Amirkabir University of Technology
(Tehran Polytechnic)

Assignment 1
Image Processing is Cool!

Homeworks Guidelines and Policies

- ***What you must hand in.*** It is expected that the students submit an assignment report (HW1_[student_id].pdf) as well as required source codes (.m or .py) into an archive file (HW1_[student_id].zip).
- ***Pay attention to problem types.*** Some problems are required to be solved *by hand* (shown by the icon), and some need to be implemented (shown by the icon).
  Please don't use implementation tools when it is asked to solve the problem by hand, otherwise you'll be penalized and lose some points.
- ***Don't bother typing!*** You are free to solve by-hand problems on a paper and include picture of them in your report. Here, cleanness and readability are of high importance. Images should also have appropriate quality.
- ***Reports are critical.*** Your work will be evaluated mostly by the quality of your report. Don't forget to explain what you have done, and provide enough discussions when it's needed.
- ***Appearance matters!*** In each homework, 5 points (out of a possible 100) belongs to compactness, expressiveness and neatness of your report and codes.
- ***Python is also allowable.*** By default, we assume you implement your codes in MATLAB. If you're using Python, you have to use equivalent functions when it is asked to use specific MATLAB functions.
- ***Be neat and tidy!*** Your codes must be separated for each question, and for each part. For example, you have to create a separate .m file for part b. of question 3. Please name it like p3b.m.
- ***Use bonus points to improve your score.*** Problems with bonus points are marked by the icon. These problems usually include uncovered related topics or those that are only mentioned briefly in the class.
- ***Moodle access is essential.*** Make sure you have access to Moodle because that's where all assignments as well as course announcements are posted on. Homework submissions are also done through Moodle.

- ***Assignment Deadline.*** Please submit your work **before the end of March 28th**.
- ***Delay policy.*** During the semester, students are given 7 free late days which they can use them in their own ways. Afterwards there will be a 25% penalty for every late day, and no more than three late days will be accepted.
- ***Collaboration policy.*** We encourage students to work together, share their findings and utilize all the resources available. However you are not allowed to share codes/answers or use works from the past semesters. Violators will receive a zero for that particular problem.
- ***Any questions?*** If there is any question, please don't hesitate to contact us through the following email addresses: **ali.the.special@gmail.com** and **dalirani@aut.ac.ir**. You may also find us in pattern recognition and image processing lab, 3rd floor, CEIT building.

1. Simple Commonplace Problems Regarding Image Acquisition (8 Pts.)

Keywords: *Image Acquisition, Image Storage, Basic Relationships of Pixels, Neighborhood, Adjacency, Connectivity, Paths*

Before getting things started, there are some basic topics need to be addressed. Understanding these topics may not be very crucial, but they are considered as the fundamentals of digital image processing and may somehow be of use.

The main focus of this problem is on **Image Acquisition**, which is actually the first stage of any vision system. Among various subjects related to image acquisition, two subjects of **Relationships of Pixels** and **Image Storage** are examined here.

First, consider the following 8×8 image. Find the shortest length of $4-$, $8-$ and $m-$path between two points p and q considering the following sets:

a. $V=\{2\}$
b. $V=\{1,2\}$
c. $V=\{1,2,3\}$

	4	**1**	**3**	**3**	**0**	**1**	**4**	**2**	
	3	**1**	**2**	**2**	**1**	**4**	**3**	**2**	*(q)*
	3	**0**	**4**	**3**	**0**	**1**	**2**	**1**	
	2	**2**	**2**	**4**	**1**	**2**	**4**	**4**	
	3	**1**	**3**	**1**	**3**	**1**	**2**	**4**	
	2	**3**	**3**	**3**	**0**	**4**	**1**	**2**	
(p)	**2**	**2**	**2**	**1**	**1**	**2**	**2**	**1**	
	0	**3**	**0**	**2**	**2**	**0**	**3**	**3**	

Now, given below are two image subsets S_1 and S_2. Considering $V=\{1\}$, determine whether these two subsets are

d. $4-$adjacent
e. $8-$adjacent
f. $m-$adjacent

	S_1				S_2				
1	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**0**
1	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**1**
1	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**1**
0	**0**	**1**	**1**	**1**	**1**	**0**	**1**	**1**	**1**

Next, Assume an $n-$bit gray image which is composed of n $1-$bit planes, where plane 1 contains the lowest-order bit of all pixels in the image and plane n contains all the highest-order bits. Consider a 1024×512, $256-$level grayscale image.

g. Determine the number of bit-planes for this image.
h. Which plane is the most significant one in terms of visual appearance?
i. Determine the number of bytes needed to store this image. Ignore image headers and compression.

Now consider a 7×7 mm CCD camera chip with 2048×2048 elements which is focused on a square, flat area placed at $1\,m$ away from its $35\,mm$ lens.

j. Determine the number of line pairs per millimetre the camera is able to resolve.

Next, assume an $8-$bit square image needs $4\,MB$ of storage space.

k. How many pixels are there in the image?

Finally, let the storage size of a 512×512 grayscale image be $256\,KB$.

l. What do you infer about this image?

2. Digging Into Color Spaces with Dr. Donald Trump! (10 Pts.)

Keywords: *Color Space, RGB Space*

Digital images, as you are probably very well aware, are matrices of numbers in mathematical terms. Therefore, most of the matrix operations are also applicable to images as well. This problem aims to get you more accustomed to this relationship. Meanwhile, you'll become more familiar with some **Color Spaces**, and also practice how to perform image operations in more efficient ways.

A set of color spaces alongside their conversion algorithm from RGB space are given in the below table. Load the RGB image "dr_trump.jpg" as the input image. For each one of the color spaces, answer the following parts.

a. Calculate the color space channel(s) pixel by pixel using loops.
b. Repeat the previous part using matrix operations.
c. Now use built-in functions to obtain the color space channel(s).
d. Display and compare the results you obtained in the previous parts.
e. Compare the execution times of each method.

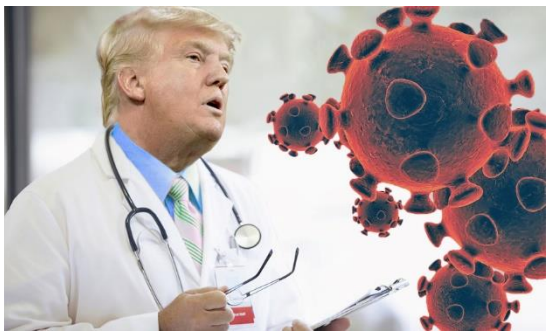


Figure 1 Dr. Trump is thinking about developing a vaccine for COVID-19 Coronavirus

$\mathbf{I} = 0.299\times R + 0.587\times G + 0.114\times B$	$R' = R/255$ $G' = G/255$ $B' = B/255$ $\mathbf{K} = 1 - \max(R', G', B')$ $\mathbf{C} = (1 - R' - \mathbf{K})/(1 - \mathbf{K})$ $\mathbf{M} = (1 - G' - \mathbf{K})/(1 - \mathbf{K})$ $\mathbf{Y} = (1 - B' - \mathbf{K})/(1 - \mathbf{K})$	$\mathbf{K} = 0.299\times R + 0.587\times G + 0.114\times B$ $\mathbf{Y} = \text{round}(0.859\times K) + 16$ $\mathbf{U} = \text{round}\left(0.496\times(B - K)\right) + 128$ $\mathbf{V} = \text{round}\left(0.627\times(R - K)\right) + 128$
Grayscale	**CMYK**	**YUV**

HSV

$$R' = R/255$$
$$G' = G/255$$
$$B' = B/255$$
$$C_{\max} = \max(R', G', B')$$
$$C_{\min} = \min(R', G', B')$$
$$\Delta = C_{\max} - C_{\min}$$
$$\mathbf{H} = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\dfrac{G' - B'}{\Delta} \bmod 6\right) & , C_{\max} = R' \\ 60^\circ \times \left(\dfrac{B' - R'}{\Delta} + 2\right) & , C_{\max} = G' \\ 60^\circ \times \left(\dfrac{R' - G'}{\Delta} + 4\right) & , C_{\max} = B' \end{cases}$$
$$\mathbf{S} = \begin{cases} 0 & , C_{\max} = 0 \\ \dfrac{\Delta}{C_{\max}} & , C_{\max} \neq 0 \end{cases}$$
$$\mathbf{V} = C_{\max}$$

HSL

$$R' = R/255$$
$$G' = G/255$$
$$B' = B/255$$
$$C_{\max} = \max(R', G', B')$$
$$C_{\min} = \min(R', G', B')$$
$$\Delta = C_{\max} - C_{\min}$$
$$\mathbf{H} = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\dfrac{G' - B'}{\Delta} \bmod 6\right) & , C_{\max} = R' \\ 60^\circ \times \left(\dfrac{B' - R'}{\Delta} + 2\right) & , C_{\max} = G' \\ 60^\circ \times \left(\dfrac{R' - G'}{\Delta} + 4\right) & , C_{\max} = B' \end{cases}$$
$$\mathbf{L} = (C_{\max} + C_{\min})/2$$
$$\mathbf{S} = \begin{cases} 0 & , \Delta = 0 \\ \dfrac{\Delta}{1 - \lvert 2L - 1\rvert} & , \Delta \neq 0 \end{cases}$$

HSL

3. Let's Play: Solving a Jigsaw Puzzle Using Basic Image Operations (16 Pts.)

Keywords: *Basic Operations on Images, Image Cropping, Image Resizing, Image Rotation, Image Translation*

It's hard to find someone who has not played a jigsaw puzzle game in his lifetime. Yet, it's also hard to find someone who has solved it using image processing techniques, the challenge you are about to face in this problem. In addition to some fun, this task helps you practice a series of image basic operations such as **Cropping**, **Resizing**, **Rotation**, **Translation** and so on.

Figure 2 shows an image containing 35 pieces of a 5×7 jigsaw puzzle depicting a miniature of Zahhak, the evil character in Shahnameh. Your job is to first extract the pieces and then put them together to create a complete image of the size 2500×3500. Although it may look arduous, it can be achieved by just performing simple image operations on the input.

Now, let's get our hands dirty. First, load the image "zahhak_miniature_puzzle.jpg".

a. Slice the image into 35 even blocks. To accomplish this, you need to divide the given 3000×4200 image into five columns and seven rows. Save all these 600×600 images and put a name on them considering the template `[raw_piece_01.jpg]` (for the first piece).
b. Extract the main pieces. Note that the background color is white, making it quite easier to extract the pieces. Save all the extracted 500×500 images with a name according to the template `[piece_01.jpg]`.
c. In order to ease up the process and reducing the execution time, resize the extracted pieces to 1/5 of their size. Save these 100×100 images and name them like `[small_piece_01.jpg]`.
d. Now comes the main part. For your convenience, the piece in the top left corner is placed in its right location. Start with this piece and find its neighbors one by one using a similarity metric of your choice. Note that some pieces need to be rotated. Therefore, it is advised to consider all 4 states (0°, 90°, 180° and 270°) when comparing pieces. The output of this step must be 5×7 matrix specifying the correct position of pieces based on their numbers.
e. Use the matrix you obtained in the previous part and put the pieces with their original size (500×500) together to complete the puzzle. Display and save the resultant image.

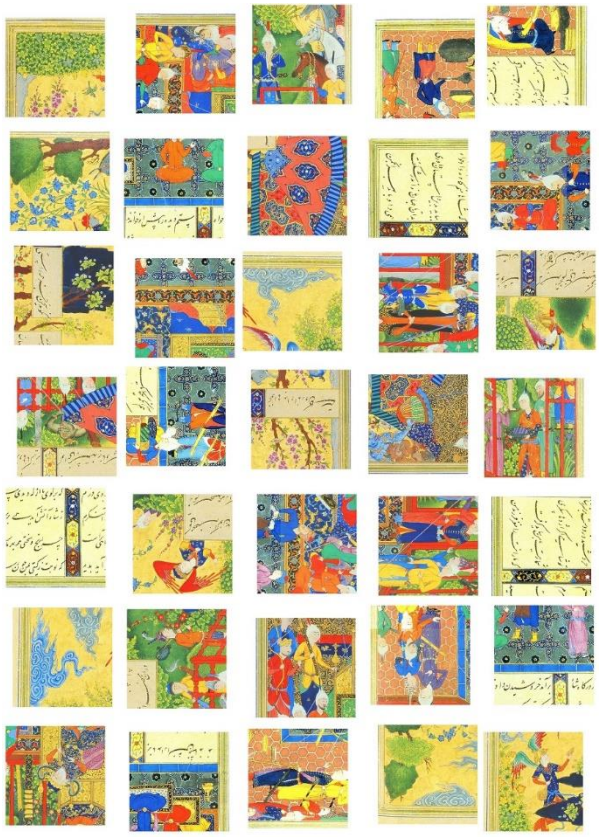


Figure 2 The input image contains pieces of a jigsaw puzzle, illustrating scenes from Shahnameh. The objective is to extract these pieces properly and combine them together to form a flawless image

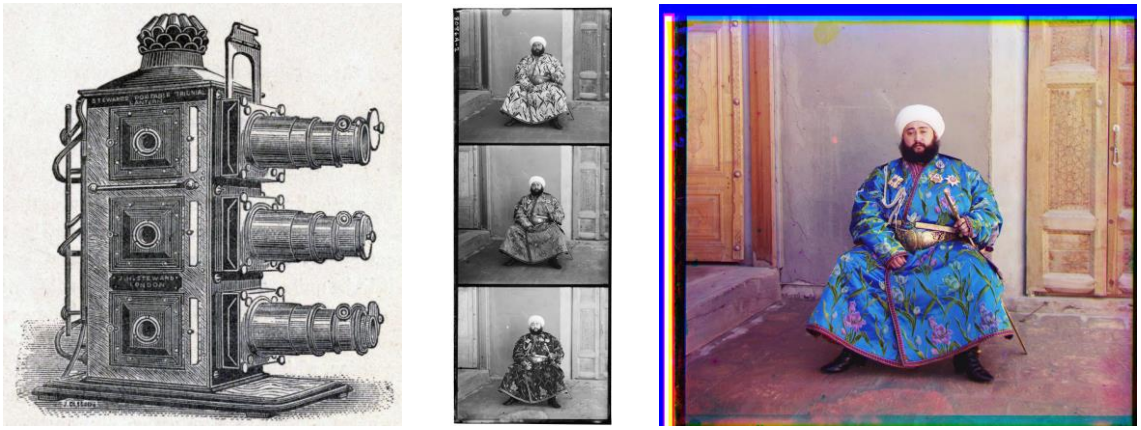
Hint: To avoid getting mixed up, save the intermediate pieces into separate folders.

4. Color Photography: Let's Make an Old Dream Come True (16 Pts.)

Keywords: *Color Photography, RGB Channels, Image Alignment, Resolution Pyramid*

Immediately after the first ever camera photograph **"***View from the Window at Le Gras"* (here) was recorded, researchers started to dream about **Color Photography**, i.e. adding real-world colors to the photos. One of the early pioneers in this area was Sergey Prokudin-Gorsky (1863-1944), a Russian chemist and photographer who travelled the Russian Empire from 1909 to 1915 and documented thousands of invaluable photos of that era.

Although color photography wasn't introduced at that time, Prokudin-Gorsky devised a method to capture colors for the photographers of the future. To do so, he captured three different grayscale pictures of every scene, each with a different color filter in front of the camera. These color filters were in red, green and blue, creating the hope that one day, a researcher of the future, would come along and produce beautiful color images by combining them together (Figure 3). He never saw his colored photos, but his legacy led to producing hundreds of high-quality color images of century-ago Russia (here).



(a) **(b)** **(c)**

Figure 3 A brief depiction of Prokudin-Gorsky's method (a) Three channel lantern projector similar to the one used by Prokudin-Gorsky (b) An example of vertical images taken by him in three channels, from Alim Khan, The Emir of Bukhara in 1911 (c) The result obtained by merging the channels together

Here, you are given some of Prokudin-Gorsky's grayscale image composites, Figure 4. These triple-framed images contain three grayscale photos taken in the early 1900s, and each frame represent the image captures with a blue, green and red filter. Choose three composites to your liking and use them as inputs in the following parts.

a. **Preparation.** Write a function `extract_channels()` which takes a triple-framed image as input, chops it vertically into three separate parts and save these channels with appropriate names.
b. **Merge.** Implement another function, `stack_channels()`, which takes three channels as input and stack each of the three images into a proper color channel and display a single colored image. It is expected that the stacked photos look wonky and unaligned. Include it in your report.
c. **Alignment.** Now write a third function, `align_results()`, which takes the image you obtained in the previous part and align it appropriately. This function must search over possible pixel offsets in the range of [-30, 30] to find the best alignment for each channel. One simple way to do so is to keep one channel fixed, and align the other two by searching over the offset range both horizontally and vertically. Pick the alignment that maximises a

similarity metric (of your choice) between the channels, e.g. dot products, normalized cross correlation, etc.

Note: For full credit, your report needs to include properly aligned images. Find a similarity metric that will accomplish this.

d. **Resolution Pyramid.** For large offsets and high resolution images, it can be computationally intensive to compare all the alignments for a specific range of displacements (e.g. [-30, 30]). It is often advised to start by estimating an alignment on a low-resolution version of the image before applying it on higher resolutions. In this part, you are asked to implement a two-level image pyramid by scaling the extracted channels down by a factor of 2, and then execute your alignment over the range of offsets [-15, 15]. After choosing the best alignment based on your similarity measure, use it as a starting place to again run the alignment in a small range [-15, 15] in the full resolution images.

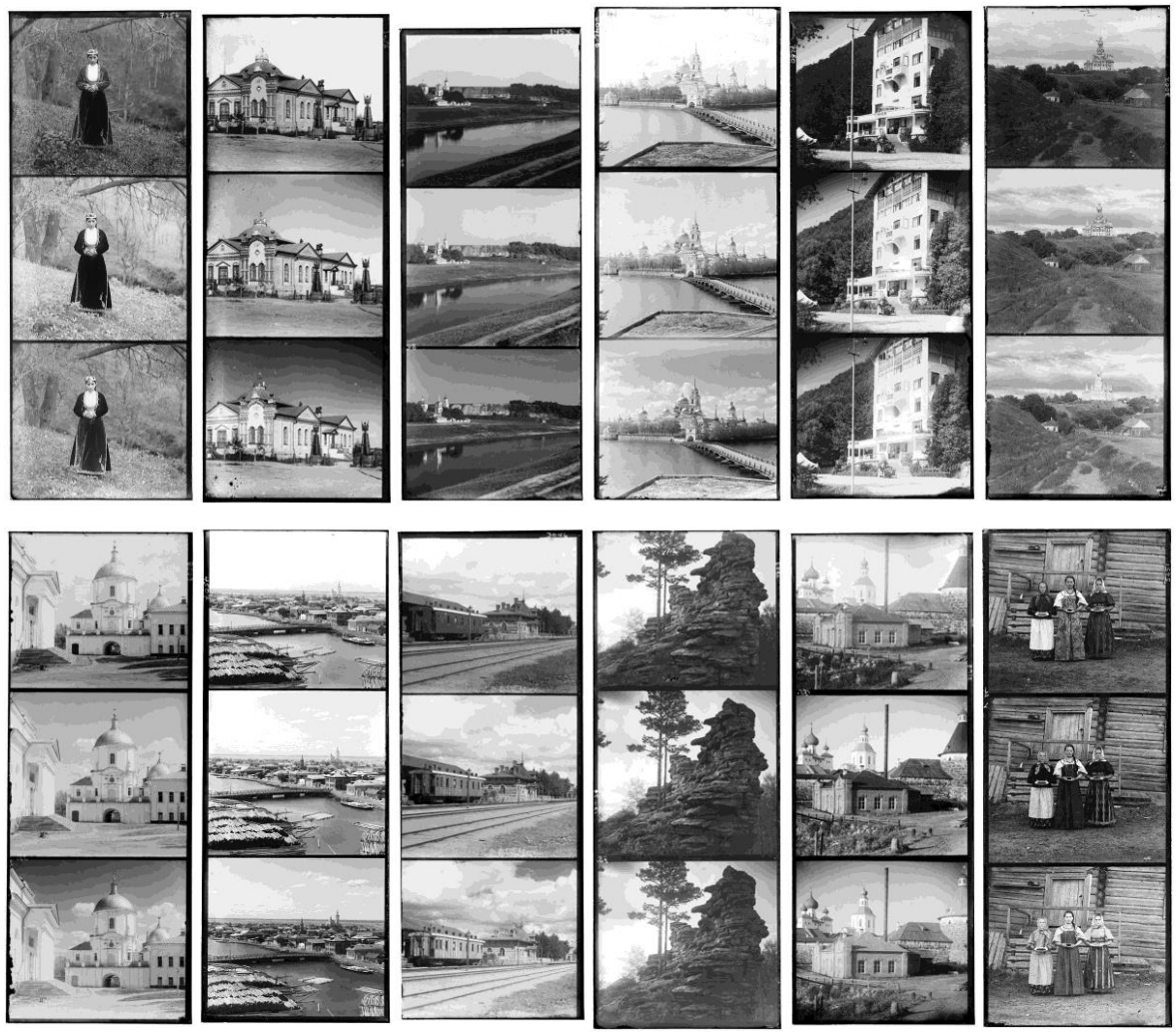


Figure 4 A dozen of Prokudin-Gorsky's negatives given here as the input. Although many of his images were lost, the majority of them are now kept in the U.S. Library of Congress

5. Become a Digital Librarian using Simple Pixel Operations! (10 Pts.)

Keywords: *Template Matching*

In this problem, we're going to take a further look at pixel operations. More specifically, here the focus is on searching and finding the location of an image in a larger image, a subject known as **Template Matching** in the area of image processing.

First, load the image "bookshelf_1.jpg". As can be seen, the image contains an imaginary bookshelf. Your task is to search for the book "book_1.jpg" over this bookshelf.

a. Find and display the position of the given book inside the bookshelf.
b. Report the number of other volumes of this book. Also display them on the image.
c. Display the matrix of errors as a heatmap.

Now let's deal with a more challenging one. Load the image "bookshelf_2.jpg" and the missing book "book_2.jpg". Although the goal is similar, the main image in this case is extremely large, making it very difficult and time-consuming to search for the book. Thus, some preprocessing steps are needed.

(a) **(b)**

Figure 5 The first task concerns finding the number and locations of the all the volumes of the given book (a) A dummy bookshelf (b) The missing book

d. Implement a function `downscale_img()`, which takes an input image and reduce its dimensions to 1/2 of the original image.
e. Downscale both images to 1/4 of their main sizes using the above function. Specify the strategy you used in case the dimensions were not divisible by 4. Also display the results.
f. Now use the resized images to search for the desired book in the bookshelf. Display the position in the original image.
g. Can you suggest other strategies to reduce the search time?

(a) **(b)**

Figure 6 Although the objective is still the same, the second task is about a real library with hundreds of real books (a) Bookshelf full of books (b) The wanted book, "The shock of the new"

6. When Image Processing Techniques Spoil the *Spot the Difference* Challenge! (15 Pts.)

Keywords: *Image Operations, Image Thresholding, Image Alignment, Image Translation, Image Rotation*

"*Spot the Difference*" is a type of puzzle you've probably came across numerous times. In this puzzle, two very similar images are given to the player and he/she is asked to find the slight differences between them. While sometimes very challenging and difficult, image processing techniques can handle these puzzles as easy as pie.

A set of four Spot the Difference puzzles is given to you. The puzzles are similar, all taken from *The Adventures of Tintin* comic books and contain 7 discrepancies. However, some of these puzzles might look more challenging than the others and need some extra works to be solved. In each part, reveal the differences by displaying a clear image showing the discrepancies among the images.

a. Puzzle 1
b. Puzzle 2
c. Puzzle 3
d. Puzzle 4

Figure 7 Four "Spot the Difference" puzzles, each with different characteristics (a) Images are grayscale, aligned and with the same size (b) Images are colored, unaligned (displacement) and with the same size (c) Images are colored, unaligned (displacement) with different sizes (d) Images are colored, unaligned (displacement and rotation) with different sizes

7. Where's Wally? Image Processing Answers (15 Pts.)

Keywords: *Image Thresholding, Template Matching*

"Where's Wally?" is a British puzzle book series which consist of a set of elaborate illustrations depicting dozens or more people doing a variety of funny things at a certain location. The challenge is to find a specific character called Wally (or Waldo in the U.S.) hidden somewhere in the image. Wally is recognised by his red-and-white-striped sweater, blue jeans, bobble hat and round glasses.

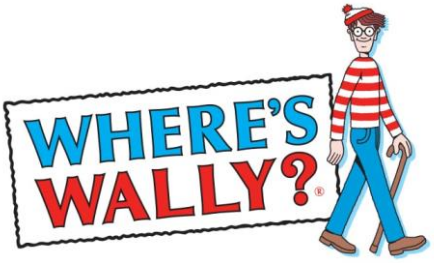


Figure 8 Wally's appearance is unchanged among all the puzzles, which is in fact the main clue to identify him between the crowds

The aim of this problem is to get you familiar with the subject of **Image Thresholding** by solving a couple of "Where's Wally?" puzzles. Bearing in mind the fact that Wally always wears a shirt with the same pattern, one can easily restrict the search area to those regions who are more similar to his shirt. In other words, the idea is to keep only red and white pixels in the images and perform the search in those areas.

You are provided with four different "Where's Wally?" puzzles. In each puzzle, your goal is to find the whereabouts of Wally and display his location in the original image. You are also given a few sample images of Wally (Figure 9) which you may find useful.

a. Find two approximate RGB values for the red and white parts of Wally's shirt.
b. Try to preserve those pixels of the puzzles similar to Wally's shirt by setting appropriate thresholds using the values you found in the previous part. You may also need to keep a specific range of pixels around them, e.g. a 50×50 pixels square. Set the values of other pixels to zero (black), and display the results you obtained for each puzzle.

Figure 9 Although his costume is always the same and might look similar to other characters, Wally may be carrying a stack of books that vary from scene to scene (a) Wally (b) His lookalikes

c. Search for Wally in the modified puzzles. You only need to search in non-black pixels, i.e. those who were detected similar to Wally's shirt.
d. Display each puzzle with Wally's position indicated by a bounding box (rectangular borders around his coordinates).

Note: There are also several Wally lookalikes in each puzzle. Try not to be fooled by them.

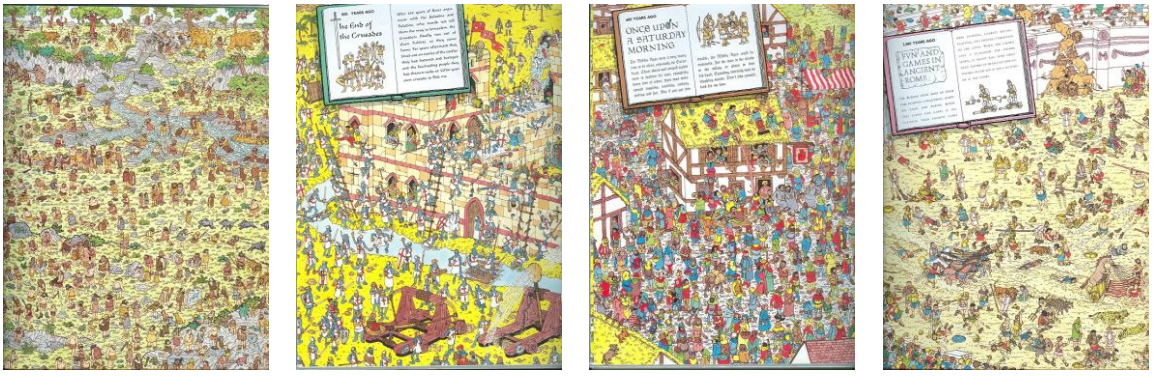


Figure 10 Different puzzles of "Where's Wally" with Wally hiding somewhere in the scene

8. Further Study: Quantum Image Representation (20 Pts.)

Keywords: *Image Representation, Image Operations, Quantum Computing, Qubit, Quantum Gates*

So far you've probably got frustrated many times by tedious and time-consuming computations of image processing. These time complexity issues may become even more annoying in the high-level applications like object recognition and pedestrian detection. During the past years, various approaches have been introduced to overcome these problems. Recently, **Quantum Image Processing (QIP)** methods has attracted attention, claiming to be hugely effective in speeding up image processing operations.

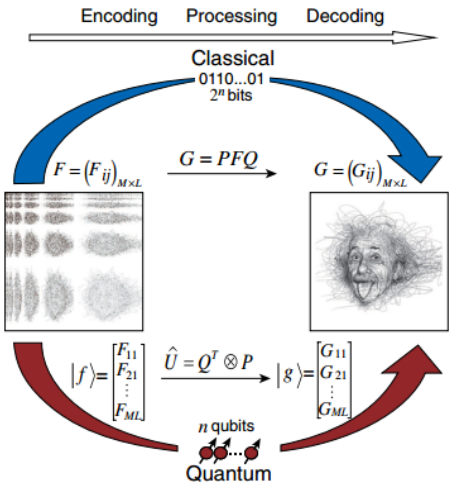


Figure 11 Classical image processing versus quantum image processing. F and G are the input and output images, respectively. In the classical image processing, an M x L image is represented as a matrix and can be encoded with at least 2n bits. Image transformation is also conducted using matrix operations. On the other hand, the same image can be represented as a quantum state and encoded in only n Qubits. In this case, the quantum image transformation can be performed by unitary evolution under a suitable Hamiltonian.

In this part, you are asked to inquire about **Quantum Image Representation**, mainly by reading a 2018 IEEE Access paper "*A quantum image representation based on bitplanes*". This paper proposes a novel image representation technique which can represent a grayscale or RGB image of the size $2^n \times 2^n$ using only $2n+4$ (grayscale image) or $2n+6$ (color image) **Qubits**. It also introduce some quantum operations such as complement of colors or bitplane translation which has been shown to be more effective than classical image processing operations.

The paper PDF file is attached to this assignment. Please read it carefully and try to make sense of it. You will come across some quantum computing terms that you may not be familiar with, therefore further reading is inevitable.

Answer the following questions based on your grasp of the paper.

a. Write down a comprehensive summary of your understanding of parts II and III of the paper.
b. How do the authors generalise their proposed representation method for color images? Explain briefly.
c. Explain one of the quantum operations introduced in the paper, and compare it with the corresponding classical approach.
d. Relation (9) shows how the least significant bit (LSB) of a grayscale image bitplane is stored. Write down the corresponding relation for the most significant bit (MSB).
e. Write a brief review of the paper based on your understanding of the proposed method.

Note 1: you are free to utilize relations and images from the paper, but a mere translation wouldn't be of much worth.

Note 2: Your answers may not be totally accurate, but your efforts are worthwhile.

9. Some Explanatory Questions (5 Pts.)

Please answer the following questions as clear as possible:

a. Which one of the color spaces in problem 2 are more appropriate to compute the distance between any two colors? What distance metrics would you use? Provide justification.
b. Explain the procedure of digitizing a continuous image in detail. Provide example if necessary.
c. What can you comment on the images captured by a 10-megapixel camera and a 20-megapixel one? Compare in terms of quality, resolution and storage.
d. Explain the concepts of re-sampling and sub-sampling in the context of image processing.
e. Is RGB to grayscale conversion possible? Provide explanations.

Good Luck!
Ali Abbasi, Farhad Dalirani