Towards Tattoo Previewing and Other Augmented/Mixed Reality Applications with Microsoft Kinect

Alexia Dreher[1], Alex Sanchez[1], Andrea Sedano[1], Kevin Chung[2], Michael Dambach[2], Shabnam Waseem[2], Blake Hament[3], Justin Zhan[3], Laxmi Gewali[3], Paul Oh[3]

Abstract:
As tattooing becomes more popular, so too does tattoo removal. Removal is painful, expensive, and not always effective. To avoid tattooing that customers will later regret, potential customers could use augmented/mixed reality to preview tattoo designs. This paper explores technical tools and methods using computer vision to overlay a digital image, like a tattoo design, on top of frames from a live video feed of the user's body. Using an RGB camera and depth camera, in this case the Microsoft Kinect, body position and volumetric data are captured. Two methods are presented for using this data for tattoo preview along with the methods' pros, cons, and state of development. This project was developed by and for high school students and educators to bridge the gap between secondary students and interesting new technology previously reserved for university-level studies.

I. Introduction

With the increasing popularity of tattoos, the number of tattoo removal clinics has increased drastically: 400% in the past decade [1]. Unfortunately, the cost of removing a tattoo can be ten times the cost of getting tattooed [2]. Tattoo previewing would help potential customers to visualize the design and make more informed decisions about tattooing. It would also save tattoo parlors time by eliminating the need for printing and adhering physical previews for every size and position a potential customer wants to preview. To achieve accurate tattoo previewing, a depth sensor should be used in addition to an RGB camera. The depth sensor is used to capture depth images which are processed for skeletonization and point cloud generation. Skeletonization is the identification and classification of joints and limbs in a body. Point clouds are collections of points in 3D space used to reconstruct 3D surfaces. See Fig. 1. for examples of skeletonization and point cloud capture. The Kinect for Xbox One is used due to its popularity, accessibility, and excellent depth imaging. Though there are many graphics libraries available, Daniel Schiffman's OpenKinect-for-Processing library and the Processing integrated development environment were selected as ideal for this project because it provides a wealth of powerful graphics tools at a level immediately accessible to beginner and intermediate programmers [3]. The Kinect is used to capture the skeleton and point cloud of a human body which can then be meshed together and textured with a 2D image to map tattoos onto the 3D surface of the body. By combining this mapped tattoo image with live video feed of the user, Augmented/Mixed Reality (A/M R) tattoo previewing can be achieved. Section II covers related work; Section III describes experimental setup; Section IV explains the basic tools used in the proposed augmented/mixed reality methods; Section V presents two methods for tattoo previewing with discussion of their pros, cons, and state of their development; and Section VI concludes.

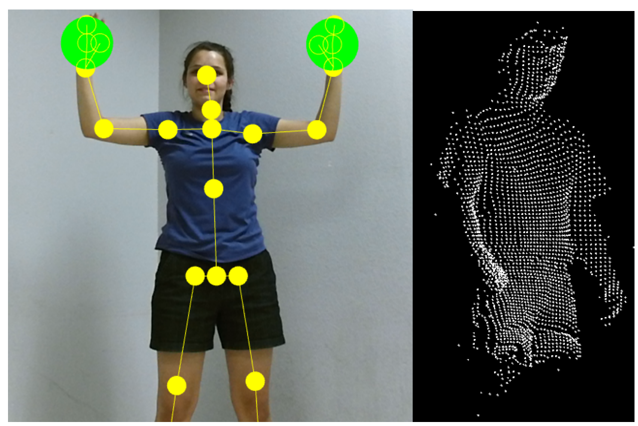

Fig. 1. In order to accurately map graphics onto a body and overlay live video, a depth camera is used to capture body skeleton (Left) and surface topology from point cloud (Right).

II. Related Work

The release of the Microsoft Kinect was a catalyst for 3D reconstruction research. Excellent work has been done to develop applications for robot navigation [4], robot control [5], scene reconstruction [6], body tracking [7], medicine [8], and more. Despite these advances, there are many Kinect owners who might have an interest in applications beyond gaming but who do not have the academic background to implement these findings. In order to bridge this gap between interest and ability, we propose developing a Kinect toolbox for tattoo previewing that will be accessible to high school students and educators who are beginner to intermediate programmers. To the researchers' knowledge at the date of submission, nothing has

*Submitted to IEEE CCWC on 11/11/2018. This work was supported by the University of Nevada, Las Vegas, the National Science Foundation, and by the US Department of Defense. Award Number: 170716.

1. Army Educational Outreach Program UNITE 2018
2. Army Educational Outreach Program RET 2018
3. University of Nevada, Las Vegas

been published on using Kinect for tattoo previewing. Many of the tools for tattoo previewing can also be applied to clothing previewing, next-generation user interface, and other mixed reality applications. This paper serves to describe and share tools developed by and for high school students and educators to empower the development of tattoo previewing and other mixed reality applications at the secondary level.

III. Experimental Setup

This section presents the hardware and software utilized in development. Table 1. compares features of the first and second-generation Kinect sensors, both of which were tested in development. Subsections A.-D. discuss Sensor, Image files, PC's, and Software libraries, respectively

Table 1. Comparison of Kinect for Xbox 360 and Kinect for Xbox One

Feature	**Xbox 360 Kinect**	**Xbox One Kinect**
Sensor	640 × 480p, 30fps	1920 x 1080p, 30fps
Camera depth	320 x 240	512 x 424
Field of Vision (horizontal)	57 degrees	70 degrees
Field of Vision (vertical)	43 Degrees	60 Degrees
Skeleton joints	20	26
Full Skeletons Tracked	2	6
Skeleton Joints	20	26

A. Sensor

In order to overlay digital images onto measured 3D surfaces in close to real time, some type of depth sensor is required. Microsoft Kinect is a popular and widely accessible sensor that combines an RGB camera, an infrared (IR) projector, and an IR depth camera. Table. 1. gives a comparison of important Kinect features [9]. The newer Xbox One Kinect has superior performance, but the original Xbox 360 Kinect suffices for the tools and methods presented here.

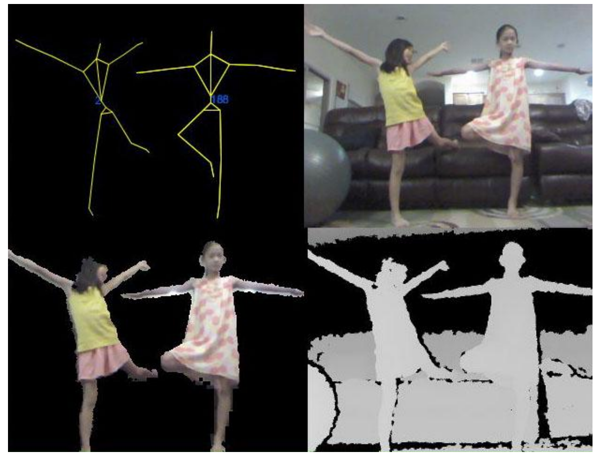

Fig. 2. Skeletonized body (Top Left) and depth thresholding (Bottom Left) tracking are accomplished with the Microsoft Kinect's combined RGB (Top Right) and depth (Bottom Right) cameras.

B. Image files

Initially, JPEG files were used to superimpose 2D images onto the 3D surfaces used for tattoo previewing. It was determined that PNG image files are a better solution. PNG files trim out negative space and preserve the main image for manipulation [10]. In contrast, JPEG files save negative space as white background. This white background then gets mapped onto our 3D reconstructions, which is unrealistic for tattoo previewing and requires extra steps for filtering out negative background or manipulating the projection mapping in other ways. The process of mapping a 2D image onto a 3D surface also requires high quality images to enable reproduction on high resolution monitors. High quality of tattoo design image is important for giving the viewer accurate information for making their final decision on tattooing.

C. PC's

The tools and methods presented were tested and developed on Mac OS 10.11.6 and 10.13.4 and Windows 10 OS Build 10240.17889. Machine RAM varied from 4 to 8 GB and Intel i5 and i7 processors were tested. On older machines, there was noticeable latency when running the code, but it did not interfere with high school students and teachers' ability to test and modify their programs and eventually get the results they desired. For those unsatisfied with code performance "out-of-the-box", it should not be a problem to integrate OpenGL and other open-source rendering software that allows for better speeds and efficiency.

D. Software Libraries

With the goal of maintaining code and software architecture friendly to beginner and intermediate coders, the Processing IDE and library was selected. The core Processing library contains many powerful graphics functions and commands with built-in debugging and powerful rendering. Additionally, there are several open-source libraries for initializing and streaming data from Kinect and using OpenNI methods that are quickly and easily linked to the Processing environment. This project uses the OpenKinect_Processing library for Mac and KinectPV2 for Windows computers. More details and code snippets for download are available at https://github.com/sudosurf/KinectAMR.

IV. Basic Tools

A. Overlay for Augmented/Mixed Reality

Overlaying is a process in which virtual graphics are superimposed on live video feed to create an experience that mixes the virtual and physical world. Virtual objects are rendered at given coordinates. These coordinates can be defined relative to specific body positions being tracked in close to real time. To accurately track bodies and identify and classify specific body parts, depth imaging is used. The depth data can be processed to generate a skeletonized representation of the body or a meshed surface approximation. The coordinates of the skeleton or surface are used to map virtual objects to the appropriate overlay position.

Fig. 3. Live video feed being overlaid with a virtual cylinder wrapped in Leonardo Da Vinci's *Mona Lisa*

B. Depth Imaging

Kinect captures depth images as matrices of pixels in which each pixel stores a depth value. This depth value is calculated from the time of flight as IR photons travel from the Kinect IR projector to the surrounding environment and back to the Kinect IR sensor. Note that infrared light is not visible to the human eye [11]. The IR projector sprays the room with photons according to the field of view (FOV) in Table. 1.

C. Point Cloud

A point cloud is a set of data points in a three-dimensional coordinate system. Occasionally, each point may also carry a fourth dimension like color or intensity. Point clouds are captured by 3D sensors, such as the Kinect, and can create a 3D figure when meshed together [12]. To generate a point cloud from a depth image, pixels are assigned X and Y coordinates in the image plane and then the depth value at every pixel is converted to a Z coordinate according to the Kinect camera parameters like project-sensor offset and focal length [13]. By connecting the points in a point cloud with lines, the points can be meshed together to create a 3D surface approximation. Furthermore, the mesh can be textured with a 2D image, which means 2D image coordinates are mapped to 3D points in the point cloud, and the mesh surface will appear wrapped in the 2D image.

D. Skeletonization

A virtual skeleton is created by identifying and joining the major joints and limbs of the user's body. Limbs can be identified from edge detection in RGB or depth images. Types of joints, such as hinge, ball-and-socket, etc., can be identified by the degrees of freedom and range of motion observed at that specific joint. Using translation and rotation matrices, coordinates are transformed from local joint frames to body and world frames. This creates a virtual skeletonized tracking of the user's body [14][15]. The skeletonization tools used in this project were developed as part of the OpenNI library.

V. Methods

Two methods are presented for tattoo previewing. Method 1 is for previewing tattoos that encircle a user's limb. Method 2 is a more general method for previewing tattoos that maps the tattoo design onto whatever 3D surface is presented. Both methods are still under development, and the progress, pros, and cons are presented for each.

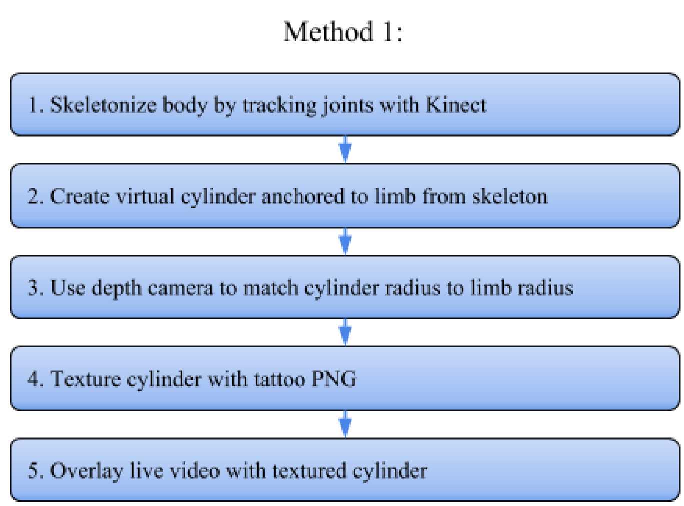

Fig. 4. Method 1 is the proposed pipeline for previewing tattoos that encircle limbs

Method 1

One limitation of skeletonization is that rotation about the axis along a limb is undetectable. To detect rotation of a rigid body, at least 3 points are required. Along a skeletonized limb and its bounding joints, there is only a line, one point in any cross-sectional plane – such there are not enough points in the plane perpendicular to the axis of rotation at points along the limb to determine rotation. Due to this limitation, this skeletonization method presents tattoo previewing for designs that encircle the user's limb in a cylindrical shape.

Popular tattoo designs like barbed wire, tribal style rings, and much more are often rotationally symmetric, such that they can be previewed just with center coordinates and radius length. Other tattoos that encircle limbs may not be rotationally invariant, but the designs can be easily rotated to the desired preview position with slight adjustments to the code.

Method 1 begins with skeletonization to track limb and joint positions. These coordinates are used to define the center of the virtual cylinder created in the next step. The height of the cylinder determines the width of the tattoo preview. To determine cylinder radius, depth values are used to identify the radius of the target limb. Next, the virtual cylinder is textured with the 2D tattoo design. Finally, the live video feed is overlaid with this virtual cylinder to create a A/M R preview of the potential tattoo. This process is outlined in Fig. 4.

Currently, stage 3 is still under development. A major challenge was that previously popular Kinect and OpenNI libraries have been close-sourced, making it difficult to find and recreate some necessary dependencies at this time.

Method 2

Method 2, outlined in Fig. 5., is a generalized tattoo previewing method that can be applied to virtually any body part. In this method, a point cloud is captured as described in Section IV. After appropriate filtering of points to capture only the desired body surface, the points are connected with lines to form a surface mesh. Fig. 5. shows how every set of four adjacent points is joined to make two triangle strips. Fig. 8. is a code snippet that illustrates looping through the Kinect depth image to create a point cloud and then filtering, meshing, and texturing points to create a 3D surface with a 2D image mapped on top. Fig. 7. is an example of mapping a 2D smiley face image on a meshed body point cloud captured from the Kinect.

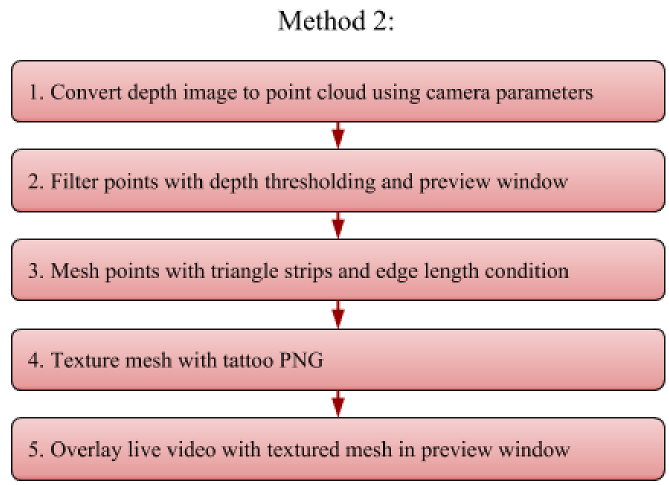

Fig. 5. Method 2 is the proposed pipeline for previewing tattoos in a bounded window within a live video feed.

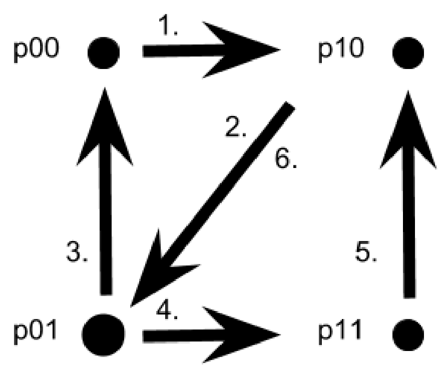

Fig. 6. After the point cloud is captured with Kinect, the points are meshed by iterating through every set of four consecutive points. For four given points, two triangle strips are created.

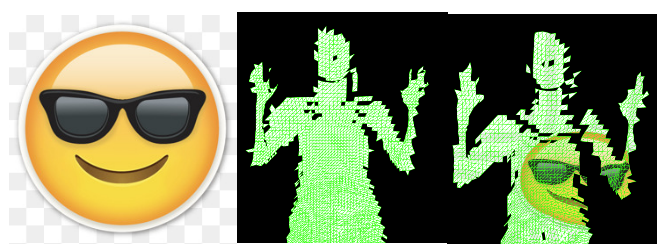

Fig. 7. A 2D image (Left) can be mapped onto a meshed surface (Center) to create a textured surface (Right). With refinement, this could be limited to a small window overlaid on live video for tattoo previewing.

```
for (int y = 0; y < kinect2.depthHeight; y+=skip) {
  for (int x = 0; x < kinect2.depthWidth; x+=skip) {
    int offset = x + y * kinect2.depthWidth;
    int d = depth[offset];
    //calculte the x, y, z camera position based on the
    //depth information and camera paramters
    PVector point = depthToPointCloudPos(x, y, d);
    pointCloud[offset]= point;
  }
}
for(int y=0;y < kdh-steps;y+=steps){
  for(int x=0;x < kdw-steps;x+=steps){
    i00 = y*kdw+x; //calculating indexes
    i01 = (y+steps)*kdw+x;
    i10 = y*kdw+x+steps;
    i11 = (y+steps)*kdw+x+steps;

    p00 = pointCloud[i00]; //indexed points
    p01 = pointCloud[i01];
    p10 = pointCloud[i10];
    p11 = pointCloud[i11];
    textureMode(NORMAL); //image coordinates normalized [0,1]
    beginShape(TRIANGLES);
    texture(smiley); // texture mesh with image
    // only positive depth points and limit edge length
    if ((p00.z > 0) && (p01.z > 0) && (p10.z > 0) &&
        (abs(p00.z-p01.z) < max_edge_len2) &&
        (abs(p10.z-p01.z) < max_edge_len2)) {
          // For vertex(x,y,x,u,v), (x,y,z) is point in 3D space
          // (u,v) is normalized image coordinate
          vertex(p00.x,p00.y,p00.z, p00.x/kdw, p00.y/kdh);
          vertex(p01.x,p01.y,p01.z, p01.x/kdw, p01.y/kdh);
          vertex(p10.x,p10.y,p10.z, p10.x/kdw, p10.y/kdh);
        }
    if ((p11.z > 0) && (p01.z > 0) && (p10.z > 0) &&
        (abs(p11.z-p01.z) < max_edge_len1) &&
        (abs(p10.z-p01.z) < max_edge_len2)) {
          vertex(p01.x,p01.y,p01.z, p01.x/kdw, p01.y/kdh);
          vertex(p11.x,p11.y,p11.z, p11.x/kdw, p11.y/kdh);
          vertex(p10.x,p10.y,p10.z, p10.x/kdw, p10.y/kdh);
        }
    endShape();
}
```

Fig. 8. Code snippet illustrating conversion of depth image to point cloud and meshing and texturing of filtered points

Currently, all modules for Method 2 are functioning, but mesh and texture results are not satisfactory. Mesh resolution has improved significantly through testing and development, but gaps persist at points with high surface gradient. More work is required to achieve robust texturing that does not significantly warp or obscure 2D image properties.

VI. Conclusion

Augmented/Mixed Reality (A/M R) is an emerging research space with many future applications that are exciting for high school students and teachers. However, there is a large gap between student interest and access. Most A/M R tools and toolboxes are too advanced for beginner or intermediate coders. In this paper we present several tools and methods at the secondary student level that can be used to develop A/M R tattoo previewing and similar applications. For instance, Method 1 can also be applied to previewing accessories like necklaces, bangles, earrings, and other cylindrical objects. Method 2 could be used to generate clothing previews for regions of cloth that are pulled tight against the user's body. In the short term, we hope to use these methods to finalize and deploy a full pipeline for tattoo previewing. This pipeline may reduce the number of people who experience buyer's remorse after getting a tattoo by finding a way for them to "preview" the tattoo beforehand. This would not only save time and increase productivity at tattoo parlors but would also reduce the amount of money customers spend on tattoo removal.

VII. Acknowledgment

The researchers would like to extend special thanks to Dr. Justin Zhan, Dr. Paul Oh, Blake Hament, the Army Education Outreach Program (AEOP), the National Science Foundation, University of Nevada, Las Vegas, and Research Experiences for Teachers (RET). RET provides secondary students and teachers the opportunity to work with graduate students and professors on university-level research projects, and this publication is a direct outcome of that collaboration.

References

[1] C. Cardellino, "Tattoo Removal Is Up 440 Percent," *Cosmopolitan*, 09-Oct-2017. [Online]. Available: https://www.cosmopolitan.com/style-beauty/beauty/news/a29182/tattoo-removal-is-up-440-percent-over-the-last-ten-years/. [Accessed: 31-May-2018].

[2] J. Lim, "Laser Tattoo Removal - Cost, Sessions, Recovery," *DocShop*, 06-Sep-2017. [Online]. Available: https://www.docshop.com/education/dermatology/body/tattoo-removal. [Accessed: 04-Jul-2018].

[3] D. Schiffman, "Open Kinect for Processing," *2018 Github Repository, ProcessingOpenKinect-for-Processing*, https://github.com/shiffman/OpenKinect-for-Processing . [Accessed: 31-May-2018].

[4] F. Dulko, "360° color and depth mapping of an indoor room:: Microsoft Kinect 2.0," *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, 2018, pp. 1-11. doi: 10.1109/LISAT.2018.8378016

[5] H. X. Nguyen *et al.*, "Performance evaluation of an inverse kinematic based control system of a humanoid robot arm using MS Kinect," *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Macau, 2017, pp. 469-474. doi: 10.1109/ROBIO.2017.8324461

[6] Y. Zhang *et al.*, "A Kinect-Based Approach for 3D Pavement Surface Reconstruction and Cracking Recognition," in *IEEE Transactions on Intelligent Transportation Systems*. doi: 10.1109/TITS.2018.2791476

[7] Q. He, Y. Ji, D. Zeng and Z. Zhang, "Volumeter: 3D human body parameters measurement with a single Kinect," in *IET Computer Vision*, vol. 12, no. 4, pp. 553-561, 6 2018. doi: 10.1049/iet-cvi.2017.0403

[8] B. Çubukçu and U. Yüzgeç, "A physiotherapy application with MS kinect for patients with shoulder joint, muscle and tendon damage," *2017 9th International Conference on Computational Intelligence and Communication Networks*

(CICN), Girne, 2017, pp. 225-228.
doi: 10.1109/CICN.2017.8319390

[9] “How Does The Kinect 2 Compare To The Kinect 1?,” *Zugara*, 19-Dec-2017. [Online]. Available: https://zugara.com/how-does-the-kinect-2-compare-to-the-kinect-1. [Accessed: 31-May-2018].

[10] E. Z. Goodnight, “What's the Difference Between JPG, PNG, and GIF?,” *How-To Geek*, 11-Jul-2017. [Online]. Available: https://www.howtogeek.com/howto/30941/whats-the-difference-between-jpg-png-and-gif/. [Accessed: 31-May-2018].

[11] “How Does The Xbox Kinect Work,” *2N3904: Major Brands : Transistor 2N3904 NPN General Purpose : ICs & Semiconductors*. [Online]. Available: https://www.jameco.com/jameco/workshop/howitworks/xbox kinect.html. [Accessed: 07-Jun-2018].

[12]“Infrared Radiation,” *We Didn't Start the Fire (Facts) History Summary from 1949-1989 by Ron Kurtus - Lessons Learned from History: School for Champions*. [Online]. Available: https://www.school-for-champions.com/science/infrared.htm#. [Accessed: 04-Jun-2018].

[13] D. Shiffman, “12.3: Raw Depth Data - Point Clouds and Thresholds - Kinect and Processing Tutorial,” *YouTube*, 25-Nov-2015. [Online]. Available: https://www.youtube.com/watch?v=E1eIg54clGo. [Accessed: 31-May-2018].

[14] Rotenberg, Steve. “Chapter 2: Skeletons.” *CSE169* , University of California, San Diego, Jan. 2016, cseweb.ucsd.edu/classes/sp16/cse169-a/readings/2-Skeleton.html.

[15] “Chapter 2: Skeletons,” *Descartes & Dualism*. [Online]. Available: https://cseweb.ucsd.edu/classes/sp16/cse169-a/readings/2-Skeleton.html. [Accessed: 31-May-2018].