1 2 9 0

UNIVERSIDADE D
COIMBRA

Beatriz Abrantes Abreu Madeira

Web-based Mosaic Creation Kiosk

Dissertation in the context of the Master in Informatics Engineering, specialization in Software Engineering, advised by Prof. Jorge Carlos dos Santos Cardoso and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2023

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ɖ
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Beatriz Abrantes Abreu Madeira

Web-based Mosaic Creation Kiosk

Dissertation in the context of the Master in Informatics Engineering, specialization in Software Engineering, advised by Prof. Jorge Carlos dos Santos Cardoso and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September 2023

Acknowledgements

I would like to express my sincere gratitude to my thesis advisor, Jorge Carlos dos Santos Cardoso, who guided me throughout this project. I would also like to thank my family and friends as well, for the support given to me during this project.

Abstract

The work developed in this thesis consisted of implementing a kiosk that allows the user to transform a photograph of his face into a Roman mosaic. The purpose of the kiosk is to be in operation in a museum to provide a playful experience for visitors. Before implementing the kiosk, algorithms that transform images into mosaics were researched. Kiosk systems used in museums were also researched in terms of functionality and interaction with the museum visitors. Research showed that Battiato's algorithm was the best choice to be used in the kiosk.

The chosen algorithm was implemented and adjusted to the kiosk's specific needs, that is, to represent some detail in a human face in the mosaics, using face detection. After implementation, an interface was developed to allow the users to use the algorithm as part of a kiosk application.

Following the implementation of the algorithm, its performance was evaluated in terms of execution time and visual quality of the mosaics. The visual assessment was made by distributing a form to identify optimal algorithm parameters. Following this, the kiosk application was then evaluated by end users, by operating it with a laptop and touch screen at the Department of Informatics Engineering at the University of Coimbra, involving students and teachers. Using their feedback, some improvements were made to the kiosk interface. The kiosk was then evaluated at the Monographic Museum of Conímbriga so that it was tested in a real environment by the museum visitors.

The results demonstrated that the system worked correctly but there are some improvements that can be made in the future such as improvement of the UI, and adding a physical structure to the kiosk in a way that it can be adapted for all users. The performance of the mosaic generation algorithm could also be improved in order to make the process faster.

Keywords

Roman Mosaic, Tessel, Kiosk, Mosaic generation algorithm

Resumo

O trabalho desenvolvido nesta dissertação consistiu na implementação de um quiosque que permite ao utilizador transformar uma fotografia do seu rosto num mosaico romano. O objetivo do quiosque é funcionar num museu para proporcionar uma experiência lúdica aos visitantes. Antes de implementar o quiosque, foram pesquisados algoritmos que transformam imagens em mosaicos. Os sistemas de quiosques utilizados nos museus também foram pesquisados em termos de funcionalidade e interação com os visitantes do museu. A pesquisa mostrou que o algoritmo de Battiato foi a melhor escolha para ser utilizado no quiosque.

O algoritmo escolhido foi implementado e ajustado às necessidades específicas do quiosque, ou seja, representar algum detalhe de um rosto humano nos mosaicos, utilizando deteção de rostos. Após a implementação, foi desenvolvida uma interface para permitir aos usuários utilizar o algoritmo como parte de uma aplicação de quiosque.

Após a implementação do algoritmo, seu desempenho foi avaliado em termos de tempo de execução e qualidade visual dos mosaicos. A avaliação visual foi feita através da distribuição de um formulário para identificação dos parâmetros ideais do algoritmo. De seguida, a aplicação quiosque foi então avaliada pelos utilizadores finais, operando-a com um computador portátil e ecrã táctil no Departamento de Engenharia Informática da Universidade de Coimbra, envolvendo alunos e professores. Usando o seu feedback, algumas melhorias foram feitas na interface do quiosque. O quiosque foi então avaliado no Museu Monográfico de Conímbriga para que fosse testado em ambiente real pelos visitantes do museu.

Os resultados demonstraram que o sistema funcionou corretamente, mas existem algumas melhorias que podem ser feitas no futuro, como melhoria da UI e adição de uma estrutura física ao quiosque de forma que possa ser adaptado para todos os utilizadores. O desempenho do algoritmo de geração de mosaico também poderia ser melhorado para tornar o processo mais rápido.

Palavras-Chave

Mosaico Romano, Tessela, Quiosque, Algoritmo de geração de mosaico

Contents

List of Figures

Chapter 1

Introduction

In this chapter, an introduction to the project is provided, including an explanation of the motivations behind it, the scope of the project, the objectives, and the structure of the document.

Mosaics are an old form of art that consists of arranging small pieces of stone, known as tesserae in order to form a certain pattern or drawing. Mosaics were made by Roman artisans who chose the color, shape, and texture of each tile to form different patterns and images.

The drawings represented in the mosaics often held symbolic meanings such as relaying stories, cultural beliefs, and historical events. Mosaics represented how life was in ancient times and historical battles, therefore, they became a very important tool to study the past. Roman Mosaics are also studied from an artistic perspective, that is, analyzing the artistic choices made by the artists of ancient Rome and exploring the evolution of artistic expression over time.

This form of art was used to decorate floors, entryways, and hallways. In the Roman world, mosaics could be found in public bathhouses, marketplaces, and homes. With the decline of the Roman Empire, this art form fell into disuse, but there are still artisans who practice it.

The presence of Roman heritage in Portugal can be observed and discovered in various regions and locations throughout the entire country. An example is 'Conimbriga' in Condeixa-a-nova, which is one of the most well-preserved and extensive Roman settlements in Portugal. Within the ruins found, there is a museum that contains historical artifacts found during the excavations that give an idea about the lifestyle of the people who lived there. Visitors can explore the ancient streets, houses, and public buildings, as well as view mosaics and other artifacts that have been uncovered during excavations [1].

In addition to Conimbriga, other notable Roman heritage locations in Portugal include "Villa Romana do Rabaçal" in Penela where we can find the remains of a large villa with a variety of architectural features such as floors with mosaics, thermal baths, and other public and private areas. These ruins also have a museum associated with them where the visitors can explore the Roman artifacts discovered in that location as well as decorative mosaics on the floors of the houses

[2].

MosaicoLab is an initiative that focuses on creating software to help explore roman mosaics and to motivate people to learn more about this subject. One example is the online editor that allows users to select a grid with different tile shapes and paint their mosaic with a pallet with pre-selected colors. It is destined to be used in workshops about mosaic creation.

1.1 Motivation

As referred above, the use of software can enhance the process of learning about ancient mosaics, using original ideas to interact with people, and new ways of exploring the mosaics.

Therefore, there is an opportunity to create a fun interactive kiosk to encourage the study of this part of history and provide interest in extending knowledge about the Roman heritage to people of all generations, attracting more visitors to these museums and providing a better museum experience.

The kiosk can therefore offer an enjoyable moment for groups of visitors or families during their museum visit, where the interaction transforms into a cherished memory of a positive and engaging experience.

1.2 Scope and Objectives

This project consists of implementing a kiosk application that would use one of the existing algorithms for mosaic creation to create a personalized mosaic, using a picture taken in real time by the user. The user should be allowed to access the mosaic on his phone so that he can save it as a souvenir or reminder of his visit to the museum.

The objective is to get the mosaics to look like Roman mosaics as much as possible, therefore the research of algorithms is focused on the ones who produce ancient mosaics and crystallization mosaics because these mosaic categories generate mosaics by joining colored pieces together each is the same principle used by ancient roman artisans to create mosaics. Therefore, other mosaic categories such as puzzle image mosaics and photo-mosaics are not explored. Puzzle image mosaics generate mosaics by joining smaller images that together form the mosaic.

Another goal is to integrate the mosaic algorithm into a kiosk application, adapting the algorithm in a way that the mosaic can be fully generated automatically with the lowest intervention needed from the user of the kiosk. The adaptation of the algorithm also involves generating the best possible result to represent human faces in mosaic.

The kiosk system is a complement to the experience of the visitor that can be used at the museum by the museum visitors without any supervision, that is, the goal is for the kiosk to be used autonomously. The most important design concern is for the kiosk to be used without too many instructions needed, that is, that can be used and easily understood by people from all age groups.

The functionalities developed should be possible to be easily integrated into the editor that is already developed by the MosaicoLab initiative.Therefore, the system was implemented using the same language as the editor, that is, it should be a web application.

1.3 Results

After extensive algorithm research, the algorithm implemented was Battiato's algorithm with some adaptations such as the incorporation of the face detection of the user so that the human features appear more clearly in the mosaic generated. The kiosk application was then developed, letting the user choose one out of three different mosaic options of mosaic, The options were generated by using different parameters in the algorithm.

Algorithm performance and user experience were evaluated in different test steps. For the algorithm's performance, some tests were made regarding the execution time and its changes with different parameters. The algorithm's visual results were also put to the test by the creation of a form where people could answer about the aesthetics of the mosaics produced. These results were then compiled and a conclusion was drawn about the previous results and the best three options were selected for the kiosk application.

When it comes to the user experience of the kiosk application, initially, the tests were carried out in an academic environment in the Department of Informatics Engineering (DEI), where participants were invited to interact with the kiosk and provide feedback through a form. Based on responses and feedback, modifications were made to the user interface to improve the clarity of the options when choosing mosaics as well as the visibility of the kiosk itself.

Subsequent testing of the kiosk application took place in the Conimbriga Monographic Museum, where visitors were able to experience the kiosk in its most appropriate and realistic setting. Opinions about the kiosk were generally positive, although there were discussions about mosaic generation time likely due to the selected tiling option and internet connection issues which affected the user experience.

1.4 Structure of the Document

The rest of the document is structured in the following way: in chapter 2 there is an overview of the relevant existing algorithms to create mosaics using an image as input. There is also an overview of kiosk software usage and overall particularities, in chapter 3 the methodology to be used is described as well as the work plan that was followed, in chapter 4 the developed work is presented, and in chapter 5 an evaluation of the algorithm and the kiosk application is made. In chapter 6 a conclusion of the document is presented, discussing the work that was developed and the future work.

Chapter 2

State of the art

Mosaics are a form of representation or construction of images using reduced-size polygonal shapes. There are several types of mosaics, each one with its specific characteristics that are explored in the next sections, as well as the existing algorithms to generate them.

Interactive kiosk applications are also addressed, discussing design considerations and concerns, the security of this type of software, the hardware and software used and the deployment and management involved. It will also be explored real use cases of kiosk applications, in particular, in a museum context.

2.1 Mosaic Generation Techniques

In this section, some existing algorithms for transforming images into mosaics are presented, as well as a discussion of these algorithms. They can be divided into categories such as Crystallization and Ancient regarding their characteristics. There are several categories of mosaics but only these are covered since they are the ones that are related to the Roman mosaic the most [3].

Regarding the Crystallization Mosaic effect generation the Haeberli, Dobashi and Faustino, and Figueiredo techniques were studied. Regarding Ancient Mosaic algorithms the Hausner Method, the Lars-Peter Fritzsche Method, the Battiato Method, and also the RenderBots Method are presented.

2.1.1 Crystallization Mosaic

The Crystallization Mosaic is a type of mosaic that is similar to stained glass (coloured glass windows). Stained glass windows were made in ancient times by cutting pieces of glass and colouring it. This form of art started to be developed by ancient Romans and Egyptians [4].Due to the manual process involved in this art, the stained glass presents a crystalline appearance and shiny shimmer effect. Therefore, the goal of the techniques presented in this section is to try to reproduce these characteristics.

Haeberli

This technique was developed by [5]. It is very simple and although it does not present the best results visually, it was the foundation for other techniques that were developed afterwards since it introduced new concepts, such as the Voronoi diagram usage, to the mosaic generation that leads to further investigations.

A Voronoi diagram is a way of dividing a plane into parcels named Voronoi regions. The partition of the plane is made in the following way: a set of dots of the plane, named sites or seeds, are chosen randomly. Then, a region for each site is calculated. The points of the plain are part of a certain region if "each location from the area surrounding a given point is closer to it than to any other point" [6], that is, the closest points to a site constitute the region of that site. The lines of the Voronoi diagram, or edges, are constituted by the points that are at the exact same distance from the two closest sites. Figure 2.1 is an example of a Voronoi diagram. For more information about Voronoi diagrams refer to [6].

To tessellate the image, a Voronoi diagram is used with a random number of sites in random positions of the image. This diagram contains information about the proximity of the points in the image, with the points represented in this diagram being equidistant to two or more sites. The colours used to fill the tesserae are taken directly from the original image. The final results obtained by this method can be seen in Figure 2.2.

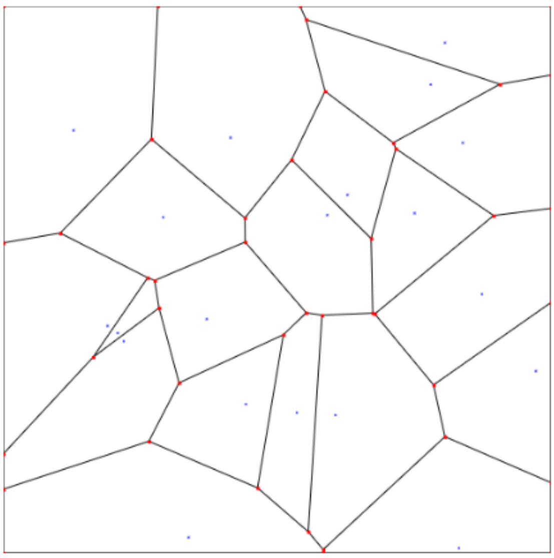


Figure 2.1: Voronoi Diagram with the sites coloured blue, the edges represented with the colour black and the vertices represented with the color red

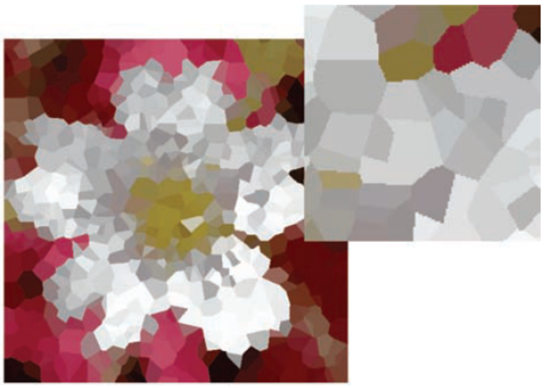


Figure 2.2: Results obtained by Haerbeli's method [5]

Dobashi

This technique, [7] performs tessellation using the Voronoi diagram, like Haeberli, but combines it with the definition of edges, which allows for highlighting objects in the image and obtaining a more aesthetically pleasing visual aspect, as shown in Figure 2.3.

The first step of the algorithm is to create the Voronoi diagram. As it is said in [7], the sites of the Voronoi diagram are positioned in the center of the Voronoi regions of the diagram. The colour of each Voronoi region is set by sampling it from the original image, that is, the position of the site of the Voronoi region corresponds to a certain pixel of the original image and the colour of that pixel is used to paint the Voronoi region and a mosaiced version of the image is obtained.

The resulting image is very inaccurate so to get a better visual result, the error of the colour is calculated, that is, the summatory of the squared difference between the colours of the image and the colours of the mosaiced version is calculated. The idea is to move the Voronoi sites and regions in a way that the difference between the colour in the Voronoi regions and the color in the original image is as minimum as possible.

The movement of the sites is done in two different stages: first, all the sites are moved if that movement decreases the value of the error of the colour, in order to capture the general features of the image.The second stage is to move each site individually in the direction that decreases the value of the error of the colour the most.

After the automated process described, the algorithm allows the user to add final touches to the mosaic interactively, such as varying the color in each region, highlighting the desired edges of the image and removing unwanted tiles.

Figure 2.3: Results obtained by Dobashi's method [7]

Faustino and Figueiredo

Another alternative for creating crystallization mosaics is to allow different sizes of tesserae to emphasize the edges of the objects of the image. This approach creates a different visual effect since the edges of the objects present in the image are not marked so abruptly visually, that is, there is no line separating one object from another, but a gradual decrease in the size of the tesserae on the edges of the image. This visual effect can be seen in Figure 2.4.

As it is said in [8], the first step of the algorithm is to sample the image. Using a quadtree, seeds are obtained and then used to compute a centroidal Voronoi diagram. From the image sampling, two sets of points are obtained: points clustered in the image edges and points in regions with less detail.

The next step of the algorithm is to compute the centroidal Voronoi diagram which is a variation of the Voronoi diagram where the sites of the diagram correspond to the centroids of each Voronoi region. [9] The diagram is computed using the points previously generated and a density function that reflects image features. The density function is used to compute the centroids of the Voronoi cells. The mosaic will have larger cells when the density is low and smaller cells when the density is high.

Finally, to paint the Voronoi regions, the authors give two options: paint the cell with the pixel's colour which is closest to the cell site or colour the Voronoi region with the average colour of the image in that corresponding place. The second alternative achieves better results, according to the authors because it takes into account color changes of the image pixels corresponding to that region which results in smoother color transitions in areas of the image where there are color changes.

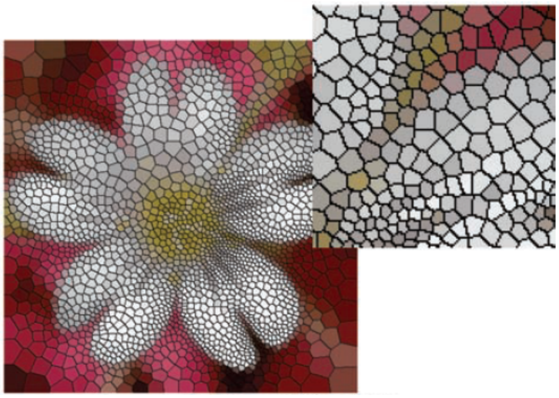


Figure 2.4: Results obtained by Faustino and Figueiredo [8]

Mould's method

David Mould [10] developed an algorithm to transform images into mosaics with a stained glass window effect that focuses on the real techniques used by the artists, having them as a reference for each step. For that, firstly the image is segmented using the EDISON (Edge Detection and Image Segmentation) system [11] that provides a GUI for the user to perform segmentation and edge detection over an image. After this, [10] describes a region smoothing technique using erosion and dilation operators. The erosion operator, as the name implies, causes the effect of an eraser, that is, it erases imperfections from the lines that delimit the region, while the dilatation operator works as a brush in the limits of the region.

From the segmentation of the image, some undesired shapes might occur such as 'islands'. Islands are regions that are fully surrounded by other regions and they are not used in stained glass windows due to the difficulty of their execution in real life, so to get around this obstacle, the region outside the island is divided in two. Another type of unwanted shape is two regions connected by a narrower bottleneck, as they are not, in general, aesthetically pleasing and would be fragile points, considering the construction in a real glass, therefore these regions are detected and then divided.

Another important aspect of the stained glass windows is the size of each region. Regions shouldn't be either too small or too large. If they are too small they will not be as highlighted as they are supposed to be, that is, they will get lost in the final mosaic. On the other hand, very large regions were not used in these types of mosaics due to the difficulty of manufacturing large pieces of glass in medieval times. If a region is too large, erosion is performed followed by connected component counting. Using this information, the algorithm checks if the region has a good subdivision. If not, the region is subdivided.

To colour the tiles, the authors opted for the heraldic pallet because its colours resemble the medieval ones.The heraldic pallet is constituted by two metals and five colours: "The metals are or, gold, or yellow; and argent, silver, or white. The colors are gules, red; azure, blue; vert, green; purpure, purple; and sable, black." [10]. The average colour of the region is calculated and then the heraldic colour that is less distant from the calculated one is chosen to paint that region.

Figure 2.5: David Mould's stained glass algorithm result [10]

An example of the final stained glass obtained can be seen in Figure 2.5.

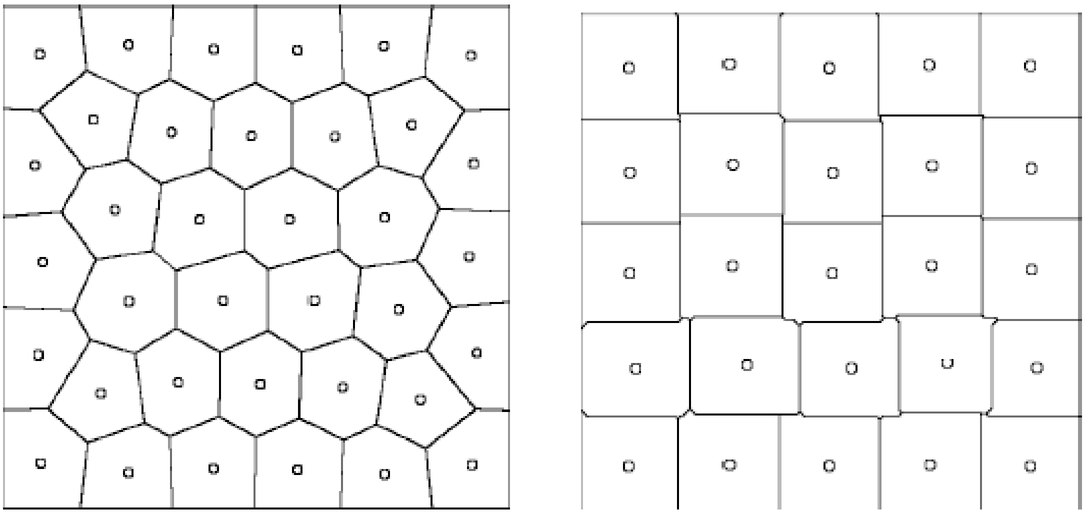
2.1.2 Ancient Mosaic

Ancient Mosaics were used for decoration purposes. They were made by hand, placing colored tiles together to form images. When trying to reproduce this type of mosaic digitally, the main problems to solve are the tile positioning and orientation. These issues are approached in different ways: by using Centroidal Voronoi Diagrams and adapting them to divide a plane into cells to perform the mosaicing of an image; or by trying to represent real mosaic creation techniques digitally, that is, positioning tiles one after the other and orienting them by tile cutting [3].

Hausner's method

Hausner developed an approach that adapts centroidal Voronoi diagrams to position the tiles in the best way possible. Centroidal Voronoi Diagrams, as explained in Faustino and Figueiredo's algorithm, are a variation of the Voronoi diagram where the sites of the diagram "are the centroids (centers of mass) of the corresponding Voronoi regions" [9].

As it was said in previous sections, the Voronoi cells are created according to the distances of the points of the plane to the sites, that is, the diagram obtained depends on the measurement of the distance between the points. Therefore, the resulting Voronoi diagram is highly influenced by how the distance between points is calculated. In the case of the Euclidean distance, the Voronoi regions tend to be shaped like hexagons Figure 2.6. On the other hand, the Manhattan corresponds to "the distance between two points measured along axes at right angles" [12] and, as a result, it does not allow the distance to be measured diagonally. When used to compute the Voronoi diagram, the plane tends to be divided into squares Figure 2.6.



(a) *Using Euclidean Distance* (b) *Using Manhattan Distance*

Figure 2.6: Centroidal Voronoi Diagrams [13]

As described in [13], the first step of the algorithm is to choose random points of the image. For each point, a square pyramid is placed with the upper vertex at the point. The square pyramid is used because its mathematical equation embodies the Manhattan distance equation. Next, each pyramid is oriented according to the direction field and then projected into an XY plane. This step generates the Voronoi diagram. The centroid of each Voronoi region is then calculated and the points are translated to their corresponding centroid. The process is iterated until it reaches a steady state, at which point the algorithm has converged.

The last step of the algorithm is to place a tile in each point, oriented by the direction field, calculated using the gradient of the Euclidean distance. The direction field must follow edge orientations by making the tiles move away from the edges so that tile edges follow image edges. The edges are specified by the user. For the color, the authors give two alternatives. The first one is to sample a pixel of the center of the tile and paint the whole tile with that color. The other alternative is to paint the tile with the average color of the pixels of the image covered by that tile.

The algorithm can be adapted to allow different sizes of tiles if you want to accentuate details of the original image in the final mosaic. For this, an alpha variable is added to the equation used in the algorithm previously. The comparison between the results with and without the tile size change can be seen in Figure 2.7.

Lars-Peter Fritzsche's Method

The purpose of this algorithm is to allow the creation of mosaics with heavy user intervention in order to produce mosaics with higher quality. Fully automated algorithms can produce "unwanted artifacts such as miss-align tiles or highlighting of unwanted tiles" [14], according to the authors. This technique can be seen

(a) *Using equal sized tiles* (b) *Using tiles of different sizes*

Figure 2.7: Hausener's method results [13]

as an extension or optimization of Hausner's algorithm because it tries to solve some of the algorithm flaws such as the alignment of the tiles, the space between the tiles uneven and some overlapping of the tiles.

These problems arise due to the automation of the algorithm, therefore an interactive tool was developed, allowing the user to intervene in all the parts of the process. Firstly, the user must provide the image it wants to process. Then the user must highlight the main objects in the image, that is, the objects he wants to highlight. He does this by drawing the lines of those objects with the cursor.

With this information, a "direction vector field" is constructed and it is used to place the tiles of the mosaic in an automated way. The shape of the tiles and the number of tiles to be placed are chosen by the user. After the tile placement, the user can insert or delete tiles to his preference. This interactive step makes this approach's results more aesthetically pleasing and a huge advantage over other approaches. The final results can be seen in Figure 2.8.

Battiato's Method

Sebastiano Battiato developed a technique, described in [15] that tries to highlight the main elements of the image in relation to its background. Thus, to transform an image into a mosaic of this type, the first step is to identify the elements that we want to highlight, which constitute the foreground and the remaining elements of the image will constitute the background. The highlight is made using different ways to place the tiles, that is, the background elements will be drawn using the technique 'Opus Musivum' that consists of placing the tiles on a square

Figure 2.8: Results using the Lars-Peter Fritzsche's Method [14]

grid. At the same time, the foreground elements will be drawn using 'Opus vermiculatum' which consists of placing tiles in a grid that follows the shape of each element.

The identification of the main elements of the image is done using the *Statistical Region Merging* (SRM) technique to segment the image into regions based on color. This technique consists of a statistical analysis whose objective is to determine if two regions should be merged or not, that is, the algorithm starts with one region per pixel and merges regions that are similar in terms of color and proximity. This process is the key to the algorithm's success because it returns the regions of the image, that ideally correspond to the objects presented in the image and the pixels that constitute the edges of the regions obtained.

After the statistical analysis and the regions are identified, it is possible to choose which regions will belong to the foreground and which will be part of the background.

To create the mosaic, the algorithm begins with the distance matrix. It determines the minimum distance from any edge pixel (obtained from SRM) and uses this information to generate the matrix (dtM). Then it uses dtM to derive two additional matrices, which are used to form chains of pixels where the tiles will be placed.

Tile placement can generate two problems: overlapped tiles and tiles that overlap an edge line. If the tiles are overlapped, the part of the new tile to be placed that will overlap the already placed tile is cut. If the tile overlaps an edge line then it is trimmed according to that line Figure 2.9.

RenderBots Method

This approach allows the usage of different image rendering styles using the same framework, through a multi-agent system. This technique has many possible

Figure 2.9: Results obtained using Opus Musivum and Opus Verniculatum [15]

applications, including the creation of mosaics.

A multi-agent system is a set of agents that share the same environment. The agents are named renderbots by the authors in [16] and they are entities that are capable of acting on their own, reacting to the environment they are in and interacting with each other. Their function is to move and paint the image we want to create.

The environment is formed by a set of information provided by image layers, that is, buffers that contain "different kinds of pixel-based information" [16]. Each layer has a unique ID so that each pixel is identifiable in the correct layer. The main layer is the input image, that is, the image to be processed. Then there is the target layer where the final image is built by the renderbots, containing the results of the algorithm and a layer that stores temporary information. There may be additional layers depending on the necessity of more information for the image generation, therefore they can provide information about image edges, gradient information, and object IDs. The additional layers are provided by the use of G-buffers. G-buffers or Geometric Buffers, contain "data about the geometric properties of the surfaces", namely "the depth or the normal vector of each pixel" [17].

Each renderbot has the following data: their position in the environment (x,y), acceleration and position values stored, the bot mass and specific state variables. These agents go through three steps. The Simulate step computes new direction and velocity values according to the data the bot collects from the environment and the other bots around him. With this information, a Force is computed and the direction of that force determines the direction of the movement. The acceleration is computed using that force and the mass of the robot. The velocity is calculated using the old velocity value, the acceleration and the difference between two-time steps in the simulation. The second step is named Move and computes the new position and performs the actual movement of the robot. Finally, in the Paint step, the image is rendered. The renderbot can either leave behind a trace

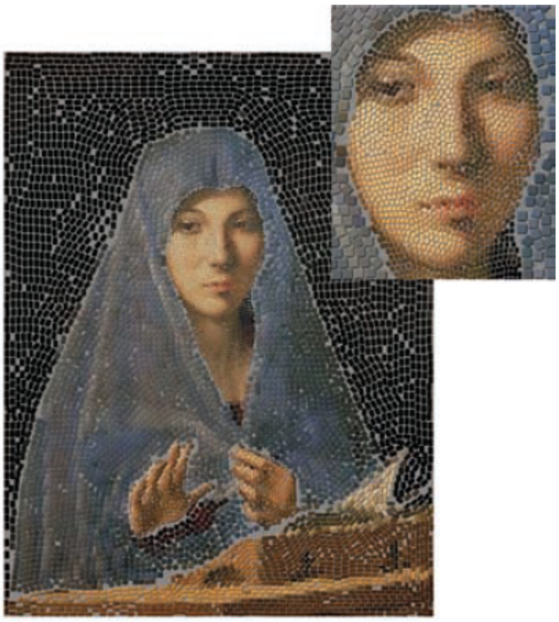


Figure 2.10: Results obtained using RenderBots [16]

or represent the stroke himself when painting.

After the creation of the environment, renderbots are created and distributed by the user or randomly. They can also clone themselves to cover the image. The bot's movement process is done in a loop until the user stops it.

An edge information layer and a specific class of bots called EdgeBots are used to draw the edges of an image. The agents have two possible states: searching and drawing. In the searching state, they search for edges using the edge layer. When they find an edge, the state of the bot switches to drawing and starts following the edge and drawing it. While it draws the edges in the results layer, the agent also leaves markers in the temporary layer so that other bots know that the area is already processed. When the edge ends the state switches to search again. This process requires edge information as input and therefore previous segmentation of the image.

In a mosaic, each bot has a rectangular shape and represents one tile. Each bot applies repelling forces so that they do not overlap. The EdgeBots also apply repelling forces so that the tiles do not overlap the edges. The bots can also rotate according to a given vector field. The color to paint each bot is sampled from the main layer from the pixel with the same position as the bot. The color can also be calculated as the average of all the pixels that the bot covers. The results obtained can be seen in Figure 2.10.

2.2 Kiosk Systems

A kiosk is a software that is always running, ready to be used by whoever approaches it, therefore, this type of software needs to be prepared to be used by a "constantly varying group of users" [18], with a simple interface that should be easy to use.

There are several types of kiosk applications, including information kiosks, which provide information to the user, for example, it may contain a map of the establishment in which the person is located, with the purpose of indicating directions to somewhere or indicating the location where the person is located in relation to the establishment.[18]

Advertising Kiosks are another kiosk type that is used for marketing purposes with the goal of presenting the products of a certain store or giving more visibility to a company or brand. In this type of kiosk, the goal is to maximize the time of interaction with the user. [18]

On the other hand, Kiosks can also be used to provide a service to the user. Service kiosks are designed to be used in a self-service context, for example in a hotel, to book a room, the user can introduce the information needed and complete the process without needing any employee. [18]

Kiosk systems are also used for entertainment purposes. This type of kiosk does not have a task associated with them as they are used mainly to entertain the user with activities, puzzles, or quizzes. [18] This type of kiosk is the most relevant for this project since this is the type of kiosk that is more adequate for the project's purposes.

2.2.1 Kiosk systems at Museums

Museums nowadays have the challenge of communicating with different generations and motivating people's interest in learning and visiting new places, as it is said in [19]. Entertainment Kiosks are a good way of doing that since they provide modern and innovative types of interaction with visitors. Some examples of museums that have implemented this type of interaction are explored in this section.

San Francisco Museum of Modern Art

One example of a museum that has implemented the use of interactive kiosks to provide a better experience to visitors is the San Francisco Museum of Modern Art (SFMOMA). In 2002, the museum announced that it would be using interactive kiosks to provide visitors with a more in-depth understanding of the art on display [20]. These kiosks allow visitors to access a wide range of information about the art and exhibitions.

Museum of Modern Art

In the Museum of Modern Art (MoMA) in New York, the kiosks are placed all over the museum and they provide information about the location of the works of art of the exhibits, helping the navigation of the visitors through the museum. It also contains information about museum events and relevant information about the authors of the art displayed [21].

Tate Modern Museum

The Tate Modern in London has an interactive project named *Bloomberg Connects*, which includes 75 interactive points, and allows visitors to leave comments and post opinions, watch videos, and suggest alternative captions for the artworks on display in the museum [22].

One of the main components of the *Bloomberg Connects* project is the Drawing Bar [22], which is constituted of interactive touchscreens that allow visitors to add their own "tile" to a virtual mosaic, as illustrated in Figure 2.12. The idea behind the Drawing Bar is to represent the imagination of Tate's visitors. Using the sketch pads provided, illustrated in Figure 2.12, visitors can trace a drawing and immediately see it become part of a large projection on the back wall, creating a collaborative digital artwork that is co-created by the visitors of the museum.

Another part of the project is Global Studios, which allows visitors to take a virtual tour of the ateliers of artists from around the world, watch videos of them working, and reply to questions posed by the public. The resulting videos are also published on the internet and displayed on a series of monitors installed in the museum, illustrated in Figure 2.11 [22]. The *Bloomberg Connects* multimedia guide provides information about the artworks, artists, and exhibitions on display at the museum and it is accessed by the visitors using the monitors installed at the museum[23].

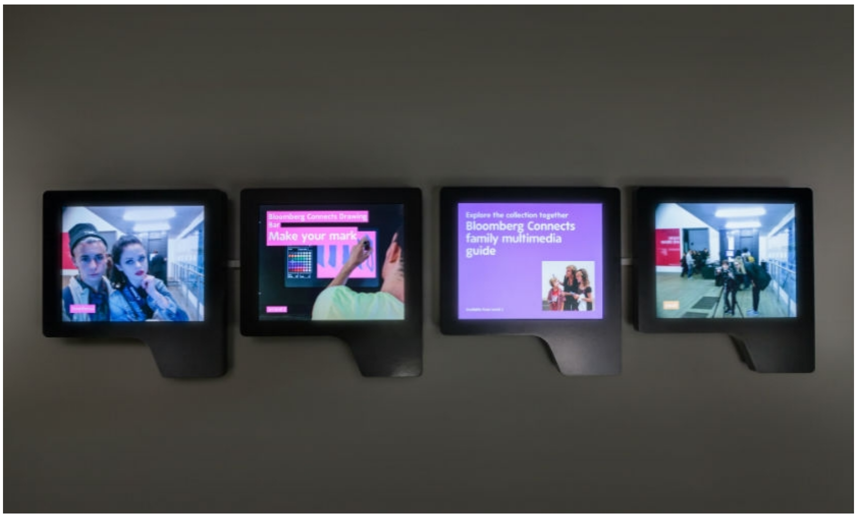
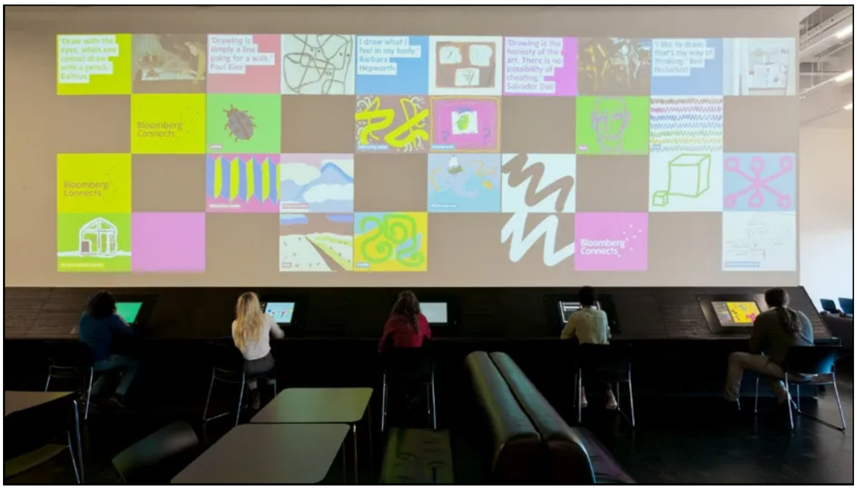


Figure 2.11: Monitors installed in Tate Modern Museum [22]



(a) *Drawing's projection in the wall [22]*

(b) *Sketch pads provided to the users [23]*

Figure 2.12: Drawing bar component of the project Bloomberg Connects at Tate Modern Museum

Heineken Experience

The Heineken Experience is a museum located in Amsterdam and takes place in the former Heineken brewery [24]. Nowadays, visitors can see the entire beer-making process by looking inside the old factory. At the end of the visit, there are several fun interactive activities that visitors can explore with the purpose of entertainment.

One of them is a kiosk that has two bicycles, a green screen, a camera to capture video, and a screen to display the video captured as shown in Figure 2.13. Visitors can sit on the bikes and take a video of them strolling through the city streets. At the end of the recording, a QR code is displayed on the screen that allows visitors to save their video as a souvenir of their visit.

Another kiosk at this museum is shown in Figure 2.14 and it is composed of a touchscreen positioned on a metal structure and a camera. In this kiosk, the visi-

Figure 2.13: Interactive kiosk with bikes in the Heineken Experience Museum [25]

tors can take a picture of their faces that will be placed in a Heineken beer-related background. The user can choose from three different background options. After choosing, the image captured by the camera appears and a timer appears. The user has 5 seconds to position themselves as they prefer. When the five-second timer ends, the picture is taken automatically and the final photo with a fun background is generated. When generated, the image is presented to the visitor as well as a QR code that allows him to save the image on the phone.

Figure 2.14: Interactive kiosk to take a picture in the Heineken Experience Museum [26]

In conclusion, the usage of interactive kiosks has evolved from being used for information and navigation at the museum to the visitor's engagement and participation in museum activities. The Kiosks can also be used to entertain the visitors and allow them to take a digital souvenir with them to remember their visit to the museum. This type of interaction may provide bounding moments between visitors and an immersive and interactive exploration of the museum exhibits.

Chapter 3

Methodology

This chapter presents the methodology, work plan, and risk evaluation related to this project.

This project aims to develop and implement one algorithm to create a mosaic and then integrate it into a kiosk application that can be used by people of all ages. The implementation of the algorithms is made using JavaScript, as this is the technology used to develop the editor, thus making integration simpler. The kiosk was implemented as a web application.

The work methodology used on this project is the following: the project was divided into phases, that is, stages of work that will be developed. The phases should be completed sequentially.

1. Phase 1 - Research

   The first phase of the project is to research the elements needed to elaborate the kiosk, that is the algorithms that were already developed and the kiosks that are in use at museums. Therefore this phase is composed by two tasks:

   - Task 1 - Research of algorithms

     To begin, an analysis of existing algorithms for transforming a picture into a mosaic will be conducted. After analyzing the existing algorithms, one will be chosen to be implemented based on its characteristics, more specifically, the resemblance with the roman mosaic.

   - Task 2 - Research the use of kiosks in museums

     To design the kiosk application interface, it is essential to research the existing kiosk applications used in museums, focusing on hardware requirements and the background knowledge needed to use them.

2. Phase 2 - Development of the prototype

   The second phase of the project consists of developing a prototype and evaluating the results obtained. Therefore this phase contains two tasks:

   - Task 1- Implement prototype

The first approach to the implementation of the project is to implement a prototype, that is, a simple web application that allows the user to take a picture and turn it into a mosaic using a simple algorithm.

- Task 2 - Evaluating results

  In this task, a reflection of the results obtained is made. This task aims for a practical understanding of the design possibilities for the kiosk implementation

3. Phase 3 - Implementation of the algorithm

   In this phase of the project the algorithm the Battiato's algorithm was implemented

   - Task 1 - Research more detail about the algorithm's implementation

     After choosing the algorithm, deeper research needs to be done in order to understand all of the implementation details and to know if any adaptation will be needed to apply the algorithm in the project context.

   - Task 2 - Implementation of tile arrangement

     The implementation of the algorithm can be divided into two stages, the tile arrangement of the mosaic and the color choice of each tile. A first implementation approach will solve the problem of the tile arrangement, that is, placing the tiles in the correct places.

   - Task 3 - Implementation of the coloring of the tiles

     After the tiles are placed in the right way, each tile will be colored as the algorithm describes.

   - Task 4 - Adaptation of the algorithm for the kiosk

     In this task, the algorithm will be adapted to represent the features of the human face better and to be fully automatic.

   - Task 5 - Evaluation of the algorithm

     The evaluation of the algorithm will be made in terms of performance and in terms of visual mosaic results.

4. Phase 4 - Implementation of Kiosk

   In this phase, the kiosk application interface will be implemented according to the information obtained from the real examples searched in phase 2. This phase contains two tasks:

   - Task 1 - Interface development

     The features that the interface should have are allowing the user to take a picture by pressing a button, present the mosaic generated to the user, generate a QR code and present it to the user to allow him to access the mosaic on his device. Although the photo-taking feature is implemented in the prototype, it could be improved with a timer, that is, when the user clicks the button to take the photo, they would have a few seconds to position themselves for the photo.

- Task 2 - Integration of mosaic algorithm

  Some changes will need to be made to the algorithm developed in order to integrate it into the kiosk.

5. Phase 5 - Kiosk Evaluation

   To evaluate the kiosk a laptop and a touch screen will be used. The kiosk will be running for other people to test it and comment on their experience. A form will be used to document the answers of the testers. Two tasks belong to this phase:

   - Task 1 - Evaluate kiosk in a lab environment

     The first tests will be performed in a lab environment, that is, in the Department of Informatics Engineering of the University of Coimbra.

   - Task 2 - Evaluate kiosk in a real environment

     Using the feedback of the tests performed in the previous task, some improvements might be made to the kiosk application. Then the kiosk will be tested at the Museum of Conimbriga.

6. Phase 6 - Documentation

   The document will be written throughout the entire project.

3.1 Work plan

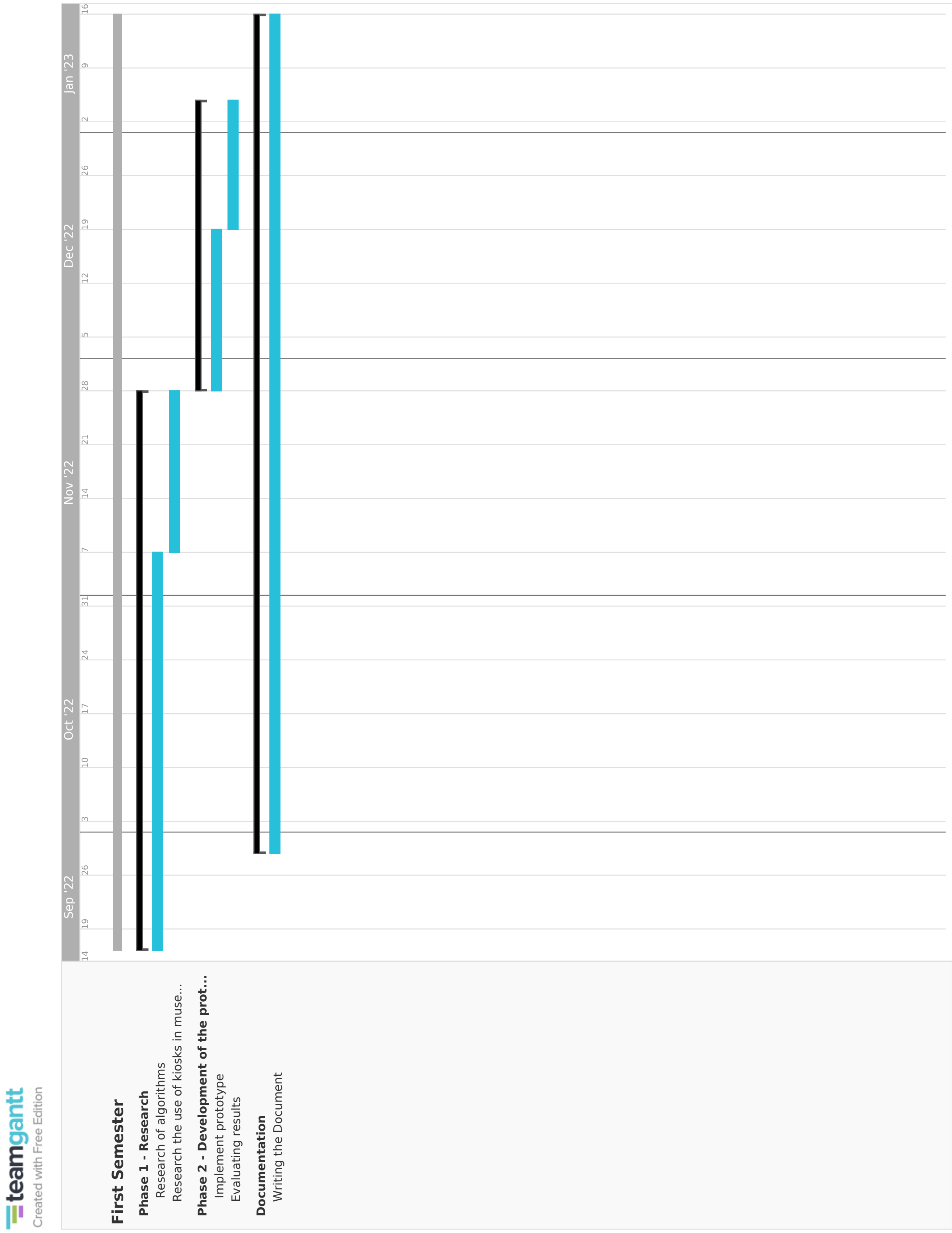
In this section, the work plan is defined throughout the entire project duration regarding the work phases already defined.

3.1.1 First Semester

The tasks related to the first semester are presented in a Gantt diagram below to provide a view of the previously defined tasks and phases distributed in a timeline.

The research of algorithms and kiosk examples was completed, corresponding to the phase 1. The research relative to the algorithms was made one algorithm at a time as well as the kiosk examples. After the research was completed, the prototype was implemented, firstly Haeberli's algorithm was implemented because this was a simpler algorithm, therefore all the steps required to implement it were easier to understand and program correctly. After the implementation was finished, the algorithm was tested with some images in order to evaluate the resulting mosaics. The results are further detailed in chapter 4.

Then, in phase 2, the design of the web application was implemented, that is, a camera to allow the user to take a picture using the webcam of the device they are using and use that picture as input of the algorithm, to be turned into a mosaic. After this, the interaction with the interface was evaluated and changed into a more pleasant layout in an iterative way until satisfying results were obtained. Using the feature of the camera, the algorithm was tested again using pictures taken with the webcam.


teamgantt
Created with Free Edition
First Semester
Phase 1 - Research
Research of algorithms
Research the use of kiosks in muse...
Phase 2 - Development of the prot...
Implement prototype
Evaluating results
Documentation
Writing the Document
Sep '22
Oct '22
Nov '22
Dec '22
Jan '23

3.1.2 Second Semester

The tasks and phases relative to the second semester are presented in the Gantt diagram below with the tasks in the color blue. This diagram differs significantly from what was expected, with some tasks taking longer to complete than expected. The initially constructed Gantt chart is shown below, with the tasks in different colors.

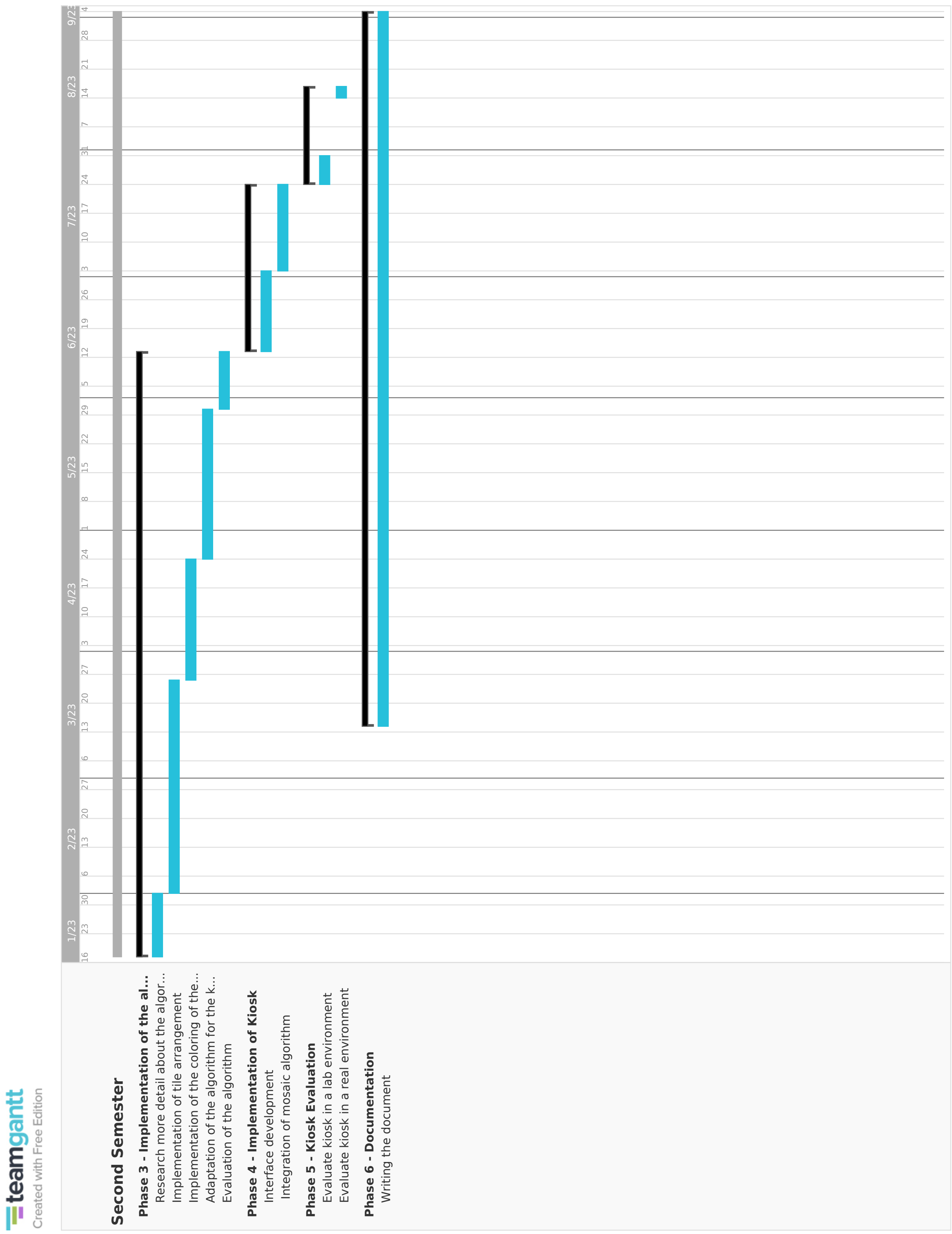
Firstly, as explained, in phase 3 there will be a need to perform deeper research to fully understand all the details of the algorithm. Then the implementation of the algorithm will be done as it is described in the article. Then an adaptation of the algorithm will be done in order to fit our case better.

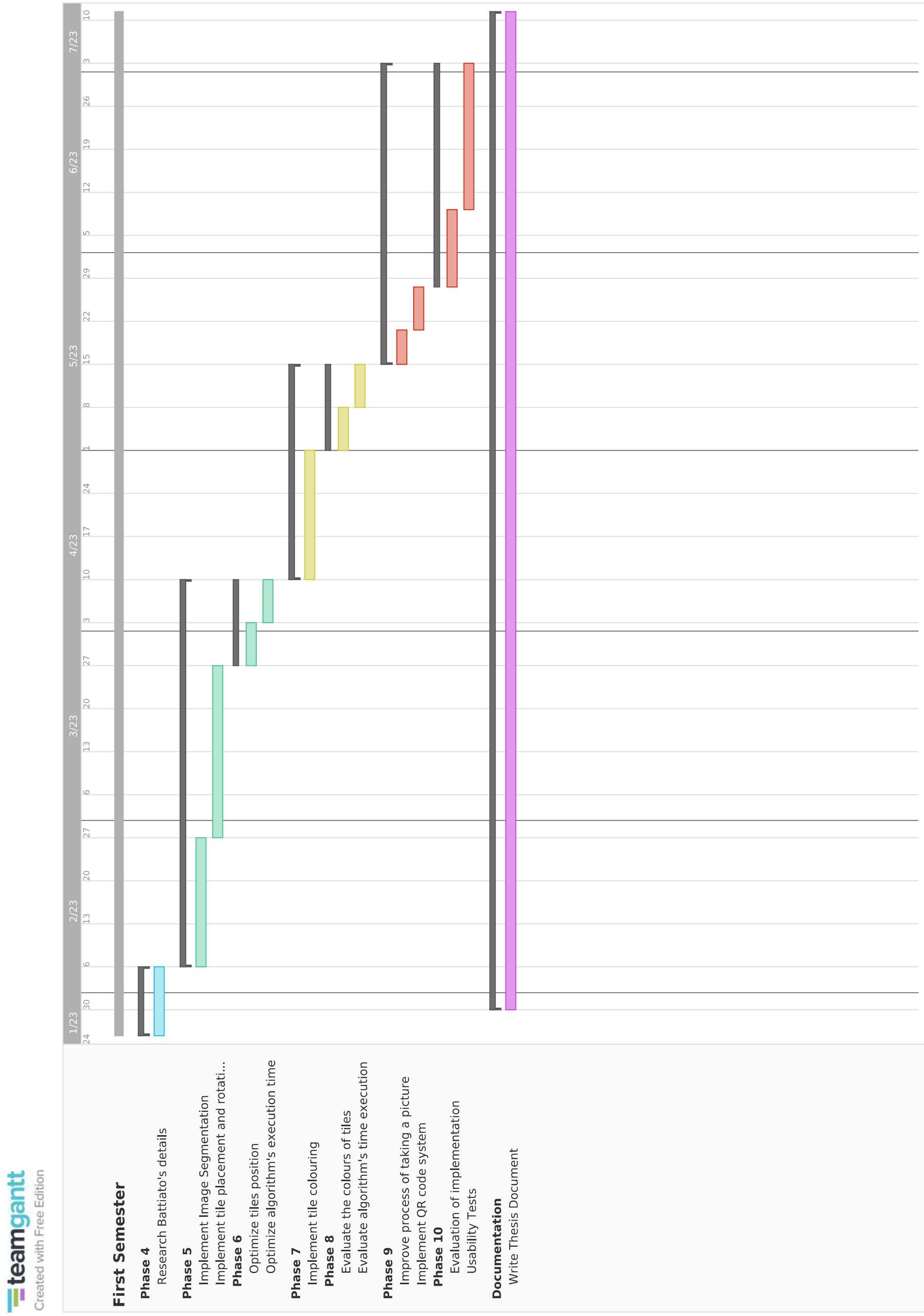
After the algorithm's implementation, the kiosk application will be implemented and the algorithm will be integrated into the kiosk application. After the implementation is made, the kiosk evaluation will be performed with real users, and the results will be analyzed.

3.1.3 Reflections

The project was developed as planned in the work plan described and all the tasks of all phases were completed. Some tasks took longer than expected to execute such as the implementation of the tile placement and the tile cutting due to complications in the implementation of these steps of the algorithm.

The phases and tasks presented had to be done sequentially since the work developed in each task is dependent on the previous tasks and phases. Therefore, the project was delayed by the implementation of the algorithm.


teamgantt
Created with Free Edition
Second Semester
Phase 3 - Implementation of the al...
Research more detail about the algor...
Implementation of tile arrangement
Implementation of the coloring of the...
Adaptation of the algorithm for the k...
Evaluation of the algorithm
Phase 4 - Implementation of Kiosk
Interface development
Integration of mosaic algorithm
Phase 5 - Kiosk Evaluation
Evaluate kiosk in a lab environment
Evaluate kiosk in a real environment
Phase 6 - Documentation
Writing the document
1/23
2/23
3/23
4/23
5/23
6/23
7/23
8/23
9/23


teamgantt
Created with Free Edition
First Semester
Phase 4
Research Battiato's details
Phase 5
Implement Image Segmentation
Implement tile placement and rotati...
Phase 6
Optimize tiles position
Optimize algorithm's execution time
Phase 7
Implement tile colouring
Phase 8
Evaluate the colours of tiles
Evaluate algorithm's time execution
Phase 9
Improve process of taking a picture
Implement QR code system
Phase 10
Evaluation of implementation
Usability Tests
Documentation
Write Thesis Document
1/23
2/23
3/23
4/23
5/23
6/23
7/23

3.2 Risk Evaluation

An evaluation of the risks of this project was made in order to predict and mitigate potential problems that might occur during the project's phases.

- **Risk 1 - Problems with understanding or implementing every detail of the chosen algorithm**

  In order to implement the algorithm, it is necessary to understand the algorithm deeply enough to implement it as it should. Since some details can be misunderstood, badly interpreted, or might be missing in the reference article, there might be a delay in the implementation. In case this situation occurs, another algorithm can be chosen with less complexity or the algorithm can be adapted to a simpler implementation.

- **Risk 2 - High execution time of the algorithm**

  The algorithm implemented might not execute fast enough to be used in a real situation. The interaction of the user with the kiosk must be fast so that he does not have to spend too much time waiting for the mosaic to be generated. To detect this problem it is necessary that the time of execution of the algorithm is measured when evaluating the algorithm's implementation. If it occurs it can be mitigated by implementing the algorithm in a more efficient way.

- **Risk 3 - Inadequate Interface of the Kiosk**

  A kiosk application implies fast interactions with the users. If the process has too many steps or if the steps are not easily understood the users might lose interest in using it. To avoid this risk, the interface should be submitted to usability tests.

- **Risk 4 - Results in real usage might be worse than the ones obtained in laboratory context**

  In real-life usage of the application, the resulting algorithms might have less quality than the ones obtained while testing because, despite the tests made to the algorithm, certain picture aspects might influence negatively the mosaic generated, for example, the background of the picture and bad lighting. To prevent this from occurring, images close to the actual context of software use will be used for testing, in order to optimize the algorithm's behavior in the best way.

After weighting the probability of occurrence of the risks presented and the severity in case they occur, a matrix was made, which is presented in Figure 3.1 in the following subsection.

3.2.1 Risk Mitigation

To mitigate the risks detected before the implementation of the project, substantial efforts were made.

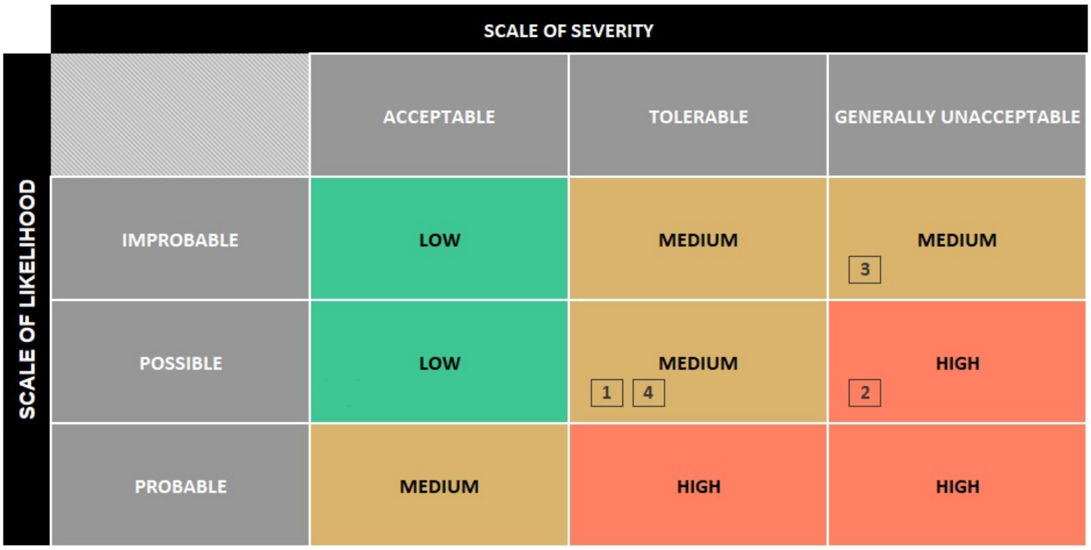


Figure 3.1: Risk matrix

The **first risk** mentioned the difficulty of understanding or implementing every detail of the chosen algorithm. This was overcome by meticulously analyzing the article where the algorithm was described, paying attention to all the steps included in the article, and consulting it throughout the implementation when doubts arose. This obstacle was well overcome.

The **second risk** identified mentioned the high execution of the algorithm. Unfortunately, this risk could not be avoided and the time of execution of the algorithm was criticized by some users of the kiosk. This problem occurred because of the difficulty in optimizing the algorithm and because it was not well defined at the outset how long would be ideal for running the algorithm, and the initially defined acceptable time limit, that is, about 30 seconds, was considered high by part of some kiosk users.

To avoid having an inadequate interface for the kiosk, as referred in **risk 3**, some examples researched in the initial phase of the project were taken as examples to design the interface. Two test phases were made, that is, there was a laboratory test phase that was made in order to identify problems with the design of the kiosk. Using the user's feedback was used to correct some problems before the tests were performed in a real environment, that is, in a museum.

As for **risk 4**, some problems occurred with regard to the luminosity of the place where the kiosk was put into operation because the place where the laboratory tests were carried out had a higher luminosity than the place where the museum tests were carried out. Thus, the mosaics produced in the museum were sometimes lacking in color due to the light in the captured photo being reduced. As for the background, in both moments of the test, it was possible to find a neutral wall that did not affect the quality of the final mosaic. This was possible due to an examination of the test site before the tests were carried out.

Chapter 4

Developed Work

In this chapter, the developed project is described. The implementation was divided into two parts: the mosaic algorithm and the kiosk application. In the algorithm's implementation process, a preliminary study was made. Then the chosen algorithm was implemented and then some adaptations were made in order to get the best result for this specific project. The kiosk interface was implemented so that it could be used by people of any age, providing a simple interaction easily understandable by anyone.

4.1 Preliminary Study

For a preliminary study, Haeberli's algorithm was implemented. This algorithm is the simplest one of the ones explored in the state of the art chapter. The algorithm was integrated into a program where the user can take a picture using a webcam and the picture is then presented in a mosaic form. The user can also change the number of cells that the mosaic will have and see the changes in real time for the same picture.

The pseudo-code for this algorithm is presented below. First, the image is loaded to a variable. Then, a Voronoi diagram is generated, using randomly located points as input. This diagram is not adapted to the features of the image, that is, it is a generic Voronoi diagram with no relation to the image. After this, the edges, vertices, and sites are drawn into a canvas, to be displayed to the user. Afterward, the cells of the diagram are painted, sampling the color used from the image. This process is done by getting the coordinates of the cell's site and getting the image's color on those coordinates. The cell of the diagram is painted with the color obtained.

```
Pseudocode:

Im <- CALL getImage()
Vd <- CALL generateVoronoi()
Draw(Vd.edges)
Draw(Vd.vertices)
Draw(Vd.sites)

For each site S
    x,y <- getPixelCoordinates(S)
    r,g,b,a <- getImageData(x,y)
    PaintVoronoiCell()
```

(a) *Input image*

(b) *Mosaic generated with 200 sites*

(c) *Mosaic generated with 10000 sites*

Figure 4.1: Results of the Haeberli algorithm for a human face

To test the algorithm it was used a picture of a person's face generated by an AI program [27]. From the results presented in Figure 4.1, we can conclude that the mosaic obtained is not the most adequate for this project's purposes. Since it does not produce squared tiles, it does not approximate enough to the roman mosaic. Furthermore, it does not define the object's edges in a satisfactory way. More specifically for this project, an individual's face requires an algorithm that captures the face features with more detail.

For these reasons, another algorithm was chosen to be used in the kiosk application. From the algorithms studied, Battiato's is the one that could represent the roman mosaic techniques better, therefore, that algorithm is the one to be implemented and integrated into the final application.

4.2 Battiato's Algorithm

After analyzing the resultant mosaics of the preliminary study, it was concluded that the algorithm implemented was not adequate for the kiosk due to its characteristics. After evaluating the studied algorithms, the one that fitted better the kiosk for its resemblance with the Roman mosaics was Battiato's algorithm. Some adaptations were made to the original algorithm in order to adapt it to our specific usage of the algorithm, that is, it was adapted for better performance when mosaicing human faces.

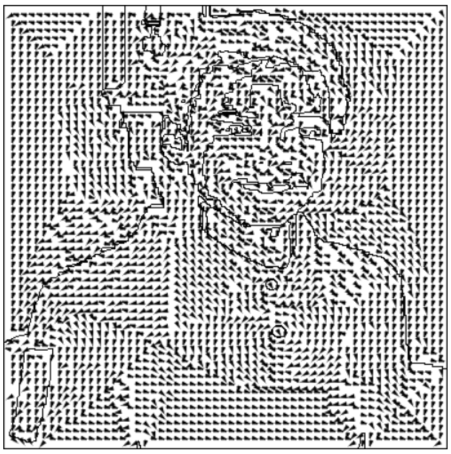
In the first approach the Battiato algorithm was implemented as it is described in [15]. The results obtained are shown in Figure 4.2e. For some images, the algorithm presents good results but in the case of the face, there are more regions detected than desired Figure 4.2b, that is, the face is divided into multiple regions but once the division is made by color, if the face has shades that will affect the segmentation process, therefore, the segmentation is not made in the most desirable way.

Another problem with the resulting mosaic is that some regions don't have any tiles on them. That's because the regions generated were so small that there was no space for a tile to be placed.

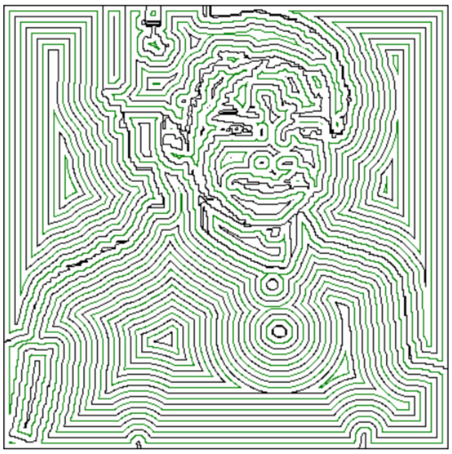
The pseudo-code of this algorithm is presented below.

(a) Segmentation with srm

(b) Guidelines



(c) Gradient matrix



(d) Level line matrix

(e) Final result

Figure 4.2: Example of the result achieved with Battiato's algorithm.

```
Pseudocode:
Im <- CALL getImage()
ImageRegions <- srm(Im, Qlevels)
```

```
guidelines <- Call getGuidelines(ImageRegions)
dtM <- CALL distanceTransformMatrixCalc(Im)
For each pixel of Im
    gM(x,y) <- arctan (dtM(x,y+1)-dtM(x,y-1) / dtM(x+1,y) - dtM(x-1,y)
    IF module (dtM(x,y), 2tileSize)=0
        llM(x,y) <- 1
    ELSEIF module (dtM(x,y), 2tileSize)=tileSize
        llM(x,y) <- 2
    ELSE
        llM(x,y) <- 0
    END IF
END FOR

Repeat
    IF llM(x,y) == 2 && is not processed
        FOR each adjacent llM(x,y) == 2
           generatedTiles <- CALL generateTile(gM, tileSize, mSize )
Until llM(x,y) == 2 are all processed

cutTiles <- CALL cutTiles(placedTiles)
CALL colorTiles(cutTiles)
```

To solve the presented issues, an adaptation of the algorithm was made and its described in the next section.

4.2.1 Battiato's Algorithm Adaptation

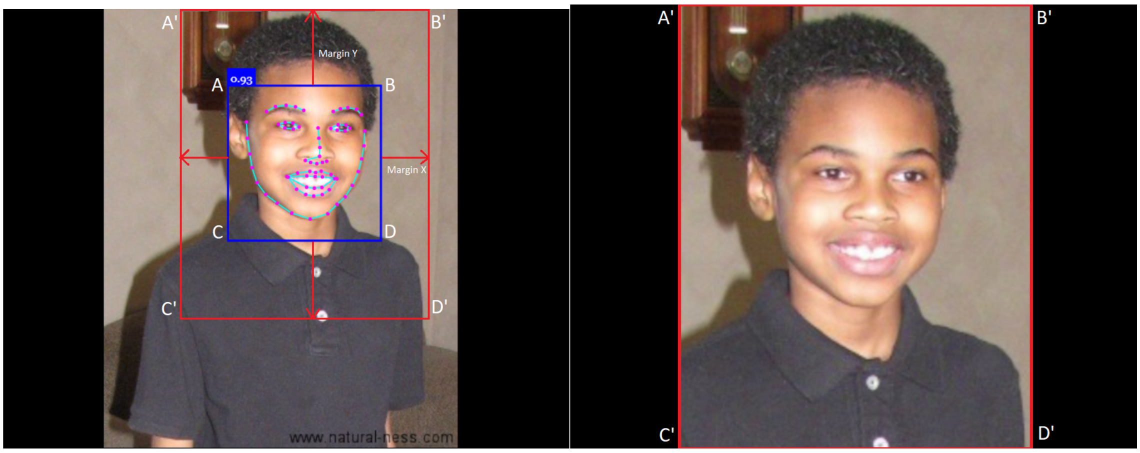
The algorithm implemented was based on Battiato's algorithm, referred in the state of the art section. This adaptation of the original algorithm has the following steps: image capture; face detection; cropping; calculation of dtm, llm, and gradient matrices; generate tiles; trim tiles; color tiles.

The first step of the algorithm is **image capture** which consists of taking a picture with a camera device. The picture will only be used to build the mosaic if a human face is detected in the picture.

The **face detection** is used to segment the image. Using the face-api.js [28], the face landmarks such as the jawline, eyebrows, eyes, mouth, and nose are detected Figure 4.5a. These landmarks form polygons that are used as guidelines for the next steps of the algorithm. The obtained guidelines can be seen in Figure 4.5b. The algorithm considers the face and its elements as the foreground and the rest of the image is considered background.

The image obtained is **cropped** to ensure that the face of the person is centered in the final mosaic Figure 4.3. First, we get the bounding box of the face (ABCD) as returned by the face detection library. Then a margin is added to the sides of the bounding box (margin x and margin y) resulting in a bigger square (A'B'C'D')

Figure 4.3a. The horizontal margin is 30% of the width of the bounding box and the vertical margin is 50% of the height of the bounding box. This square is then mapped in a new image that has the final intended dimensions Figure 4.3b.



(a) Picture before the crop

(b) Picture after the crop

Figure 4.3: Example of cropping mechanism

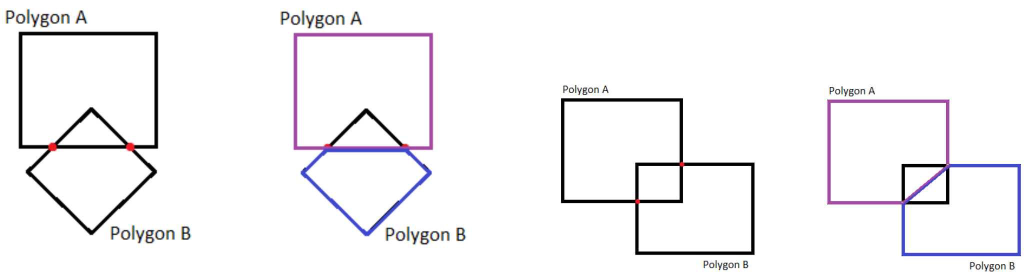
Then the **calculation of dtm** (distance transform matrix) is made. That consists of calculating the minimum Euclidean distance of each pixel to any guideline and storing the values in a matrix. Using the values of the dtm matrix, two other matrices are calculated, the **gradient matrix** and the LLM (level line matrix). The gradient values will determine the rotation of the tiles of the mosaic. The **level line matrix** (LLM) determines the lines where the tiles will be placed.

After the calculations are done, the tiles are placed in the mosaic, following the lines stored in the LLM.In the case of the face elements like mouth, eyes, and eyebrows, the placement of the tiles is made using a value of the size of the tiles that is 60% smaller than the one used for the background and for the rest of the face, defined by an input parameter of the algorithm. To get the effect of having extra lines around the foreground objects, in this case, the face, The jawline polygon obtained by the face detection is scaled in a way that allows a certain number of extra rows, defined by the parameter snakesize, to fit inside the polygon. The 'Opus vermiculatum' technique will be used in the pixels inside the scaled polygon.

The next step of the algorithm is to **generate the tiles**. The tiles are generated with their centers separated by the size of a side of the square. The size of the side of the square is previously defined before the start of the algorithm. The placed tiles are stored as polygons in a quadtree structure containing their width, height, center coordinates, and the vertex points of the square with the rotation of the gradient already applied to them. When a square is generated, the space it uses from the LLM is marked as occupied. The placement of tiles ends when every place of the LLM is marked. The result of the placement of the tiles is shown in Figure 4.5e.

After the tile placing two problems occur: the tiles may overlap each other and the tiles may overlap a guideline. To solve those problems the **tiles are trimmed**.

To detect the overlapping tiles, a quadtree is used and it detects the existing collisions. When the tiles are overlapping two cases may occur: if at least one vertex of one tile is inside the other tile, then only the tile that has its vertex inside the other is cut, as shown in Figure 4.4a; if both tiles have vertices inside one another, both tiles are cut as shown in Figure 4.4b.



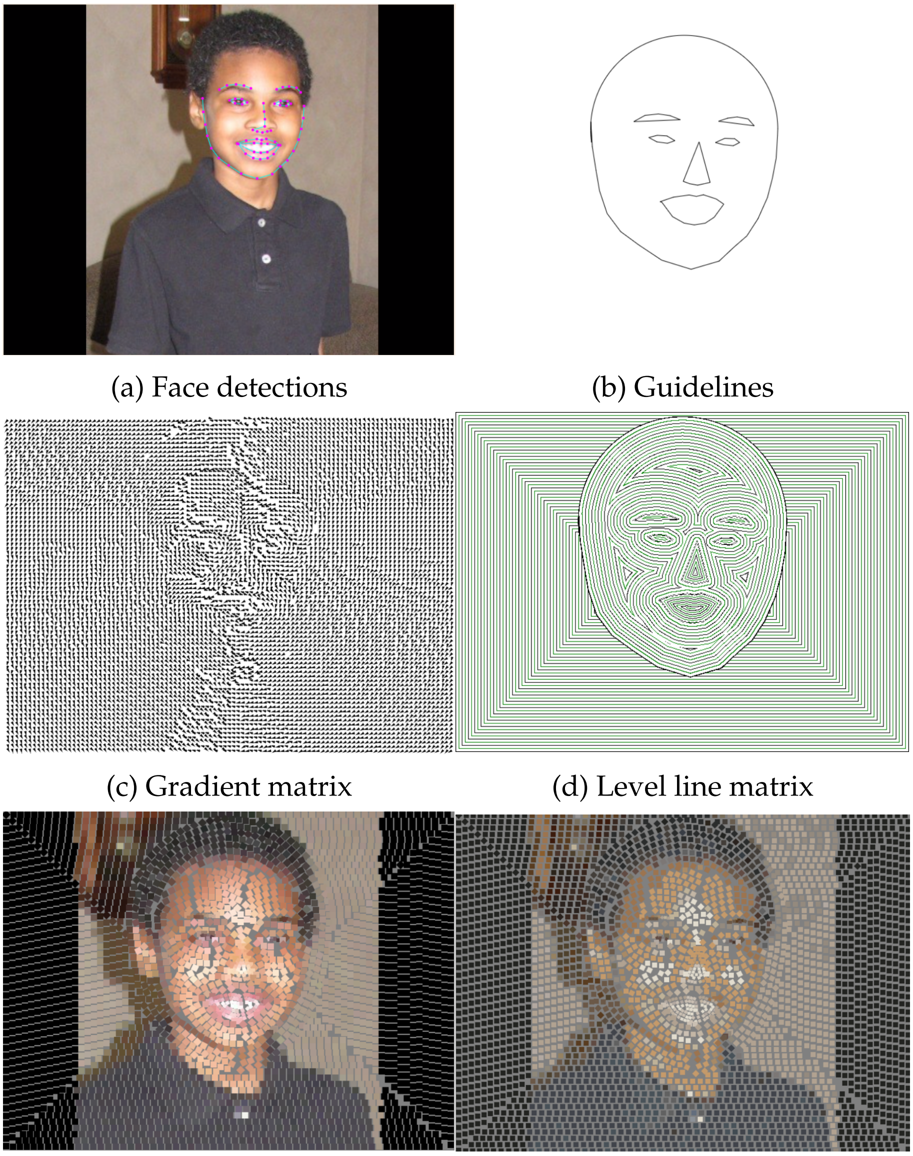
(a) One vertex of one polygon intersects the other polygon

(b) both polygons intersect each other with one vertex

Figure 4.4: Cut cases

The final step of the algorithm is to **color the tiles**. Each tile is drawn on the canvas using the coordinates of its vertices. To color the tiles, a color palette is used. To generate the color palette we used the Image Picker tool from the Colors website [29] to automatically generate a 39 color palette from 4 photographs of Roman mosaics from the Conimbriga site. To apply a color to a tile, our algorithm samples the RGB color of the original image from the correspondent coordinates of the center point of a square and compares that color with all colors of the palette, calculating the difference between colors using the color space CIELAB, once this color space is more precise than RBG because the color is represented in three dimensions. The color of the palette with less difference from the sampled color is the one used to paint the square. The generated mosaic is shown in Figure 4.5f.

This algorithm produces mosaics that have more similarities with the Roman mosaic then the Haeberli's due to the tile shape being squared and the tiles being separated, giving the effect of the real mosaics.



(a) Face detections (b) Guidelines

(c) Gradient matrix (d) Level line matrix

(e) Tile Placement (f) Final result

Figure 4.5: Example of the result achieved with Battiato's algorithm Adaptation

```
Pseudocode:
mSize > 0
tileSize > 0
0 < space <= 7

Im <- CALL RetrievePhoto()
faceLandmarks <- FaceDetection(Im)
guidelines <- processFaceLandmarks(faceLandmarks)
foreground <- FacePolygon + FaceElements
background <- Im - FacePolyon
dtM <- CALL distanceTransformMatrix(guidelines)
For each pixel of Im
    gM(x,y) <- arctan (dtM(x,y+1)-dtM(x,y-1) / dtM(x+1,y) - dtM(x-1,y)
    IF pixel belongs to FaceElements
        tileSizeAux=tileSize*0.6;
    Else
        tileSizeAux=tileSize;
    END IF

    IF  (dtM(x,y) mod 2tileSize) = 0
        llM(x,y) <- 1
    ELSEIF (dtM(x,y) mod 2tileSize) = tileSize
        llM(x,y) <- 2
    ELSE
        llM(x,y) <- 0
    END IF
END FOR

Repeat
    IF llM(x,y) = 2 AND is not processed
        FOR each adjacent llM(x,y) = 2
            generatedTiles <- CALL generateTile(gM, tileSize, mSize )
Until llM(x,y) = 2 are all processed

cutTiles <- CALL cutTiles(generatedTiles)
CALL colorTiles(cutTiles,pallete)
```

4.3 Kiosk Application

The kiosk application objective is to be used by museum visitors, complementing the overall experience with a quick interaction with the kiosk. The kiosk allows the user to take a picture of himself and to save the mosaic generated with that picture. The user can also choose which type of mosaic they want, having three different options to choose from.

The application flow shown in Figure 4.6 was thought to be simple and to be used by the visitors autonomously. It starts with a homepage with a slideshow of mosaics and a start button. When the start button is pressed, the user is directed to a page with the mosaic options to choose from. After choosing, the user can take a picture of his face that will be transformed into a mosaic or return to the homepage. If he takes a picture, the process of creating the mosaic starts and while it is being generated the user is presented with a page with an animation over the picture taken and a loading bar. When the mosaic is created the user is presented with a page with the mosaic and a QR code that allows the user to save the mosaic on a mobile device. After that, the user can press the button "Restart" to do the process again. In case of inactivity for one minute, the kiosk redirects to the homepage.

The kiosk implementation was made using HTML, CSS, JavaScript, and PHP. These technologies were used because the kiosk was meant to work locally with the minimum use of external resources. The interfaces use only HTML and CSS. The algorithm runs in JavaScript. When the execution of the algorithm ends, the mosaic is saved in a PHP server. Then the server generates a Qr code with the link to the mosaic saved. The Qr code is also saved in a server folder.

The interfaces are shown in Figure 4.8. The Homepage Figure 4.7a has a carousel of images of generated mosaics in order to get the attention of the visitors that pass in front of the screen. There is a toggle button in the top right corner that allows the user to change the language to English if needed. The start button and all of the buttons, in general, are green because that color is associated with a positive outcome.

The screen Figure 4.7b is where the user chooses the type of mosaic they would like to generate. The different types of mosaics are a result of different values for algorithm parameters. The differences between each mosaic are detailed in the next chapter.

To take the picture, in the screen Figure 4.7c, after pressing the button "take photo" a timer with 5 seconds appears on the screen, so the user has 5 seconds to position themselves for the photo, preferably with their face inside the red circle, whose purpose is to help to frame the face in the photo. The button "back" is red because this color is typically associated with caution or stopping.

The loading bar and the animation in the screen Figure 4.8a were made to let the user know that the process of generating the mosaic is running and to give an idea of how much time the process will take.

The last screen is the one shown in Figure 4.8b where the mosaic generated is presented to the user and the QR Code to save it on the phone is also presented. The button to restart is green for the same reason as the others.

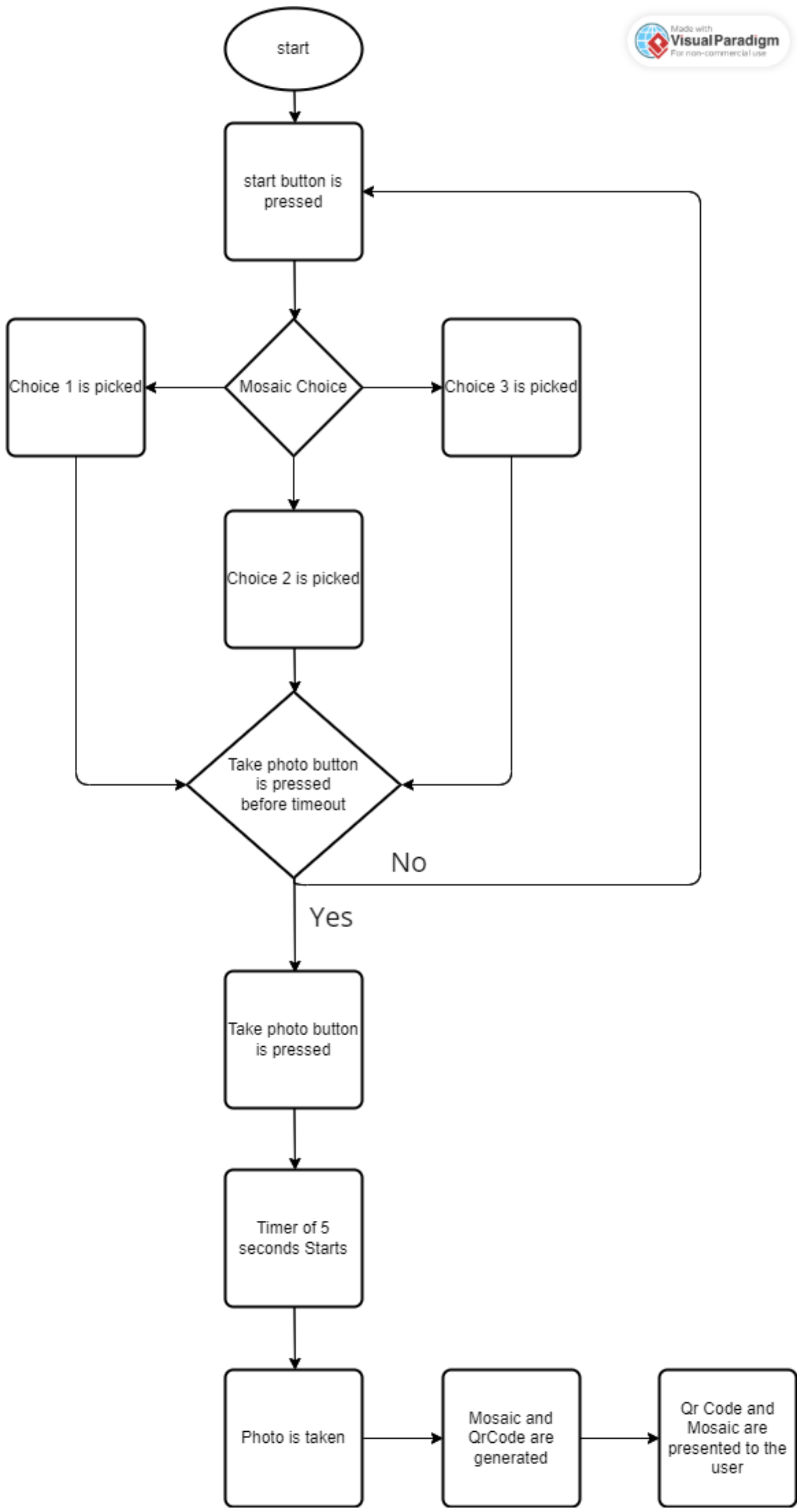
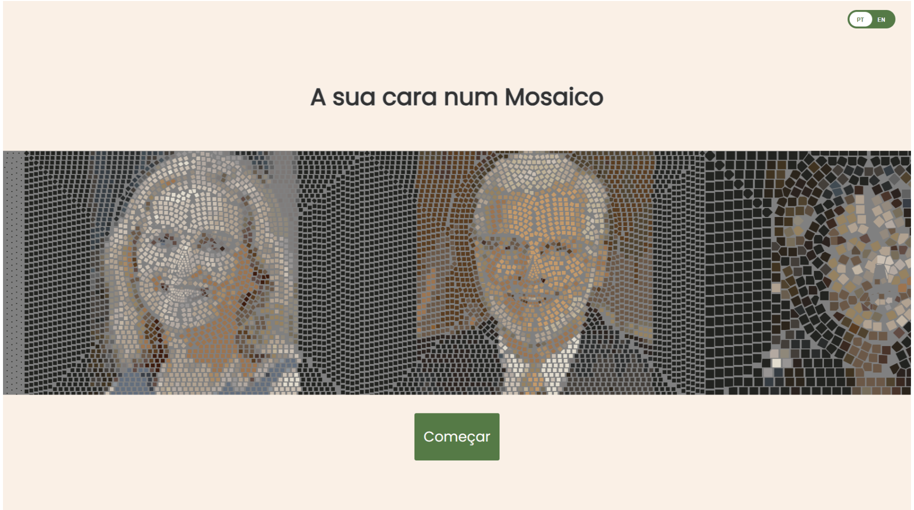


Figure 4.6: Flow chart of the kiosk application

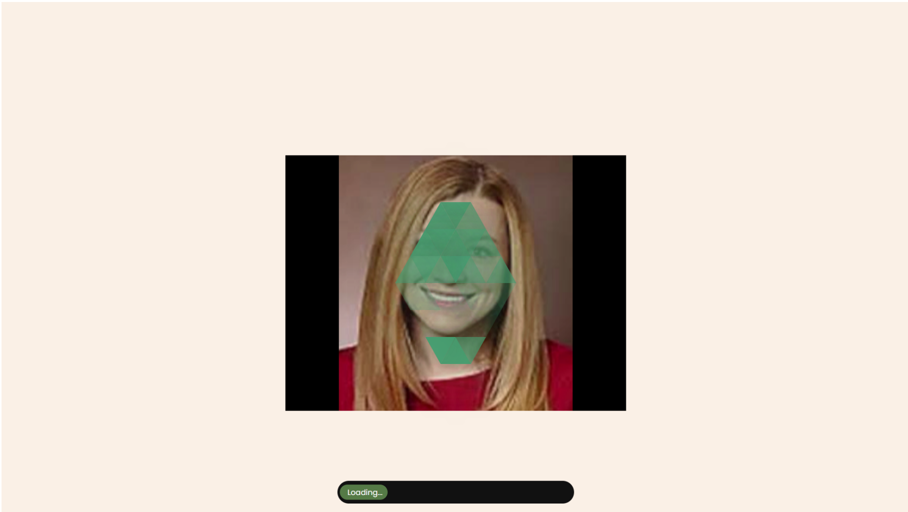


(a) Homepage

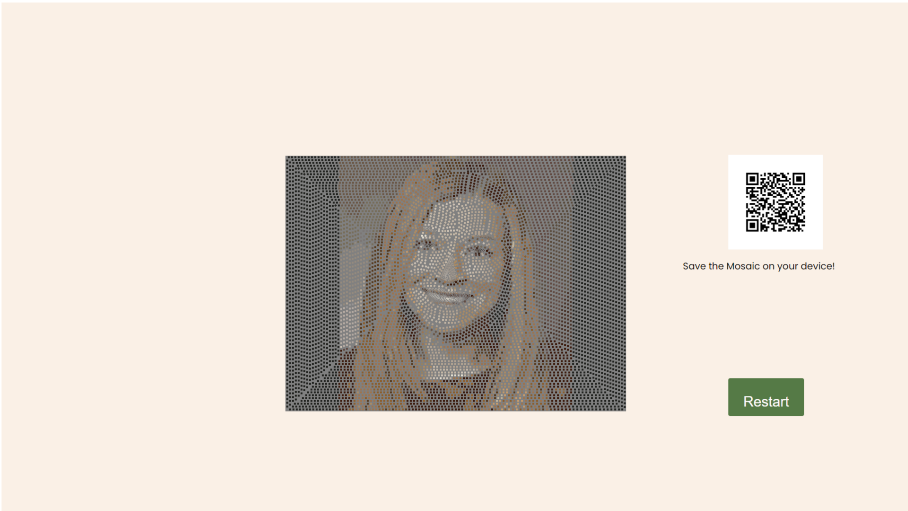
(b) Mosaic Choices

(c) Page to take picture

Figure 4.7: Kiosk interface



(a) Loading page



(b) Page with generated mosaic

Figure 4.8: Kiosk interface

Chapter 5

Evaluation

The evaluation of the algorithm was done in four ways at different stages of implementation. Firstly, after implementing the algorithm, it was evaluated in terms of performance, which is a very important aspect for the kiosk, as mosaic generation can only take as long as the user is willing to wait. The algorithm was also evaluated regarding the visual appearance of the mosaics, using a form to collect information about the preferences of the different types of mosaics generated.

After implementing the kiosk, it was evaluated in another testing phase in which the kiosk was put into operation so that it could be tested by real users. First, this evaluation was carried out in a laboratory, that is, in the Department of Informatics Engineering at the University of Coimbra. In a second testing phase, the kiosk was evaluated at the Monographic Museum of Conímbriga by its visitors.

5.1 Performance Evaluation

After the algorithm implementation, the performance evaluation was made in order to verify the time of execution of the algorithm. This evaluation is very important because the time needed to generate the mosaics has a high impact on the user experience of the kiosk. Furthermore, in this evaluation, the mosaics were also analyzed in aesthetic terms, with a reflection on the parameters that work best for the representation of faces with the implemented algorithm.

To test our algorithm, we used the UTKFace (In-the-wild Faces) dataset – an image data set of human faces [30] categorized according to the person's age, ethnicity, and gender. We selected a small subset of 36 images, making sure that all possible combinations of ethnicity and gender for the ages of 10, 20, 40, and 70 years old. These age values were selected to provide a wide range of different facial features. The purpose is to verify whether the visual results differ when the face detection and mosaic rendering algorithm is applied to faces of people of different ages, races, and genders. In this preliminary evaluation, however, we report only the results of running our algorithm over 4 photos (see Figure 5.1), classified by UTKFace as: 10 years old Black Male; 20 years old Asian Female; 40 years old White Female; and 70 years old Indian Female.

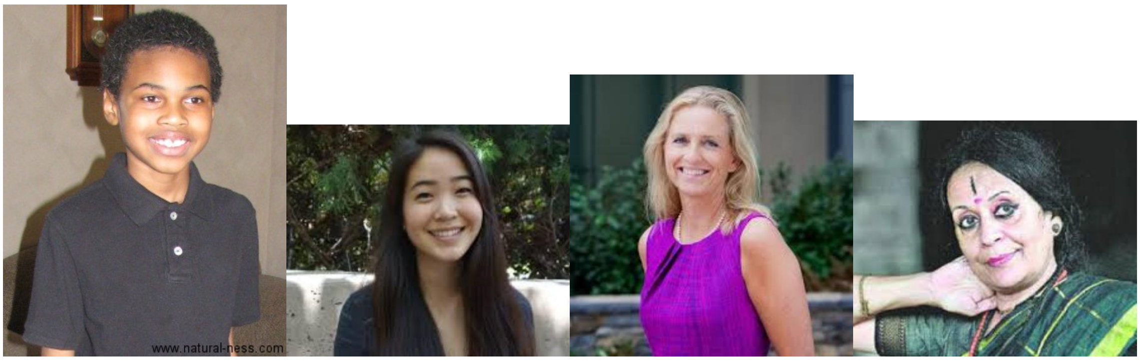


Figure 5.1: Images used for tests.

For each image, we varied the main algorithm's parameters:

- hSize - the height of the tile box
- wSize - the width of the tile box
- snakesize - the number of extra lines around the face
- space - space between tiles

The tile box is a box that includes the painted tile itself but also the spacing around other tiles. So the final tile size is equal to the size of the box minus the space between tiles.

We chose 6 combinations of parameters that allows us to assess a small degree of apriori sensible variations of these values.

The results, including the specific parameter's values, are presented in Table 5.1 and Figure 5.2.

From the tests, we can conclude that, as expected, the time of execution of the algorithm depends mostly on the number of tiles generated. The cutting of the tiles is the step of the algorithm that consumes the most time.

The number of generated tiles depends mostly on the size of the tile box: smaller tiles result in a need for more tiles to cover the same area. A slightly unexpected result is that the rectangular tile (Test 6) seems a lot less efficient than the square ones: to cover roughly the same area as in Tests 1 through 4, Test 6 required around more 3000 tiles than would be required in a perfect situation. This is explained by the fact that when tiles curved surfaces there is a lot of overlap, which then needs to be trimmed, resulting in extra processing time.

Visually, these results also allow us to understand what values work and what don't. Larger tiles (10x10) (Figure 5.2e) result in images without enough detail on the face. In Figure 5.2b there is too much space between the tiles, which gives a faded look to the mosaic, meaning that a spacing of 2 is too much (relative to the box size). The mosaics in Figures 5.2d and 5.2f represent a high level of detail but they took much longer to execute than the other ones. Since the algorithm is meant to be used in an interactive kiosk, a more reasonable value for

Table 5.1: Average time results

Parameter	Test1	Test2	Test3	Test4	Test5	Test6
hSize	5	5	5	3	10	3
wSize	5	5	5	3	10	5
snakeSize	0	3	3	8	3	3
space	1	2	1	1	1	1
# generated tiles	3 433	3 434	3 434	9 413	954	8 725
Time spent in						
guidelines,crop, facedetection (s)	6,35	7,28	5,89	10,74	13,32	7,68
matrices (s)	7,13	8,48	8,54	8,73	8,45	8,49
generate tiles (ms)	60,90	64,60	63,05	143,95	26,55	129,00
trim tiles (s)	9,38	9,77	9,94	40,61	1,69	53,44
color tiles (ms)	216,93	213,05	215,85	530,52	67,12	512,43
Total (s)	23,13	25,82	24,65	60,76	23,56	70,25

the maximum execution time should be less than 30 seconds. Therefore, tests of Figures 5.2a, 5.2b and 5.2c seem the more reasonable ones.

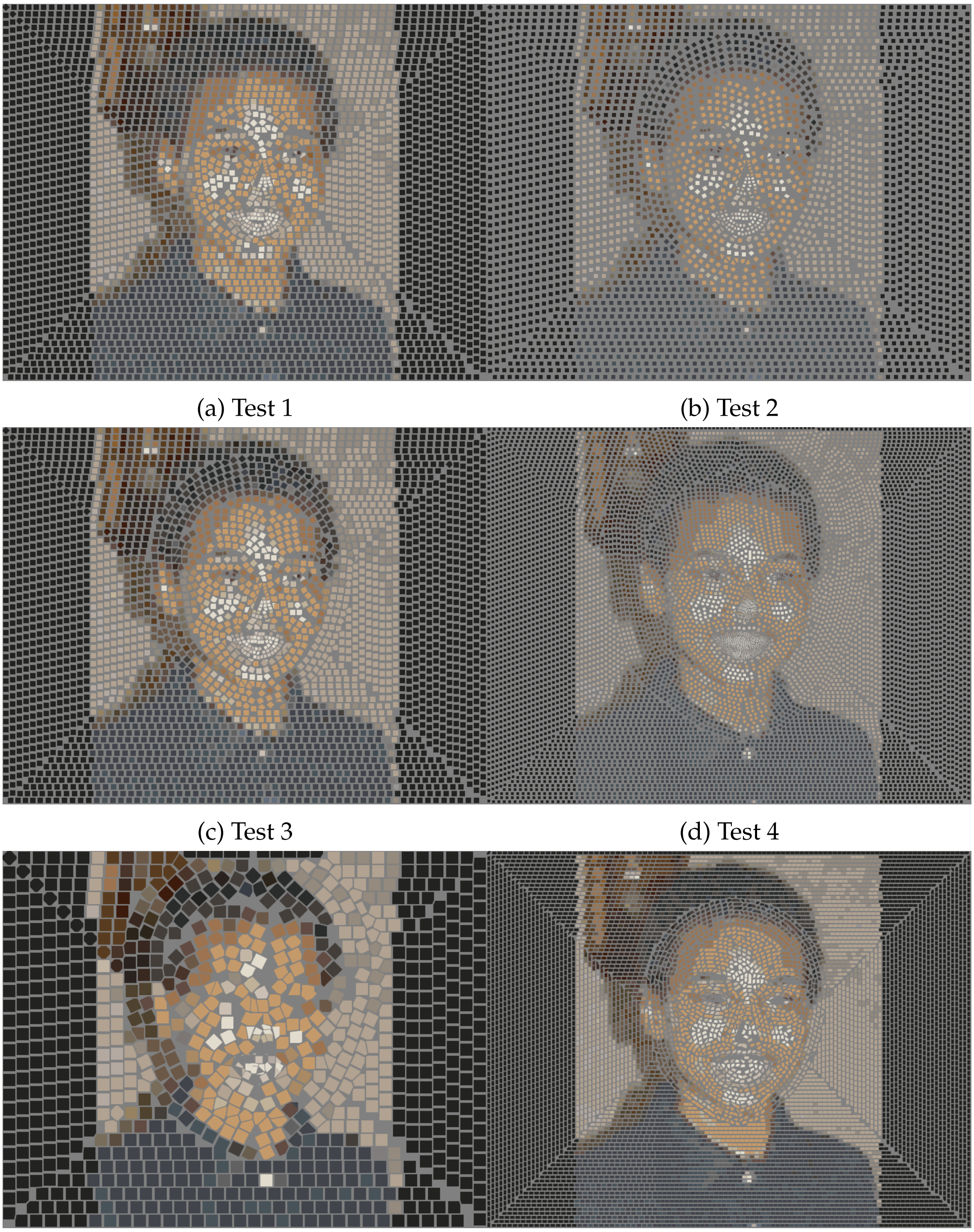
An interesting result is also the difficulty in setting an appropriate value for the snakeSize parameter due to the variations in the hair height. The hair is not a feature detected by the face detection and given the variations in hair styles, it is difficult to find an heuristic. A possibility is for this to be adjusted interactively by the user somehow.

5.2 Visual Evaluation

After evaluating the algorithm, further evaluation of the visual aspect of the mosaics that were generated was made. In this evaluation, a form was distributed so that more people could answer. This was important because the quality of the visual result is subjective and therefore, we need to collect the opinions of different people.

The goal of this evaluation is to determine which parameters produce the most visually pleasing mosaics, therefore, after the data analysis, the objective is to extract the best parameter combinations to generate mosaics with the algorithm implemented.

To collect data about mosaic preferences, the same photos from the previous evaluation phase were used, that is, the images shown in figure Figure 5.1 were used.



(a) Test 1 (b) Test 2

(c) Test 3 (d) Test 4

(e) Test 5 (f) Test 6

Figure 5.2: Visual test results

The different mosaics were obtained by changing the algorithm's parameters: hSize, wSize, snakesize, and space. For each of the images, 6 mosaics were generated using the parameters shown in Table 5.2. An example of the visual aspect of each mosaic can be seen in Figure 5.2.

The generated mosaics were then placed in a form where participants would have

	hSize	wSize	snakesize	space
Mosaic 1	3	5	3	1
Mosaic 2	3	3	8	1
Mosaic 3	5	5	0	1
Mosaic 4	5	5	3	1
Mosaic 5	5	5	3	2
Mosaic 6	10	10	3	1

Table 5.2: Parameters used to generate each mosaic variation

to rank their three favorite mosaics based on their personal tastes. The form also collected some demographic data such as age and gender. Information was also collected on whether the participant had already worked with mosaic generation.

The form is then structured by asking for the person's demographic data and experience in generating mosaics. The response to age is done in intervals to facilitate data analysis as shown in figure Figure 5.3.

Then the mosaics were presented in four sets, each one relative to an image. Each set had six mosaics for the participant to choose the best three. The mosaics were firstly shown with more detail on a bigger scale and then a question to order the best three mosaics was presented. An example of the question in the form is shown in figure Figure 5.4. The complete form used to collect data is available in Appendix A.

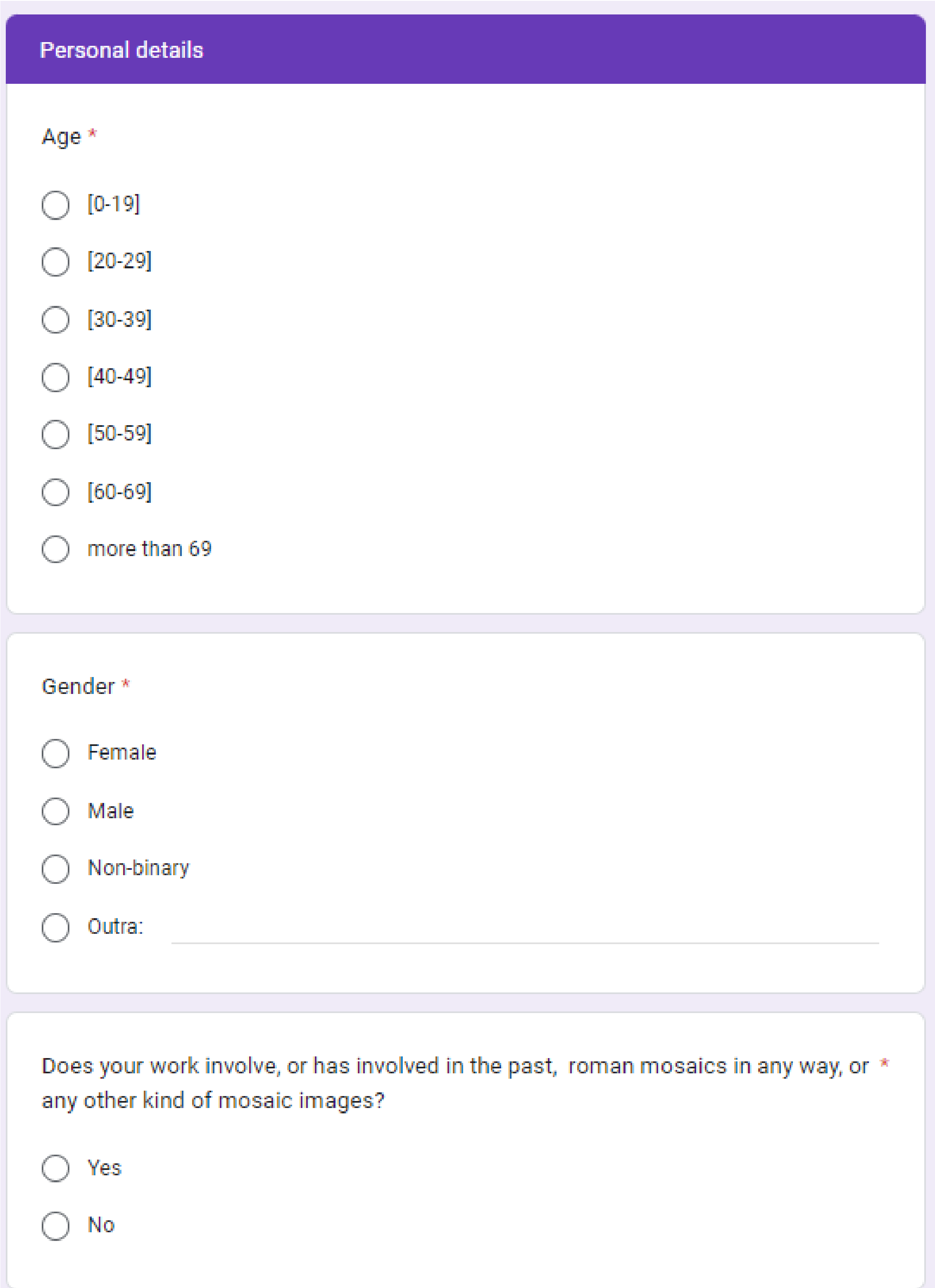
About 50 responses were obtained on this form. Visualizing the graphs Figure 5.5, it is possible to observe that there was a great variation in terms of age, with participants of almost all age ranges. This is very positive as the kiosk is intended to be used by people of all ages.

As for gender, the sample is quite balanced between females and males. As for experience in matters related to Roman mosaics, in general, the participants did not have knowledge about the subject, therefore, the answers obtained were given by people with basic knowledge about the subject, that is, without specific knowledge about mosaics.

The answers to each question are represented in Figure 5.6. For each set of mosaics, three graphs were generated. They represent the first, second, and third choices of mosaics of the participants. Each set of mosaics represented a different ethnicity and age group. The answers were similar between sets of mosaics, that is, the ethnicity and age of the person represented in the image did not influence the quality of the mosaics generated.

Looking at the generated graphics, it is clear that mosaic 1 was chosen as the most pleasing of all sets of mosaics. The rest of the choices are difficult to analyze, that is, it is difficult to know which tiles have the most votes in general.

The information present on these graphs was combined into one graph for better evaluation of results, that is, the responses of each set were added. The first choices were given a weight of 3, the second choices had a weight of 2, and the third choices had a weight of 1. The values obtained can be seen in Figure 5.7.



Personal details

Age *

[0-19]

[20-29]

[30-39]

[40-49]

[50-59]

[60-69]

more than 69

Gender *

Female

Male

Non-binary

Outra:

Does your work involve, or has involved in the past, roman mosaics in any way, or any other kind of mosaic images? *

Yes

No

Figure 5.3: Collecting demographic data on the form

In this graph, the mosaics 1 and 4 were the most voted. Mosaic 2 was the least voted. The mosaics 3, 5, and 6 have a very close number of votes.

In the form, a participant commented: "In 6, I have a feeling a human artist would sacrifice fidelity to the ground truth in order to get human features across more

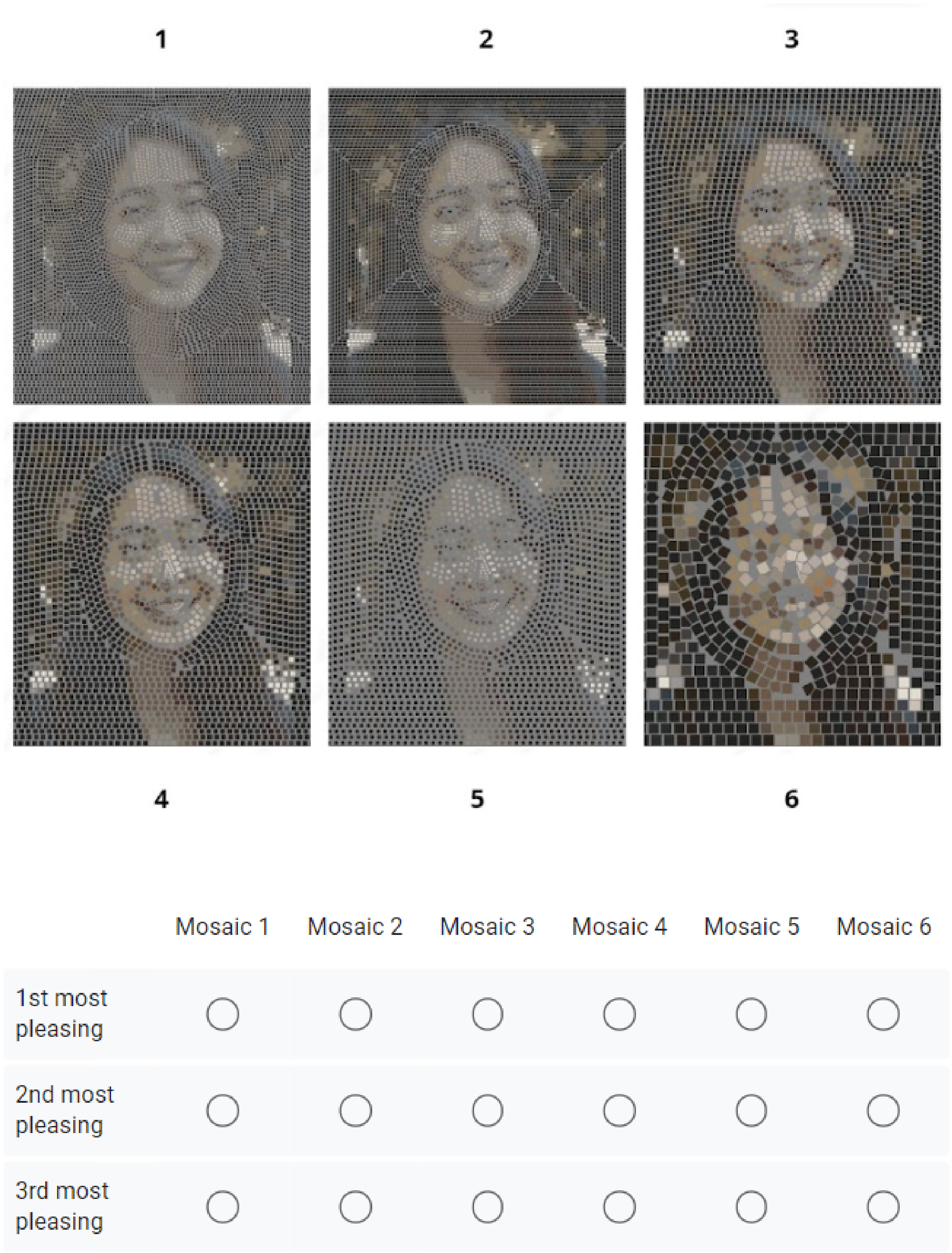
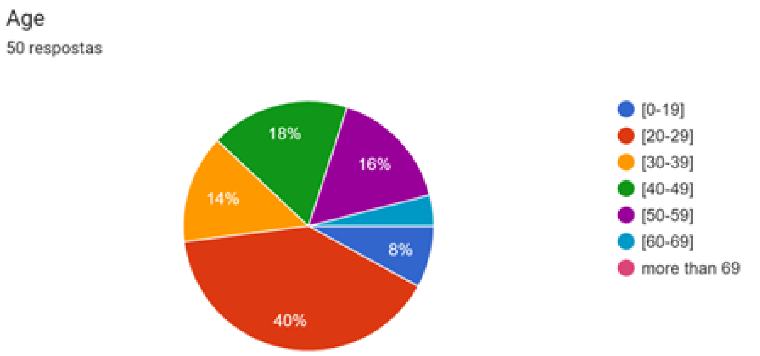
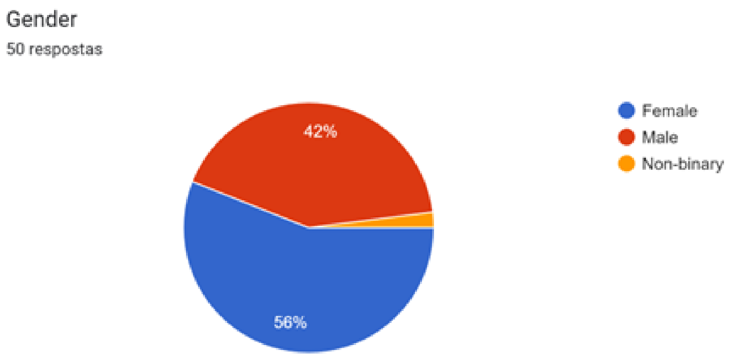


Figure 5.4: Visual Evaluation Form question example

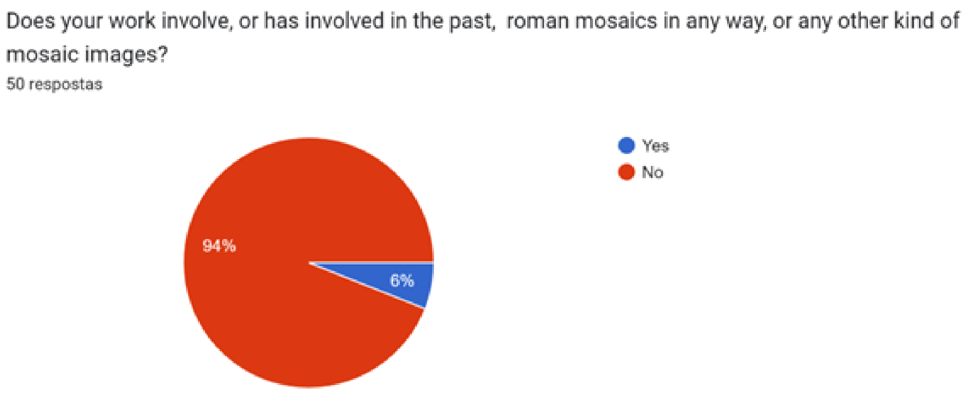
clearly.". Another participant commented: "The mosaics with more "stones" look a bit uncanny". The mosaic 6 is the one that generated the least number of stones. However, the features of the face are not clearly represented. In this case, the good representation of the human features was prioritized. Therefore, mosaic 5 was chosen over mosaic 3 and mosaic 6 because it represents the human features better than 6 and generates fewer tiles than 3. From this analysis, the mosaics chosen to be in the kiosk were 1, 4, and 5.



(a) Age of the participants

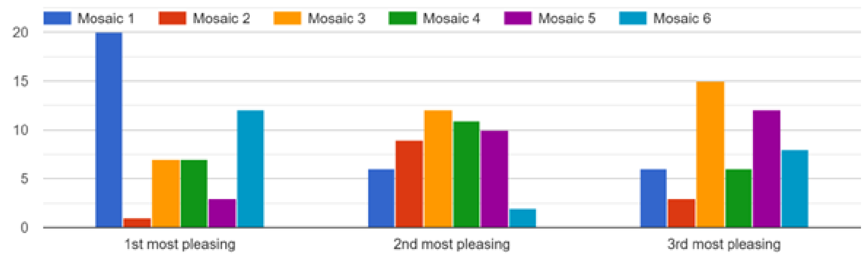


(b) Gender of participants

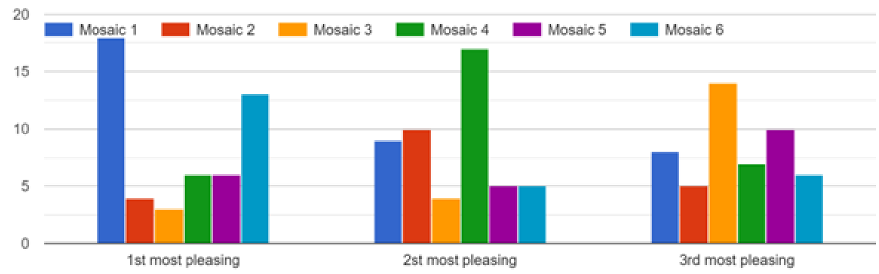


(c) Mosaic experience of participants

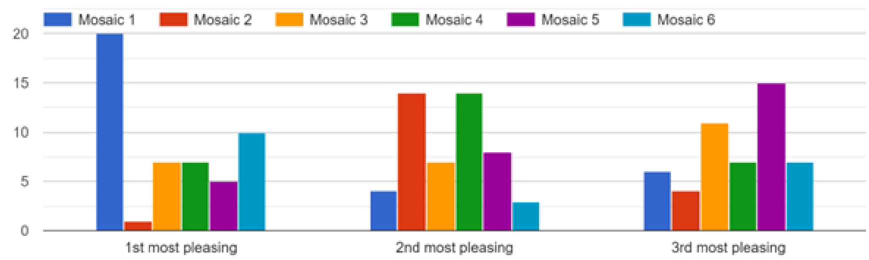
Figure 5.5: Demographic data



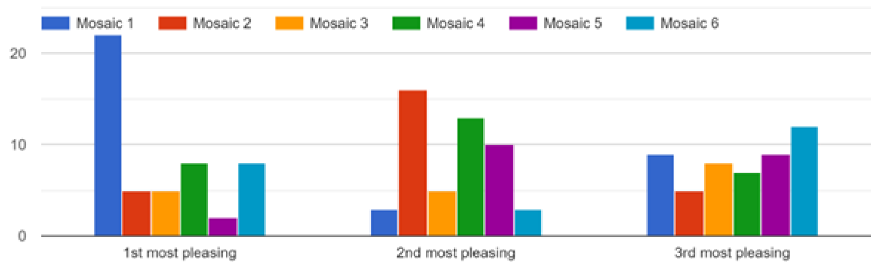
(a) First Set of mosaics



(b) Second Set of mosaics



(c) Third Set of mosaics



(d) Fourth Set of mosaics

Figure 5.6: Visual test results

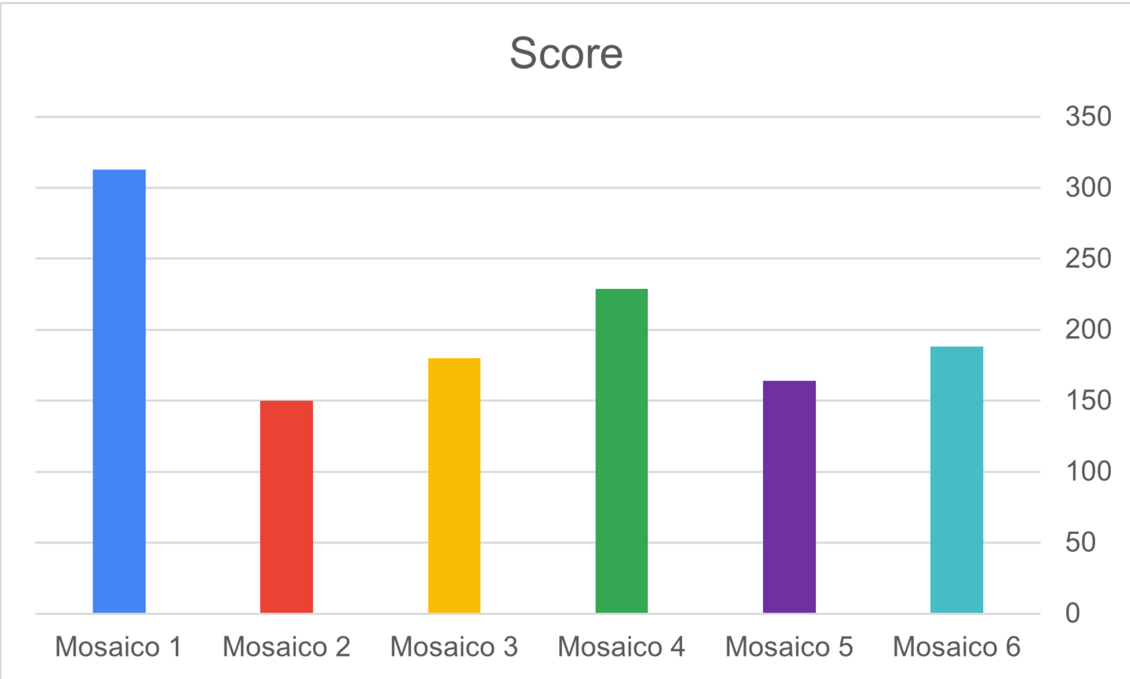


Figure 5.7: Interpretation of form data

5.3 Lab Kiosk user Evaluation

The evaluation of the kiosk was made to test the quality of the kiosk application in terms of simplicity of usage, quality of the mosaic generated, and to evaluate the overall experience.

The tests were made with a touch screen, a camera, and a computer. The users would use the touch screen to interact with the kiosk and then they would answer a form describing their experience. The experience was performed at first in the Department of Informatics Engineering of the University of Coimbra and then they were performed in the Museum of Conimbriga.

The kiosk was tested first in the Department of Informatics Engineering of the University of Coimbra. When passing the hallway, people were asked to try the kiosk. The display of the computer and the screen used in the tests is shown in Figure 5.8. The QR code on the computer was used for the testers to have access to the form on their phones so they could answer more easily.

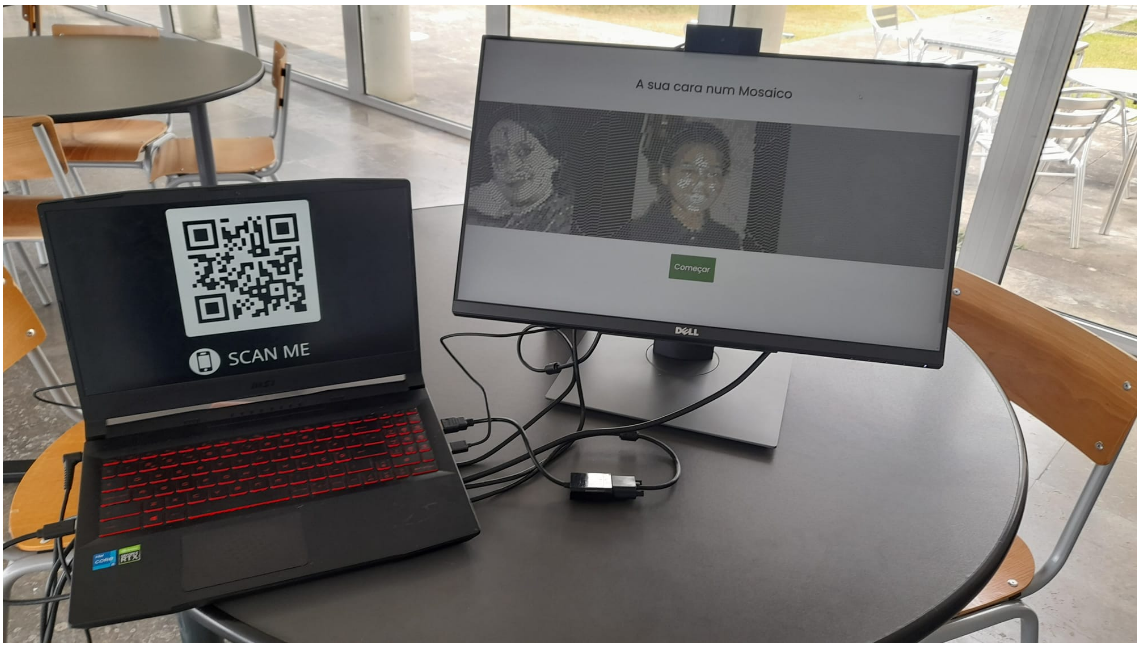
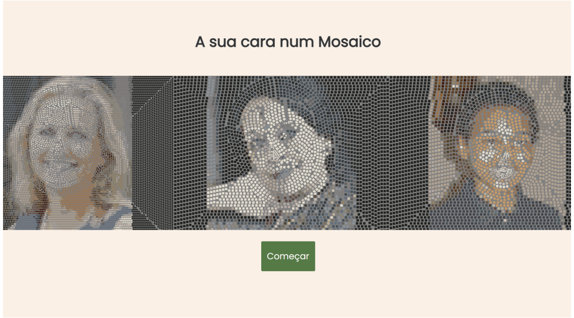


Figure 5.8: Display of kiosk

The context given to the users before trying the kiosk was minimal since we wanted to evaluate if the application was easy enough to use autonomously without further explanations. With that being said, the users were given the following information:

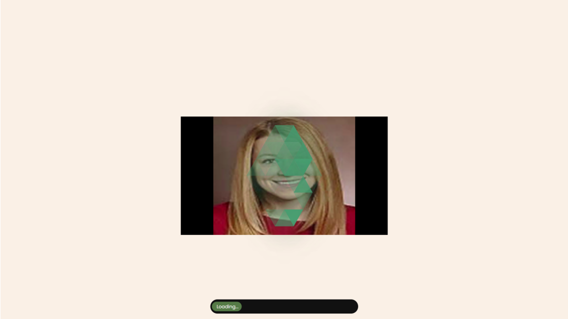
- The kiosk is running on the touchscreen
- The kiosk's purpose is to transform the face into a Mosaic
- The kiosk was thought to be used in a museum by its visitors

The interfaces of the kiosk used in these tests were Figure 5.9. They were changed later for the tests at the museum, described in the next section, due to the comments obtained in this test phase.

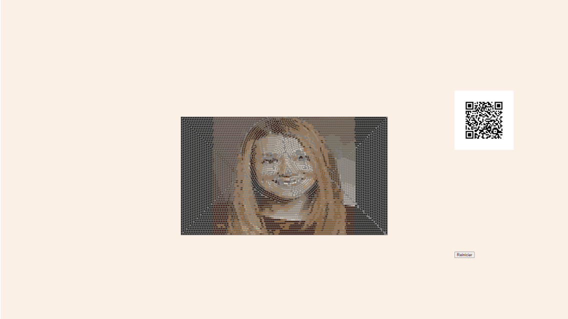


(a) Homepage

(b) Page to take picture



(c) Loading page



(d) Page with generated mosaic

Figure 5.9: First version of kiosk interface

After using the kiosk the users were asked to answer a form about the experience and they were encouraged to comment about its performance and aspects that could be improved. The form was composed of nine questions. The questions asked were multiple choice or a scale from 1 to 6 in which 1 represented "Disagree" and 6 represented "Agree". The first question asked what mosaic type was chosen. The user could choose more than one option in case he generated more than one mosaic. The second question asked if the user liked the visual aspect of the mosaics generated. The third question asked, just in case the user generated more than one mosaic, which one did they like best? It was also asked which one they liked least. Next, they were asked whether the kiosk was easy to use and whether it was appealing to attract the attention of museum visitors. A question was also asked about how long the mosaic took to be generated and whether any problems were found in the kiosk. In the end, the users could leave comments and suggestions about the kiosk and their experience. The form used can be consulted in Appendix B.

The users in this phase of testing consisted of students and teachers of the Department of Informatics Engineering. The answers obtained in the form are presented in Figure 5.12. Some users left comments about points to improve when interacting with the kiosk. One comment of the testers was that the mosaic choice should be more evident in the UI, that is, the buttons to choose the mosaic type were not clear enough. Another user said "I think the choice of mosaic should be more implied. In my experience, I did not immediately realize that there were 3 possible options to choose from.".

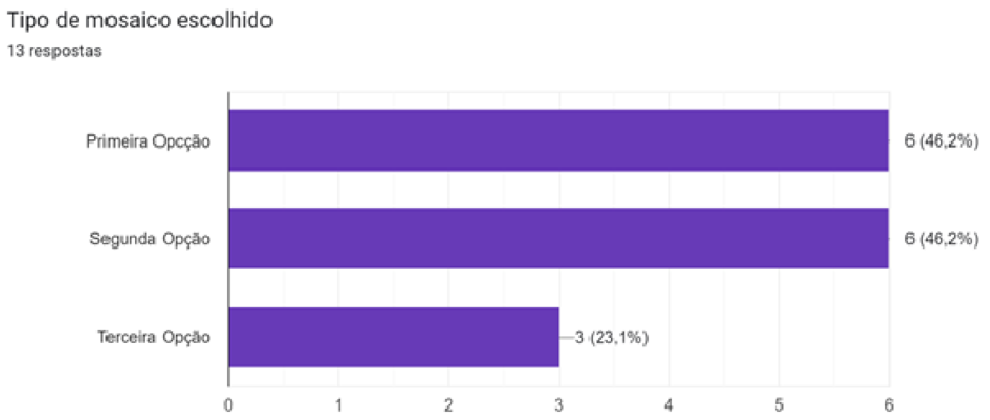
About the loading page, a user wrote "While we wait it was fun to have some curiosity to read or something like that about mosaics in order to be more fun". Regarding the last page shown to the user, a user commented that it would be better to "improve the design of the final page (restart button" and "indicate that the QR code is for downloading)". A user also said that "the camera image could be mirrored to help the user". Users also commented that the final mosaic's resolution could be more squared instead of being a rectangle.

Looking at the answers given in the form, the first and second mosaic types were chosen the same amount of times Figure 5.10a. These mosaic types were chosen more times than the third option. This result was expected since in the previous tests to define what mosaic types would be available in the kiosk, the first option was the most preferred one, followed by the second and third, therefore, the users chose to generate the mosaics that visually were more appealing to them in the examples given in the interface when choosing the mosaic type to generate.

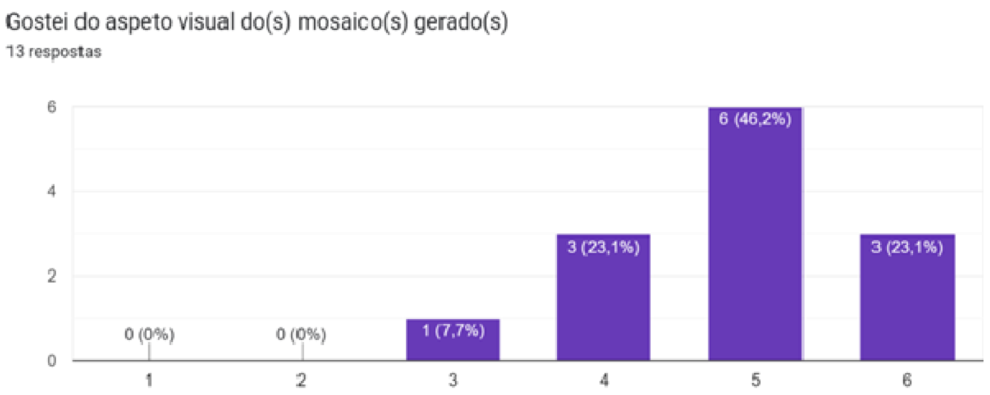
When answering about the visual aspect of the mosaics generated, the opinions of the users were divided Figure 5.10b. However, the answers were closer to the side of "Agree" which means that overall the users liked the aesthetics of the mosaics generated.

Regarding the mosaic most liked by the users Figure 5.10c, the second one was the most voted. This result was not expected since the first option was the most voted as most pleasing in the visual evaluation of the algorithm in section 5.1. However, as this question is related to the personal preferences of the users, it is normal that the preferred mosaic type varies.

In graph Figure 5.11a we can conclude that most users found the kiosk easy



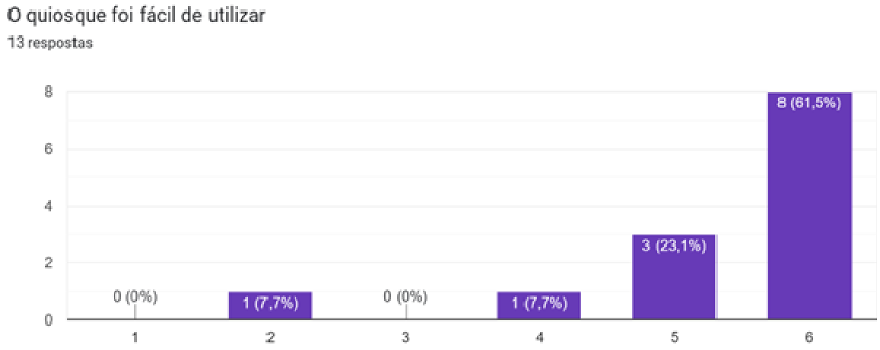
(a) Chosen Mosaic



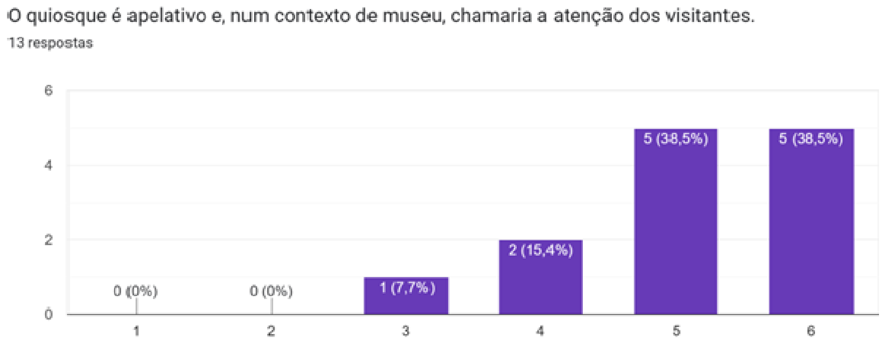
(b) Mosaic's visual aspect

(c) Most liked mosaic

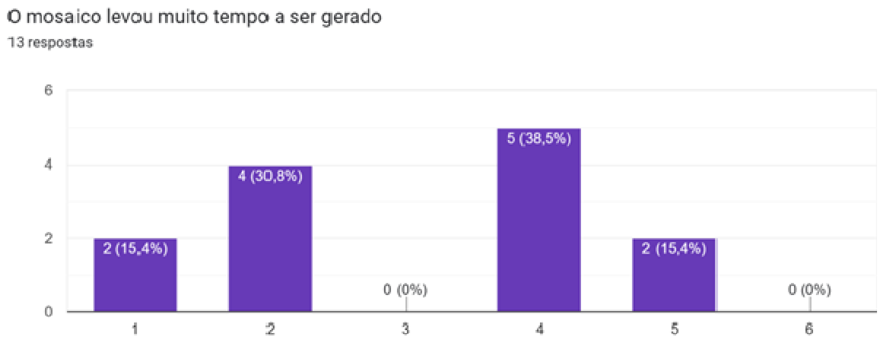
Figure 5.10: User tests answers



(a) Easy to use



(b) Kiosk Appealing to users

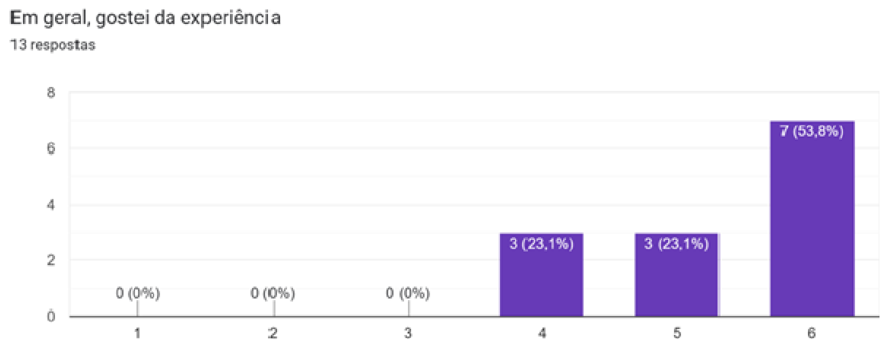


(c) Execution Time

Figure 5.11: User tests answers

to use but some users were confused with some functionalities, as some commented that they did not understand that they could choose the type of mosaic they wanted to generate as they did not notice where to choose them. Therefore, this aspect of the UI was changed in order to provide a better user experience.

According to users, the kiosk could be more appealing to attract more attention



(a) Rating overall Experience

Figure 5.12: User tests answers

from people who pass by it, and the UI is quite simple Figure 5.11b. While users generally liked the kiosk's interface, many commented that it could be improved.

The most dividing answer of the form was about the time of execution as ii is shown in Figure 5.11c. Some users considered that the waiting time for the mosaic to be generated was long and some considered it adequate. The different answers occur since the time of execution can vary due to the type of mosaic chosen.

In general, the users enjoyed the overall experience which is good feedback. Considering the tester's comments and answers to the form, some changes were made to the UI in order to provide a better user experience. The changes made in the UI are described in the next section.

5.4 Kiosk Evaluation at the Museum

These tests were made in the Museum of Conimbriga in Coimbra with the goal of testing the kiosk in a real environment, that is, with museum visitors. To perform the tests at the Museum, some changes to the UI were made, regarding the previous tester's comments. The interfaces used in these tests were shown in section 4.3. In the final version of the interfaces, the option of changing the language of the kiosk was added. The choice of the mosaic type was changed to another page so that this step was more clear to the user. The resolution of the camera was also changed in order to be more squared and a text indicating the functionality of the QR code was added next to it.

The kiosk was situated at the end of the visit so, they had the opportunity to interact with the kiosk after seeing the exhibition and the ruins at the museum. The installation of the kiosk is shown in Figure 5.13 and it was available for one day at the museum.

This phase of testing consisted of two different approaches. In one approach the

Figure 5.13: Display of kiosk at the Museum

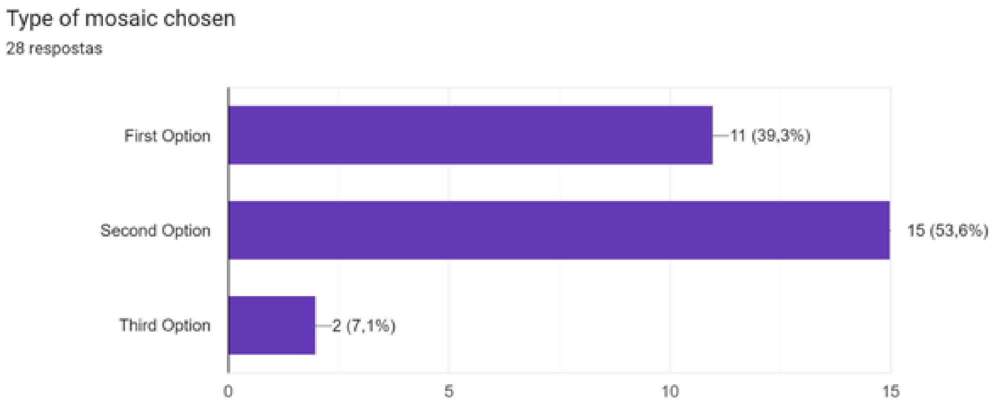
users were invited to try the kiosk and then relate their experience in a form Appendix C or comment orally on their opinion of the kiosk. The form used was similar to the one used in the laboratory tests, however, an English version was made. When approaching the visitors the context given was:

- This was a project made for a master's thesis at the University of Coimbra
- The project consists of a kiosk that transforms the face into a mosaic
- In the end there is a form about the experience that you can fill out if you would like

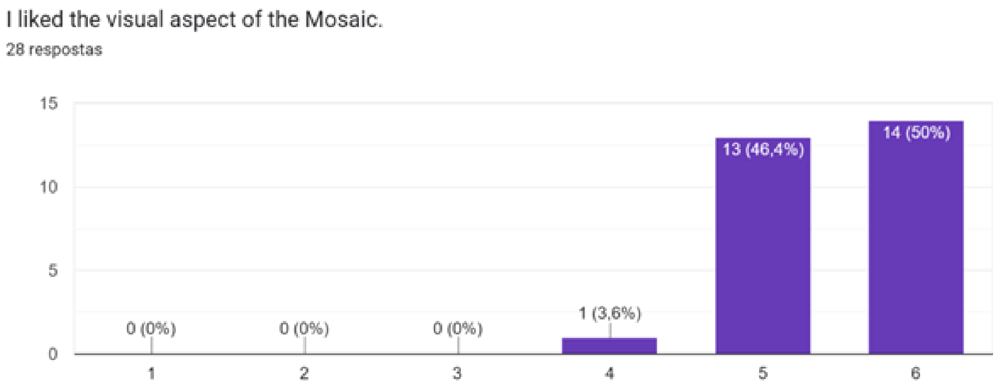
On another approach, the behavior of the visitors was observed from a distance, that is, as they passed through the kiosk we observed if they would interact with it voluntarily or if they were interested in it. In this case, after they generated their mosaics, they were asked to comment on their experience and on the kiosk's positive and negative aspects. The users were also encouraged to answer the form if they would like.

In this phase of tests, there were 28 answers to the form. The results obtained in the form are presented in Figure 5.15. The questions asked were similar to the form used in the previous evaluation, that is, some were multiple choice and others were a scale from 1 to 6 in which 1 represented "Disagree" and 6 represented "Agree". Some of the mosaics generated at the museum by the visitors are shown in Figure 5.16.

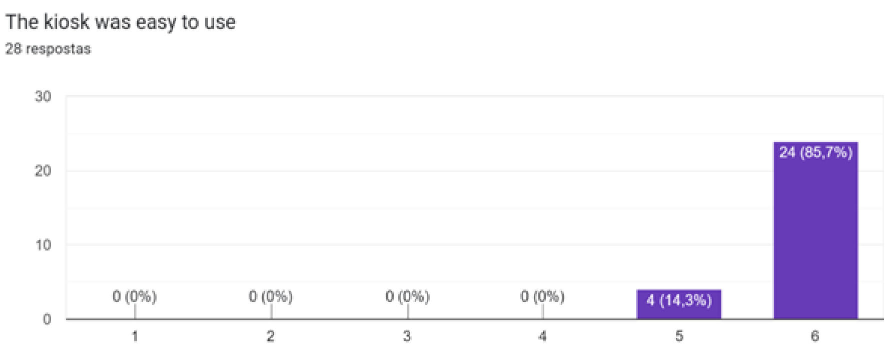
Regarding the generated mosaic some visitors commented about the lack of colour in the mosaic. A visitor referenced that the internet problems spoiled the experience since the QR code did not work to save the mosaic.



(a) Chosen Mosaic



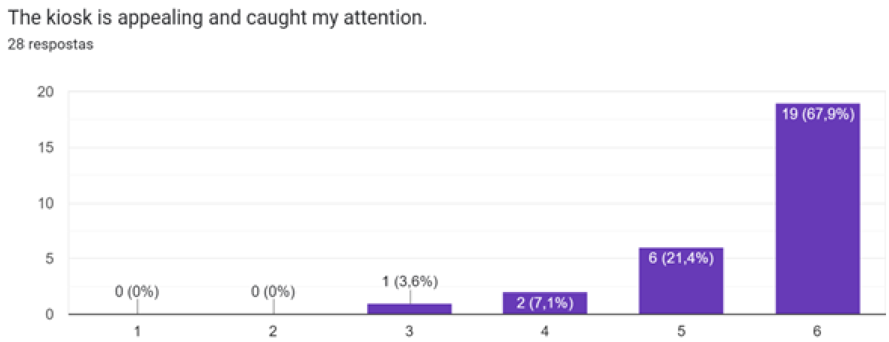
(b) Mosaic's visual aspect



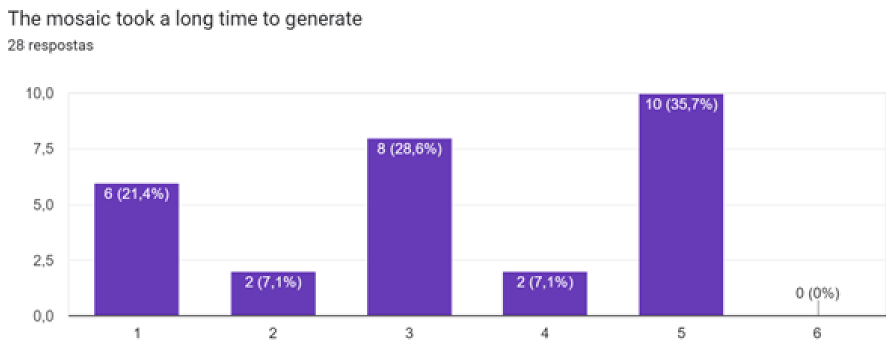
(c) Easy to use

Figure 5.14: Museum Visitor's answers to the form

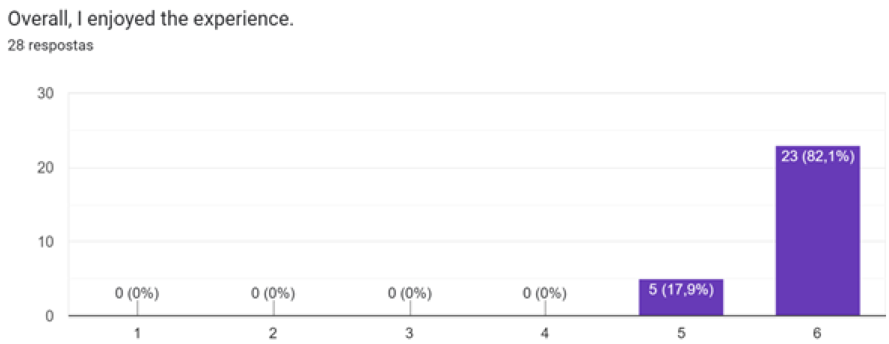
The visitors in general made positive comments about the kiosk, thinking of the concept as original and adequate for the museum. They thought it was a fun



(a) Kiosk Appealing to users



(b) Execution Time



(c) Rating overall Experience

Figure 5.15: Museum Visitor's answers to the form

experience but pointed out some aspects that could be better such as the time for the mosaic to be generated and the failures of the internet connection.

Without approaching the people directly, there were about three visitors that interacted with the kiosk but most of them did not notice it was in there. This

Figure 5.16: Mosaics Generated at the museum

indicates that the kiosk was not as appealing as needed to attract more of the visitor's attention. In figure Figure 5.11b it is noticeable that the visitors did not find the kiosk very appealing. That could have happened due to the lack of indication of the presence of the kiosk, that is, with no further indications that the kiosk was there and that it was to be used by the visitors, most of them did not realize its presence. On the other hand, when asked to try the kiosk, the visitors were interested, both children and adults. Every visitor that used the kiosk saved the generated mosaics using the QR code.

The mosaic that the visitors chose the most was the second option as illustrated in Figure 5.14a. This mosaic option was also the most chosen in the tests made in the Department of Informatics Engineering, therefore we can assume that it is the mosaic type that the public in general prefers.

In general, the visitors liked the visual aspect of the mosaic generated and gave positive feedback on this aspect of the experience as seen in Figure 5.14b. As for ease of use, compared to previous tests, the feedback improved, and the responses were more positive, as can be seen in Figure 5.14c. This result means that the changes made to the interface have improved the user's interaction with the kiosk, namely the addition of a screen dedicated exclusively to choosing the type of mosaic desired. This new screen made the possibility of choosing between types of mosaics more evident, and this functionality did not go unnoticed. In addition, the images of the types of mosaics to be chosen could be placed in a larger size, since the screen was only showing this information, which facilitated the visualization of the differences between the types of mosaics.

Due to the place where the museum is located, the internet has some failures which made the mosaic take more time to be presented to the visitor in some cases since the generated mosaic is saved in a PHP server before a QR code is created and then shown to the user. Therefore, when answering about the time that the mosaic took to be generated, the opinions were very divided and negative in general as it is shown in Figure 5.15b. Another factor that interferes with the time of execution is the mosaic option chosen.

In conclusion, the visitors answered that the kiosk was easy to use and there was no confusion regarding the choice of the mosaic type. There could be a description for each choice describing the differences between the choices since some visitors reported that although they could see that the choices were different, they would prefer to have some information about the mosaic they were choosing. Despite the less positive notes about the kiosk, the visitors enjoyed the experience as it is shown in Figure 5.15c.

Chapter 6

Conclusion

Within the scope of this master's thesis, an algorithm was developed to generate mosaics from photographs and integrate it into a kiosk interface. In this document, an extensive search of algorithms that transform images into mosaics is presented. The research was focused on algorithms that produce mosaics with characteristics similar to Roman mosaics. After the research, the Battiato algorithm was chosen to be used in the kiosk. Research was also carried out on kiosks existing in museums today, focusing on their functionality and mode of operation. The implementation of the kiosk application took place in two phases: First, the algorithm that generates the mosaics was implemented, and then the kiosk application was implemented, where the algorithm was subsequently integrated. The algorithm was subject to some adaptations to make it more suitable for the specific project, namely using face detection to obtain the characteristics of the user's face.

After the implementation of the algorithm was finished, an evaluation was made of the algorithm in terms of performance, that is, algorithm execution time, and in visual terms, that is, the aesthetic quality of the generated mosaics. The visual evaluation of the generated mosaics was done through the distribution of a form that was used to identify the best parameterizations of the algorithm. After carrying out this evaluation, evaluations were made of the kiosk. For this purpose, the kiosk was put into operation using a laptop connected to a touch screen. First, the evaluation of the kiosk took place at the Department of Informatics Engineering at the University of Coimbra with students and professors from that department. Users were invited to fill in a form about their experience.

By analyzing the feedback received, some changes were made to the kiosk interface. Afterward, an evaluation of the kiosk at the Monographic Museum of Conímbriga was carried out, where visitors to the museum tried the kiosk and also answered a form to facilitate the reporting of their experience.

The results obtained were satisfactory and the feedback regarding the visual appearance of the mosaics created was positive in all phases of evaluation, both in tests in the laboratory and in tests carried out in the museum. Users in general, although some pointed out aspects for improvement, such as the mosaic creation time and the interface being more elaborate, made positive comments about the kiosk, saying that it is a fun, original concept and quite appropriate to the theme of the Museum in which it was placed.

As future work, improvement of the UI can be considered so that the kiosk is more appealing and catches the attention of the visitors of the museum. The physical structure of the kiosk could also be done with a touch screen in a structure that allows the user to control better the height of the screen so that the kiosk can be easily used by every person of any age group independently of their height. Another aspect that could be improved in the future is the performance of the algorithm, that is, making it more efficient so that the user does not have to wait so long for the mosaic to be generated.

References

[1] . Descobrir - património romano | villasicó. http://www.villasico.com/descobrir/patrimonio-romano/pag2, June 2012.

[2] DGPC. Dgpc | museus e monumentos | rede portuguesa de museus | museu da villa romana do rabaçal.

[3] S. Battiato, G. Di Blasi, G. M. Farinella, and G. Gallo. Digital mosaic frameworks - an overview. In *COMPUTER GRAPHICS forum Volume 26, number 4 pp. 794–812*, 2007.

[4] History of stained glass | the stained glass association of america.

[5] Paul Haeberli. Paint by numbers: Abstract image representation. In *SIGGRAPH1990*, 1990.

[6] Wojciech Pokojski and Paulina Pokojska. Voronoi diagrams – inventor, method, applications. In *Polish Cartographical Review Vol. 50, 2018, no. 3, pp. 141–150*, 2018.

[7] J. Dobashi, HAGA T., JOHAN H., and NISHITA T. A method for creating mosaic images using voronoi diagrams. In *Eurographics2002*, 2002.

[8] Geisa Faustino and Luiz Figueiredo. Simple adaptive mosaic effects. In *SIBGRAPI2005*, 2005.

[9] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41:637–676, 1999.

[10] David Mould. A stained glass image filter. In *Eurographics Symposium on Rendering 2003 Per Christensen and Daniel Cohen-Or (Editors)*, 2003.

[11] C.M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *2002 International Conference on Pattern Recognition*, volume 4, pages 150–155 vol.4, 2002.

[12] Paul E. Black. Manhattan distance. Available from: https://www.nist.gov/dads/HTML/manhattanDistance.html, 2019.

[13] Alejo Hausner. Simulating decorative mosaics. In *Proc. SIGGRAPH2001, 573–580.*, 2001.

[14] FRITZSCHE L. P., HELLWIG H., HILLER S., and DEUSSEN O. Interactive design of authentic looking mosaics using voronoi structures. In *Proc. 2nd International Symposium on Voronoi Diagrams in Science and Engineering VD 2005 Conference*, 2005.

[15] S. Battiato, DI BLASI G., FARINELLA G. M., and GALLO G. Sa novel technique for opus vermiculatum mosaic rendering. In *Proc. ACM/WSCG*, 2006.

[16] Stefan Schlechtweg, Tobias Germer, and Thomas Strothotte. Renderbots – multi-agent systems for direct image generation. In *Computer Graphics Forum 24, 2 , 137–148*, 2005.

[17] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes, 1990.

[18] Getting it across layout issues for kiosk systems.

[19] FLORIN-CĂTĂLIN TOFAN, ANCA TUDOR-ANDREI, and DANIEL-ŞTEFAN GHIURCA. Interactive applications in museum activities. 2018.

[20] Sfmoma unveils groundbreaking educational facility new center unique among american art museums.

[21] J. Boys and A. Boddington. *Museums and Higher Education Working Together: Challenges and Opportunities*. Taylor & Francis, 2016.

[22] Riccardo Bianchini. Everyone can make art at the tate modern! | inexhibit, 2019.

[23] Tate modern museum digital experience | segd.

[24] . Heineken experience. https://www.heinekenexperience.com/pt/, August 2023.

[25] . Heineken experience, amsterdam. https://www.amsterdamsights.com/attractions/heineken. August 2007.

[26] . Get your guide - amsterdam: Heineken experience ticket. https://www.getyourguide.com/amsterdam-l36/amsterdam-heineken-experience-ticket-t205012/, August 2008.

[27] This person does not exist.

[28] Vincent Mühler. Face-api.js. https://justadudewhohacks.github.io/face-api.js/docs/index.html, June 2023.

[29] Fabrizio Bianchi. Extract palette from image - Coolors. https://coolors.co/image-picker, June 2023.

[30] Yang Song and Zhifei Zhang. UTKFace, June 2023.

Appendices

Appendix A

Apendix A - Form used for visual evaluation of the mosaics

Ranking image mosaics

The project consists of creating a fun interactive kiosk to encourage the study of the Roman mosaic history and provide interest in extending knowledge about the Roman heritage to people of all generations, attracting more visitors to museums and provide a better museum experience.

The kiosk application uses an algorithm to create a personalized mosaic, using a picture of the user taken from a web camera.

What is the purpose of this questionnaire?
The purpose of this questionnaire is to evaluate the mosaic images obtained with the algorithm implemented, comparing mosaics generated with different parameters.
Four images of different people from different ethnicities and ages were used. For each image, six different mosaics were generated by altering a set of parameters in the mosaic generation algorithm. The goal of this form is to determine the best combination of parameters for the mosaic generation algorithm based on your choices.

Who can participate?
To participate in this study, the following criteria must be met:
- Be of any nationality but have written understanding of English.

Do I have o participate in this questionnaire?
Your participation is voluntary and you may refuse to participate. Your responses and data will be strictly confidential and anonymous.
Your participation in this study is voluntary and you may refuse to participate. If you decide to participate, you will have the right to access, change, delete, oppose, and limit the processing of your personal data or to withdraw from the study at any time, without any reprisals and/or need to justify the reason. To exercise these rights, you should contact the researchers responsible for this study, to the following email address: beatriz.a.a.madeira@gmail.com.

How will the results of the study be disseminated?
The results of this study will be disseminated through its publication in scientific journals and presentations in scientific meetings (national and international), as, for example, conferences. The main results will be shared in the media (e.g., UC press releases), as well as in study dissemination web pages.

To whom should I address questions about the study?
Beatriz Abrantes Abreu Madeira (beatriz.a.a.madeira@gmail.com), Master's student at the Department of Informatics Engineering (DEI) of the Faculty of Science and Technology of the University of Coimbra (FCTUC), supervised by Professor Jorge Cardoso (jorgecardoso@dei.uc.pt) DEI/FCTUC.

* Indica uma pergunta obrigatória

1. Informed, Free and Clarified Consent *

Marcar tudo o que for aplicável.

- [ ] I declare that I have read and understood all the information presented and I agree to respond to the questionnaire.

Personal details

2. Age *

Marcar apenas uma oval.

- [0-19]
- [20-29]
- [30-39]
- [40-49]
- [50-59]
- [60-69]
- more than 69

3. Gender *

Marcar apenas uma oval.

- Female
- Male
- Non-binary
- Outra: ____________________

4. Does your work involve, or has involved in the past, roman mosaics in any way, or any other kind of mosaic images? *

Marcar apenas uma oval.

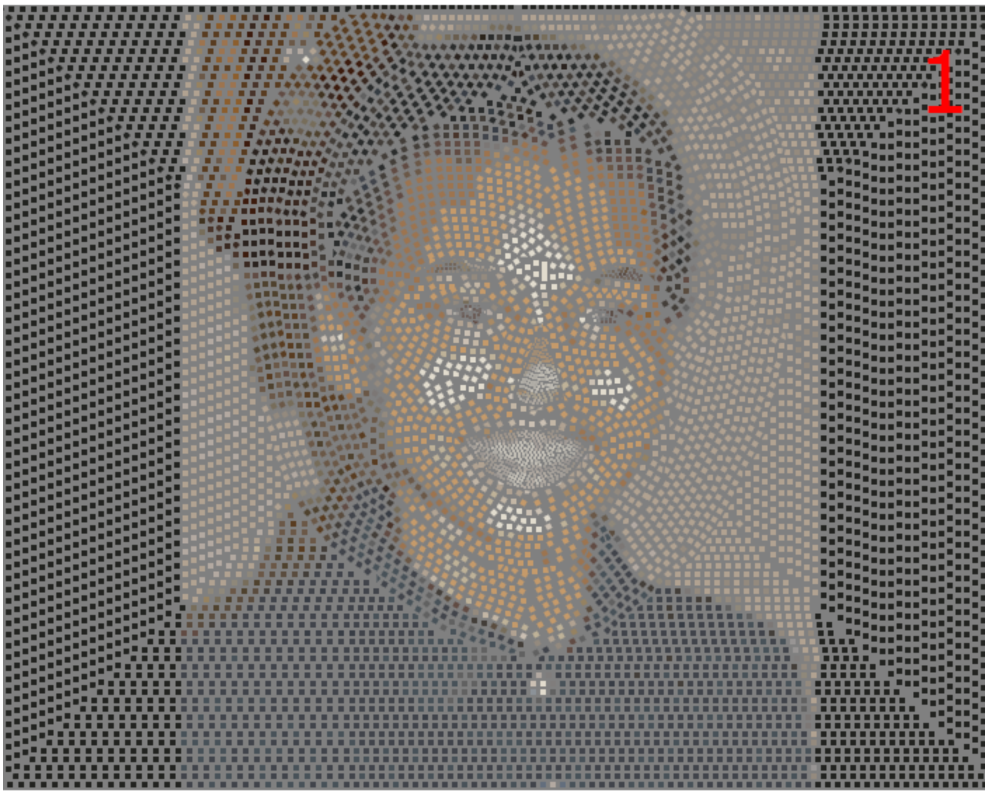
- Yes
- No

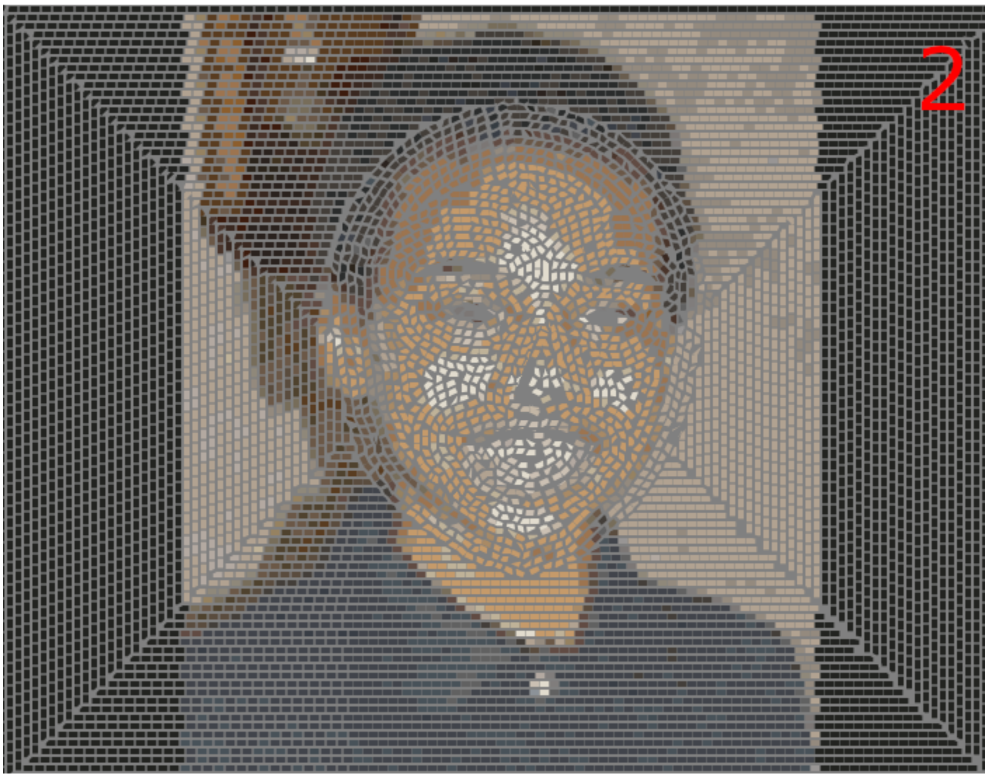
First set of mosaics

Mosaics were generated using the image below.

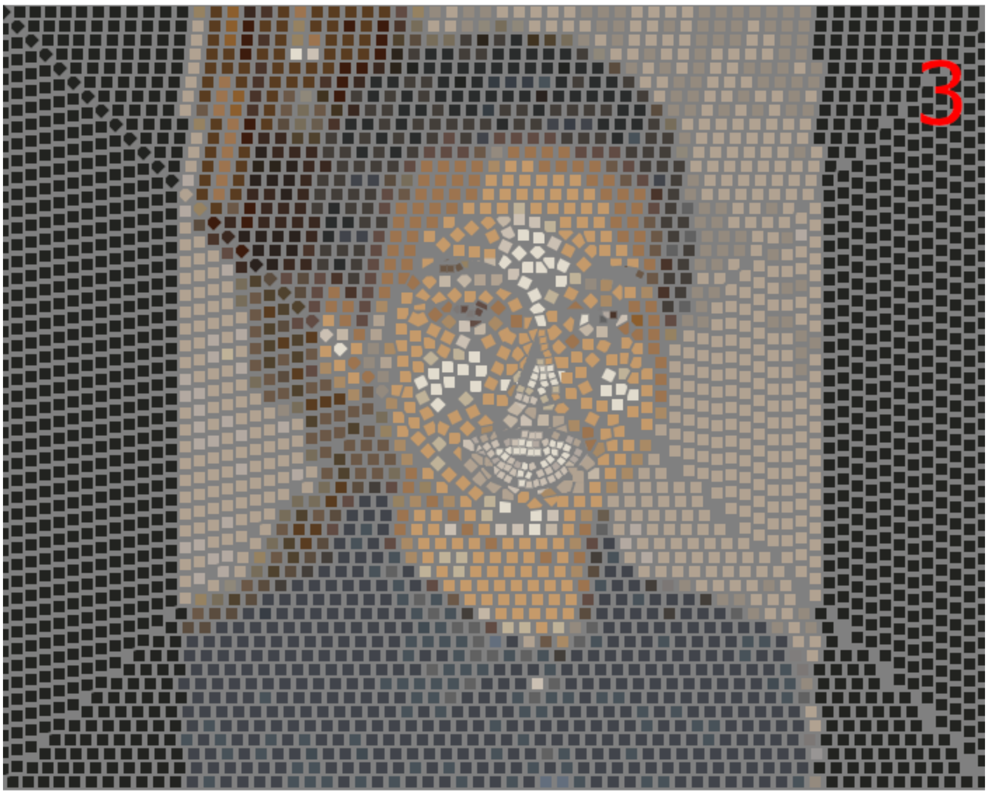
The six generated mosaics are shown below, identified with a red number.

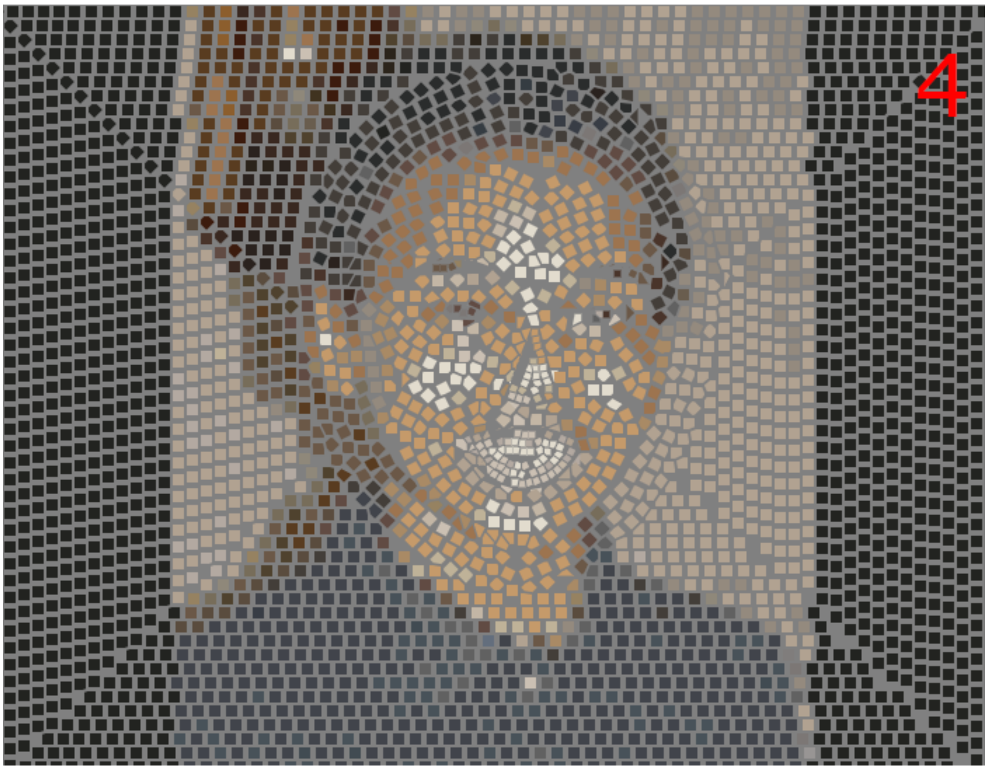
24/08/23, 15:50 Ranking image mosaics

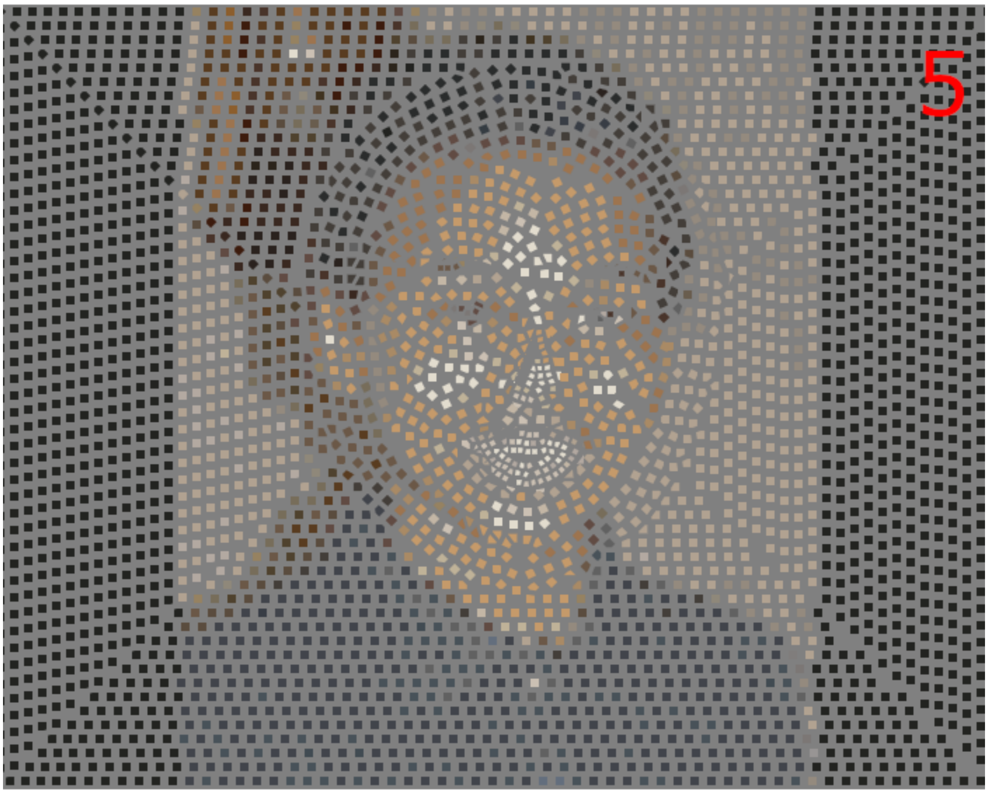


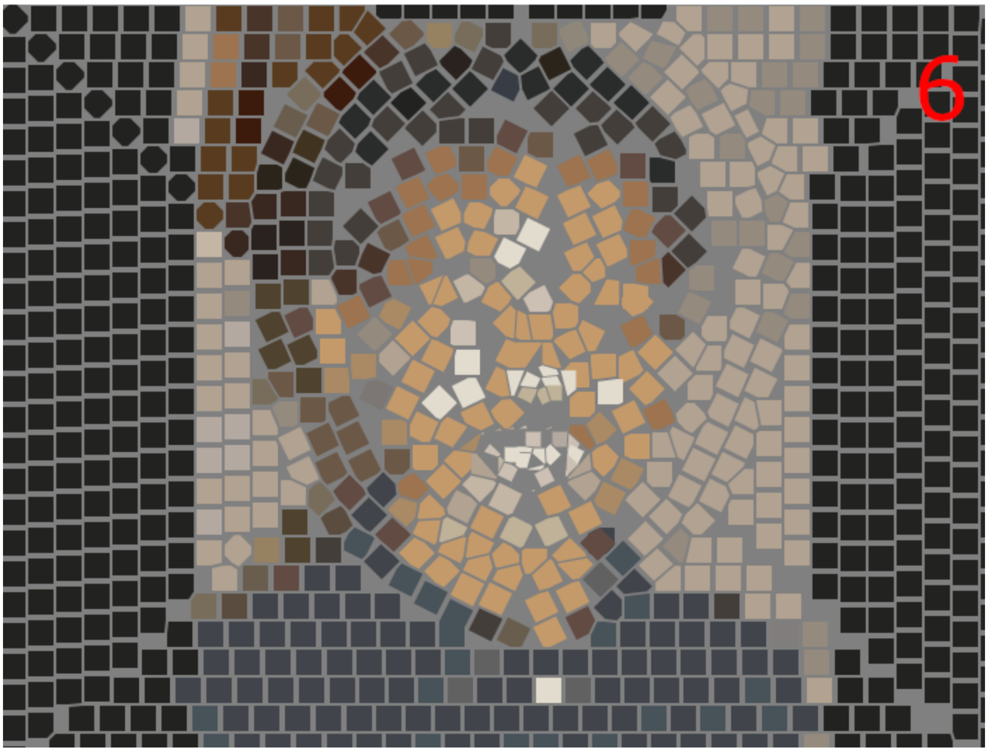


https://docs.google.com/forms/d/1LPBfK3IpEZaIz-0WsU32E1a0NY1CIWZSF4PThiWLm10/edit 5/32

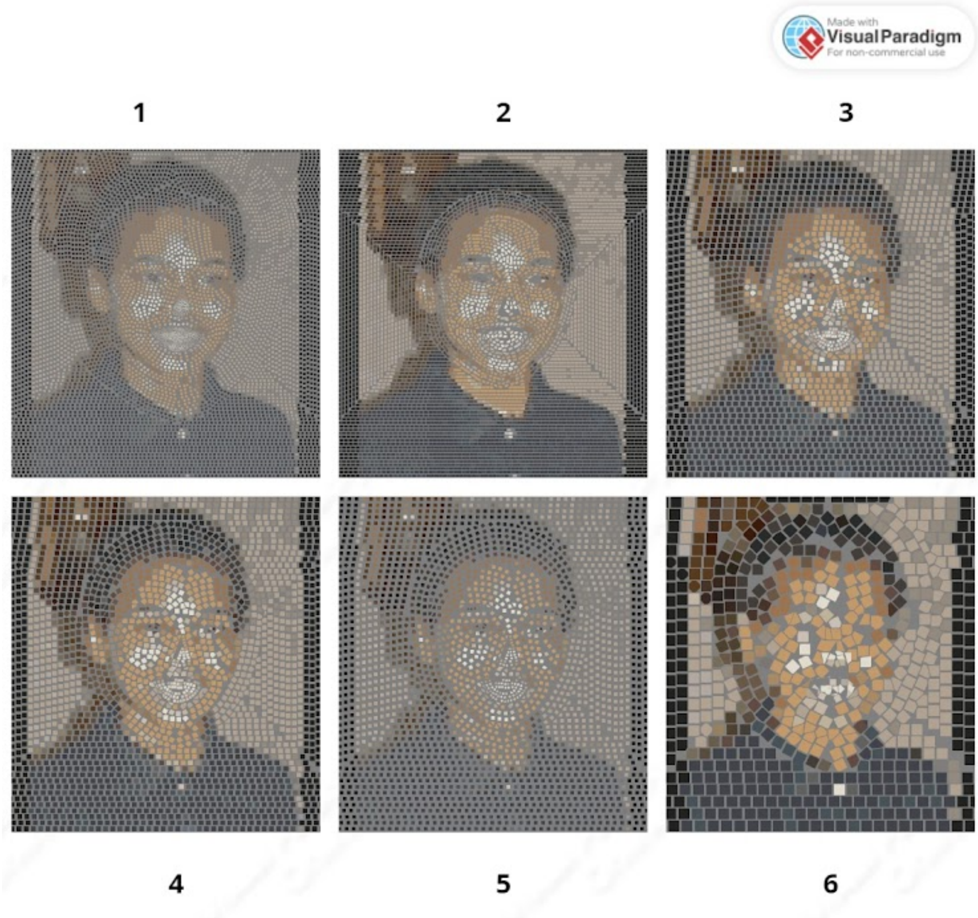

3


4


5


6

5. Order the 3 most aesthetically pleasing mosaic images, based on your personal preference. *



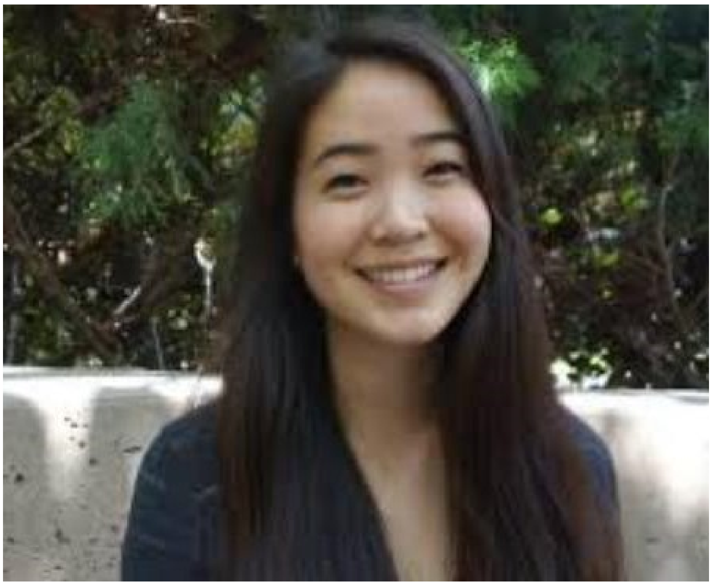
Marcar apenas uma oval por linha.

	Mosaic 1	Mosaic 2	Mosaic 3	Mosaic 4	Mosaic 5	Mosaic 6
1st most pleasing	⬭	⬭	⬭	⬭	⬭	⬭
2st most pleasing	⬭	⬭	⬭	⬭	⬭	⬭
3rd most pleasing	⬭	⬭	⬭	⬭	⬭	⬭

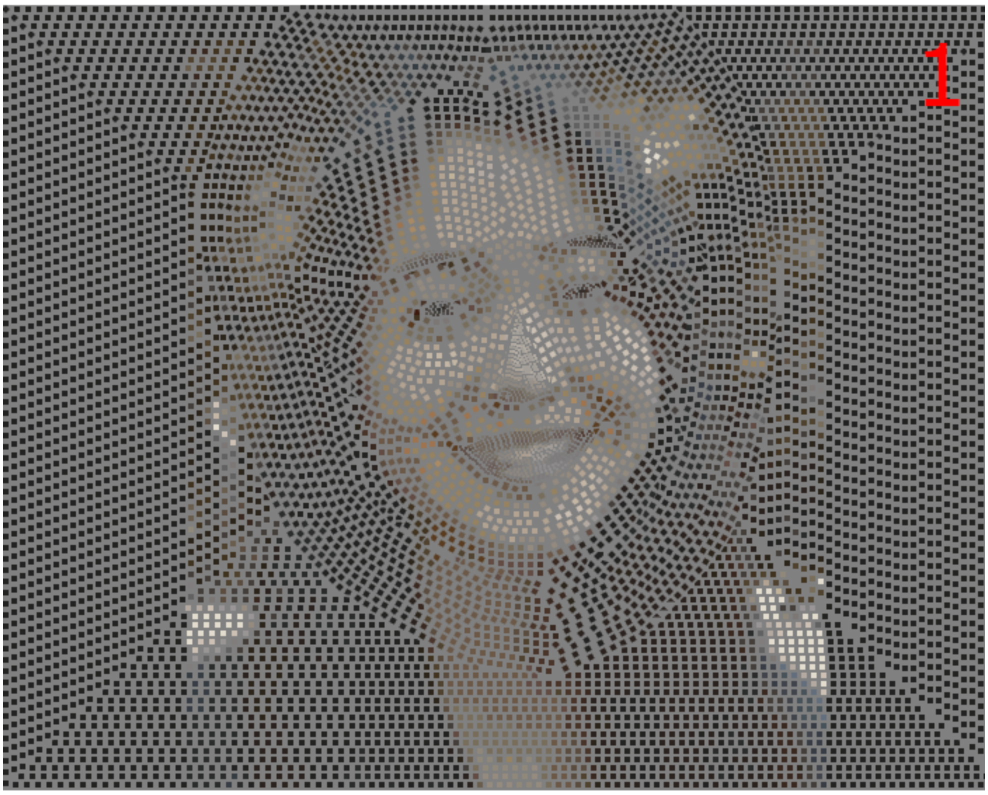
6. Comments or any feedback regarding the generated mosaic images

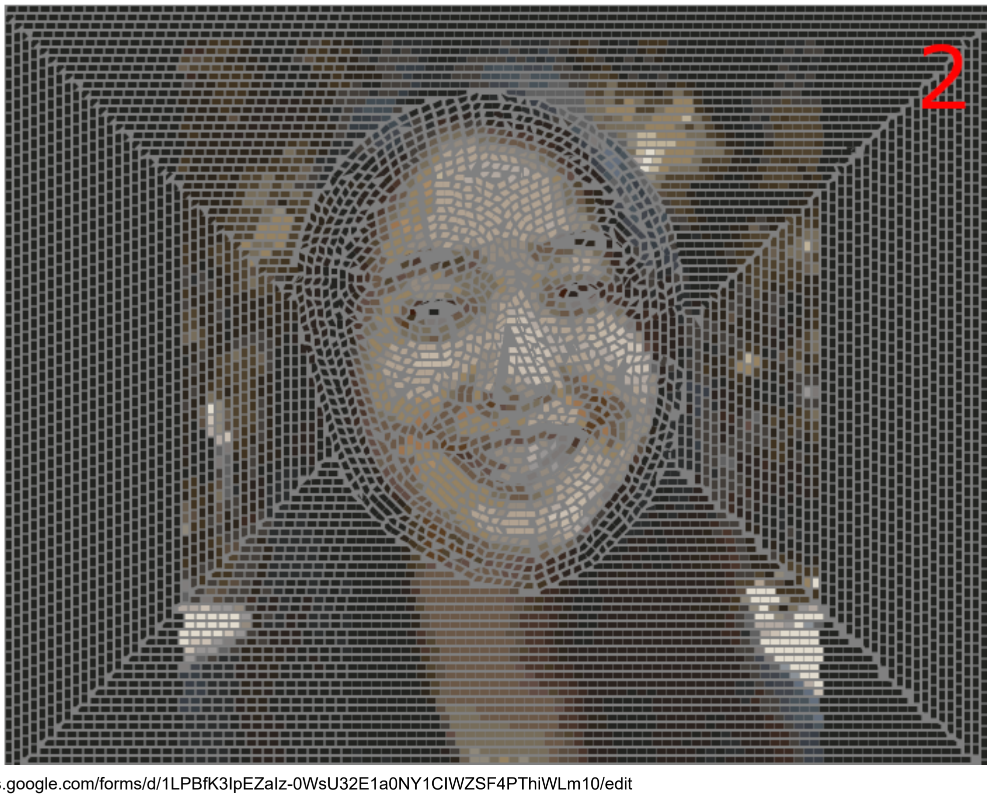
Second set of mosaics

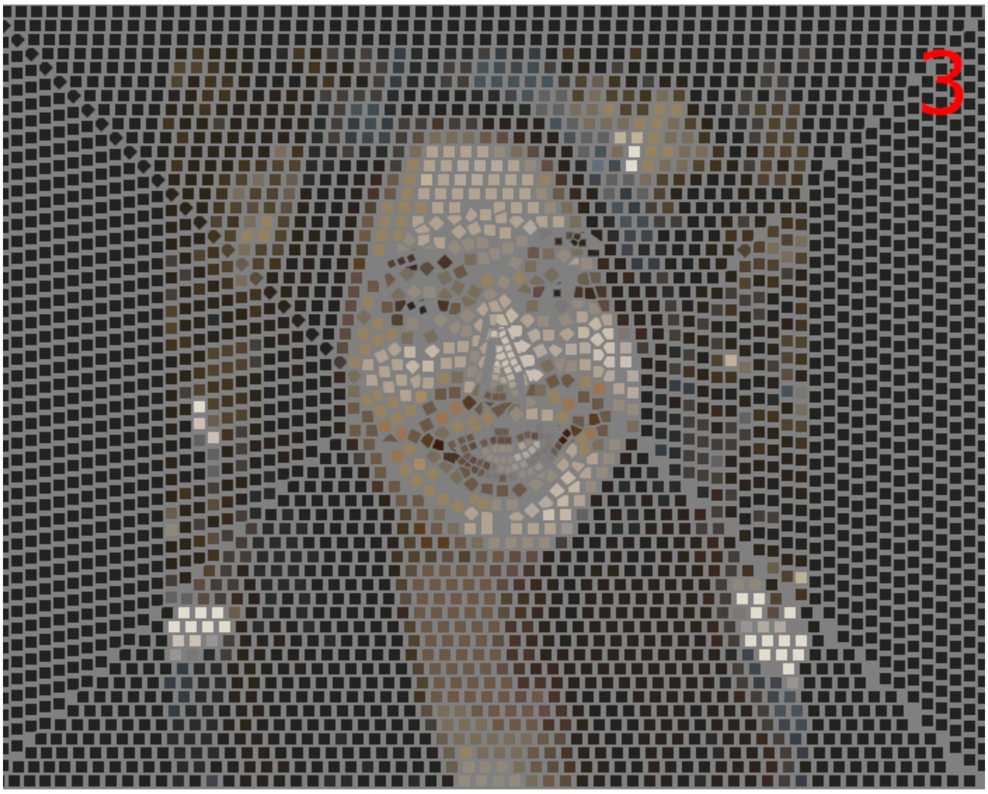
Mosaics were generated using the image below.

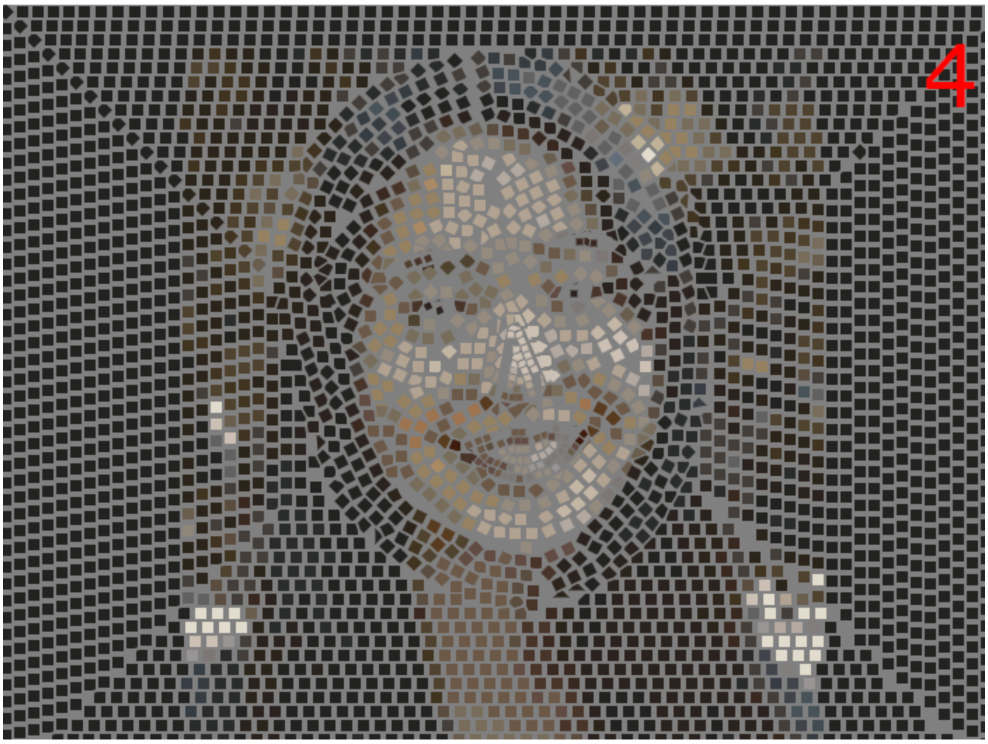


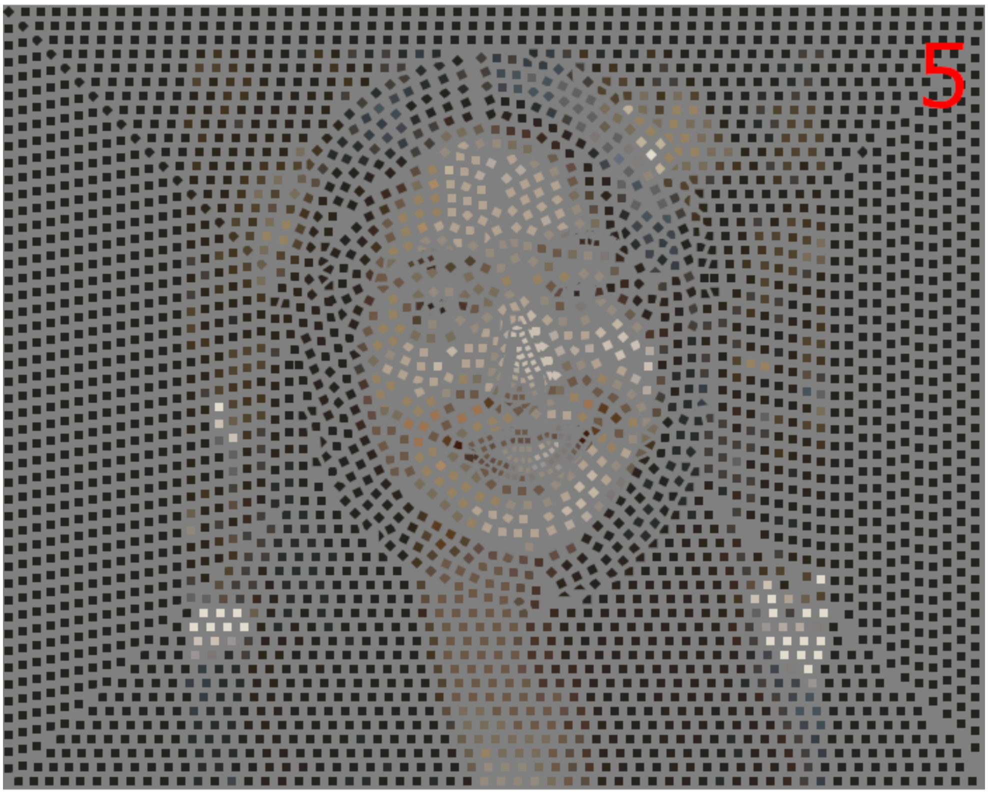
The six generated mosaics are shown below, identified with a red number.

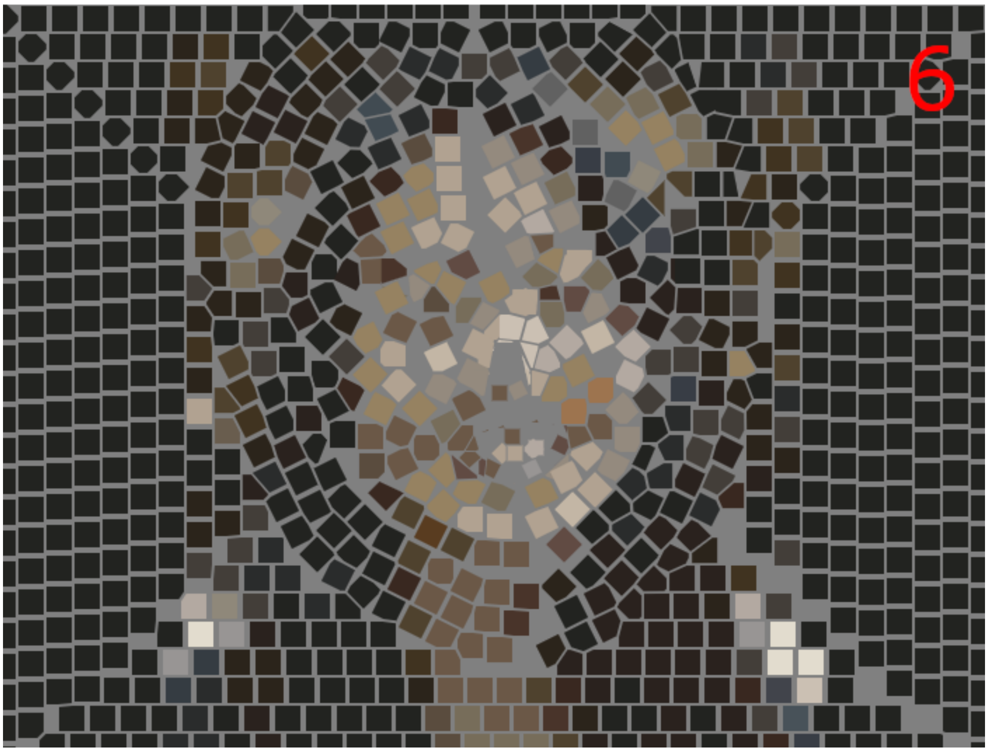

1


2

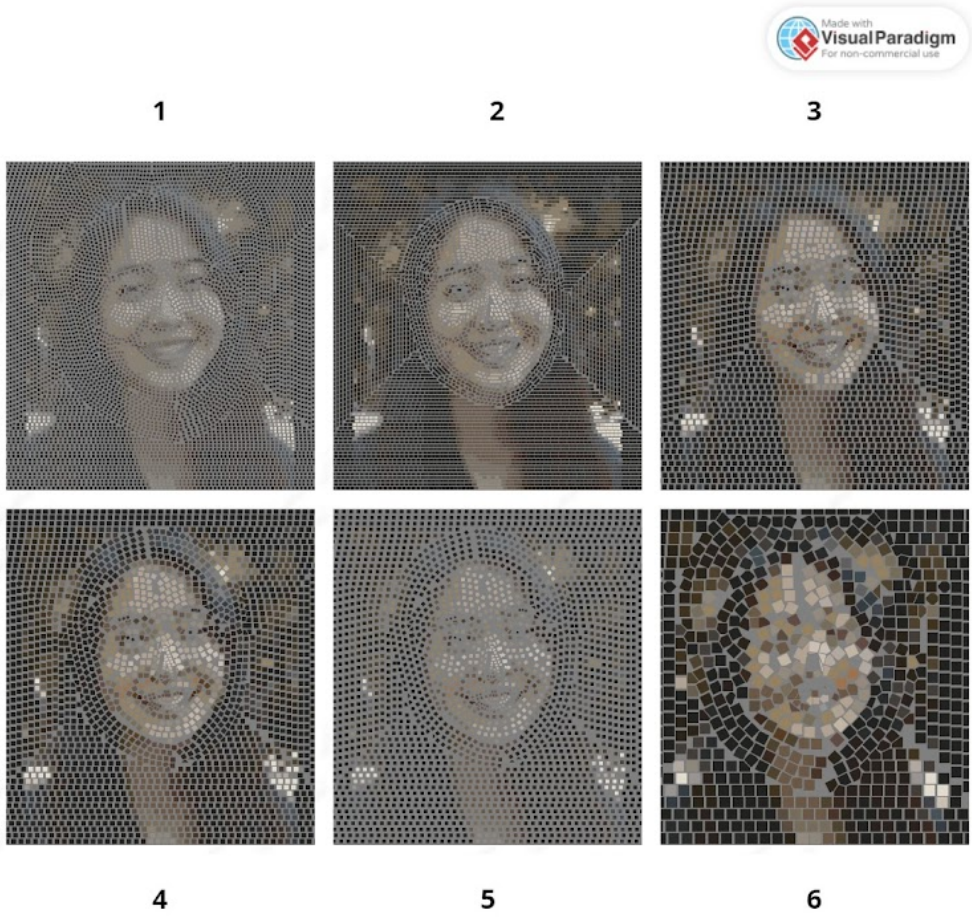

3


4


5


6

7. Order the 3 most aesthetically pleasing mosaic images, based on your personal preference. *



Marcar apenas uma oval por linha.

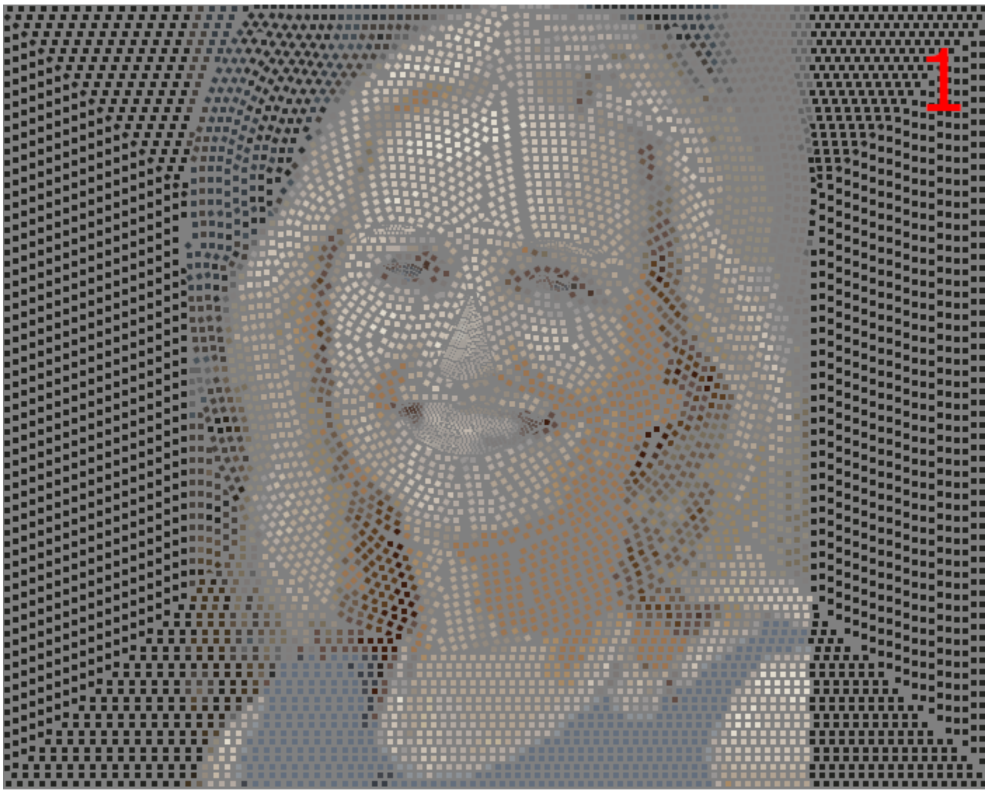
	Mosaic 1	Mosaic 2	Mosaic 3	Mosaic 4	Mosaic 5	Mosaic 6
1st most pleasing	⬭	⬭	⬭	⬭	⬭	⬭
2nd most pleasing	⬭	⬭	⬭	⬭	⬭	⬭
3rd most pleasing	⬭	⬭	⬭	⬭	⬭	⬭

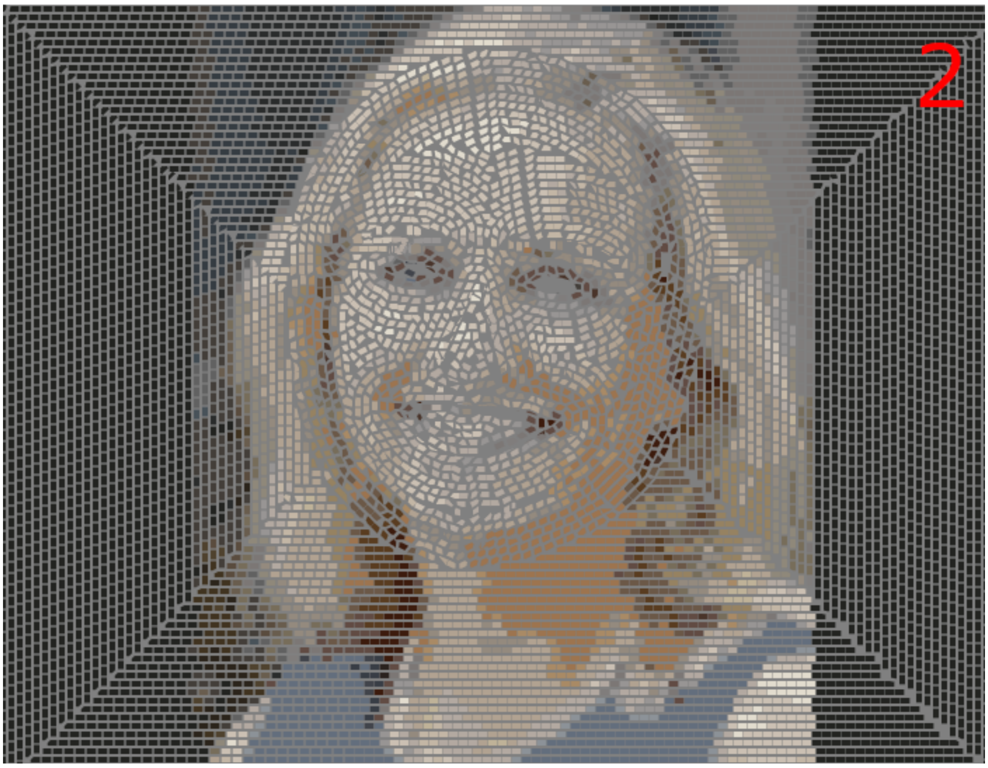
8. Comments or any feedback regarding the generated mosaic images

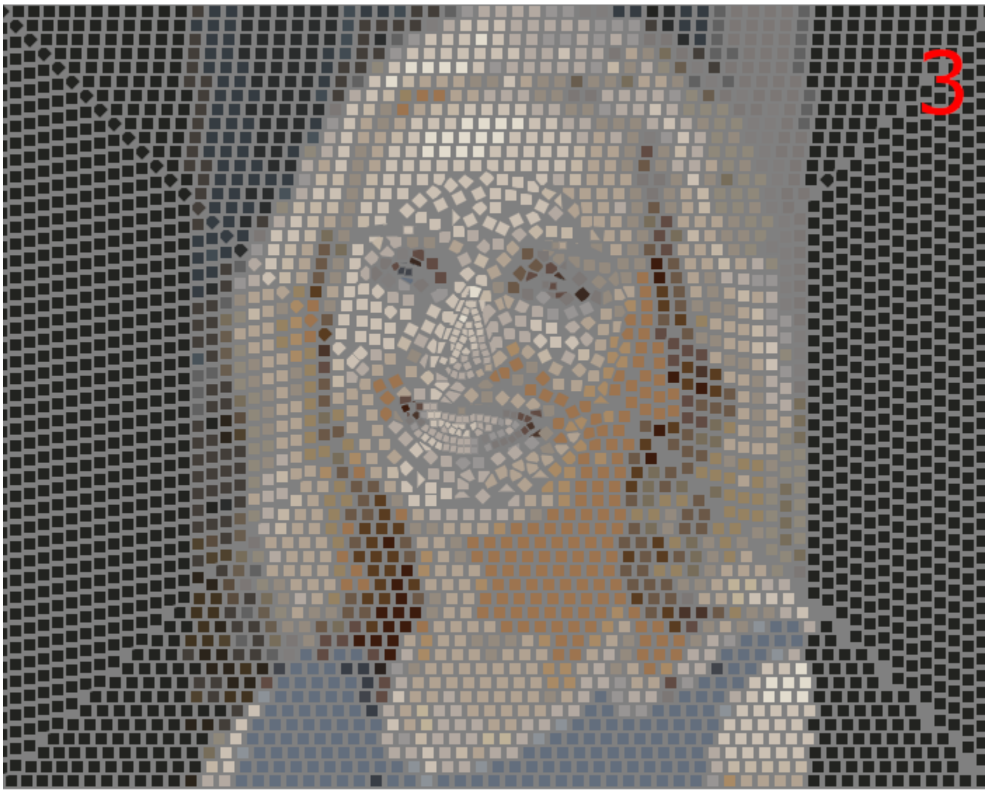
Third set of mosaics

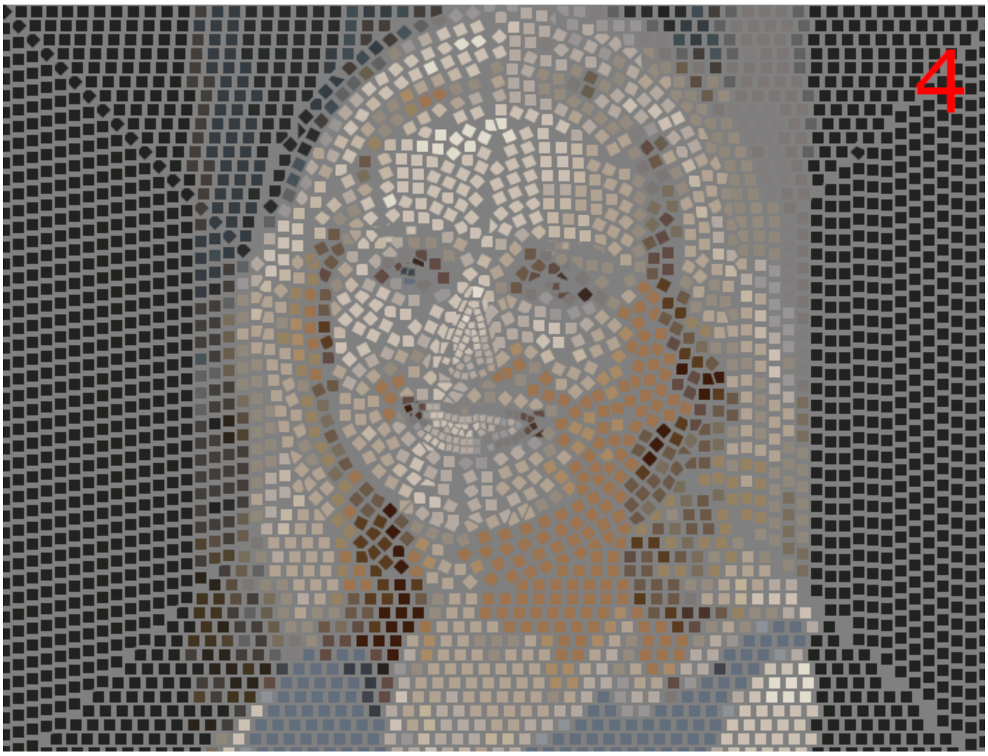
Mosaics were generated using the image below.

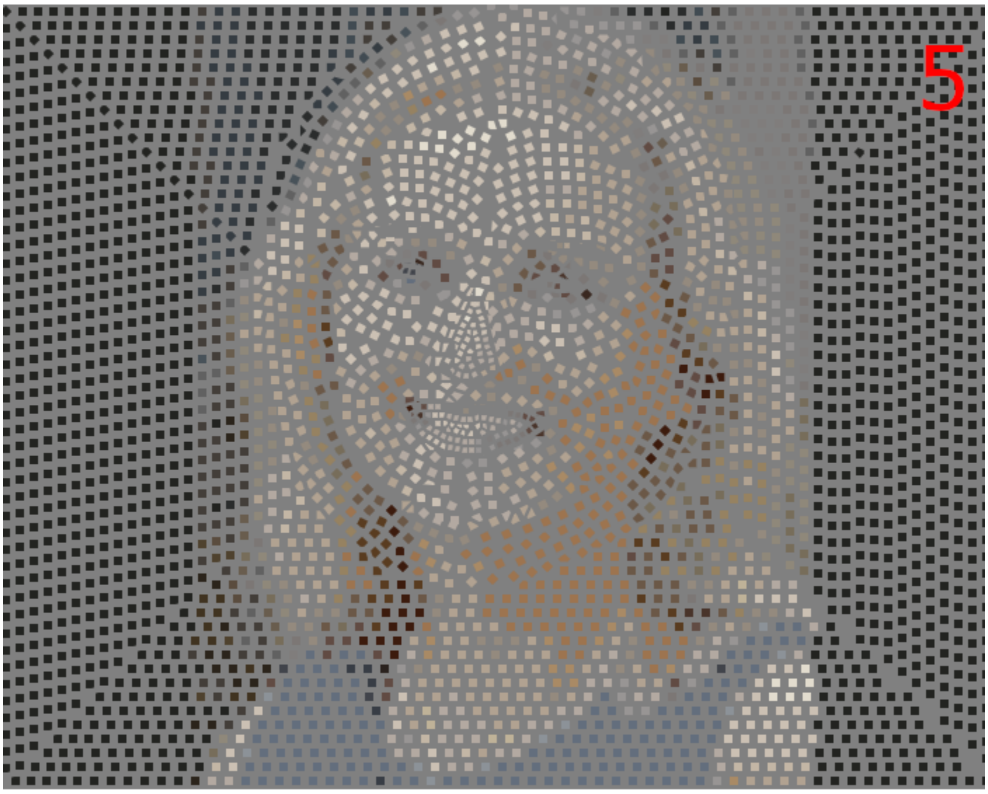
The six generated mosaics are shown below, identified with a red number.

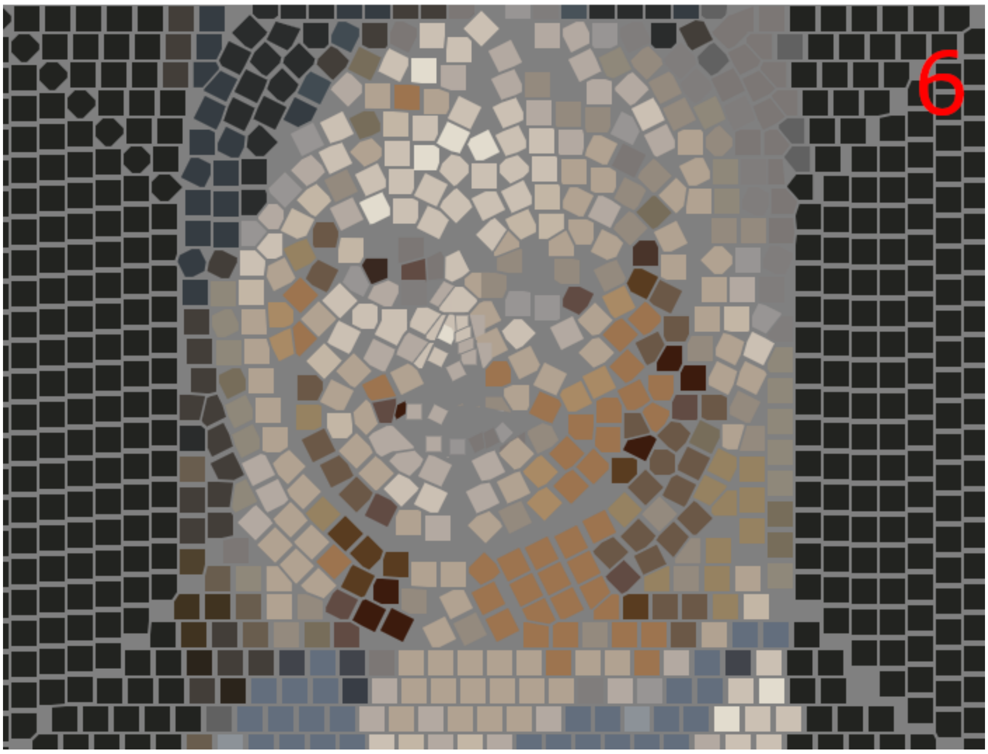

1


2

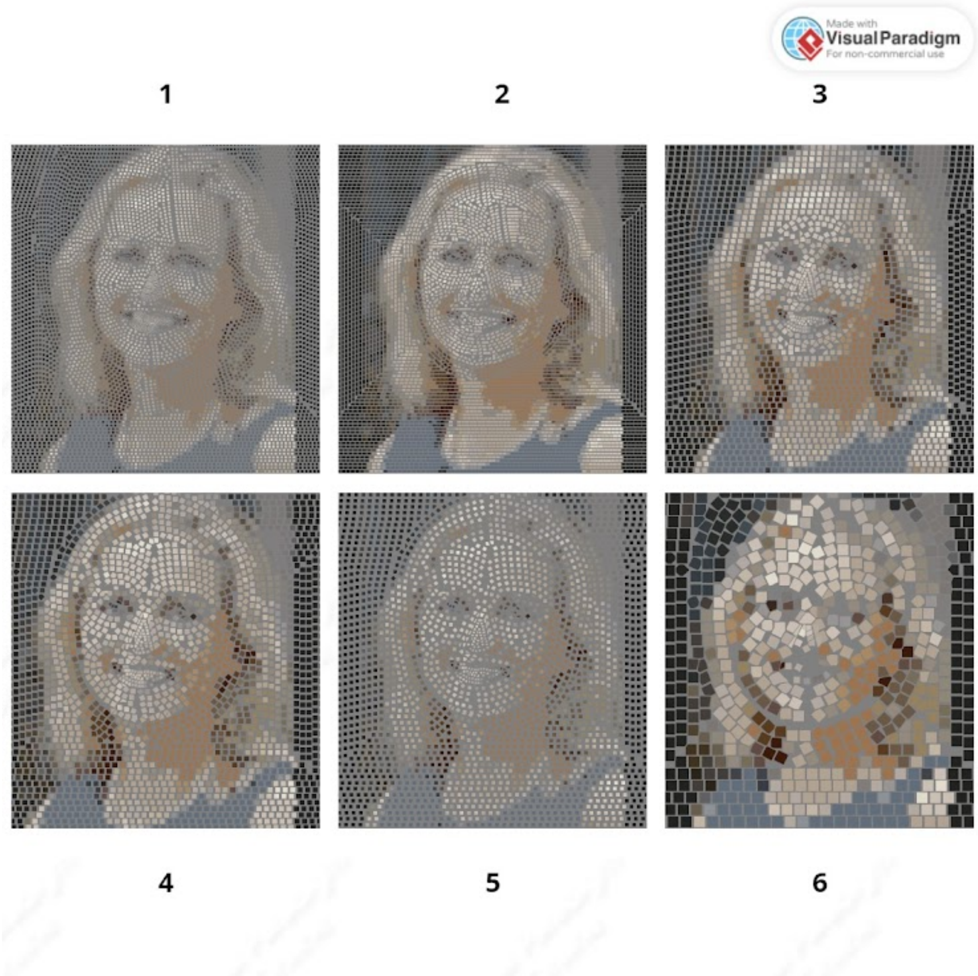

3


4


5


6

9. Order the 3 most aesthetically pleasing mosaic images, based on your personal preference. *



Marcar apenas uma oval por linha.

	Mosaic 1	Mosaic 2	Mosaic 3	Mosaic 4	Mosaic 5	Mosaic 6
1st most pleasing						
2nd most pleasing						
3rd most pleasing						

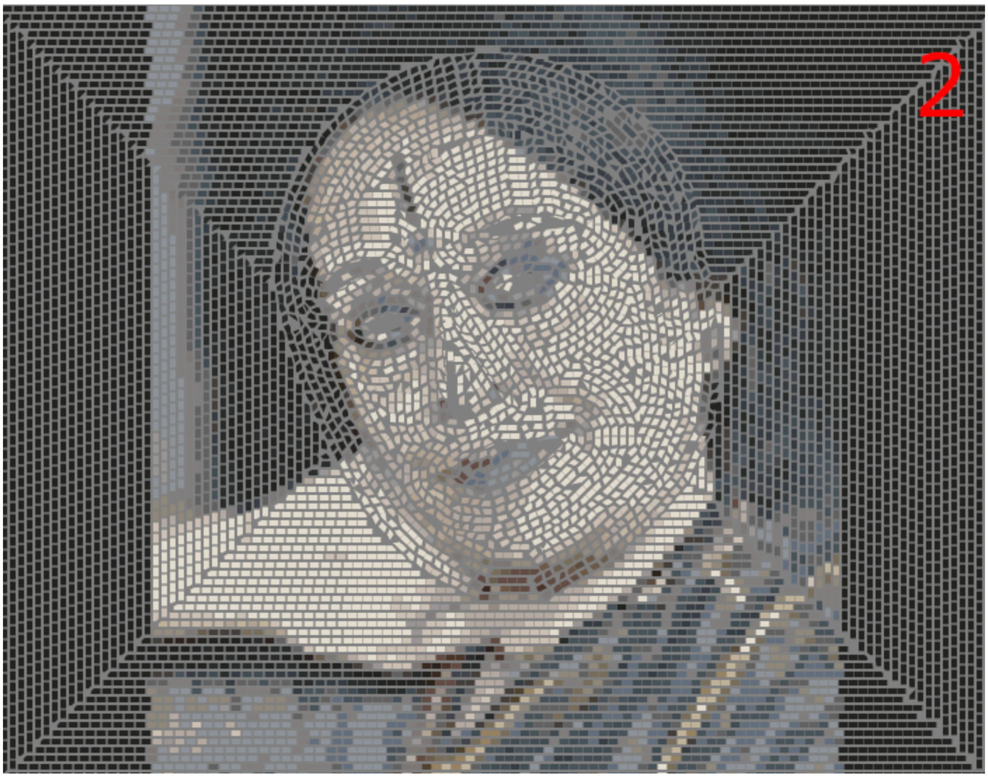
10. Comments or any feedback regarding the generated mosaic images

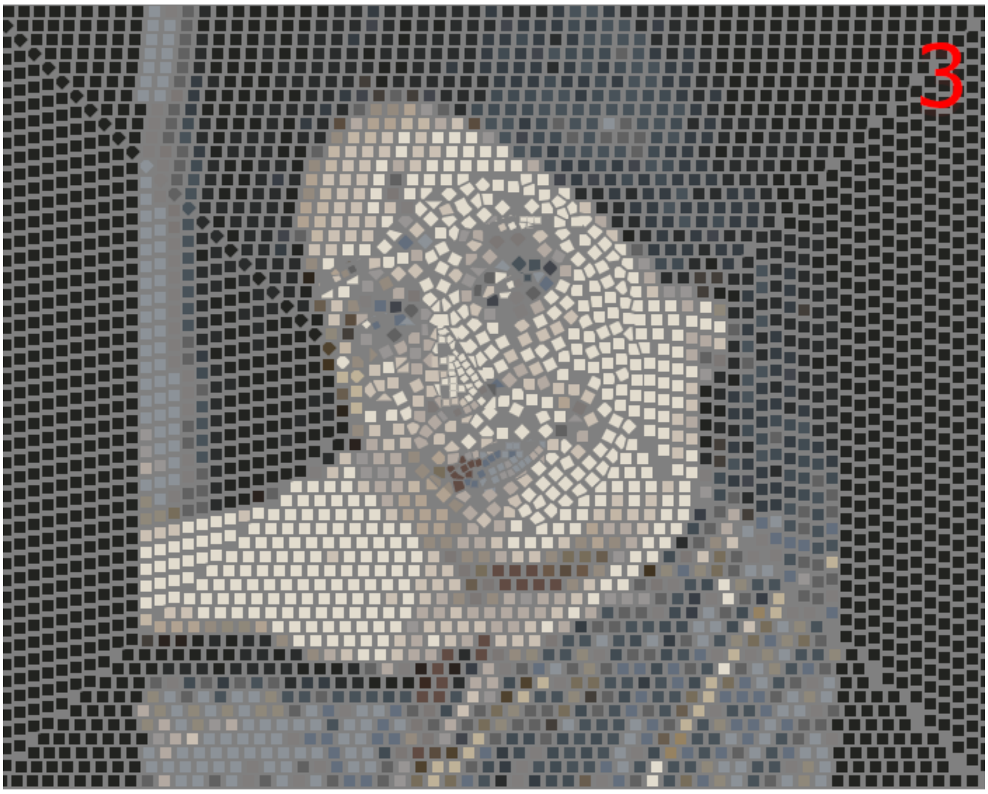
Fourth set of mosaics

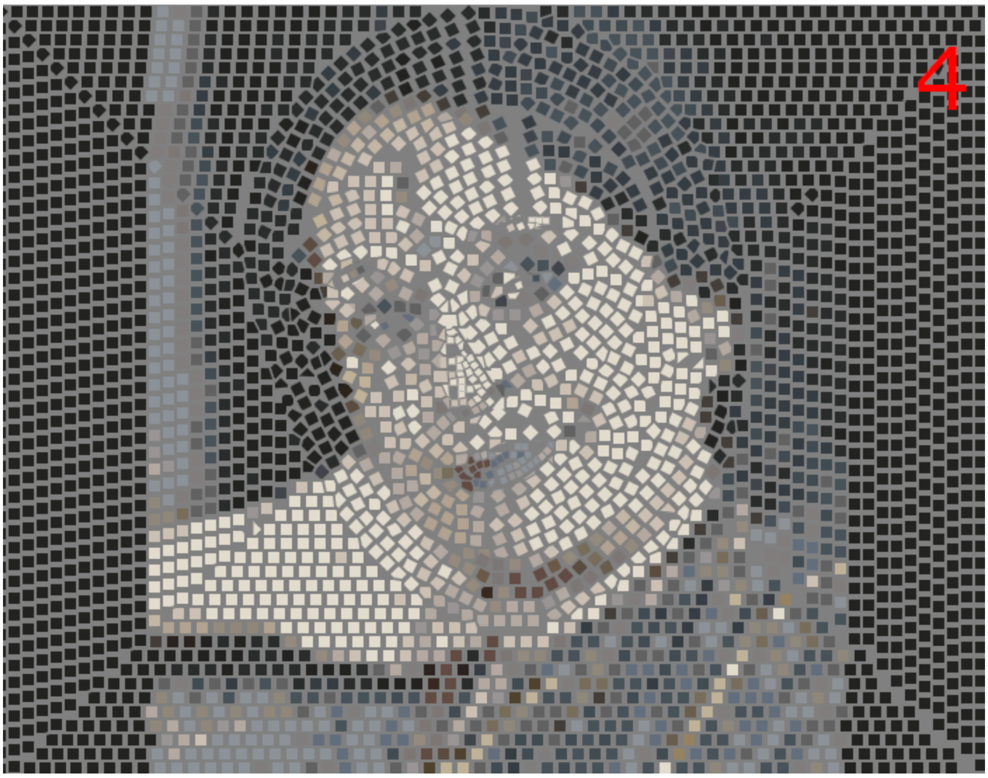
Mosaics were generated using the image below.

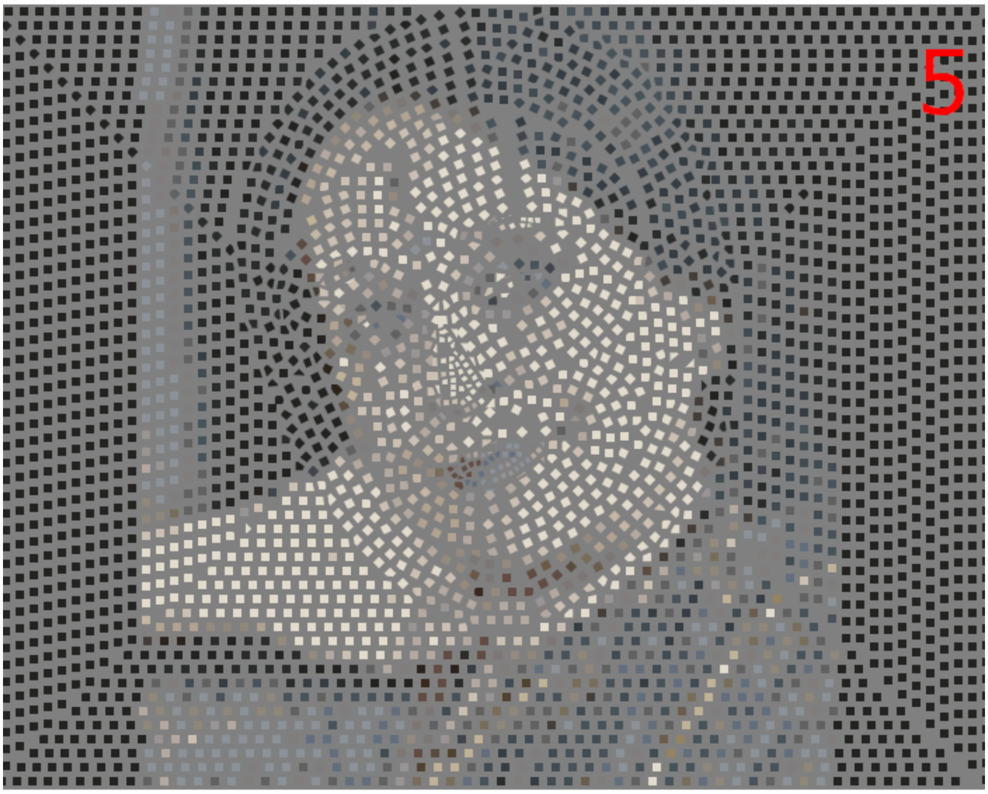
The six generated mosaics are shown below, identified with a red number.

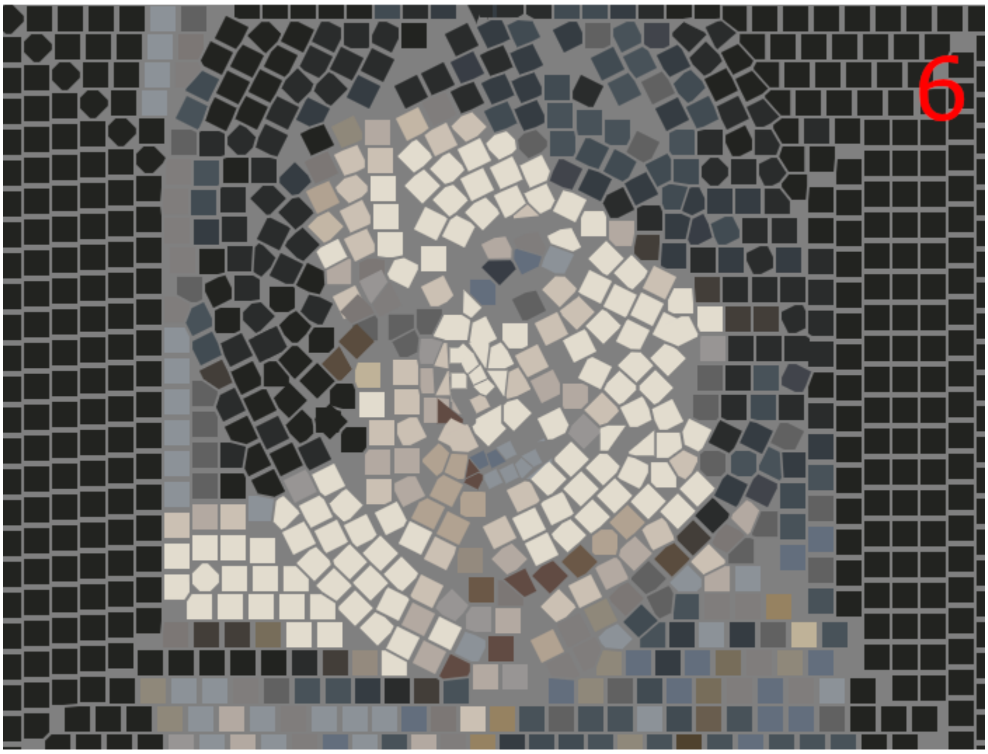

1


2


3


4


5


6

11. Order the 3 most aesthetically pleasing mosaic images, based on your personal preference. *

Marcar apenas uma oval por linha.

	Mosaic 1	Mosaic 2	Mosaic 3	Mosaic 4	Mosaic 5	Mosaic 6
1st most pleasing						
2nd most pleasing						
3rd most pleasing						

12. Comments or any feedback regarding the generated mosaic images

Appendix B

Apendix B - Form used for Kiosk evaluation with users in lab environment

Experiência de utilização do Quiosque

* Indica uma pergunta obrigatória

1. Tipo de mosaico escolhido *

Marcar tudo o que for aplicável.

- [ ] Primeira Opcção
- [ ] Segunda Opção
- [ ] Terceira Opção

2. Gostei do aspeto visual do(s) mosaico(s) gerado(s) *

Marcar apenas uma oval.

	Discordo
1	⬭
2	⬭
3	⬭
4	⬭
5	⬭
6	⬭
	Concordo

3. Caso tenha gerado mais do que um tipo de mosaico, indique o que gostou mais.

   Marcar apenas uma oval.

   - Primeira Opção
   - Segunda Opção
   - Terceira Opção

4. Caso tenha gerado mais do que um tipo de mosaico, indique o que gostou menos.

   Marcar apenas uma oval.

   - Primeira Opção
   - Segunda Opção
   - Terceira Opção

5. O quiosque foi fácil de utilizar *

   Marcar apenas uma oval.

	Discordo
1	
2	
3	
4	
5	
6	
	Concordo

6. O quiosque é apelativo e, num contexto de museu, chamaria a atenção dos visitantes. *

Marcar apenas uma oval.

Discordo

1 ⬭

2 ⬭

3 ⬭

4 ⬭

5 ⬭

6 ⬭

Concordo

7. O mosaico levou muito tempo a ser gerado *

Marcar apenas uma oval.

Discordo

1 ◯

2 ◯

3 ◯

4 ◯

5 ◯

6 ◯

Concordo

8. Em geral, gostei da experiência *

Marcar apenas uma oval.

	Discordo
1	◯
2	◯
3	◯
4	◯
5	◯
6	◯
	Concordo

9. Encontrou algum problema no quiosque? *

Marcar tudo o que for aplicável.

- [ ] Sim
- [ ] Não

10. Se sim, descreva os problemas encontrados

11. Comentários e aspetos a melhorar

Appendix C

Apendix C - Form used for Kiosk evaluation with Museum Visitors

User experience on the kiosk

* Indica uma pergunta obrigatória

1. Type of mosaic chosen *

Marcar tudo o que for aplicável.

- [ ] First Option
- [ ] Second Option
- [ ] Third Option

2. I liked the visual aspect of the Mosaic. *

Marcar apenas uma oval.

	Disagree
1	◯
2	◯
3	◯
4	◯
5	◯
6	◯
	Agree

3. In case you generated more than one mosaic, which one did you prefer?

Marcar apenas uma oval.

- First Option
- Second Option
- Third Option

4. In case you generated more than one mosaic, which one did you like the least?

Marcar apenas uma oval.

- First Option
- Second Option
- Third Option

5. The kiosk was easy to use *

Marcar apenas uma oval.

	Disagree
1	
2	
3	
4	
5	
6	
	Agree

6. The kiosk is appealing and caught my attention. *

Marcar apenas uma oval.

	Disagree
1	◯
2	◯
3	◯
4	◯
5	◯
6	◯
	Agree

7. The mosaic took a long time to generate *

Marcar apenas uma oval.

Disagree

1 ◯

2 ◯

3 ◯

4 ◯

5 ◯

6 ◯

Agree

8. Overall, I enjoyed the experience. *

Marcar apenas uma oval.

Disagree

1 ⬭
2 ⬭
3 ⬭
4 ⬭
5 ⬭
6 ⬭

Agree

9. Did u find a problem with the kiosk? *

Marcar apenas uma oval.

⬭ Yes
⬭ No

10. If u did, describe the problems encountered.

11. Comments and aspects to improve