City University of Hong Kong
School of Creative Media: BScCM
Department of Computer Science

BScCM Final Year Project 2020 – 2021
Phase II Final Report

<< Group Name: NNN ; Group No.: 26 >>

<< CREATOR >>

Student Name/EID : ZHENG Kexin / kexizheng3

Student Name/EID :

Student Name/EID :

Programme Code : SM4602

Supervisors SCM : FU Hongbo
CS : WANG Shiqi

Date : 09 May 2021

Table of Content

Introduction

People use sketches to express their creativity and illustrate their ideas. It is an entertaining and casual way to transform their thoughts into visual representations. In recent years, there are serval applications and research that combine the 2D-to-3D sketch-based modeling system with AR technology to create AR scenes, such as Rakugaki AR [16], MagicToon [4], and Wonder Painter [2]. These applications capture the user's sketches using a mobile camera and transform them into 3D models in real-time. Some even support simple on-screen interactions with the models.

However, this kind of system requires the sketch to be physically presented, which puts constraints on the user scenarios. In addition, the animations of those applications are usually predefined, which also limits the user's imagination. The CREATOR proposes an AR mobile application that can transform the user's digital sketches on the screen into 3D models and animate them with short keyframe animation. The user can draw with their finger on the screen to create different sketches and the AR software will analyse the sketches and turns them from 2D drawings into 3D models with our meshing algorithm. The user can deform or animate the model with our deformation & animation system and create moving path in the 3D space with the AR system. With integrated AR plane detection and AR occlusion method, the user can create visual effects that are closely related to the real environment.

With this system, 3D models can be created from simple 2D digital sketches, without other physical material reference or environment constraints. The AR system forms a bridge between the physical world and the created 3D model, which enhances the immersive experience for the user.

Objective

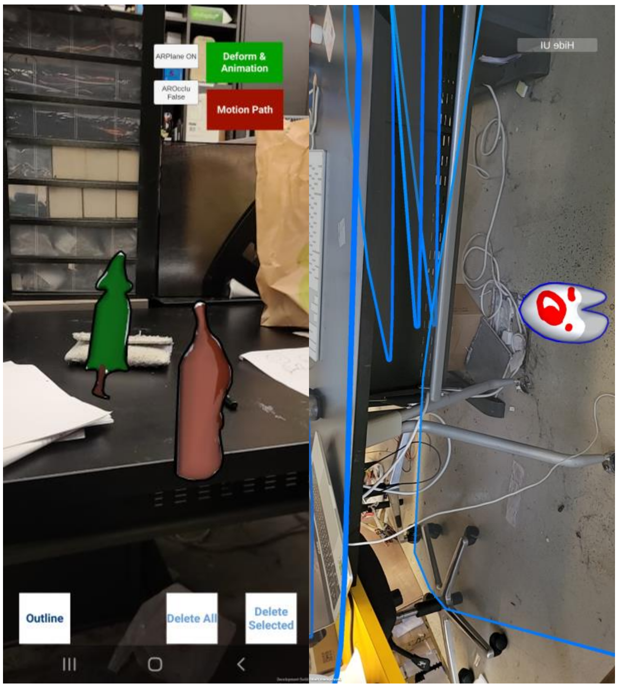
• Design a system that can perform 2D-to-3D modeling based on digital sketches on the mobile phone.

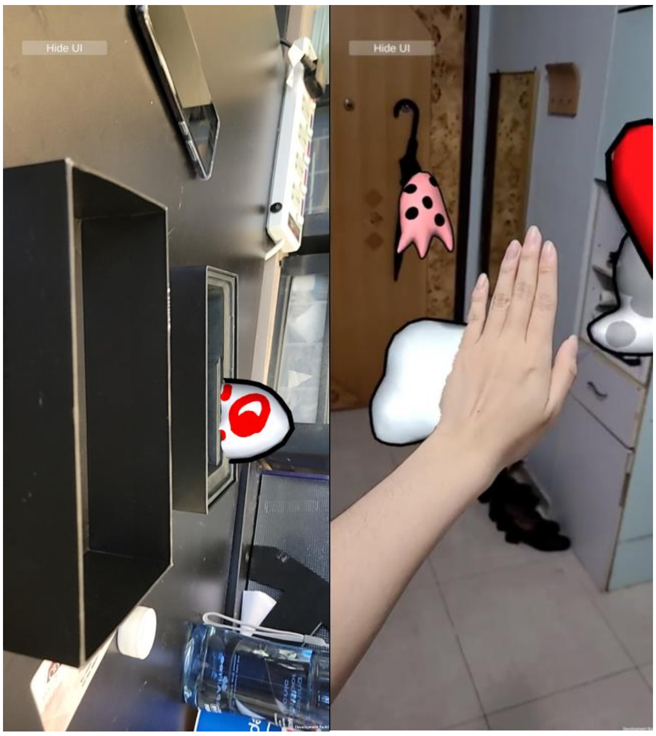
• Encourage people deform and animate model on mobile phone.

• Encourage people to create their own AR scene that closely related to the real envrionment.

Deliverables

Screenshots





Work Plan

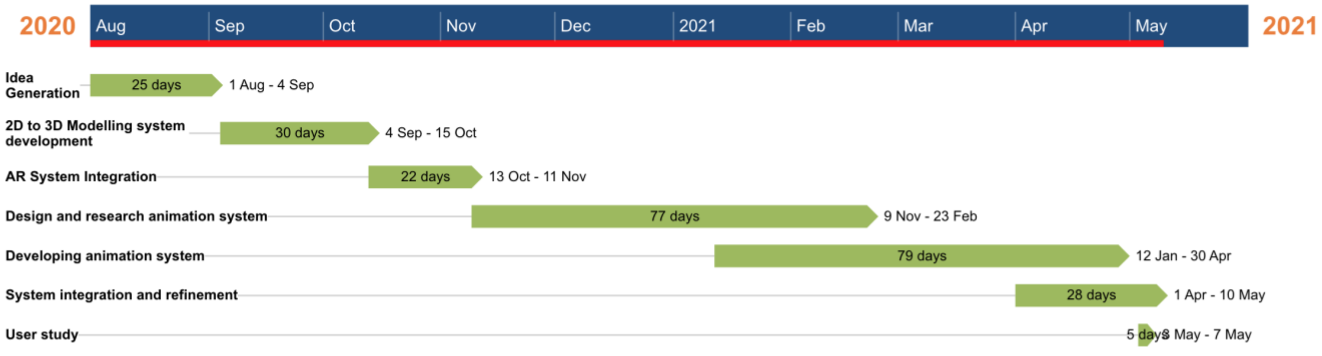


Figure 1 Project Plan

Background Research

1. Sketch-based Modelling Systems

• MagicToon [4]

This paper proposes an interactive system that can create 3D models based on physical sketches. The user can animate the model with a given skeleton.

• Wonder Painter [2]

Wonder Painter is a mobile application that can turn physical sketches/objects into simple 3D models. The user can create interactive stories with the models. The animation effect of Wonder Painter is predefined by the system.

• NaturaSketch [12]

In NaturaSketch, the user can create sketch-based 3D models with simple annotations. Following the annotations, the system will create hole and extrusion part on the model.

• Rakugaki AR [16]

The Rakugaki AR will turn physical sketches into paper-like 3D models. The user can perform simple on-screen interaction with the created models.

	MagicToon	WonderPainter	NaturaSketch	RakugakiAR	CREATOR
MODELING					
2D-to-3D modeling	Yes	Yes	Yes	Yes	Yes
Complex Shape	No	No	Yes	No	No
INTERACTION					
On-screen Interaction	No	No	No	Yes	Yes
Interactive Story	No	Yes	No	No	No
ANIMATION					
Mesh Deformation	Yes	Yes	No	Yes	Yes
Skeleton-based Animation	Yes	No	No	No	Yes
Keyframe Animation	No	No	No	No	Yes

Table 1 Comparison Table of CREATOR and the existing applications/research

2. Mesh deformation & Animation

Interactive mesh deformation based on arbitrarily defined handles is an interesting and challenging topic in the field of Computer Graphics. For 2D meshes, an early approach is proposed by Igarashi [6] which based on the optimization of local rigid transformations and preservation of differential coordinates. The idea of preserving local rigidity is further developed in 3D space to create well-defined model manipulation. [13] However, these approaches are not good choices for real-time mesh deformation on mobile devices because they need to compute optimization result at pose time, which could be very slow.

Computing the vertices transformation based on weighted bones are less computationally expensive and are widely used in the industry, such as linear blend skinning [9] and dual quaternions skinning [8]. These methods generally based on predefined bone weights that are painted by the creator or compute automatically by algorithms. There are various methods to compute skinning weights automatically. Currently, a notable one that achieves the best performance is the bounded biharmonic weight method, which based on minimizing the Laplacian weight. [7] The method is initially developed for 2D meshes but can also be used for 3D mesh deformation. It is also proved to achieve good performance for 2D shape manipulation on mobile devices [10]. However, this method requires discretization of the volume for 3D meshes, which could be very slow in general [5]. Another popular algorithm to compute the weight is the bone heat diffusion method [1]. In comparison with the bounded biharmonic method [7], it has some disadvantages on preserving features. However, it is relatively easy to implement and requires less complex matrix computation

3. AR environment

The development of DepthLab [3] provides more well-defined environment information that is useful for AR development on Android devices. It transfers the data captured by the camera into different categories so that the virtual objects can interact with the environment more naturally.

Framework

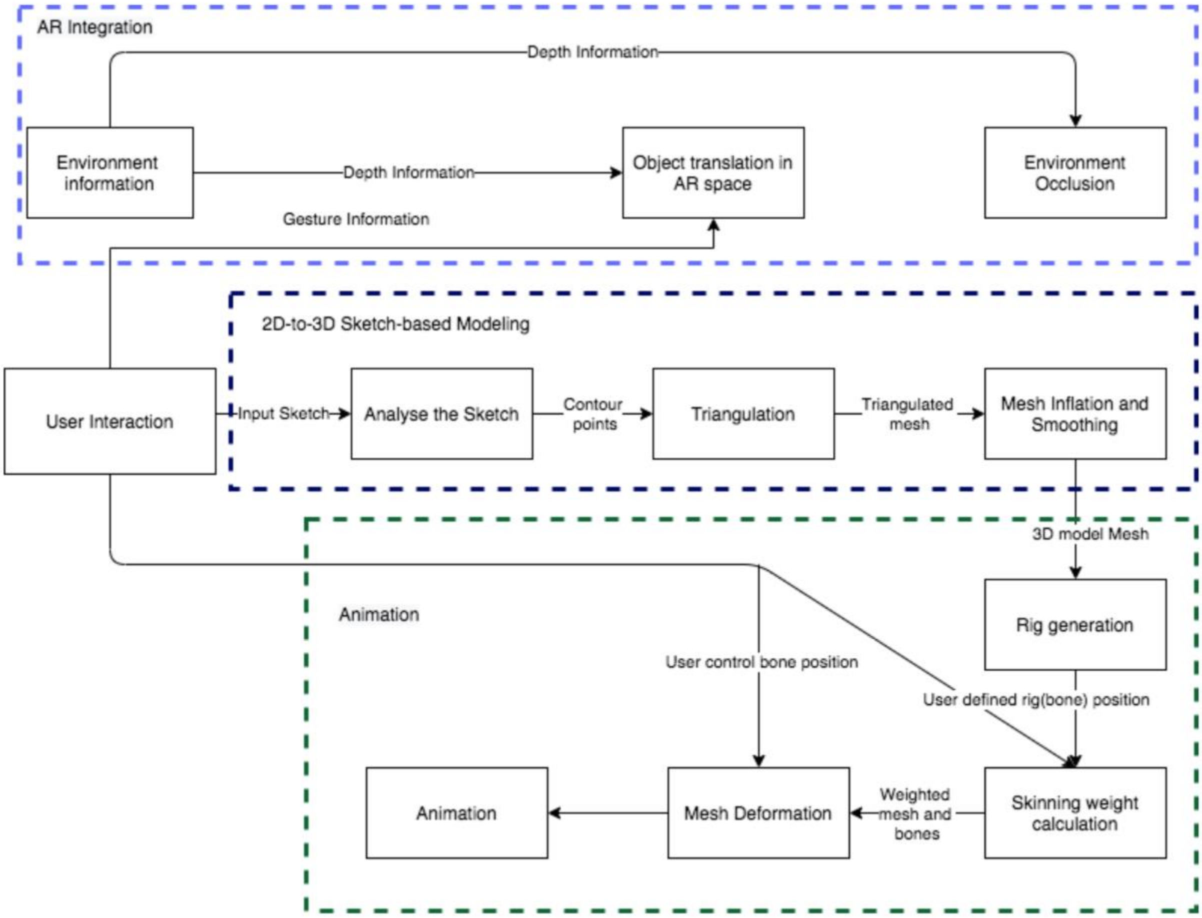


Figure 2 System framework

Methodology

1. Design

   The general idea of CREATOR is to develop an application that enable the user to create their own 3D AR scenes with simple sketches.

   The system will transform the user's input sketches into 3D model and enable them to interact with those models. To analyze the user's sketches, this application will use a special algorithm to turn the 2D drawings into 3D models.

   A simple rigging method will be provided to enable the user to create skeleton on the generation model for mesh deformation. Combining with key frame animation system, the user will be able to create short animation easily on the generated model.

   Our system will make use of the advantages of augmented reality and use data such as device's position and orientation to create meaningful interaction for the model. Other environment data will be processed by AR Core and use as visual effect reference.

2. Technology

   2.1. Unity

   Unity Engine is a powerful development platform that support software development for different devices. It is easy an intuitive to construct a good project structure. It also have powerful graphical programming support, which is an wonderful choice for this project.

   2.2. AR Core

   AR Core service provided by Google is the basis for Augmented Reality applications on android devices. It have many powerful feature such as augmented image, point cloud detection and plane detection. With the newly released Depth API, AR Core will be able to get the depth data from the environment and create visual effect that closely related to the real environment, such as object occlusion.

Implementation

The current application consists of three main systems that provides different functions.

- 2D-to-3D Modeling System
- Deformation and Animation System
- AR System

2D – to – 3D Modeling System

1. Contour Analysis

   The winding algorithm[17] is implemented to detect whether an input sketch polygon point is inside or outside the previous polygon shape. Every time when the user input a new sketch stroke, a new polygon shape will be generated for analysis. The final contour will erase the edge points that located inside the polygon shape for the triangulation processing.

2. Delaunay Triangulation & Ruppert's Mesh Refinement Algorithm

   The Delaunay triangulation is performed to segment the 2D sketch into triangle regions. It maximizes the minimum angle of all the angles of the triangles in the triangulation. After performing the Delaunay triangulation, Ruppert's algorithm is further applied to add additional vertices into the mesh until minimum angle constraints are met. [14]

3. Mesh Inflation & Laplacian smoothing

   Based on the triangulated 2D sketch, the system performs mesh inflation according to the distance map of the sketch. It transforms each vertex's z position based on the distance from that vertex to the nearest edge point. This method is also used by Natura Sketch [12] and MagicToon [4]. After the mesh inflation, the system symmetrizes all the non-edge vertices in order to create a complete 3D model. Using distance map to inflate mesh might cause some sharpness in the meshes. As a result, Laplacian smoothing is applied to change the position of the vertices in order to create a smooth surface. [15]

4. Texture Mapping

   For each vertex, the system calculates its UV coordinate by finding the relative x position and y position of that vertex to the bounding rectangle of the whole mesh. The texture is generated with the smallest area that contains all the coloring pixels.

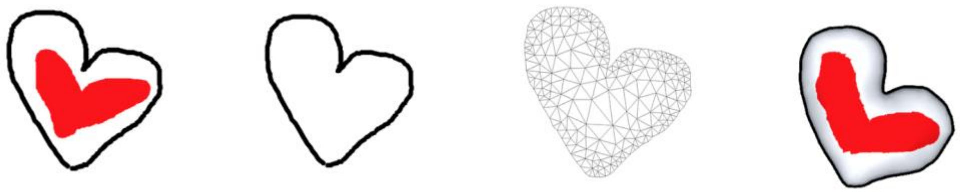


Figure 3 From left to right: Input 2D sketch, 2D sketch contour, triangulation result, final 3D model

Deformation and Animation System

A complete model with rigged skeleton is presented with the following model group structure. The 'ModelGroup' controls the animation and transform of the whole model, while the 'ModelMeshObj' contains the mesh of the model. The 'SphereHandel' is movable component that serves as the deformation joints for the model. Each object under the structure is attached with corresponding script for different functions.

The system will automatically generated the skeleton and calculate the corresponding weight with our special algorithm inspired by Niskanen[5]

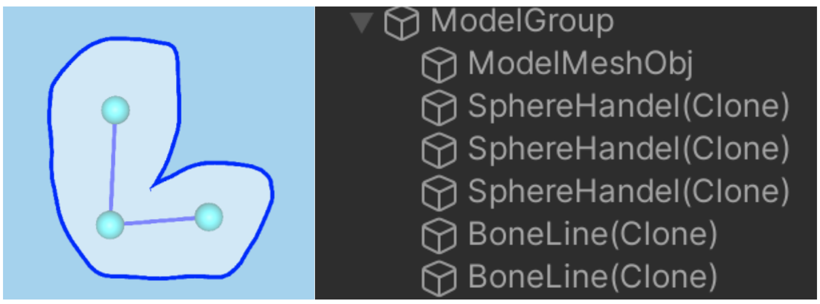


Figure 4 Model with generated skeleton (Left) Model Group Structure (Right)

1. Auto-rigging with Sphere Mesh Approximation

Our rigging Sphere Mesh is based on the shape approximation method proposed by Thiery et al. [19] It use spheres to approximate a planar surface. By recursively collapsing the connected edges between spheres with minimal error, an simplified sphere mesh will be generated. The Spherical Quadric Error Metrics(SQEM) for an sphere *s* at position *p* for an target plane with normal *n* is defined as

$$SQEM(s) = Q(s) = \frac{1}{2}s^T * A * s - b^T * s + c$$

Where

$$A = 2 * \begin{bmatrix} n * n^t & n \\ n^t & 1 \end{bmatrix},$$

$$b = 2 * (n^t * p) * \begin{bmatrix} n \\ 1 \end{bmatrix} ,$$

$$\text{and } c = (n * p)^2$$

In the initialization phase, each vertex *i* in the mesh will be assigned with an sphere *s* with radius of zero and the corresponding SQEM will be calculated by approximate each triangle planes t_j in the barycenteric cell of that vertex:

$$Q_{i_{(s)}} = \sum_{t_j \in T(v_i)} \frac{area()}{3} * Q(s, p_{t_j}, n_{t_j})$$

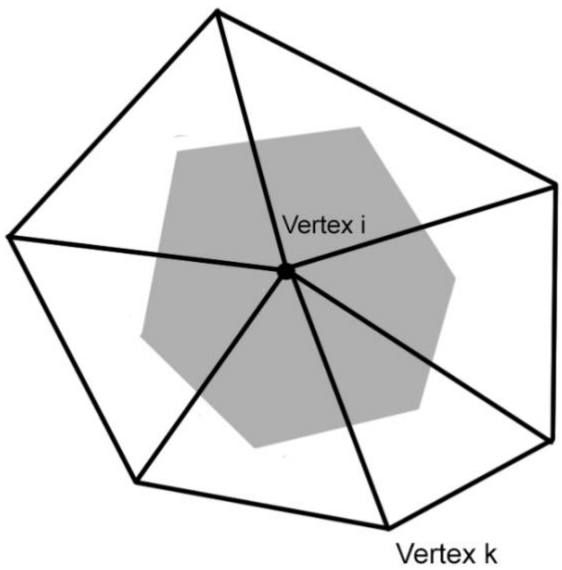


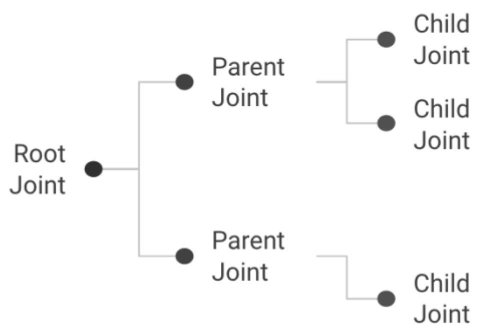
Figure 5 The barycentric cell of vertex i is highlighted in gray

The SQEM for an edge *ik* which connects vertex *i* and vertex *k* will be calculated with:

$$Q_{ik} = Q_{i_{(s)}} + Q_{k_{(s)}}$$

The system will use a priority queue to store the SQEM for each edges and recursively collapsing the one that has the minimum error. Collapsed sphere will be replaced with an new one until the system has reach the desired number of decimation, which is defined by the user as the number of bones they wanted.

Our system generate the skeleton partially based on the user's input. The user need to drag across the connected bones generated with sphere approximation and the system will automatically generate the bone structure. In our system we use the skeleton structure that consists of root joint, parent joints and child joints. The root joint is not transformable while the user can perform simple rotation on parent joint and translation and rotation on child joint.



2. Bone hear skinning weight calculation

Initially, the As-rigid-as-possible shape manipulation method [6] is chosen for the mesh deformation. However, in the implementation, it shows a very slow performance for real-time deformation on the triangulated sketch mesh. As a result, I turned to another direction by using the linear blending skinning method [9] with user-defined handles and automatically calculated weights. To calculate the weight, the initial choice is bounded biharmonic weight method [7], which possibly provides the result. However, the performance will be slow on a mobile device because it needs the 3D model to be volumetrically discretized before the weight calculation. As a result, the final project implemented the bone-heat skinning method [1]. It will automatically calculate weights for a given mesh and bone structure. It treats the 3D model as a heat-conducting body and calculates the weight of each bone separately by calculating the equilibrium temperature at each surface vertex.

The equilibrium for the bone I can be calculated with the following equation:

$$\frac{\partial w^i}{\partial t} = \Delta w^i + H(p^i - w^i) = 0$$

The equation can be written as:

$$-\Delta w^i + Hw^i = Hp^i$$

Where Δ represents the discrete surface Laplacian calculated with the cotangent formula[20] and p^i is the vector that has $p_j^i = \frac{1}{n}$ if a vertex j is controlled by n bones, including the bone i. H is a disfonal matrix with $H_{jj} = \frac{1}{d(j)^2}$, where d(j) us the distance from vertex j to the nearest bone.

3. Linear Blend Skinning

Linear blend skinning[9] is a deformation method that are widely used in the animation industry for its fast and simple matrix calculation. Even though it will introduce some occasional artefact, we still choose it as the deformation method for our system considering the matrix calculation complexity. The deformation matrix for a vertex i is defined as:

$$v_i' = \sum_{j}^{N} w_{ij} T_j v_i$$

Where w_{ij} is the weight of bone j for vertex I and T_j is the transformation matrix for bone j.

In our system, each skeleton joint will store the matrices for the later transformation matrix calculation. They are listed as follows:

- A: Parent joint transformation matrix relative to model center
- B: Joint transformation matrix relative to parent
- C: Joint Initial position matrix relative to parent
- D: Vertex position matrix relative to joint
- E: Vertex position matrix relative to model center

Each time the user moves a joint, a new transformation matrix B will be calculated and the old matrix will be replaced. To define the deformation of vertex *i*, we will calculate the following matrices:

The final transformation matrix for a vertex i is computed with

$$M(i) = A * b * C * D * E^{-1}$$

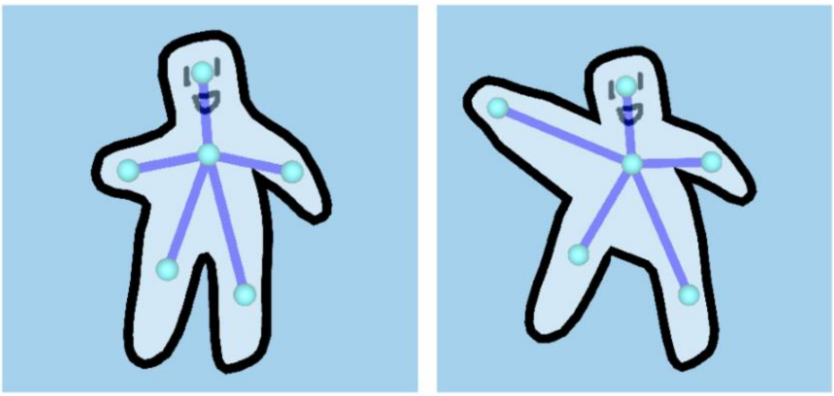


Figure 6 Original mesh with skeletons (Left) Deformed mesh (Right)

4. Keyframe Animation

The current system can create animation with framerate of 24, which means there will be 24 frames be presented at each second. The system provides 6 key positions for the user to define the model's deformation. The user can create both skeleton animation and normal scaling & rotation animation. Each time when the user defines a key, the system will store the corresponding data in the animation class and automatically calculate the frames between keys using sphere linear interpolation for natural animation effect.

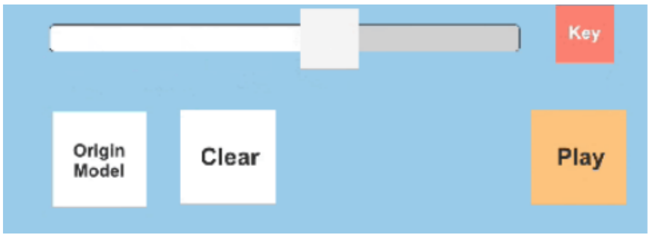


Figure 7 User Interface for keyframe animation

Our system uses two different classes for bone animation and normal deformation animation respectively. They has similar structure with key positions, keyframes and interpolation calculation functions. For bone animation, the recorded data at the key position will be the translation vector for rotation and quaternion for rotation. For normal deformation animation, the recorded data at the key position will be the rotation and scale of the whole model. To avoid large computation, our system will store optimized the mesh data when the bone animation in the first playing loop and retrieves it in the following loops.

AR System

1. Motion Path

   Motion path has been studied by many researcher for animation generation in recent years. An earlier one is the space-time sketching[21] that can use simple sketch lines to manipulation character animation. For the usage of Motion path in mixed reality, there are some recent developments such as the Spatial Motion Doodle introduced by Garcia et al.[22] that use VR controller to set the motion path for animation generation. And the AR Animator presented by Ye et al.[23], shows an interesting application that use mobile gesture and motion path to create animation that closely-related to the real environment. The projects shows that the movement of device is expressive and serve as an intuitive approach to create animation in the mixed reality environment.

   Therefore, we decided to adopt a similar method that uses the position of mobile phone as the 3D moving path for the generated model. Our system use the AR anchor to track the position of the starting point of the path in the space and the rest position on the moving track will be stored in our motion path class for later reference. To avoid the turbulence effect caused by sudden track loss and hand shaking, we use a simple mid-point smoothing method to smooth the curve after the user finish drawing a motion path.

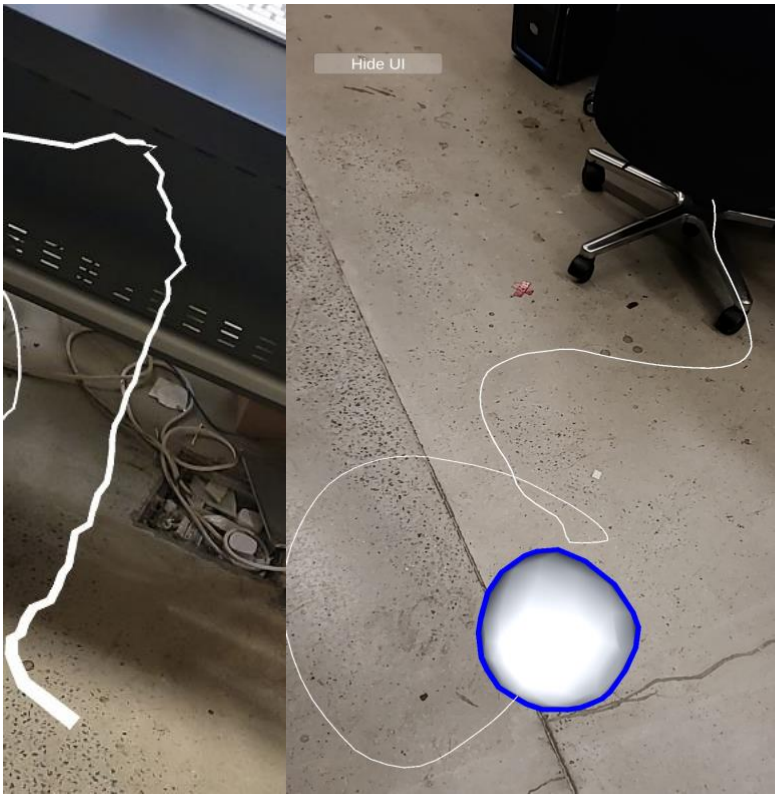


Figure 8 Mobile motion path without smoothing (Left) Generated model with motion path (Right)

2. AR Plane detection and AR Occlusion

   Our system integrated the AR Plane detection and AR Occlusion method provided by Google AR Core. The user can switch between different effect using our UI. The corresponding effect manager will be called when the user activated an AR effect.

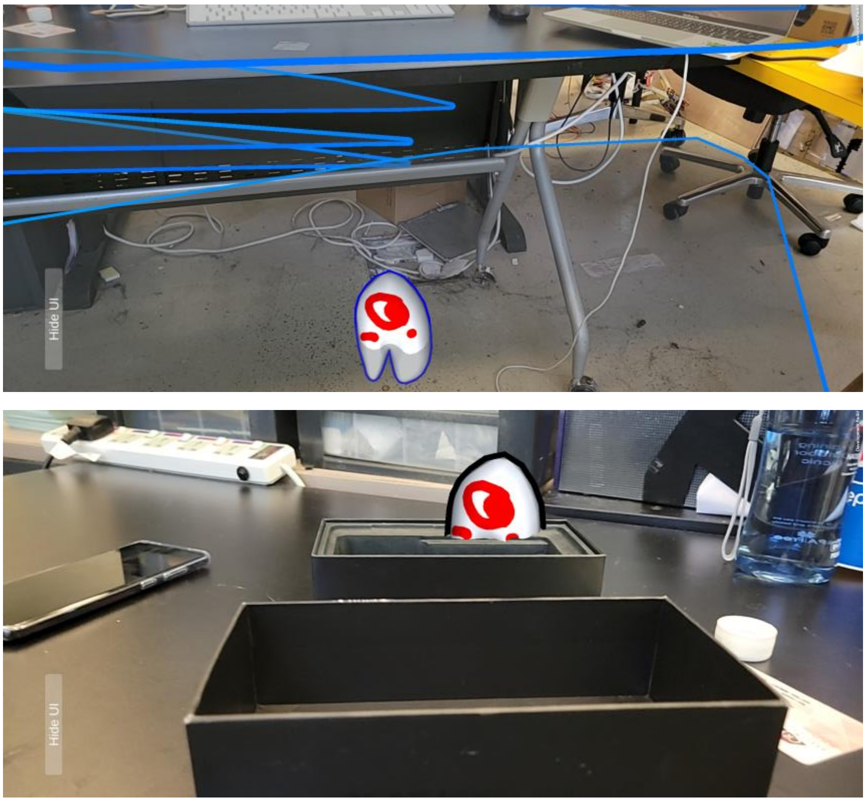


Figure 9 Model placed on AR Plane (Top) Model occluded with AR Occlusion (Bottom)

Usability Study

In order to evaluate the system , we ran a small-scale usability study with 8 participants, which includes 3 male and 4 female aging from 21 to 30. Half of them have previous experience in mobile AR application, with three quarters of them have 3D modelling or animation experience or majoring in creative related area.

Procedural

Each participants were given 10 minutes training on using the application to create model, key animation, and a complete AR scene. After that, they were asked to accomplish three exploration tasks for testing different functions of the system.

- T1: use CREATOR to model and deform 3 models
- T2: use CREATOR to create keyframe animations for 3 models
- T3: use CREATOR to create an AR scene that combining all the functions provided by the system.

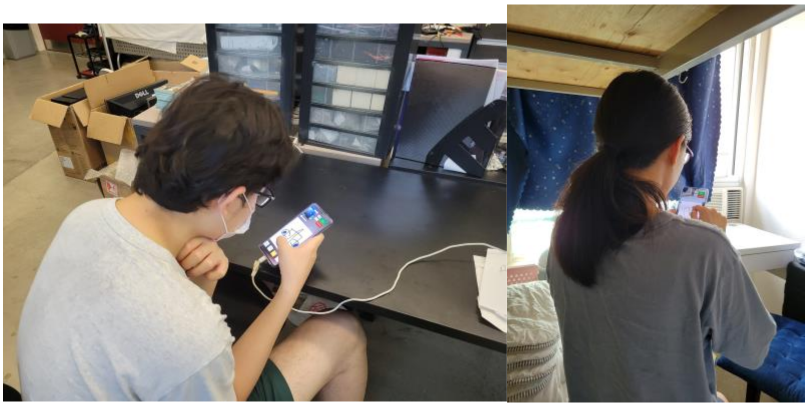


Figure 10 Participants are using the application

After finishing each task, the participants were asked to fill in a NASA Task Load Index[11] (NASA-TLX, 1 – very low to 5 = very high) to investigate on the workload for each task. At the end of the study, they need to fill in an System Usability Scale[18] (SUS, 1 = strongly disagree to 5 = strongly agree) form to evaluate the usability of the whole system.

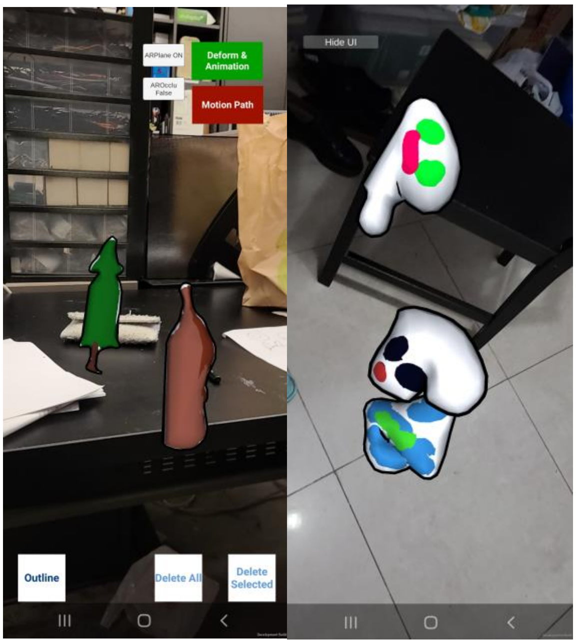


Figure 11 Models created by the participants

1.1. Result

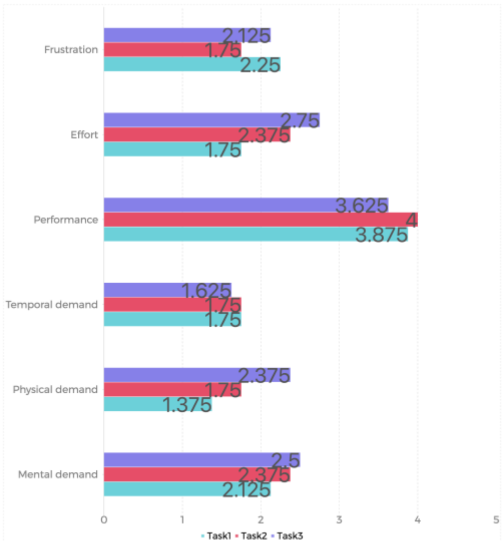


Figure 12 Mean values of NASA-TLX

From the NASA-TLX results, we found that for task 1 and task 2, the demands and effort that the participants experienced were low to moderate, while the performance score are relatively high. It shows that our system can enable the user to create their own model and animate them easily, and the effects are generally satisfying. For task 3, the mental demand and effort score has increased. It is possibly due to the task requires the participants to combine the functions that are also tested in the previous task, which increases the complexity. The performance score has decreased a little in comparison to the previous 2 task, possibly because of similar reason. The physical demand for task 3 shows a clear increase, possibly due to the motion path generation, which requires the participants to move the phone. The temporal demand for three tasks are low, which implies that our system is not very time-consuming. In conclusion, the NASA-TLX shows that our system can enable the users to generate AR scenes that are generally satisfying with moderate efforts.

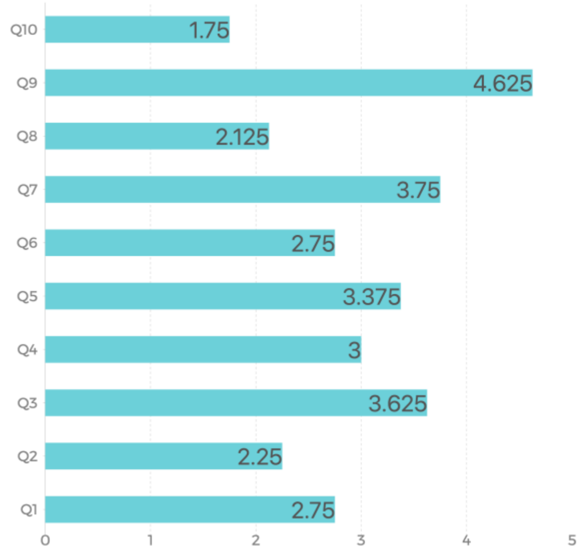


Figure 13 Mean values of SUS

The SUS form consists of the following 10 questions:

- Q1: I think I would like to use this tool frequently
- Q2: I found this system unnecessarily complex
- Q3: I found this system was easy to use
- Q4: I think that I would need the support of a technical person to be able to use this system
- Q5: I found the various functions in this system were well integrated
- Q6: I thought there was too much inconsistency in this system
- Q7: I would imagine that most people would learn to use this system very quickly
- Q8: I found the system very cumbersome to use
- Q9: I felt very confident using the system
- Q10: I needed to learn a lot of things before I could get going with this system

From the collected data for Q2, Q3, Q8, and Q10, we can conclude that our system is relatively easy to use for novice user and it does not requires previous knowledge. The result of Q9 also implies that the participants are satisfied with their performance when using our system. However, the overall SUS result shows that our system still has some issues. From the results for Q1, Q4 and Q5 and some user feedback, we have realize we need to design the system workflow in a more novice-user friendly style, such as providing tutorial stage or improving the UI design. We also need to reconsider the integration of different functions.

2. User Feedback

In the study, all participants were able to learn to use the system quickly and perform the task on their own after the training session. In general, they like the visual effect they created and are satisfied with the user study. They have provided useful feedback and improvement suggestion for possible future development.

A1: *"I would like to use this effect in my movie project, can it be used with video editing software on computer?"*

A4: *"I think the visual effect is cool."*

A5: *"The integration of AR occlusion effect is very interesting"*

A8: *"The generated model is very cute, I like the cartoonish style.*

Discussion

1. Limitation

   In the current system, the contour analysis generally based on polygon analysis, which might result in unwanted artefact when modelling. It might be addressed by using matrix system like computer vision to help better subtract the contour from the input sketches.

   The auto-rigging system using sphere approximation method provide an simple method for skeleton generation without too much user input. However, it also limit the position of the joint that the user can select from. Therefore, it might be better to improve the system by letting the user to freely move the bones using the result from sphere approximation as the reference positions. In addition, the current bone-heat weight calculation method is very time-consuming when performing on mobile phone because it need to calculate the nearest bone for each vertex.

   The deformation of the current system are mainly 2D, which limits the degree of freedom that the joints can be rotated around. Using linear blend skinning[9] also might cause mesh collapsing with performing rotation around joints.

2. Future Development

   The current animation supported by our system are simple with limited effects. It also does not involve any responses to human interaction. It could be an interesting development direction by combining human-related tracking technique, such as hand-tracking or face-tracking to add more interaction into the scene, and possibly generate trigger-based animation.

   The deformation & animation system still has many issues regarding the rigging. The current auto-rig method might cause some occasional artefacts or wrong calculation result. For future development, we want to polish the current system and improve its performance by adding editing features and fast matrix computation method.

Concluding Remarks

In conclusion, our system has demonstrate relative good performance on 2D-to-3D modelling and animation for AR scene creation. The current function are well-integrated and the system can generated satisfying visual effect for the users. However, there are still some problems that can be improved, such as the deformation mesh collapsing error and the slow weight computation time. The user study result shows that we need to put more emphasis on improving the usability of our system in the future. It is possible because we want to implement too many functions into our application that make the system a bit complex to use, and there are performance issues when combining all the functions together.

Summary

We presented an interactive system that allows novice user to create AR scenes from simple sketches on mobile devices. With simple inputs, our system will automatically generate 3D model with complete texture. We also designed an auto-rigging system that can rigged the generated model automatically with simple user inputs. We have implemented an interactive animation system that enable the user to create short key-frame animation on their models and further overlay them in the real environment. A possible implementation of our system is to be used as prototyping tool for simple visual effect generation or entertaining educational tool for the children.

We have also conduct a small-scale user study with 8 participants. The study result shows that our system is easy to use and are not time-consuming or laborious. However, it also implies some problem on our current system workflow and UI design. The user study provides very useful feedback and comment for the future development of our project.

Acknowledgements

I would like to express my sincere gratitude towards Prof. FU Hongbo and Dr. WANG Shiqi for being my advisor, who have given me many useful guidance and advice. Their comments on my project has helped me develop and refine my system. I was encouraged by their positive feedback, and motivated by their critical suggestions. Without their help, I will not be able to design and develop the whole system and integrate so many different functions together.

I also would like to thank Dr. ZHU Kening, for allowing me using the device in his laboratory. As my major advisor, he has given me many useful suggestion and introduced me to many great opportunities.

I also want to thank my friends WANG Keru, Lai Zhaoxuan, WU Keying, HAN Fei and the participants of my user study for giving me support and useful comments on the system design.

My profound thanks are given to my parent. I want to thank them for their unconditional support and encouragement throughout my life.

Reference List

[1] Baran, I., & Popović, J. (2007). Automatic rigging and animation of 3d characters. *ACM transactions on Graphics (TOG)*, 26(3), 72-es.

[2] Cao, X. (2018). Wonder painter: turn anything into animation. In *ACM SIGGRAPH 2018 Real-Time Live!* (pp. 1-1).

[3] Du, R., Turner, E., Dzitsiuk, M., Prasso, L., Duarte, I., Dourgarian, J., ... & Kim, D. (2020, October). DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (pp. 829-843).

[4] Feng, L., Yang, X., & Xiao, S. (2017, March). Magictoon: A 2d-to-3d creative cartoon modeling system with mobile ar. In *2017 IEEE Virtual Reality (VR)* (pp. 195-204). IEEE.

[5] Niskanen, O. (2019). An approach for automated rigging to facilitate 3D modeling in mobile augmented reality.

[6] Igarashi, T., Moscovich, T., & Hughes, J. F. (2005). As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)*, *24*(3), 1134-1141.

[7] Jacobson, A., Baran, I., Popovic, J., & Sorkine, O. (2011). Bounded biharmonic weights for real-time deformation. *ACM transactions on Graphics (TOG)*, *30*(4), 78.

[8] Kavan, L., Collins, S., Žára, J. ř., & O'Sullivan, C. (2008). Geometric skinning with approximate dual quaternion blending. *ACM transactions on Graphics (TOG)*, 27(4), 1–23.

[9] Magnenat-Thalmann, N., Laperrire, R., & Thalmann, D. (1988). Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface'88*.

[10] Messmer, S., Fleischmann, S., & Sorkine-Hornung, O. (2016). Animato: 2D shape deformation and animation on mobile devices. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications* (pp. 1-4).

[11] Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology* (Vol. 52, pp. 139-183). North-Holland.

[12] Olsen, L., Samavati, F., & Jorge, J. (2011). Naturasketch: Modeling from images and natural sketches. *IEEE Computer Graphics and Applications*, *31*(6), 24-34.

[13] Sorkine, O., & Alexa, M. (2007, July). As-rigid-as-possible surface modeling. In *Symposium on Geometry processing* (Vol. 4, pp. 109-116).

[14] Ruppert, J. (1995). A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of algorithms*, *18*(3), 548-585.

[15] Vollmer, J., Mencl, R., & Mueller, H. (1999, September). Improved laplacian smoothing of noisy surface meshes. In *Computer graphics forum* (Vol. 18, No. 3, pp. 131-138). Oxford, UK and Boston, USA: Blackwell Publishers Ltd.

[16] Whatever Inc. Rakugaki AR. whatever.co/en/post/rakugakiar/.

[17] Alciatore, D., & Miranda, R. (1995). A winding number and point-in-polygon algorithm. *Glaxo Virtual Anatomy Project Research Report, Department of Mechanical Engineering, Colorado State University.*

[18] Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, *24*(6), 574-594.

[19] Thiery, J. M., Guy, É., & Boubekeur, T. (2013). Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM Transactions on Graphics (TOG)*, *32*(6), 1-12.

[20] Meyer, M., Desbrun, M., Schröder, P., & Barr, A. H. (2003). Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III* (pp. 35-57). Springer, Berlin, Heidelberg.

[21] Guay, M., Ronfard, R., Gleicher, M., & Cani, M. P. (2015). Space-time sketching of character animation. *ACM Transactions on Graphics (TOG)*, *34*(4), 1-10.

[22] Garcia, M., Ronfard, R., & Cani, M. P. (2019). Spatial Motion Doodles: Sketching Animation in VR Using Hand Gestures and Laban Motion Analysis. In *Motion, Interaction and Games* (pp. 1-10).

[23] Ye, H., Kwan, K. C., Su, W., & Fu, H. (2020). ARAnimator: in-situ character animation in mobile AR with user-defined motion gestures. *ACM Transactions on Graphics (TOG)*, *39*(4), 83-1.