Particle based 3D Hair Reconstruction using Kinect and High Resolution Camera

by

Zhongrui Li

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements for the degree

Master of Applied Science in Electrical and Computer Engineering

Ottawa-Carleton Institute for Computer Science

School of Information Technology and Engineering

University of Ottawa

Abstract

Hair modeling based on real-life capturing is a rising and challenging topic in the field of human modeling and animation. Typical automatic hair capture methods use several 2D images to reconstruct 3D hair model. Most of them usually adopt 3D polygons to present hair strands, and a few recent strand-based methods require heavy hardware settings.

We introduce an approach to capture real hair using affordable and common devices such as a depth sensor and a camera to reconstruct a 3D hair model based on particle system. Kinect™ sensor from *Microsoft* is chosen to capture 3D depth data. However, as Kinect 3D depth data are known to be noisy and 2D texture image to be of low quality, an additional DSLR camera is employed in the system in order to capture high resolution image for hair strand extraction. The proposed approach registers the 3D hair point cloud and high resolution image in the same space, extracts the hair strands manually from the image, and then generates 3D hair strands based on Kinect depth information. Eventually, a particle based 3D hair model is reconstructed.

The proposed method captures 360-degree views by collecting datasets of real-life hair with four sets of Kinect sensors and DSLR cameras in four viewpoints. We register the DSLR camera image in the space of Kinect to build the mapping relationship between 2D and 3D. Therefore, the image from the DSLR camera can be mapped on the point cloud replacing the existing Kinect texture image, resulting in a new high-quality texture image of the 3D data. Next we manually select the hair strands in the high resolution image and we use control points to represent hair strand as a spline curve. These 2D control points are then projected on the 3D point cloud in order to obtain the corresponding 3D information. In 2D image, some hair strands are partially occluded by some other hair strands, the result is that the occluded hair strand is separated into two segments in 3D. An algorithm is applied to analyze and build the connection

between the hair strand segments. Meanwhile some refinement works are done with the 3D hair strands, filtering and interpolation techniques are utilized on the 3D hair strand splines to generate smoother 3D hair strands. Finally we reconstruct the 3D hair model, where the strands are represented in the particle system.

Our method, combining a depth sensor and an high resolution camera, is novel and has many advantages which other approaches do not have; (i) hardware setting is simple and affordable; (ii) combination of high-quality image of DSLR and depth of Kinect takes advantage of each of them; (iii) the 2D and 3D combined method allows us to repair and refine the 3D data; (iv) Spline-based hair representation can be used to construct a hair particle system which has many advantages of hair animation and simulation.

Acknowledgments

I must give my deepest appreciations and sincerest gratitude to my supervisor Prof. WonSook Lee. Without her continuous guidance and support during my research, this thesis is impossible. Her positive attitude towards life and work influenced me during my research work.

I also need to thank Dr. Alberto Chavez, his excellent work and invaluable assistance, guidance and feedback made my research successful. I should give my thanks to my colleagues, Fatemeh Cheraghchi and Chao Sun as well. That was a wonderful experience of working with them in the hair modeling project.

My thanks and appreciations also go to other colleagues at CG++ group, Ali, Kaveh, Iman, Aseem, Jing, Shiyi and Yongji. These greatest colleagues and friends helped me during my master study, not only in the research field but also throughout my life in Ottawa.

For sure, I need to give my gratitude to my girlfriend Xin, her understanding, support, company and the love made my life full of meaning.

Finally, I am grateful to my parents. Their patience, support and encouragement made my study and work possible, this thesis is dedicated to them.

Table of Contents

List of Figure

List of Table

Abbreviations

2D	Two Dimensional
3D	Three Dimensional
IR	infrared Radiation
ICP	Iterative Closest Points
DSLR	Digital Single-Lens Reflex
NURBS	Non-uniform Rational B-Spline
SPH	Smoothed-Particle Hydrodynamics
SIFT	Scale Invariant Feature Transform
CCD	Charge-coupled device
PSD	Position sensitive device
ToF	Time of Flight

Chapter 1. Introduction

3D human model is a necessary component for many kinds of computer graphics applications, such as computer game, 3D computer animation and virtual environment. As an identity characteristic of human appearance, hair has become an important research object of 3D human modeling. 3D hair reconstruction based on hair capture is an interesting and challenging work in the 3D reconstruction field. Compare to other parts of human body, hair is the most complex part that contains a huge amount of narrowly structured strands. The brightness, colours and styles of hair affect the difficulty of capture, analysis and reconstruction.

In the 3D hair modeling field, there are two kinds modeling methods: geometry polygon-based and particle-based. Traditionally, a computer graphics designer uses a modeling tool to interactively design hair shape using the polygonal geometry and texture mapping. Even after more automatic hair modeling methodology has been introduced, most existing methods are still the geometry polygon plus texture mapping type; in this method, hair strands are presented as thick, long polygon geometry virtually, and they are usually of low quality due to the detail limitation of captured data and lack of graphics performance.

With the advancements of the hardware nowadays, the strand-based approach for hair reconstruction is achievable. Different than the thick geometry of the hair strand model, the thin strand-based hair modeling provides a more realistic visualize 3D effect, and also supports dynamic hair simulation using particle-based space curves hair representation. Many recent high-quality hair animation systems are based on particle system. Particle system is a powerful tool and flexible option for hair modeling. With the individual hair strands represented by the numerous particles, the particle-based hair modeling provides realistic and natural hair modeling

results. Benefiting from the advantages of physical simulation of the particle system, many mature researches make optimization for dynamic hair modeling and hair animation. However, the numerous particles and a large amount of computation require higher graphics performance and decrease the rendering speed. An existing particle–based hair modeling system, such as *Blender*™ [3], is capable of generating a huge amount of hair strands by assigning comparatively few guide hair strands. Obtaining the 3D position of spline curve control points of guide hair strands is the core problem for the strand-based and particle-based method. Our 3D hair reconstruction research is carried out step by step based on this motivation.

In the capture-based 3D hair reconstruction field, the depth data are essential for the existing hair reconstruction systems. As hair structure is detailed and complex, a traditional depth sensor, such as a laser scanner, always shows low capability of capturing real hair because of the specular and diffuse reflection of the hair surface. Many groups select the image multi-view stereo based 3D reconstruction approach to create the hair depth and orientation map. However, the relative expensive and complex experiment devices setup limits this type of research to be commonly used. In 2010, Microsoft released Kinect sensor, this low-cost (about $200/each) and compact device is capable for capturing depth information in a rapid way. The proposed system in this thesis employs Kinect to capture the 3D point cloud of real hair directly. Therefore, the 3D orientation analysis work is simplified. Unfortunately, the accuracy of Kinect is quite limited for both depth and texture image to recognize the tiny structure of hair strands, the resolution of Kinect colour camera is only up to 1280*960. The low resolution limits our image processing based 2D strand extraction method, so we introduce an additional of high-resolution digital single lens reflection (DSLR) camera into the capturing system, and build the mapping between high resolution image and depth frame. It is also important to point out that the quality of the

image is also affected by the setting of the camera, such as f-number, ISO value, etc… Our approach provides freedom to choose a camera with the parameters necessary for the quality of the image related to image processing.

1.1 Overview of the System

As shown in Figure 1.1, the 3D hair reconstruction system is divided into three stages

- Data capture and registration stage (Chapter 3)
- Data analysis stage (Chapter 4)
- Rendering stage (Chapter 5)

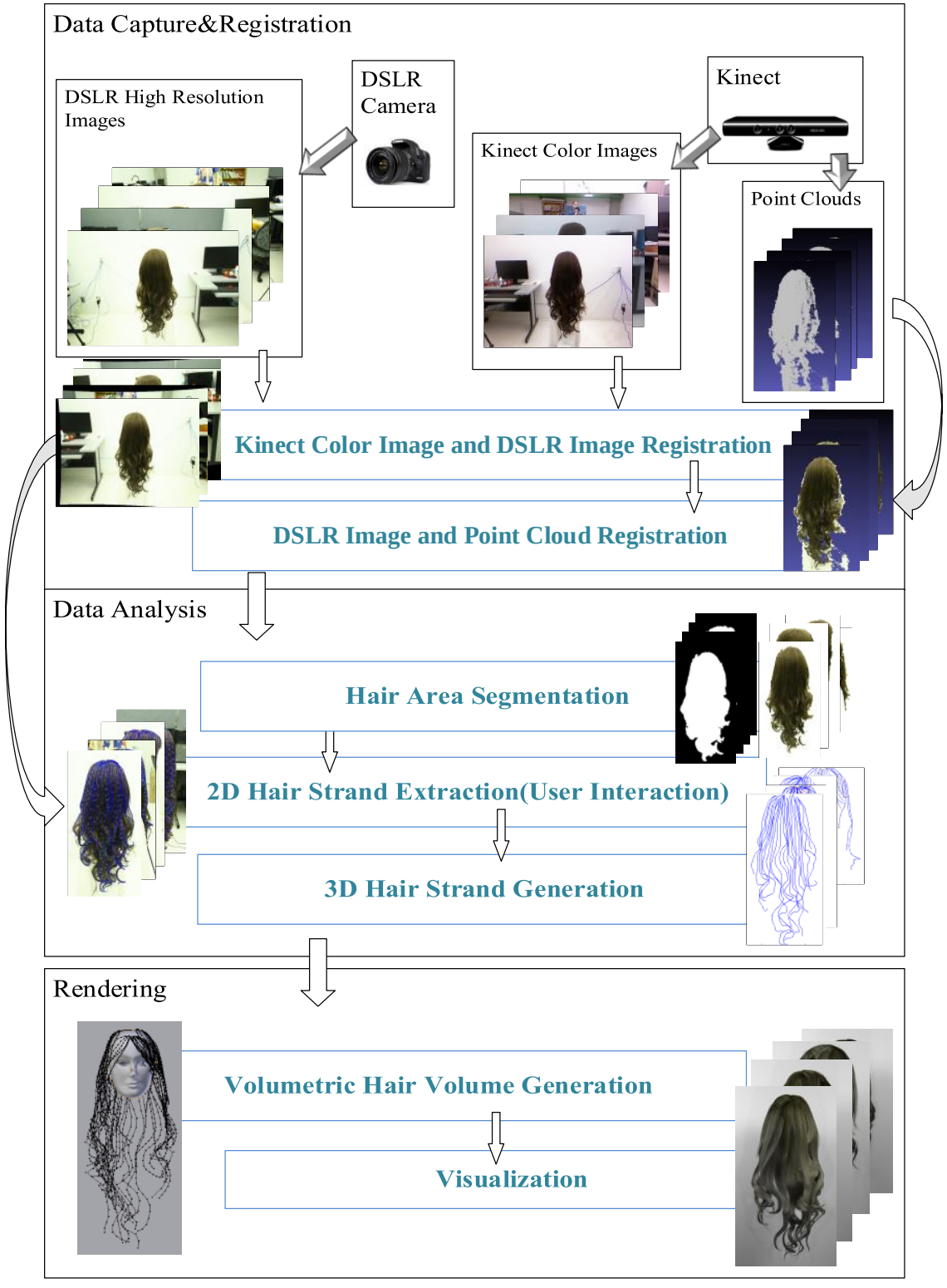


Figure 1.1: Particle system-based 3D hair reconstruction system pipeline. The 2D hair strands extraction procedure can be automatic using image processing, but we extract manually in this thesis.

1.1.1. Data Capture and Registration Stage

In the first stage, after the capturing devices, four Kinect sensors and DSLR cameras, were positioned in the fixed locations, two pre-procedures are required: (i) Kinect sensor pairing calibration and (ii) Kinect colour camera and DSLR camera registration.

To calibrate the Kinect sensors, we capture a set of feature images and corresponding point clouds by two Kinect sensors to calculate the relative position and orientation between the two Kinect sensors. This calibration runs for all combinations of the two out of the four sensors. The camera registration procedure is performed between each pair of Kinect colour camera and DSLR camera. We register the feature images and obtain the transformation between the images form Kinect colour camera and DSLR camera.

After the capture devices setup and calibration, the target hair model is placed in the designated position. We use face-tracking-based segmentation on the depth frame to remove the background. And the segmentation is applied on the capture symmetric capture system. Moreover, we adopt a temporal filter like approach to improve the quality of point clouds. For each point cloud, two corresponding colour images are captured by Kinect colour camera and by DSLR camera respectively.

To generate a 360^{o} point cloud with the point clouds of four viewpoints, the system applies point clouds alignment using the relative position and orientation information from the Kinect calibration procedure. Iterative closet point (ICP) algorithm is also applied to refine the point clouds alignment procedure. Using the transformation between the Kinect colour camera and DSLR camera, the system registers the hair images from the Kinect colour camera and DSLR camera, and then the system registers the Kinect depth frame and the high-resolution image to build the mapping relationship between 2D and 3D data.

1.1.2. Data Analysis Stage

In the data analysis stage, we analyze and process the captured point clouds and images. During this stage, there are three main steps: (i) hair area segmentation, (ii) manual 2D hair strands extraction and (iii) 3D hair strands generation.

The point clouds are already segmented and the background points are removed in the data capture stage. And we also apply hair area edge detection to segment the hair area in the point cloud thereby removing other noisy and unnecessary points. Afterwards, a complete hair area point cloud is obtained, and this point cloud is used to determine a scalp area which will be used as a particle emitter. In the data analysis stage, the most important work is to extract the hair strands in the high-resolution images. However, this work is not a part of the content of this thesis. The data analysis stage is based on the assumption that there are enough available hair strands extracted from the high-resolution images. Under this assumption, the conspicuous hair strands in the high-resolution images are traced and drawn manually, and we adopt a user interactive tool to select the control points of each strand. Therefore, with the mapping relationship, the corresponding 3D control points in the point cloud are identified, and then the 3D strands can be generated. The hair strands, which are partially occluded by some other hair strands in 2D image, are divided into two segments in 3D space. The short and randomly distributed strand segments are unavailable for particle-based hair reconstruction. In order to solve this problem, an algorithm is implemented to build the connection and interpolates points between segments. With this algorithm, all strand segments are merged as complete hair strands. Eventually, the system generates 3D hair strands which are represented by spline curves.

1.1.3. Particle-based Hair Modeling and Rendering Stage

In the last stage, we generate particle-base hair model using the processed data and render the hair model in the *Blender*. A series of uniformly distributed control points are selected based on the curvilinear function of each hair strand. These control points and spherical skull are imported into the particle system to generate the 3D hair strands. With these inputs, and some specific settings like child number, hair material and lighting setting, etc…, the particle system-based 3D hair model is reconstructed. We use *Blender Cycle* render engine for realistic hair rendering. Figure 1.1 shows the proposed particle-based hair reconstruction system pipeline.

1.2 Organization of the Thesis

This thesis is organized as follows:

- Chapter 2 contains literature reviews about the background knowledge and other related researches about hair modeling, depth data capturing and image processing.
- Chapter 3 describes the details about the data capture and registration stage.
- Chapter 4 details various techniques on data analysis stage for 3D hair strand and skull generation.
- Chapter 5 introduces the process work in rendering stage.
- Chapter 6 provides the experiment result and system validation to demonstrate the system performance.
- Chapter 7 concludes the thesis, summarizes the presented methods and main contributions. At end it remarks the potential direction of further researches.

Chapter 2. Literature Review

The hair modeling methods can be classified as manual modeling and automatic modeling. Hair reconstruction based on 3D data acquisition is more popular in the automatic modeling field. Therefore, in this chapter we delineate some of the techniques and background knowledge about 3D hair modeling as well as techniques of data processing in the reconstruction process, such as data registration, segmentation and extraction.

2.1 3D Hair Modeling

3D Hair modeling is an indispensable work of the human character modeling. Generally, hair modeling methods can be classified into two types, first one is manual modeling which relies on a designer's professional skill of computer graphic and visual art, second is automatic hair modeling, where by a large portion of the work is done by hardware and computer programming. There is a well-done survey paper about hair modeling presented by Ward et al. [7].

For the manual modeling method, professional 3D computer graphics software allows the artist to model 3D hair for human character. For example, 3D computer graphics software by *Autodesk*, *Maya*™, provides a *hair system* to help the user generate and modify 3D hair model rapidly [1]. The popular software-based hair modeling method is built on polygon geometry with texture mapping. Some other hair systems represent hair not as polygonal system, but particle systems that is known to provide more natural and high-quality animation. Hadap et al. [2] show a particle system-based hair modeling method which provides dynamic hair motion control. And *Blender*™ , an open source computer graphics software, supports a system to create a hair model with particle system [3].

For the automatic hair modeling, there exist two types of approaches, some are based on synthetic data, and others are based on real-life hair capture. In recent years, capture based 3D hair reconstruction has become a popular direction of hair modeling. A few recent papers show successful results of reconstructing the realistic looking hair model by capturing real-life hair. The first hair capture approach, which generates hair from photographs, was proposed by Kong et al. [6] and various hardware equipped hair modeling systems are being done afterward. This kind of image-based hair modeling methods capture the hair geometry and analyze the hair volume information in the images, then reconstruct the hair in 3D space.

The following subsections describe the related works of hair modeling, covering geometry polygon-based hair modeling, particle-based hair modeling, capture-based hair modeling and hair animation.

2.1.1 Geometry Polygon-based Hair Modeling

Geometry polygon-based hair modeling is a part of the general polygon object modeling. This modeling approach employs a 3D polygon mesh and deforms the mesh to represent hair wisp. A parametric surface such as a NURBS surface is used in some of these methods, thus hair strands can be grouped into strips and modeled geometrically by parametric surface [8]. Texture of alpha map hair images can be mapped on the polygon surface to reveal the texture, colour and light brightness of hair strands. The geometry polygon hair modeling has been widely used as a basic approach to create hair models in both manual and automatic modeling fields.

Some geometry polygon methods have been developed and proposed in the past few years. W. Liang et al. present a method based on 2D strips hair model using U-shape polygon meshes [5], this method improves the volumetric visual effect of the model. Another polygonal hair

modeling method, *Hair meshes*, is proposed by Yukselet et al. [9]. This method provides the polygon meshes to the artist to help them design the desired hair style. This method generates individual hair strands of different hair styles on the mesh surface with various topological operations. Scheuermann [10] introduces a real-time hair rendering algorithm with the polygon hair model, which is illustrated in Figure 2.1.

Figure 2.1: The real-time rendering and shading result of a polygon hair model. A simple of geometry polygon and texture mapping mode of hair modeling, reprinted from [10].

2.1.2 Particle-based hair Modeling

Particle system is a useful technique in computer graphics which is extensively adopted in the fuzzy simulation field. Many chaotic systems that contain a large number of fuzzy objects, such as fire, explosions, smoke, moving water and other kinds of natural phenomena, can commonly be replicated by particle systems [11]. Particle systems can be divided into two types, animated particles and static particles. In practice, the static particle shows the overall trajectory of the particle, it is the result of the simultaneous rendered entire cycle of each particle. The parameters of the particle can be some physical attributes that make the simulation of relative collision between the particles possible. Therefore, the dynamic hair modeling method with particle is

possible to achieve. A patent for hair simulating and rendering was demonstrated in 1998 [12]. This patent uses a simulator, which contains a particle system that emits particles from the surface, and a renderer is also presented to calculate fuzzy segment generation, self-shadowing and lighting in 2D image rendering from 3D space [12]. Bando et al. [13] utilize a set of SPH particles for hair modeling, and this method can simulate the interaction between the hair particles during the hair motion.

In recent years, particle system-based hair modeling has rapidly become a mature hair modeling method in some 3D computer graphics software. For example, *Blender* provides a brief procedure to create a particle system-based hair model. The first step is to assign a mesh area of the head object as particle emitter, and then an initial static particle system, which contains a specified amount of straight guide hair strand particles, can be created. The user can easily model the hairstyle by moving the control points for each guide hair strand. Afterwards, the user can assign specified amount of child hair, which surround the guide hair particles. After modifying the material and colour, a complete particle system based hair model is completed. An example of hair modeling with *Blender* is demonstrated in Figure 2.2. Furthermore, *Blender* also supports hair animation rendering using a particle system.

Figure 2.2: Hair Modeling using *Blender* Hair Particle System reprinted from [14].

2.1.3 Capture-based Hair Modeling

Hair capture is a challenging work in the realistic hair reconstruction field. The 3D information of the hair is necessary for reconstructing a 3D hair model. However, traditional 3D capture devices have many limitations for hair capture. For example, for 3D laser scanners, the erroneous result is always caused by the complex specular and diffuse reflection of real-life human hair.

Paris et al. present a hair capture method without 3D scanner device [15], as only a fixed camera and moving light source are utilized in their system. They analyze the 2D image, and recover the 2D orientation of the hair strand segment. The corresponding normal of the same segment is obtained by 3D analysis of the illumination variations with image sequences. A plane which contains the segment in the 3D space can be identified by the normal, and the 3D orientation of the segment is the intersection of the two planes. This method is widely usable with its easy experiment setup. However it also has limitations, for example, it only works with thin and orientation visible hair strands and it cannot handle curls. Afterward, Paris et al. [16] present an improved capture method using multiple projectors and cameras address the challenges raised by the reflectance and intricate geometry of hair and the result is outstanding. However this method relies on input from the giant and relative expensive hardware setting (16 digital cameras, 150 LED light sources and 3 DLP projectors).

An automatic single view hair modeling method of user interactive application is presented by Chai et al. [17]. This method adopts an effective high-precision 2D strand tracing algorithm and a hair depth optimization algorithm, but the method has not been proven for solid case studies. Beeler et al. [18] focus on the high-quality representation of hair fibres and underlying skin

surface, their method focuses on short hair such as short beards and moustaches on a face even though the result is outstanding.

Based on Paris' image hair orientation analysis method, Luo et al. [19] propose a multi-view hair capture method using orientation fields. This method analyzes the orientation field of input multi-view images and generates depth maps based on the orientation field. Then stereo matching technique is applied to construct a merged high frequency details hair mesh. This method observes hair colour vary quickly with changes in viewpoint.

Figure 2.3: The capture system used in "Structure-Aware Hair Capture" method [20] , a robotic gantry and SLR camera to capture images for the wigs (left), a camera array of 30 SLR cameras for real hairstyle (right).

And Luo et al. [20] demonstrate an improved work for complex structures of real-life hairstyles using a complex capture system (Figure 2.3). The new method addresses the problem that the point cloud of the complex structure hairstyle, which is generated from the 3D orientation-based depth map, could not contain the complete details of the hair strand orientation; it adopts ribbon mesh to continually cover the adjacent strand segments until a locally coherent wisp structure is revealed. The hair point cloud is reconstructed with this synthesized hair strands

(ribbon), and the new hair model reveals the hair detail closer to the real-life hair; meanwhile, the hair simulation is also achievable by this method.

2.1.4 Hair Animation

Hair animation involves the dynamic motion of hair, and it is the major topic of hair physical simulation, including collision detection between the hair strands. The mass-spring system [21] and the projective angular momentum [22] are early physically-based hair simulation approaches that treat each individual hair strand as a string of particle. These approaches have sufficient degrees of freedom in translation and rotation but cannot handle hair twist.

Mechanical models have been used to simulate more accurate hair strand dynamics. Bertals et al. [23] use an energy minimization method to calculate the strand position with physical force and a super-helices model [24] for hair dynamic. In this method hair strand is represented as a sequence of helices, the torsion and curvature are general coordinates, which control the helix centerline, for computation of the dry friction in hair dynamics.

Some improvements of the mass-spring system are introduced. An altitude spring model is proposed by Molino et al. [25] that can prevent volume collapse and make hair strands revert to their original shapes. The improved mass-spring method is more convenient to integrate in the optimization procedure, and mechanical-based helices are more powerful when simulating complex hairstyle. Particle–based hair simulation is adopted in the video-based hair dynamics modeling method which is introduced by Zhang et al. [26]. They combine image-based hair capture and hair physical simulation. Figure 2.4 shows the result of this approach.

Figure 2.4: The full swinging motion of three kinds of hair in approach of Zhang et al. [26].

2.2 Depth Data Acquisition and Processing

Depth data acquisition is an important procedure of capture-based hair modeling. Since the common capture device cannot provide full detailed data, most methods reconstruct initial hair geometry, and refine it using various approaches. Commonly, there are two types of depth data acquisition methods: image-based and 3D scanner-based. The former always provides high accuracy result and the latter always responds immediately. Since the accuracy of the acquired depth data directly affects the final hair reconstruction result, many researches adopt the image-based method to reconstruct 3D data to avoid the risk caused by the low accurate depth data.

3D scanner is a category of the device that analyzes and collects object's depth data. 3D scanner can be constructed with some different technologies. The traditional 3D scanner emits some kinds of light or laser; the light will be reflected if they contact the surface of the object and the sensor receives the reflected light and estimates the distance. The limitation of the 3D scanner is indicated by the physical properties of the emitted light or laser, the working range and accuracy are very different between different technologies. Depth sensor has become a well-known device represented by *Microsoft* Kinect, this type of devices adopt infrared projector to

project the pattern of infrared light on the object, and it reads the pattern using the infrared camera to work out the depth.

The following subsections introduce the multi-view stereo method (image-based) and different kinds of 3D scanners.

2.2.1 Multi-view Stereo

Multi-view stereo (stereo vision) is commonly adopted in the aspect of recovering the geometry of a scene or the target object from a collection of images taken from known camera viewpoints [27]. Generally, in order to estimate the relative transformation between each viewpoint, most approaches adopt the calibrated camera viewpoint system. The shape and the motion of the object in the scene can be estimated by photometric properties-based measurement. Moreover, Lhuillier and Quan [28] introduce a quasi-dense approach for surface reconstruction from uncalibrated images, and this approach computes the corresponding information and geometry based on the quasi-dense point features.

The general multi-view stereo method always constructs a rough initial geometry and applies refinement to the initial geometry to with optimizing cross-view photo-consistency and surface smoothness [29]. Another important way to refine the reconstruction is to improve the capture device setup. Some proposed methods use the wide-baseline camera setup based on multi-view stereo. This setup is desired to find robust feature correspondence between input images. Scale-invariant feature transform (SIFT) descriptor, which is presented by Lowe, can find the feature points in images with image scaling, translation and rotation, and partially invariant to illumination changes and affine or 3D projection [33].

Seitz et al. propose two evaluation properties about the multi-view stereo based reconstruction method, one is accuracy which indicates that how close the reconstruction result is to the ground truth model, another one is completeness which indicates that how much of the ground truth model is modeled by the reconstruction result [27].

Since hair is highly specular, and colour varies quickly with the moving of viewpoint, the capture hair image always lacks obvious features like reliable texture or corner points. The reconstruction results of generic multi-view algorithm always lack of hair details.

2.2.2 Laser Scanner

The laser scanner is frequently adopted 3D data capture device in various reconstruction systems. This kind of devices scan the real-life object or environment using laser and collect the 3D data of the target shape and appearance to construct the digital 3D model. The laser scanner is a type of non-contact active 3D scanner which doesn't contact the object, but emits a laser to detect and identify the surface of the object.

Triangulation-based 3D laser scanner utilizes laser to probe the environment. It employs a laser emitter to radiate laser to the object, besides a Charge-Coupled Device (CCD) or Position Sensitive Device (PSD) sensor is used to track the laser dots on the object. The location of the tracked laser dot is related to the depth of the laser dot plane. The laser dots, the CCD/PSD sensor and the laser emitter form a triangle [32]. The Time-of-flight (ToF) laser scanner also belongs to the laser scanner, ToF scanner calculates the depth data of the target surface by timing the round-trip time of a pulse of laser [34]. And ToF sensor is commonly used in the 3D data capture field, Cui et al. [35] present a reliable shape data capture and alignment method with ToF camera alone. And Schuon et al. [36] show the super resolution methods that were originally

developed for colour images can be applied to capture higher resolution 3D geometry with a ToF depth camera.

Advantage and disadvantage of the mentioned laser scanners are: the triangulation-based 3D scanner has relatively high accuracy but it has limited range of some meters, the ToF scanner is just the opposite [34].

2.2.3 Structured Light Depth Sensor - Kinect

In recent years, with the release of Microsoft Kinect, the structured light depth sensor attracts more attention. This kind of 3D scanning device uses structured light to project light patterns and analyzes the stripe patterns for collecting the depth data and measuring the shape of an object.

Kinect is the most popular depth sensor presently, it was released by Microsoft in 2010 [37]. Kinect received a lot of attention because of its rapid human pose recognition system. As a low-cost depth sensor, Kinect was an attractive alternative to the high price and large scale laser scanner in applications for developers and researchers; its reliability and speed of measurement make it an ideal depth sensor for indoor robotics, 3D scene reconstruction and object recognition [38]. So far, numerous Kinect open source drivers have been released, making it possible to use Kinect as a depth data capture tool. Kinect provides the point cloud without large systematic errors compared with laser scanning data. However, the random error of depth measurement increases quadratically and density of point decreases with increasing the distance to the sensor [39].

Kinect is the major capture device in our system and the subsections to follow introduce the technical details and accuracy of Kinect.

2.2.3.1 Technical Description of Kinect

Figure 2.5: Microsoft Kinect

As shown in Figure 2.5, the Kinect sensor consists of one laser-based IR projector, one RGB colour camera and one IR camera. The IR camera is used to capture the infrared laser projected from the IR projector. The IR camera and the IR projector constitute a stereo pair with a baseline of approximately 7.5 cm [40]. The triangulation-based depth measurement method and structured light technology are adopted by Kinect (shown in Figure 2.6 below). In order to estimate the depth frame, the IR projector projects a predefined pattern which consists of tiny dots, and the infrared camera captures the projected light pattern then compares it against the predefined reference pattern [41]. The introduction of an object in the field of view of the depth sensor deforms the reference pattern. The resulting shift of the dots in the pattern from the reference image is measured as a disparity in the image plane of the IR camera [40]. The Kinect sensor calculates the correlation between points on the reference pattern *O* and points on the deformed pattern *K*. By this, depth can be calculated by the triangulation against a known pattern from the projector [39]. The depth frame created by the Kinect sensor is from the Kinect IR frame, and there is a fixed offset between these two data frames.

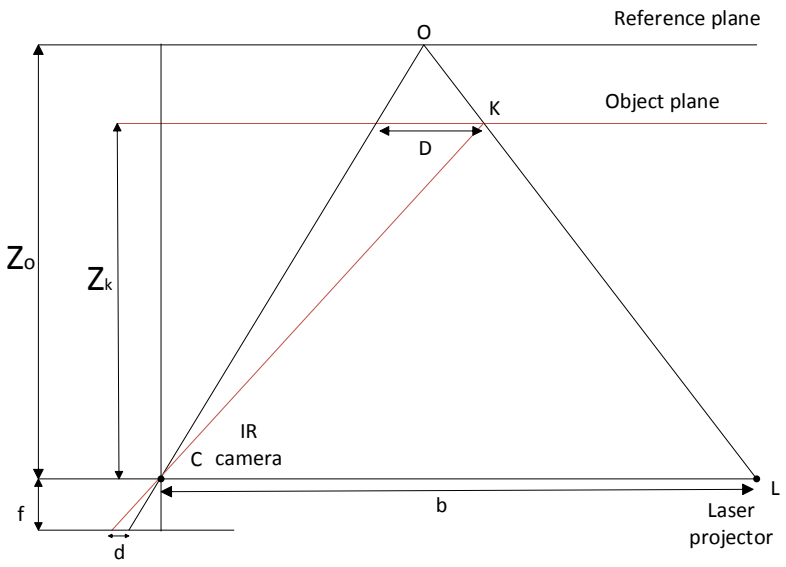


Figure 2.6: Relation between relative depth and measured disparity, reprinted from [42]

2.2.3.2 Accuracy of Kinect

The accuracy and density of Kinect depth data (point cloud) are important measures for evaluating the device. The error and imperfection of Kinect data can be caused by the properties of the sensor, the system setup and the condition of the object surface [42].

The expected error on Kinect depth measurements is proportional to distance squared, because the Kinect is essentially a stereo camera. The increased distance to Kinect caused random depth error. Figure 2.7 below shows how far off the Kinect sensor measurements are from that average distance [43]; this data are measured using a RANSAC-based method, this method extracts planes through the measured point cloud and calculates the distance of the points in the point cloud to the plane. The error is minimized when the distance is around 0.5 to 1.0 meter.

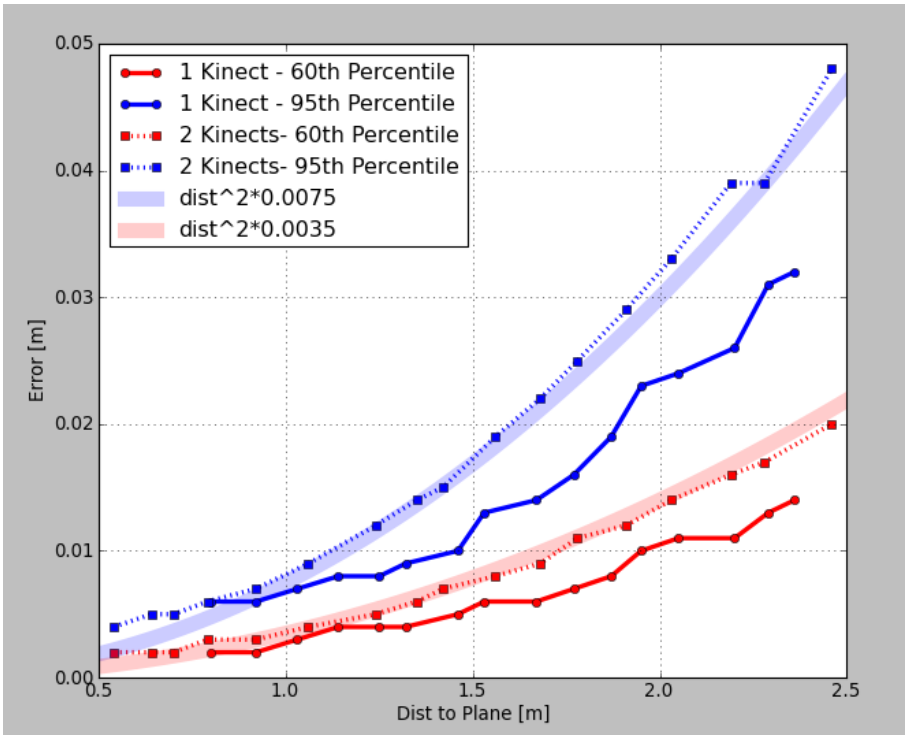


Figure 2.7: Precision of Kinect Sensor, relationship between distance to Plane and the error, reprinted from [43].

The measurement error can also be caused by the lighting condition, since Kinect adopted a laser projector and sensor to measure depth, the different lighting conditions affect the amount of laser speckles on the object surface; in practice, in a strong light field the laser speckle should appear in low contrast in the infrared image [42].

The accuracy of Kinect is directly affected by its intrinsic and extrinsic parameters. One group had compared the Kinect depth image and a calibrated monocular camera with a chessboard, and measures the distance with a measuring tape [43]. The collected data demonstrate that the accuracy of Kinect is ±1mm [43].

2.2.4 Time-of-Flight Depth Sensor-Kinect V2

Microsoft released Kinect V2, which is designed for the Xbox One video game console, in July 15th 2014. As an all-new designed Natural User Interface (NUI) controller, Kinect V2 adopts Microsoft's proprietary ToF technology, which is an entirely different technology of acquiring depth data than the structured light approach of the first generation Kinect sensor. The working principle of ToF had been introduced in Section 2.2.2; it estimates the depth by measuring the round-trip time (RRT) of the photons that are emitted by the sensor and reflected back. Kinect V2 has a depth sensor with resolution of 512×424 pixels. With higher depth fidelity and a significantly improved noise floor, the new Kinect has a better ability of 3D visualization, smaller objects detection and skeletal tracking [44]. Kinect V2 has a 1080p camera with the resolution up to 1920×1080, its new camera provides higher quality RGB image than the first generation of Kinect. Figure 2.8 illuminates the depth frame comparison between Kinect V1 and V2.

Figure 2.8: A comparison between the depth frames from Kinect v1 and v2 of a same scene, Kinect v2 provides higher depth fidelity than Kinect v1, left) depth image from Kinect v1, right) depth image from Kinect v2 [44].

2.3 Data Analysis

This section introduces the background knowledge about the kernel techniques that are adopted in the proposed system, such as image registration, edge detection, mathematical morphology and some geometry knowledge about spline curve.

2.3.1 Point Cloud Processing

Point cloud is a 3D data format, it records *X*, *Y* and *Z* coordinates of a set of points which are collected by engineering measurement. Point cloud is data collection of the surface appearance of an object. And point cloud also records RGB data and intensity information. Point cloud can be directly rendered and inspected [45]; it is used to represent a digital model of several objects, such as terrain, building, city model, human body, et… [46]. With the advent of new and low-cost depth sensors such as Kinect, and continued efforts in advanced point cloud processing, this 3D data format has become more and more popular and important in modeling, robotics and some other fields [47].

2.3.1.1 Point Cloud Alignment

Point cloud alignment is a type of 3D data registration technique. The purposes of point cloud alignment are registering the overlap part of the two point clouds from different capture sensors in different positions or viewpoints, merging point clouds in a same coordinate and obtaining a relative position and orientation between them. The point cloud alignment techniques are usually based on feature points. The general method search a group of corresponding feature points in both point clouds, and computes the relative transformation between these corresponding points. The search of the corresponding feature points is the important topic of point cloud alignment.

Rusu et al. [48] present a method for computing persistent point feature histograms for the corresponding search in 3D point clouds. They also propose two novel 3D robust features which characterize the local geometry around a point named Point Feature Histogram (PFH) and Fast PFH (FPFH) to be candidates for point correspondence search in 3D point cloud alignment [49]. Rabbani et. al [50] propose an automatic point cloud registration method which efficiently searches corresponding objects, and uses the corresponding objects as features to register the target objects. This method reduces the manual labour for point cloud registration, though it cannot be applied to all scenes.

Another category of point cloud alignment approach is frame-alignment-based, which sets up a calibrated capture camera frame. The relative position and orientation of cameras' poses are estimated during the frame alignment, therefore the system can register the 3D point clouds from the fixed camera. Zhao et al. [51] demonstrate a general framework and results using frame alignment for automatic video-to-3D registration.

2.3.1.2 Iterative Closest Point Algorithm

Iterative closest point (ICP) algorithm is presented by Besl et al. [52]. ICP is designed as an automatic algorithm for solving the 3D shape alignment problem. This algorithm is extensively used and developed in the point cloud (depth data) alignment field. And it only requires a procedure to find the closest point on a geometric entity, and it revises the transformation of the source point cloud to best match the reference point cloud whose location is fixed during this procedure. The procedure is iterative invoked until the distance between the two point clouds is minimized. The inputs of the algorithm are source point cloud and reference point cloud, and the

output is the transformation between the two point clouds. Essentially, every iterative procedure includes three steps [53]:

1. For each point in the source point cloud, find the closest point in the reference point cloud.
2. Estimating the transformation using a mean squared error cost function that will best align each source point to its match found in the previous step.
3. Transforming the source points using the obtained transformation.

A modified K-D tree algorithm for efficient closest point computation is proposed by Zhang [54]. Chen et al. [56] present the point-to-point closest point matching method to develop the ICP algorithm. And Bergevin et al. [57] propose the point-to-plane closest point matching method, which usually performs better in practice than point-to-point type.

2.3.2 Image Registration

Image registration is the process that compares and matches the images which contain the same object. The images may be captured from different devices, times, depths and viewpoints [58]. Specifically, for a set of two images, image registration is used to find out the spatial transformation that can build the correspondence between the mappings of the same spatial point in two separate images for the purpose of information fusion. This technology is widely used in the application fields of computer vision, medical image processing, biological imaging and brain mapping, military automatic target recognition and material mechanics. Since 20^{th} century, medical imaging technology has developed rapidly from static to dynamic, from 2D to 3D. Combining a variety of images to display information on the same image, providing multiple

data mostly for clinical diagnostic image information, these have become highly valued technologies, and image registration is the key technique in this field.

There are two kinds of image registration algorithms: intensity-based and feature-based [59]. Pre-process procedure is necessary for the feature-based method, and this procedure consists of image segmentation and feature extraction. The feature-based method matches tracked characteristics between two images and establishes the spatial transformation them. There are some frequently used features, such as feature points (including corners, high curvature points, etc.), straight line segments, edge, contour, closed area and so on. Corner point is one of the frequently used image features. There is a very wide range of applications of the image corners in pattern recognition and computer vision.

Spatial transformation of image can be divided into rigid transformation and non-rigid transformation. Rigid transformation includes affine transformation, projection transformation and transformation curves. Non-rigid transformation includes radial basis function, physical continuum models and large deformation models.

Piecewise linear mapping function is a kind of rigid transformation, with the given coordinates of corresponding control points in two images of the same scene, the images can be divided into triangular regions by triangulating the control points[60]. The mapping algorithm can register each pair of corresponding triangular regions of the images and calculate a linear mapping function. An overall mapping function is obtained [60] after piecing together the linear mapping function. There were three steps to get the transformation using piecewise linear mapping function. Firstly the algorithm determines the triangulation in the reference image according to the selected feature points. The triangulation of the target image was obtained automatically through the corresponding control points. For the corresponding triangles in both

images, the mapping functions $X = f(x, y)$ and $Y = g(x, y)$ were determined and this pair of triangles were registered. Thus, for each pair of triangles, a transformation correspondence was calculated, and triangular area inside the triangles' vertexes was registered. The area outside the triangle area used determined function for mapping by extending the boundary triangle planes. Intersections of neighbouring planes limited the extent of the boundary planes that should be used in the extrapolation [60].

2.3.3 Edge Detection

Edge detection is the essential problem and fundamental tool of image processing and computer vision, particularly in the field of feature detection and feature extraction [61]. The purpose of the edge detection is to identify the obvious brightness variation points in the digital image. The significant changes of image attributes include the depth discontinuity, a discontinuous surface direction, changes of material properties and scene illumination. Edge detection greatly reduces the amount of data, removes the relevant information and retains the important structural attributes.

Most of edge detection methods can be divided into two major types: search-based and zero-crossing based [62]. A search-based method detects the image edge by finding out the maximum and minimum values of the first-order derivative. The boundary is usually positioned in the direction of the maximum gradient. A zero-crossing-based method detects the image edge by finding out the zero crossings in a second-order derivative expression which is obtained from the image.

Natural image edge is not always ideal step edges, instead they are usually affected by one or more factors listed below

- Focus blur is caused by the limited depth of the scene.
- Light generated fuzzy caused by the non-zero radius shadow penumbra.
- Shadow of the object smooth edges.
- Partially specular reflection or diffuse near the edge of the object.

Some researchers have used Gussian smoothing step as ideal step edge model edge blurring effect in the practical application of simple extension [63] [64][65].

2.3.3.1 Sobel Operator

Sobel operator is an operator which is mainly used for edge detection. It is a discrete differential operator for gradient approximation of computing the image brightness function. Using this operator at any point of the image will produce a corresponding gradient vector or its normal vector. This operator contains two groups of 3×3 matrixes which are horizontal and vertical. The approximations of the derivatives can be calculated by convolving the operator with the original image. If G_x and G_y are two images which contain the horizontal and vertical derivative approximations at each point, and A is the source image, G_x and G_y can be calculated by equations [66]:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \tag{2.1}$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \tag{2.2}$$

Horizontal and vertical gradient approximations of each pixel in the image can be combined with the following formula to calculate the magnitude of the gradient:

$$\mathrm{G} = \sqrt{{G_x}^2 + {G_y}^2} \tag{2.3}$$

And the gradient direction can be calculated by the formula:

$$\Theta = \arctan(\frac{G_y}{G_x}) \tag{2.4}$$

If Θ equals 0, that means there is a longitudinal edge in this part of the image and the left side is darker than the right side [67][68].

2.3.3.2 Flood Fill Algorithm

Fill algorithm is a classification of computer vision algorithm; it fills the internal pixels of specifying irregular region with fill colour in an image. Filling algorithm is extensively used in the field of computer-aided design and image processing, it includes injection fill area algorithm, seed filling algorithm and scan line filling algorithm.

The flood fill algorithm is a classical fill algorithm which extracts a number of connecting points from one region to separate with other neighbouring regions. It operates in the same way floods spread from one area to another. Flood fill algorithm changes the value of a pixel of the internal region to change the overall property of the region [69].

2.3.3.3 Mathematical Morphology

Mathematical morphology is a technique and theory in the geometrical structures analysis and process frequently used on digital images, graphs, surface meshes, solids, and many other spatial structures. Mathematical morphology is based on set theory, lattice theory, topology and random functions. It was originally developed for binary image and had been extended to grayscale

images and functions [70]. Erosion, dilation, opening and closing are basic operators of mathematical morphological [71].

Dilation operation probes and expands the shapes in the image with structuring element, it was first expanded to grayscale images and then to complete lattices. Dilation integrate all the background points connect with the dilated object, into the object, it expands the edge of the object and fills the holes inside. In binary morphology, dilation is a shift-invariant operator, strongly related to Minkowski addition [72].

Dilation inserts pixels along the boundary of objects in an image, while erosion removes pixels along object boundary. The amount of pixels that are inserted or removed depends on the size and shape of the structuring element. In the morphological dilation and erosion operations, the state of a given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbours in the input image.

2.3.4 Spline Curve

Spline curve is generated by a given set of control points, and the shape of the curve is controlled by these control points. A spline is defined by a numeric piecewise polynomial function that has a sufficiently high degree of smoothness at the polynomial pieces connect point, which is knot [73][74]. There are two usages of the spline: spline interpolation and spline approximation. Spline interpolation is usually used to digitize a drawing or animation design, and spline approximation is usually used to structure the object surface.

Spline interpolation is also a method of interpolation in the mathematical field of numerical analysis which is preferred over polynomial interpolation, because the interpolation error can be small even when using low degree polynomials for the spline [75]. A Bezier spline is a spline of

Bezier form. B-spline, which is a generalization of Bezier spline, is a special representation of spline, any spline function of given degree can be expressed as a linear combination of B-spline of that degree [76]. B-spline can be further extended to non-uniform rational B-spline (NURBS), which is used to construct a precise general geometry model.

Chapter 3. Data Capturing and Registration

This chapter gives the technical details about the data capture method with *Microsoft* Kinect sensor and High Resolution Camera (DSLR camera), and the process approaches of both depth and colour data.

For depth data (point cloud) capture, we set up a Kinect network using four Kinect sensors to capture 360 ° point cloud. The DSLR camera was introduced into the system to capture high resolution images from each viewpoint. After the data capturing, three kinds of data from four viewpoints were collected. They are point clouds, colour images from Kinect, and high resolution images which were captured by DSLR camera.

In order to build the correspondence between the 2D point in the high resolution image and 3D point in the point cloud, we registered the Kinect colour image and depth frame first, and then we registered the Kinect colour image and high resolution image.

3.1 Kinect Sensors Calibration

3.1.1 Kinect Network Setup

The Kinect sensor contains three major components: a 3D microphone, a RGB colour camera and a depth sensor. The depth sensor consists of an infrared emitter and an infrared sensor. These components constitute a multi-view system which is able to provide three kinds of data: RGB image, infrared image and depth image. To set up a collaborative group, which consists of several Kinect sensors, to capture real-life hair with 360 ° view, three objectives must be achieved: (i) projecting depth image to metric 3D space, (ii) precise mapping between colour and depth image, (iii) accurate estimating of the relative translation and orientation of each pair of two

Kinect sensors. The partial implementation of this part of the work was done by the post doctor alumnus Alberto Chavez, he introduced the idea of Kinect network and did some experimental work and we implemented the system based on his work.

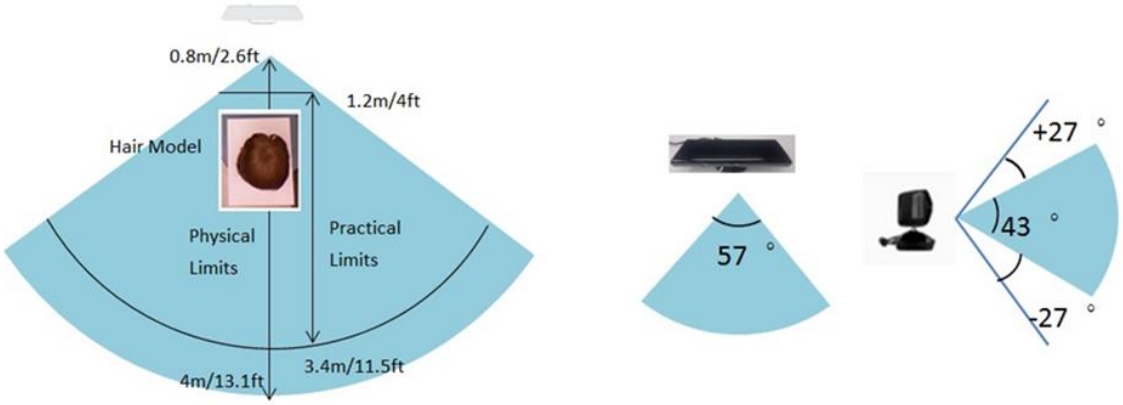


Figure 3.1: Microsoft Kinect Specifications. Left: the physical distance limitation is from 0.8m/2.6 ft to 4m/13.1 ft, the practical distance limitation is from1.2m/4 ft to 3.4m/11.5 ft. Right: the view angle is 57 degree horizontal and 43±27 degree vertical.

The work limits of Kinect are shown in Figure 3.1, the minimum capturing distance ranges from 0.8 m to 1.2 m, and it has 43 °vertical by 57 °horizontal field of view. In order to collect the data of the 360 ° view of hair, we designed a Kinect network which consists of four Kinect sensors, the devices were positioned on the four sides of the target hair model. The whole experiment framework is shown in Figure 3.2 below. The distance between the hair model and each Kinect was 1.2 m. The height from the ground of four Kinect sensors were same, the vertical view angle of them were same as well.

With the Kinect sensor collaborative group, the system collected depth and colour data of the target hair model from four viewpoints. Since the captured point clouds from different Kinect sensors were in the different coordinates, we adopted a calibration process to estimate the relative position and orientation between each pair of Kinect sensors, the point clouds were

integrated into the same coordinate and an 360 ° hair point cloud was generated in the point clouds alignment stage.

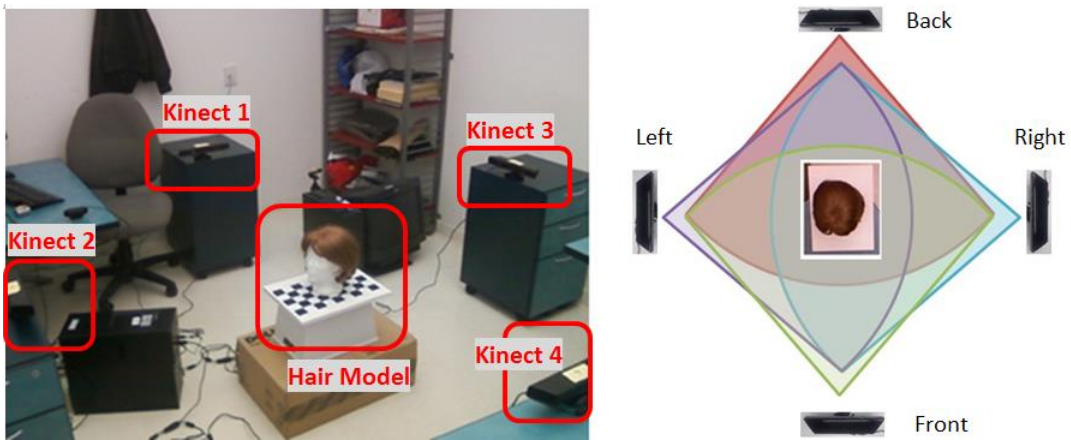


Figure 3.2: Experiment framework Setup. Left: real experimental setup by Alberto Chavez. Right: schematic plot of the data capturing system, the distance between head model and Kinect is 1.2 m.

3.1.2 Kinect Colour and Depth Data Registration

During the data capture stage, the Kinect sensor provided colour data (RGB image) and depth data (point cloud) streams separately, but it did not register colour and depth frames. Thus, it was necessary to perform a registration between the two data streams.

Before the Kinect colour and depth data registration, we estimated the intrinsic and extrinsic parameters of the build-in Kinect cameras using Zhang's [77] camera calibration method. The intrinsic parameters included focal length, principal point, and lens distortion coefficients of both colour and IR camera. The extrinsic parameter, which is estimated by stereo calibration, is the relative location of the colour camera and depth camera within each Kinect. The registration of colour and depth frame was based on the estimated internal intrinsic and extrinsic parameters of a given Kinect sensor, and the registration procedure is performed as follows: (i) First, the system removed the offset between the depth image and IR image, which was five pixels in the horizontal direction and four pixels in the vertical direction [38][30]. Then, the system deformed

the colour image and the calibrated depth images to correct the radial and tangential lens distortion using OpenCV [31] to register the colour image and depth image. (ii) Next, the system projected each pixel of the depth image to metric 3D space using the depth camera intrinsic parameter. Therefore, each pixel of the depth image exactly mapped a 3D point in the real-world 3D space, and a corresponding point cloud was generated. (iii) Last, the colour image was projected to the point cloud, thus the RGB information from the colour image was linked to the point cloud.

The Kinect calibration method, which is presented by Burrus[55] was referred for projecting depth image pixels to the 3D space, registering the colour image and 3D point cloud. Let x and y represent the pixel coordinate of the depth image and colour image, the 3D space coordinate values in the real-world x_R, y_R and z_R can be calculated by the following equations:

$$x_R = (x - c_{xd}) * depth * f_{xd} \tag{3.1}$$

$$y_R = (c_{yd} - y) * depth * f_{yd} \tag{3.2}$$

$$z_R = depth \tag{3.3}$$

Where focal length (f_{xd},f_{yd}) and principal point (c_{xd}, c_{yd}) are the intrinsic coefficients of the given Kinects depth camera, and $depth$ is the depth data from the depth image related to the (x, y) pixel. Then, in the registration procedure each pixel of colour image was projected to the point cloud by the following equations:

$$P_{3D} = \begin{bmatrix} x_R \\ y_R \\ z_R \\ 1 \end{bmatrix} * H \tag{3.4}$$

$$x_{rgb} = \frac{X_{P_{3D}}}{Z_{P_{3D}}} * f_{x_c} + c_{x_c} \tag{3.5}$$

$$y_{rgb} = \frac{Y_{P_{3D}}}{Z_{P_{3D}}} * f_{y_c} + c_{y_c} \tag{3.6}$$

Where *H* is the homogeneous transformation matrix (3*3 rotation matrix *R* and 3*1 translation matrix *T*) of the built-in Kinect cameras, and focal length (f_{x_c}, f_{y_c}) and principal point (c_{x_c}, c_{y_c}) are the intrinsic coefficients of Kinect RGB colour camera. By this method, the pixel (x_{rgb}, y_{rgb}) in the colour image was exactly mapped to the point (x_R, y_R, z_R) in the point cloud.

3.1.3 Calibration between two Kinect Sensors

The point clouds from four Kinect sensors in different locations were in separate 3D space coordinates. In order to merge them in a same coordinate, we performed a pairing calibration to estimate the relative location and orientation of each pair of Kinect sensors. The calibration process is shown in Figure 3.3. A chessboard was positioned in the overlap detection scope of the two Kinect sensors, and then the devices captured a pair of chessboard images and corresponding point clouds. The system defined the coordinates of the point clouds by finding the normal vector and the center of the chessboard plane, and then we can calculate the certain transformation (calibration parameter) between the coordinates of the Kinect sensors.

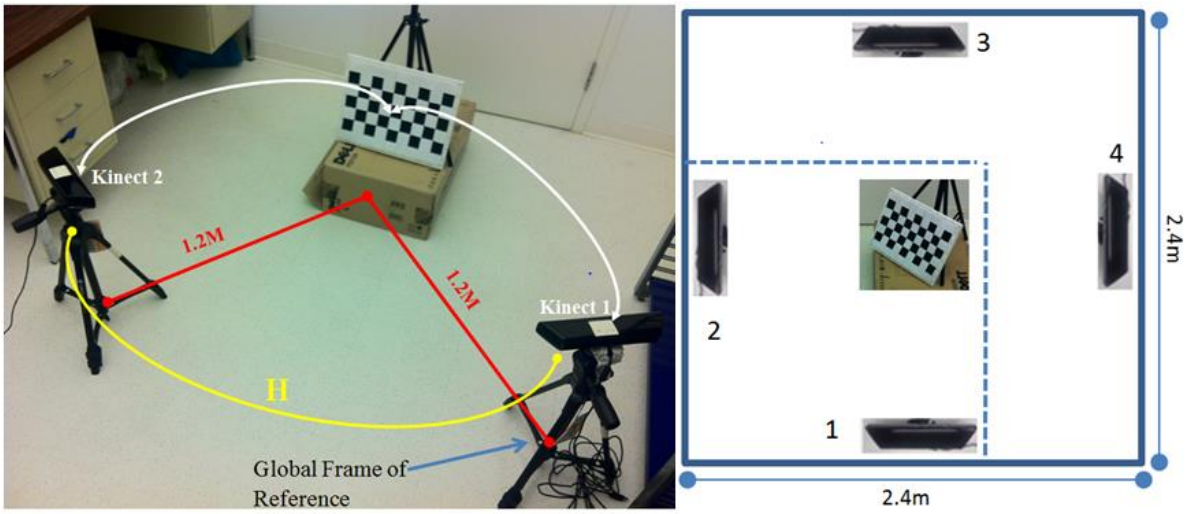


Figure 3.3: A calibration test of a pair of two Kinect sensors, *H* is the homogeneous transformation.

We adopted the calibration method, which is presented by Macknojia et al. [78]. We recognized the corner points of the chessboard in colour image, and their 3D positions were identified in the point cloud.

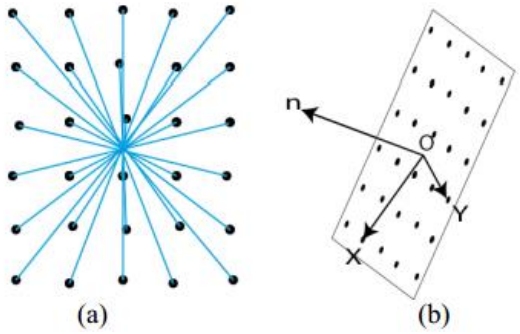


Figure 3.4: a) possible combination of lines passing through the center of the chessboard, b) the normal vector and the center of a chessboard target reprinted from [41].

As shown in Figure 3.4, this method defined the center point O by all the possible lines passing close to the center, the closest point to all intersection was selected as the center point. Two points X and Y were selected on the plane, and vectors $\overrightarrow{OX}$ and $\overrightarrow{OY}$ were defined, the normal $\vec{n}$ of the plane was defined by the equation:

$$\vec{n} = \overrightarrow{OX} \times \frac{\overrightarrow{OY}}{\left|\overrightarrow{OX} \times \overrightarrow{OY}\right|} \tag{3.7}$$

The relative orientation and the location between two Kinect sensors were calculated with the normal vectors and the plane center points, which were defined with respect to both Kinects sensors. Let the center points of the chessboard plane of Kinect 1 and Kinect 2 were O_1 and O_2, and normal vectors were $\overrightarrow{n_1}$ and $\overrightarrow{n_2}$, the rotation angle θ can be defined as:

$$\theta = \cos^{-1}(\overrightarrow{n_1} \cdot \overrightarrow{n_2}) \tag{3.8}$$

And the normal vector $\overrightarrow{V_n}$ of both vectors was the rotation axis, which is defined as:

$$\overrightarrow{V_n} = \overrightarrow{n_1} \cdot \overrightarrow{n_2} \tag{3.9}$$

The axis-angle can be defined in quaternion form as:

$$q = [q_0 q_1 q_2 q_3] \tag{3.10}$$

Where $q_0 = cos(\theta/2)$, $q_1 = \overrightarrow{V_{nx}} \sin(\theta/2)$, $q_2 = \overrightarrow{V_{ny}} \sin(\theta/2)$, and $q_3 = \overrightarrow{V_{nz}} \sin(\theta/2)$.

So that a rotation matrix R was calculated by the quaternion:

$$R = \begin{bmatrix} 1 - 2{q_2}^2 - 2{q_3}^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 1 - 2{q_1}^2 - 2{q_3}^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 1 - 2{q_1}^2 - 2{q_2}^2 \end{bmatrix} \tag{3.11}$$

And the translation T can be calculated with the center points O_1 and O_2 of the chessboard by the equation:

$$T = O_1 - RO_2 \tag{3.12}$$

With this method, the relative translation and rotation between the coordinates of each pair of Kinect sensors were estimated, and this method was expanded to the Kinect network, which consisted of four Kinects sensors. This calibration ran four times for all combinations of the two adjacent Kinect sensors. The calculated transformations of the experiment shown in Figure 3.2 are shown in the Table 1 below.

Table 1: Calibration parameters of the Kinect network.

Calibration Parameter for Kinect Network						
Sensors (Cali Pair)	Translation			Rotation		
	X	Y	Z	X	Y	Z
1,(1,2)	-9.34495	-3.78145	100	-0.64592	46.2937	4.47915
2,(2,3)	-0.24606	-2.13008	112	-0.50815	46.4532	3.31961
3,(3,4)	0.28278	-3.57201	99.3	1.61183	45.8776	2.72148
4,(4,1)	-1.82663	0.48801	105	0.55623	45.8518	5.75233

3.2 Point Cloud Alignment

In this stage, we processed each individual point cloud at first, a face tracking-based segmentation was utilized to remove the background of the point cloud, and a temporal filtering method was adopted to improve the quality of each point cloud. Then, with the obtained calibration parameters, we performed point clouds alignment to integrate the point clouds from the different captured device in one coordinate system. Besides the regular point cloud alignment, we also adopted the ICP algorithm to refine the alignment process. A sample of calibration-based plus ICP alignment of a pair of point clouds is shown in Figure 3.6. The partial implementation was done by the post doctor alumnus Alberto Chavez in CG++ group, he built the first version of test program for alignment and temporal filter, and we implemented the system based on his work.

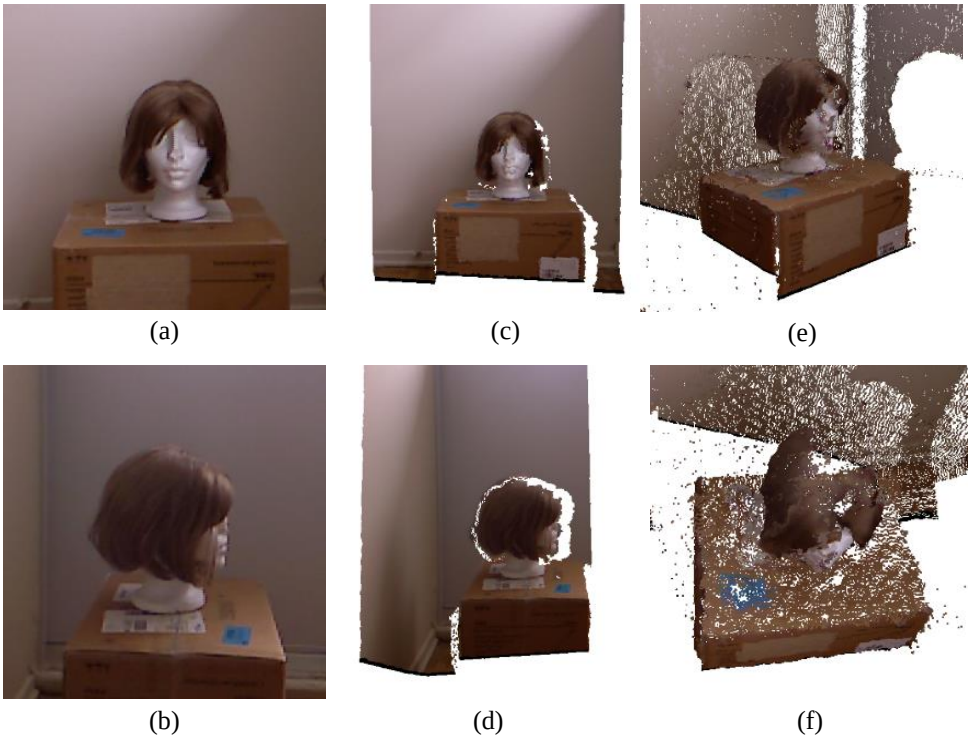


Figure 3.5: a) Front view captured by front Kinect, b) Left view captured by left Kinect , c) Point cloud from front Kinect, d) Point cloud from left Kinect, e)f) Alignment result of the two point clouds.

3.2.1 Head and Hair Region Segmentation based on Face Tracking

The maximum detection range of the Kinect sensor is three meters. The hair model in the experiment was placed 1.2 m from the sensor, as a result the background, which was useless for hair reconstruction, was also captured by the Kinect sensors. To remove the background points for reducing scale of the point cloud, we used a face tracking-based segmentation to segment the head region from the background automatically.

Microsoft Face Tracking Software Development Kit for Kinect (Face Tracking SDK) provides a face tracking function, it enables the Kinect sensor to track human face in real time (Figure 3.6). The face tracking engine analyzes the input from the Kinect and deduces the head

pose and facial expressions, it detects facial feature points in real time and acquires the depth information of the facial points.

(a) (b)

Figure 3.6: Face tracking on model with Kinect

The face-tracking function only works on the front viewpoint. We estimated a bounding box for the head region using the tracked facial points. Figure 3.7 shows the experimental sample, an average depth value $AveDepth$ of the facial points was calculated, and the front boundary equal $Avedepth - 15$, and the back boundary equal $AveDepth + 30$. The segmentation was carried out along Z-axis, while the X and Y axes remained pending for the solution, so that the head region was segmented from the background.

(a) (b) (c)

Figure 3.7: Face tracking-based segmentation, a) bounding box of head region, b) initial real-time point cloud, c) the background is removed after segmentation.

Since the four Kinect sensors were positioned around the head model symmetrically. In theory the relative location between the head model and Kinect sensors are the same. Thus, the system shared the segmentation parameter $AveDepth$ in the Kinect network, and the segmentation was applied to the point clouds from other viewpoints. The background removal point clouds are shown in Figure 3.8.

(a) (b) (c) (d)

Figure 3.8: Applying same segmentation to the point clouds from the four Kinect sensors.

3.2.2 Point Cloud Refinement

In order to generate a depth map of the scene, the Kinect emitter projects a pseudo random pattern (points) to detect the space object. Since the boundary and some part of the object that cannot reflect the infrared, cannot be detected by the depth sensor, the consecutive readings of Kinect may be different from each other. In order to refine the quality of the readings, a temporal filtering like method was adopted to clean up the signal and reduce the holes in the depth frame. We captured a sequence of depth frames, and the frames were overlapped, thus the gaps of one frame can be recovered by same part of other frames. For each point in the depth frame, we removed the null depth value and calculated an average value of other non-null values, after that a new integrate depth frame was generated. And the point clouds were enhanced using this temporal filtering method. Figure 3.9 demonstrates the mechanism of point cloud refinement.

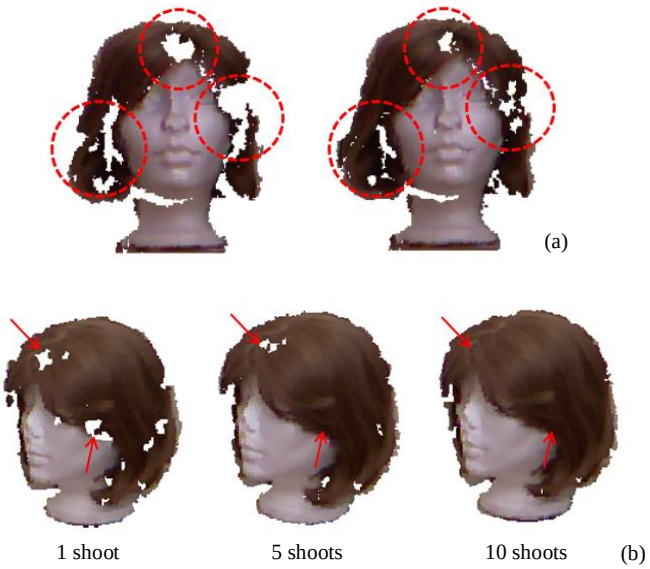


Figure 3.9: a) The point clouds of the same model captured in different time are not completely equivalent. b) Temporal filter refines the quality of the point cloud. With the increasing of shot times, the gaps indicated by red arrow are shrank.

3.2.3 ICP Point Cloud Registration

Iterative Closest Points (ICP) algorithm is used to minimize the differences between two point clouds using the geometric translation and rotation. The proposed system adopted the point-to-plane ICP algorithm to improve the point cloud alignment process. In ICP algorithm, one point cloud, the reference or target, kept fixed, while the other one, the source, was transformed to best match the reference.

The algorithm iteratively revised the transformation as translation and rotation needed to minimize the distance from the source to the reference point cloud. Due to the fact that the image planes of the reading were orthogonal, they shared many points. The rigid transformation between the two systems was calculated by the given coordinates of point clouds in two Cartesian coordinates.

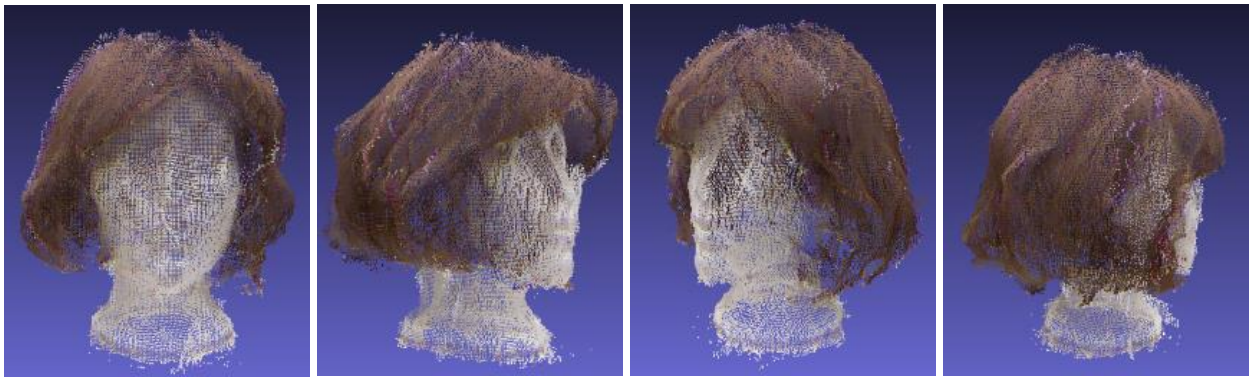
We applied the transformation to the point clouds from four viewpoints. In this procedure, ICP ran three times for front-left, front-right view and right-back registrations. The algorithm calculated three sets of transformation parameters which are shown in Figure 3.10, the transformations are:

$$T_1 = \begin{bmatrix} 0.0865 & -0.0530 & 0.9948 & -300.3875 \\ 0.0644 & 0.9968 & 0.0475 & -9.4369 \\ -0.9942 & 0.0600 & 0.0896 & 263.2604 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

$$T_2 = \begin{bmatrix} 0.1925 & 0.0190 & -0.9811 & 282.7549 \\ -0.0494 & 0.9987 & 0.0097 & -2.9876 \\ -0.9800 & 0.0466 & 0.1932 & 248.6811 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

$$T_3 = \begin{bmatrix} 0.0878 & 0.0331 & -0.9956 & 293.8087 \\ -0.0242 & 0.9992 & 0.0311 & -5.4365 \\ 0.9958 & 0.0213 & 0.0886 & 282.0514 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

Where T_1, $T_2$and T_3 are for front-left, front-right and right-back.



(a) (b) (c) (d)

Figure 3.10: Point clouds alignment result.

3.3 High Resolution Image and Point Cloud Registration

Since the resolution of the Kinect RGB colour camera is only up to $1280 * 960$, it could not meet the requirement of the image processing procedure for hair strand extraction. We adopted a

DSLR camera to capture high resolution hair image from each viewpoint for the data analysis stage. For the purpose of obtaining the 3D positions of the 2D hair strand control points in the data analysis phase, the system estimated the mapping relationship between the 2D point in the high resolution image and 3D point in the point cloud. This mapping relationship could be built through the correspondence between the depth data and colour data from Kinect sensor, shown in Figure 3.11.

Figure 3.11: Build the correspondence between depth data and high resolution image through the connections of depth data and colour data, and colour data and high resolution image.

Since a calibration procedure had been carried out to register the Kinect colour data and depth data, the system only needs to register the high resolution image and the Kinect colour image. Thus, the mapping relationship between the pixels in two images was obtained, and the 2D points in the high resolution image could be mapped to the 3D points of the point cloud as well. The system could obtain the real-world 3D position of each 2D pixel.

3.3.1 Kinect Colour Image and DSLR camera Image Registration

We introduced a Canon 500D DSLR camera into the system, it provided the high resolution images with resolution up to $4752 \times 3168\ pixels$. In the image analysis and process stage, the hair strands were extracted from this high resolution image. Figure 3.14 shows the comparison between the images from different devices.

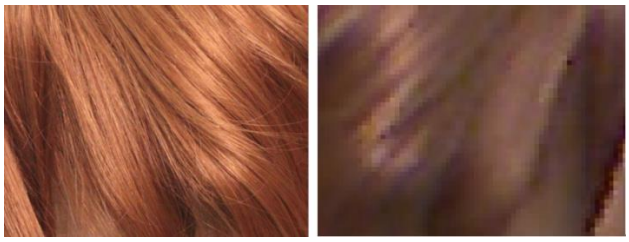
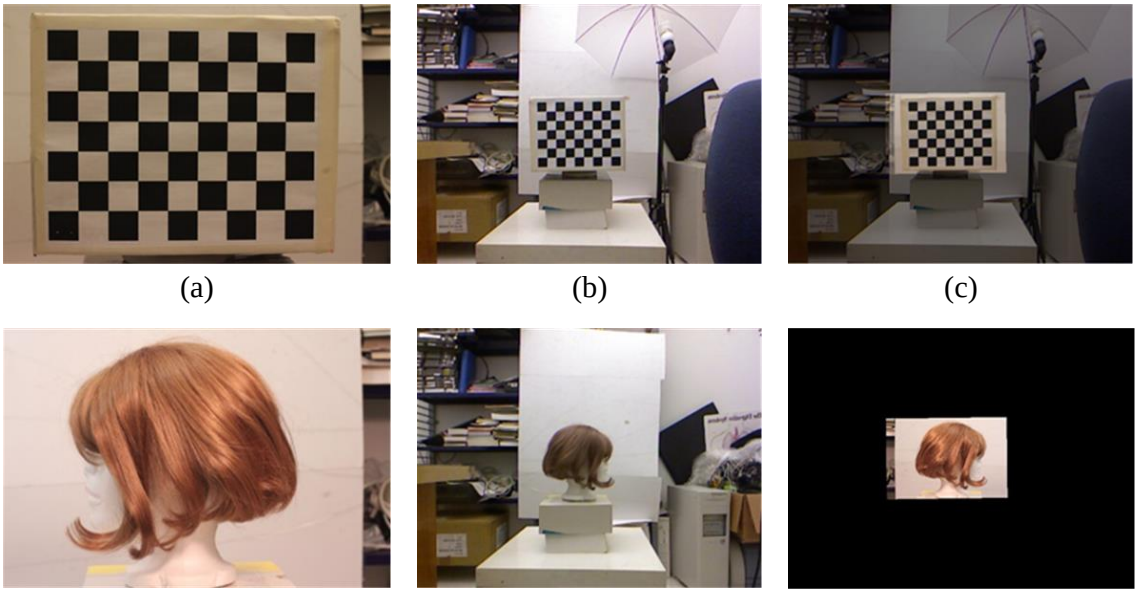


Figure 3.12: Same region images captured by DSLR camera (left) and Kinect colour camera (right).

Image registration is the process of overlaying two images of the same scene. We utilized the feature-based registration, which consists of two steps: (1) selecting and matching feature points in the images, (2) determining a mapping function that can match the rest of the points in the images using information of the matched feature points [58]. For the purpose of using this feature-base registration technique, the system captured two images of a calibration chessboard from each camera. Then the system recognized the corner points of both images and used them as the corresponding points for registration. Since the position and view angle of Kinect sensor and DSLR camera were fixed, the transformation, which was calculated with the chessboard image registration, could be applied to the subsequent hair images. The cameras' setup is shown in Figure 3.14 below.

Figure 3.14: The Kinect sensor and DSLR camera used in the experiment.

We adopted piecewise linear transformation during the registration to estimate the transformation between two images. This transformation algorithm can be used when parts of the image appear distorted differently. After the transformation between the two chessboard images was calculated, the system applied this transformation to the high resolution hair images captured by the DSLR camera. This registration only localized in the chessboard area and Figure 3.16 shows the procedure of the image registration.



(a) (b) (c)

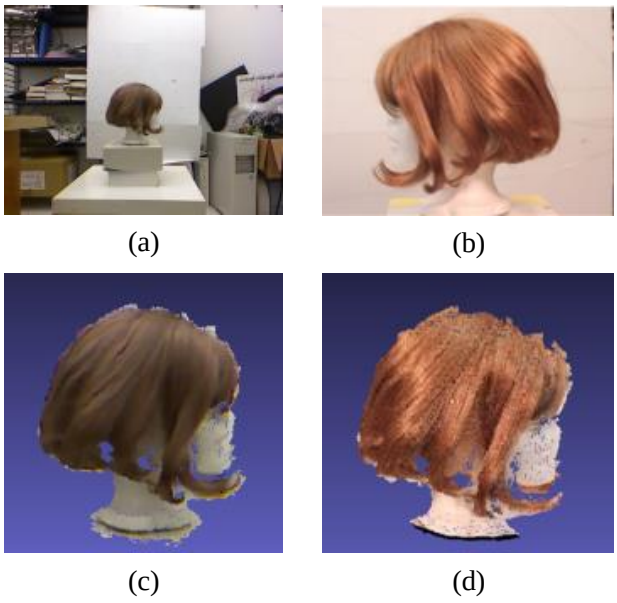
(d) (e) (f)

Figure 3.15: a) d) The high resolution images of chessboard and hair model captured by DSLR camera. b) e) Images of same scene captured by Kinect colour camera. c) Image registration result of two images. f) Applying registration transformation to high resolution image of hair model.

3.3.2 DSLR camera Image and Point Cloud Registration

After high resolution image and Kinect colour image registration, the system was able to project the high resolution image on the point cloud. For each 2D point in the high resolution image, the corresponding 3D point can be identified.

During the capture phase, the Kinect captured a colour frame for each point cloud. We maintained a vector to save the mapping between 3D point and 2D point, and a struct to save the point cloud data. For each 3D point, the struct saved the 3D coordinates xyz and corresponding image pixel coordinates XY, and colour data RGB. Because the size of the struct is 640×480 same as the resolution of Kinect frame, in theory it could save up to 37200 points for one frame. Thus, we can easily identify the 3D data for a 2D point in the high resolution with this struct which save the mapping relationship between 2D and 3D.



(a) (b)

(c) (d)

Figure 3.16: a) The Kinect colour image b) The high resolution image c) The point cloud with Kinect colour data. d) The point cloud with the high resolution image.

Figure 3.16 demonstrates the mapping results. The point cloud with the high resolution image reveals more details in the hair region. In order to map the high resolution image on the point cloud, high resolution image was compressed to 640×480, and this process resulted in data loss.

Chapter 4. Data Analysis and 3D Hair Strand Modeling

This chapter describes the details about the data extraction and analysis stage. In this stage, there are three steps of data processing: (i) we used the image processing-based method to segment the hair area from the background and to remove the facial part in both 2D images and 3D point clouds. (ii) According to the obtained hair point cloud, we estimated the center point position and radius of a spherical skull. (iii) We analyzed the separate hair strand segments, built connection between hair strand segments and interpolated points in the connection area.

The hair strand extraction is an important part of the system. In the project, we adopted image processing-based techniques for automatic 2D hair strands extraction, however the details of automatic method are not included in this thesis. Instead we used a user interactive tool for the manual hair strand extraction.

In 2D image, some hair strands may be partially occluded by other hair strands, and these hair strands were segmented into two segments. Most of time the algorithm could not track a hair strand from root to tip to extract a continuous 2D hair strand. In order to connect these segments and recover lost data in the occlusion area. We designed algorithms to analyze and build the connection between hair strand segments, and we also interpolated points in the connection area.

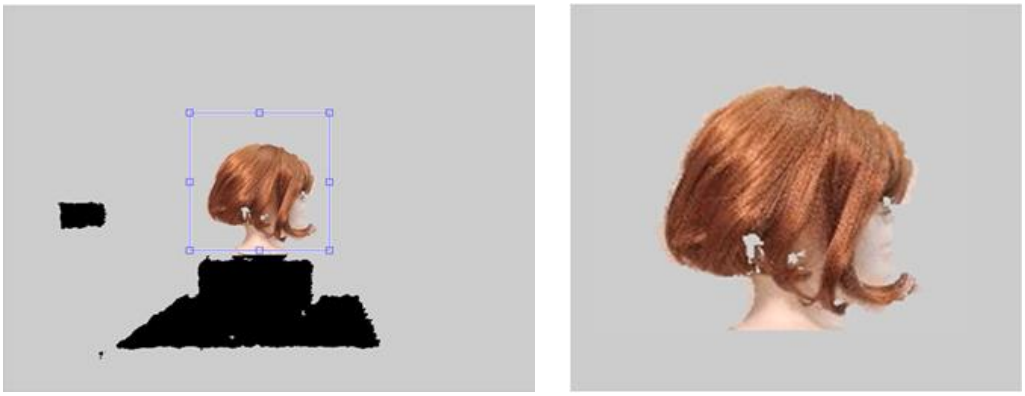
4.1 Hair Area Segmentation

We adopted the edge detection on 2D image to track the hair area contour and segmented the hair area. Firstly, we projected the point cloud, where the background had been removed already, to 2D plan and generated a background removal colour image. Afterwards, a user interactive tool was employed to help select an approximately hair area in the image, and then an edge detection technique was applied to this area to detect the hair contour. After that, the point cloud was

segmented based on the mapping relationship between 2D image and point cloud. The corresponding 3D points inside the hair contour were extracted from the point cloud. The unnecessary points, such as the points of the facial area and neck area, were removed. So that the hair strands can be extracted in the segmented hair image and the corresponding 3D hair strands were obtained from segmented point cloud. The scale of the target point cloud was reduced to help improve the system`s processing speed.

4.1.1 Image Background Removal

For each viewpoint, there were three kinds of data: Kinect colour image, Kinect depth data (point cloud) and DSLR image. In the point cloud, the background points were removed using the face-tracking-based segmentation technique, and we projected the coloured point cloud on the image plane to generate a background removal image inversely. In this step, the background was replaced with the low grayscale pixels, however, the points inside the bonding box such as the face and neck parts were retained as well. In order to remove these points, and detect the hair edge easily, a user interactive tool was utilized to select the major hair and head area in the image, this tool provides user a rectangular box to for segmentation, as shown in Figure 4.1. Therefore, a cropped image which only contained hair and head region was generated for the next edge detection phase.



(a) (b)

Figure 4.1: a) the background removal image, the head area is cropped with a rectangular box, b) cropped hair image.

4.1.2 Sobel Operator-based Edge Detection

After cropping the target image, the system converted the image to a binary image using thresholding. The threshold, which was related to the properties of the original image, affected the conversion result. For different hair and skin colour, the differentiation between colour of the facial and neck area and background is different; therefore, the proper threshold could filter the non-hair pixel in the binary image during binary conversion process.

Then, the Sobel-based edge detection was performed in the binary image to detect the hair contour, and the edge detection procedure is shown in Figure 4.2 below. Because the Sobel operator is a discrete differentiation operator, the gradient change around the hair edge and inside the hair region could be detected by this operator. The edges along the hair contour and some other edges with obvious gradient change inside were recognized.

A close area is a set of background pixels that cannot be reached by filling in the background from the edge of the image [79]. The hair area was the largest close area in the image, the hair contour which was detected in the previous Sobel-based detection is not always a closed curve, there were some disturbed factors like small detected edges around the contour. In practice, the

system could not directly determine the final hair contour after the first time Sobel-based detection, unnecessary short edges always exist inside the hair contour and, above all, the detected hair contour was not a closed curve. In order to close the hair contour, the image dilation algorithm was employed to insert pixels along the hair contour edge and we adopted flood fill algorithm to fill hair contour inside. Thus, a unique, clear and closed hair contour edge can be recognized by the second time Sobel-based edge detection. For the complex condition in the hair area, the Sobel detection, dilation algorithm and flood fill algorithm may need to be invoked iteratively.

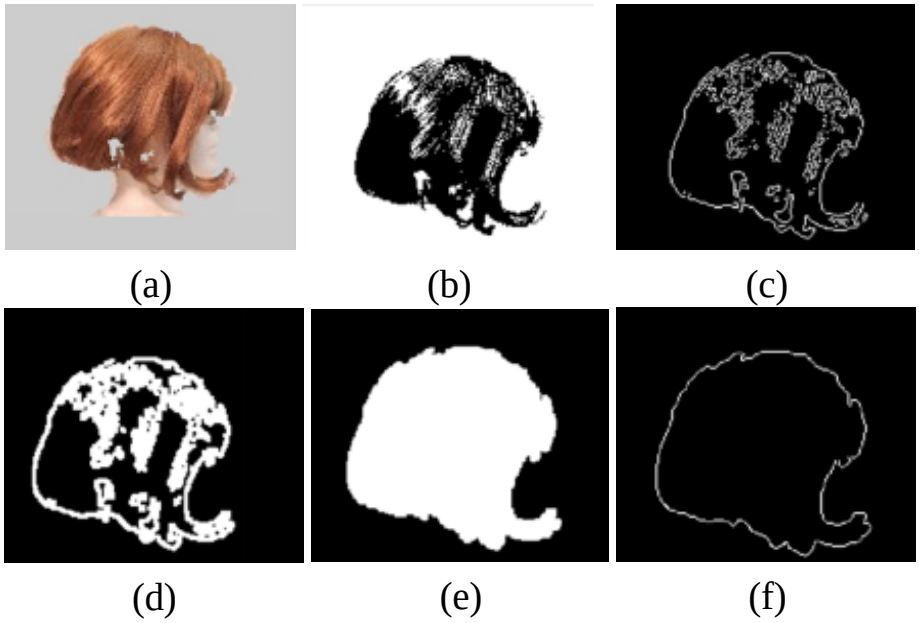


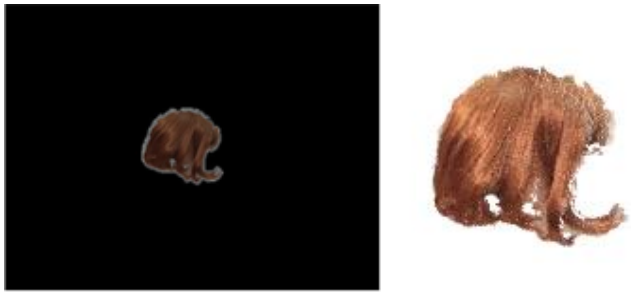
Figure 4.2: The hair contour detection using Sobel operator, a) the original input image, b) binary image using threshold-based conversion, c) first Sobel-based edge detection, d) morphology dilation result, e) flood fill algorithm, f) the final hair contour.

4.1.3 Hair Area Segmentation

The hair area extraction in the 2D image could be directly and easily implemented using the detected final hair contour. The system obtained the position of all pixels along the hair contour

by searching the pixels that had the binary value 1(white colour) in the binary hair contour image. The flood fill algorithm changed the pixels inside the hair area to binary value “1” (white colour), while the binary value of outside pixels are all “0” (black colour), thus the system could distinguish the 2D hair area points in the image.

In order to perform the 3D segmentation, for each 2D hair point, we acquired its corresponding 3D point from point cloud using the mapping relationship between 2D and 3D. Eventually, the 3D hair area was extracted from the point cloud. Figure 4.3 shows the hair area segmentation on 2D image and 3D point cloud.



(a) (b)

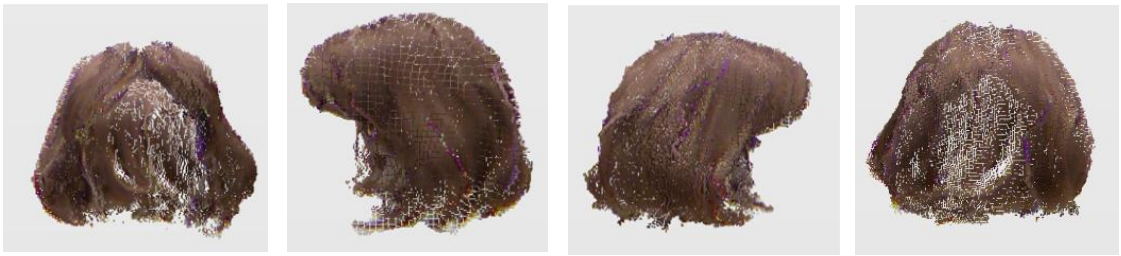
Figure 4.3: a) Hair area segmentation in 2D image, the pixels outside the hair contour are all black, b) segmented hair area point cloud from a single viewpoint point cloud.

4.2 Skull Generation

In order to create a hair particle system, a particle emitter is required because the hair growth is dependent on the surface of the emitter. Thus, a head object was created before the hair particle system generation phase, and the pose of the head object was estimated based on the 3D hair point cloud. To reduce the complexity of the system, a sphere was supposed to represent the head in the particle system, and its position and radius were the only requirements.

4.2.1 Hair Area Point Cloud

Since the size and position of the sphere were related to the hair point cloud, determination of the basic spherical skull's attributes involved the use of the complete hair point cloud. The hair point clouds were extracted from the point cloud of each viewpoint. The point cloud alignment was performed with the four hair area point clouds to generate a complete hair point cloud (shown in Figure 4.4). The radius and center point position of the spherical skull could be calculated with this complete hair area point cloud, and the sphere was generated inside the hair area point cloud to be wrapped by the hair point cloud.



(a) (b) (c) (d)

Figure 4.4: the complete hair area point cloud of four viewpoints, a) front view, b) back view, c) right view, d) left view.

4.2.2 Spherical Skull Attributes

There are two basic attributes of a sphere, radius and center position, since the posture or the orientation can be ignored. These attributes were related to the size and position of the hair point cloud. Since the hair strands particles were always emitted from the scalp area which is the top area of the spherical skull, the system only consider the top area of the hair point which covered the scalp area.

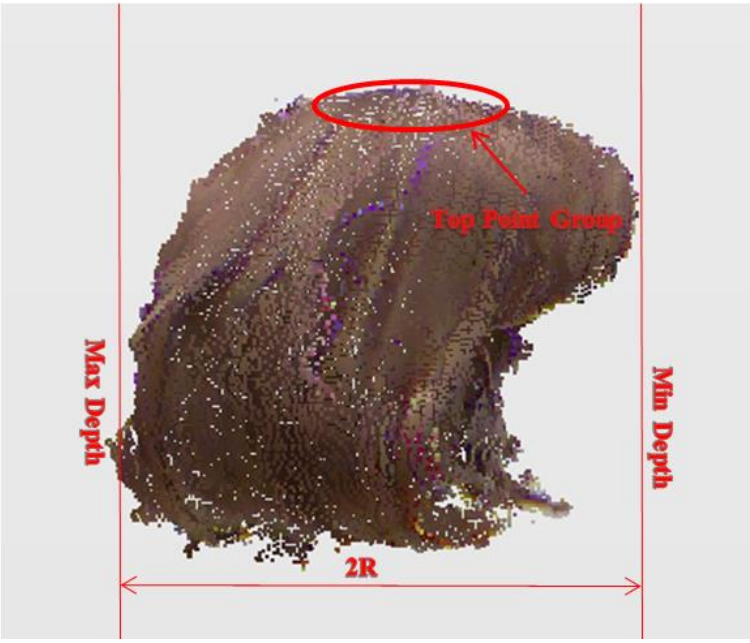


Figure 4.5: Illumination of related features of spherical skull on the left side hair point cloud.

As shown in Figure 4.5, we analyzed the sagittal plane of the hair point cloud. The diameter of the skull sphere is equal to the width of the sagittal plane of the hair point cloud. Let $MaxDepth$ and $MinDepth$ to represent the maximum and minimum depth value, the radius R and the Z-axis value of the sphere center point $CentZ$ can be calculated by equations:

$$CentZ = (MaxDepth + MinDepth)/2 \tag{4.1}$$

$$R = (MaxDepth - MinDepth)/2 \tag{4.2}$$

The X-axis value of the sphere center point $CentX$ is equal to the average X-axis value of the points that are located in the top area of the hair point cloud, and they were named top points group. We defined $MaxY$ as the maximum Y-axis value of the point cloud, and the top points group consisted of all the points whose Y-axis value is greater than $MaxY - 1$, and we calculated an average X-axis value $AvgX$ of these points. So the X-axis value of the sphere center point $CentX$ was defined as:

$$CentX = AvgX \tag{4.3}$$

The Y-axis value of the sphere center point $CentY$ equals the average Y-axis value of the top point group of the hair point cloud minus radius. Let $AvgY$ be the average Y-axis value of the top point group, the Y-axis value of the sphere center point $CentY$ was defined as:

$$CentY = AvgY - R \tag{4.4}$$

With the calculated radius and center point position, the spherical skull can be created during the hair particle system generation. Figure 4.6 shows the sphere with the hair point cloud.

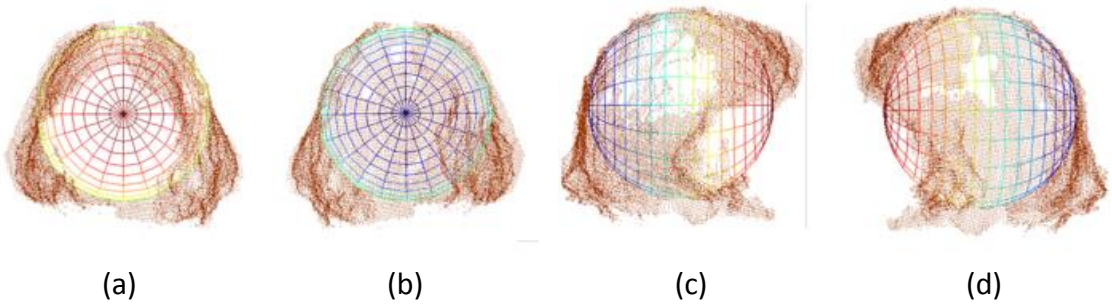


Figure 4.6: An inserted spherical skull inside the hair point cloud, a) front view, b) back view, c) left view, d) right view.

4.3 Connection Analysis between Hair Strand Segments

In some case, some hair strands were occluded by some other hair strands in the image. The hair strands were separated into two segments by other hair strand in this occlusion situation. In 2D image, this kind of hair strand was regarded as two strand segments, and the hair strand information in occlusion area was lost. In order to generate continuous 3D hair strands, a rough algorithm was presented to analyze the properties of the strand segments and build the connection between them. In addition, a point interpolation algorithm was presented to interpolate control points in the blank connection area. In this procedure, the hair strand segments were merged into other hair strands.

This part of the work was based on the assumption that hair strands can be tracked automatically using image processing techniques. We adopted manual method instead of automatic method for hair strand extraction in this thesis. The hair strands were manually selected in the high resolution image using a user interactive tool.

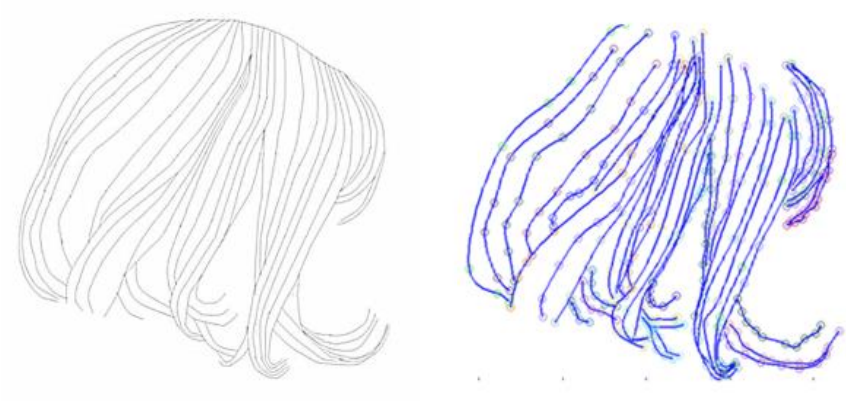
4.3.1 Hair Strands Control Points

In the experiment, a hair strand map was drawn manually based in the high resolution image. On the strand map, the strokes represented the recognisable hair strands in the hair image. As shown in Figure 4.7, this map serves as the result of the 2D hair strand extraction.

(a) (b)

Figure 4.7: a) The original hair image from DSLR camera, b) the manually drawn hair edge map based on the original hair image.

A user interactive tool was used to select the control points along each recognisable hair strand on the hair strand map. For each hair strand, a number of control points were selected, and the control points were uniformly distributed on the hair strand from root to tip, as shown in Figure 4.8. The system obtained position of each control point in 2D space. With the mapping relationship between 2D and 3D, the 3D information of each control point was obtained.



(a) (b)

Figure 4.8: a) The horizontal flip of the manual draw hair edge map, b) manually selected control points on the hair strand, and the points constitute hair curves, the points are 3D points from the hair point cloud.

4.3.2 Space Curve Generation and Hair Strand Classification

Our system generated space spline curves based on the 3D control points to fit the hair strands. Piecewise polynomial interpolation was used to generate a parametric variational spline curve passing through all the control points of a hair strand. The parameter value $t(j)$ of the j th point was chosen by Eugene Lee's [80] centripetal scheme. Then the system analyzed the properties of curves, such as slope, curvature, and straight distance etc.

A simple experimental sample was designed for simplifying the situation. The hairstyle sample in the experiment was shown in Figure 4.7, in which only a few hair strands were occluded by other hair strands. In this occlusion situation, the upper layer hair strands cover some part of the lower layer hair strands and the lower layer hair strands were separated into two strand segments. The hair strands could be classified in two types for this occlusion: 1) the strand segment, which was the tip end segment of the hair strand, was defined as piece strand; 2) the other strands that grown from the root end include strand segments and complete strands, defined

as reference strand. The head end of reference strand was always located in the top area of the head, and the piece strand was exactly opposite.

We designed an algorithm for hair strand segment classification, it was used to determine whether a strand is a piece strand or reference strand. According to the capture method, the real–world vertical orientation and the horizontal orientation were represented by Y-axis and X-axis in 3D space respectively. The algorithm defined the strand's type by analyzing the position of the strand`s head end. The Y-axis value was used as the feature value and defined as the height value of the strand control point in the determination process. For general situation, the hair strands occlusion situation always happened around the lower part of the hair structure. Thus the strand head end height value could be used to distinguish the strand type, piece or reference.

The algorithm computed the average height of all hair strands' head ends, so a boundary value can be calculated by the equation:

$$B \qquad ary = \frac{\sum Y_{headPoint}}{n} - \delta \tag{4.5}$$

where n is the hair strand's amount in calculation, and δ is a floating parameter. For different size of hair point cloud, we set the proper value of floating parameter δ. The strand whose height value (Y-axis value) of head end is higher than the boundary value is reference strand, otherwise the strand is piece strand.

4.3.3 Connection Analysis

After the hair strand classification, the hair strands were divided into two groups: piece strands and reference strands. The connection analysis algorithm focused on the piece strands. A brief method, used to identify the proper connections between piece strands and reference strands is described below.

Step1: The system calculated a connection-weight value of connection for each piece strand with every reference strand.

Step2: Then for the target piece strand, linked it to the reference strand which it had the highest connection-weight with.

Step3: For the reference strand that connected with more than one piece strand, kept it linked with the piece strand which it had the highest connection-weight value with, and removed the other links. The remained link was a confirmed connection, and the strands (piece strand and reference strand) in this connection quit the next round analysis.

Step4: Removed the reference and piece strands of the determinate connection from their groups.

Iteratively invoked Step1 to Step4 until the size of the piece strand group became zero, so that all piece strands had connected to a reference strand. This is a two-way selection algorithm, for a piece strand, it connected to the reference strand which it had the highest connection-weight with, and for reference strand, it connected to the piece strand which it had highest connection-weight with.

The following subsections describe the measurement of connection-weight, the point interpolation strategy and the curve smoothing approaches.

4.3.3.1 Connection Weight

The connection-weight was introduced into the determination process as a value that can measure the possibility of the connection between two strands (one reference strand and one piece strand). The value of connection-weight presented that how high possibility is of the connection exists between the two strands. To calculate the connection-weight of two strands, the system analyzed the curve fitting function of hair strand. In 2D space, a curve can be defined

by the curvature at each point, and the shape of a 3D space curve is well defined by curvature and torsion at each point.

Curvature of a plane curve at a point is a measure of how sensitive its tangent line is to move the point to other nearby points [81]. For a plane curve that has a parametric equation $c(t) = (x(t), y(t))$, the curvature κ_{2D} of this plane curve on the point (x, y) can be calculated by the equation:

$$\kappa_{2D} = \frac{|x'(t)y''(t) - x''(t)y'(t)|}{(x'^2(t) + y'^2(t))^{3/2}} \tag{4.6}$$

As in the case of 2D curves, the curvature of regular space curve in the 3D is the magnitude of the acceleration of a particle moving with unit speed along a curve. For a space curve with parametrical definition given in Cartesian coordinates by $\gamma(t) = (x(t), y(t), z(t))$, the curvature κ_{3D} on point (x, y, z) can be calculated by equation:

$$\kappa_{3D} = \frac{\sqrt{(z''y' - y''z')^2 + (x''z' - z''x')^2 + (y''x' - x''y')^2}}{(x'^2 + y'^2 + z'^2)^{3/2}} \tag{4.7}$$

And the torsion of a curve measures how sharply it is twisting out of the plane of curvature [82]. Taken together, the curvature and the torsion of a space curve are analogous to the curvature of a plane curve [82]. The torsion τ of a space curve on point (x, y, z) can be calculated by the formula:

$$\tau = \frac{x'''(y'z'' - y''z') + y'''(x''z' - x'z'') + z'''(x'y'' - x''y')}{(y'z'' - y''z')^2 + (x''z' - x'z'')^2 + (x'y'' - x''y')^2} \tag{4.8}$$

Slope is also an important concept related to the strand orientation in 2D. Slope of curve describes both the direction and the steepness of the 2D curve at a point. And slope m of the curve at point (x, y) can be calculated by the equation:

$$m = \frac{dy}{dx} \tag{4.9}$$

Meanwhile, the algorithm calculating the straight-line distance between the tail end and the head end of two curves in 2D and 3D space. The distance d_{2D} , d_{3D} between two points (x_1, y_1) and (x_2, y_2)in 2D, and (x_1, y_1, z_1) and (x_2, y_2, z_2) in 3D can be computed by the equations:

$$d_{2D} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4.10}$$

$$d_{3D} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{4.11}$$

The algorithm analyzed the connection-weight of two curves using these properties. For reference strand curve C_r , it has slope S_{rt} and curvature κ_{rt} on the tail end, and for piece strand curve C_p, it has slope S_{ph} and curvature κ_{ph} on the head end. And the distance between the reference tail end and piece head end is d. A rough connection-weight *w* was defined by the equation:

$$w = \alpha / \left| S_{rt} - S_{ph} \right| + \frac{\beta}{d} + \gamma / \left| \kappa_{rt} - \kappa_{ph} \right| \tag{4.12}$$

Where α, β, γ are the coefficients, these coefficients were related to the data of the points. By adjusting the coefficients, the connection-weight was calculated. Therefore, the system measured the connection-weight between two space curves. It should to be noted that concerned the precision of depth data from Kinect sensor, the connection analysis algorithm ignored some depth information of the space curves. It can be analogous to a plane curve, which was the projection of the original 3D space curve on the zero-depth plane. Only end point slope, plane curvature and space straight-line distance were under consideration.

4.3.3.2 Points Interpolation

By applying the connection analysis algorithm, the system identified the connection between a piece strand curve and a reference strand curve. However, the data loss existed in the occlusion region between the two strand segments. Thus, in order to connect two curves smoothly and naturally, a point interpolation procedure was presented to fill the blank with some points. The algorithm of interpolation is described below (Figure 4.9).

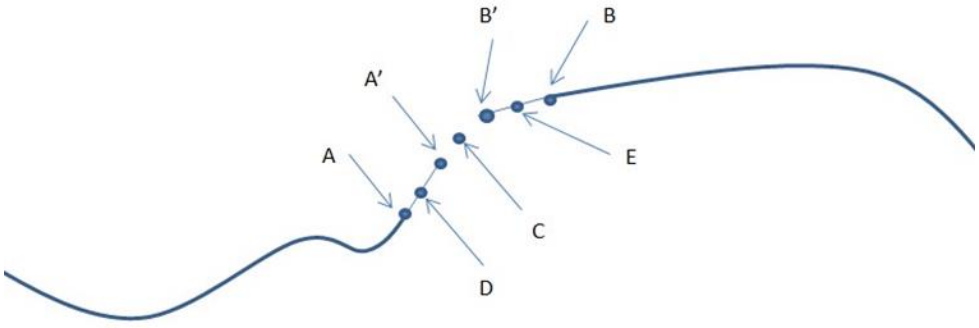


Figure 4.9: Illumination of point interpolation.

The system aimed to interpolate three points in the connection region. For the left curve C_1 with parametric equation:

$$C_1 = f(t_1) \tag{4.13}$$

The range of t_1 is from 0 to n. We defined end point $A = f(n)$ and the extended point $A' = f(n+1)$, the first interpolated point in the left side is the middle point D of A and A':

$$D = (f(n) + f(n+1))/2 \tag{4.14}$$

For the right curve C_2 with parametric equation:

$$C_2 = g(t_2) \tag{4.15}$$

The range of t_2 is from 0 to m. We defined end point $B = g(0)$ and the extended point $B' = g(-1)$, the second interpolated point is the middle point E of B and B' in this side:

$$E = (g(0) + g(-1))/2 \tag{4.16}$$

The third interpolated point C is the middle point between D and E:

$$C = (f(n) + f(n+1) + g(0) + g(-1))/4 \tag{4.17}$$

According to the control points of each hair strand, the piecewise polynomial curve parametric function for each hair strand was easily calculated by the system. By this interpolation approach, the system connected the strand segments with a smooth curve successfully.

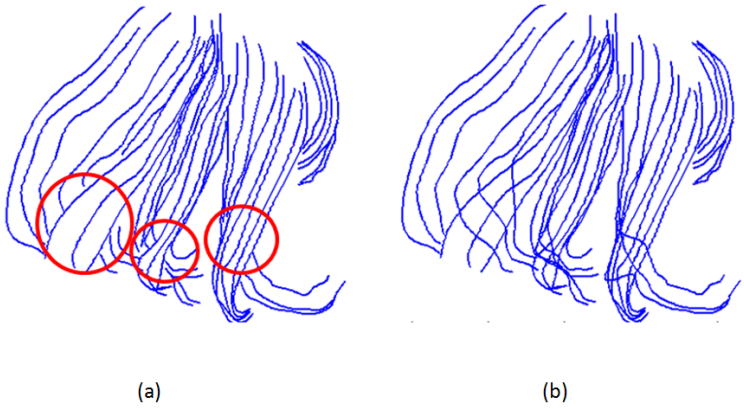


Figure 4.10: Connection analysis and establishment, a) the red cycles indicate the hair occlusion situation, b) the connection result, the disconnections were recovered in the red cycles area in (a).

4.3.3.3 Spline Curve Smoothing

We adopted the median filter and smoothing filter to process the spline curve from the previous phase. In the system, the median filter was used to reduce the noise point in the 3D hair strand curve. With curvilinear parametric function, we set 30 control points along the curve, and the median filter considered each control point and looked at eight nearby neighbour points to decide whether or not it is representative of its surroundings. It replaced the point being considered with a median value which was calculated by first sorting all the neighbour point values in numerical order and then finding the middle value. After the first time filtering, we

adopted a moving average filter to smooth the curve. The moving average filter is a kind of low pass filter, which operates by averaging a number of points from the input to produce each point in the output [83], and its equation form is written:

$$y_i = \frac{1}{M}\sum_{j=0}^{M-1} x_{i+j} \tag{4.18}$$

Where x_{i+j} is the input point array, y_i is the output point, M is the point amount, this filter replaced the value of point in consideration by the average value of following points, a moving window of M values is used to calculate the average of the data sequence. In practice, we use window size 5 to calculate the average. Figure 4.11 shows the filtering result of the hair strands.

(a) (b)

Figure 4.11: (a) Smoothing result with median filter, (b) smoothing result with smoothing filter

Chapter 5. 3D Hair Reconstruction using Particle System

This chapter introduces the particle-based 3D hair reconstruction stage. In this stage, the hair strands were represented by the curvilinear parametric function. Our system selected uniformly distributed control points on the curve for the purpose of generating hair strands using the particle system. Different from the common animated particle system, a static particle system that consists of a distinct number of curves was adopted. Furthermore, the spline curves (hair strands) could be controlled with velocity vectors and force fields, those important attributes make hair animation possible.

5.1 Hair Particle System

Similar to the animated particle system, an emitter is also necessary for a static particle system. In the previous stage, the system had calculated the radius and center position of a spherical skull. The sphere was used as the head and emitter in the hair particle system. The hair strands could grow on the sphere.

In the hair particle system, each hair strand was an independent particle. The length of the hair strand was equal to the particle path during its lifetime. In order to create the hair model, the system first selected uniformly distributed points from the curvilinear parametric functions. These points were used as control points in order to generate the guide hair. The orientation of the guide hair strands represented the hair wisp orientation. The particle system also contributed to the growth of child hair strands surrounding each guide hair. These child hairs are subparticles which are assigned to the guide hair and their properties are inherited from their parent.

In the reconstruction phase, the input information of each hair strand was the curvilinear parametric function. We selected equal amount, uniformly distributed control points, for each curve. For a curve *C* with parametric equation:

$$(x, y, z) = C(t), t = [0 \dots \dots n] \tag{5.1}$$

Parameter *t* is from 0 to *n*, if *m* is defined as the number of control points, the control points' positions are $C\left(\frac{n}{m}\right), C\left(\frac{2n}{m}\right), C\left(\frac{3n}{m}\right) \dots \dots C(n)$. In the system *m* was set to be 25, for each strand the system obtained 25 control points. Child hair strands are hair and emitter particles assigned subparticles. They make it possible to work primarily with a relatively low amount of parent particles, for which the physics are calculated. They carried the same material as the guide hair and coloured according to the exact place from where they were emitted. The amount of child hair affects the density and appearance of hair structure. The children were then aligned with their parents, without recalculating the physics the number and visualization of the children can be changed.

5.2 3D Hair Reconstruction and Rendering

The realistic rendering of the 3D hair was influenced by many factors, such as the local and global hair properties and some other environmental factors. The hair material and the light setting are important components for rendering. A rendering process can be applied to generate decent results with the proper internal and external settings.

5.2.1 3D Hair Generation

To reconstruct a particle-based 3D hair model, the first task is generating a skull object in the 3D space. The position and size of this spherical skull was calculated in the previous spherical skull extraction phase. Theoretically, a spherical skull is suitable for particle generation.

However, in practice, the surface of a sphere always affects the growth path of hair strands, because the curvature of the sphere surface is generally greater than the curvature of the hair strand curve. We deformed the sphere to make its shape more close to the human skull, and the deformed sphere was implanted into a human head model, it acted as the particle emitter in the experiments.

In the particle generation phase, "*steps*" is an important parameter of the hair strand curve. It equals the number of segments of a strand curve, which affects the smoothness of the curve. In practice, 10 segments could provide a sufficient smoothness for very long hair. B-spline was adopted by the particle system to interpolate hair between control points. In practice, the hair strands were smoother and the visualization of reconstruction result is more realistic with B-spline.

5.2.2 Lighting and Particle System Settings

With regard to the light source, two area light sources were created and positioned in the left-top and right-top locations in the front of the hair model, and the strength of two light sources was 4000 units in *Blender* light setting. For different hairstyles, and rendering effect, the system could adopt different lighting solutions.

The number of child hair strands always affects the 3D visualization and rendering effect. Besides the number of hair strands, the radius of child hair cluster also determines the hair density of the hair model. For the realistic hair rendering, we made proper particles settings in the *Blender* particle system, such as the number of the child hair strand is 350 in rendering for each guide hair strand, and the radius of the strand wisp is 0.4. And we used thick representation for hair strand in rendering. The details of the setting are shown in Figure 5.1 below.

Figure 5.1: The particle settings.

In Figure 5.1, the *Simple* mode means that the children are emitted from the parent particles, *Display* is the number of children in the 3D window, *Render* is the number of children to be rendered, *Size* indicates a multiplier for children size and *Random* is the value of random variation to the size of child particles. For the *Effects* settings, *Clump* value which is from -1.0 to 1.0, controls the location where children meet at, the children may meet at their tip when it is 1.0 or start together at their root when it is -1.0. *Shape* indicates the form of clump, either inverse parabolic (0.99) or exponentially (-0.99). The *Length* is the length of child paths, and *Threshold* controls the amount of particles left untouched by child path length. *Radius* controls the radius in which the children are distributed around their parents. *Roundness* controls the roundness of the

children around their parents, either in a sphere (1.0) or in plane (0.0). Seed indicates offset in the random number table for child particles, to get a different randomized result. For *Roughness* settings, *Uniform* controls amount of location dependent rough, it is based on children location so it varies the paths in similar way when the children are near, and *Size* is the size of location dependent rough. *Endpoint* controls amount of endpoint rough, "Rough End" randomizes path ends, *Shape* is shape value of endpoint rough. And *Random* controls amount of randomness rough. It is based on a random vector so it's not the same for nearby children, *Size* and *Threshold* control the size and amount of particles left touched random rough [85].

5.2.3 Rendering and Simulation

To reconstruct the captured hair, the colour and material were also important characteristics of the hair strands, and the diffuse and specular were the principal parameters of the hair material. In the rendering stage, we adopted *Blender Cycles* Rendering Engine for realistic hair rendering. The rendering result was directly affected by the hair strand's material. A shader system (Figure 5.2) was built following the Trammell's *Blender* and *Cycles* tutorial [86] to simulate the real hair material under the natural light. In order to reproduce the real hair colour, we calculated an average RGB value *avgRGB* based on the colour data in the hair image, the *avgRGB* was the basic colour inputs of two *MixRGB* shaders in the shader system. For the '*Add*' type *MixRGB* shader, the value of '*Color2*' equals *avgRGB/255*, and for another *MixRGB* shader, the inputs '*Color1*' equals *avgRGB/10000*, and '*Color2*' equals *avgRGB/1000*. A colour ramp shader was used to simulate the gradual change of the colour from the hair root to tip, and we used Voronoi texture on the scalp to change the greyscale in different areas. Some other shaders were used to simulate the reflection effect, such as diffuse and translucent shader. This shader system was applied to the rendering results, which are shown in Chapter 6.

Our particle-based hair model is available for dynamic hair simulation, with the physical structure which consists of the strands and skull, the simulation such as head motion and hair collision can be produced in *Blender*. For simulation, a series of field weights attributes of hair strands should be set, such as gravity, force and magnetic. And pre-defined head and object movements are scripted to drive the hair motion. The guide hair strand is driven by the head movement and the child hair is driven by the guide hair. The simulation processing and computation are all achieved by *Blender*. Some frames of simulation result are shown in Chapter 6, and more results can be found in the accompanying video of this thesis.

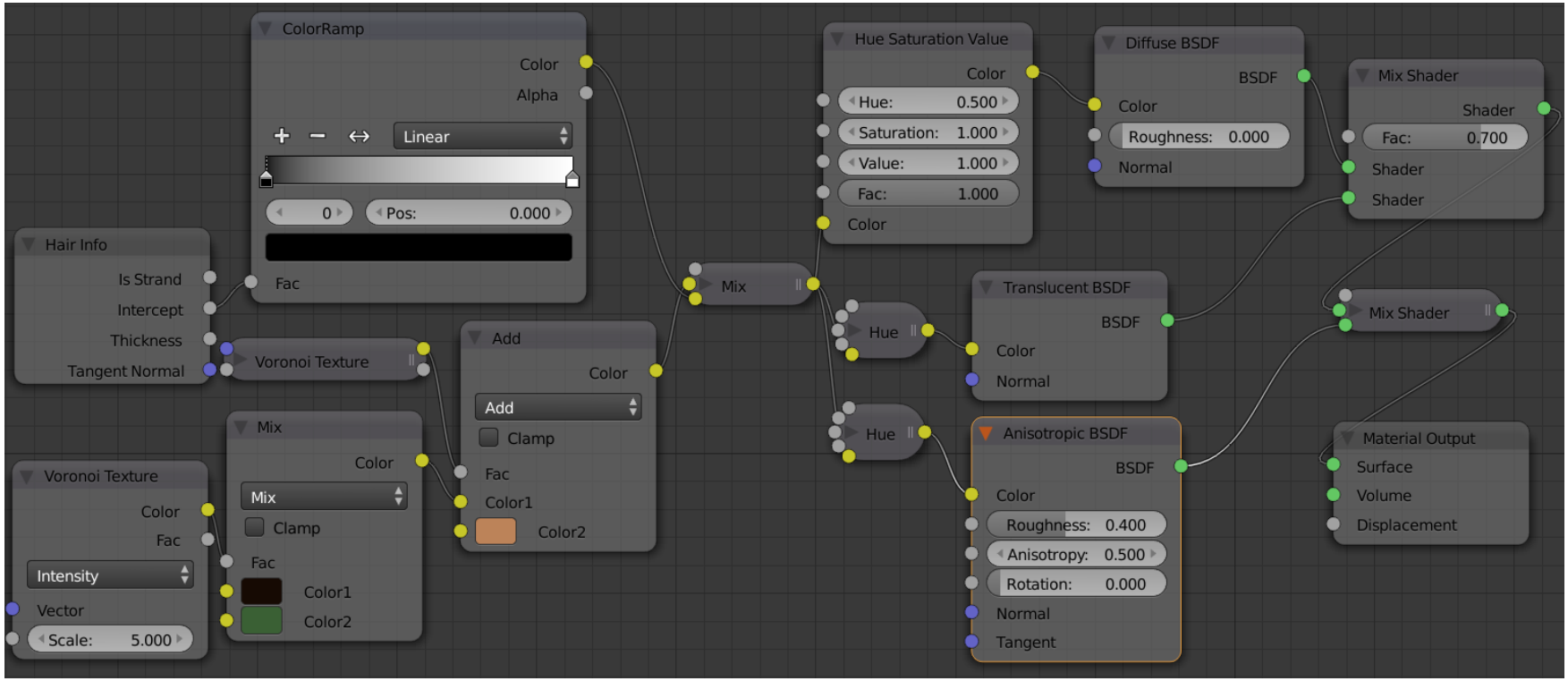


Figure 5.2: Shader system for hair modeling using *Blender* and *Cycles* [86].

Chapter 6. Result and Validation

In this chapter, we demonstrate the experimental results, the workflow of the system, and we also represent the validation of the proposed 3D hair reconstruction system to verify the feasibility, stability and efficiency of the system implementation. The evaluation and discussion are also included in this chapter to discuss the features, advantages, general issues and limitations of the proposed system.

6.1 Experiment Result

Through the experiment, our system shows the ability of handling variety of relative simple hair styles, we ran our system with female short straight hair (wig), long straight hair and long curly hair (wig). In addition, our system can handle different hair colour, like blond, brown and light black. In this section, due to the space, we only present the results of five samples.

Subsection 6.1.1 demonstrates the results of each process during the reconstruction of a curly hair wig. For each viewpoint, the results of image registration, 2D image and 3D point cloud hair area segmentation and the hair strands extraction are demonstrated. In addition, the results of point clouds alignment, reconstruction and rendering of particle-based 3D hair model are also shown below. The 2D hair strands extraction was done manually with a user interactive tool.

Subsection 6.1.2 demonstrates the results of each stage during the reconstruction of single back view of a real female's hair. In this experiment, we adopted the data from automatic 2D hair strands extraction method which is proposed by Fatemeh [87] and Sun [88]. The research work of this approach is in processing.

Subsection 6.1.3 demonstrates some frames from the hair simulation videos which were produced using *Blender*.

The input high resolution images were captured by a Canon 500D DSLR camera with a Sigma 10-20mm/ 3.5F lens. The regular images and point clouds were captured by Kinect sensors.

6.1.1 Reconstruction of 360 Degree View Long Curly Wig

The figures of the following sections illustrate the sequential process stage of the proposed system as follows:

Figure 6.1, Figure 6.4, Figure 6.7 and Figure 6.10 demonstrate the image registration stage of the back, right, left and front respectively. The major processes in this stage include:

- Registering Kinect colour image and high resolution image.
- Projecting the registered image on the point cloud and generating an inverse-mapped image which only contains the corresponding pixels of the collected 3D points.

Figure 6.2, Figure 6.5, Figure 6.8 and Figure 6.11 demonstrate the 2D and 3D hair area segmentation stage of each viewpoint.

And Figure 6.3, Figure 6.6, Figure 6.9 and Figure 6.12 demonstrate hair strand extraction stage of each viewpoint. The processes in this stage include:

- Extracting 2D hair strands on image using user interactive tool.
- Extracting corresponding 3D hair strands points in the point cloud.
- Analyzing the occlusion and building the connection between hair strand segments.

Figure 6.13 demonstrates the point cloud alignment stage, Figure 6.14 demonstrates particle-based 3D hair model and the rendering results of four viewpoints.

6.1.1.1 Back View

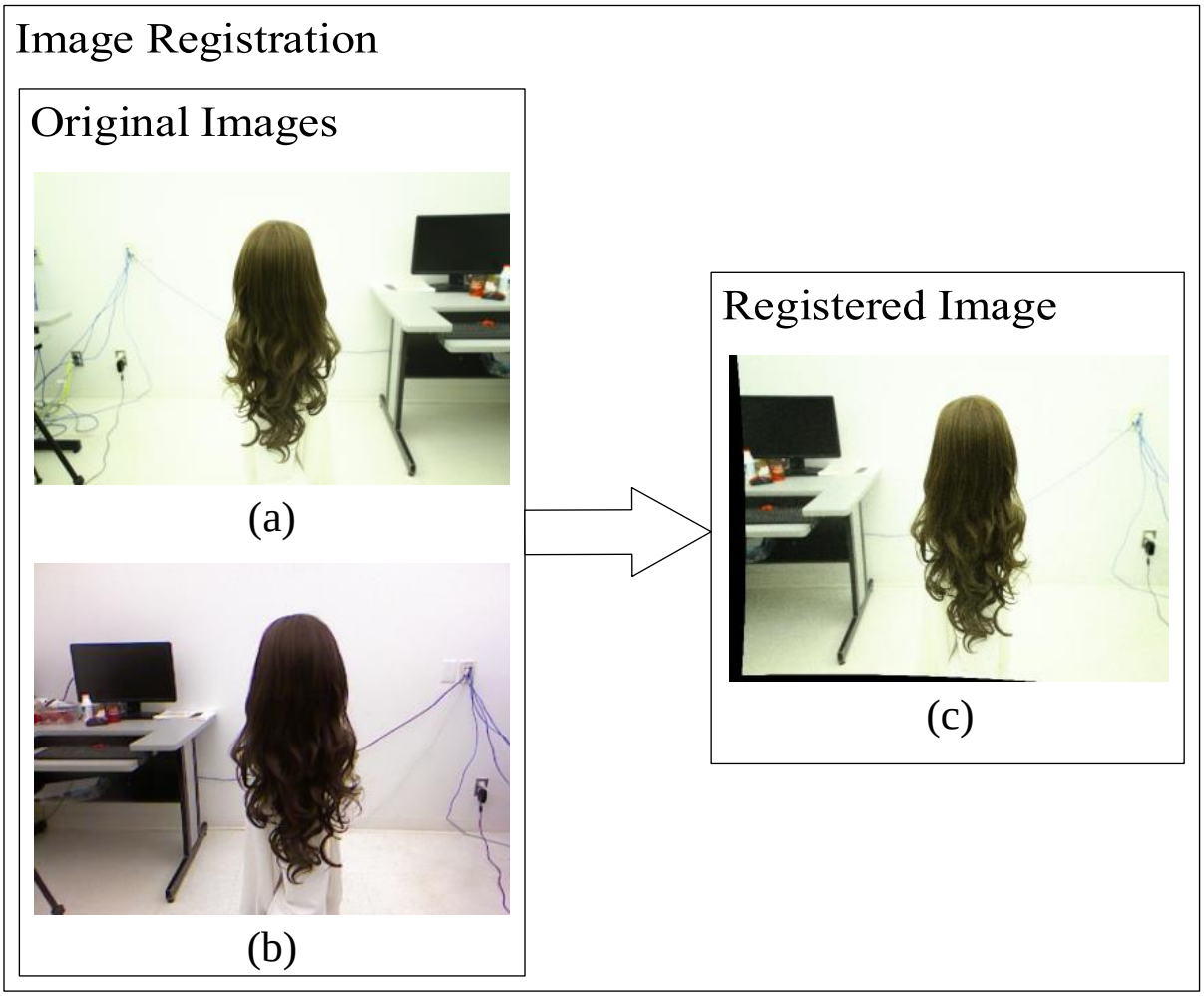


Figure 6.1: Back view image registration phase, a) high resolution image from DSLR camera, b) Kinect RGB colour image, c) registered high resolution image.

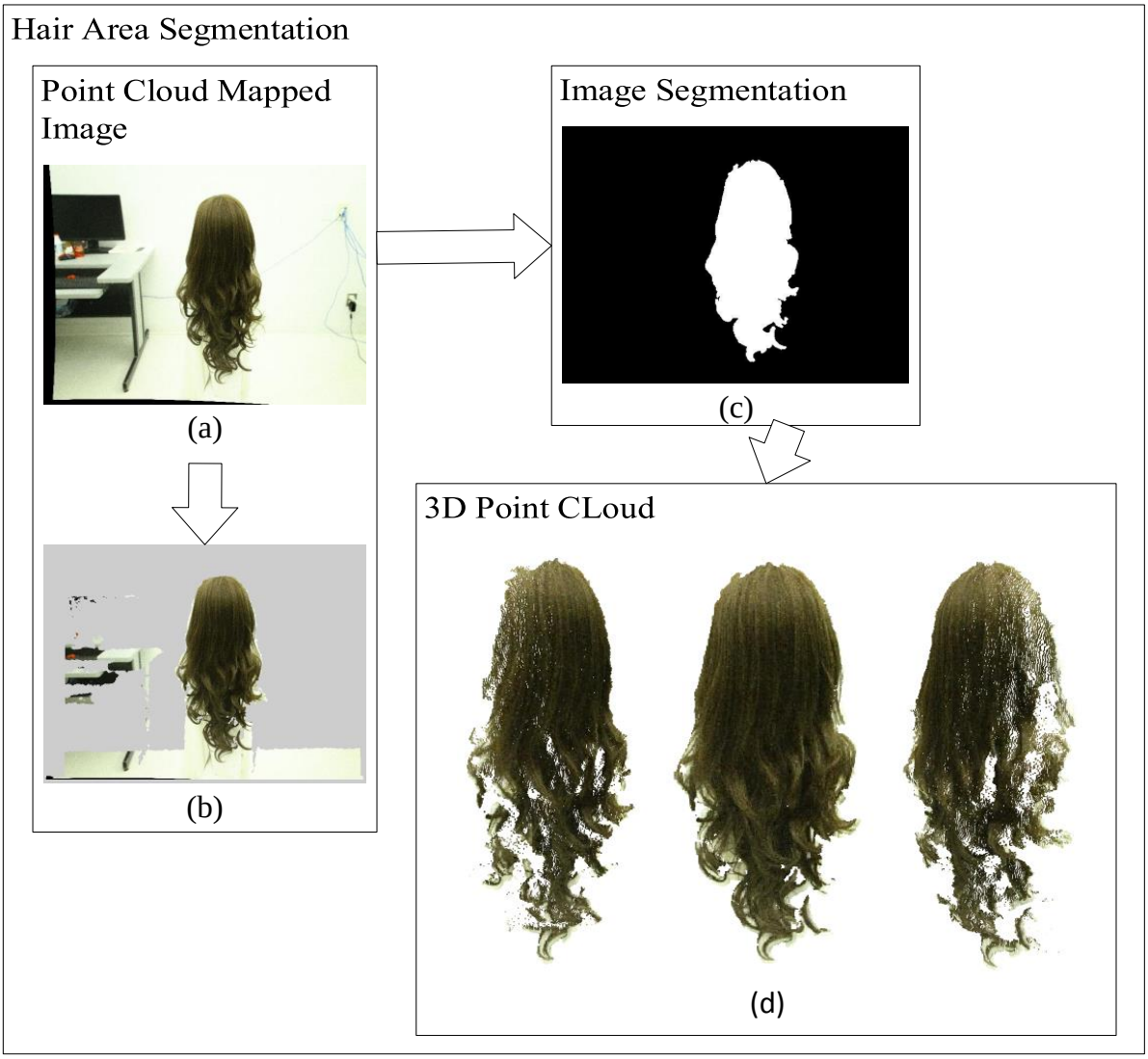


Figure 6.2: Back view hair area segmentation, a) registered high resolution camera image, b) point cloud inverse mapped image, c) 2D image hair area segmentation based on (b), d) 3D point cloud hair area segmentation.

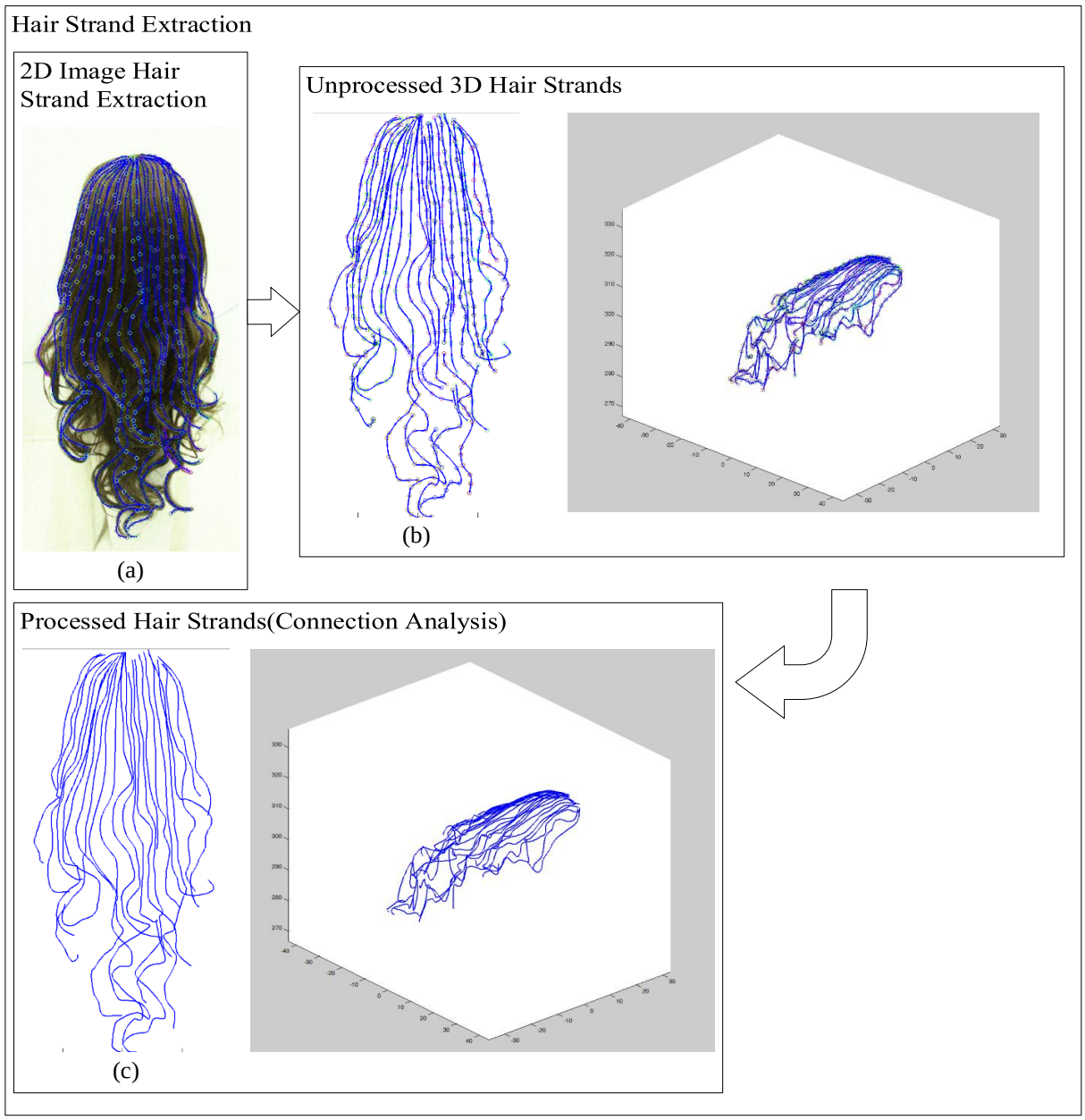


Figure 6.3: Back view hair strands extraction phase, a) extract hair strands from 2D image(from high resolution camera), b) extracting the corresponding 3D hair strands in point cloud, c) processed hair strands after connection analysis.

6.1.1.2 Right View

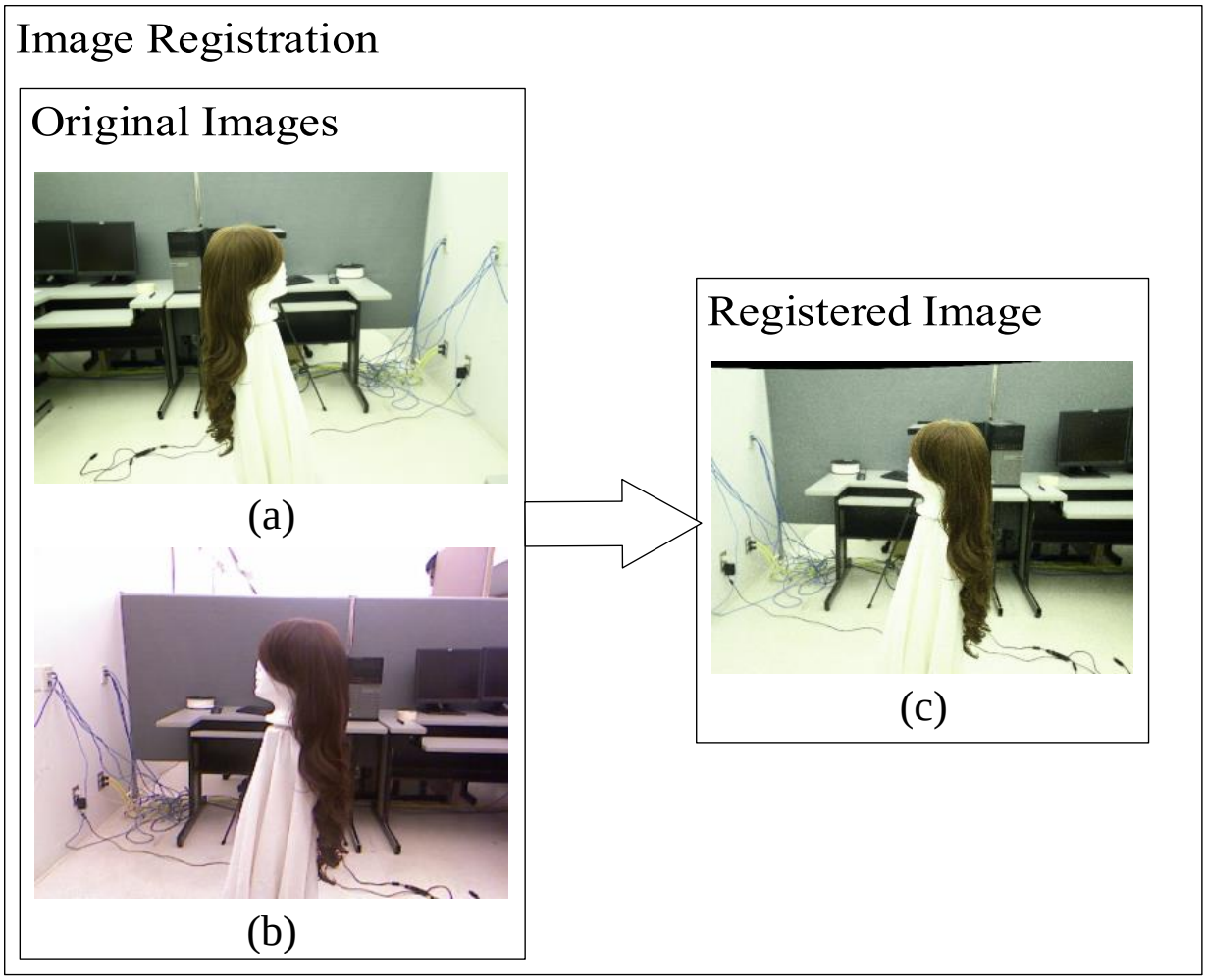


Figure 6.4: Right view image registration phase, a) high resolution image from DSLR camera, b) Kinect RGB colour image, c) registered high resolution image.

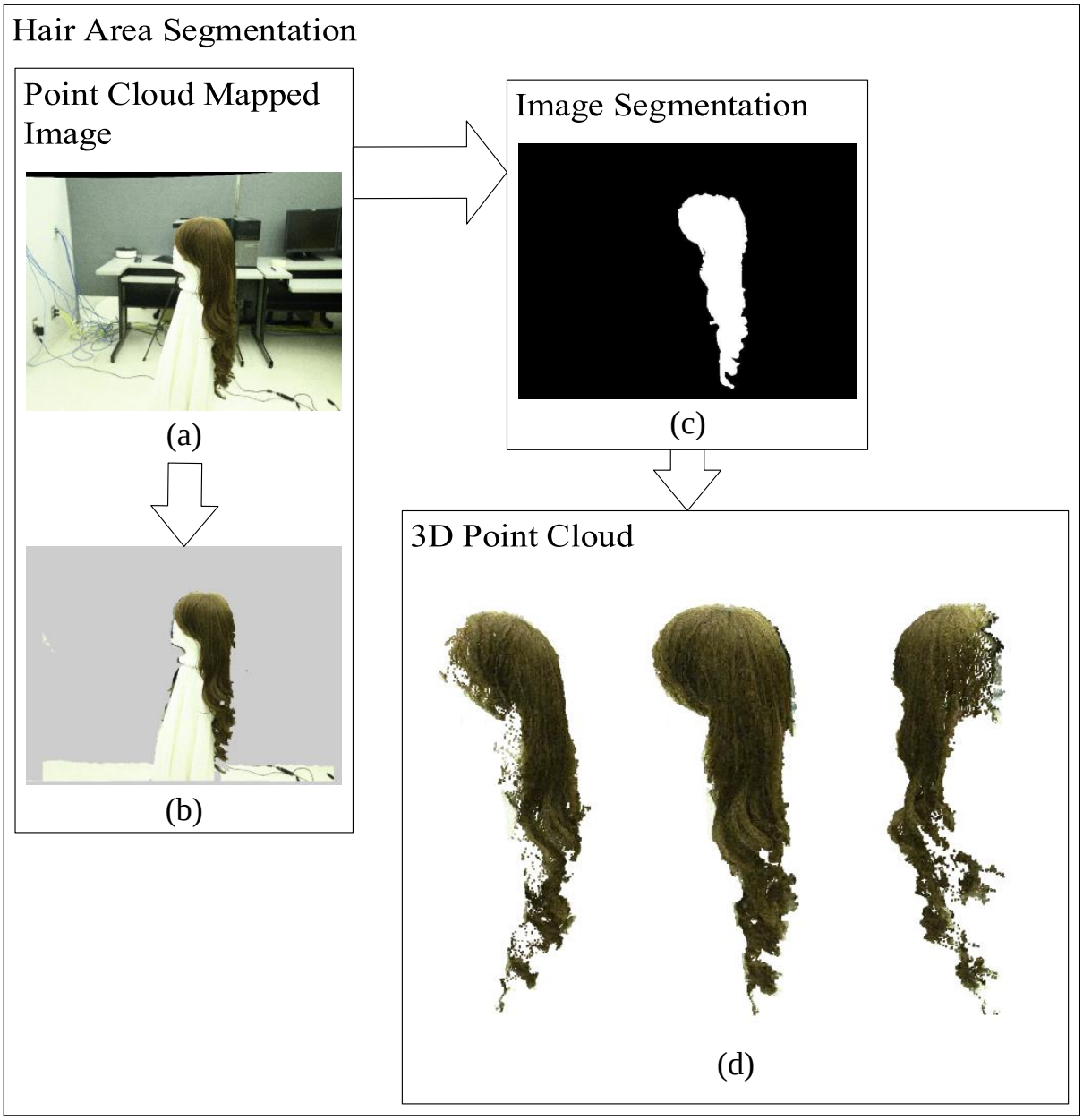


Figure 6.5: Right view hair area segmentation, a) registered high resolution camera image, b) point cloud inverse mapped image, c) 2D image hair area segmentation based on (b), d) 3D point cloud hair area segmentation.

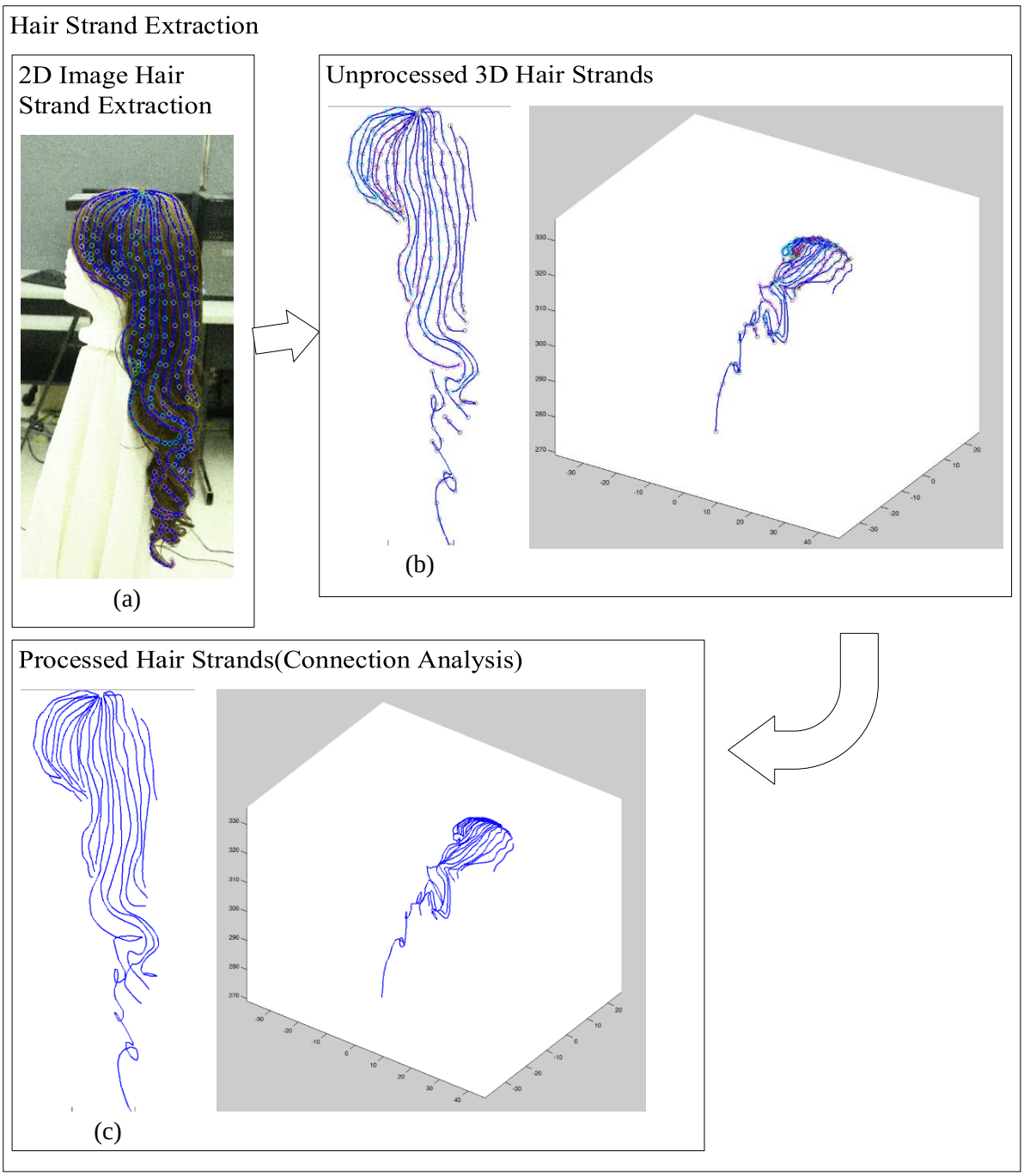


Figure 6.6: Right view hair strands extraction phase, a) extract hair strands from 2D image(from high resolution camera), b) extracting the corresponding 3D hair strands in point cloud, c) processed hair strands after connection analysis.

6.1.1.3 Left View

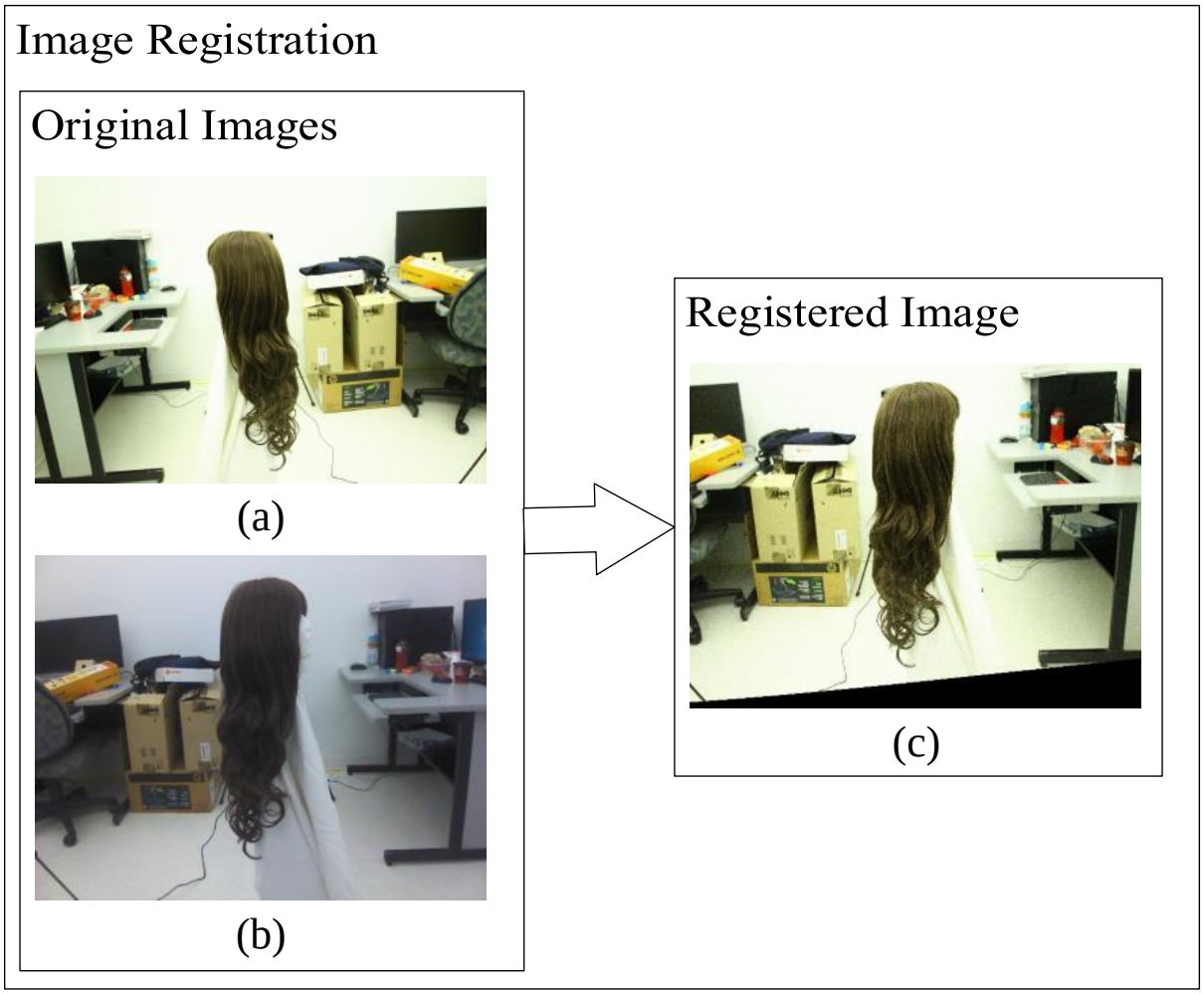


Figure 6.7: Left view image registration phase, a) high resolution image from DSLR camera, b) Kinect RGB colour image, c) registered high resolution image.

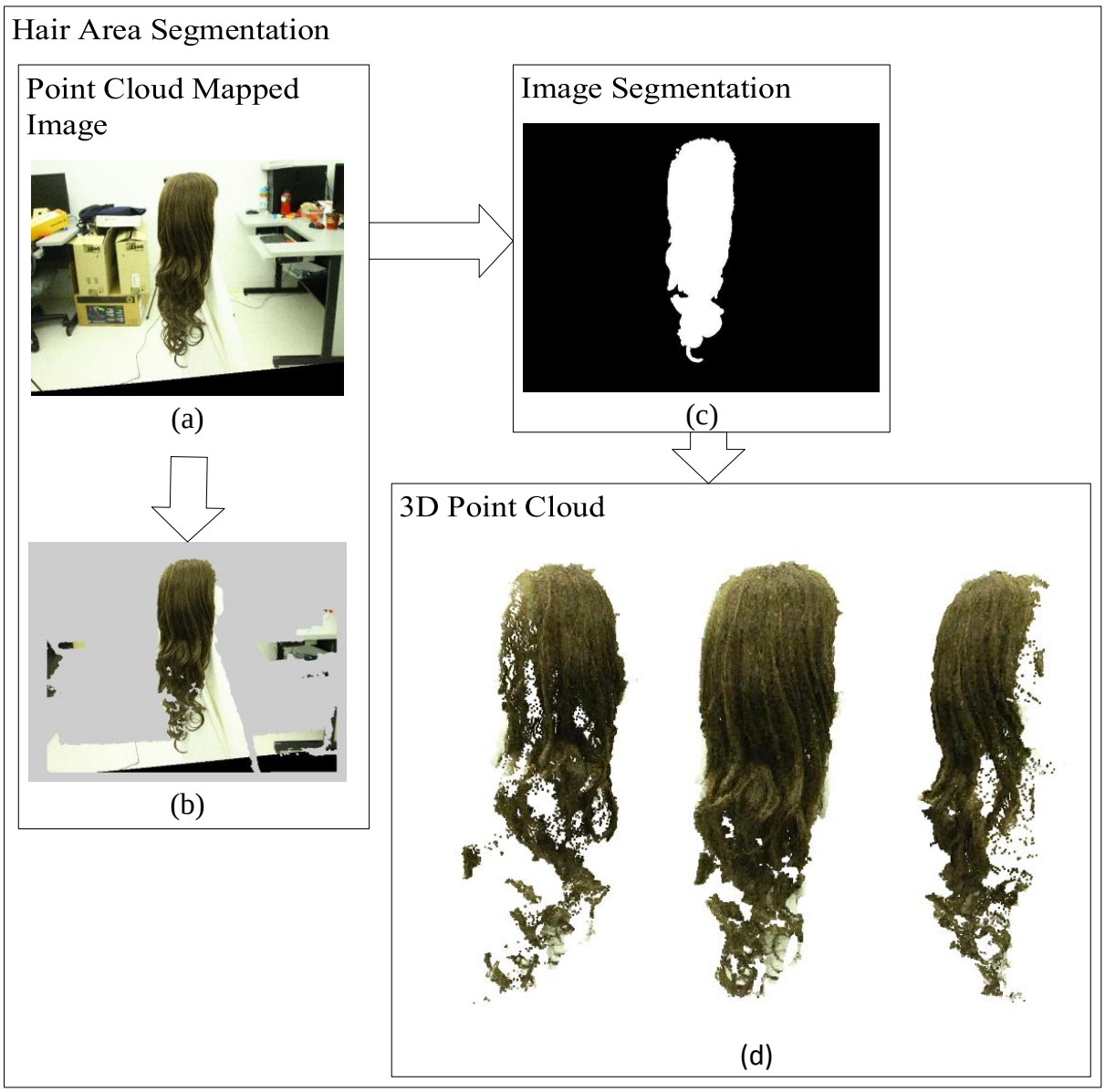


Figure 6.8: Left view hair area segmentation, a) registered high resolution camera image, b) point cloud inverse mapped image, c) 2D image hair area segmentation based on (b), d) 3D point cloud hair area segmentation.

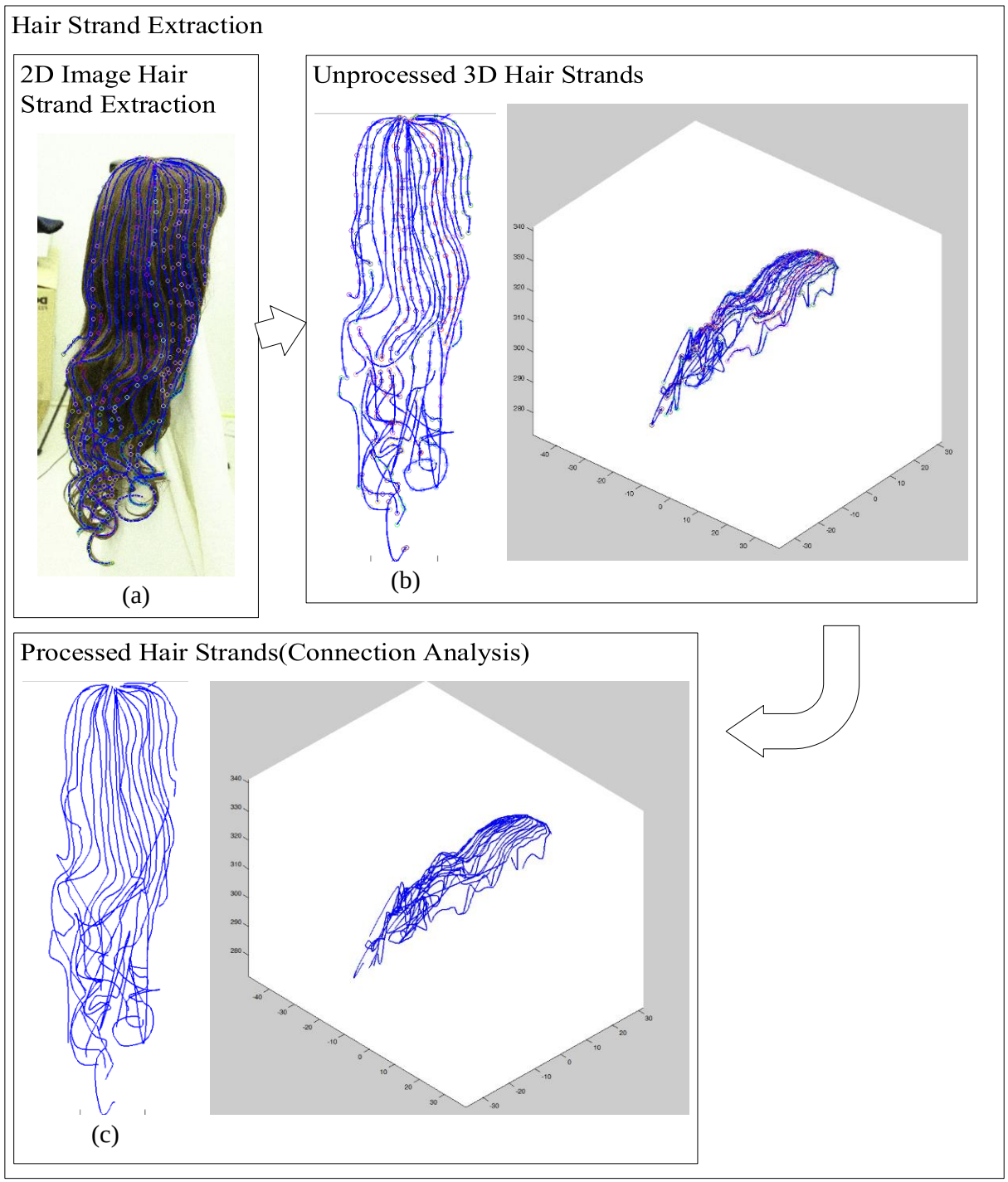


Figure 6.9: Left view hair strands extraction phase, a) extract hair strands from 2D image(from high resolution camera), b) extracting the corresponding 3D hair strands in point cloud, c) processed hair strands after connection analysis.

6.1.1.4 Front View

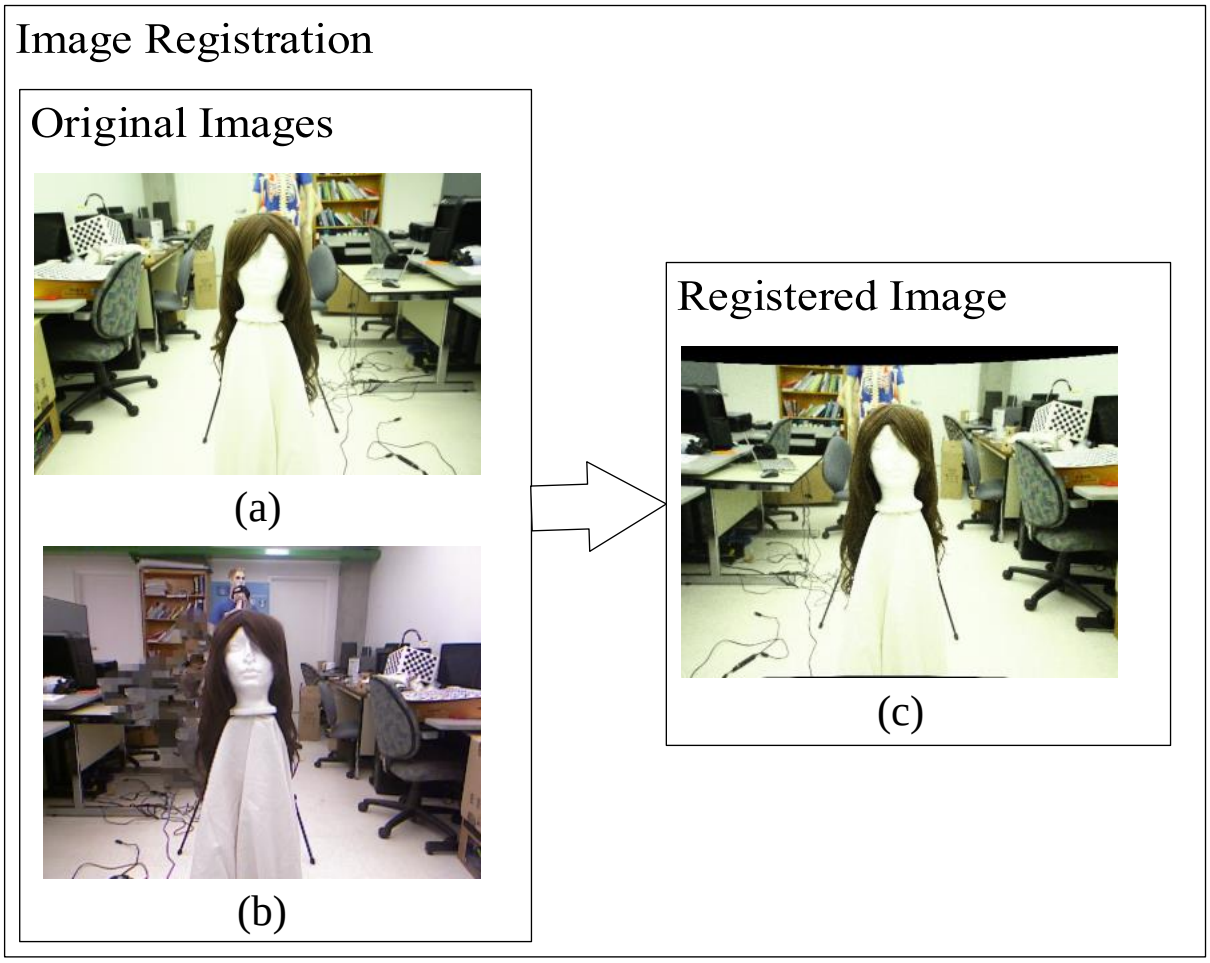


Figure 6.10: Front view image registration phase, a) high resolution image from DSLR camera, b) Kinect RGB colour image, c) registered high resolution image.

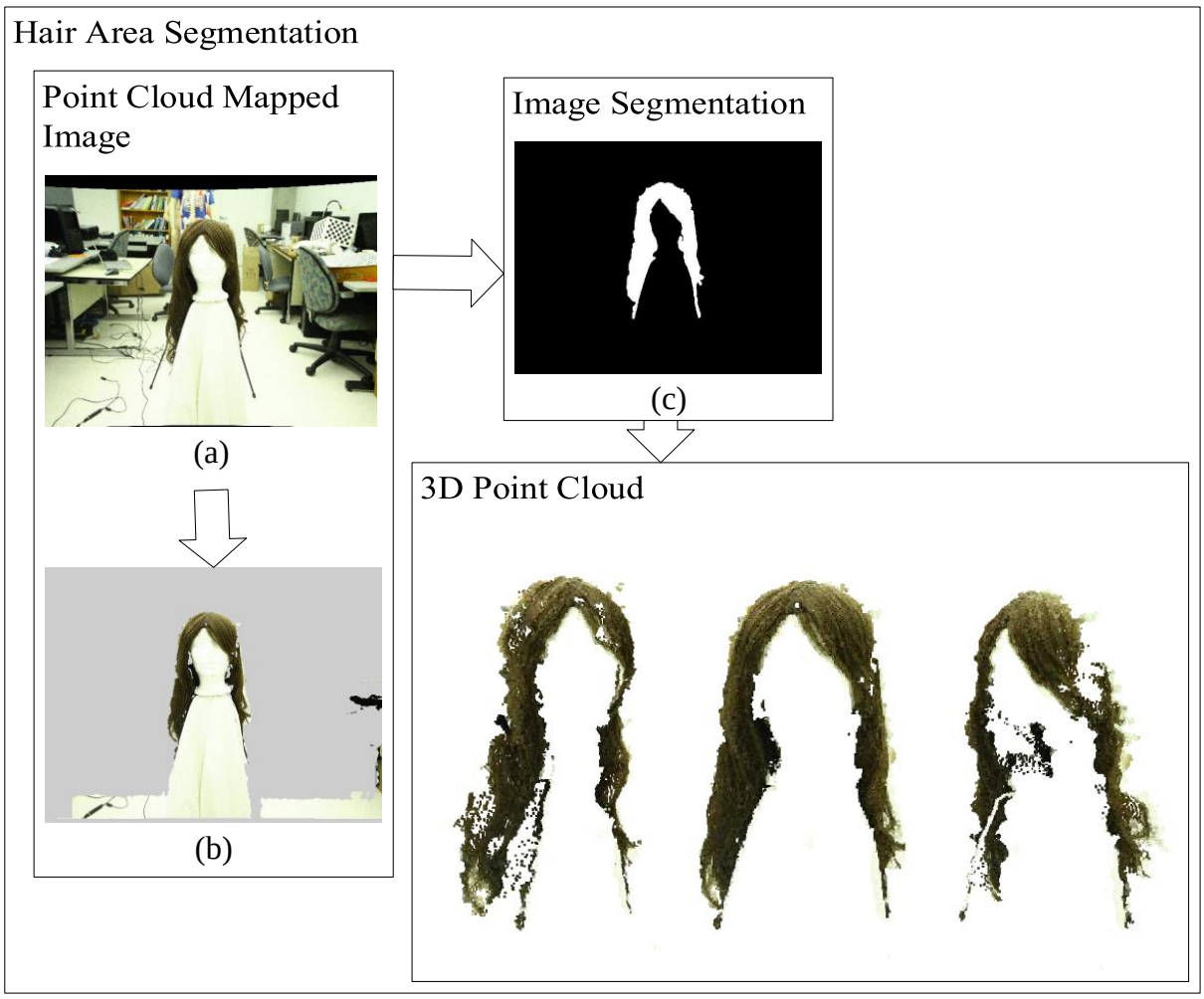


Figure 6.11: Front view hair area segmentation, a) registered high resolution camera image, b) point cloud inverse mapped image, c) 2D image hair area segmentation based on (b), d) 3D point cloud hair area segmentation.

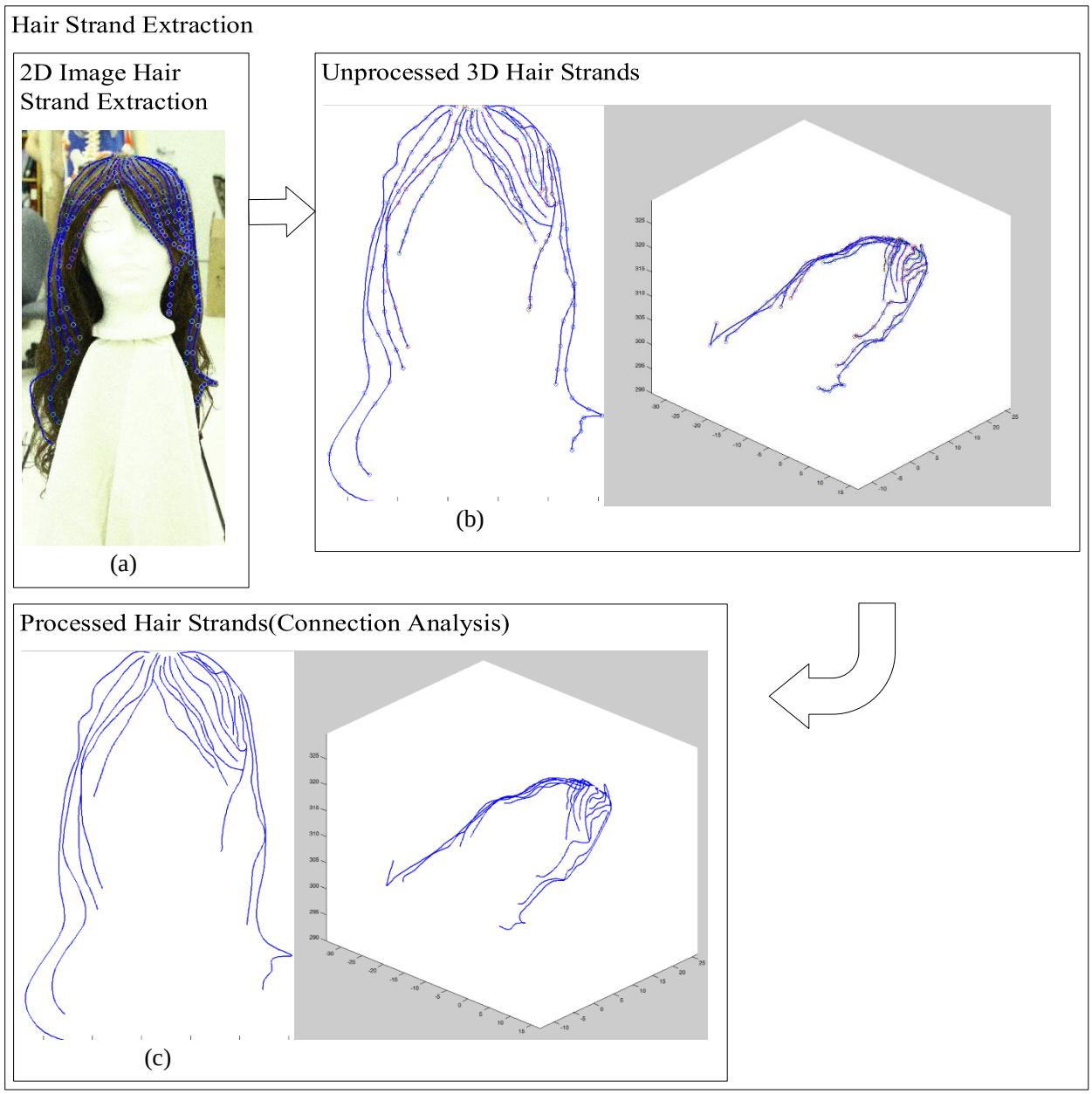


Figure 6.12: Front view hair strands extraction phase, a) extract hair strands from 2D image(from high resolution camera), b) extracting the corresponding 3D hair strands in point cloud, c) processed hair strands after connection analysis.

6.1.1.5 Point Cloud Alignment

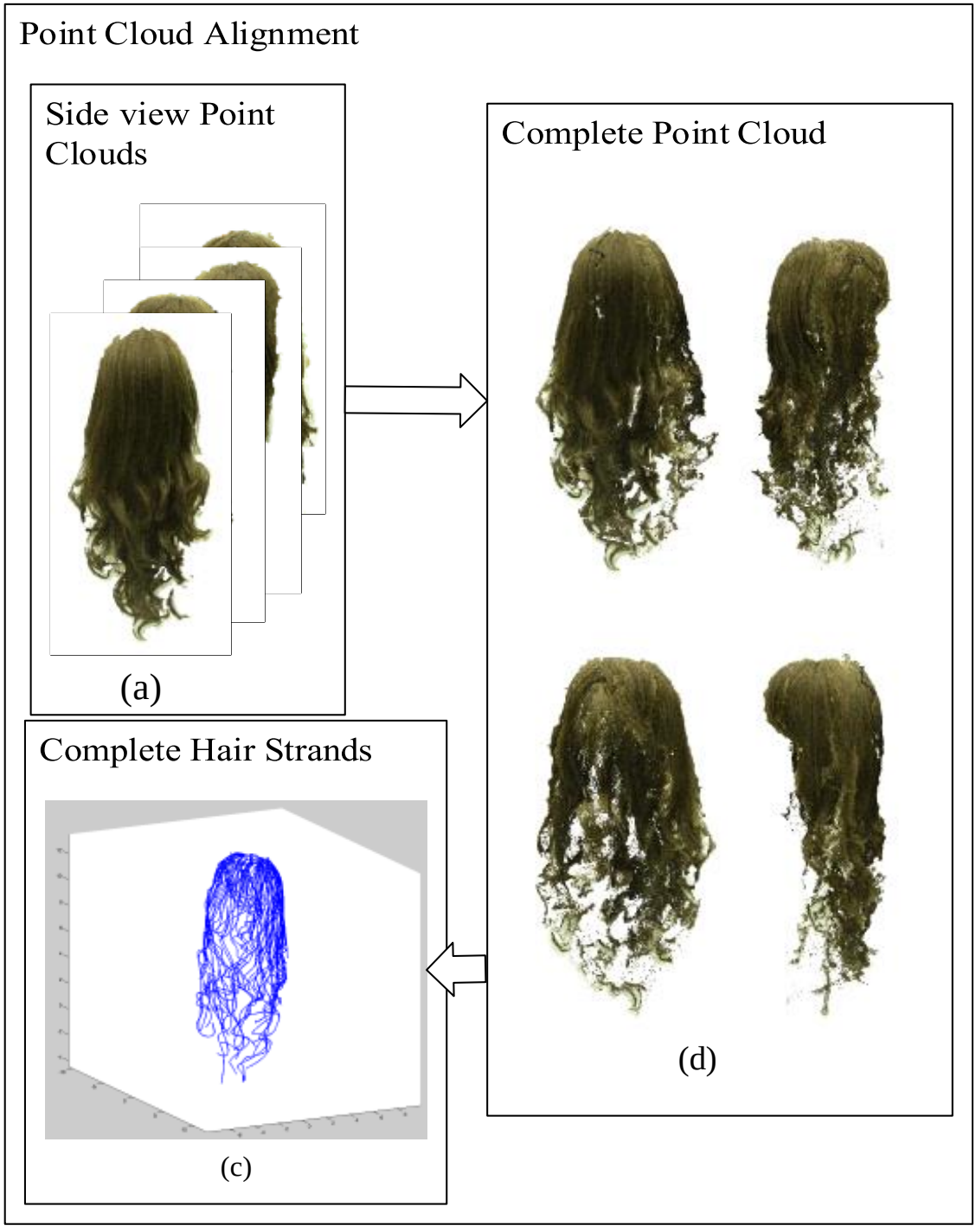


Figure 6.13: Point cloud alignment phase, a) four independent point clouds of four viewpoints, b) align the point clouds to get a complete hair point cloud, c) apply transformation to each view hair strand from independent point cloud to merge the hair strands into one coordinate.

6.1.1.6 Particle-based Hair Model and Rendering

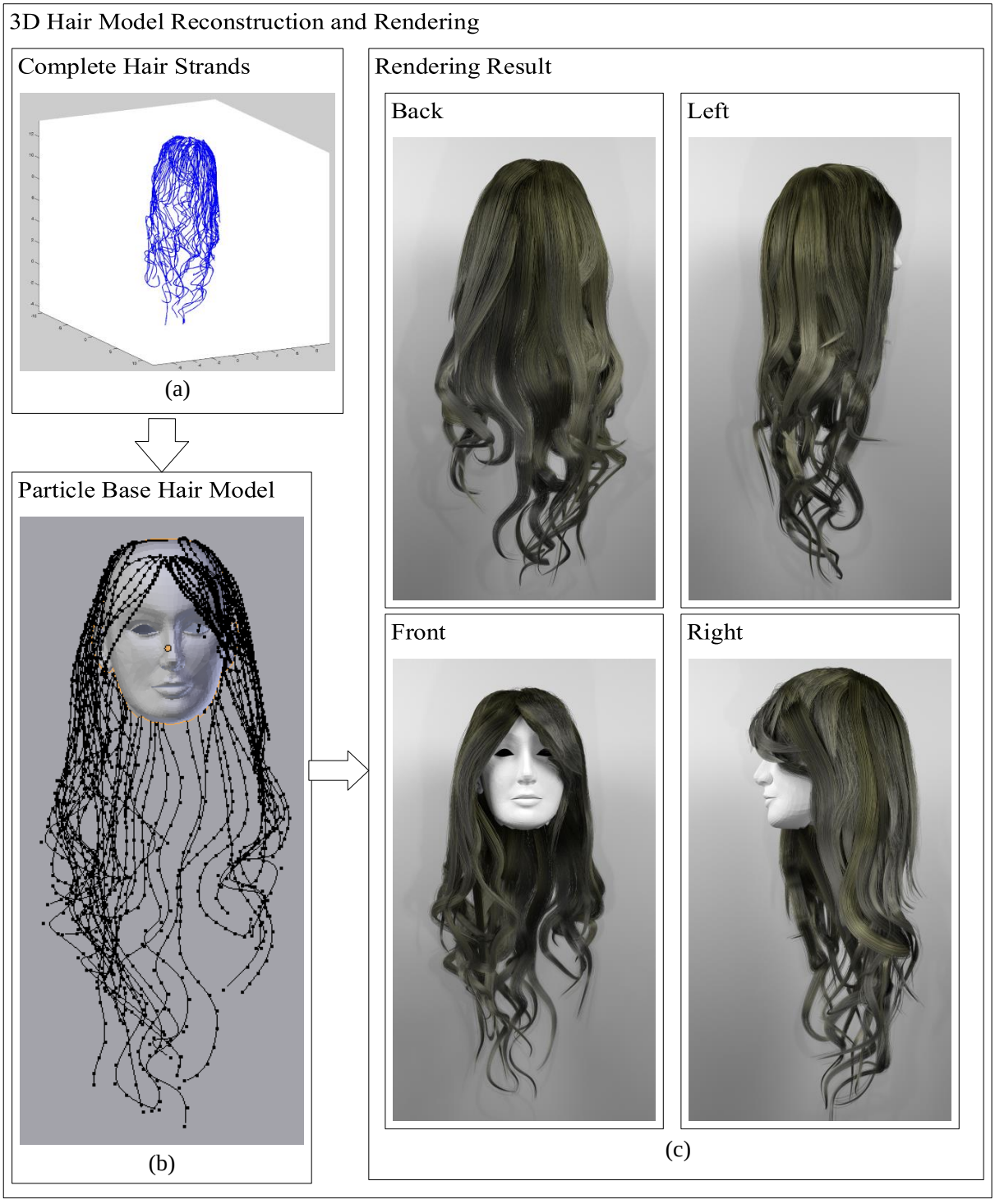


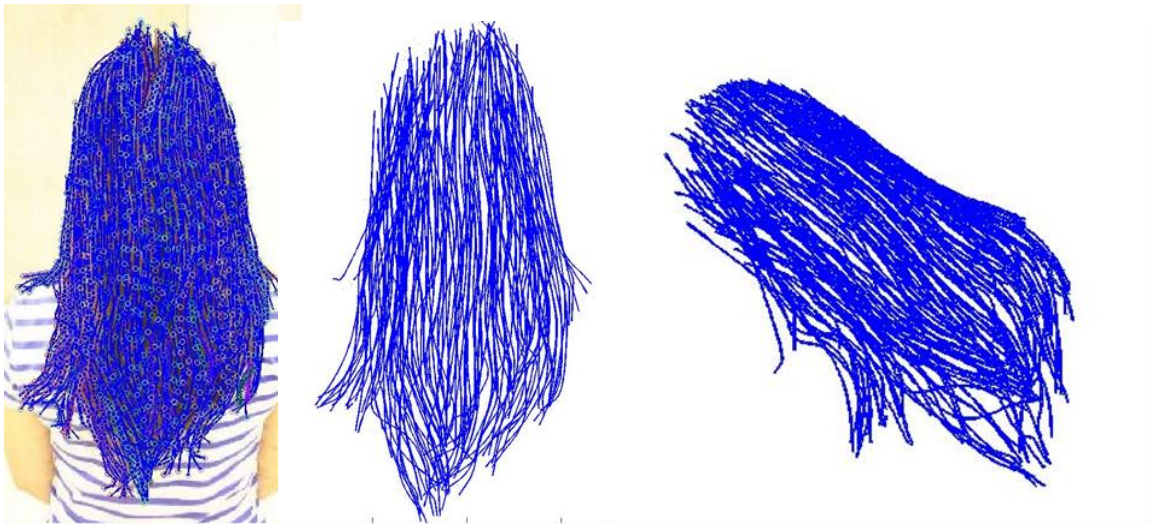
Figure 6.14: Particle-based 3D hair model reconstruction phase, a) complete hair strands after alignment, b) reconstruct a particle base hair model in *Blender*, c) rendering results of the reconstructed hair model.

6.1.2 Back View Reconstruction of Real Female's Hair

We also applied our pipeline with a real female's hair. We only captured and tested the back view of the hair. And the 2D hair strand extraction was performed automatically with Sun's automatic hair strands extraction approach [88]. The results of each stage are revealed in the following figures. Figure 6.15 shows the captured images and the point cloud, Figure 6.16 demonstrates the automatic hair strand extraction result, and Figure 6.17 demonstrates the 3D hair reconstruction and rendering result.

(a) (b) (c)

Figure 6.15: Data from data capture stage: a) Kinect colour image, b) high resolution image c) point cloud from Kinect.



(a) (b)

Figure 6.16: Hair strand extraction stage: a) automatic 2D hair strand extraction, b) corresponding 3D hair strand after connection analysis.

(a) (b) (c)

Figure 6.17: 3D hair reconstruction and rendering result, a) particle-based hair reconstruction using *Blender*, b) reconstruction with child particles, c) rendering result using *Blender Cycles*.

6.1.3 Simulation

Figure 6.18 shows some results of hair physical simulation, for this hair model, we have around 300 guide hair strands, and for each guide hair, there are 300 child hair stands during rendering. More results can be found in the accompanying video of this thesis.

Figure 6.18: Six frames from two physical simulation videos, up: head motion, down: hair collision.

6.2 System Validation

Our capture system consists of four *Microsoft* Kinects for *Windows* and four digital SLR cameras, the price of Kinect is $200/each, the price of the adopted digital SLR camera is $600/each.

The proposed system and the demonstrated experiments were all performed using a current configuration desktop computer. The configuration of the computer is shown below in Table 2.

Table 2: The computer configuration table

Computer Configuration	
Processer	Intel(R) Core(TM) i7-3370 CPU@3.4GHz
Memory	16.0GB
Graphics	AMD Radeon HD 7500 Series
Operating System	Windows & Enterprise 64bit

The capture system is developed and implemented with Kinect for Windows SDK 1.7, Kinect for Windows Runtime 1.7, Kinect for Windows Developer Toolkit 1.7 and OpenGL 4.4 (Open Graphics Library) and OpenCV 2.46 (Open Source Computer Vision), using C++ in Microsoft Visual Studio 2012. The data analysis subsystem is implemented in Matlab 2013b, and the hair reconstruction and rendering subsystem are developed and implemented using Python with *Blender* Python API in *Blender* 2.67.

For the realistic hair rendering in *Blender*, the render time of each frame strongly depends on the computer performance and the rendering setting, such as the child hair amount, sampling and picture resolution. Using the PC with the regular configuration shown in Table 2, and the light and particle settings which are described in section 5.2.2, the render time of each frame is around one minute. For the simulation video which is about 30 seconds and 30 frames per second, the total rendering time is around 15 hours.

In order to evaluate the system performance, we examined the experiment results from four quantifiable phases, and we evaluated the usage of the major data units in the experiments, such

as 3D point, 2D point (pixel), 2D and 3D strand, which were the important data flows in analysis phases. We also provide the comparisons between our rendering results and the original hair image as an open evaluation for reviewers.

In order to validate the system robustness, we adopted the robustness evaluation method using the varying qualities inputs, which is used in the hair capture method presented by Liwen Hu et. al [84], and structure-aware hair capture method presented by Linjie Luo et. al [20].

6.2.1 System Performance

During the hair area segmentation phase, an initial segmentation was applied to the 2D image firstly, thus the system could determine the hair area pixels in the 2D image. Then the system collected the corresponding 3D hair points from the point cloud. In theory, each pixel of the 2D image has a corresponding 3D point. However, the point missing exists and results in holes inside the 3D hair area. Thus, the amount of detected 2D hair pixels is always greater than the amount of collected 3D hair points. After segmentation, the system kept the hair area points and removed the useless points. This procedure decreased the size of the point cloud.

Table 3 shows the numbers of 2D hair pixels and the 3D hair point, the original point cloud size of four viewpoints, the point cloud size is obviously decreased after hair segmentation.

Table 3: Comparison of the point number of original point cloud and the segmented hair area point cloud.

Comparison between Hair Area Point and Original Point Cloud		
Side	Hair Area(Points/Pixels)	Point Amount of the Original Point Cloud
Front	15405/16020	109185
Back	41354/42774	80953
Right	21759/25501	72888
Left	34440/36042	107238

The hair strands were manually outlined in the high-resolution image with the user interactive tool. According to these 2D strands, the corresponding 3D hair strands were obtained from the related point cloud. For the reason described previously, the amount of 3D points might be less than the amount of 2D points, fortunately, it doesn't affect the amount of the strands, and this data loss could be ignored when data set is large. Table 4 shows the amount of extracted strand and control point in both 2D and 3D of each viewpoint.

Table 4: Comparison of 2D and 3D Hair Strand Extraction.

Comparison of 2D and 3D Hair Strand Extraction		
Side	2D Hair Strand Extraction Strands/Points	3D Hair Strand Extraction Strands/Points
Front	17/226	17/204
Back	36/554	36/500
Right	22/353	22/319
Left	39/527	39/472

In the data analysis stage, a connection analysis algorithm was introduced to solve the hair strand occlusion problem. The algorithm analyzed the curve's properties to identify the connection. It merged some strand segments into one strand, thus the amount of strands is decreased after processing. The comparison is shown in Table 5.

Table 5: Strand amount of before-and-after connection analysis phase.

Comparison between before and after Connection Analysis		
Side	Before Connection Analysis	After Connection Analysis
Front	17	17
Back	36	24
Right	22	19
Left	39	21

6.2.2 System Evaluation

For system evaluation, we provide an open evaluation for reviewers with six sets of comparisons between the original hair images and the rendering results, we also provide rendering images of colour-coded hair strand to help participants recognize the position and orientation of hair strand easily. The six sets of comparisons are four viewpoints of the long curly wig (Figure 6.19 and Figure 6.20), back view of a real female's hair (Figure 6.21) and a set of close-up hair detail comparisons (Figure 6.22).

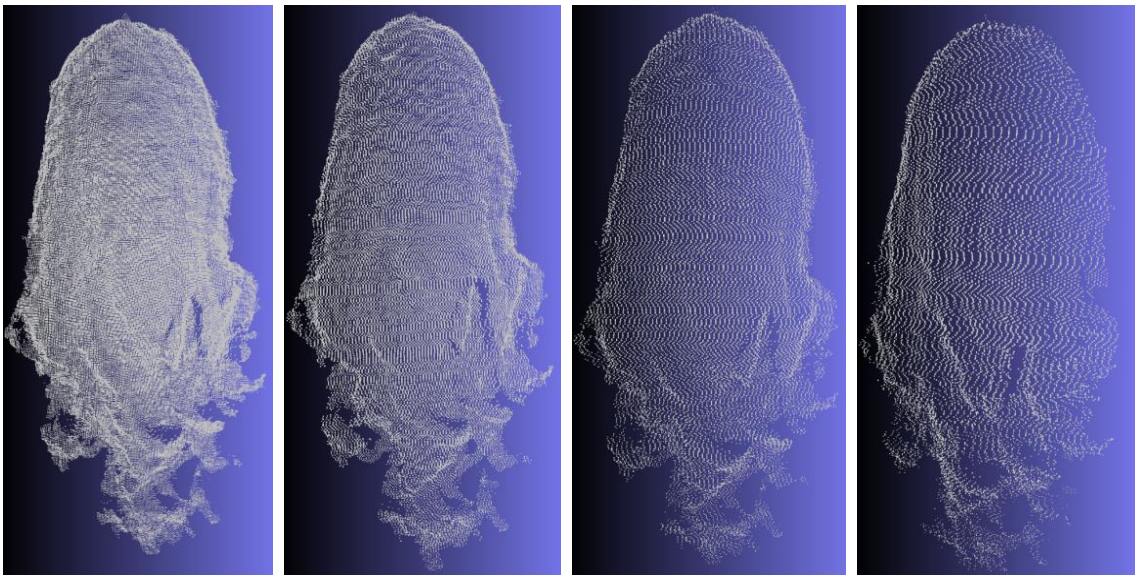
Figure 6.19: The comparisons of the front and left view data set, the left row images are the input high resolution images, the middle row images are the regular rendering results, and the right row images are rendering results with colour-coded hair strand.

Figure 6.20: The comparisons of the right and back view data set, the left row images are the input high resolution images, the middle row images are the regular rendering results, and the right row images are rendering results with colour-coded hair strand.

Figure 6.21: The comparisons of the back view of real female's hair data set, the left image is the input high resolution images, the middle image is the regular rendering results, and the right image is rendering results with colour-coded hair strand.

Figure 6.22: Close-up comparison of the hair details between the input reference high resolution image and our reconstruction rendering result.

The robustness of the proposed reconstruction system is evaluated using varying input. We provided four point cloud datasets with different size as the system inputs (Figure 6.23); these inputs are all hair area point clouds of long curl wig back view. Sample A is a full resolution point cloud, sample B is half size of sample A, sample C is 1/3 size of sample A, and sample D is ¼ size of sample A. The quantities of each sample are listed in Table 6.



(a) (b) (c) (d)

Figure 6.23: The input point clouds with different size, a) sample A, the original resolution point cloud with 49460 points, b) sample B is half size, c) sample C is 1/3 size, d) sample D is ¼ size.

A comparison work was done with four different point cloud sizes, as the 2D hair strands were mapped on the four samples and the system collected the corresponding 3D hair strands. The 3D hair strand extraction results with 4 different size point cloud are shown in Table 6 below.

Table 6: 3D hair strand extraction results with 4 different size point clouds.

Input Point Cloud Samples		
Index	Amount of Point	3D Hair Strand Extraction (Strands/Points)
Sample A	41354	35/500
Sample B	20677	35/491
Sample C	13784	35/489
Sample D	10338	35/481

This set of data indicates that the decreasing of the point cloud size doesn't affect the 3D hair strand extraction phase too much. The system adopted a nearby-search strategy for mapping failure, if the algorithm cannot find the corresponding 3D point for a 2D point, it will select a proper nearby 3D point as the corresponding 3D point for this 2D point. Figure 6.24 below shows the 3D hair strand extraction results with four different inputs. The results are almost the same, with only few differences.

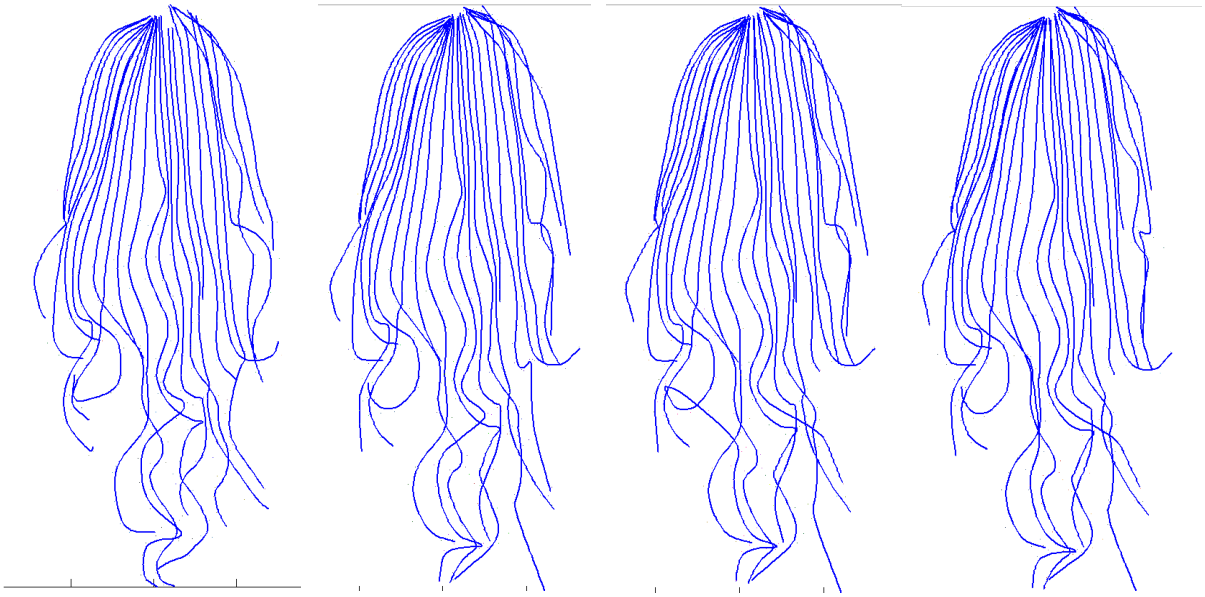


Figure 6.24: The 3D hair extraction results after connection analysis, a) result of sample A, b) result of sample B, c) result of sample C, d) result of sample D.

6.3 Discussion

The experimental results and system validation prove that this particle-based hair capture method has the ability of 3D hair reconstruction. And it can reconstruct a similar 3D hair model for the target real hair. As shown in section 6.2.2, the system presents capability and stability of processing complex hairstyles with different input quality. And the connection analysis algorithm has the capability to roughly solve the hair strands occlusion and data loss problem. The biggest limitation of the proposed system is caused by Kinect. For example, in order to adapt the resolution of the Kinect depth frame, the system had to compress the high resolution image during image registration, and data loss exists in this stage. And the 2D hair extraction, which affects the reconstruction and rendering result directly, is the key process of the whole system. The experimental results obviously prove that, although the manual hair strand extraction method can help to recover the key hair strands, but it caused more detail loss compare to the automatic one which recovers more hair detail.

Developing and refining the automatic hair strand extraction can help to improve the final 3D hair reconstruction result. And adopting some other new and high performance depth sensors to replace Kinect can be helpful for improving the final reconstruction result as well. These two improvement directions can be discussed in the future.

Chapter 7. Conclusion

This chapter summarizes the presented system, contributions of this research work and points out the direction of future research work.

7.1 Summary

This thesis presents a novel particle based 3D hair reconstruction method using Kinect sensor and DSLR camera. This method combines the 2D image and 3D point cloud to acquire the 3D information of hair strands which are represented as a spatial curve. The proposed system provides high quality detailed hair capture, and its device setting is simpler than many previous capture systems. Our method takes advantage of the 3D depth data from Kinect depth sensors, while the detailed hair strand extraction is done based on high resolution images. We adopted user interactive tool to manually select the hair strand in the high resolution images. After mapping 2D strands to point cloud, the 3D strands are obtained and represented as a set of 3D spline curves rather than a set of primitive surface elements (polygons or patches). Furthermore in order to solve the hair strand occlusion situation, we developed algorithms to analyze the strand segments and build the connection between them. The particle-based 3D hair reconstruction system utilizes the advantages of the particle system, such as the more realistic visualization on hair modeling, mature computation and processing on light tracing, physical simulation, to generate the more realistic hair model which is available for realistic hair rendering and dynamic simulation.

7.2 Contribution

The contribution of this particle-base hair modeling research can be summarized as follows:

- Proposing a novel, cheap and easy setup of hair capture method using a RGB-D camera. The system is able to handle some characteristic hairstyles in practice; it takes the advantage of depth sensor and decreases the workload on the aspect of 3D data acquisition.
- Integrating RGB-D camera and DSLR camera into the system, replacing the Kinect colour image with high resolution image, make the 2D hair strands extraction possible.
- Introducing algorithms to analyze the spatial curve and build the connection between strand segments, the algorithms roughly solve the hair strand occlusion issues.

7.3 Future Work

Although this thesis presents a complete solution of Kinect-based hair capture and reconstruction, the limitations which are brought by the hardware objectively restrict the accuracy and precision of the system. Some future work directions can be discussed to refine the performance of the proposed system. Here, the future work is divided into two aspects: hardware aspect and algorithm aspect.

In hardware aspect, in order to increase the accuracy of the mapping between the depth map and the colour image, the calibration of the Kinect depth sensor and colour camera should be refined. The normal camera calibration method is not suitable for the variable focal length lens camera which is adopted in the presented system. That resulted in error during image registration. Furthermore, a small offset between the colour image and point cloud was hard to be corrected. In order to solve the above issues, more works are needed. The new Kinect sensor Kinect V2 and all-new SDK were released in the summer of 2014. The Kinect V2 has better performance than the older generation Kinect. The data accuracy and precision of Kinect V2 had been greatly improved, we need more test to confirm the performance of Kinect V2, if this device meets the

requirements of the proposed method, the DSLR camera can be removed and the capture system setup can be simplified as well.

In the algorithm aspect, there is plenty of space for performance improvement. With the improvement of the device precision, the reconstruction result can be refined as well. In addition, the presented connection analysis algorithms cannot handle complex hair structure. And the input strand curve was considered as a 2D curve. In the future, with the improvement of data accuracy, the strand curve should be considered as spatial curve in the connection algorithms and the torsion of the spatial curve can be considered in the connection-weight computation. These refinements can help to increase the applicability of the system. With the more accurate data and optimized analysis, each processing stages and the reconstruction result can be improved.

References

[1] Autodesk Maya, available online: http://www.autodesk.com/products/autodesk-maya/overview, Nov 2014.

[2] Sunil Hadap, Nadia Magnenat-Thalmann, "Modeling Dynamic Hair as a Continuum", *Computer Graphics Forum*, Volume 20 Issue 3, pages 329–338, September 2001.

[3] Blender Particle Hair, available online:

[4] http://wiki.blender.org/index.php/Doc:2.6/Manual/Physics/Particles/Hair, Nov 2014.

[5] Wenqi Liang, Zhiyong Huang, "An enhanced frame work for real-tie hair animation", *Pacific Graphics Conference on Computer Graphic and Applications*, page 467, October 2003.

[6] Waiming Kong, Masayuki Nakajima, "Generation of 3d hair model from multiple pictures", in *Proceedings of Multimedia Modeling*, pages 183-196, 1997.

[7] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R. Marschner, Marie-Paule Cani, Ming C. Lin, "A Survey on Hair Modeling: Styling, Simulation, and Rendering", *IEEE Transaction on Visualization and Computer Graphics*, pages 213-234, 2006.

[8] Chuan Koon Koh, Zhiyong Huang, "Real-time animation of human hair modelled in strips", in *Computer Graphics International (CGI*), pages 578-581, Jun 2004.

[9] Cem Yukesel, Scott Schaefer, and John Keyser, "Hair Meshes", *ACM Trans on Graphic (TOG)*, Volume 28 Issue 5 No.166, December 2009.

[10] Thorsten Scheuermann, "Practical Real-Time Hair Rendering and Shading", in *Proceeding of ACM SIGGRAPH 2004 Sketches*, page 147, 2004.

[11] Particle System, available online: http://en.wikipedia.org/wiki/Particle_system, Nov 2014.

[12] Duncan R. Brinsmead, "Patent: Computer based method US 5777619 A", *Silicon Graphics, Inc.*, available online: http://www.google.com/patents/US5777619, Jul 7 1998.

[13] Yosuke Bando, Bingyu Chen, Tomoyuki Nishita, "Animating hair with loosely connected particles", *Computer Graphics Forum (proceedings of Eurographics 2003*), Volume 22 No. 3, pages 411-418, 2003.

[14] Creating Long Hairstyles in Blender, available online: http://cgcookie.com/blender/coming-soon/creating-long-hairstyles-in-blender/, Dec 2014.

[15] Sylvain Paris, Hector Briceno, Franois X. Sillion, "Capture of hair geometry from multiple images", *ACM Transactions on Graphics (Proceedings of the SIGGRAPH conference)*, pages 712-719, 2004.

[16] Sylvain Paris, Will Chang, Wojciech Jarosz, leg Kozhushnyan, Wojciech Matusik, Matthias Zwicker, Frédo Durand, "Hair Photobooth: Geometric and Photometric Acquisition of Real Hairstyles", *ACM Transactions on Graphics (proceedings of the ACM SIGGRAPH conference)*, Volume 27 Issue 3 No.30, Aug 2008.

[17] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guoy Kun Zhou, "Single-View Hair Modeling for Portrait Manipulation", *Siggraph 2012, ACM Transactions on Graphics (TOG)*, Volume 31 Issue 4 No.116, 2012.

[18] Thabo Beeler, Bernd Bickel, Gioacchino Noris, Paul Beardsley, Steve Marschner, Robert W. Sumner, Markus Gross, "Coupled 3D reconstruction of sparse facial hair and skin", *ACM Transactions on Graphics (TOG) - SIGGRAPH 2012 Conference Proceedings TOG*, Homepage archive, Volume 31 Issue 4 No.117, July 2012.

[19] Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, Szymon Rusinkewicz, "Multi-View Hair Capture using Orientation Fields", in *Proceeding of CVPR 2012*, pages 1490-1497, 2012.

[20] Linjie Luo, Hao Li, Szymon Rusinkiewicz, "Structure-Aware Hair Capture", *ACM Transactions on Graphics (TOG)*, Volume 32 Issue 4 No.76, July 2013.

[21] Robert E. Rosenblum, Wayne E. Carlson, Edwin Tripp III, "Simulating the structure and dynamics of human hair: Modeling, rending and animation", *The Journal of Visualization and Computer Animation 2*, pages 141-148, 1991.

[22] Ken-ichi Anjyo, Yoshiaki Usami, Tsuneya Kurihara, "A simple method for extracting the natural beauty of hair", In *Siggraph 1992*, pages 111-120, 1992.

[23] Florence Bertils, Basile Audoly, Bernard Querleux, Frederic Leroy, Jean-Luc Leveque, Marie-Paule Cani, "Predicting nautral hair shapes by solving the statics of flexible rod", *Eurographics (short papers)*, 2005.

[24] Florence Bertils, Basile Audoly, Bernard Querleux, Frederic Leroy, Jean-Luc Leveque, Marie-Paule Cani, "Super-helies for predicting the dynamics of natural hair", *ACM Trans* on Graphics. 25 3, 2006.

[25] Neil Molino, Robert Bridson, Joseph Teran, Ronald Fedkiw, "A crystalline, red green strategy for meshing highly deformable objects with tetrahedra", *In 12th Int. Meshing Roundtable*, pages 103-114, 2003.

[26] Qing Zhang, Jing Tong, Huamin Wang, Zhigeng Pan, Ruigang Yang, "Simulation Guided Hair Dynamics Modeling from Video",*Computer Graphic Forum*, Volume 31 Issue 7, pages 2003-2010, 2012.

[27] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, Richard Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms", *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference*, Volume 1, pages 519-528, June 2006.

[28] Maxime Lhuillier, Long Quan, "A Quasi-dense Approach to Surface Reconstruction from uncalibrated images", *IEEE Trans, On Pattern Analysis and Machine Intelligence*, Volume 27 Issue 3, pages 418-433, 2005.

[29] Linjie Luo, Hao Li, Zhengyou Zhang, Szymon Rusinkiewicz, "Wide-baseline Hair Capture using Strand-based Refinement", *Computer Vision and Pattern Recognition (CVPR)*, pages 265-272, June 2013.

[30] Kurt Konolige and Patrick Mihelich, "Technical description of Kinect calibration", available online: http://www.ros.org/wiki/kinect_calibration/techgnical, Dec 2014.

[31] OpenCV, available online: http://opencv.willowgarage.com/wiki/, Nov 2014.

[32] Wolfgang Boehler, Andreas Marbs, "3D scanning instruments", *CIPA – ISPRS workshop on scanning for cultural heritage recording*, Corfu, Greece, available online: http://www.i3mainz.de/sites/default/files/public/data/p05_Boehler.pdf.

[33] David G. Lowe. "Object recognition from local scale-invariant features", *Computer Vision* (*Proceedings of the Seventh IEEE International Conference)*, pages 1150-1157, 1999.

[34] 3D Scanner, avalable online: http://en.wikipedia.org/wiki/3D_scanner#Technology, Dec 2014.

[35] Yan Cui, Sebastian Schuon, Derek Chan, Sebastian Thrun, Christian Theobalt, "3D shape scanning with a time-of-flight camera", *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on, pages 1173-1180, Jun 2010.

[36] Sebastian Schuon, Christian Theobalt, James Davis, Sebastian Thrun, "High-quality scanning using time-of-flight depth superresolution", *CVPRW '08 (Computer Vision and Pattern Recognition Workshops) IEEE Computer Society Conference*, June 2008.

[37] Kinect for windows, available online: http://www.microsoft.com/en-us/kinectforwindows/, Nov 2014.

[38] Jan Smisek, Michal Jancosek, Tomas Pajdla, "3D with Kinect", *Consumer Depth Cameras for Computer Vision Advances in Computer Vision and Pattern Recognition 2013*, pages 3-25, 2013.

[39] Kourosh Khoshelham "Accuracy Analysis of Kinect Depth Data", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XXXVIII-5/W12, 2011 ISPRS Calgary 2011 Workshop, pages 29-31, August 2011.

[40] Technical Description of Kinect, available online: http://wiki.ros.org/kinect_calibration/technical, Dec 2014.

[41] Rizwan Macknojia, Alberto Chavez-Aragon, Pierre Payeur, Robert Laganiere, "Experimental characterization of two generations of Kinect's depth sensors", *Robotic and Sensors Environments (ROSE) 2012 IEEE International Symposium*, pages 150-155, Nov 2012.

[42] Kourosh Khoshelham, Sander Oude Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications", *Sensors 2012*, 12(2), pages 1437-1454, 2012.

[43] Precision of the Kinect Sensor, available online: http://ros.informatik.uni-freiburg.de/roswiki/openni_kinect (2f) kinect_accuracy.html, Dec 2014.

[44] Kinect for Windows Features, available online: http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx, Nov 2014.

[45] Szymon Rusinkiewicz, Marc Levoy, "Splat: a multiresolution point rendering system for large meshes", In ACM *Siggraph 2000*, New York, NY, pages 343–352, DOI= http://doi.acm.org/10.1145/344779.344940, 2000.

[46] Karim Hammoudi, Fadi Dornaika, Bahman Soheilian, Nicolas Paparoditis, "Extracting Wire-frame Models of Street Facades from 3D Point Clouds and the Corresponding Cadastral Map", *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS)*, Volume 38 Part 3A, pages 91–96, September 2010.

[47] Radu Bogdan Rusu, Steve Cousins, Willow Garage, "3D is here: Point Cloud Library (PCL)", *Robotics and Automation (ICRA), 2011 IEEE International Conference*, pages 1-4, May 2011.

[48] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, Michael Beetz, "Aligning Point Cloud Views using Persistent Feature Histograms", I*ntelligent Robots and Systems, IROS 2008, IEEE/RSJ International Conference on*, pages 3384-3391, September 2008.

[49] Radu Bogdan Rusu, Nico Blodow, Michael Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration", *Robotics and Automation, ICRA '09, IEEE International Conference*, pages 3212-3217, May 2009.

[50] Tahir Rabbani, Frank van den Heuvel, "Automatic Point Cloud Registration using Constrained Search for Corresponding", *7th Conference on Optical*, 2005.

[51] Wenyi Zhao, David Nister, Steve Hsu, "Alignment of Continuous Video onto 3D Point Clouds", *Pattern Analysis and Machine Intelligence*, *IEEE Transactions*, Volume 27 Issue 8, page 1305-1318, Aug 2008.

[52] Paul J. Besl, Neil D. McKay, "A Method for Registration of 3D Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 14 No 2, February 1992.

[53] Iterative closest point, available online: http://en.wikipedia.org/wiki/Iterative_closest_point, Dec 2014.

[54] Zhang, Zhengyou, "Iterative point matching for registration of free-form curves and surfaces", *International Journal of Computer Vision*, Springer 13 (12), pages 119–152, 1994.

[55] Nicolas Burrus, "Kinect Calibration", available online: http://nicolas.burrus.name/index.php/Research/KinectCalibration.

[56] Yang Chen, Gerard Medioni, "Object Modeling by Registration of Multiple Range Images", *Image and Vision Computing*, 10: 145-155, 1992.

[57] Robert Bergevin, Marc Soucy, Herve Gagnon, Denis Laurendeau, "Towards a general multi-view registration technique", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 540-547, 1996.

[58] Lisa Gottesfeld Brown, "A survey of image registration techniques (abstract)", *ACM Computing Surveys (CSUR)*, archive, Volume 24 Issue 4, pages 325 – 376, December 1992.

[59] Ardeshir Goshtasby, "2-D and 3-D Image Registration for Medical", *Remote Sensing and Industrial Applications*, Wiley Press, 2005.

[60] Ardeshir Goshtasby "Piece wise linear mapping functions for image registration", *Pattern Recognition*, Volume 19 Issue 6, pages 459-466, 1989.

[61] Scott E Umbaugh, "Digital image processing and analysis: human and computer vision applications with CVIP tools (2nd ed.)", Boca Raton, FL: CRC Press, 2010.

[62] Suchendra M. Bhandarkar, Yiqing Zhang, Walter D. Potter, "An edge detection technique using genetic algorithm-based optimization", *Pattern Recognition*, Volume 27 Issue 9, pages 1159-1180, September 1994.

[63] Edge Detection, available online: http://en.wikipedia.org/wiki/Edge_detection#cite_note-5, Nov 2014.

[64] Tony Lindeberg, "Edge detection and ridge detection with automatic scale selection", *Computer Vision and Pattern Recognition (CVPR '96)*, pages 465-470, 1996.

[65] Wei Zhang, Fredrik Bergholm, "Multi-scale blur estimation and edge type classification for scene analysis", *International Journal of Computer Vision*, Volume 24 Issue 3, pages 219 – 250, 1997.

[66] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Addison Wesley, pages 414 – 428, 1992.

[67] Roger Boyle, Richard C. Thomas, "Computer Vision: A First Course", Blackwell Scientific Publications, pages 48 – 50, 1988.

[68] Sobel Edge Detector, available online: http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.html, Nov 2014.

[69] Pierre Soille, "Morphological Image Analysis: Principles and Applications", Springer-Verlag, pages 173-174, 1999.

[70] Edward R. Dougherty, "An Introduction to Morphological Image Processing", ISBN 0-8194-0845-X, 1992.

[71] Jean Serra, Pierre Soille (Eds.), "Mathematical Morphology and Its Applications to Image Processing", *proceedings of the 2nd international symposium on mathematical morphology (ISMM'94)*, ISBN 0-7923-3093-5, 1994.

[72] Freericc P. Miller, Agnes F. Vandome, John McBrewster, "Dilation (Morphology)", *VDM Publishing*, Jul 2010.

[73] Kenneth L Judd, "Numerical Methods in Economics", MIT Press, page 225, ISBN 978-0-262-10071-7, 1998.

[74] Wai-Kai Chen, "Feedback, Nonlinear, and Distributed Circuits", CRC Press, pages 9–20. ISBN 978-1-4200-5881-9, 2009.

[75] Spline Interpolation, available online: http://en.wikipedia.org/wiki/Spline_interpolation, Nov 2014.

[76] Les Piegl, Wayne Tiller, "The NURBS Book (2nd. edition ed.)", Springer, ISBN 3-540-61545-8, 1997.

[77] Zhengyong Zhang, "A Flexible New Technique for Camera Calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 22 no 11, pages 1330-1334, 2000.

[78] Izwan Macknojia, Alberto Chavez-Aragon, Pierre Payeur, Robert Laganiere, "Calibration of Network of Kinect Sensors for Robotic Inspection over a Large Workspace", *Robot Vision(WORV), 2013 IEEE Workshop*, pages 184-190, Jan 2013.

[79] Imfill Fill image regions and holes, available online: http://www.mathworks.com/help/images/ref/imfill.html, Nov 2014.

[80] E. T. Y. Lee. "Choosing nodes in parametric curve interpolation", *Computer-Aided Design* 21, pages 363–370, 1989.

[81] Curvature, available online: http://en.wikipedia.org/wiki/Curvature, Nov 2014.

[82] Torsion of a Curve, available online: http://en.wikipedia.org/wiki/Torsion_of_a_curve, Nov 2014.

[83] Moving Average Filter, available online: http://lorien.ncl.ac.uk/ming/filter/filmav.html, Nov 2014.

[84] Liwen Hu, Chongyang Ma, Linjie Luo, Hao Li, "Robust Hair Capture Using Simulated Examples", *ACM Transactions on Graphics*, *Proceedings of the 41st ACM SIGGRAPH Conference and Exhibition*, Volume 33 Issue 4, 2014.

[85] Blender 2.6 Manual, Particles Children, available online: http://wiki.blender.org/index.php/Doc:2.6/Manual/Physics/Particles/Children, Nov 2014.

[86] Kent Trammell, "Styling and Rendering Long Hair with Blender and Cycles", available online: http://cgcookie.com/blender/cgc-courses/styling-and-rendering-long-hair-with-blender-and-cycles/, Nov 2014.

[87] Fatemeh Cheraghchi, Won-Sook Lee, "Hair Strand Extraction by Template Matching using Gabor Filter and Histogram", *4th International Conference on Image Processing Theory*, Tools and Applications, Paris France ,October 2014.

[88] Chao Sun, Fatemeh Cheraghchi, Won-Sook Lee, "2D Hair Strands Generation Based on Template Matching", *IMULTECH2014 (4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications)*, Vienna Austria, August 2014.

Related Publications by the Author

1. Zhongrui Li, Chao Sun, Won-Sook Lee and Ig-Jae Kim, "Hair Modeling using Kinect Sensors and DSLR Cameras", *International Conference on Ubiquitous Robots and Ambient Intellivence (URAI 2014)*, Kuala Lumpur Malaysia, Nov 2014.
2. Chao Sun, Zhongrui Li, Won-Sook Lee, "Relatively straight medium to long hair reconstruction using Kinect Sensors", *The International Conference on Computer Graphics Theory and Applications (Grapp 2015)*, Berlin Germany, March 2015.