Depth Map Preparation and Salient Object Segmentation using Focal stack

A thesis

submitted to

The Indian Institute of Science Education and Research Pune

in partial fulfillment of the requirements for the

BS-MS Dual degree program

Submitted by:

Ankit Kumar Vishwakarma (20191088)

Supervisor:

Ravindra Chaugule

Principal Engineer Renishaw Metrology Systems, Pune, India

Expert:

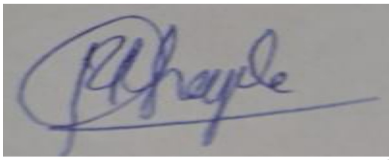
Joy Merwin Monteiro

Assistant Professor Earth and Climate Science Data Science (Joint), IISER Pune

May 17, 2024

Certificate

This is to certify that this dissertation entitled Depth Map preparation and Salient Object Segmentation using Focal stack towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Ankit Kumar Vishwakarma under the supervision of Ravindra Chaugule Principal Engineer ,Renishaw Metrology Systems, Pune during the academic year 2023-2024.



Ravindra Chaugule

Committee:
Ravindra Chaugule
Prof. Joy Merwin Monteiro

Declaration

I hereby declare that the matter embodied in the report entitled Depth Map preparation and Salient Object Segmentation using Focal stack are the results of the work carried out by me at Renishaw Metrology Systems, Pune, under the supervision of Ravindra Chaugule, and the same has not been submitted elsewhere for any other degree. Wherever others contribute, every effort is made to indicate this clearly, with due reference to the literature and acknowledgement of collaborative research and discussions.

Ankit Kumar Vishwakarma

Reg. No. - 20191088

Acknowledgments

I would like to express my heartfelt gratitude to my supervisor, Ravindra Chaugule, for his invaluable guidance and mentorship during my internship at Renishaw Metrology. Under his supervision, I had the privilege of learning and growing significantly. I am particularly grateful for his teachings on not being complacent with my work. In retrospect, I realize that his firmness and high expectations were essential in shaping me professionally. I am deeply grateful to my expert, Joy, for his guidance, support during times of difficulty. Joy's mentorship and assistance helped me navigating through challenging situations. I would like to express my sincere gratitude to my friends Aniket, Samarth, Mehul, Shubhankar, Raghav, Vedant, Varun, Sanket, Rishikesh, Arindam, and Rajeet for their support, companionship, and guidance throughout the course of my BS-MS degree. Their presence made the journey enjoyable and memorable. I am immensely grateful for their encouragement, fun times, and valuable insights. I missed the presence of Shubhankar and Raghav during the fifth year, and I appreciate their support despite not being around as much. Their friendship has been invaluable, and I am truly fortunate to have them in my life.

Abstract

This thesis presents a method for generating depth maps from a focal stack of images and utilizing the depth information for salient object detection and binary segmentation. The focal stack dataset used is the Mobile Depth dataset captured using a mobile phone camera. The approach involves aligning the frames in the focal stack, computing a sharpness map using a discrete cosine transform (DCT) based focus measure, refining the sharpness map through edge-preserving filtering, and estimating the depth map by a weighted combination of frame indices. The depth map is then employed as a saliency map for salient object detection. An adaptive thresholding technique based on Otsu's method generates a trimap, which is fed into the GrabCut algorithm to produce a high-quality binary segmentation mask of the salient object. Challenges addressed include handling textureless regions, achieving accurate depth estimation with limited sampling frequency, and preserving edge details during filtering. The proposed method aims to leverage depth information from focal stacks to enhance salient object detection and segmentation performance, with potential applications in areas such as computer vision and image processing.

Contents

Chapter 1

Introduction

1.1 Background

Estimating scene depth from image captured from an RGB camera is a fundamental task in computer vision. Active and passive depth estimation are two fundamental approaches used in computer vision and depth sensing to determine the distance or depth of objects within a scene. i) Active depth estimation involves the use of external stimuli, such as light or sound, which are emitted or projected onto a scene. For example, Laser scanning systems use laser beams to sweep across a scene, measuring the time it takes for the laser pulses to reflect back to the sensor. The distance to objects is determined based on the time-of-flight of the pulses. But active methods require complex hardware setups and are expensive. ii) Passive depth estimation relies on analyzing the information present in the captured images without the need for emitting external stimuli. These methods include:
a) Stereo Vision: Stereo vision (7) systems use multiple cameras or images captured from different viewpoints to calculate depth by triangulating corresponding points in the images. Depth is determined based on the disparity between corresponding image points.
b) Depth-from-Defocus (DFD): Depth from Defocus (1) techniques analyze the blur or sharpness of objects in images captured with varying focus settings. By analyzing

the degree of defocus, these methods can estimate depth.
c) Depth-from-Focus (DFF): Depth from Focus ([5])focuses on estimating depth information based on the sharpness or focus quality of image patches across images captured at different focus settings. DFF techniques typically involve determining the optimal focus setting for each image patch to estimate depth.
d) Structure from Motion (SfM): SfM ([6]) techniques analyze the motion of objects or features across multiple images to infer depth information. Depth is calculated based on the relative motion of objects in the scene.
Passive methods are often simpler and less expensive to implement but passive methods may be limited by factors such as textureless regions, occlusions, or low-contrast scenes.

I opted to utilize Depth-from-Focus (DFF) method. Depth from Focus (DFF) leverages a focal stack to estimate scene depth. When you take a photo with your mobile phone, the camera lens sweeps the focal plane through the scene and adjusts its focus to ensure that the object of significance in the scene appear sharp and clear in the captured image. This a collection of images captured of the same scene with varying focal distances is known as focal stack or focus bracket. DFF and DFD ([2; 3]) are closely related techniques used for estimating depth information from images captured at different focus settings. Both methods exploit variations in focus across the image to infer depth. In the simplest case, DFD may use only two frames: one captured in sharp focus and the other intentionally defocused. By comparing the amount of blur between these two frames, DFD can estimate the depth of objects in the scene. In DFD, the amount of defocus blur is often quantified using the concept of Circle of Confusion (CoC). When a point light source is unfocused, the light rays converge either in front of or behind the image plane, producing a blurred circle on the image plane. Whereas when the point light source is in focus, it will create an infinitely small point on the image plane, resulting in the sharpest projection with the minimum Circle of Confusion (CoC) as shown in below figure.

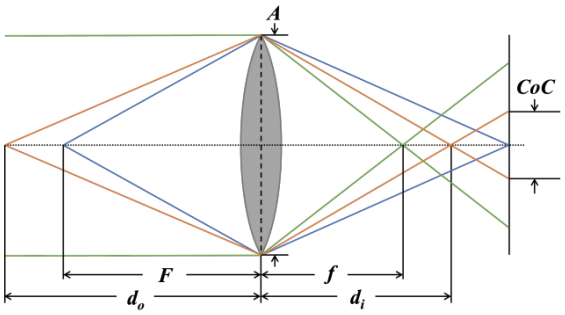


Figure 1.1: Illustration of the thin-lens equation

According to lens laws, the CoC can be expressed using the following equation:

$$\text{CoC} = A \left| \frac{d_o - F}{d_o} \right| \frac{f}{F - f}$$

where A is the aperture diameter, d_o is the object distance, or depth, F is the focus distance, f is the focal length. In practice, the f-number, N = f /A, is commonly used to describe the aperture size. Even though DFD requires less number of frames as compared to DFF, it relies on establishing a mathematical relationship between depth and defocus patterns. This necessitates assumptions that may not accurately represent real-world scenarios. Specifically, DFD commonly assumes that object points are centralized at a small plane and that the Point Spread Function (PSF) follows a particular distribution (25). However, these assumptions may not hold true in practical situations, leading to inaccuracies in depth estimation. DFF operates under the assumption that there exists a single optimal-focused frame for each pixel within a focal stack. This condition is theoretically assured (24) for thin lens cameras, provided that the sampling of focal distances is sufficiently dense. In this scenario, the focal distance corresponding to the best-focused frame can be utilized for pixel depth estimation.

1.2 Research Objectives

The objective of this thesis is to develop a method for generating a depth map from a focal stack, utilizing it as a saliency map for Salient Object Detection (SOD) and to produce binarized salient object mask. Depth maps provide a comprehensive understanding of the spatial layout, distinguishing between foreground and background elements. This involves exploring techniques to accurately extract depth information from the focal stack, followed by employing the derived depth map to enhance the performance of SOD and binary segmentation algorithms. The focal stack dataset which i used is Mobile Depth dataset ([2]). It comprises real-world Depth from Focus (DFF) data captured using a mobile phone.

In Depth from Focus (DFF) techniques, a focus measure is commonly used to determine the sharpness of each pixel of the image. The focus measure is a mathematical function that quantifies the degree of focus in an image region. It helps identify the areas in the image that are in focus and those that are blurred. But one of the primary challenges in DFF is the design of effective focus measures. There are various focus measures available, each with its own strengths and weaknesses. These are gradient based measures([4]), Laplacian-based measures([20; 21]), frequency transformation-based measures([22]), and statistic based measures([23]). There are following challenges we encounter in DFF methods.

i) Textureless Regions: Another challenge arises in textureless regions of images, where traditional focus measures may produce low responses. When inferring depth based on the image index with the maximum sharpness value for each pixel (i.e., argmax), the initial depth estimate is highly noisy and far from accurate. In such areas, relying solely on focus measures may not be sufficient to accurately determine the focus status. To address this issue, additional context information ([8]) is often required to infer the focus status accurately. This could involve incorporating surrounding pixel information or employing sophisticated algorithms to enhance focus estimation in textureless regions.

ii) High Sampling Frequency Requirement: DFF techniques typically require capturing multiple images at different focus distances to estimate depth information

accurately. However, achieving a high sampling frequency can be challenging in practice, especially in real-time applications. If the sampling frequency is insufficient, there may not be a clear distinction between the sharpest pixel across the focal sweep, leading to inaccuracies in depth estimation. Traditional DFF methods often mitigate this issue by utilizing a large number of input frames (9), which can significantly impact computational efficiency and limit the method's real-time performance.

1.3 Approach

The focal stack dataset which i used is Mobile Depth dataset (2). It comprises real-world Depth from Focus (DFF) data captured using a mobile phone. The focal stack undergoes several processing stages to achieve the final segmentation. These stages include the following:

i) Focal Stack Alignment: It is essential to align the images in focal stack before proceeding with further processing steps. Alignment is very important to ensure that the focal stack accurately represents the scene and that subsequent depth estimation or image fusion techniques yield reliable results. The alignment step aims to rectify the parallax and viewpoint discrepancies that naturally occur when capturing a focal stack using a camera. The objective is to ensure that the aligned focal stack closely resembles one obtained from a static, telecentric camera, where parallax and viewpoint variations are minimized or non-existent. In addition to addressing parallax and viewpoint changes, there is a minute magnification changes that occur while capturing a focal stack due to focus adjustment. The alignment could be done using optical flow (11), homography (12), or any other methods.

ii) Choice of Focus Measure: I opted for a Discrete Cosine Transform (DCT) (10) based focus measure from the array of available options primarily due to its sensitivity to high-frequency details within the image, which is crucial for accurately assessing sharpness and focus. In the DCT-based approach, the image or a specific region of interest is transformed into the frequency domain using the DCT. The en-

ergy distribution in the transformed domain is then analyzed to assess the sharpness or focus quality. Typically, higher energy concentrations at lower frequencies correspond to blurred or out-of-focus regions, while higher frequencies represent sharper areas.

iii) Sharpness Map Refinement: I begin by employing a DCT-based focus measure to generate individual sharpness maps for each image frame within the focal stack. Next, I enhance the emphasis on salient regions in the image by weighting the sharpness map by a local entropy map derived from the sharpness map itself. The obtained sharpness exhibits noise and abrupt transitions among neighboring pixels. The final sharpness map undergoes smoothing through edge-preserving filters (18) to mitigate the impact of outliers and uphold boundary integrity.

iv) For depth map estimation we devised an probabilistic prediction mechanism where instead of taking frame index of most focused pixel we weight the frame index by its sharpness value for each pixel. The shaprness value is used as the probability of the best-focused frame. This method ensures that even if certain frames are not directly visible in the focal stack, their contribution to the depth map is captured through the weighted pixel intensity, allowing for a comprehensive representation of the scene's depth.

v)Grab Cut: Given the depth map, the subsequent step involves extracting the salient object. The result is represented as a binary depiction of the salient object. While past studies have utilized fixed thresholding for segmentation, employing a single fixed threshold across a large dataset presents challenges. To address this, we initially compute an adaptive threshold, denoted as T_b, using the Otsu (14) method for each image based on its depth map. This adaptive thresholding provides a preliminary estimation of the object and background. Subsequently, we create a trimap based on T_b to mitigate errors in the saliency maps. Finally, the trimap is inputted into the GrabCut framework (15) to generate a high-quality salient object mask.

vi) Post processing: After the initial segmentation using GrabCut, the resulting binary mask may contain noise contours or isolated regions, which can degrade the quality of the segmentation. To address this issue, a post-processing step is performed to refine the binary mask. This involves applying connected components

analysis, a technique used to identify connected regions of pixels with similar properties. In connected components analysis, each connected region in the binary mask is assigned a unique label. By analyzing the properties of these connected regions, such as their area, we can identify and eliminate noise contours or isolated regions that do not correspond to the desired foreground or background.

Chapter 2

Methods

2.1 Optical Flow

Optical flow (11) is a technique used in computer vision to track the motion of objects in a sequence of images. Optical flow estimation aims to determine the displacement of pixels between consecutive frames of a focal stack. One common application of optical flow is image alignment, where it can be used to register or align images so that they are spatially coherent.

The Lucas-Kanade (26) method is a popular optical flow algorithm that estimates the motion field by solving a system of linear equations. Here's a detailed explanation of the Lucas-Kanade optical flow method for image alignment:

Assumptions:

i)Brightness Constancy: It assumes that the brightness of a pixel remains constant between consecutive frames. This assumption is expressed as:

$$I(x, y, t) = I(x + u, y + v, t + 1)$$

where (x, y) are pixel coordinates in the image, t is the frame index, and u and v represent the horizontal and vertical components of the optical flow at pixel (x, y).

ii)Spatial Coherence: It assumes that nearby pixels in the image have similar motion. This assumption is used to regularize the optical flow estimation.

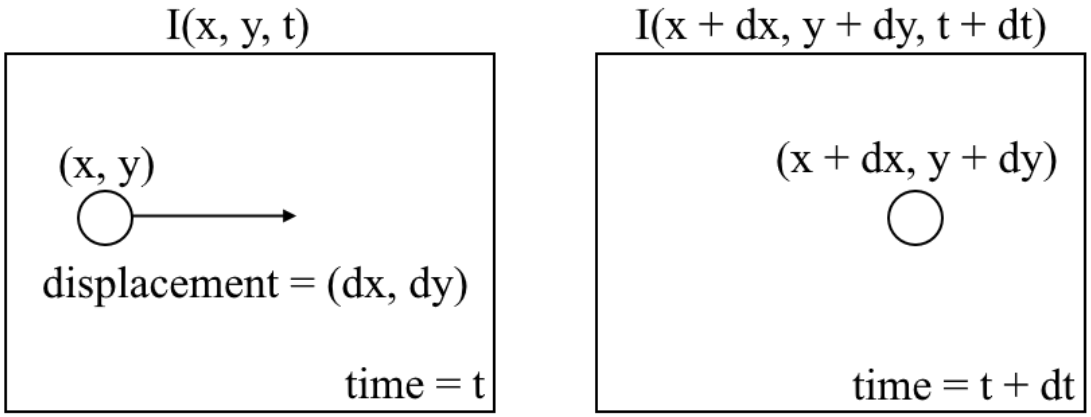


Figure 2.1: Optical flow problem

Optical Flow Equation: The optical flow equation is derived from the brightness constancy assumption and Taylor expansion.According to the brightness constancy assumption, we have:

$I(x, y, t) = I(x + u, y + v, t + 1)$

Taylor series expansion of $I(x + u, y + v, t + 1)$ around (x, y, t) gives:

$I(x + u, y + v, t + 1) \approx I(x, y, t + 1) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}$

We can neglect the higher-order terms, resulting in the optical flow equation:

$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0$

Lucas-Kanade Algorithm:

- **Window Selection:** Divide the image into small patches or windows.
- **Local Approximation:** Assume that optical flow remains constant within each window. This leads to the formation of a system of linear equations for each window.
- **Solution:** Solve the overdetermined system of equations using least squares estimation to find the optical flow vectors (u, v) that minimize the error.

- **Matrix Formulation:** The system of equations for each window can be represented in matrix form as:

$$A^T A \begin{pmatrix} u \\ v \end{pmatrix} = A^T b$$

where A is the Jacobian matrix formed by stacking the spatial gradients of intensity within the window, and b is the temporal gradient vector.

Implementation:

- Define a local window around each pixel.
- Compute spatial gradients $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ within the window.
- Compute temporal gradient $\frac{\partial I}{\partial t}$ by subtracting corresponding pixel intensities between frames.
- Solve the system of linear equations using least squares to estimate (u, v) for each pixel.

Image Alignment: Once the optical flow vectors (u, v) are estimated for each pixel or region, they can be used to warp one image onto another to achieve alignment. This involves warping the second image based on the estimated flow field. Warping involves transforming the pixels of an image from one location to another based on obtained Optical flow vector field.

Interpolation: After obtaining the flow field at each level of the pyramid, the algorithm interpolates the flow vectors to compute the flow field at the original resolution of the input frames. This interpolation process ensures that the final dense optical flow field is aligned with the original image resolution.

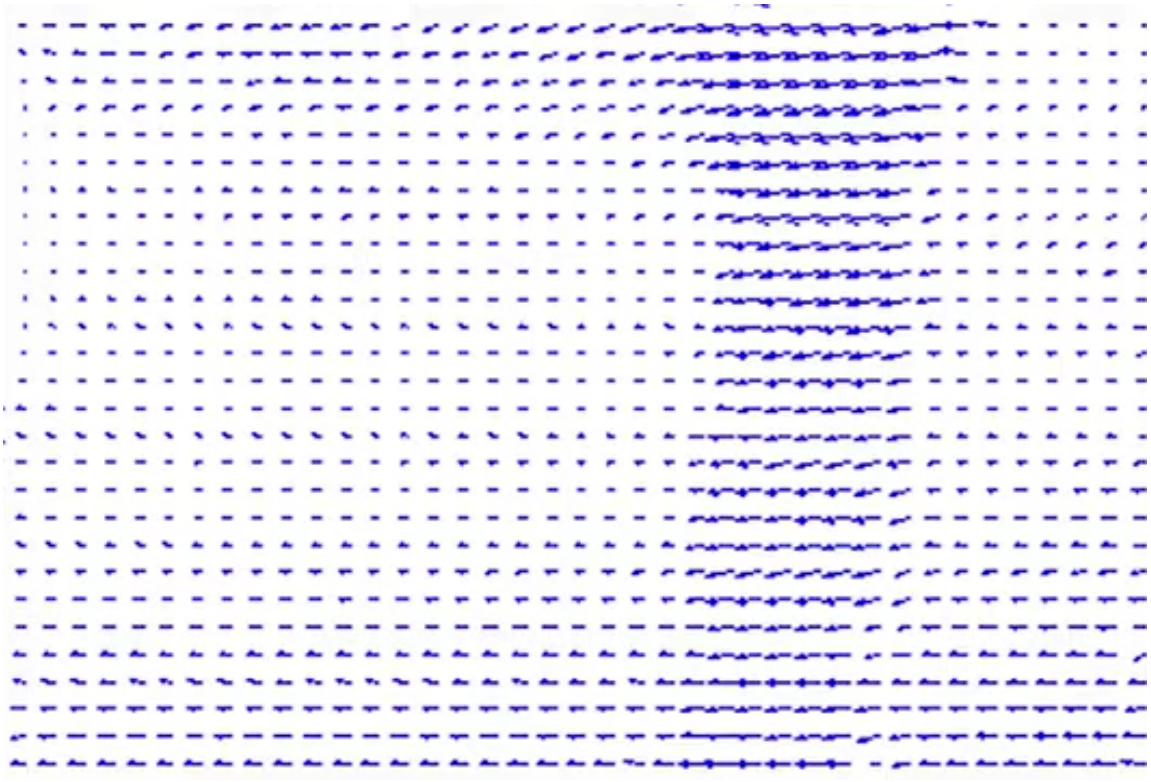
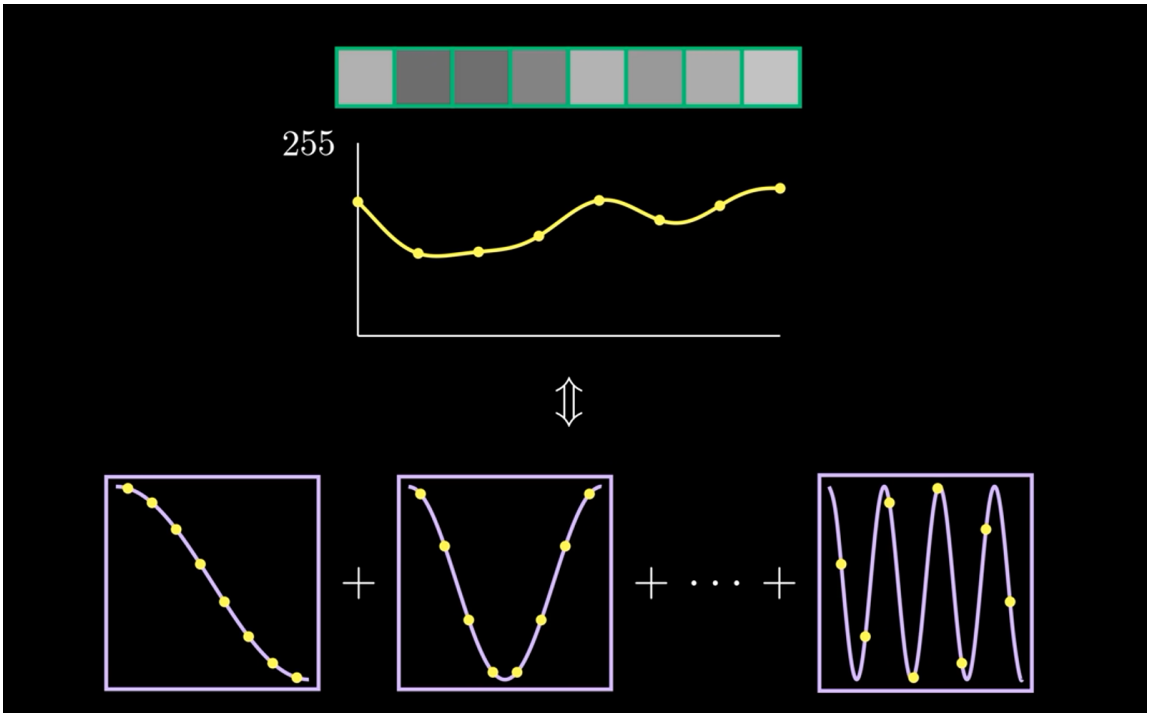


Figure 2.2: Optical Flow: It is 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second.

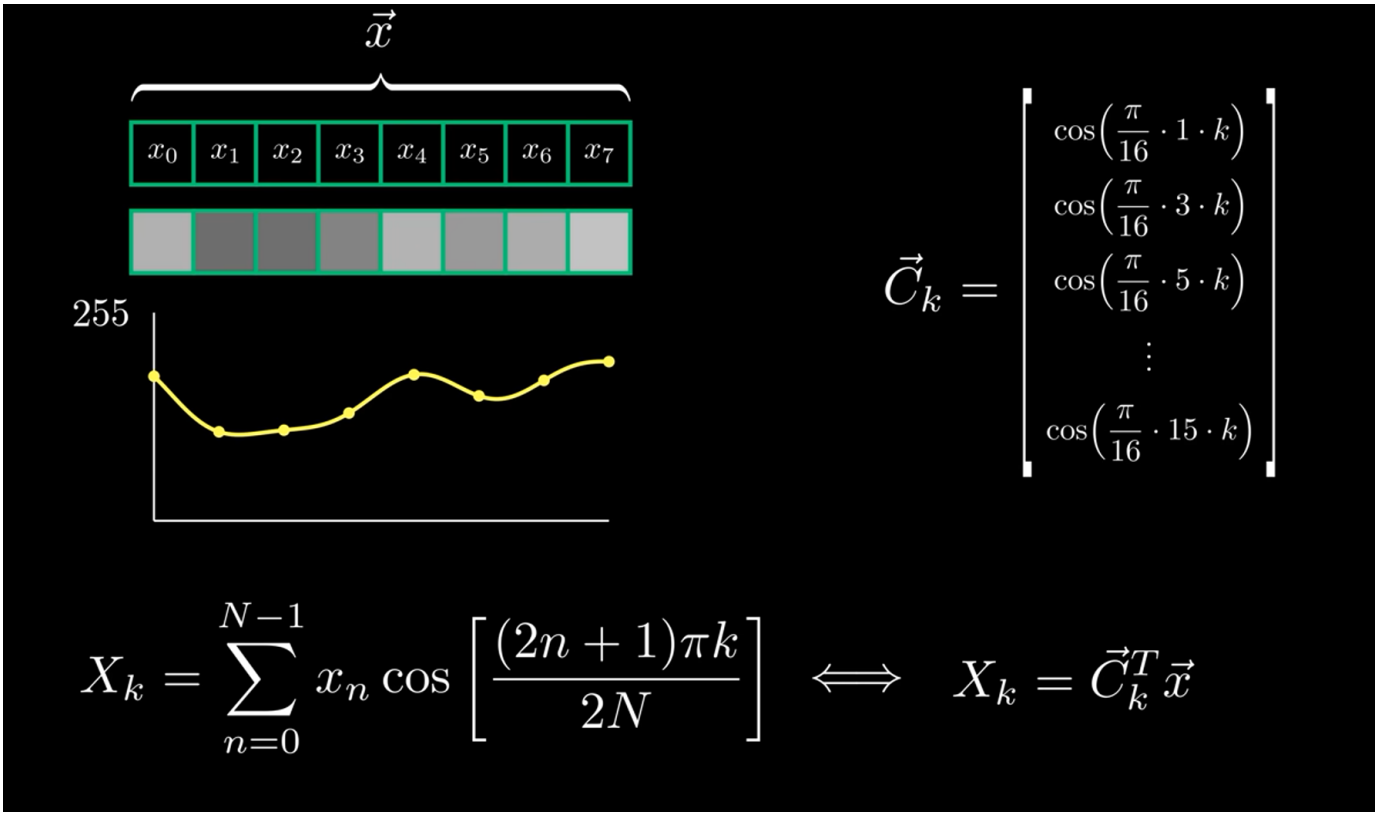
2.2 Focus Measure: Discrete Cosine Transform (DCT)

We begin with description of what exactly is Discrete Cosine Transform. Visualising images as signals allows us to talk about frequency component within an image. Let's take an input of 8 pixels and suppose it forms some sort of signal. The clever idea behind DCT (10) is to represent these eight points as sum of sample points from cosine waves of various frequencies.



DCT takes an input of sampled points from our original signal and gives us an output of the same size. we refer to the outputs of the dct as coefficients. These coefficients represent the weights of cosine waves of different frequencies that contribute to the original signal. The cosine term in the below equation represent the sampled points from a cosine wave with frequency k. In order to get the DCT coefficient corresponding to K_{th} frequency we are summing over a product of each sampled points from input signal with the samples from the cosine wave.

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{(2n+1)\pi k}{2N}\right) \iff X_k = \vec{C}_k^T \vec{x}$$



By sampling method we can think of the entire dct as a matrix vector product. All we are doing is linear transformation. The rows of the matrix are sampled points from the cosine wave of a particular frequency. we see DCT as a way of decomposing a signal into a coefficient representation of weights associated with cosine waves.

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} = \begin{bmatrix} \longleftarrow & C_0^T & \longrightarrow \\ \longleftarrow & C_1^T & \longrightarrow \\ \longleftarrow & C_2^T & \longrightarrow \\ \longleftarrow & C_3^T & \longrightarrow \\ \longleftarrow & C_4^T & \longrightarrow \\ \longleftarrow & C_5^T & \longrightarrow \\ \longleftarrow & C_6^T & \longrightarrow \\ \longleftarrow & C_7^T & \longrightarrow \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \tag{2.1}$$

finally, he formula for the DCT transform coefficient $F(u, v)$ of an 2D image patch $f(x, y)$ at coordinates (u, v) in the frequency domain is given by:

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \tag{2.2}$$

Where:

- $F(u, v)$ is the DCT coefficient at spatial frequency coordinates (u, v) in the transformed domain.
- $f(x, y)$ is the pixel intensity at spatial coordinates (x, y) in the original image.
- N is the size of the image (assuming it's square).
- $C(u)$ and $C(v)$ are normalization factors defined as:

$$C(u) = \begin{cases} \sqrt{\frac{1}{2}}, & \text{for } u = 0 \\ 1, & \text{for } u > 0 \end{cases}$$

Normalizing terms help preserve the energy of the signal/image during transformation. This means that the sum of the squared values of the transformed coefficients is equal to the sum of the squared values of the original signal/image

2.3 Sharpness Map Preparation

We propose a discrete cosine transform (DCT) based focus measures that uses mutiscale context window to compute the sharpness score of each frame in the focal stack. The context window help us to address the challenges in textureless region. The primary issue is that sharpness metrics often rely on detecting edges and high-frequency details to determine the level of sharpness. In textureless regions, there are few or no

edges and minimal intensity variations, making it difficult for these metrics to provide accurate sharpness evaluations. By considering a larger context window around the textureless region, the sharpness metric can incorporate neighboring areas that may contain edges and details. This helps in providing a more balanced assessment of the overall image sharpness.

Let I represent the input image consisting of $N1 \times N2$ pixels. Initially, we perform the Discrete Cosine Transform (DCT) on the input image in a patch-wise fashion. Here, $P_{M_{i,j}}(i_0, j_0)$ signifies a patch with dimensions $M \times M$ centered at pixel (i, j). The patch's center lies within the range $i - \lfloor \frac{M}{2} \rfloor \leq i_0 \leq i + \lfloor \frac{M}{2} \rfloor$ and $j - \lfloor \frac{M}{2} \rfloor \leq j_0 \leq j + \lfloor \frac{M}{2} \rfloor$, where $\lfloor \frac{M}{2} \rfloor$ denotes the floor function applied to $\frac{M}{2}$. Additionally, $\hat{P}_{M_{i,j}}(\nu, \mu)$, with $0 \leq \nu, \mu \leq M - 1$, denotes the DCT of the patch $P_{M_{i,j}}(i_0, j_0)$.

The Discrete Cosine Transform (DCT) of an image patch can be decomposed into components representing low, medium, and high-frequency content. These frequency components describe different aspects of the image's visual characteristics.

i)Low-Frequency Content: Low-frequency components capture the overall structure and smooth variations in intensity across the image patch. These components represent large-scale patterns and gradients, such as gradual changes in brightness or color. Low-frequency content corresponds to slowly varying signals in the spatial domain.

ii)Medium-Frequency Content: Medium-frequency components represent intermediate-scale patterns and details in the image. These components capture moderate variations in intensity, such as edges, textures, and contours. Medium-frequency content corresponds to moderately varying signals in the spatial domain.

iii)High-Frequency Content: High-frequency components encode fine details and rapid changes in intensity within the image patch. These components represent sharp transitions, fine textures, and high-contrast features. High-frequency content corresponds to rapidly varying signals, such as sharp edges and fine details, in the spatial domain.

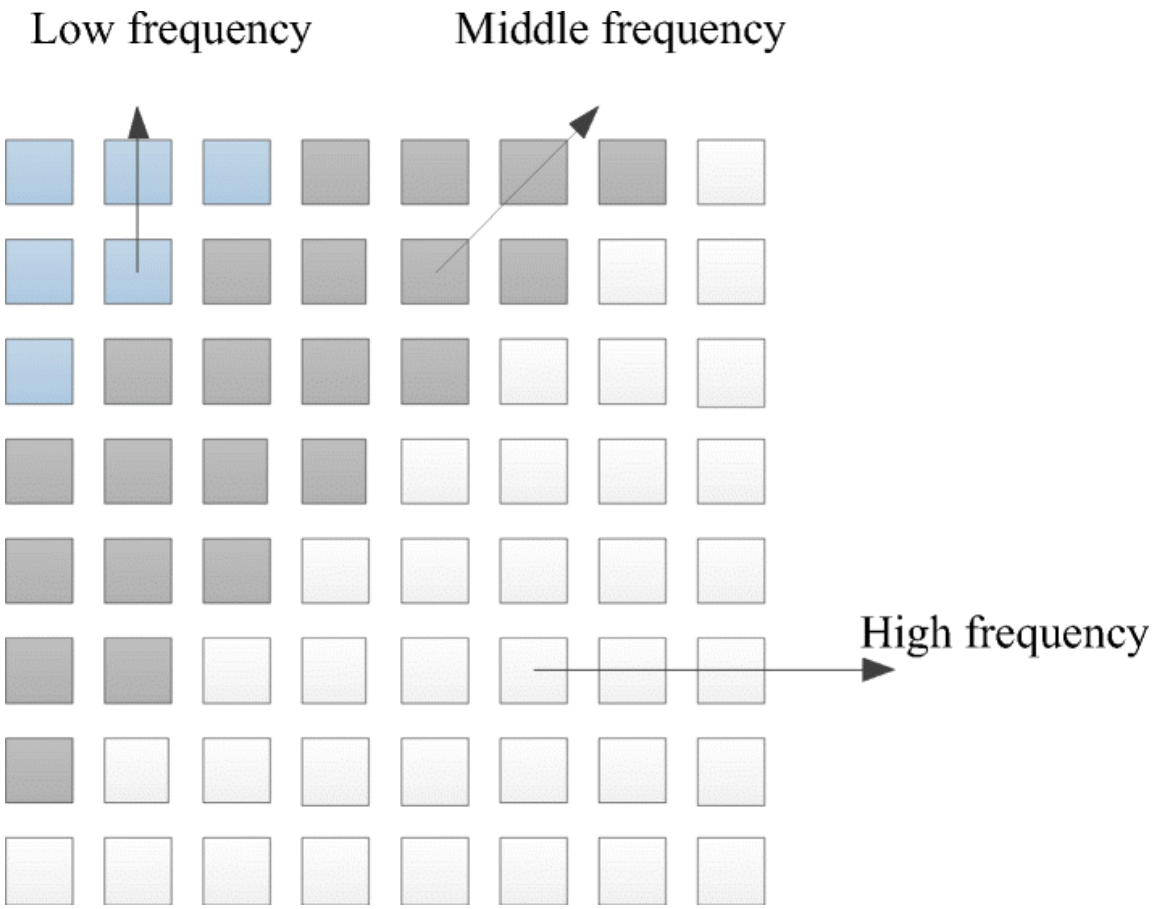


Figure 2.3: Illustration of DCT coefficients for a 8×8 block while dividing them into low-, middle-, and high-frequency bands.

To obtain sharpness measure from the DCT transform of an image patch we analyze the high-frequency components of the DCT coefficients. One common sharpness metric is to compute the energy of the high-frequency DCT coefficients. The energy of the high-frequency Discrete Cosine Transform (DCT) coefficients refers to the sum of the squared magnitudes of these coefficients. It measures the overall strength or magnitude of high-frequency components in the DCT domain.

Mathematically, the energy E of the high-frequency DCT coefficients in an image

patch can be computed as:

$$E = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} |C_{ij}|^2$$

Focusing solely on a single resolution might not provide an accurate indication of whether an image or patch is blurred. Scale ambiguity has been a subject of investigation across multiple applications (16; 17). Hence, we incorporate a multiscale model to integrate information from various scales. Through our multiscale analysis, our approach can effectively address blurring in both small-scale and large-scale structures, enhancing the detection of blurred and unblurred regions. This is done by repeating the DCT transform for each pixel with multiscale window around it and concatanating the obtained energy of the image patch from all the scales. In this work we took $2^r - 1$ scales for $r = 1, 2, 3$.

We weighted the initially computed sharpness map with the with its entropy map for the following reasons:
i)To enhance salient/focused regions: The entropy map captures the local variation in the high frequency DCT coefficients. Regions with high entropy values correspond to areas with more variation in coefficients, which are more likely to be sharp, in-focus, and salient regions in the image. By weighting the initial blur map with the entropy map, it gives more emphasis and higher weights to these salient, likely in-focus regions. This helps make the focused object boundaries and details more prominent in the final depth map.
ii)To suppress noise and insignificant variations: On the other hand, blurred regions tend to have more uniform coefficients, resulting in lower entropy values. By weighting with the entropy map, the influence of these low-entropy, likely blurred regions is suppressed in the final map. This reduces noise and insignificant variations in the smooth, defocused regions of the map.
iii) Perceptual importance: Salient, in-focus regions are generally more perceptually important for depth perception and applications like refocusing. By amplifying these regions using entropy weighting, the final blurdepth map becomes more aligned with

perceptually important areas.

To calculate the entropy for each pixel in the sharpness map, a window is considered around each pixel. The resulting entropy map has the same dimensions as the sharpness map. The entropy $E(x, y)$ at each pixel location (x, y) is computed using the following equation:

$$E(x, y) = -\sum_{i=0}^{N-1} p_i \log_2(p_i)$$

Where p_i represents the probability of occurrence of intensity level i within the window centered at pixel (x, y), and N is the total number of intensity levels.

After computing the entropy map, we perform an element-wise Hadamard product between the entropy map E and the sharpness map S. This operation weights the sharpness map by its corresponding entropy value. Mathematically, the weighted sharpness map W is obtained as:

$$W(x, y) = E(x, y) \cdot S(x, y)$$

Where $W(x, y)$ represents the pixel value in the weighted sharpness map at location (x, y).

2.4 Filtering

The obtained sharpness map is noisy and contain outliers which need to be smoothened. Traditional filtering techniques often blur edges and fine details along with noise reduction, which can lead to loss of important image features. Here we used Domain Transform filter (18). The Domain Transform for Edge-Aware Image (and Video) Filtering is a technique introduced by Eduardo S. L. Gastal and Manuel M. Oliveira in their 2011 paper. It's designed to perform edge-aware filtering on images or videos, preserving edges and fine details while effectively reducing noise and other unwanted artifacts.

The below equation describes the local scaling factor applied to a 1D signal I

by the domain transform $ct(x)$. This scaling factor, denoted as $ct_0(x)$, plays a crucial role in determining the amount of local smoothing introduced by a filter when applied to the signal.

The equation is expressed as:

$$ct_0(x) = 1 + \frac{\sigma_r}{\sigma_s}\left|\frac{dI}{dx}\right|$$

Here's a breakdown of the components of Equation:

- $ct_0(x)$: Represents the local scaling factor applied to the signal I at position x.
- σ_s and σ_r: Parameters that control the behavior of the filter in the spatial and range domains, respectively.
- $\left|\frac{dI}{dx}\right|$: Denotes the derivative of the signal I with respect to x. This derivative captures the rate of change of the signal intensity at position x, providing information about the local image structure and gradients.

The equation indicates that the local scaling factor $ct_0(x)$ depends on the ratio $\frac{\sigma_r}{\sigma_s}$ and the magnitude of the derivative $\left|\frac{dI}{dx}\right|$ of the signal I. Essentially, higher values of $\left|\frac{dI}{dx}\right|$ correspond to regions with significant intensity changes or edges in the signal, while the ratio $\frac{\sigma_r}{\sigma_s}$ controls the relative influence of spatial and range domains on the filtering process.

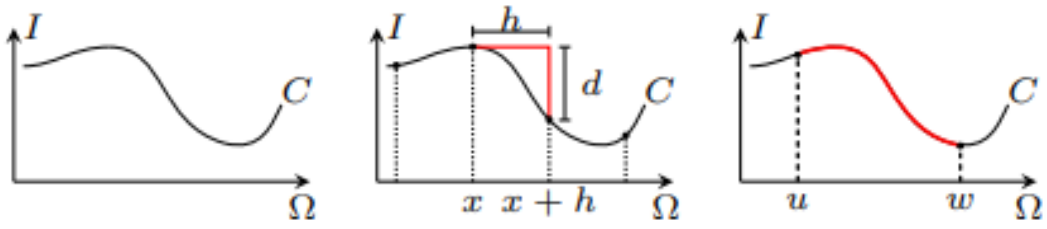


Figure 2.4: Curve C defined by the graph $(x, I(x))$, $x \in \Omega$ (left). In ℓ_1 norm, $\|(x+h, I(x+h)) - (x, I(x))\| = h + d = h + |I(x+h) - I(x)|$ (center). Arc length of C, from u to w (right).

We can understand that the local smoothing introduced by the filter is influenced by the local image structure and the balance between spatial and range(color) parameters. Higher derivative values indicate sharper transitions or edges in the signal, leading to less smoothing in those regions. Conversely, lower derivative values imply smoother regions where more smoothing can be applied.

A recursive edge-preserving filter can be defined in the transformed domain as ([19]):

$$J[n] = (1 - ad)I[n] + adJ[n - 1]$$

As the difference, denoted as d, between $ct(x_n)$ and $ct(x_{n-1})$ increases, the corresponding increment a_d approaches zero. This halts the chain of propagation, effectively preserving edges in the transformed domain (Ωw).

2.5 Depth Map Preparation

After generating a sharpness map for a single frame, the next step involves constructing a depth map of the scene using a focal stack. Here we introduced a probabilistic prediction mechanism to locate the best-focused frame for each pixel instead of traditional DFF methods which can only identify the sharpest frame among the given input frames, so their accuracy is fundamentally limited by the input sampling density. The key advantage of this probabilistic prediction is that it allows us to estimate the best-focused frame location at a sub-frame accuracy, even if that frame is not present in the sparse input focal stack (i.e., sparse sampling of the focal stack).

Here's how it works:

i) Initially, we compute a sharpness map for every frame within the focal stack. Subsequently, we normalize each pixel of the sharpness matrix along the frame axis.

ii) Now we utilizing the normalized pixel intensity as the probabilistic weight for the best-focused frame index number of each particular pixel. We proceed to calculate the weighted sum of frame index numbers. This process results in the generation of our final depth map.

2.6 Grab Cut Segmentation

The GrabCut algorithm is a method used for foreground/background segmentation in images. It was proposed by Carsten Rother, Vladimir Kolmogorov, and Andrew Blake in their paper titled "GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts" in 2004 (15). GrabCut works with a user-defined bounding box or a trimap around the object of interest. It leverages the initial information provided by the user to guide the segmentation process. A trimap divides the image into three regions: foreground, background, and unknown. Grabcut effectively segment objects from images by combining color information and spatial coherence The inner working of the algorithm is as following:

Trimap Generation: We use Otsu (14) thresholding for trimap generation. Otsu's method is a popular technique for automatically segmenting objects from the background in an image. It works by finding an optimal threshold value to separate the foreground (object) from the background based on the image's histogram. This threshold value maximizes the inter-class variance between the foreground and background pixel intensities.Once we have the optimal threshold obtained from Otsu's method, we can use it to create a trimap. A trimap is an image where pixels are labeled as either foreground, background, or unknown. Pixels within a small margin around the threshold value are considered uncertain and labeled as unknown. Pixels with intensity values above the threshold are labeled as foreground, while those below are labeled as background. I determined the unknown margin of 50 pixels through an iterative process involving trial and error. This approach allowed me to systematically evaluate various margin sizes and assess their impact on the segmentation results. After rigorous testing, the margin of 50 pixels emerged as the optimal choice, striking a balance between capturing sufficient contextual information and minimizing segmentation errors

Gaussian Mixture Model (GMM):

GrabCut models the color distribution of foreground and background pixels using a Gaussian Mixture Model (GMM). The GMM is represented by parameters such as mean ($\boldsymbol{\mu}$), covariance (Σ), and mixture coefficients ($\boldsymbol{\pi}$) for each Gaussian component.

Unary Energy (Data Term):

The unary energy term $U(\alpha, k, \theta, z)$ measures the likelihood of a pixel z belonging to the foreground or background class. It is computed based on the color similarity between the pixel and the model distributions for foreground and background. The unary energy is typically defined as the negative log-likelihood of the pixel color under the GMM.

Pairwise Energy (Smoothness Term):

The pairwise energy term $V(\alpha, z)$ encourages spatial smoothness in the segmentation mask. It penalizes abrupt changes in the labeling of neighboring pixels, promoting coherent segmentation boundaries. The pairwise energy is often defined based on the spatial proximity and color similarity between neighboring pixels.

Gibbs Energy:

The Gibbs energy $E(\alpha, k, \theta, z)$ represents the overall energy of the segmentation problem. It combines the unary and pairwise energies to capture both data fidelity and spatial coherence. The Gibbs energy is minimized to find the optimal segmentation mask.

Optimization Objective:

Given the unary and pairwise energy terms, GrabCut aims to find the optimal segmentation mask α that minimizes the Gibbs energy $E(\alpha, k, \theta, z)$. This optimization problem is typically solved iteratively using techniques like graph cuts or variational methods.

Iterative Optimization:

GrabCut iterates between updating the pixel labels (foreground or background) based on the unary energy and refining the segmentation boundaries based on the pairwise energy. During each iteration of GrabCut, the algorithm updates the foreground and background regions based on the pixel values and the likelihood of each pixel belonging to either the foreground or background. The iterative optimization process continues until convergence criteria are met, such as reaching a stable segmentation mask or minimizing the change in energy between iterations.

Output: The output of the GrabCut algorithm is a binary segmentation mask, where pixels belonging to the foreground are marked as white (or 1), and pixels be-

longing to the background are marked as black (or 0). This segmentation mask can then be used for various applications, such as object extraction, image editing, or further analysis.

2.7 Implementation Details

The entire code is written using Python programming language. I have utilized OpenCV (Open Source Computer Vision Library) for implementing key components of my proposed method, including optical flow calculation, Otsu's thresholding, and GrabCut. The Python implementation of the proposed method is structured into four major blocks as follows:

Optical Flow: I have used OpenCV's cv2.calcOpticalFlowPyrLK() function for optical flow estimation. One challenge we encountered was that defocus alters the appearance of each frame differently depending on the focus settings. Running an optical flow algorithm between each frame and the reference may fail as frames that are far from the reference in the focal stack appear vastly different. We overcome this problem by concatenating flows between consecutive frames in the focal stack, which ensures that defocus between two input frames to the optical flow appear similar.

The primary output of cv2.calcOpticalFlowPyrLK is the 2D flow field, which is typically stored as a 2- channel (2D) numpy array. The two channels represent the horizontal and vertical components of the motion vectors, providing information about the direction and magnitude of pixel motion. Finally I used cv2.remap function in OpenCV for image remapping, which involves transforming the pixels of an image from one location to another based on a mapping function(optical flow).

Sharpness Map Preparation: I personally designed this part of the code, incorporating functions from OpenCV and SciPy for tasks such as Gaussian blurring, gradient calculation and entropy calculation of images.

To prepare a sharpness map for each frame in the focal stack, the process is as follows: First, the input image is smoothed using Gaussian blur, and then the

gradient magnitude of the smoothed image is computed. The image is analyzed at multiple scales using a scale pyramid. For each scale, DCT coefficients are computed for image patches, and high-frequency components are extracted. The frequency bands are used to identify areas with significant high-frequency content. Sharpness is quantified by computing the energy of this high-frequency content. Additionally, the local entropy of the frequency content is calculated to weigh the significance of different regions.

Sharpness Map Smoothening: To smooth and remove noise from the obtained sharpness map, I implemented the domain transform edge-preserving filtering technique, as described by Eduardo S. L. Gastal and Manuel M. Oliveira in their 2011 paper. Specifically, I implemented equation 14 from their paper.

Trimap And GrabCut In my implementation, the generation of the trimap is achieved using Otsu's thresholding method, which is facilitated by the OpenCV library.

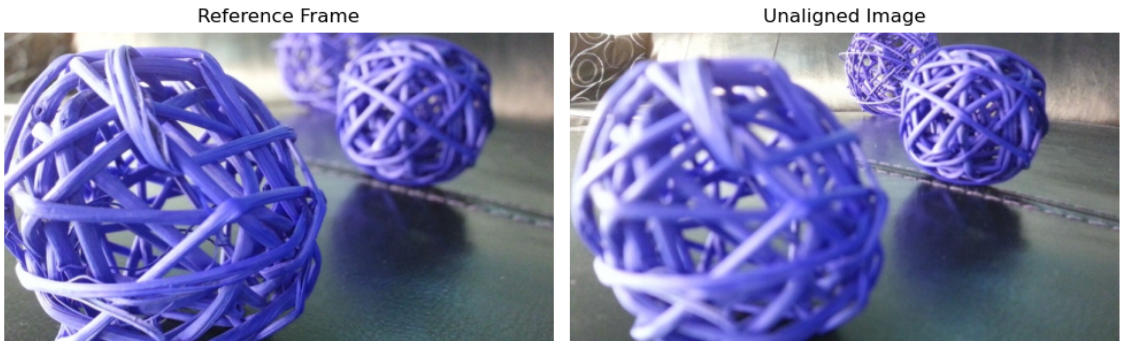
Following the trimap generation, I employed the GrabCut algorithm, also implemented using OpenCV, to refine the segmentation. The GrabCut algorithm iteratively improves the trimap by modeling the foreground and background as Gaussian Mixture Models (GMMs) and then using graph cuts to minimize the segmentation energy function. This process results in a more accurate delineation of the object boundaries, providing a high-quality segmentation output.

Chapter 3

Results and Discussion

3.1 Optical Flow

Below is a visualization of a properly aligned frame. Pay close attention to the top left corner of both the reference and unaligned images. Here, you can observe noticeable magnification changes. Misalignment caused by these changes can introduce artifacts into the final depth map. These artifacts might manifest as ghosting, blurring, or misregistered features, significantly affecting the quality of the depth map.



The images depicted below are aligned frames. No magnification changes are present, and each pixel in the reference frame perfectly corresponds to its counterpart in the aligned image frame.

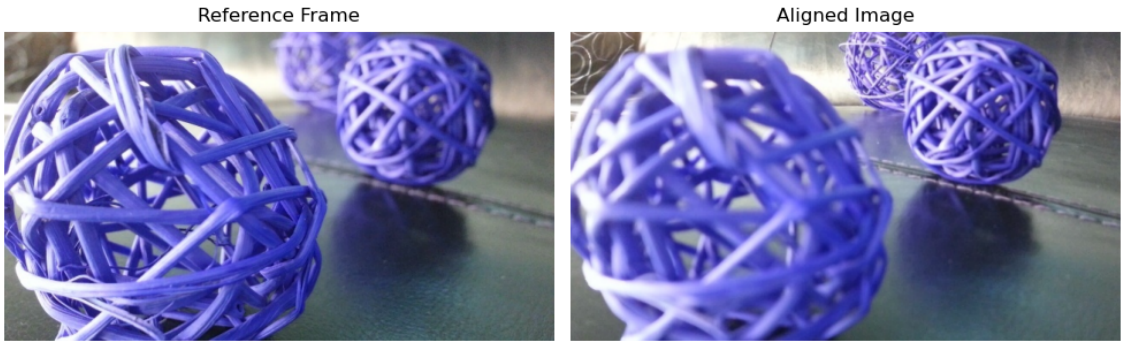


Figure 3.1: Unaligned(above) and Aligned(below) images of a focal stack

3.2 Sharpness Map

In our work we used the gradient magnitude of the image instead of the original image for the Discrete Cosine Transform (DCT). Here's how it was useful:

i)Edge Preservation: The gradient magnitude emphasizes edges and transitions in the image. By using it instead of the original image, the DCT transform focuses more on preserving edge information, which can be crucial for tasks like edge-aware filtering or feature extraction.

ii)Noise Robustness: Since gradients capture local changes in intensity, they are less affected by noise compared to the original image. Using the gradient magnitude can lead to a more robust representation, particularly in noisy environments.

iii)Dimensionality Reduction: Gradients provide a compact representation of image structure compared to the original image. Utilizing gradient magnitude for the DCT transform in patches can lead to a more efficient representation, especially when dealing with high-dimensional data.

iv)Feature Extraction: The gradient magnitude highlights areas of significant change in the image, making it useful for feature extraction tasks. By incorporating gradient information into the DCT transform, the resulting representation may capture important features more effectively.

The Gradient magnitude of the image looks like as following:

Figure 3.2: Gradient Magnitude of the image

The proposed DCT-based sharpness measure algorithm receives the gradient magnitude of the original image as input. The image below represents the raw sharpness map derived from the input image. The presence of noise in the map can be attributed to inherent noise in the image as well as non-textured regions.

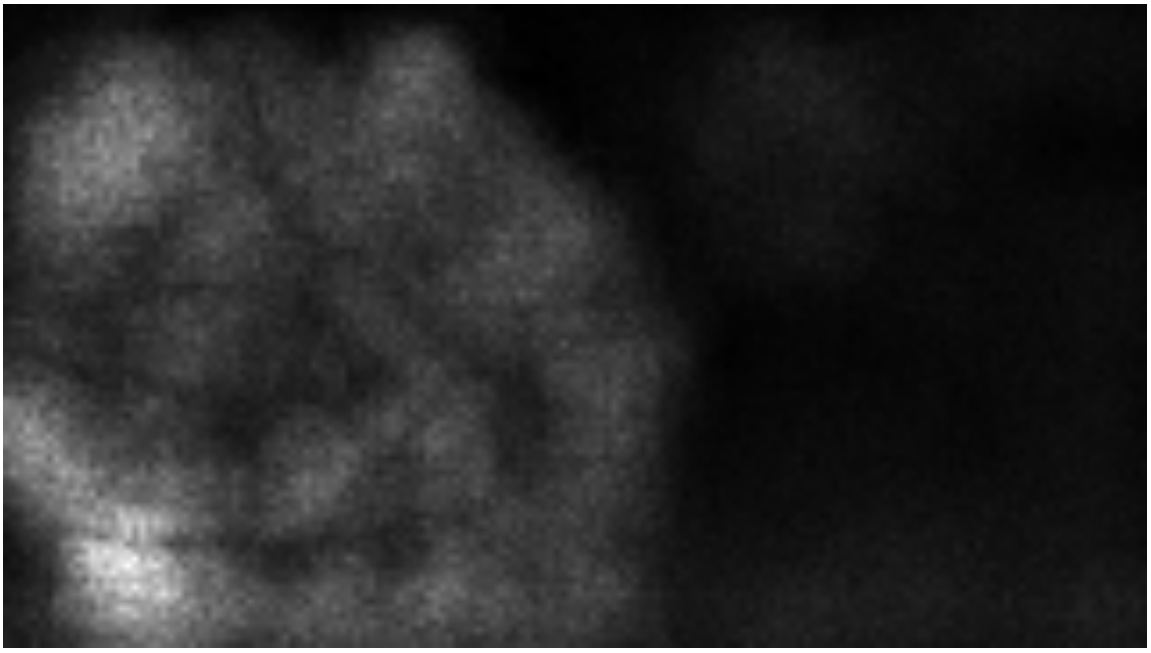


Figure 3.3: Unfiltered Sharpness Map

The below is the filtered image. The Domain Transform for Edge-Preserving applies a transformation to the image domain based on local image properties, such as gradient magnitude or color differences, to adjust the filter's behavior dynamically across different regions of the image. This approach ensures that regions with sharp transitions or edges are preserved while smoother areas are effectively smoothed.

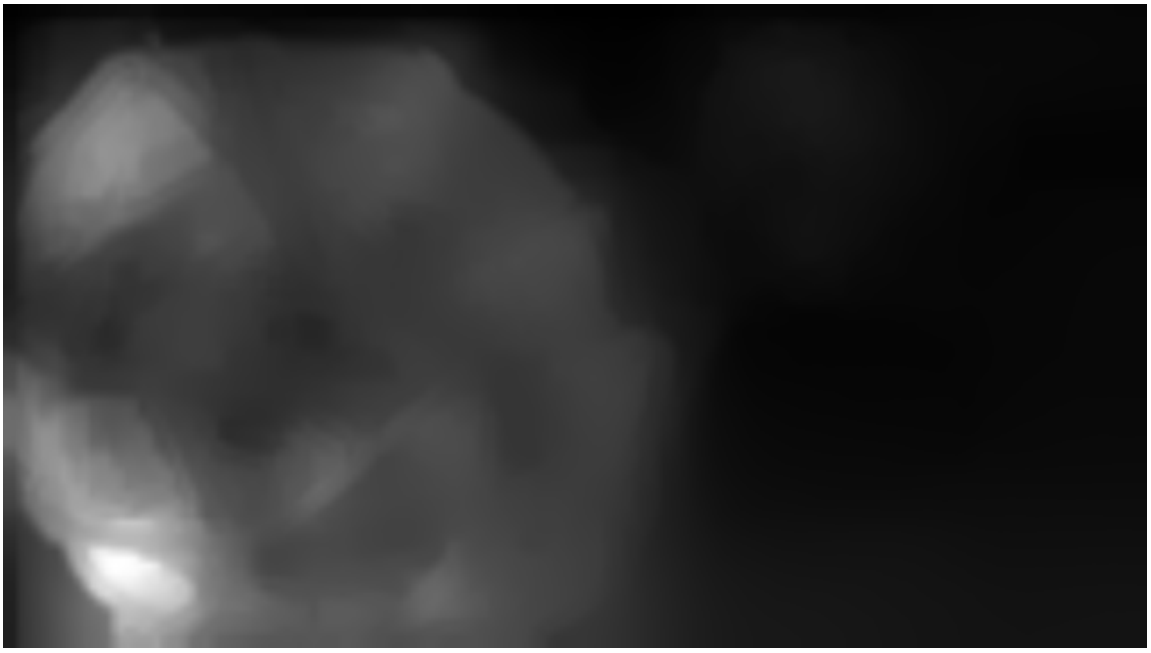


Figure 3.4: Smoothened Sharpness Map after applying Domain Transform edge preserving filter

3.3 Depth Map Preparation

The focal stack intended for generating the depth map is depicted below.

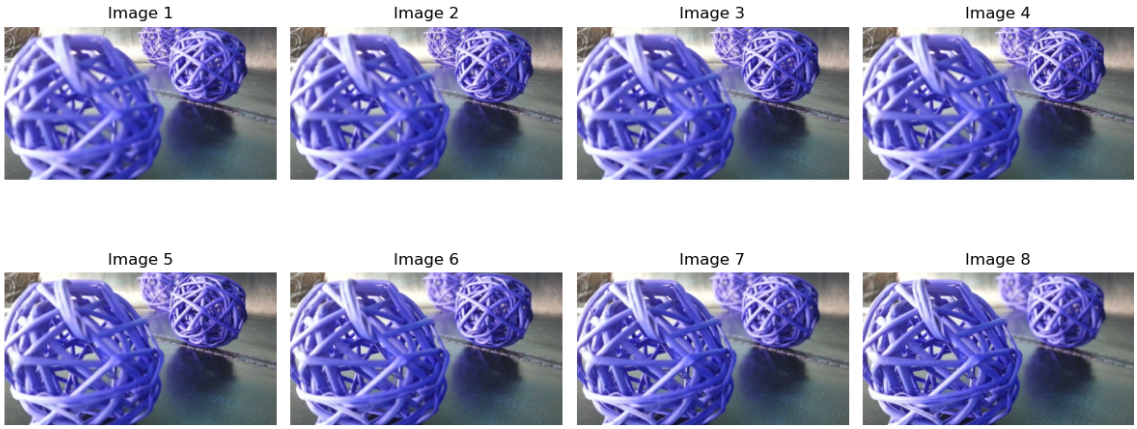


Figure 3.5: Focal Stack

The image below illustrates the expected appearance of a depth map when employing the traditional DFF method, which selects the sharpest pixel from the focal stack for depth map estimation. The resulting depth map may exhibit inaccurate contours.

Figure 3.6: Argmax Depth Map

The following image depicts the depth map generated using the weighting method outlined in the methodology section. The advantage of this approach lies in its ability to yield satisfactory results even with sparsely sampled focal stacks. By employing the weighting method, the requirement for a densely sampled focal stack is circumvented.

Figure 3.7: weighted Depth Map

Figure 3.8 displays the depth map outcomes for the Mobile depth dataset using several deep learning methods, excluding MobileDFF, which is an optimization-based approach that utilizes the entire focal stack, spanning from 14 to 30 images per stack. In our study, we utilized only 7 frames.
Figures 3.9 to 3.11 depict the depth maps generated by our algorithm. Notably, these depth maps exhibit comparable performance to those produced by deep learning methods. Specifically, in Figure 3.10, our algorithm demonstrates precise differentiation between the ball and the background tree. Moreover, even in the background, our algorithm effectively separates the tree from the house. In contrast, the trained model displays a stark distinction between the foreground ball and background tree,

failing to distinguish the background tree from the background house.

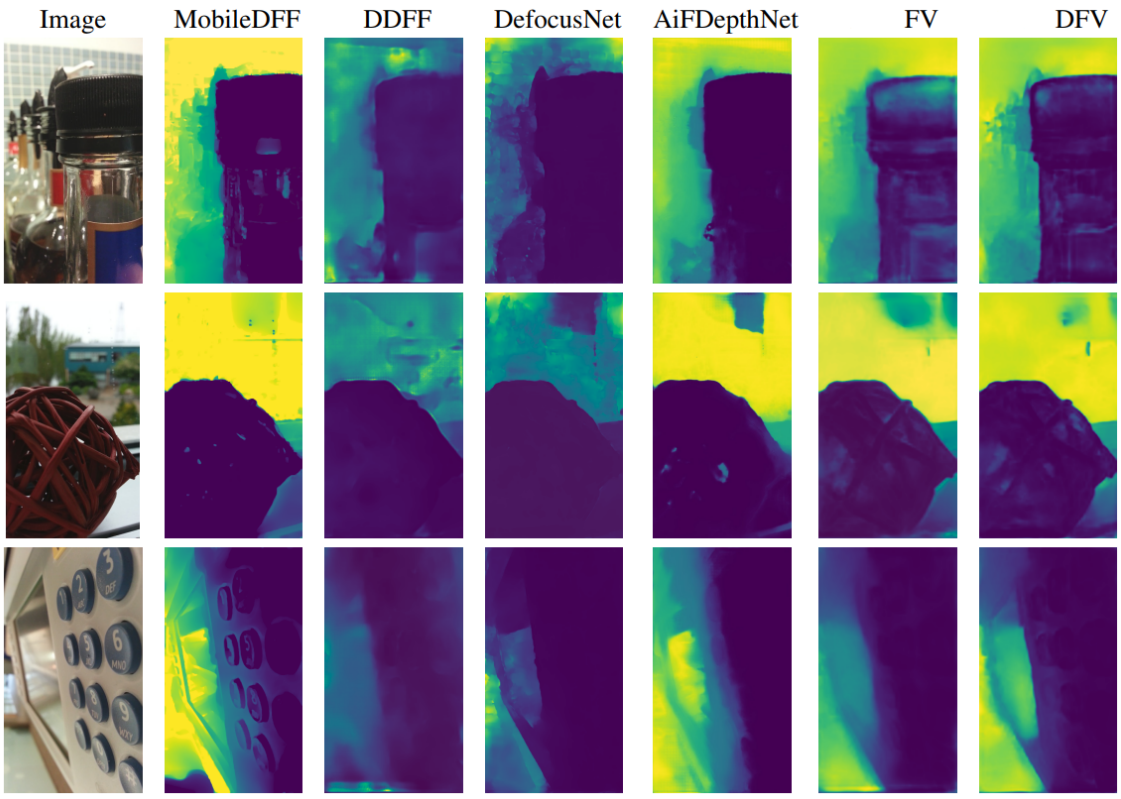


Figure 3.8: Qualitative results on Mobile depth dataset. The warmer the color, the larger the depth value.

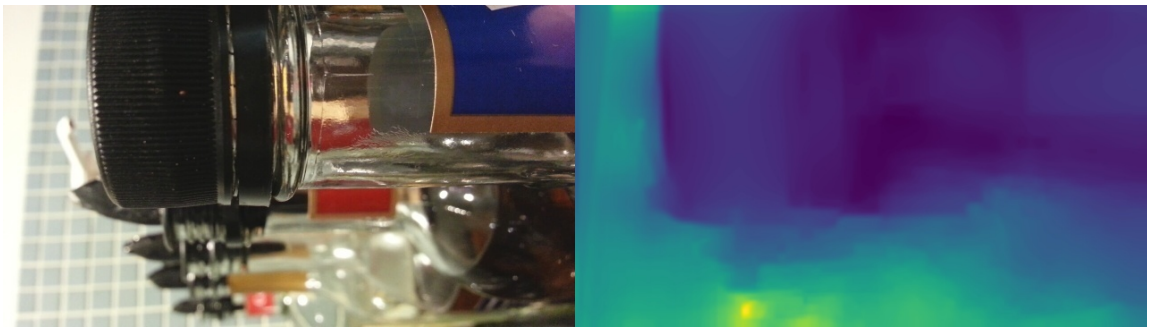


Figure 3.9: Bottle

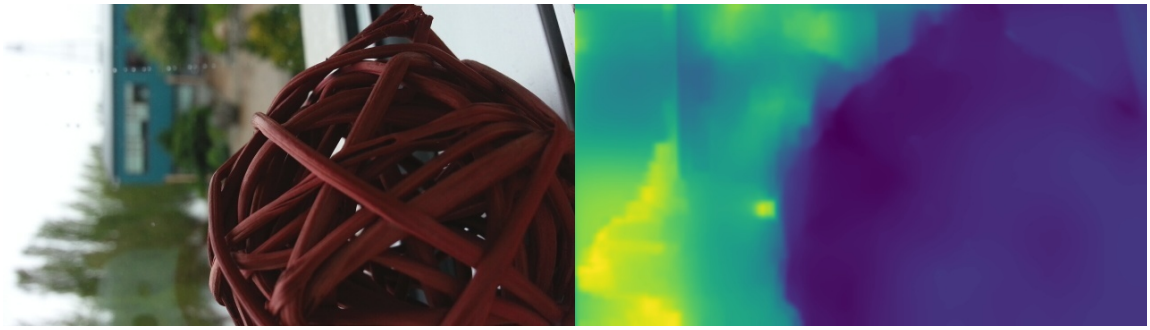


Figure 3.10: Window

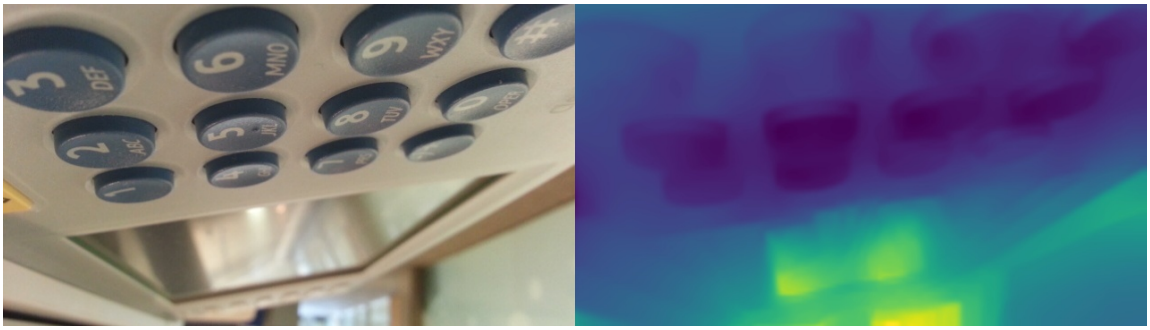


Figure 3.11: Telephone

3.4 Grab Cut

Figure 3.12 below illustrates a trimap generated from intensity values extracted from the depth map, with certain regions left as unknown. This trimap serves the purpose of delineating the foreground from the background, with areas of uncertainty indicated as unknown. However, he GrabCut algorithm, when provided with a trimap and original image, segments the image into foreground and background regions based on the initial user guidance. However, due to factors such as image noise or inaccuracies in the trimap, the resulting mask from GrabCut may contain small isolated regions or noise artifacts as we can see in the figure 3.13. To clean up this mask and obtain a more accurate segmentation, we can employ the connected components algorithm. Connected components identifies connected regions of pixels with the

same label or intensity, allowing us to distinguish between meaningful regions and noise. By removing small connected components or regions that do not correspond to the desired foreground or background, we effectively eliminate noise from the segmentation mask. This post-processing step enhances the quality of the segmentation results and ensures a smoother and more refined output.

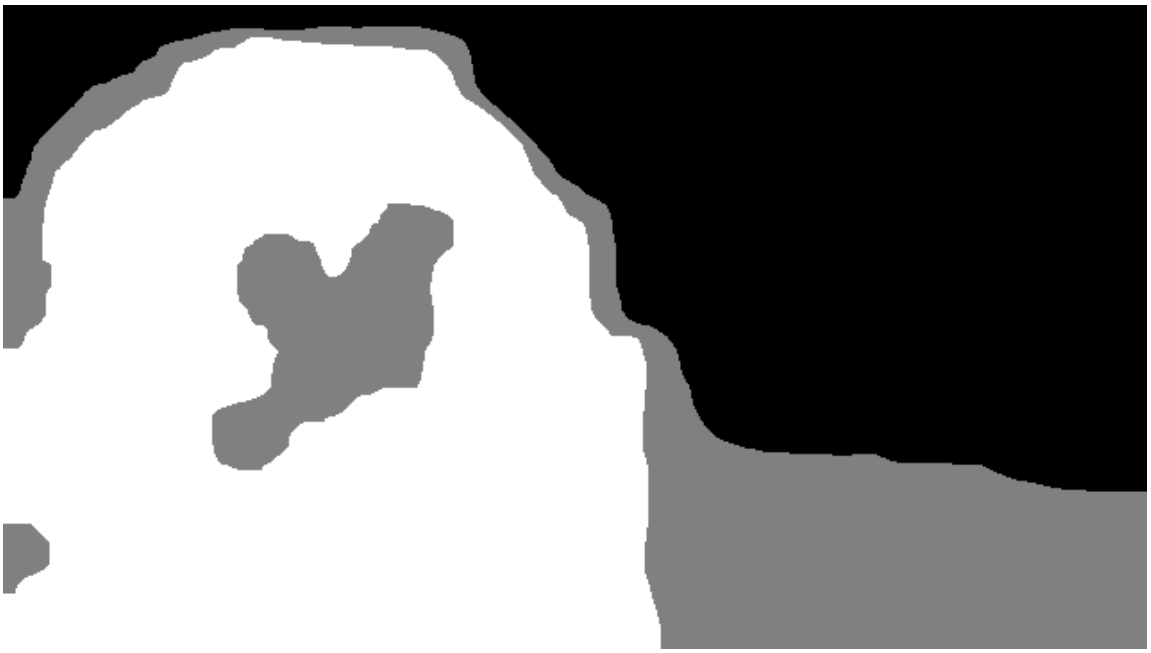


Figure 3.12: Tri-Map

Figure 3.13: Raw binary mask generated by GrabCut

Figure 3.14: Processed and Clean binary mask

The image below shows the segmented salient object obtained by utilizing the clean noise free binary mask along with the original RGB image.

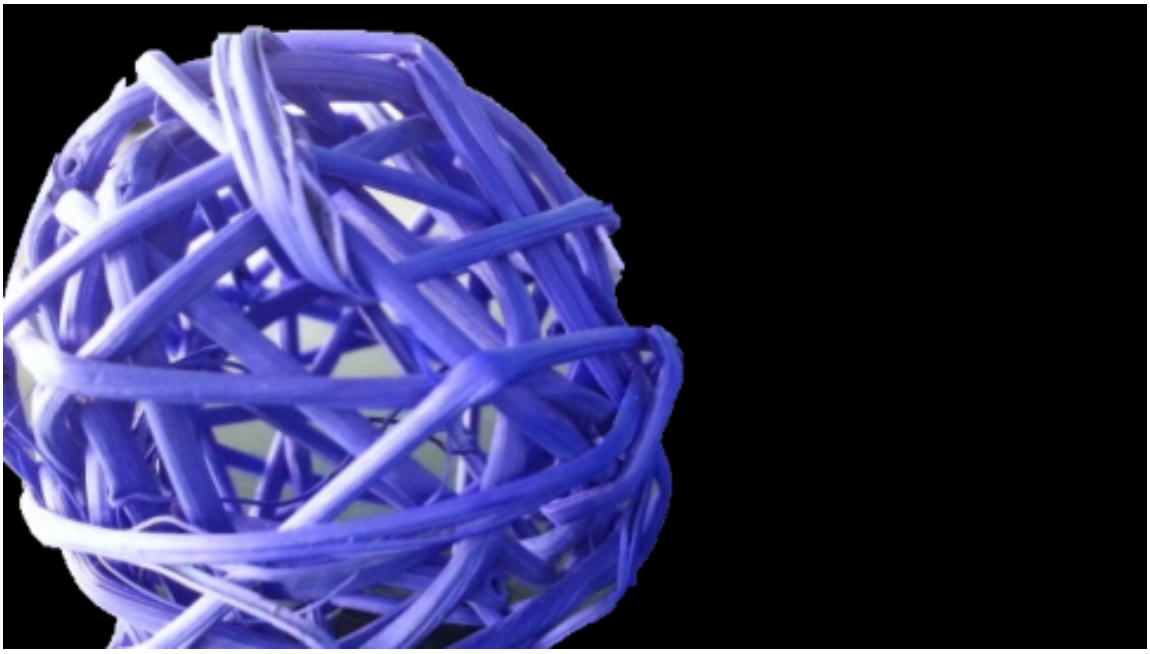


Figure 3.15: Segmented Image

3.5 Conclusion

Chapter 4

Conclusion

This thesis has presented a novel approach for generating depth maps from focal stacks and leveraging the derived depth information for salient object detection and binary segmentation. The proposed method effectively addresses several challenges associated with depth estimation from defocused images, including handling textureless regions, achieving accurate depth estimation with sparse sampling, and preserving edge details during filtering. The key contributions of this work can be summarized as follows:

i) A multi-scale Discrete Cosine Transform (DCT) based focus measure that incorporates contextual information to robustly handle textureless regions within the images. ii) An entropy-weighted sharpness map that enhances the prominence of salient regions while suppressing noise and insignificant variations. iii) A probabilistic depth map estimation technique that enables sub-frame accuracy estimation, alleviating the need for densely sampled focal stacks. iv) An adaptive thresholding approach based on Otsu's method for generating trimaps, which are subsequently refined using the GrabCut algorithm to produce high-quality binary segmentation masks. v) Effective handling of noise and artifacts in the segmentation masks through post-processing techniques, such as connected components analysis.

The experimental results demonstrated the efficacy of the proposed method in generating accurate depth maps and segmenting salient objects from challenging real-world scenarios. The depth maps produced by the algorithm exhibited comparable

or superior performance to those generated by deep learning methods, particularly in scenarios involving complex backgrounds or intricate object boundaries. While this work has made significant strides in leveraging depth information from focal stacks for salient object detection and segmentation, there remain avenues for further exploration and improvement. Future research could investigate the incorporation of additional cues, such as motion or semantic information, to enhance the robustness and accuracy of the depth estimation process. Additionally, exploring ways to extend the proposed method to video sequences or real-time applications could broaden its practical applicability. Overall, this thesis has contributed to the field of computer vision and image processing by presenting a novel and effective approach for depth map generation and salient object segmentation from focal stacks. The proposed method demonstrates promising results and opens up new possibilities for leveraging depth information in various applications.

Bibliography

[1] Yang, Fengting, Xiaolei Huang, and Zihan Zhou. "Deep depth from focus with differential focus volume." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12642-12651. 2022.

[2] Suwajanakorn, Supasorn, Carlos Hernandez, and Steven M. Seitz. "Depth from focus with your mobile phone." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3497-3506. 2015.

[3] Favaro, Paolo, Andrea Mennucci, and Stefano Soatto. "Observing shape from defocused images." International Journal of Computer Vision 52 (2003): 25-43.

[4] Nair, Hari N., and Charles V. Stewart. "Robust focus ranging." In Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 309-310.

[5] Pentland, Alex Paul. "A new sense for depth of field." IEEE transactions on pattern analysis and machine intelligence 4 (1987): 523-531. IEEE Computer Society, 1992

[6] Schonberger, Johannes L., and Jan-Michael Frahm. "Structure-from-motion revisited." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4104-4113. 2016.

[7] Garg, Ravi, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. "Unsupervised cnn for single view depth estimation: Geometry to the rescue." In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands,

October 11-14, 2016, Proceedings, Part VIII 14, pp. 740-756. Springer International Publishing, 2016.

[8] Fan, Tiantian, and Hongbin Yu. "A novel shape from focus method based on 3D steerable filters for improved performance on treating textureless region." Optics Communications 410 (2018): 254-261.

[9] Moeller, Michael, Martin Benning, Carola Schönlieb, and Daniel Cremers. "Variational depth from focus reconstruction." IEEE Transactions on Image Processing 24, no. 12 (2015): 5369-5378.

[10] Ahmed, Nasir, $\mathrm{T}_N atarajan, and Kamisetty R. Rao."Discrete cosine transform."IEEE transactio$
$90-93. Brox, Thomas, Andrés Bruhn, Nils Papenberg, and Joachim Weickert."High accuracy opti$
$ECCV 2004 : 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-$
$14, 2004. Proceedings, Part IV 8, pp. 25 - 36. Springer Berlin Heidelberg, 2004.$

[12] Jeon, Hae-Gon, Jaeheung Surh, Sunghoon Im, and In So Kweon. "Ring difference filter for fast and noise robust depth from focus." IEEE Transactions on Image Processing 29 (2019): 1045-1060.

[13] Gastal, Eduardo SL, and Manuel M. Oliveira. "Domain transform for edge-aware image and video processing." In ACM SIGGRAPH 2011 papers, pp. 1-12. 2011.

[14] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." Automatica 11, no. 285-296 (1975): 23-27.

[15] Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. "" GrabCut" interactive foreground extraction using iterated graph cuts." ACM transactions on graphics (TOG) 23, no. 3 (2004): 309-314.

[16] Shi, Jianping, Li Xu, and Jiaya Jia. "Discriminative blur detection features." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2965-2972. 2014.

[17] Yi, Xin, and Mark Eramian. "LBP-based segmentation of defocus blur." IEEE transactions on image processing 25, no. 4 (2016): 1626-1638.

[18] Gastal, Eduardo SL, and Manuel M. Oliveira. "Domain transform for edge-aware image and video processing." In ACM SIGGRAPH 2011 papers, pp. 1-12. 2011.

[19] Smith, Julius Orion. Introduction to digital filters: with audio applications. Vol. 2. Julius Smith, 2007.

[20] Ahmad, Muhammad Bilal, and Tae Sun Choi. "Application of three dimensional shape from image focus in LCD/TFT displays manufacturing." IEEE Transactions on Consumer Electronics 53, no. 1 (2007): 1-4.

[21] Nayar, Shree K., and Yasuo Nakagawa. "Shape from focus." IEEE Transactions on Pattern analysis and machine intelligence 16, no. 8 (1994): 824-831.

[22] Xie, Hui, Weibin Rong, and Lining Sun. "Wavelet-based focus measure and 3-d surface reconstruction method for microscopy images." In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 229-234. IEEE, 2006.

[23] Krotkov, Eric. "Focusing." International Journal of Computer Vision 1, no. 3 (1988): 223-237

[24] Schechner, Yoav Y., and Nahum Kiryati. "Depth from defocus vs. stereo: How different really are they?." International Journal of Computer Vision 39 (2000): 141-162.

[25] Mannan, Fahim, and Michael S. Langer. "What is a good model for depth from defocus?." In 2016 13th Conference on Computer and Robot Vision (CRV), pp. 273-280. IEEE, 2016.

[26] Bruhn, Andrés, Joachim Weickert, and Christoph Schnörr. "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods." International journal of computer vision 61 (2005): 211-231.