

DEPARTMENT OF COMPUTER SCIENCE FINAL EXAM

Summer 2024

|  |  |
| --- | --- |
| Class: | **Visual Computing and Mixed Reality** |
| Instructor: | **Dr. Tam Nguyen** |
| Date: | 1 august 2024 |
| 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), .m files and .blend file to Canvas. If you have multiple files, please compress them to a .zip file and submit it to Canvas.

# Problem 1

1. Why do we need to do feature extraction? **(5 points) Answer:**

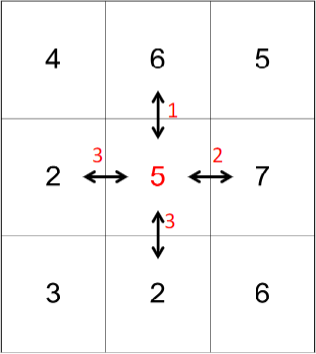
We need to do feature extraction to identify and highlight important parts of an image. By focusing on key elements, we can make photos more interesting and visually appealing. It helps in creating a good composition by using techniques like the Golden Ratio and the Rule of Thirds, making sure the photo looks balanced and professional. Feature extraction simplifies the process of identifying and emphasizing the main subjects, leading to better aesthetic outcomes. It helps distinguishing the image and provides useful for object recognition, robot navigation and other things.

1. Is histogram of a grayscale image a good feature? Explain why. **(5 points) Answer:**

Since histogram gives representation for the Numerical Data in graphical way, numerical data for an image is it’s RGB values. Since we have a Grayscale image, yes, a histogram of a grayscale image is a good feature. It shows how pixel values are distributed, which helps in understanding the overall contrast and brightness. This can be useful for identifying patterns and ensuring balanced exposure in a photo. By analyzing the histogram, we can adjust improve the aesthetic quality of the image, making sure it looks pleasing and well-composed.

# Problem 2

In Seam Carving, the first step is to compute the importance map. Below is an example of computing the importance value.



The importance value of a pixel is the sum of the difference of this pixel to the four neighboring pixels (left, right, upper, and lower). In the example above, the importance value is 3 + 2 + 1 + 3 = 9.

You are asked to write a MATLAB function to compute the importance map of an input grayscale image. The input is a grayscale image, and the output is the importance map. You need to submit the

.m file as well **(15 points) Answer:**

function importanceMap = Importance(grayscaleImage)

% Get the size of the grayscale image

[height, width] = size(grayscaleImage);

% Initialize the importance map with zeros

importanceMap = zeros(height, width);

% Loop over each pixel (excluding border pixels)

for i = 2:height-1

for j = 2:width-1

% Get the current pixel and its neighbors

currentPixel = grayscaleImage(i, j);

leftPixel = grayscaleImage(i, j-1);

rightPixel = grayscaleImage(i, j+1);

upperPixel = grayscaleImage(i-1, j);

lowerPixel = grayscaleImage(i+1, j);

% Compute the importance value for the current pixel

importanceValue = abs(currentPixel - leftPixel) + ...

abs(currentPixel - rightPixel) + ...

abs(currentPixel - upperPixel) + ...

abs(currentPixel - lowerPixel);

% Store the importance value in the map

importanceMap(i, j) = importanceValue;

end

end

% Border pixels are left as zero

end

% Read a grayscale image (or convert a color image to grayscale)

img = imread('singapore.jpg');

if size(img, 3) == 3

img = rgb2gray(img); % Convert to grayscale if it's a color image

end

% Compute the importance map

importanceMap = Importance(img);

% Display the original grayscale image and the computed importance map

figure;

subplot(1, 2, 1); % Create a subplot with 1 row and 2 columns, active on the first plot

imshow(img); % Show the original grayscale image

title('Original Grayscale Image'); % Add a title to the first plot

subplot(1, 2, 2); % Switch to the second plot

imshow(importanceMap, []); % Show the importance map

title('Importance Map'); % Add a title to the second plot

# Problem 3

1. Why is VR popularly used for flight simulation, but AR is not? **(5 points) Answer:**

AR for airplane driving simulation would not be useful because , We cant create a realistic reality with it like Virtual Reality Does. VR is popular for flight simulation because it creates a fully immersive environment. Pilots can experience realistic controls, scenery, and conditions, enhancing their training experience. AR, on the other hand, overlays digital elements onto the real world, which isn't as effective for creating the complete, controlled environments needed for flight training. VR's ability to simulate various scenarios and conditions makes it the better choice for flight simulation.

1. What are common issues in VR? Name 4 of them. Please suggest your solution to overcome these issues? **(5 points)**

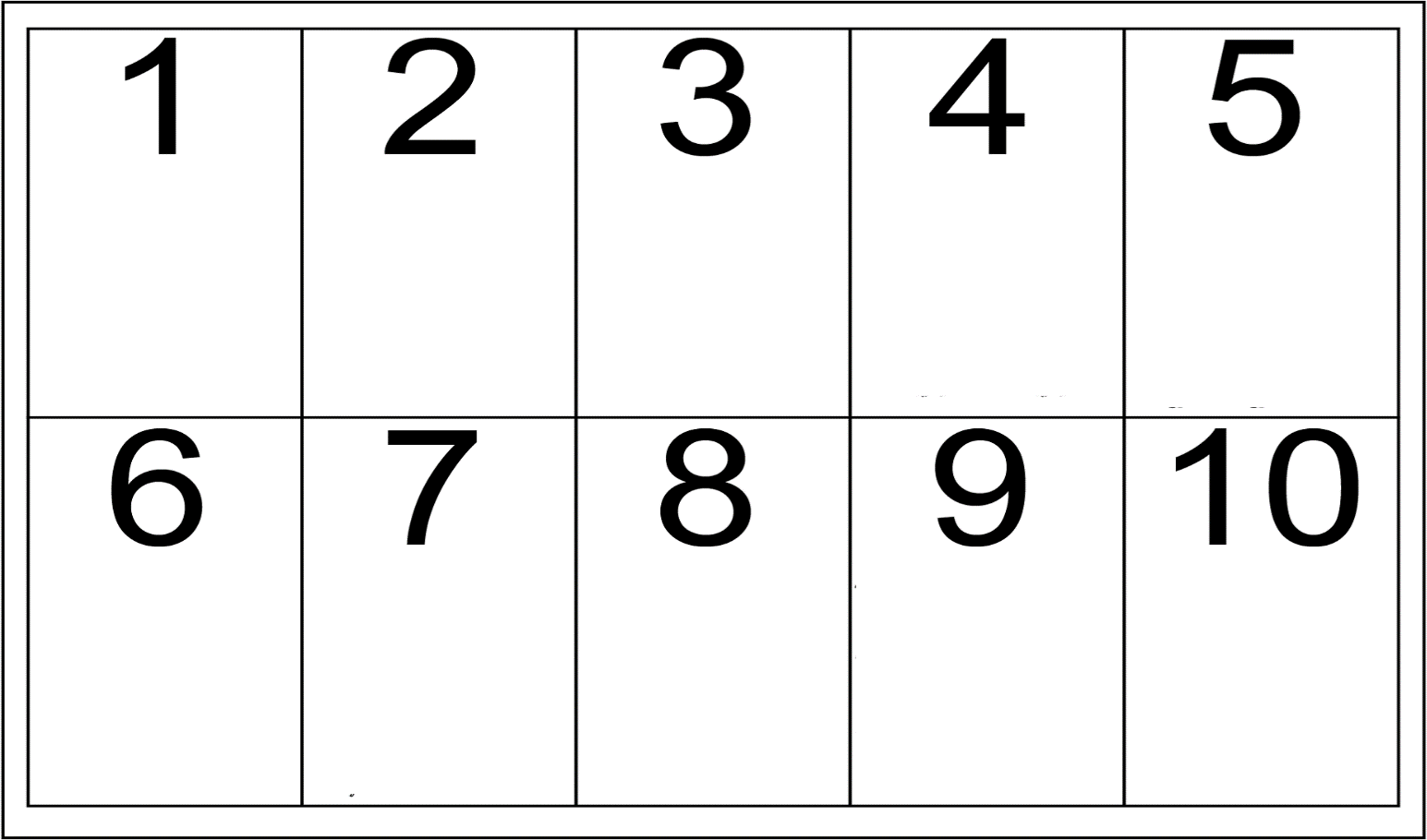
# Answer:

Most common Issues in VR

1. **Rendering Failures:** Environment elements like trees or water might not render properly.  
   Solution: Use optimized assets and ensure your VR system has sufficient processing power.
2. **Motion Sickness:** Users may feel nauseous due to lag or mismatched visual and motion cues.  
   Solution: Improve frame rates and reduce latency. Use comfort settings like teleportation for movement.
3. **Limited Field of View (FOV):** The FOV in VR headsets can feel restrictive.  
   Solution: Choose headsets with a wider FOV and adjust the lens settings for a better view.
4. **Poor Tracking:** Controllers or movements may not track accurately.  
   Solution: Ensure sensors are properly set up and calibrate tracking devices regularly.
5. You are asked to design the markers (targets) for an Augmented Reality (AR) application to teach

children how to count from 1 to 10. Each marker includes **a number** and **a corresponding number of objects** to engage children in learning to count. Please complete the design of markers by drawing different objects in the table below. **(10 points)**

# Answer:



**Problem 4**

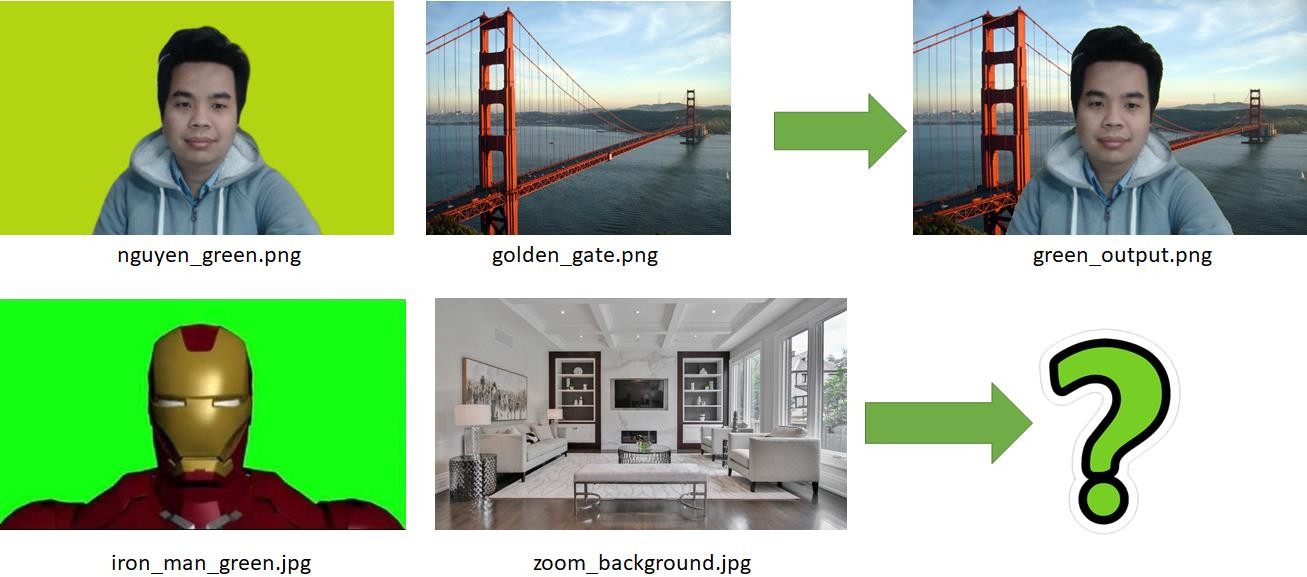
Dr. Nguyen is meeting students on Zoom. He uses a green screen as a background (**nguyen\_green.png**). He applied augmented reality to change the background to **golden\_gate.png**.

Iron Man (**iron\_man\_green.jpg**) also wants to do it. In particular, he wants to use the

**zoom\_background.jpg** but he does not know how. Please help Iron Man solve the problem.

* 1. (**zoom\_background.jpg**) to Iron Man’s green screen (**iron\_man\_green.jpg**). You may check the example below.
  2. Please write a MATLAB script to help Iron Man blend the Zoom background

# (25 points)



%% Your MATLAB script

%% You need to submit the .m file as well

% Read the input images

ironManImage = imread('iron\_man\_green.jpg'); % Image with Iron Man and green background

backgroundImage = imread('zoom\_background.jpg'); % New background image

% Convert the images to double precision for processing

ironManImage = double(ironManImage); % Convert Iron Man image to double

backgroundImage = double(backgroundImage); % Convert background image to double

% Convert to HSV color space for easier color segmentation

ironManHSV = rgb2hsv(ironManImage); % Convert Iron Man image to HSV color space

% Define the green color range for masking

hueMin = 0.2; % Minimum hue value for green

hueMax = 0.4; % Maximum hue value for green

saturationMin = 0.2; % Minimum saturation for the green color

valueMin = 0.2; % Minimum value for the green color

% Create the mask for green color

greenMask = (ironManHSV(:,:,1) >= hueMin & ironManHSV(:,:,1) <= hueMax) & ...

(ironManHSV(:,:,2) >= saturationMin) & ...

(ironManHSV(:,:,3) >= valueMin); % Mask where the green color is present

% Convert the mask to logical type

greenMask = logical(greenMask); % Convert mask to logical type (true/false)

% Resize the background image to match the size of Iron Man image

backgroundImage = imresize(backgroundImage, [size(ironManImage, 1), size(ironManImage, 2)]); % Resize background

% Create the output image

outputImage = zeros(size(ironManImage), 'double'); % Initialize output image

% Replace the green screen with the new background

outputImage = backgroundImage; % Start with the background image

outputImage(repmat(~greenMask, [1 1 3])) = ironManImage(repmat(~greenMask, [1 1 3])); % Replace non-green areas with Iron Man image

% Convert the output image to uint8 for display

outputImage = uint8(outputImage); % Convert to uint8 for proper display

% Display the result

figure;

imshow(outputImage); % Show the final image

title('Iron Man with New Background'); % Title for the displayed image

* 1. Display your results on **nguyen\_green.png** and **zoom\_background.jpg**. **(5 points)**

A screenshot of a computer

Description automatically generated

# Problem 5

Open Blender **v2.69**, and model a Pinata object as below. You can change the color or size. Be creative.

# View 1 View 2



**View 3 View 4**

1. Then, you save your model in a .blend file and submit the .blend file as well **(20 points)**.
2. Copy and paste the screenshot of your Pinata model into the box below **(5 points)**.

A cartoon of a toy

Description automatically generated A cartoon of a toy animal

Description automatically generated