

DEPARTMENT OF COMPUTER SCIENCE MID-TERM EXAM

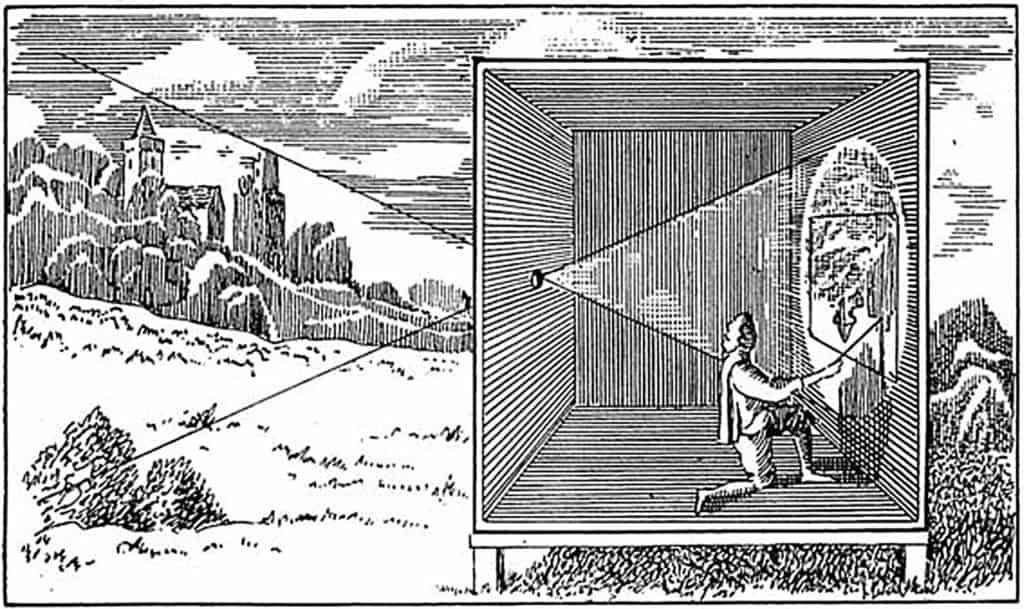
|  |  |
| --- | --- |
| Class: | **CPS564 - Visual Computing & Mixed Reality Summer 2024** |
| Instructor: | **Dr. Tam Nguyen** |
| Date: | 25 june |
| Student’s Name: | Jenil Sanjaykumar Pandya |
| Student’s Email: | Pandyaj2@udayton.edu |

***You can refer to the materials on Canvas. However, using ChatGPT, WhatsApp, Facebook Messenger, or similar tools is not allowed. Plagiarism and cheating will not be tolerated. Any student caught plagiarizing or cheating will not pass the class.***

# Instruction (Very Important)

* Write your name and email on the first page of your answer script.
* There is no teamwork in the exam.
* You can use materials on Canvas.
* Using ChatGPT, WhatsApp, Facebook Messenger, or similar tools is not allowed.
* Do not leave any question unanswered.
* You submit your answer script (.doc, .docx, or .pdf file) to Canvas. If you have multiple files, please compress them to a .zip file and submit it to Canvas.

# Question 1



What is the person **doing** in the image above? Write your answer into the box below **(10 points).**

He is colouring the reflection projected by the lens of his camera, since we didn't have color images in the past, he is producing a coloured photograph without a traditional camera or film. It is the light, refracting through his lens onto a canvas, that he captured and outlined the lines of the landscape which he saw. After that, he colours it as per the actual hues existing in the landscape. he makes use of a lens to project an inverted image of the outside world onto a flat canvas inside a dark box. Even though the device does not capture images permanently by itself, it can be used to make very realistic portraits of scenes. Sometimes even artists of this time use this device to compose very realistic paintings.

# Question 2

How was the photo below taken, i.e., aperture and shutter speed?



Give your answer to the box below **(10 points).**

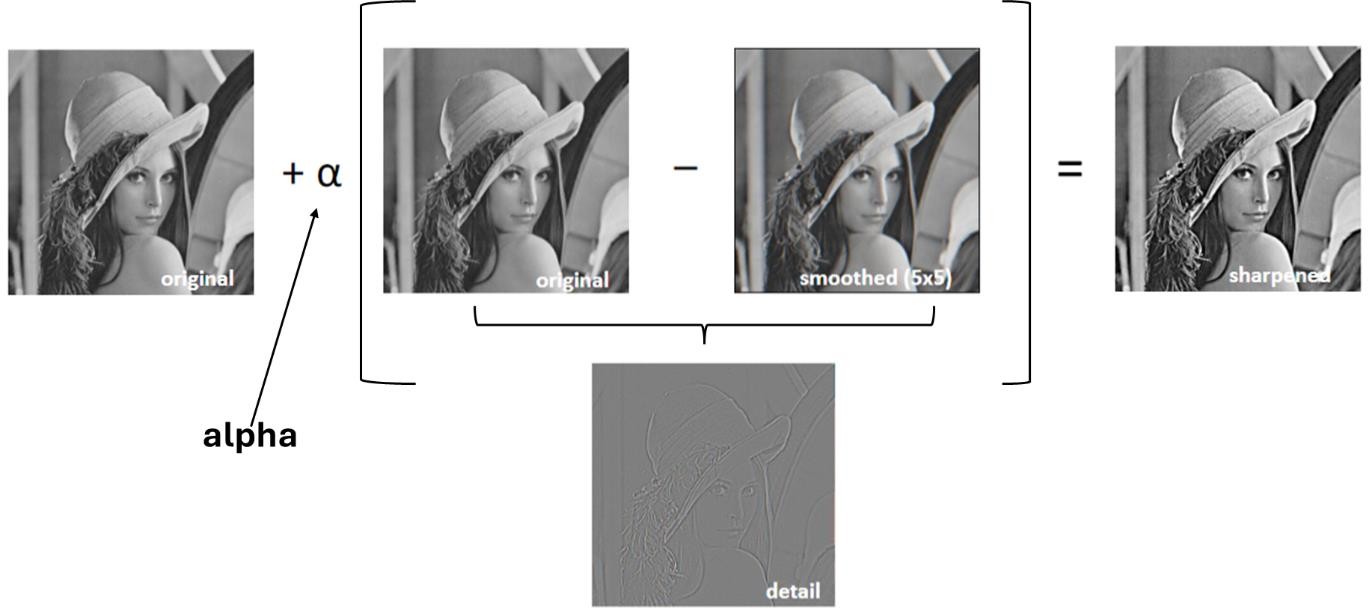
This Picture is of a very fast-moving car that can only be seen by a large aperture camera, because the camera must have large aperture it captures the lightest that any other cameras can, that is why because of a large aperture it is showing blurred background and the car in the sharp focus

Because large aperture allowed lighter to get into the camera lens it can capture the car in low light condition and good focus with a blurred background.

The fact that car is in focus shutter speed must have been high, photographer must have taken many pictures to take sharp car image, so the shutter speed is also fast otherwise slow shutter speed car would have been blur, so according to me the shutter speed must be around 250 pictures per second

# Question 3

Write your MATLAB code to perform the sharpening filter with **alpha**. Please refer to the image below for the demonstration of **alpha**.



Note: you smooth the original image with **Gaussian filter (with the kernel size 5x5)**. Write your MATLAB code into the box below **(20 points)**.

clear all;

close all;

clc;

% Read the mountain image

img = imread('mountain.jpg');

figure, imshow(img);

title("original Image");

% Define Gaussian filter kernel as instructed in the problem

gaussian\_kernel = fspecial('gaussian', [5 5], 1);

% Apply Gaussian filter to smooth the image as it is mentioned

img\_smooth = imfilter(img, gaussian\_kernel, 'replicate');

figure, imshow(img\_smooth);

title("Gaussian Image");

% using 1.5 to submit and 0.5 and 1 for other questions

alpha = 1.5;

% Sharpen the image using the formula: it can be seen in the question

img\_sharp = (1 + alpha) \* double(img) - alpha \* double(img\_smooth);

% Convert the sharpened image back to uint8 otherwise no good result

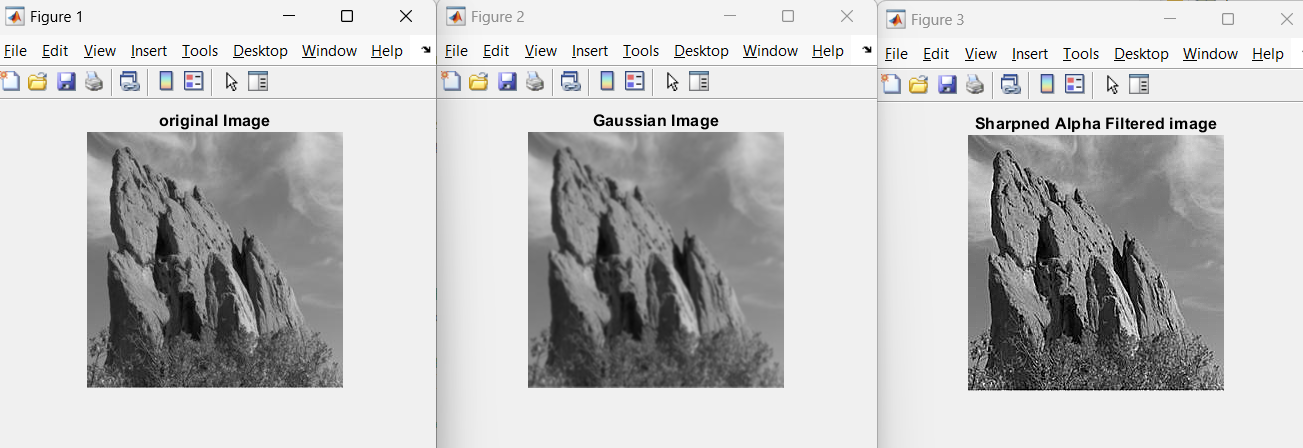
img\_sharp = uint8(img\_sharp);

% Display the final sharpened image

figure, imshow(img\_sharp);

title("Sharpned Alpha Filtered image")

%original Output Pasted below



Use your code to test on **mountain.jpg** with 2 alpha values: 0.5 and 1.

A sharped alpha 2 rock formation

Description automatically generated

Show the output images to the box below **(10 points)**.

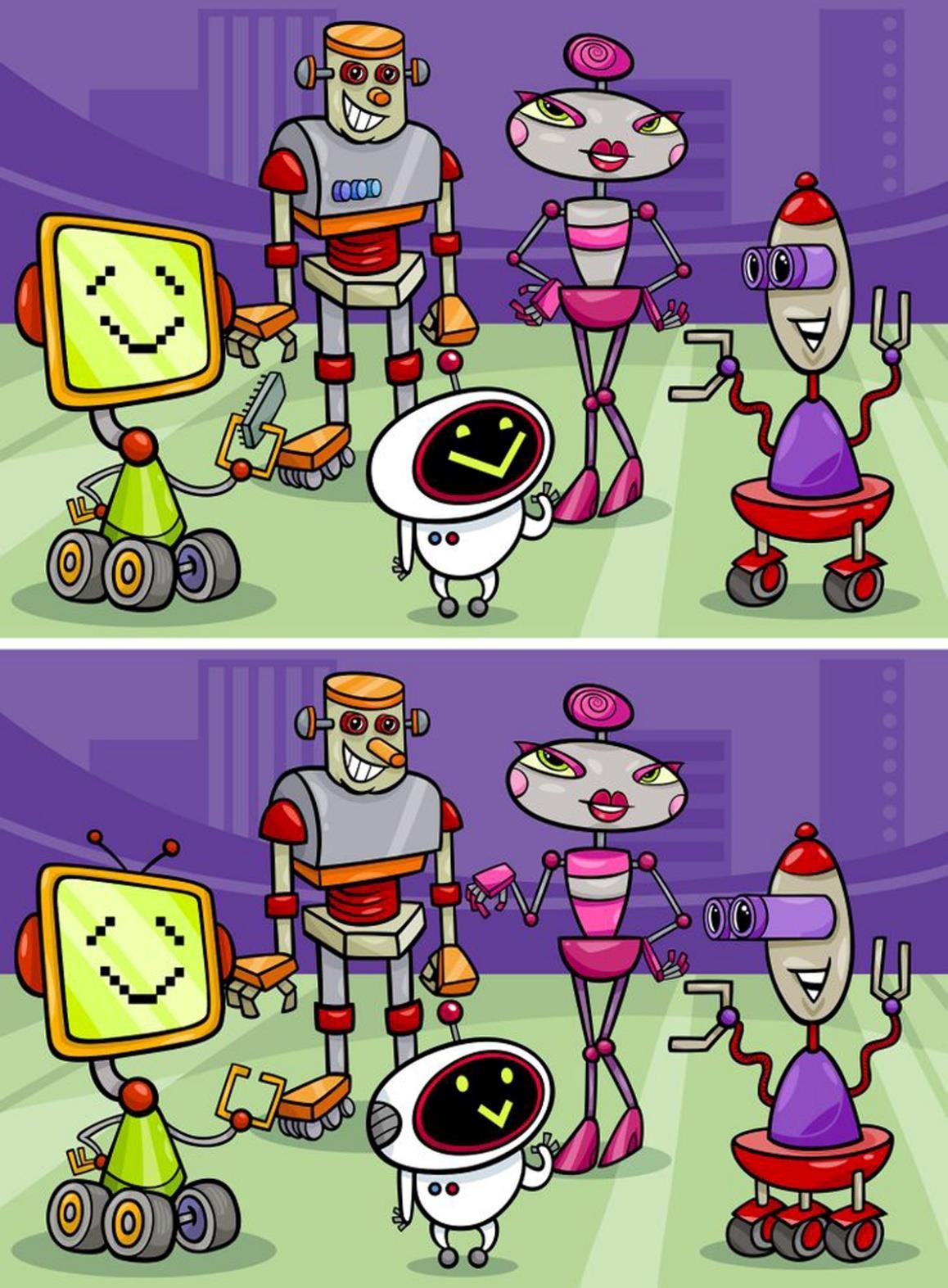
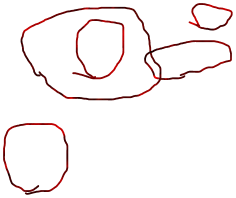
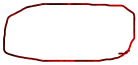
A sharped alpha 0. 5

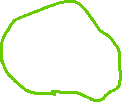
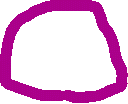
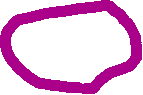
Description automatically generated

|  |  |
| --- | --- |
| **alpha = 0.5** | **alpha = 1** |
|  | A close-up of a rock formation  Description automatically generated |

# Question 4

Find 10 differences between two images below. Circle the differences **(10 points).**





**Explain how you did it. Is it bottom-up or top-down saliency? Why? (10 points)**

**Write your answer to the box below.**

I used Top Town Saliency for this to find the differences, considering top image to be original and bottom one to have differences and found all ten by comparing each robot in both images. Overall Scan: First, I scanned both photographs to gain a basic understanding of the situation.

Detail Comparison: Next, I compared various areas one by one (for example, robot faces, limbs, and wheels).

Spotting Differences: I noticed differences by focusing on one robot at a time and comparing the appropriate parts.

It is top-down Saliency because I’m using the knowledge from the past means original image to get the difference between the new image, the original image is guiding me

Only noticing the similar parts of the image like robots and their features to find the difference between them

# Question 5

***Write your MATLAB code to extract Baymax from the image baymax.png.This image can be found on Canvas.***

***Hint: use saliency code to extract the saliency map. From the saliency map, you compute the binary map. Then, you extract Baymax from the binary map.***

|  |  |
| --- | --- |
| **Input (baymax.png)** | **Expected Output** |
|  | A white cartoon character with black eyes  Description automatically generated |

Write your MATLAB code to the box below **(30 points)**.

% Clearing workspace and closing figures

close all;

clear all;

clc;

% Reading the image

inputImage = imread('baymax.png');

figure, imshow(inputImage);

title("Original Baymax");

% Converting RGB to LAB color space

labOutput = rgb2lab(inputImage);

% Extracting lightness, a, and b channels

lightnessChannel = labOutput(:,:,1);

aChannelOutput = labOutput(:,:,2);

bChannelOutput = labOutput(:,:,3);

% Calculating mean values for each channel

lightnessMeanValue = mean2(lightnessChannel);

aMeanValue = mean2(aChannelOutput);

bMeanValue = mean2(bChannelOutput);

% Calculating saliency map

saliencyMapOutput = (lightnessChannel - lightnessMeanValue).^2 + ... % Compute saliency map

(aChannelOutput - aMeanValue).^2 + ...

(bChannelOutput - bMeanValue).^2;

% Thresholding the saliency map

thresholdValue = mean(saliencyMapOutput(:));

saliencyMapOutput(saliencyMapOutput < thresholdValue) = 1; % Threshold to create binary map

saliencyMapOutput(saliencyMapOutput >= thresholdValue) = 0;

% Applying saliency map

for channelIndex = 1:3

outputImage(:,:,channelIndex) = inputImage(:,:,channelIndex) .\* uint8(saliencyMapOutput); % Apply binary mask to each channel

end

% Displaying the output image

figure, imshow(outputImage);

title("Extracted Baymax");