Amirkabir University of Technology (Tehran Polytechnic)

# Assignment 4
## Pixels Gone Wild: Restoring, Compressing, and Morphing

## Homeworks Guidelines and Policies

- ***What you must hand in.*** It is expected that the students submit an assignment report (HW4_[student_id].pdf) as well as required source codes (.m or .py) into an archive file (HW4_[student_id].zip). Please combine all your reports just into a single .pdf file.
- ***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 do not use implementation tools when it is asked to solve the problem by hand, otherwise you will be penalized and lose some points.
- ***Don't bother typing!*** You are free to solve by-hand problems on a paper and include their pictures 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. Do not forget to explain your answers clearly, and provide enough discussions when needed.
- ***Appearance matters!*** In each homework, 5 points (out of a possible 100) belong to compactness, expressiveness, and neatness of your report and codes.
- ***MATLAB is also allowable.*** By default, we assume you implement your codes in Python. If you are using MATLAB, you have to use the equivalent functions when it is asked to use specific Python 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, which must be named '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 is where all assignments as well as course announcements are posted. Homework submissions are also made through Moodle.

---

- ***Assignment Deadline.*** Please submit your work **before the end of July 7th**.
- ***Delay policy.*** During the semester, students are given only 7 free late days which they can use them in their own ways. Afterwards, there will be a 20% penalty for every late day, and no more than four 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 do not hesitate to contact us through the following email addresses: **ebp.mohsen@gmail.com** and **ali.the.special@gmail.com**.

## 1. It Don't Matter If You're Trump or Biden! (20 Pts.)

**Keywords**: *Image Morphing, Facial Morphing, Facial Features, Cross Dissolve Method, Delaunay Triangulation, Affine Transformation*

In November 1991, Michael Jackson unveiled the music video "Black or White," inspired by his hit single from the album Dangerous. Like his previous works, "Black or White" rapidly gained immense popularity across the globe, amassing a staggering audience of 500 million views and setting a new world record for the most-watched music video. Alongside its comedic narrative, the video captivated viewers with its groundbreaking visual techniques, introducing unprecedented innovations that were hailed as a cinematic masterpiece during that era.

*Figure 1 In the last minute of the music video, several people from different races dance as their face change into another, while repeatedly singing "black or white".*

Towards the conclusion of the music video, a diverse group of individuals representing various ethnicities and nationalities (Figure 1) engage in a captivating dance sequence, seamlessly transforming from one person to another. This powerful visual effect symbolizes the core message of the song, centered around the idea of racial equality encapsulated by the main verse, *"It don't matter if you're black or white."* This innovative technique, now recognized as **Image Morphing**, had not been previously employed elsewhere. Today, it is commonly utilized as an entertaining means to morph one face or object into another (Figure 2).

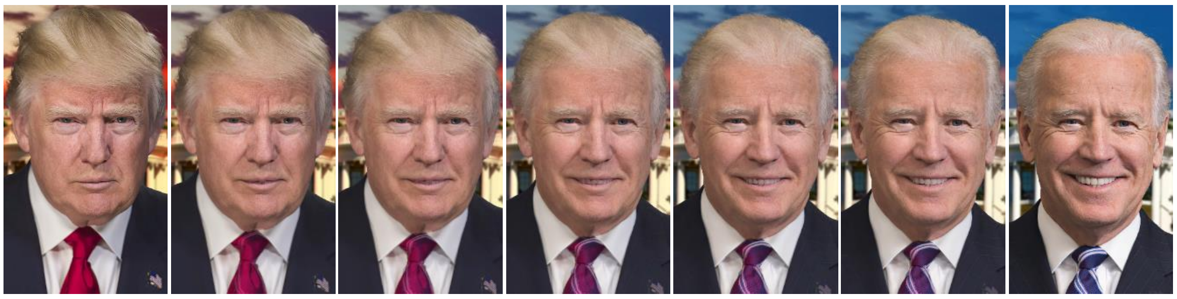


*Figure 2 From Trump to Biden. Face morphing is often used in media to convert a celebrity face into another*

Image Morphing can be performed through various approaches. Here, we first introduce a naïve way to handle the problem, and then define an algorithm based on **Geometric Transformation**, which successfully overcomes the first method's deficiency.

### I. Naïve Method

The first idea which may come to the mind is to simply interpolate whole images. Assuming $M$ as the initial image and $N$ as the target image, each pixel in the morphed image $I$ can be interpolated as follows:

$$I(i,j) = (1-\alpha)\cdot M(i,j) + \alpha \cdot N(i,j), \quad 0 \le \alpha \le 1$$

where $\alpha$ controls the gradual transformation from $M$ to $N$. This technique is known as *cross-dissolve* in the film industry, and is able to balance coloration in the middle frames. However, it leaves a "ghosting" effects in the intermediate images, especially if the two faces aren't perfectly aligned.

## II. Triangulation Method

In the previous method, only the "photometric" side of the transformation was addressed, and the "geometric" side was ignored. We therefore need to find a one-to-one correspondence between certain parts of the two images (e.g. lips, eyes, nose, etc.), so as to know which part of the initial image should be morphed into which part of the target image. This can be done through a new algorithm, which is defined in the following steps:

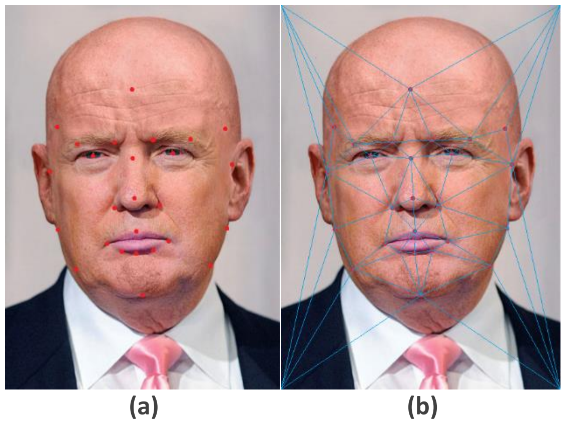


*Figure 3 The process of face triangulation (a) Specific landmarks are detected, denoting important regions of the face (b) Triangulation algorithm is performed*

**i. Finding corresponding points.** First, a set of corresponding keypoints must be extracted from each images. These keypoints play a major role in highlighting prominent facial features so as the interpolation becomes more specific. Here, we perform this step manually, using the reference image shown in Figure 3. In this image, 26 facial feature points are highlighted with red dots. In order to include the background into the morphing frames, at least 4 points (corners) must also be selected. Let $P_M = \{p_M^1, p_M^2, \dots, p_M^k\}$ and $P_N = \{p_N^1, p_N^2, \dots, p_N^k\}$ be the two set of points extracted from images $M$ and $N$, respectively.

**ii. Delaunay Triangulation.** Next, a weighted mean of the points in the two sets must be calculated. More precisely, using $P_M$ and $P_N$, we obtain another set of points $P_I = \{p_I^1, p_I^2, \dots, p_I^k\}$ in which

$$p_I^j = (1-\alpha)\, p_M^j + \alpha\, p_N^j$$

These points are the corresponding features in the intermediate image, and will be used later to apply the triangulation to the set $P_I$.

Triangulation operation divides the plane of the image into small triangles (Figure 3), using the points of the set $P_I$ as vertices. We opt to use *Delaunay Triangulation*, since it does not produce overly skinny triangles. The triangulation should be performed for all the sets $P_M$, $P_N$ and $P_I$, giving a one-to-one correspondence between triangles in the images $M$, $N$ and $I$. We now have three sets of corresponding triangles, $T_M = \{t_M^1, t_M^2, \dots, t_M^l\}$, $T_N = \{t_N^1, t_N^2, \dots, t_N^l\}$ and $T_I = \{t_I^1, t_I^2, \dots, t_I^l\}$.

**iii. Affine transformation.** Now, we select a triangle $t_M^j$ from the image $M$ as well as its corresponding triangle $t_I^j$ from the image $I$, and calculate the affine transform which converts the triangle $t_M^j$ to $t_I^j$. Similarly, we calculate the transformation matrix of the corresponding triangle $t_N^j$ in the image $N$ to the triangle $t_I^j$. Then we apply the obtained transformations to all the triangles of both images $M$ and $N$, and obtain warped images $M'$ and $N'$. Finally, we calculate the intermediate image $I$ by using the same equation in the naïve method:

$$I(i,j) = (1-\alpha)\cdot M'(i,j) + \alpha \cdot N'(i,j), \quad 0 \le \alpha \le 1$$

Hence, a series of intermediate images based on different values of $\alpha$ will be obtained. When shown in rapid succession, these images gives the effect of one image being converted into the other.

Two images 'rouhani.png' and 'raisi.png' are given. Consider the second as the target image.

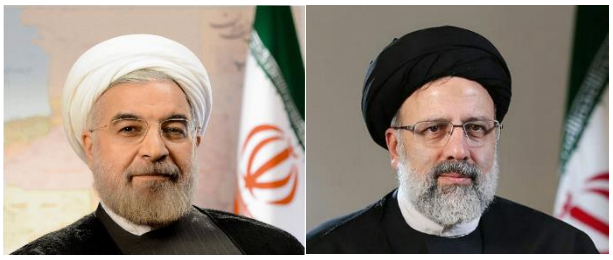


*Figure 4 Input images given for this task (a) Initial image (b) Target image*

a. Use the naïve method to convert the initial image to the target one. Calculate and display 10 intermediate images. Comment on the results.
b. Manually select keypoints in two images. Display the detected points.
c. Apply triangulation to both images, and display the results.
d. Calculate the transformations, and display 10 intermediate images. Compare the results with part (a).
e. Generate a two-second animation containing 61 frames. Frame 0 must be identical to the initial image, and frame 60 must be identical to the target image. In the video, each frame will be displayed for 1/30 of a second.

**Note 1:** The transition between frames should be smooth and the intermediate frames must be as realistic as possible.

**Note 2:** There's no restrictions on the usage of built-in functions and libraries.

**Recommended MATLAB functions**: `cpselect()`, `ginput()`, `delaunay()`, `imtransform()`, `cp2tform()`, `maketform()`, `tsearch()`, `interp2()`, `VideoWriter()`, `writeVideo()`

## 2. Exploring Google Earth Post-Processing Techniques (22 Pts.)

**Keywords**: *Image Stitching, Panorama, Feature Extraction, Feature Matching, Geometric Transformation, Image Blending, Scale-Invariant Feature Transform (SIFT), Random Sample Consensus (RANSAC)*

Google Earth is a remarkable tool that grants us the ability to traverse the globe virtually and delve into a satellite perspective of our own locality. It offers the opportunity to witness three-dimensional representations of renowned landmarks and maneuver through bustling streets of major cities, all within the confines of our homes. The captivating imagery and data utilized by Google Earth originate from raw snapshots acquired by satellites and aircraft, gathered through collaborative efforts with esteemed organizations such as NASA and National Geographic. However, how do these initial raw images undergo a transformative process to culminate in the visually appealing and seamless maps we admire?

Google acquires an immense volume of images captured under diverse conditions. Once these images are enhanced through techniques like noise reduction, haze removal, and contrast and brightness adjustments, the challenging endeavor lies in connecting these images seamlessly. To achieve this, Google employs a straightforward yet highly efficient algorithm called **Image Stitching**, which is akin to the technique your smartphone utilizes for capturing panoramic photos.

The process of image stitching can be divided into four steps:

**i. Feature Detection**. First, a set of interest points are detected in the images. Here, we consider the *Scale-Invariant Feature Transform (SIFT)* detector.

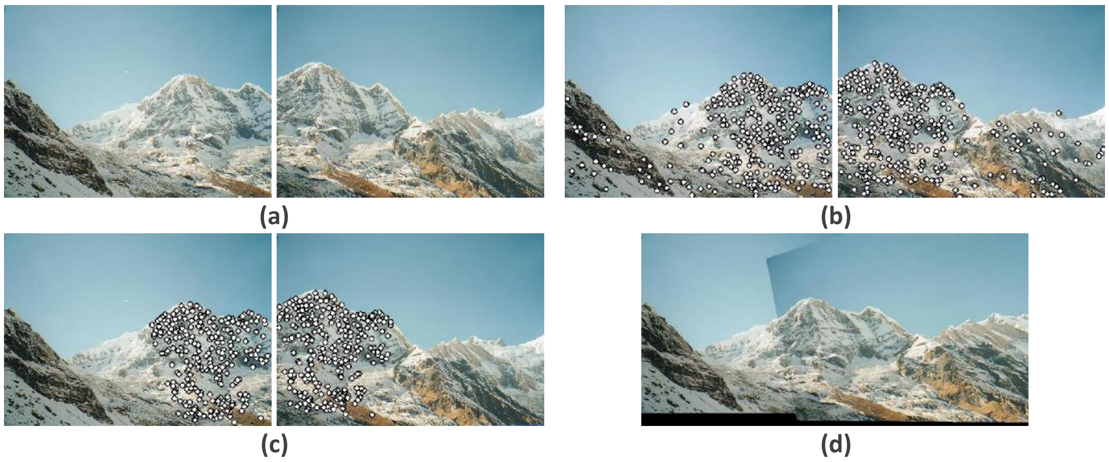


Figure 5 The process of image stitching (a) Initial images (b) Keypoints are detected in both images (c) A set of matched keypoints between two images are obtained, from which the geometric transformation between the two images is estimated (d) After applying the geometric transformation, images are blended and create the final panorama result

**ii. Feature Matching**. Next, similar keypoints between the pair of images are detected. To perform this, a distance measure (e.g. Euclidean distance) between feature vectors must be considered. Then a threshold must be used to determine whether a pair of keypoints from two images shall be considered as a match.

**iii. Image Registration**. The goal of this step is to use the corresponding points in the reference image and the target image to estimate a geometric relationship between the two images. For this purpose, we consider *RANdom SAmple Consensus (RANSAC)* algorithm which attempts to find a transformation which is less affected by mismatched points.

**iv. Image Blending**. Finally, the transformed image(s) must be combined by averaging the pixel values in the overlapped regions from two or more images. Another way to implement this step is to apply weighted averaging of overlapping pixel values where the weights vary linearly according to the distance of a pixel in the overlapped region to the centre of either one of the images.

In this problem, we aim to investigate the usage of this technique in three different features of Google Earth.

## I. Satellite View

From its initial release in 2005, Google Earth has provided satellite views of earth varying in quality and resolution. Imagery of a certain area may be a combination of various satellite images, each taken from different angles, scales and positions. Hence, they are required to be stitched to make one integrated image of the area with no visible border, Figure 6.

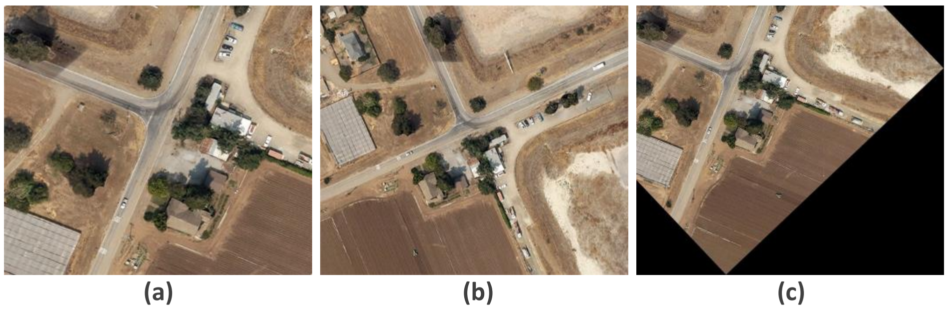


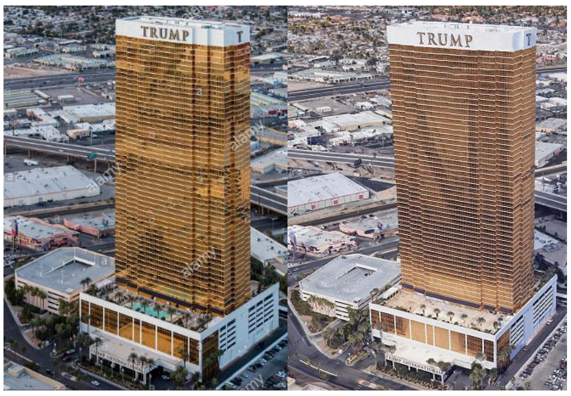
Figure 6 An example of stitching two satellite images (a) First image (b) Another image of the same scene, with minor change in camera position (c) Resultant image

a. Three satellite images of our university are provided. Although they share similar scales, the camera has been slightly rotated when capturing these images. Use the procedure described above to obtain an integrated satellite image of our university campus.

## II. 3D Aerial View

In 2012, Google started to deploy small, camera-equipped airplanes to fly above various cities and take pictures of their buildings. The goal was to utilize these images and create 3D maps of those areas. Today, Google Earth 3D imagery has been expanded to over 60 countries, including every U.S. state. By allowing the users to view photorealistic 3D imagery of buildings (and even objects like cars, trees and billboards), this feature has brought a whole new experience to Google Earth users, Figure 7.

The first step in the process of generating 3D reconstruction of buildings is to stitch various images of similar scenes, which are indeed taken from different angles and scales.



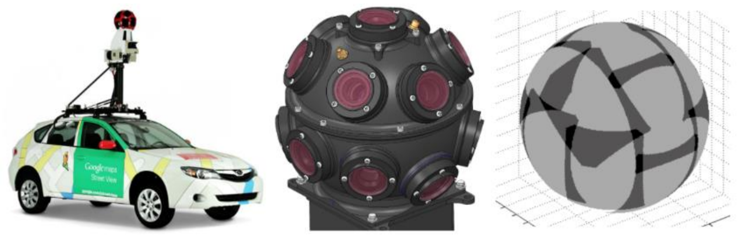
**(a)** **(b)**

*Figure 7 Comparison between the aerial photo and photorealistic 3D imagery of Trump International Hotel in Las Vegas (a) Aerial photo taken from an airplane (b) 3D reconstruction of the building*

b. Use the input images which are taken from the Camp Nou stadium in Barcelona, and find the stitched image containing all parts.

## III. Street View

Another cool feature which was added to Google Earth in its 2008 release is the Street View. This feature enables user to view 360° panoramic streel-level photos of major cities and their surroundings. These photos are taken by cameras installed on automobiles, and can be viewed at different scales and from many angles.

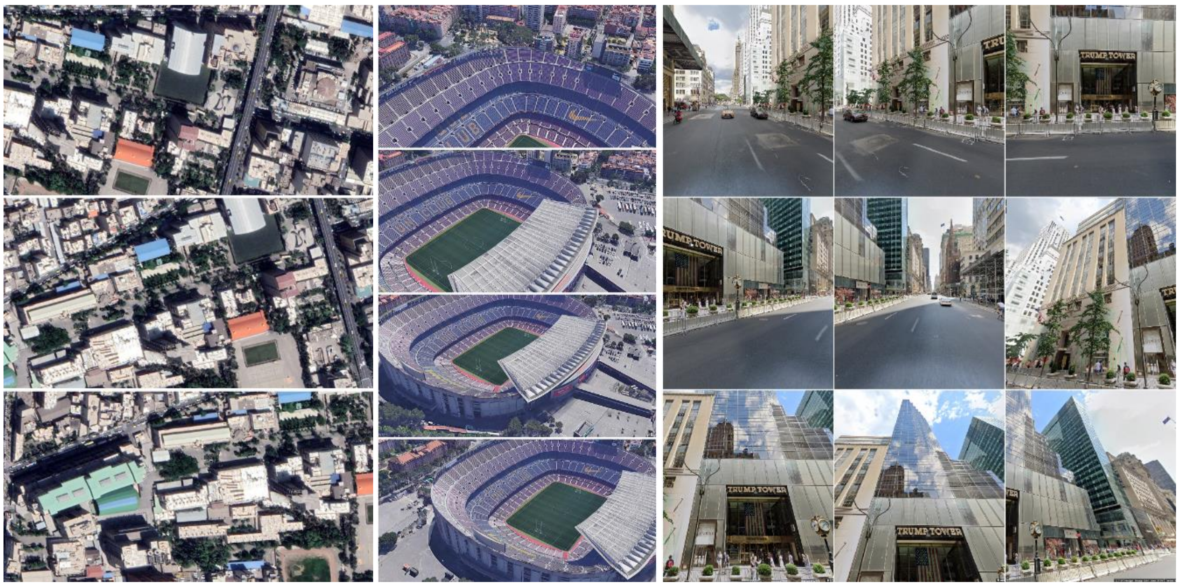


**(a)** **(b)** **(c)**

*Figure 8 Each Google car is equipped with cameras which simultaneously take images in multiple directions (a) A street view car (b) A close-up view of the rosette, which is made up of 15 cameras (c) A visualization of the spatial coverage of each camera, with overlapping regions shown in darker gray*

While driving, Google cars capture multiple images in different directions at specific time intervals. More specifically, each car is equipped with a multi-camera rosette which is made up of 15 cameras (Figure 8). The images taken at each shot are later overlapped and stitched together into a single 360-degree image.

c. Several images taken in the vicinity of Trump Tower in New York City are given. Create a single panoramic image of the area.



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

*Figure 9 All input images given for each of the three sections (a) Satellite images of Amirkabir University in Tehran (b) Aerial photos taken from Camp Nou in Barcelona (c) Several shots taken from the entrance of Trump Tower in NYC*

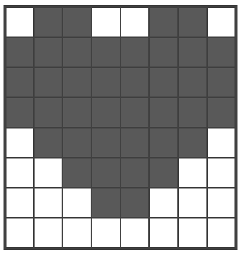
**Note:** Except for the final step (blending), you are allowed to make use of built-in functions and libraries. However, you may need to tune some parameters manually to obtain the best possible output.

## 3. A Review on the Basics of Image Compression (15 Pts.)

**Keywords**: *Image Compression, Huffman Coding, Zig-Zag Ordering, Run-Length Coding, Entropy Coding, Root-Mean Square Error, Peak Signal-to-Noise Ratio*

The goal of this problem is to evaluate your fundamental knowledge of **Image Compression** techniques.

First, consider the following binary image.



a. Find the runlength representation. Note that you must assume that a line always starts with "white", and specify the white runlength and then the black runlength alternatively. Put an "EOL" symbol at the end of each line.
b. Find the corresponding zig-zag ordered vector to this image.
c. We want to use a single Huffman codebook to code the white and black runlengths. Write down the probability distribution of the symbols to be coded.
d. Next, generate the Huffman code for all possible runlength symbols. You must consider "EOL" symbols as well.
e. Calculate the average number of bits per coded symbol. How does it compare with the entropy of the symbol?
f. What is the average bit rate (bits/pixel) for this method? What about the compression ratio compared to when using 1 bit for each pixel to indicate whether the pixel is black or white?

Next, assume an 8-bit gray-scale image given as below.

| | | | |
|---|---|---|---|
| 0 | 18 | 48 | 64 |
| 12 | 100 | 120 | 180 |
| 32 | 150 | 180 | 200 |
| 96 | 180 | 220 | 255 |

g. Instead of 8 bits, we would like to use 2 bits to represent each pixel. Find the value assigned to each pixel. If we intend to keep the value of each pixel between 0-255, explain how you will do such mapping.

h. Find the root-mean square error and mean-square signal-to-noise ratio for the output image you obtained in the previous part.

Now consider the following table as the initial dictionary for LZW algorithm.

| Index | Entry |
|---|---|
| 1 | a |
| 2 | r |
| 3 | - |
| 4 | t |

i. Output of the LFW encoder on a sequence is given below. Decode the sequence.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 1 | 3 | 4 | 8 | 6 | 3 | 2 | 10 | 5 | 13 | 8 | 12 | 11 | 3 |

j. Encode the decoded sequence using the same initial dictionary. Does your answer match the sequence given above? Justify your answer.

Finally, consider a JPEG compression procedure in which instead of 8×8 image blocks, blocks of the size 4×4 are used. The coefficients generated by applying DCT on a 5×5 block of luminance value as well as the quantisation table are given below.

| | | | |
|---|---|---|---|
| 3) 120 | 4) 30 | 5) 20 | 6) 8 |
| 7) 25 | 8) 20 | 9) 12 | 10) 5 |
| 11) 12 | 12) 10 | 13) 5 | 14) 3 |
| 15) 10 | 16) 7 | 17) 4 | 18) 1 |

| | | | |
|---|---|---|---|
| 2 | 15 | 20 | 40 |
| 18 | 14 | 25 | 40 |
| 20 | 24 | 32 | 50 |
| 24 | 36 | 48 | 60 |

k. Find the values assigned to each pixel after performing quantisation.

l. Assume the quantised table is transmitted as a one dimensional vector. In what order will it be transmitted, and why? Write down the ordered vector.

m. Find the run length encoding of the vector in the previous part.

n. What kind of additional compression techniques can be used when several such spatially neighboring 4×4 blocks are transmitted? Justify your answer.

## 4. Warming Up: Image Morphology for Binary Images (12 Pts.)

**Keywords**: *Image Morphology, Image Logical Operations, Structuring Element, Image Dilation, Image Erosion, Image Opening, Image Closing*

**Image Morphological Operations** can be more clearly explained in binary images. The effect of each morphological operation on an image and the role of structuring element and its properties like position of the origin and size are more perceivable when these operations are applied to 1-bit images. For this reason, let's start with some binary images first. Note that you are not allowed to use built-in functions to perform morphological operations, i.e. you have to implement them yourself.

a. Consider the image in Figure 10, part (a). Use an appropriate procedure consisting of proper morphological operation(s) to fill the holes inside the figure. The final output must be as clear as possible with the fewest artefacts detectable.

b. Morphological operations can also be used to extract relevant features from an image. Try to use these operations in order to remove certain features (like diagonal lines) from the image given in part (b) of Figure 10.

c. Assume the binary image in Figure 10, part (c). Use thickening alongside other morphological operations to process this image. Reduce all lines to a single pixel width and obtain their maximum length.

d. Figure 10, part (d) displays a letter which is said to be wrote by Amirkabir to Naser al-Din Shah Qajar. As can be seen, the letters are separated and sentences can hardly be read. Try to fill the gaps and obtain a clear readable result.

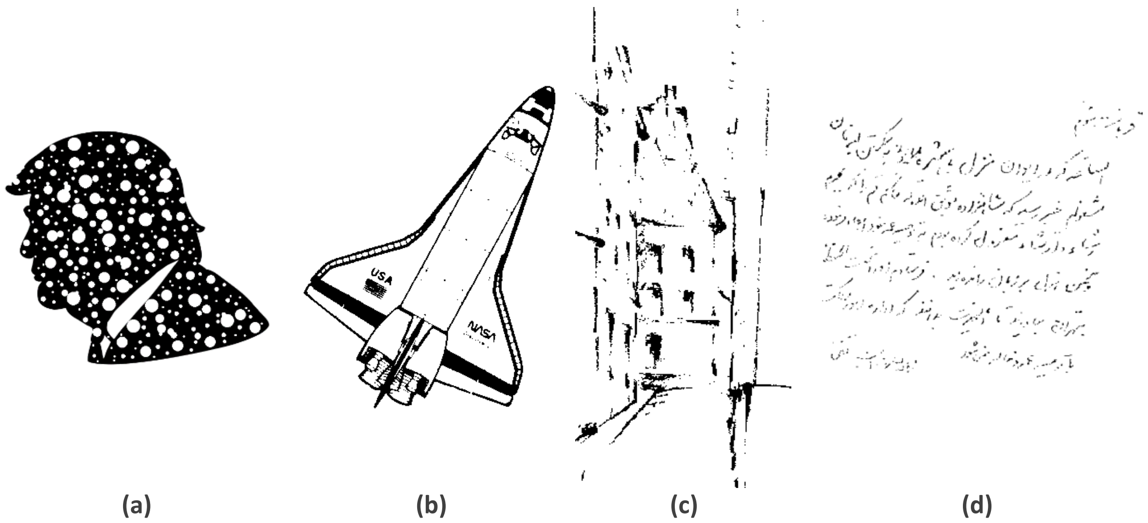


*Figure 10 Input images provided for each part (a) Binary image of Donal Trump with white holes of different sizes (b) A schematic of a NASA shuttle (c) A painting consisting of some houses in an alley (d) Handwriting attributed to Amirkabir*

## 5. Beyond Binary Images: Broadening Image Morphology Applications (16 Pts.)

**Keywords**: *Image Morphology, Structuring Element*

As you are very well aware, the scope of **Image Morphological Operations** is not limited to binary images. One can apply these operations on grayscale images and obtain fine results based on the intended application.

Here you are going to practice some of the related problems. Note that you are not necessarily supposed to obtain perfect results, as some of the tasks are a bit difficult to deal with. You must try your best nonetheless.

a. Given below are two microscopic images of human body cells which can be used in various medical applications. Your task is to implement an algorithm to count the number of cells inside each image.

b. Now comes the third task, in which the goal is to create a street map from a satellite image taken from the city of Berlin at night. You must use appropriate morphological

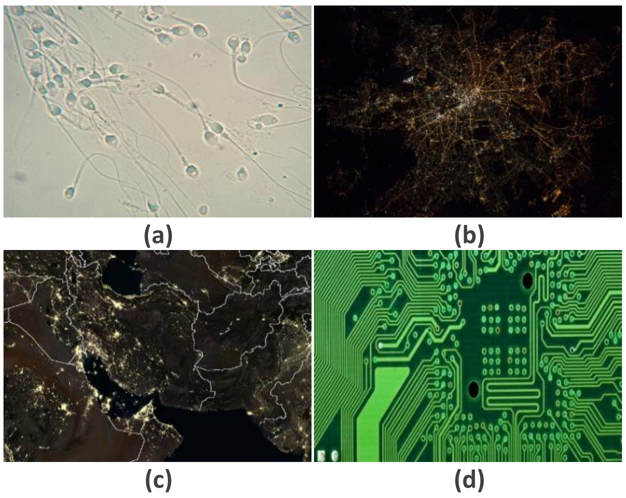


*Figure 11 Input images (a) Microscopic image of human body cells (b) Satellite image of Berlin at night (c) Satellite image of Iran (d) An unknown printed circuit board*

operations to detect the main roads and remove the remaining areas of the image. Highlight the streets properly.

c. This part is somehow similar to the previous one, as it utilizes another satellite image taken at night. This time the goal is to find and display the most crowded cities in this map in the Figure 11, part (c). Display the map with the whereabouts of the 10 most crowded (or brightest) cities clearly highlighted with colored points.

d. Finally, you are an image of a printed circuit board. Use morphological operations to remove the lines within the board, while keeping the circles. Then try to remove the circles while keeping the lines. Note that the output of the colored input must also be colored.

## 6. Some Explanatory Questions (10 Pts.)

Please answer the following questions as clear as possible:

a. Can image warping be reversed? Explain.

b. What are the definitions of "spatial scalability" and "amplitude scalability," and how does the JPEG 2000 image coding method incorporate these concepts?

c. How does assuming 0 instead of 1 as the origin of a structuring element impact the results?

d. Can a method be proposed for performing morphological thinning and thickening on grayscale images? If so, how can it be accomplished? If not, what are the reasons behind this limitation?

e. What makes dilation an inappropriate approach for filling small noisy holes in objects? Justify your answer.

*Good Luck!*
*Mohsen Ebadpour, Ali Abbasi*