**CMP 314: COMPUTER GRAPHICS AND IMAGE PROCESSING**

**IMAGE DEBLURER**

**GROUP 3**

**ADIGWE CHIBUZOR DU\110**

1. **Provided the images**
2. **Python code to blur and deblur images**

**AKPAN ALFRED DU\115**

1. **Prepared the slides**

**FAGBESAN ELIZABETH DU\131**

1. **Generated the report**

**OCHEI NAOMI DU\146**

1. **Generated the report**
2. **Matlab code to blur and deblur images**

**INTRODUCTION**

In image processing world, the blur can be caused by many factors such as defocus, unbalance, motion, noise and others. In human being, the vision is one of the important senses in our body. So the image processing also plays an important role in our life. An important problem in image processing is its blurring problem which degrades its performance and quality. Deblurring is the process of removing blurring artifacts from images, such as blur caused by defocus aberration or motion blur. The blur is typically modeled as the convolution of a (sometimes space- or time-varying) point spread function(PSF) with a hypothetical sharp input image, where both the sharp input image (which is to be recovered) and the point spread function are unknown. Image deblurring is used to make pictures sharp and useful by using mathematical model. Image deblurring (or restoration) is an old problem in image processing, but it continues to attract the attention of researchers and practitioners alike. A number of real world problems from astronomy to consumer imaging find applications for image restoration algorithms. Image restoration is an easily visualized example of a larger class of inverse problems that arise in all kinds of scientific, medical, industrial and theoretical problems. To deblur the image, a mathematical description can be used. It can be started with a shift-invariant model, meaning that every point in the original image spreads out the same way in forming the blurry image.

**DEBLURRING**

Deconvolution for deblurring

Images can be distorted by blur, such as motion blur or blur resulting from an out-of-focus lens. Blur is represented by a distortion operator, also called the point spread function (PSF). Different deblurring algorithms estimate and remove blur based on how much knowledge you have of the PSF and noise in the image.

**Functions**

|  |  |
| --- | --- |
| [deconvblind](https://www.mathworks.com/help/images/ref/deconvblind.html) | Deblur image using blind deconvolution |
| [deconvlucy](https://www.mathworks.com/help/images/ref/deconvlucy.html) | Deblur image using Lucy-Richardson method |
| [deconvreg](https://www.mathworks.com/help/images/ref/deconvreg.html) | Deblur image using regularized filter |
| [deconvwnr](https://www.mathworks.com/help/images/ref/deconvwnr.html) | Deblur image using Wiener filter |
| [edgetaper](https://www.mathworks.com/help/images/ref/edgetaper.html) | Taper discontinuities along image edges |
| [otf2psf](https://www.mathworks.com/help/images/ref/otf2psf.html) | Convert optical transfer function to point-spread function |
| [psf2otf](https://www.mathworks.com/help/images/ref/psf2otf.html) | Convert point-spread function to optical transfer function |
| [padarray](https://www.mathworks.com/help/images/ref/padarray.html) | Pad array |

**METHODOLOGY**

**Data Collection:**

The images were obtained from the internet (). The images were automatically blurred with Mathlab and Python code and de-blurred with Mathlab code.

**Image Processing Operations:**

The images processes were to blur and de-blur.

**Mathlab Code**: It was used to blur and de-blur

**%%original image**

**Image = im2double((imread('pic1.jpg')));**

**subplot(1,3,1), imshow(Image), title('Original Image');**

**%%blurred image**

**Len = 100;**

**theta = 50;**

**psf = fspecial ('gaussian', len, theta);**

**blurred = imfilter (Image, psf,'conv', 'circular');**

**subplot (1,3,2), imshow (blurred), title('Blurred Image');**

**%%recovered(de-blur) image**

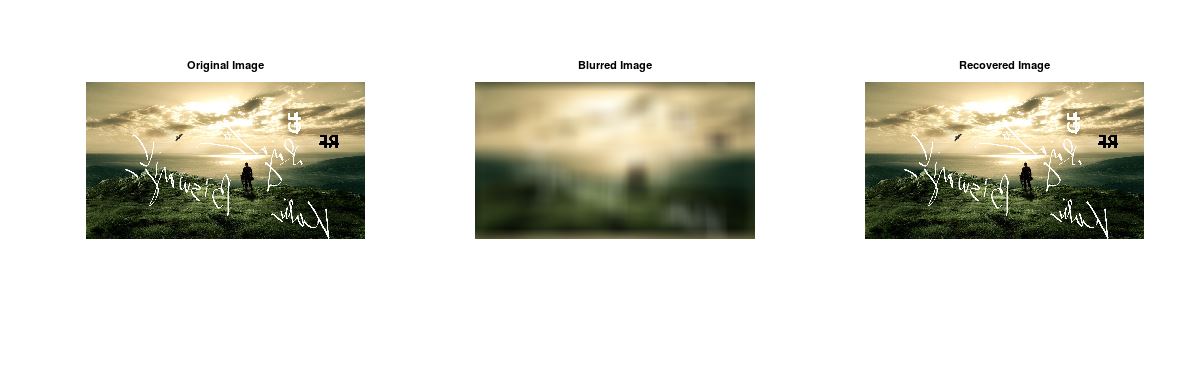
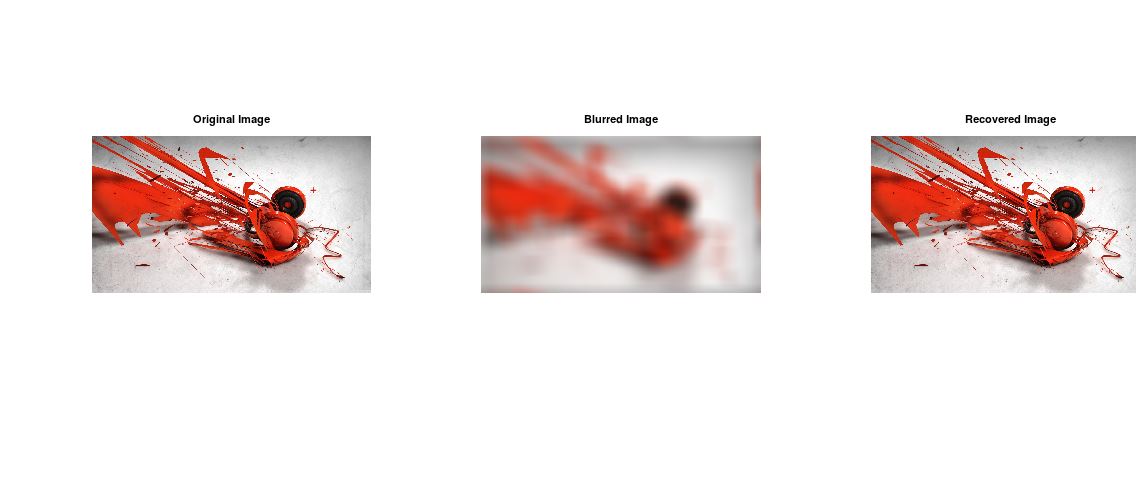
**Recovered = deconvwnr(blurred,psf,0);**

**Subplot (1,3,3), imshow (Recovered), title('Recovered Image');**

**Implementation:**

1. The images were obtained from the internet.
2. The images were put in a folder to and uploaded on Github.
3. The Python code was written and uploaded on Github to blur the images.
4. Math lab code was written in GNU Octave to blur and de-blur images.

**RESULTS**

****

**Conclusion**

The images were used to generate appropriate results with Math lab and Python Code respectively.

The Python code was uploaded on Github alongside the images that were used to perform the image processing operations. Github link <https://github.com/iamphantasm0/ImageProcessingBlurAndUnblur>

**References**

1. <https://www.mathworks.com/help/images/image-restoration-deblurring.html>
2. [IRJET-V3I12241.pdf](https://www.irjet.net/archives/V3/i12/IRJET-V3I12241.pdf#:~:text=Image%20deblurring%20is%20a%20process%2C%20which%20is%20used,e.g.%2C%20remove%20the%20effect%20of%20imaging%20system%20response.)
3. <https://www.mediafire.com/file/j8gr3z1316gum62/Download_500%252B_3D_Wallpapers_in_a_ZIP_File_%2528HD_Wallpapers_2017%2529.zip/file>