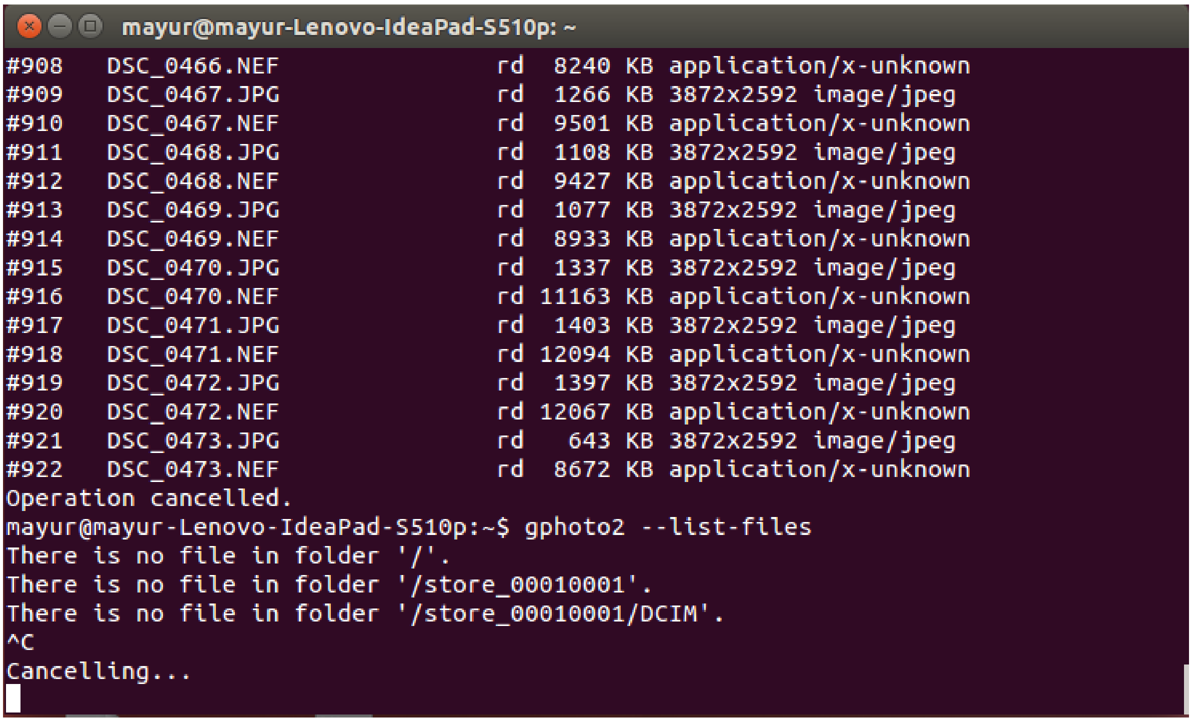


Figure 8.10: Listing of files in memory

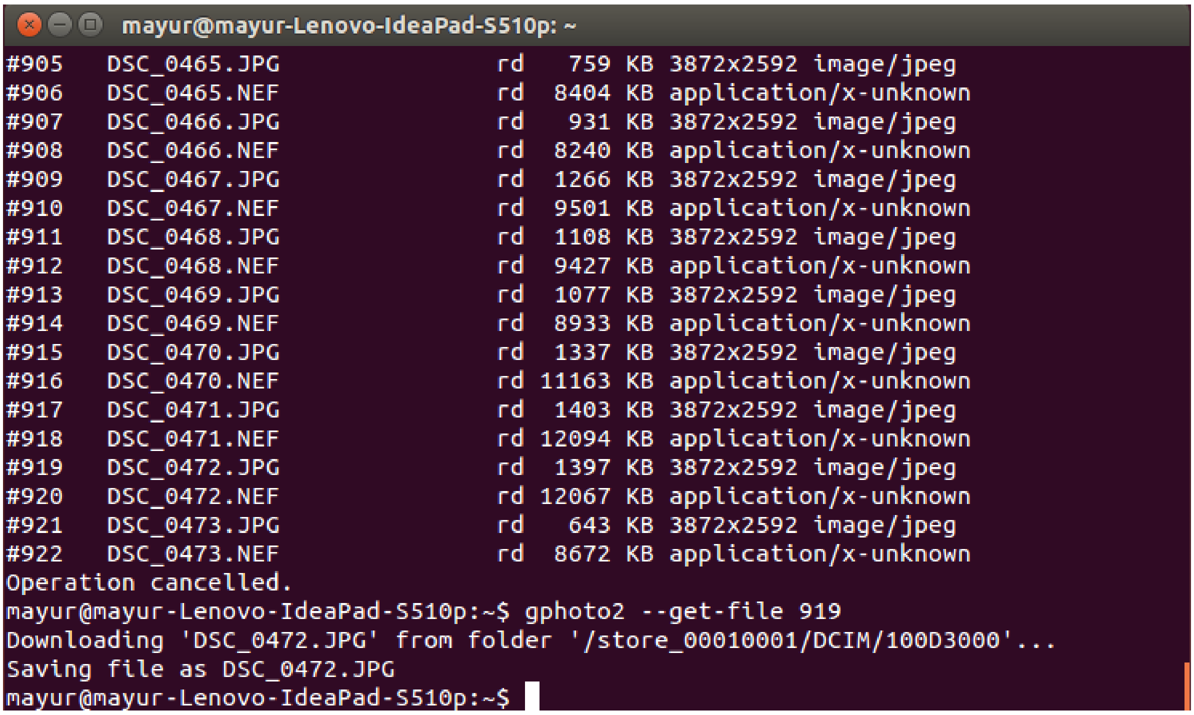


Figure 8.11: Fetching image file from the camera to PC

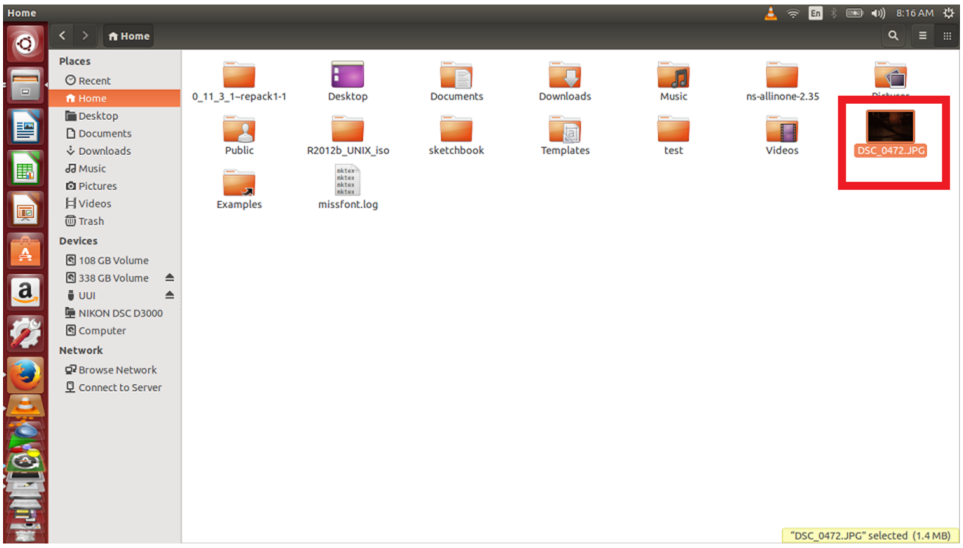


Figure 8.12: Output of fetched image in PC

Chapter 9

Curve Fitting

Our algorithm quantifies blur in the captured image. To get the shutter speed for which the quantified blur value is present in training data samples, we have used curve fitting tool. This tool helps to plot 2 dimentional curves and returns equations to respective curve.We have used cubical polynomial and 2nd order sum of sin curves to obtain coeffiecients. To collect data samples, we have kept a constant aperture and varying shutter speed we could obtain blur, contrast and entropy for training data.

9.0.1 Parameter Table

Following table represents coefficients of cubic polynomial for shutter speed vs. blur at Intensity 230

Table 9.1: Coefficients for cubic polynomial

Aperture	P1	P2	P3	P4
5	5.011e-009	1.955e-006	1.955e-006	-0.001221
6.3	3.381e-007	-3.32e-005	0.001193	-0.01229
8	2.73e-008	-5.116e-006	0.0005474	-0.007489
10	-2.088e-006	0.0001529	-0.00303	0.01866
13	-6.394e-005	0.003657	-0.06733	0.4026
16	0.0004518	-0.0221	0.3593	-1.94
20	0.0002719	-0.01317	0.2129	-1.146

9.0.2 Parameter Table

Following table represents coefficients of 2nd degree sum of sin for shutter speed vs. blur at Intensity 230

Table 9.2: Coefficients for sum of sin

Aperture	A1	B1	C1	A2	B2	C2
5	0.01156	0.06881	-1.653	0.005964	0.1077	0.267
6.3	0.01072	0.07832	-1.806	0.005519	0.1175	0.1176
8	0.01051	0.0346	-0.5463	0.0001863	0.4665	-2.145
10	0.01189	0.1304	-2.198	0.004773	0.2608	-1.562
13	0.01099	0.3299	1.176	0.005796	0.6597	-1.086
16	0.004949	0.7122	1.872	0.003603	2.849	-1.434
20	0.009389	0.2378	9.064	0.002774	2.854	-0.8747

9.0.3 Parameter Table

Following table represents coefficients of cubic polynomial for shutter speed vs. blur at Intensity 190

Table 9.3: Coefficients for cubic polynomial

Aperture	P1	P2	P3	P4
5	6.47e-006	-0.0003627	0.007395	-0.05066
6.3	4.598e-005	-0.002089	0.03222	-0.1668
8	4.84e-005	-0.002182	0.03354	-0.1735
10	-1.418e-005	0.001044	-0.02029	0.1196
13	0.0001521	-0.0064	0.09129	-0.4392
16	0.0008641	-0.03822	0.5651	-2.788
20	0.001464	-0.06393	0.9321	-4.532

9.0.4 Parameter Table

Following table represents coefficients of 2nd degree sum of sin for shutter speed vs. blur at Intensity 190

Table 9.4: Coefficients for sum of sin

Aperture	A1	B1	C1	A2	B2	C2
5	0.01981	0.1633	3.457	0.009688	0.2558	4.756
6.3	0.0114	0.4121	0.2082	0.006676	0.6387	6.034
8	0.03081	0.2977	1.71	0.01965	0.4518	2.353
10	0.01987	0.2237	3.097	0.005716	0.6009	0.4911
13	-0.4392	0.5542	4.726	0.005687	1.198	4.5
16	0.03799	0.6143	3.792	0.02468	0.8836	9.199
20	0.01935	1.004	-1.042	0.007764	4.122	-2.769

9.0.5 Parameter Table

Following table represents coefficients of cubic polynomial for shutter speed vs. blur at Intensity 150

Table 9.5: Coefficients for cubic polynomial

Aperture	P1	P2	P3	P4
5	0.0005462	-0.02538	0.3932	-2.027
6.3	0.0005887	-0.02556	0.3689	-1.769
8	-7.644e-05	0.008139	-0.1873	1.235
10	0.002607	-0.1147	1.679	-8.175
13	0.001995	-0.08456	1.196	-5.642
16	0.005362	-0.2302	3.299	-15.76
20	0.007005	-0.3021	4.35	-20.87

9.0.6 Parameter Table

Following table represents coefficients of 2nd degree sum of sin for shutter speed vs. blur at Intensity 150

Table 9.6: Coefficients for sum of sin

Aperture	A1	B1	C1	A2	B2	C2
5	0.1095	0.1824	3.481	0.0394	0.5535	7.117
6.3	0.06684	0.552	-2.106	0.05243	0.8451	2.796
8	0.1132	0.6729	-3.726	0.0964	0.941	1.715
10	0.1534	0.3044	8.145	0.07323	0.7189	11.5
13	0.1151	0.6549	3.142	0.08603	0.9897	7.67
16	0.0769	0.2814	8.714	0.01624	3.995	-0.9768
20	0.138	0.183	10.16	0.008378	3.729	-2.932

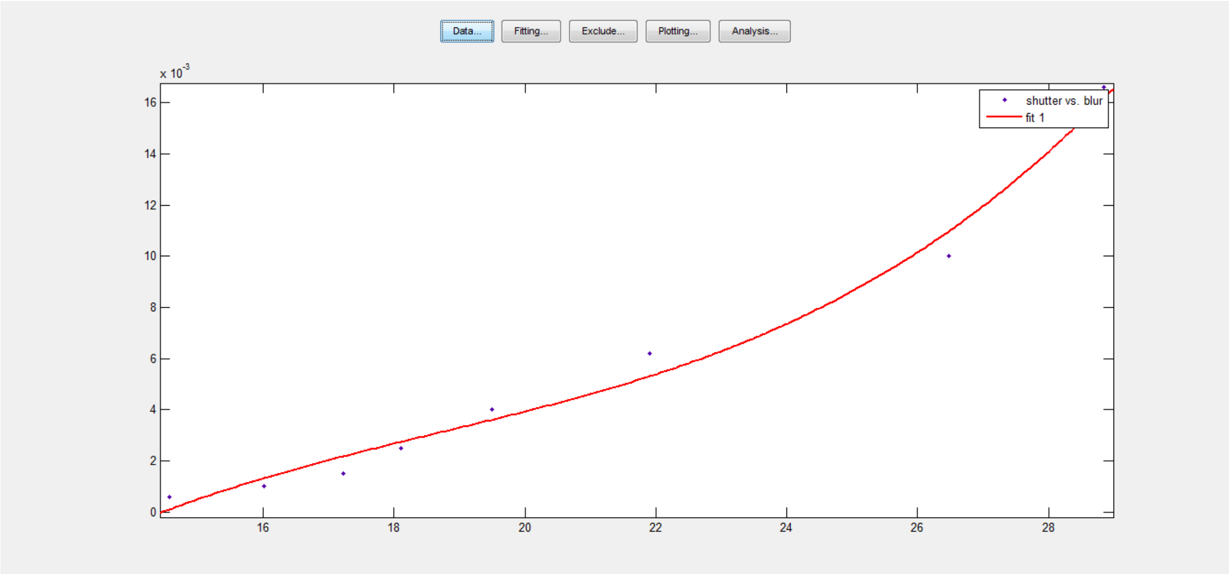


Figure 9.1: Cubic polynimial curve for Shutter vs. Blur at F=5

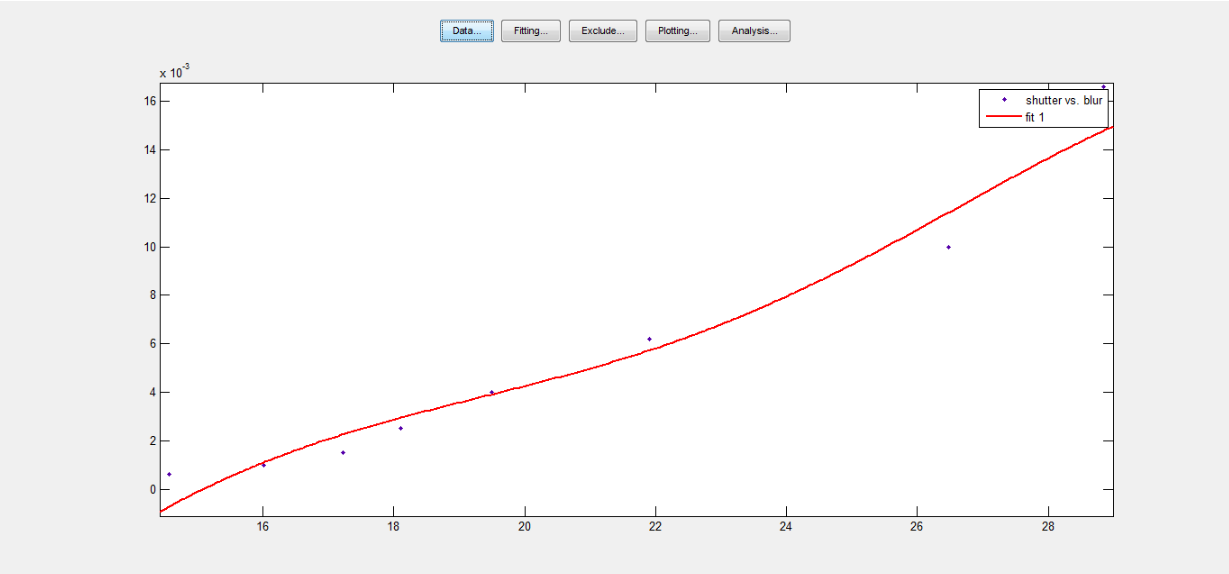


Figure 9.2: Sum of sin curve for Shutter vs. Blur at F=5

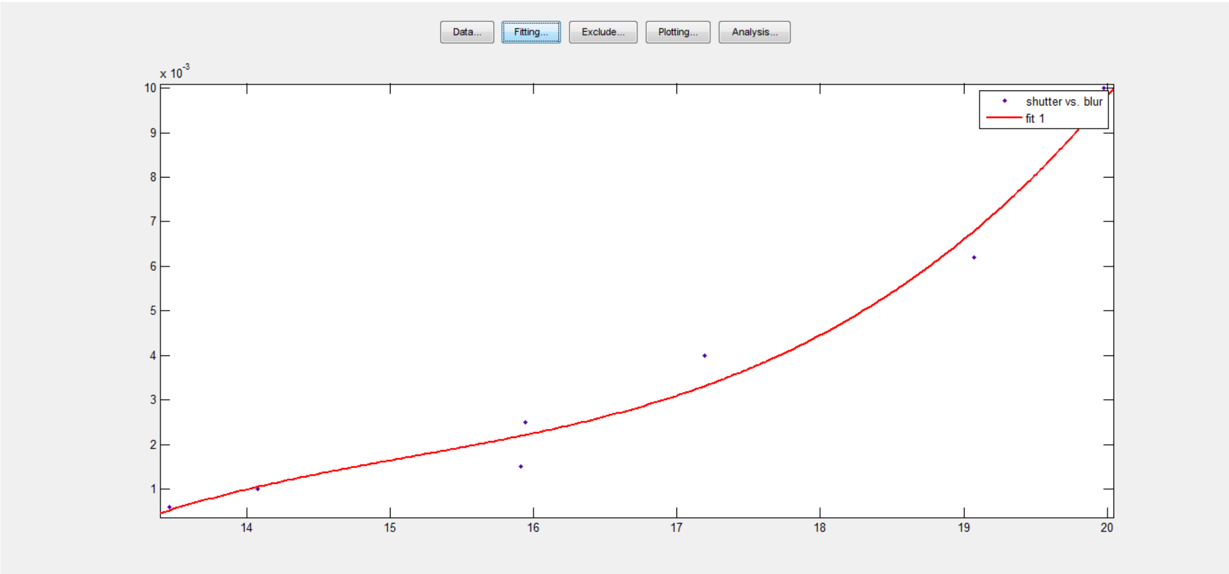


Figure 9.3: Cubic polynimial curve for Shutter vs. Blur at F=6.3

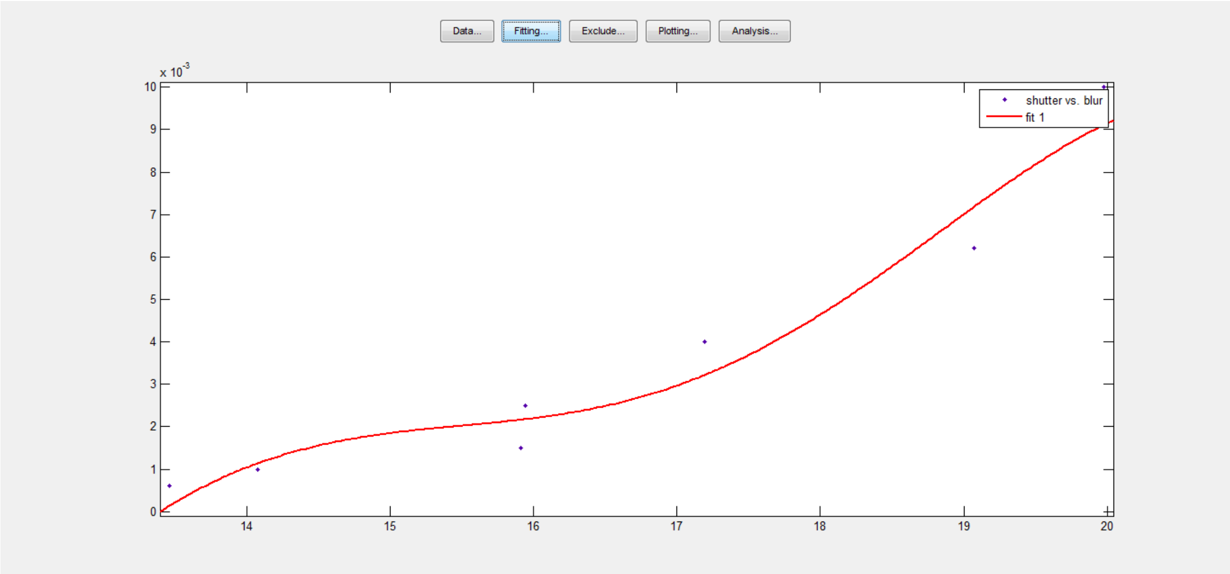


Figure 9.4: Sum of sin curve for Shutter vs. Blur at F=6.3

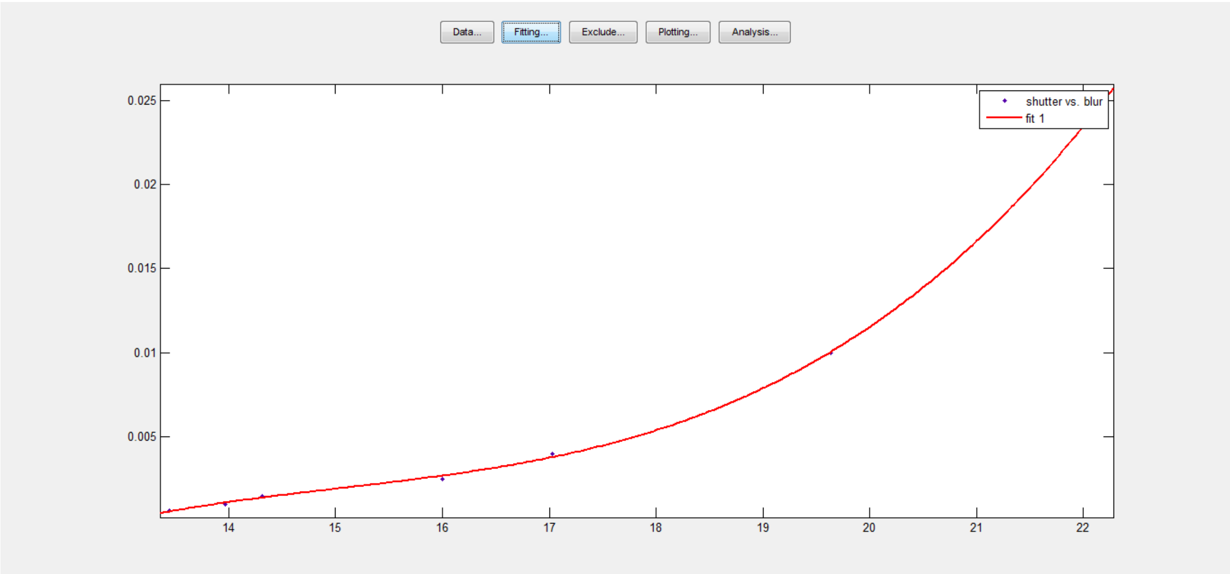


Figure 9.5: Cubic polynimial curve for Shutter vs. Blur at F=8

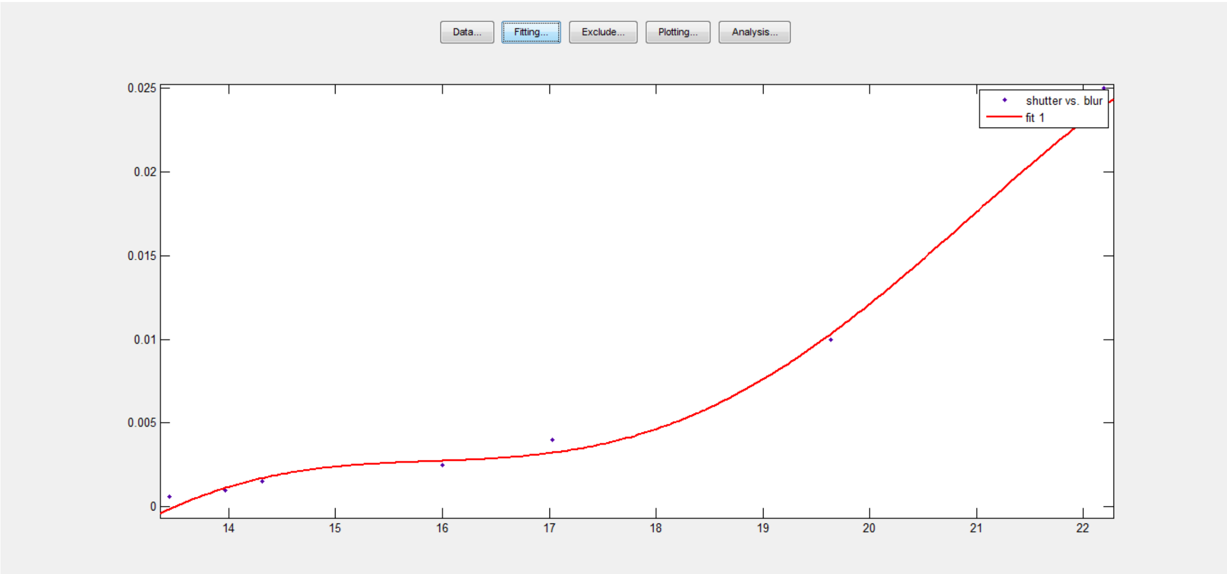


Figure 9.6: Sum of sin curve for Shutter vs. Blur at F=8

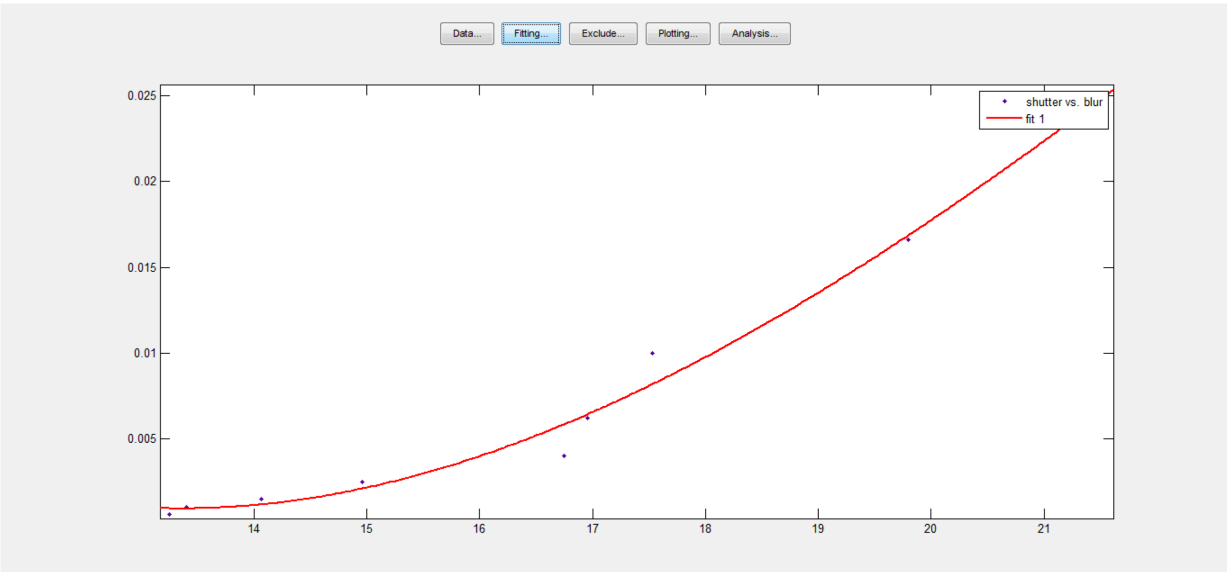


Figure 9.7: Cubic polynimial curve for Shutter vs. Blur at F=10

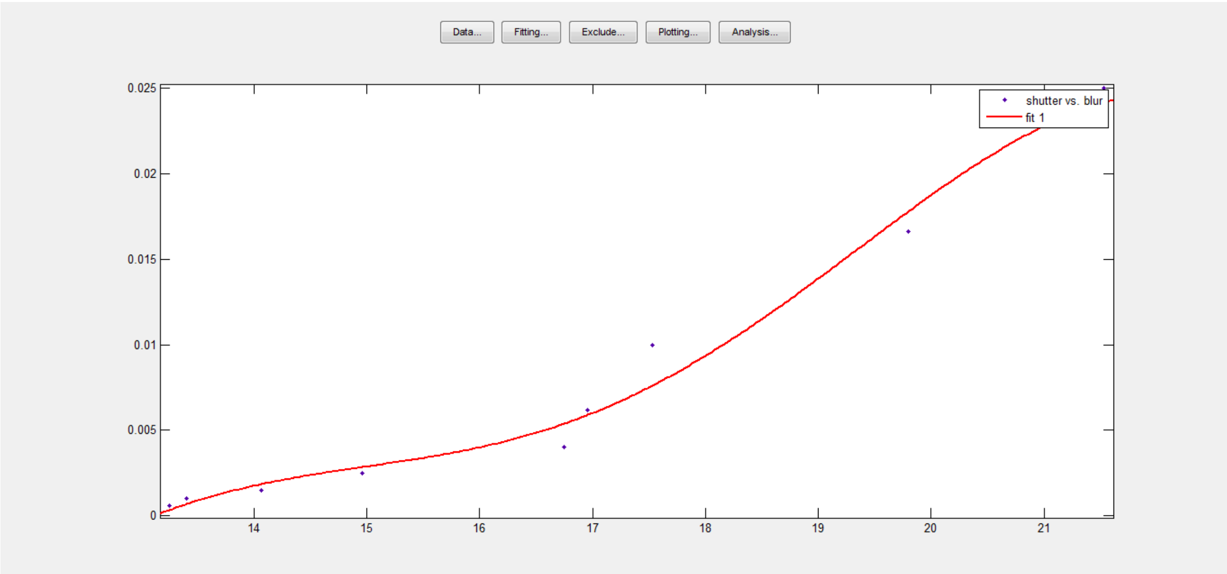


Figure 9.8: Sum of sin curve for Shutter vs. Blur at F=10

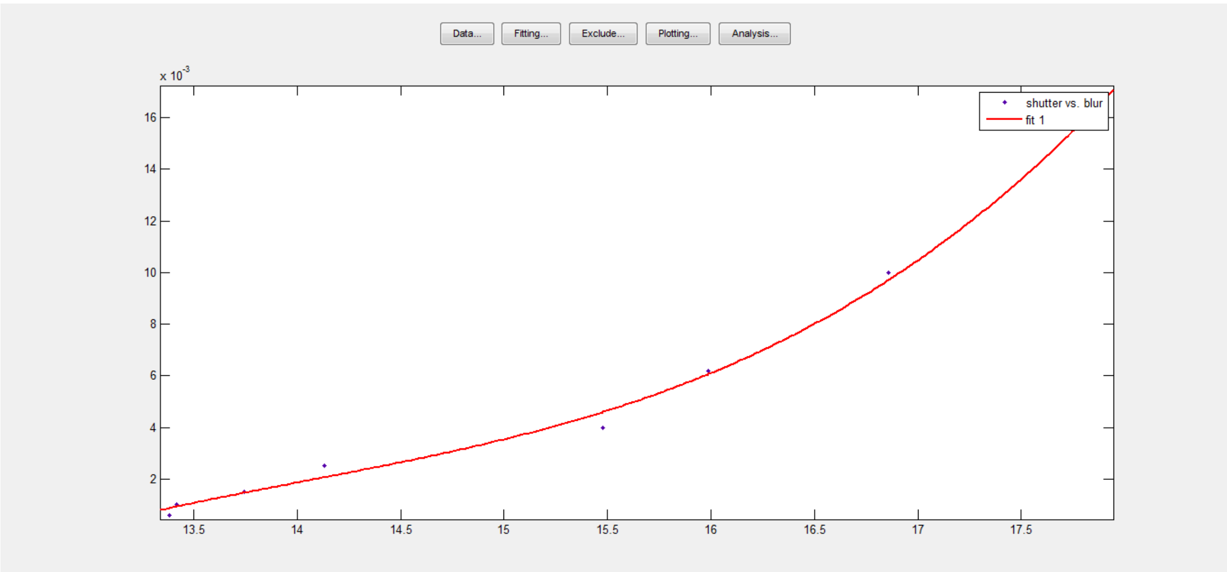


Figure 9.9: Cubic polynimial curve for Shutter vs. Blur at F=13

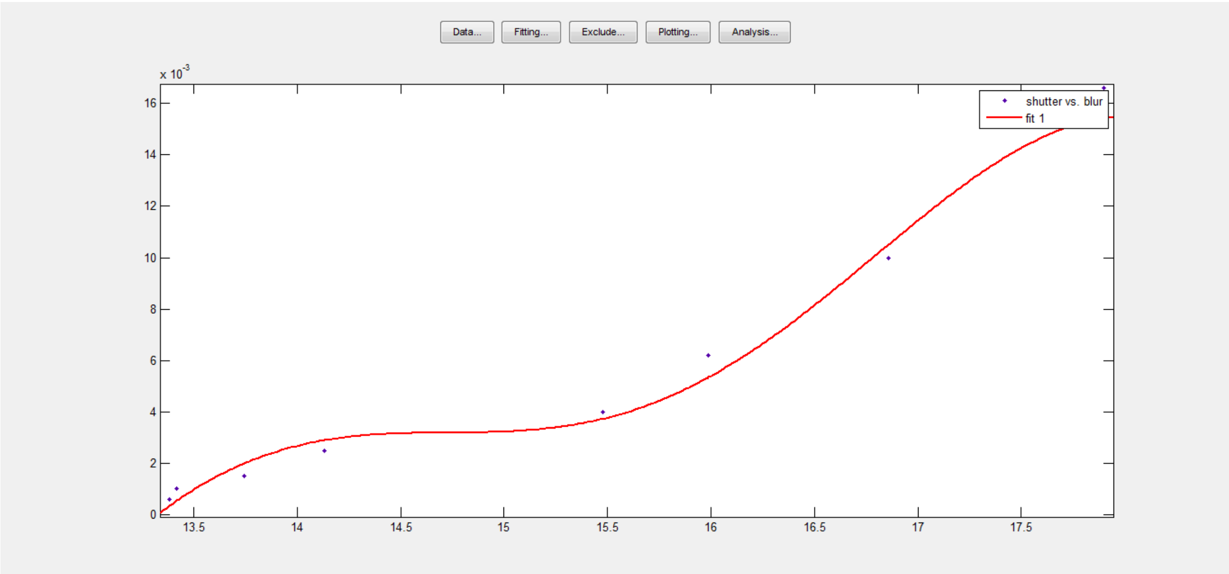


Figure 9.10: Sum of sin curve for Shutter vs. Blur at F=13

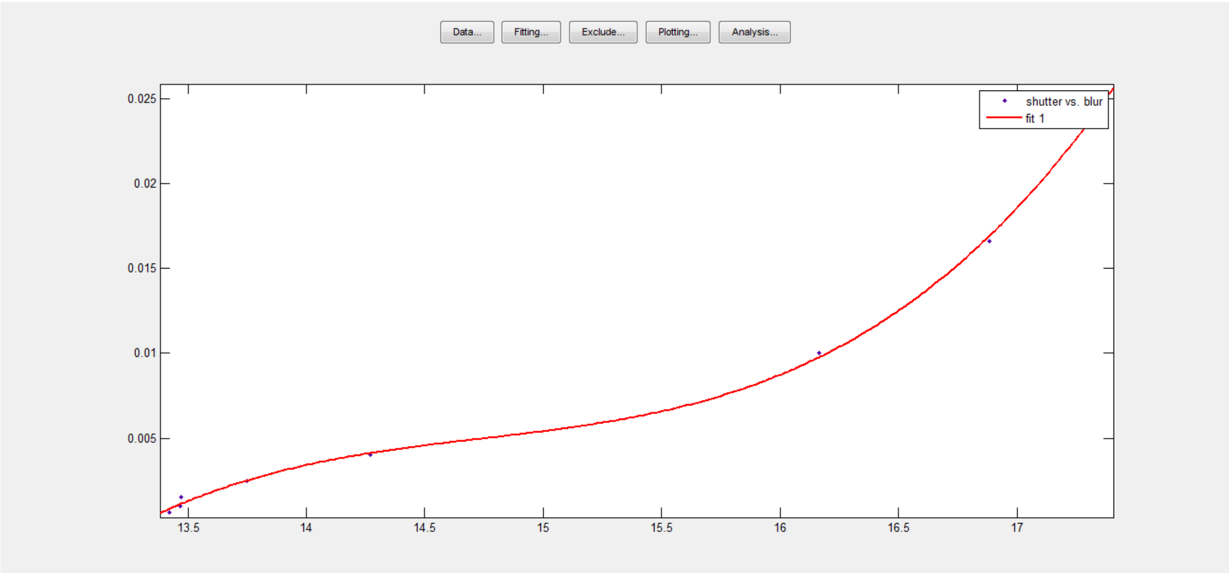


Figure 9.11: Cubic polynimial curve for Shutter vs. Blur at F=16

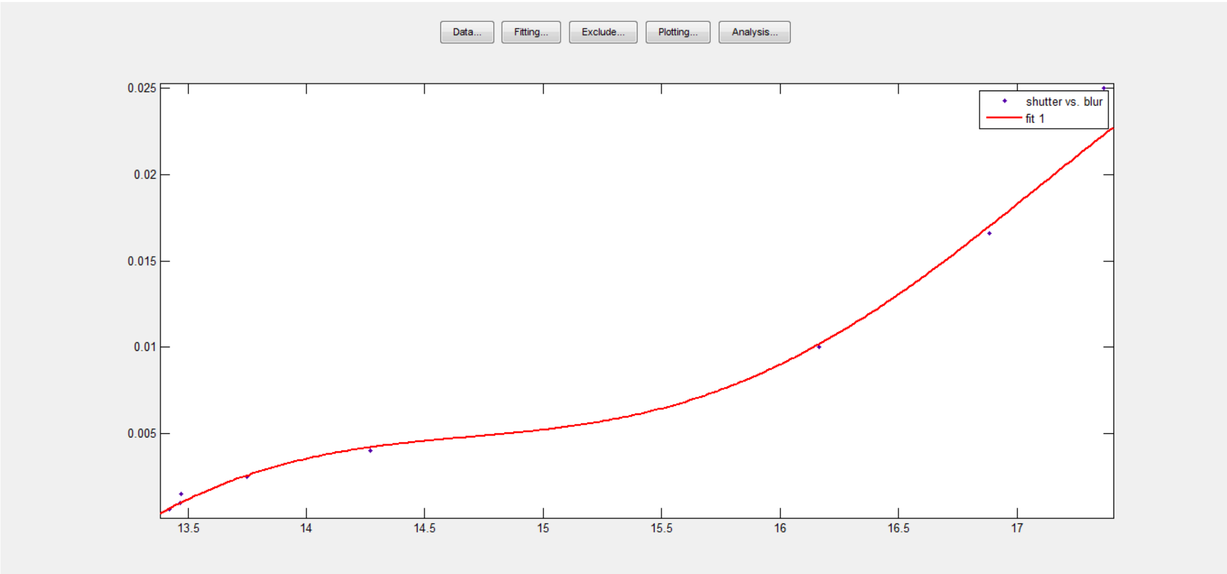


Figure 9.12: Sum of sin curve for Shutter vs. Blur at F=16

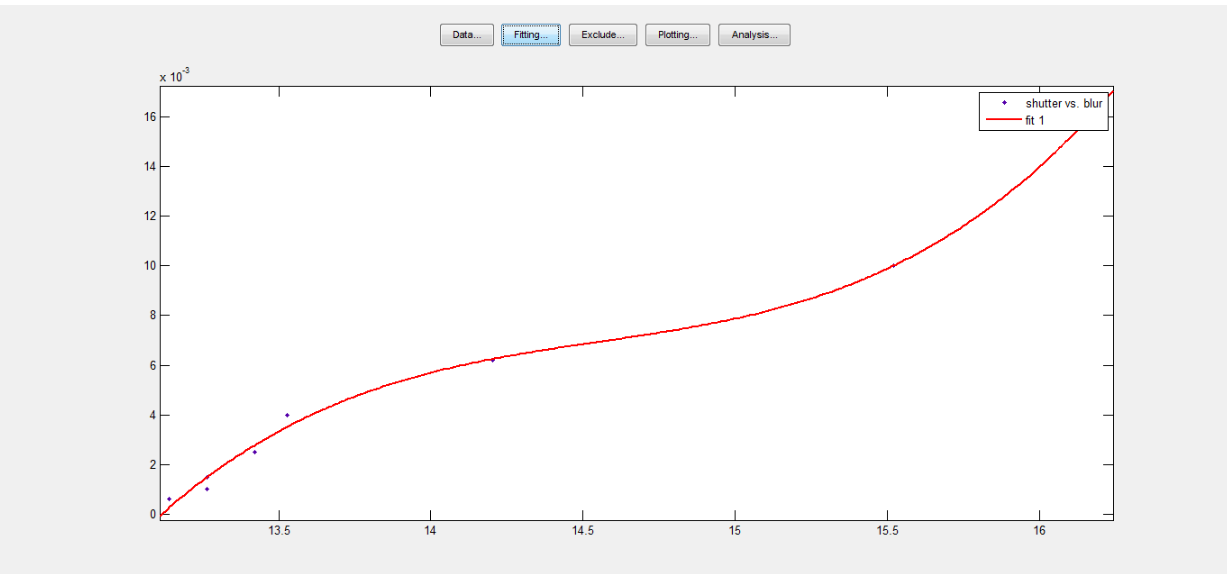


Figure 9.13: Cubic polynimial curve for Shutter vs. Blur at F=20

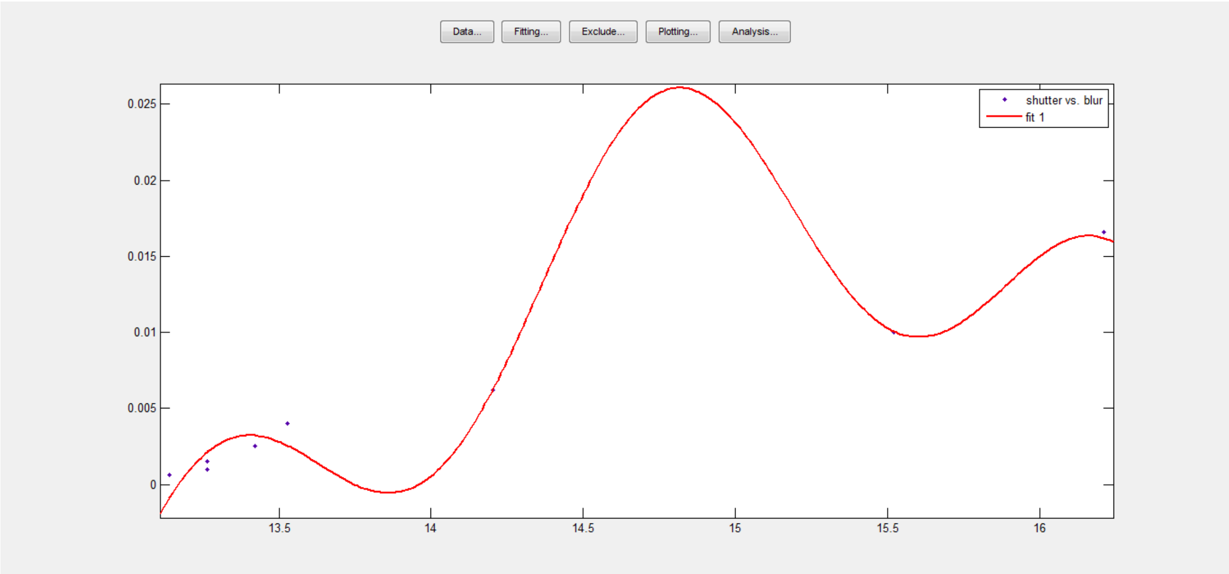


Figure 9.14: Sum of sin curve for Shutter vs. Blur at F=20

Chapter 10

Conclusion

We have designed algorithm which involves three alternatives for estimating best possible transient parameters. Using our primary database, we are in a position to successfully determine quantitative contrast and blur in any captured image. Current work involves use of curve tting tool for obtaining curves for different data inputs. Input quantity of blur determines range of possible shutter speeds, and cyclically shutter speed determines aperture for best possible image which is blur free. Processing of colored image and normalizing RGB is critical phase in project. So, three alternative methods are implemented. Future work involves study of support vector machine and adaptive processing techniques. Number of images required for computation can be reduced using adaptive technique. However reducing shutter speed doesnt allow lenses to capture enough light to produce bright image. Hence, using one of image parameter aperture, which also used to produce equation is linked with shutter speed and blur. For getting real time conditions to increase efciency in parameters, we introduced readings for different lighting conditions. Also now the concept if adaptive behavioral entity of camera parameters as a core to the main algorithm which is being used for suggestion generation. This entity is now one camera specific now. The entity can be modified as basis form supported for all cameras so that the user will be able to get all camera conditions by adapting to real time conditions automatically. This hypothesis can be related in camera application when one want to explore camera parameters effect.

Bibliography

[1] Guangcan Liu, Yi Ma, Blind Image Deblurring by Spectral Properties of Convolution Operators

[2] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, Blind motion deblurring from a single image using sparse approximation, *CVPR*,2009

[3] N. Joshi, R. Szeliski, and D. J. Kriegman, Psf estimation using sharp edge prediction, *CVPR*,2009

[4] N. Wiener, Extrapolation, Interpolation, and Smoothing of Stationary Time Series. *The MIT Press*,1964

[5] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman, Blur kernel estimation using the radon transform, *CVPR*,2011

[6] J. Jia, Single image motion deblurring using transparency, *CVPR*,2007

[7] Levin, Blind motion deblurring using image statistics, *NIPS*,2006

[8] L. Xu and J. Jia, Two-phase kernel estimation for robust motion deblurring, *ECCV*,2006

[9] M.Talole, P.Talole.Hetertogenous Database Migration using ODTDM Supported with SAX and SDM algorithms, International Journal Of Engineering And -Computer Science, Volume 3 Issue (10 October 2014).

[10] Q. Shan, J. Jia, and A. Agarwala, High-quality motion deblurring from a single image, *ACM Trans. Graph*,2008

[11] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, Removing camera shake from a single photograph, *ACM Trans. Graph*,2008

[12] M.Talole, R.Somwanshi, C.Kulkarni. Safety using Road Automated Wireless Communicating Smart Helmet Application (SURACSHA) ", International Journal of Engineering Research and Technology (IJERT), Vol. 3 - Issue 9 (September - 2014)

[13] X. Meng and Z. Hu. A new easy camera calibration technique based on circular points. Pattern Recognition, 36(5), 2003.

[14] S. M. Bryant, A Case-based Reasoning Approach to Bankruptcy Prediction Modeling, Intelligent System Accounting, Financial and Management., Vol. 6, pp.195-214. 1997.

[15] J. Weng, P. Cohen and M. Herniou. Camera calibration with distortion models and accuracy evaluation. IEEE Trans. PAMI 14(1992) 965-980.

[16] Tsai, Roger Y. (1987) A Versatile Camera Calibration Technique for HighAccuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses, IEEE Journal of Robotics and Automation, Vol. RA3, No. 4, August 1987, pp. 323344.

[17] J. C. Platt, Advances in Kernel Methods: Support Vector Machines, B. Schlkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, Dec. 1998. Fast training of support vector machines using sequential minimal optimization

[18] Z. Zhang. A flexible new technique for camera calibration. PAMI, 22(11), 2000.

[19] Naslavsky L., Ziv H., Richardson D.J., "A Model-Based Regression Test Selection Technique", ICSM 2009.

[20] Hecker, C., McHugh, L., Dyckho, M.A.,, M.: Three approaches to Halo-style behavior tree ai. In: Game Developers Conference (2007).