Advances in Computer and Communications, 2023, 4(1), 25-36
https://www.hillpublisher.com/journals/acc/
ISSN Online: 2767-2875

Colouring the Sculpture through Corresponding Area from 2D to 3D with Augmented Reality

Yaozhong Zhang

University of Nottingham, Nottingham, UK.

How to cite this paper: Yaozhong Zhang. (2023) Colouring the Sculpture through Corresponding Area from 2D to 3D with Augmented Reality. *Advances in Computer and Communication*, 4(1), 25-36.
DOI: 10.26855/acc.2023.02.004

Received: February 15, 2023
Accepted: March 10, 2023
Published: April 7, 2023

***Corresponding author:** Yaozhong Zhang, University of Nottingham, Nottingham, UK.

Abstract

With the development of 3D modelling techniques and AR techniques, the traditional methods of establishing 2D to 3D relations are no longer sufficient to meet the demand for complex models and rapid relation building. This paper presents a prototype development implemented that creates many-to-many correspondences by marking images and 3D model regions that can be used to colouring 3D model by colouring images for the end user. Further development related to the creation of rendered textures by changing the Zeus bust model texture and building in the AR environment. The results of testing each part of the prototype show the viability of creating render textures relation method. The advantages of it are easy to build many-to-many relations and adaptability for any model with proper UV mapping. But there are still three main limitations and future work will focus on solving them, building a database to store relation information data and 3D printing coloured models in the real world.

Keywords

Augmented reality, AR colouring, render texture, Unity, 2D to 3D relation

1. Introduction

Augmented reality colouring technology creates a more vivid experience for the user by combining it with traditional colouring methods [1-3]. The most appealing aspect of AR colouring is that it is not limited by space or conditions, which means that in today's world of smart devices, the user can access the program in any scenario by using any set object as a target while using the tools within the program to complete the colouring operation.

Many sculptures have faded due to historical or human factors, and with the help of modern science, archaeologists and scientists have discovered the colours of some ancient sculptures to unveil their original appearance [4-6]. In 2008, the reconstructions made by archaeologists Vinzenz Brinkmann and Ulrike Koch-Brinkmann in the exhibition Gods in Colour showed to viewers many vibrant pictures of the former polychrome sculptures [7]. Normally, experts often use artificial repair work and 3D modelling software, such as Maya or Blender to restore or recreate sculptures model. Acropolis Museum uses digital superimposition of original colours onto a 3D computer-generated image of the Peplos Kore sculpture and also creates a 2D online digital interactive game Colour the Peplos Kore [8] for visitors using colours and brush they choose [9]. Using digitization tools to quickly help scientists and archaeologists construct and repair polychrome sculptures not only saves time and resources but also can provide visitors with an educational and fun experience.

However, due to the volume and surface complexity of the sculptures, some existing ways have to reduce the model's volume or ignore the details [10], which would compromise the physical characteristics of the original sculpture. The majority of the AR colouring work so far has been based on textbooks or pages, focusing on hand-drawn textures for display on 3D models [1, 2, 11], and some of the work is based on direct colouring to 3D models, interacting with models manipulating in virtual space [12, 13]. Therefore, the establishment of 2D-3D correspondence relations through AR technology still has broad application potential.

After investigating and experimenting with currently viable methods of model colouring and comparing them, first of all, I found that they all have strengths and weaknesses that are difficult to compensate for each other, they all directly satisfy the users with colouring needs using the matching established by program's producer [1-3, 10, 11]. In addition, the models are also relatively simple, which may be related to the amount of workload and ease degree of modelling of the program, as well as the final commercial presentation [28, 30].

As the technical barriers to modelling become lower and lower, and a large number of models are now available for free download online, the fastest way to improve efficiency is to build a program that allows users to create their own correspondence relations between models and images, the relations also could be used by others, so that more models could be developed and used. It would directly impact the museum's work to restore the polychrome sculpture.

With the aim of allowing users to quickly obtain the 2D-3D correspondence and use it to colour the 3d model through images with AR, this paper describes a prototype design that can colour 3D sculptures through images. In the prototype, users can use a mobile device with a camera to scan the image target into the program interface and use a touch screen or capacitive pen to create a matching area between the model and the image. By identifying the location of a finger or capacitive pen pixel on the image, the user can colour the area according to their preference or reference colour. The prototype can be used to colour pages with different image targets in textbooks, and electronic devices in the home or museum environment for helping restoration, inspiring artists, and providing education for the public and children.

2. Related Work

2.1. Three-Dimensions Paint

In 3D computer graphics, a common method for increasing the apparent detail of objects without increasing the complexity of the models [14]. In the last three decades, the development of 3D painting has been pursuing a way to change the texture map of 3D models using brushes and colours simulated by computer technology in virtual space. Developers realized that textures have a very important influence on the appearance of the model, so much of the basis for 3D painting is also based on textures.

In 1990, Hanrahan and Haeberli introduced an interactive paint program, which allows users to paint different types of paints and materials directly onto 3D shapes [12]. Users can utilise the tablet to control the position of brushes containing shapes and paints, and the created material properties are stored as a set of associated texture maps in the end. The main contribution of this work is using rendering techniques as the user paints to create the appearance of the surface of the object and the result of the interactive colouring is a reproduction of the 3D shape. The author also gives a full description of the problems of creating and mapping texture maps, firstly the problem of defining the coordinates and orientation of textures, and secondly the problem of deformation of texture maps due to parameterization. Subsequent works in 3D painting have also evolved in the direction of a more realistic view.

Distinct from traditional 3D painting programs that use predefined UV maps, Takeo Igarashi and Dennis Cosgrove present a technique for creating effective UV maps for texture painting programs [15], this system of systems dynamically creates tailored UV maps for newly drawn polygons during the painting process. The user can draw un-deformed strokes over the entire surface from any direction. Dynamic texture distribution allows the user to draw smooth strokes at any zoom level. This technique can be effectively implemented using standard 3D rendering capabilities without the need to write code for pixel-level manipulation. The result of the drawing is stored as a standard texture polygon model and can be read by various graphics applications. This work gets inspiration for texture generation and preservation for 3D painting. Lu et al. allow the user to generate texture maps by drawing directly on the surface of the 3D model using a defined UV map in relation [16]. During the painting process, when the user paints strokes in the 3D view of the object, the system subsequently re-projects the user's strokes to the corresponding positions in the 2D texture bitmap based on predefined UV mappings (texture coordinates).

The proliferation of touch screens has made 3D painting no longer limited to computer programs. Many previous works have demonstrated that using a haptic device to draw textures directly from the surface of an object is an effective technique for creating textured images intuitively and interactively [17-19]. In 2008, Wakita et al. propose a system for designing haptic models based on textures for haptic devices [20]. In the system, differences in tactile impressions are rendered by dynamically varying the magnitude and/or direction of the reaction force based on the pixel values of the object surface, which maps the particular texture image and converts roughness, stiffness, and friction into a 2D image.

With the advancement of technology, the development of software has made it more realistic for 3D painting to be done. One of the most important developments is the render texture. Render textures are special types of textures that are created and updated by specifying a camera at run time in Unity software [21]. The Camera component has a Target Texture variable, which can be set to a Render Texture object to output the camera's view to the texture rather than the screen [22]. Monitoring the display is a good metaphor for how it works: the camera is the camera, the rendered textures are the video recording, and the material is the screen your video is projected onto.

In 2015, the RealTime Painting Project [23] was published in Unity and assessed by Rodrigo Fernández. The developer uses render textures to simulate painting the texture of the material of the mesh in real time. In this project, the camera of the rendered texture looks at a quad with the UV map of the model. And then anything that is placed in front of the camera will be added texture. Painting mesh in this way will allow us to have a lot of freedom and will have very clean results for the end users, for now, after merging texture it won't allow users to roll back the changes made.

2.2. Augmented Reality Colouring

The definition of augmented reality (AR) used in this project refer to the survey proposed by Ronald Azuma in 1997 and a book written by Alain Craig in 2013. The survey defines AR as any system that has the following three characteristics: combines real and virtual; interactive in real time; registered in 3D [24]. In 2013, Alain Craig further extends the augmented reality definition, not simply including the essential components of AR, but also defining it from users' perspective. He defined augmented reality as a medium of the user's experience that is interactive, at least by changing perspective dependent on the location and perspective of the viewer and the digital information is overlaid on the physical world, that is in both spatial and temporal registration with the physical world and that is interactive in real-time [25].

The target users of the majority of AR colouring works are children and based on the context of colouring books or colouring pages. In early AR colouring works, it is common to use image registration and image inpainting techniques to map or extract the image texture to the visible and occluded regions of the 3D model. The users could utilise the smart devices with camera scanning and recognize the colours and brush strokes to form the model-wrapped texture map on the screen. In 2012, Clark and Dunser at The HIT Lab NZ created an interactive AR colouring book that automatically performs registration and texture extraction from colouring book pages. Allowing the user to in-novate the colour of the book's content and automatically map it to virtual pop-up 3D scenes and models based on what the user draws with a coloured pen. The key to the technique is image registration of pages that have been colour removed by computing the RGB colour threshold within a pixel through natural feature registration. The software identifies the black and white line art as the baseline by the light intensity and the saturation of colour on the drawing [1]. Currently, Quiver [26], a company developed by The HIT Lab NZ, is producing and releasing commercial products about AR colouring.

Lee and Choi use Vuforia SDK (Software Development Kit) with Unity 3D to bring more possibilities to the AR colouring book. They propose an algorithm to extract a specific rectangular region from the live video and use it for colouring book's texture map. The work involves the processing of image deformations by calculating the transformation matrix from local coordinates to world coordinates, and from the world, coordinates to screen coordinates. In addition, the perspective created in reality due to the deviation of the camera from the page angle was corrected [27]. Based on the key technology, the AR colouring puzzle, Dinosaur Sketchbook [28], was built and then developed commercially, which provides its technical viability [3]. After that, Cho et al. used colour image acquisition and processing algorithms, matrix computation and image processing algorithms to build a colouring game-Playing House. The aim is to establish connections between the real world, the AR world, and the VR (virtual reality) world, so that all worlds are seamlessly connected together, enabling users to access the same content in different visual ways [29].

Norraji and Sunar built wARna (Wonderful Augmented Reality and Arts), a mo-bile-based interactive augmented reality colouring book, by texture extraction and mapping using benchmarking and image processing techniques, the colouring system extracts textures within the frame and maps them to the corresponding 3D model by detecting the marker frame [11]. After that, they developed colour recognition and colouring programs for preschool learning based on the wARna project for the educational purpose [30]. This work uses printed line art provided by the manufacturers, thus avoiding the problem of curved pages caused by bookbinding.

A major limitation in these works is that users cannot fully create 3D content. While users can colour 3D content, they will always need to prepare publisher-supplied print art lines. In order to increase interactivity and address the cause of user boredom caused by the post-colouring model's view-only functionality, many colouring works are also actively exploring the use of post-colouring models.

To the problem of the camera's angle deviation from the page arising in the use of the system, Magnenat et al. present a real-time variable surface tracking and texture synthesis method for the AR colouring process. Deformable surface tracking allows warped painted pages to be tracked in real-time on mobile devices, increasing the interactivity of the experience [2]. Real-time texture compositing uses textures drawn by the user on the 2D image and applied in real-time to solve mirroring and parameterization seams artefacts, which proposed algorithm is based on both diffusion-based and patch-based methods, as it both extends the known parts of the image to the unknown ones and copies pixels know regions to unknown ones. According to the testing results, texture composites aren't perfect, but acceptable to most users. Unlike another AR colouring, which can just see the coloured model after colouring, live colouring is more interactive for the user in real-time. MagicToon solves the problem that AR content cannot be freely created by users, the system can automatically build 3D models according to user-drawn 2D images and interactively edit more complex scenes on the AR screen [10]. The system used Natu-raSketch [31] inflation method including diffusion-based and patch-based to calculate a distance map to encode the distance of every pixel to the nearest black pixel and use the inflation method to expand the drawing to the entire region. This method can be used for symmetric 3D models and the limitation is the original texture on the image is used as textures for both the front and back surfaces of the 3D model.

From the above summary of works, most of the works are based on changing or generating texture maps for implementing colour modification. For this project, redesigning the user interface is important when developing a mobile AR colouring prototype. Moreover, establishing one-to-multipoint mapping relations requires consideration of finding ways from the model generation steps.

3. Prototype Development

3.1. Background

This project is going to use the prototype development method to try a technical task of colouring sculpture, the focus is on the feasibility of the idea to evaluate its performance or the performance of some of its components. The crux of this work is in building a mapping relation between a front-view image of the 3D model and a texture map wrapped on the 3D model. Ideally, this relation can be established by different users, then the end users could utilise it to transform the colour of the 3D model by painting on the image.

This project utilises the Unity 2019.2.8 f1 version and Vuforlia version 8.5.9. The advantages of Vuforia are multi-platform release and 3D object target. The project can be loaded on iOS, Windows, and Android system, which provide convenience for subsequent expansion of the system and full platform coverage. Moreover, 3D object targets might provide a realistic way for users to trigger systems and view completed AR works in the real world.

For the model, I referenced sculptures from the British Museum, many of which are now digitally conserved, including the creation of digital models of the statues for public study. As the statues themselves have been damaged for historical reasons, the fragmentation of the models can affect the effect of the colouring. So, I chose to base my work on a more complete bust of the Zeus digital model (Figure 1). A bust is a composite object with the head being antique roman, nose, body, and socket base an 18th-century restoration made for fitting the head, which was probably an over-life size statue originally [32]. The model was updated by Thomas Flynn in 2016 by Agisoft PhotoScan using 171 photos [33]. The other reason for choosing this 3D model is its UV mapping properly, and in texture mapping, it is difficult to get the model position by pixels using the method in the above work. This is challenging for the next prototype development.

3.2. Target User Group

The target group of users for this project includes museum colour restorers, as well as members of the public with colouring needs. The end users can take advantage of a relationship that can be created in a relatively simple way by the creator.

Most of the works previously described in Chapter 2 were designed to face the user directly for colouring, with the developer specifying the 2D-3D correspondence, and the user using this relation to colour the model, generating colours or brushstrokes. But with the increasing simplicity of 3D modelling technology, such as RealityCapture software [34], which only needs to take multi-angle photos of a model, there will be more and more resources could be used to develop in the future. Therefore, the workload of developing programs by traditional workflow is enormous for companies or individuals.

In this prototype, I consider two types of users: the creator, who can use the system to quickly establish correspondence; and the end user, who uses the correspondence to colour sculptures.

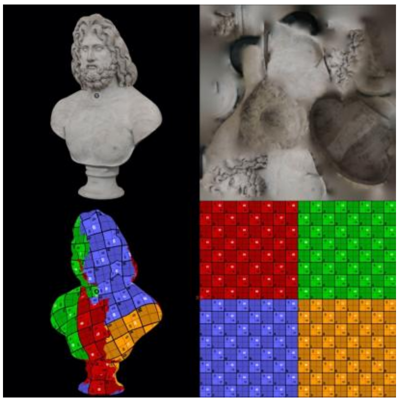

Figure 1. Colossal marble bust of Zeus model [36], 3D model (top left) and texture map (top right) 3D UV map (bottom left) and 2D UV map (bottom right).

3.3. System design

In this method, I tried to modify the texture map itself, and create a mapping between 2D and 3D. Inspired by the aforementioned RealTime Painting project, the 3D model can be coloured using render texture in an interactive way, the same can be utilised in the 2D images as well. So, my conception is that there are two modes in the system, one for the creator mode and the other for the user mode. In the creator mode, the creator can establish a matching relation between the rendered texture map and the image by painting the colours and saving it locally. In the user mode, the user can change the colour of the 3D model by colouring the front view of the model, using the matching relations that have been established by the creator.

In Unity software, there are three scenes that connect the entire program. Scene0 is the menu page, which includes creator mode and user mode. These two modes are settled in Scene1 and Scene2 respectively.

Scene 0 is a menu interface with *Creator* and *User* buttons in the middle of the window, corresponding to creator mode and user mode respectively. In *Scene1*, the interface consists of the following parts: the 3D model of the sphere that used in the concept design is shown in the middle of the screen; the colour panel is at the bottom left corner for picking the desired colour by users; the brush control bar is on the top of the colour panel, which is for controlling the size of the brush by dragging to the left or right and the system is to use the maximum brush by default; the *Revoke* button that will appear after painting and the Save button are both in the lower middle; the front view area of the model, which is circular in the prototype in the lower left; the last one is the *Back* button in the upper left corner, which is used to return to the menu. The layout in *Scene2* is the same as in *Scene1*, except that *Scene2* doesn't need the *Revoke* button, the *Save* button, and the brush size control bar.

The user flow diagram, as figure 2, shows how the two user groups operate the program. In the beginning, the program will enter the Creator mode by default. In the Creator mode, the creators can select the colour by the colour panel to mark the desired area in the 2D and 3D parts. First, use the brush to paint on the model to mark the area, then perform the same operation on the front view image. In the process of marking the areas, if the creator is not satisfied with one of them, the creator can revoke it. Besides, if one of the parts has no mark after clicking the Save button each time, the relation cannot be saved, and the system will prompt by missing data. After saving, the creator can choose to continue to create the relation using other colours or return to the menu interface. The user can enter the User mode through the menu interface, they only need to fill in the image with any chosen colour, and at the same time the corresponding area of the 3D model will be coloured as well.

Technically, in the process of implementing the prototype, the main situations are as follows:

1) The relation between the UV Map and the point of interaction;
2) How to mark on render textures;
3) How to save the relation between render textures for user colouring.

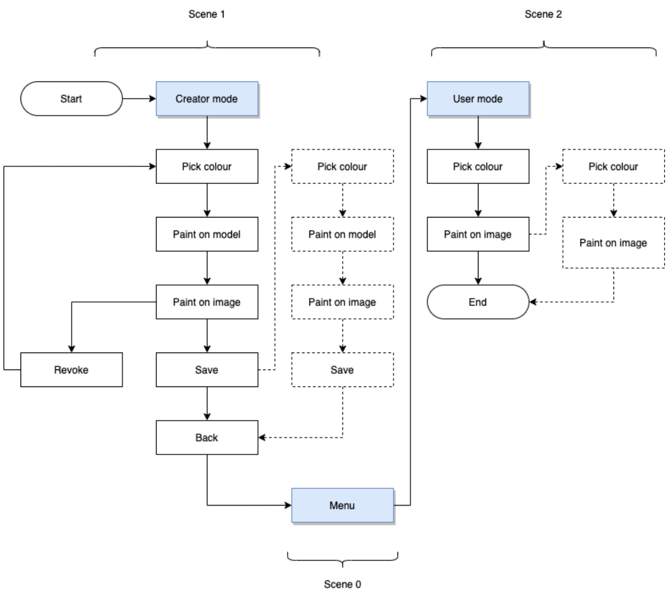

Figure 2. Creator and User flow.

3.4. Render Texture Relation

In order to establish a relation, firstly, the creator needs the marking the mesh. I used the render texture to draw the mesh material texture in real-time. The cameras of the render textures of the 3D model and image look at materials named *MeshBaseMat* and *UIBaseMat* with the UV map of the model, the only difference between them is the render texture on the image doesn't need to map to any model. In the prototype, there is no texture on them so that just shows a white background for colouring. The materials should have *'Unlit /Texture'* as a shader to avoid unusual colouring. So, anything that is placed in front of our camera will be added texture.

To mark the sphere, I create a material named *MeshMat* using the texture for mesh with *'Standard' Shader* and adding on *DrawSphere*. After adding the material to the mesh, I need to use *Raycasting* and a Unity function for determining which part of the UV map are users hitting, and then users can place things in front of the camera to simulate colouring over it. The front view will be marked in the same way as above mentioned. In order to display the circle image, I have put it on the texture of the *DrawPanel*, and then all markers will be within the scope of this circle.

For marking on render texture without having too many draw calls, I used *Sprites* to change brush instance colour without creating a material instance to allow Unity to group these sprites together. When the creator starts marking, the Sprites will be generated for each created brush under *BrushContainer* (Figure 3).

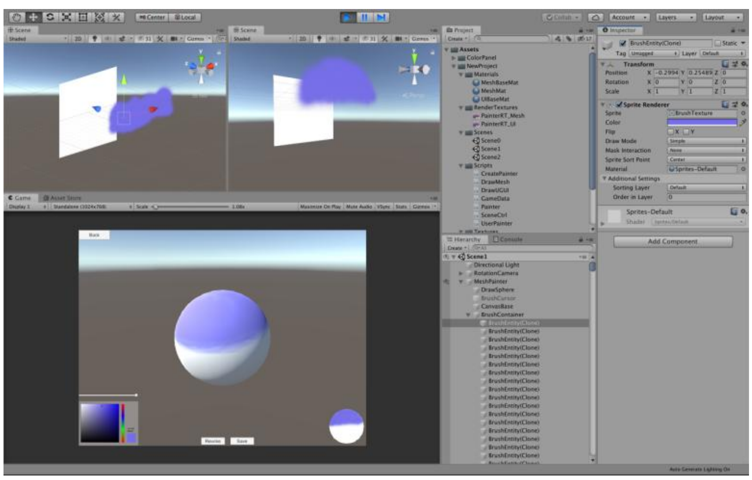

Figure 3. Created brush sprites for marking sphere.

When Sprites are generated, the user can use the *Revoke* button to cancel the current marking regions by deleting the child object under *BrushContainer* and then recreate the new relation. The *Save* button is for merging spawned brushes to base textures which are *MeshBaseMat (instance)* and *UIBaseMat (instance)*. After saving, it will clear marks before (Figure 4).

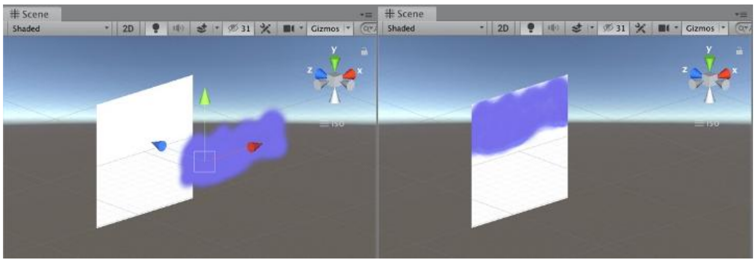

Figure 4. Unsaved texture (left) and saved texture (right).

In order to saving the relation data, once the marked parts of the 3D model and image are saved in texture, the coordinate data of those will be temporarily stored in *Game data*, which the end user will already be able to paint images with these relations.

In the User mode, after the end user selects the colour and fills in the image, the image and 3D model both have new textures with that colour. Figure 5 shows after colour the top of the circle image, the colour of top area of sphere is also be changed.

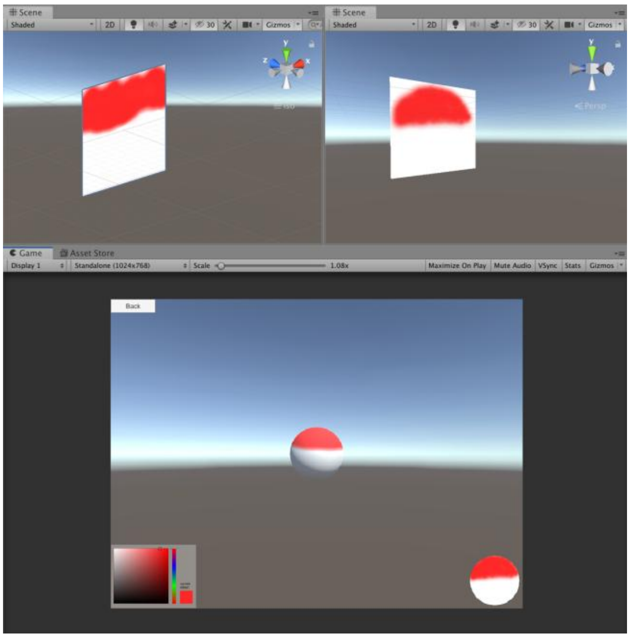

Figure 5. Filling red in User mode, 3D texture map (top left), 2D texture map (top right), The effect in the scene (below).

The biggest and first advantage of this method is that it can be applied to any model in which the UV map is a mesh. It will change the workflow for creating such 2D to 3D model mappings as establishing the correspondence in a very simple way for the creator. Secondly, the correspondence is much finer than the traditional method, but it depends on the size of the brush and the distance variable. Thirdly, there is no limit to the size and number of correspondence areas in the relationship-building process, it is completely free to be created by the creator. The limitation is still not perfect for fine structures such as eyes, and the user cannot change the colour of an area after filling it.

3.5. Detailed Design in Augmented Reality

3.5.1. The Front View Image

In order to achieve the aim of this project, I changed the model in the detail design to a more complex sculptural model, which is the bust of the Zeus model with a mesh UV map. The front view in the prototype will need to be turned into a black and white line drawing for created and end users building and utilizing the correspondences. The front view of the sculpture has a lot of lines representing the different areas of the sculpture. To solve this problem, I extracted the lines from the black and white line drawing as a PNG file so that the rest parts are transparent. In the new image, I set it to the source image. Eventually, after adding the lines, I got the ideal front-view image of the bust.

3.5.2. Image Trager

Images are detected based on natural features that are extracted from the image target and then compared at run time with features in the live camera image. To create a trackable that is accurately detected, Images need to have rich detail, and good contrast, avoid repetitive patterns and featureless areas, and formatting must be 8- or 24-bit PNG and JPG formats; less than 2 MB in size; JPGs must be RGB or greyscale (no CMYK). I use this black-and-white line drawing as the image target (Figure 6).

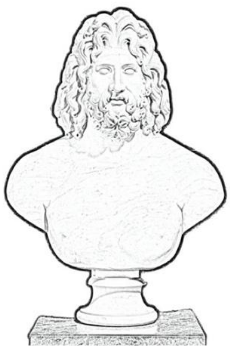

Figure 6. Image target.

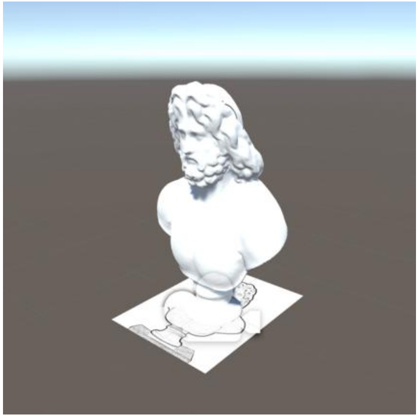

Figure 7. Zeus bust on image target in Unity.

For the setting of *ImageTarget*, I place the model vertically on the image target allowing the user to interact with the model by simply placing the image target in a horizontal position when running the program (Figure 7).

3.5.3. AR environment Interface

In order to be able to test it on an iPad Pro, I set the screen to 1149*843, which will fit the screen. The basic components of Interfaced consist of a ColourPanel, buttons, a black and white line drawing, and a 3D model with the same layout as in the prototype (Figure 8). The user flow of interaction in the AR system is also identical to that of the prototype, except that the model rotation can follow the rotation of the camera angle of view.

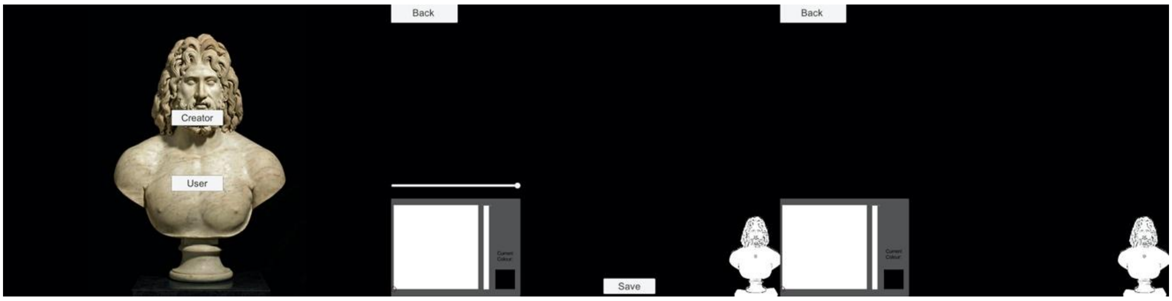

Figure 8. Interface of Scene0 (left), Scene1 (middle), and Scene2 (right).

So far, all the set up and build of the prototype are complete. The next step is to install it on the iPad pro for testing.

4. Testing

The testing chapter will design the test case as an independent module for each part of system, and each module will be subdivided into tests for each component. A portion of the concept prototype testing including the impact of the number of brushes on the system and detection of distance variable (f value) will be on a laptop, and the AR prototype portion will be on an iPad. The technical specifications of the testing device will be described in detail below.

4.1. Testing Devices

This project used Apple MacBook Pro 2020 and Apple iPad Pro 12.9-inch with 2nd generation Apple pencil for testing the performance of the prototype. The processor of the Apple MacBook Pro 2020 is a 2.0GHz 10th-generation quad-core Intel Core i5 and 16GB of 3733MHz LPDDR4X onboard memory [35]. The technical specifications of the iPad are Apple A12Z Bionic chip with 64-bit architecture embedded M12 coprocessor. The rear camera with wide 12MP and ultra-wide 10MP, and the screen is a 2732x2048-pixel resolution at 264 pixels per inch (PPI) with LED-backlit Multi-Touch display and ProMotion technology with refresh rates of up to 120Hz [36].

4.2. Brush Sprites

When the creator uses brushes, no matter how many brushes sprites are generated before the user presses the Save button, all sprites are not lost until the system crashes, which means that reaching a certain number of sprites will cause the system to run slowly. In extreme tests, it is generated more than 6000 sprites of brushes in 4 minutes, and the system still runs smoothly, but if more than 3000 brush sprites are generated in a single area, the system will crash. So, it must be saved before that to reduce the cache pressure.

4.3. Distance Variable

To test the distance variable (f value), I chose 11 values from 0.1 to 1000, and the goal of the test is to find numbers that I could just colour without bias. After the test, I find that it is difficult to colouring directly at 0.1 to 1f, although this value range creates a more exact match, and allows accurate colouring even two areas are closer together. In 3f to 5f range, it cannot be coloured directly on the matching area every time. It is very easy to colour the matching areas from 8f. After 50f, if more than one matching area is created close to each other, the colouring area will be biased resulting in colouring errors (Table 1). In the end, I chose 8f as the standard f value for the prototype.

4.4. Running Performance in Augmented Reality

For creating the mapping, the creator could use colours to mark areas of the bust model and image by adjusting the size of the brush. Then, they could rotate the camera or image target to get a different view of the model (Figure 9). While the program is running, the creator could temporarily remove the iPad's camera from the image target and then get the bust model by aligning the camera with the image target in any direction without losing the previous marker information.

Figure 9. Testing marking in AR system.

For colouring, a standard distance of 8f is viable for this statue filling colour and there is no bias in the colouring process (Figure 10). The end users can rotate the camera around the image target while the system tracks it to see the coloured bust at any angle. If the model is lost from the screen, they can get the coloured bust by putting the camera refacing to the image target again. A discussion of the results and limitations of the test is presented in the next chapter.

Figure 10. Testing filling colour in AR system.

5. Discussion and Future Work

As a result of the tests above, there are still some limitations in the prototype. First, in the interface design, the model's front view area is too small which makes it difficult to create areas and fills using brushes, the size of the brushes needs to be adjusted more appropriately to the interface design. Second, in the process of building relationships, although losing the model on the screen will not lose the mapping information, holding the iPad for a long time will be very tiring for the users, the interactive flow and the presentation of the model need to be iteratively optimized. Third, it is not friendly that the end user can't change the colour again after colouring, which should be a function to be added in the future.

From a wide perspective, there are two areas of work that could be undertaken in the future, including building a database and connecting 3D prints. The database is a system for organizing, storing, and managing data according to data structures. One of the ways in which future applications will work is to connect to the database, use this connection to store the data generated by the application, and then colour the model using the mapping relationships established by the creator within the database once the user is connected to the network.

3D printing is a type of rapid prototyping technology in which a computer-designed three-dimensional digital model is broken down into several layers of flat slices, and then a 3D printer stacks the different bondable materials layer by layer in a slicing pattern to eventually build up a complete object. The ultimate future presentation of this project could be a 3D printed object as the final output, with the foreseeable future being the use of coloured texture information to generate new model data that can be exported as a solid coloured model via a pre-programmed 3D printing device. If features such as using more advanced methods to improve the system's mapping relationships, adding texture maps to compensate for model detail and linking to more advanced large-scale 3D printers working in conjunction, which could provide a cheaper and more convenient option for the museum's restoration efforts and reconnecting to the real world. There has been a case of painted models being 3D printed, but there are still some 3D printing material limitations that may cause the colours to darken or become fragile [37].

6. Conclusion

This project implements the method used marker colours to establish correspondence between rendered textures of the image and the 3D model. Although it temporarily cannot to add the original texture map and change the filled colour, this method can handle any model with UV mapping properly, which reduces the large amount of workload of modelling for the creator to establish correspondence. Such correspondences support many-to-many relations, and there is no limit to the number of them.

In order to achieve the aim of simpler mapping creation, I chose the render texture mapping concept and eventually tested the Zeus bust model in the AR environment. The test results showed that the idea of pointing to the UV map, marking on render textures, and saving relations involved in the concept of render texture mapping meet the project aim, and the tests discover and then solve some of the problems that appeared in the prototype including the number of brushes sprites, distance detection, setting of model front view image. Three major limitations and future work are 1. image size needs to be adjusted to fit the brush size and layout; 2. tiredness during utilizing requires iterative optimization of the interaction process using additional user data and device selection; 3. the end user cannot yet change the colour after filling in the region. The broad future expectation is that the program can utilise the database to provide end users with mapped data produced by the creator, even each user can create and use the corresponding data as both creator and end user, and finally, by connecting to advanced 3D printers to produce physical models.

Acknowledgments

Thank you for Associate Professor Paul Tennent, your guidance is really helpful on my project. You give me more inspiration when we discuss the project during every online meeting. Thank you for your work, I really enjoyed discussing with you.

References

[1] Clark, A.; Dünser, A.; Grasset, R. An interactive augmented reality coloring book. *2012 IEEE Symposium on 3D User Interfaces* 2012, 7-10. https://doi.org/10.1109/3DUI.2012.6184168.

[2] Magnenat, S.; Ngo, D.T.; Zünd, F.; Ryffel, M.; Noris, G.; Rothlin, G; Marra, A.; Nitti, M.; Fua, P.; Gross, M.; Sumner, R.W. Live texturing of augmented reality characters from colored drawings. *IEEE transactions on visualization and computer graphics* 2015, *21*, 1201-1210. https://doi.org/10.1109/TVCG.2015.2459871.

[3] Lee, Y. AR coloring jigsaw puzzles with texture extraction and auto-UV mapping algorithm. *International Conference on Distributed, Ambient, and Pervasive Interactions* 2015, *9189*,182-187. https://doi.org/10.1007/978-3-319-20804-6_17.

[4] Panzanelli, R.; Schmidt, E.D.; Lapatin,K, A.; Brinkmann, V.; Ostergaard, J.S.; Collareta, M.; Potts, A. *The color of life: polychromy in sculpture from antiquity to the present*, 1st ed.; Getty Research Institute: Los Angeles, United States, 2008; pp. 18–39. https://www.amazon.com/Color-Life-Polychromy-Sculpture-Antiquity/dp/0892369175#customerReviews

[5] Bradley, M. The importance of colour on ancient marble sculpture. *Art history* 2009, *32*, 427-457. https://doi.org/10.1111/j.1467-8365.2009.00666.x

[6] Skovmøller, A.; Brøns, C.; Sargent, M. L. The polychromy of the sculpted garments on the statue of Gaius Fundilius Doctus in the Ny Carlsberg Glyptotek. *Polychromy in Ancient Sculpture and Architecture* 2018, 93-100. https://amis.ku.dk/people/?pure=en%2Fpublications%2Fthe-polychromy-of-the-sculpted-garments-on-the-statue-of-gaius-fundilius-doctus-in-the-ny-carlsberg-glyptotek(87ccd98f-9592-43c4-88cf-a7339bf906c8).html.

[7] Gods in Color. Available online: https://buntegoetter.liebieghaus.de/en/ (accessed on 07 September 2022).

[8] Color the Peplos Kore: http://www.yanniskoutsoukos.com/en/peploforos (accessed on 07 September 2022).

[9] Archaic Colors: Acropolis Museum. Available online: https://theacropolismuseum.gr/en/exhibition-programs/archaic-colours (accessed on 07 September 2022).

[10] Feng, L.; Yang, X.; Xiao, S. Magictoon: A 2d-to-3d creative cartoon modeling system with mobile ar. *2017 IEEE Virtual Reality (VR)* 2017, 195-204. https://doi.org/10.1109/VR.2017.7892247.

[11] Norraji, M. F.; Sunar, M. S. wARna—Mobile-based augmented reality colouring book. *2015 4th International Conference on Interactive Digital Media (ICIDM)* 2015, 1-4. https://doi.org/10.1109/IC3e.2018.8632639.

[12] Hanrahan, P.; Haeberli, P. Direct WYSIWYG painting and texturing on 3D shapes: (An error occurred during the printing of this article that reversed the print order of pages 118 and 119. While we have corrected the sort order of the 2 pages in the DL, the PDF did not allow us to repaginate the 2 pages.). *ACM SIGGRAPH computer graphics* 1990, *24*, 215-223. https://doi.org/10.1145/97880.97903.

[13] Adams, B.; Wicke, M.; Dutré, P.; Gross, M.; H.; Pauly, M.; Teschner, M. Interactive 3D Painting on Point-Sampled Objects. *PBG* 2004, 57-66. https://doi.org/10.2312/SPBG/SPBG04/057-066.

[14] Williams, L. 3D paint. *SIGGRAPH Comput. Graph* 1990, *24*, 225-233. https://doi.org/10.1145/91394.91450.

[15] Igarashi, T.; Cosgrove, D. Adaptive unwrapping for interactive texture painting. *Proceedings of the 2001 symposium on Interactive 3D graphics* 2001, 209-216. https://doi.org/10.1145/91385.91450 .

[16] Lu, S.; Mok, P.Y.; Jin, X. A new design concept: 3D to 2D textile pattern design for garments. *Comput. Aided Des.* 2017, *89*, 35-49. https://doi.org/10.1016/j.cad.2017.03.002.

[17] Zilles, C. B.; Salisbury, J. K. A constraint-based god-object method for haptic display. *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots* 1995, *3*, 146-151. https://doi.org/10.1109/IROS.1995.525876.

[18] Basdogan, C.; Srinivasan, M. A. Haptic rendering in virtual environments. In *Handbook of virtual environments*, 1st ed.; Hale, K. S., Stanney, K. M., Eds.; CRC Press: Boca Rato, United State, 2002; pp. 117-134. https://www.amazon.com/Handbook-Virtual-Environments-Implementation-Applications/dp/080583270X.

[19] Theoktisto, V.; Fairén, M.; Navazo, I.; Monclús, E. Rendering detailed haptic textures. *Second Workshop in Virtual Reality Interactions and Physical Simulations (VRIPHYS 05)* 2005, 16-23. https://www.academia.edu/50205980/Rendering_detailed_haptic_textures.

[20] Wakita, W.; Murakami, K.; Ido, S. A texture-based haptic model design with 3D brush. *Proc. 18th International Conference on Artificial Reality and Telexistence (ICAT2008*) 2008, 51-56. https://icat.vrsj.org/ICAT2008_Proceedings/Papers/PS2_2.pdf.

[21] Manipulation of Render Texture. Available online: https://docs.unity3d.com/2019.4/Documentation/Manual/class-RenderTexture.html (accessed on 07 September 2022).

[22] Description of Render Texture. Available online: https://docs.unity3d.com/ScriptReference/RenderTexture.html (accessed on 07 September 2022).

[23] RealTime Painting: Tutorials: Unity Asset Store. Available online: from https://assetstore.unity.com/packages/templates/tutorials/realtime-painting-33506 (accessed on 07 September 2022).

[24] Azuma, R. T. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments* 1997, *6*, 355-385. https://doi.org/10.1162/pres.1997.6.4.355.

[25] Craig, A. B. *Understanding augmented reality: Concepts and applications*, 1st ed.; Morgan Kaufmann: Burlington, United States, 2013; pp. 1-36. https://doi.org/10.1016/B978-0-240-82408-6.00001-1.

[26] QuiverVision. Available online: https://quivervision.com/ (accessed on 07 September 2022).

[27] Lee, Y.; Choi, J. Texture Extraction from Video and Image Warping for AR Coloring Book*. *Computer Science and its Applications. Lecture Notes in Electrical Engineering* 2015, *330*, 361-362. https://doi.org/10.1007/978-3-662-45402-2_55.

[28] Dinosaur Sketchbook. VUiDEA. Available online: http://www.vuidea.com/ (accessed on 07 September 2022).

[29] Cho, K.; Kim, H.; Lee, Y. Augmented reality coloring book with transitional user interface. *Indian Journal of Science and Technology* 2016, *9*, 1-5. https://doi.org/10.17485/ijst/2016/v9i20/94692.

[30] Mokhtar, M. K.; Mohamed, F.; Sunar, M. S.; Arshad,M. A. M. ; Mohd Sidik, M. K. Development of Mobile-Based Augmented Reality Colouring for Preschool Learning. *IEEE Conference on e-Learning, e-Management and e-Services (IC3e)* 2018, 11-16. https://doi.org/10.1109/IC3e.2018.8632639.

[31] Olsen, L.; Samavati, F.; Jorge, J. NaturaSketch: Modeling from Images and Natural Sketches. *IEEE Computer Graphics and Applications* 2011, *31*, 24-34. https://doi.org/10.1109/MCG.2011.84.

[32] Colossal marble bust of Zeus: British Museum. (n.d.). Available online: https://www.britishmuseum.org/collection/object/G_1836-0709-1 (accessed on 07 September 2022).

[33] Colossal marble bust of Zeus 3D model. Available online: https://sketchfab.com/3d-models/colossal-marble-bust-of-zeus-9246efe1279543a784a08be62e6b96a8 (accessed on 07 September 2022).

[34] Explore the possibilities of RealityCapture. Available online: http://www.capturingreality.com/ (accessed on 07 September 2022).

[35] MacBook Pro 13-inch - Technical Specifications. Available online: https://www.apple.com/uk/macbook-pro-13/specs/ (accessed on 07 September 2022).

[36] iPad Pro - Technical Specifications. Available online: https://www.apple.com/uk/ipad-pro/specs/ (accessed on 07 September 2022).

[37] Mccarthy, J.; Sebo, E.; Wilkinson, B.; Sheehan, F. Open Workflows for Polychromatic Reconstruction of Historical Sculptural Monuments in 3D. *J. Comput. Cult. Herit* 2020, *13*, 1-16. https://doi.org/10.1145/3386314。

[38] The paper is published in a preprint version with different structure and acknowledgement in arXiv. Available online: https://doi.org/10.48550/arXiv.2202.04259 (accessed on 07 September 2022).