University of Tartu

Faculty of Science and Technology

Institute of Technology

Egils Avots

Garment retexturing using Kinect V2.0

Masters Thesis (30 ECTS)
Robotics and Computer Engineering

Supervisors:

Assoc. Prof. Gholamreza Anbarjafari
Assoc. Prof. Sergio Escalera

Tartu 2017

Resümee/Abstract

Rõivaste tekstureerimine kasutades Kinect V2.0

Selles töös kirjeldatakse 3 uut rõivaste tekstureerimismeetodi. Need meetodid on mõeldud FitsMe virtuaalriietusruumi rakendustele ning nendes kasutatakse Microsoft Kinect II RGB-D kaameraid.

Esimeses meetodis rakendatakse ühe ainsa RGB-D pildi ja infrapuna informatsiooni Kinekt II'st, et automatiseerida rõivaste tekstuuri loomist. Selle meetodi puhul segmenteeritakse rõivad pildist välja, kasutades GrabCut või sügavuse põhjal segmenteerimist. Järgnevalt arvutatakse iga rõiva piksli jaoks tekstuuri domeeni koordinaadid, kasutades normaliseeritud 3D informatsiooni. Tekstureeritud pildi abil määratakse värvidele varjutused.

Teine selles töös kirjeldatud meetod kasutab 2D'st 3D rõivaste tekstureerimist, kus mannekeeni või inimeste rõivaste segmendid ühildatakse uue allika rõivaga ja tekstureeritakse. Tulemusena tekivad augmenteeritud pildid, kus uued rõivad on kantud mannekeenile või inimesele. Rõivaste tekstureerimise probleem jagatakse kaheks osaks. Esimene neist on rõivaste piirjoonte tuvastus, kus kasutades seadepunkti registreerimisel põhinevaid Gaussian segu mudeleid. Sisemised punktid interpoleeritakse geodeetilise tee pinna topoloogiaga. Tulemusena tekib palju realistlikum tekstuur kui olemas olevate meetoditega.

Viimane osa antud teesis tutvustab uut 3D rõiva tekstuuri resolutsiooni suurendamise meetodit. Selles meetodis kasutatakse samu Kinecti kaadreid, mis olid rakendatud mudeli loomises. Esiteks luuakse 3D mudelist struktureeritud polügooni pind, ning selle saavutamiseks mähitakse baasmudeli ümber 3D mudel. Baasmudelil on varasemalt defineeritud servad ja tekstuuri märkmed. Kaadrid ühendatakse loodud mudeliga ja kiirte jälituse abil kujutatakse Kinekti kaadri värvi väärtused 3D mudeli UV kaardil.

CERCS: T111 Pilditehnika

Märksõnad: Kinekt, Infrapuna pilt, Rõivaste seadistamine, Rõivaste tekstureerimine, Tekstuuri kaardistamine

Garment retexturing using Kinect V2.0

This thesis describes three new garment retexturing methods for FitsMe virtual fitting room applications using data from Microsoft Kinect II RGB-D camera.

The first method, which is introduced, is an automatic technique for garment retexturing using a single RGB-D image and infrared information obtained from Kinect II. First, the garment is segmented out from the image using GrabCut or depth segmentation. Then texture domain coordinates are computed for each pixel belonging to the garment using normalized 3D information. Afterwards, shading is applied to the new colors from the texture image.

The second method proposed in this work is about 2D to 3D garment retexturing where a segmented garment of a manikin or person is matched to a new source garment and retextured, resulting in augmented images in which the new source garment is transferred to the manikin or person. The problem is divided into garment boundary matching based on point set registration which uses Gaussian mixture models and then interpolate inner points using surface topology extracted through geodesic paths, which leads to a more realistic result than standard approaches.

The final contribution of this thesis is by introducing another novel method which is used for increasing the texture quality of a 3D model of a garment, by using the same Kinect frame sequence which was used in the model creation. Firstly, a structured mesh must be created from the 3D model, therefore the 3D model is wrapped to a base model with defined seams and texture map. Afterwards frames are matched to the newly created model and by process of ray casting the color values of the Kinect frames are mapped to the UV map of the 3D model.

CERCS: T111 Imaging, image processing

Keywords: Kinect, Infrared Image, Garment matching, Garment Retexturing, Texture Mapping

Table of contents

List of figures

List of tables

Abbreviations, constants, definitions

CPD - Coherent Point Drift

FHD - Full High Definition images with size of 1920x1080 pixels

GUI - Graphical User Interface

ICP - Iterative Closes Point

IR - Infrared

IRT - IR-based Retexturing Method

LPQ - Local Phase Quantization

MOS - Mean Opinion Score

MSE - Mean Square Error

NRICP - Non-Rigid Iterative Closes Point

RGB - Red, Green, Blue color space used in digital image systems

RGB-D - image in RGB color space and a Depth image for the same scene

SDK - Software Development Kit

TPS - Thin Plate Splines

UV - normalized texture coordinates where x,y axis are represented as u,v

YCbCr - Luminance, Chroma blue, Chroma red color space used in digital image systems

1 Introduction

In fashion industry companies choose to display their products in digital format. Mostly this digital media consists of 2D images of garments. As one could predict the next steps could be augmented garments or garment 3D models. Such step has taken company FitsMe with an aim to create something similar to virtual fitting room. Currently their top selling service is fit adviser which suggest the best garment size based on body parameters provided by the user. In addition to the fit advice, it also displays an example of how the garment would look like if a person with the given body shape was wearing it. Such images were created with special robots who can change their body shape within given parameters. Usually the service would show a similar or a base model of the garment, this is because producing the image database for all sizes is an expensive process, therefore crating a database for all garments at FitsMe disposal is not profitable. To find a more automated solution a research and development project with FitsMe and University of Tartu was created. The project itself was divided in two major groups: 3D model creation and garment retexturing. It was decided to use Kinect 2 RGB-D camera as the main data capturing device, therefore the model creation and retextuirng is also based on data provided by Kinect 2.

This document will mostly address the proposed retexturing methods and few steps of the 3D model creation as well. By having the ability to retexture garments it is possible to reduce the amount of scans needed for the virtual fitting room. The retexturing can be divided in three sub sections: plain texture, matched texture and UV mapped texture for 3D models. A plain texture can be used to change the appearance of an garment form single Kinect frame. This is useful if the color or pattern of an garment needs to be changed. The matched texture addresses the problem of matching a flat image of a garment to a garment captured by single Kinect frame. Thirdly, the UV texture mapping refers to mapping a texture from the Kinect scans for a 3D model which was also captured and created using data from Kinect.

2 Hardware

Kinect 2 is a RGB-D device (Fig. 2.1), with FHD color image and 512x424 depth map. For depth estimation Kinect 2 uses time-of-flight (ToF) sensors. These sensors calculate distance based on speed of light, where time-of-flight of the light signal is measured between the sensor and the scene. ToF sensors only need one light pulse to capture entire scene, thats why such sensors are classified as scanner less LIDAR. Kinect 2 uses an array of emitters to send out square wave modulate infrared signals (in 860 nm range) that are sent out in the scene, the signals get reflected back and captured by CCD (charge-coupled device) image sensor. Additionally, it is possible to obtain an image showing how the scene has absorbed or reflected the emitted infrared light. In depth information about the process of depth estimation using Kinect 2 can be found in [5, 6].

The Kinect 2 SDK software offers a verity of applications, for example skeleton tracking, face tracking and many more. For the purpose of proposed applications, a Kinect capture sample code was used to extract color image, depth information, depth to color mapping, infrared information and depth to space table which contains Kinect 2 calibration information. Using this data it is possible to represent the scene as a colored point cloud.

Kinect 2 is a excellent depth capturing device for gaming, but there are few limitations when using it as a depth measuring device. Firstly, errors appear around objects edges, this happens when the emitted light hits a surface in such an angle, that the reflected signals is not detectable by Kinect. In such cases Kinect assigns zero depth, meaning that the data is missing. Secondly, a similar error is observed when scene contains reflective/polished surfaces, which can also crate randomly scattered points. Another problem is directly related to the stability of the system, meaning that the measured depth distance fluctuates, due to this effect the surfaces have wave like patterns. For static scenes, frame averaging is used to get more stable output.

Figure 2.1: Kinect V2.0 camera [1]

According to [7], Kinect 2 can capture frames starting from 0.5 meters and has depth accuracy error smaller than 2 mm in the cater part of the frame. The error increases towards edges of the frame, and it also increases with greater measurement distances. The best distance for scanning objects is the 0.5 to 2m range. To achieve the best depth resolution, the people and manikins were scanned at a distance of 1.5 to 2 meters where the error in the horizontal and vertical plane is the smallest.

3 IRT

3.1 Introduction

The following method describes how a Kinect frame can be retextured with a simple texture. In this case a simple texture refers to an image containing an texture or a different color. There are several steps between taking an RGB-D picture and displaying the final result with a retextured garment. These steps involve segmentation of the garment, mapping the texture to the Kinect frame and surface shading. The novelty of this approach lies in the retexturing part, which involves several challenges. First, a coordinate map must be created between the image of the new texture and the image that is being retextured. This problem is especially difficult in the case of non-rigid and easily transformable surfaces like clothes. Another challenge is to shade the new texture correctly without knowing the lighting, intensity and the original colour of the surface.

A texture is necessary to give 3D objects life like appearance where colour information is incorporated into a model, which may be the result of a 3-Dimensional (3D) reconstruction process [8]. To add colour onto the surface of the object, a mapping is used that maps planar texture to the surface which is referred as texture mapping [9]. Few examples of techniques for doing such task are using an intermediate 3D shape [10], direct drawing onto the object [11, 12] and curve fitting [13–15].

Although texture mapping can be done to any 3D objects with different methods, this document will focus on non-rigid garment retexturing using a recently proposed retexturing method [2] referred to as the IR-based Retexturing Method (IRT), which overcomes the problem of not knowing the shading parameters, and offers reliable, automatic performance. The underlying notion and the main novelty of the IRT lies in the shading process, which is colour- and texture-invariant.

3.2 Literature review

Due to increase in depth cameras like Kinect [16], new advances and research in object retexturing are emerging. The following section will shortly describe the current advances in object retextuing. The advances in 3D virtual garment appearance has encouraged many researchers to engage in attempts to improve the retexturing quality during the last decade [17–19].

White and Forsyth describe a method [20] that uses prior knowledge of the texture of a surface and that allows them to choose markers. They are trying to match the markers with the ones on the transformed surface and to create a map between them. They are using a simplified approach where they have only a small number of colours, which enables them to guess the original colour on the image and to obtain lighting intensity at that point.

A rather popular texture fitting method was proposed in [21], allowing the users to sketch garment contours directly onto a 2-Dimensional (2D) view of a mannequin. Improved versions where proposed by others [22, 23].

Pilet et al. [24] also perform image registration between an image in its original form with no rotations in uniform lighting conditions and the same image, which has different lighting, occlusions or rotations. They are using the Expectation Maximisation framework that also provides visibility and lighting maps. The resulting method is able to handle difficult lighting conditions and occlusions.

Guo et al. [25] describe a method that operates under the assumption that locally, the depth gradients are proportional to the intensity gradients. They compute the texture coordinates by solving an energy minimisation problem. The lighting problem is solved by converting to $YCbCr$ colour space and using the Y component as the intensity value for the new texture colour.

Shen et al. describe a somewhat similar retexturing algorithm [26] that works on high dynamic range images. They use image luminance to create an approximate depth map, calculate gradients of the depth map, and compute new texture coordinates and colours based on the previously obtained depth gradient map.

Zelinka et al. describe a material replacement system [27] where the interactively chosen area is retextured by first approximating the surface normals by assuming the Lambertian reflection model and improving it with the Gaussian mixture model. The new texture is synthesised with the jump-map technique.

Hilsmann and Eisert describe a method [28] that assumes a uniformly coloured surface with a small, easily detectable textured area on it. The texture is tracked with a video sequence using optical flow, and the occlusions are estimated based on that motion. As the main original colour of the surface is previously known, it is possible to easily estimate the lighting. In the textured area the lighting is estimated by interpolation using the light intensity of the uniformly coloured area.

Possible approaches to perform image registration for transformed non-rigid surfaces are discussed by Bartoli and Zisserman [29]. They are using Radial Basis Functions for regularising the optical flow field. Chui and Rangarajan [30] also discuss the same problem and propose a general point matching framework for non-rigid surfaces that is based on thin-plate splines.

Pizarro and Bartoli [31] focused on solving the self-occlusion problem; they used marker-based templates to reconstruct a deformed surface, which enabled them to achieve good results in retexturing.

Khan et al. [4] create a rough reconstruction of a 3D surface by assuming that 3D distances can be approximated using image intensity. Texture coordinates are approximated based on the obtained gradients of the constructed surface, and the shading is based on interpolating between the original colour and the new texture colour.

A somewhat similar approach is taken by Shen et al. [3], but they use gradients of image intensity instead of a triangulation. Based on the gradient field the Poisson equation is solved and the results are used to compute texture domain coordinates. The shading procedure is also based on the $YCbCr$ colour space chroma replacement.

Kerl et al. [32] describe a simple method for estimating the pure albedo of the texture, in order to remove illumination effects from IR and colour images using Kinect II RGB-D sensors. They estimate the IR albedo from infrared and depth images, and transfer the former to the colour image. Using this approach, it is possible to create a colour shading model which includes all illumination effects and the colour albedo image.

3.3 Description of IRT method

The IRT uses images from the Microsoft Kinect II camera, along with the depth information and real-world coordinates. Algorithm replaces the texture of the shirt with a desired reference texture, through performing the following three tasks: segmentation, retexturing and shading. In other words, the part of the image corresponding to the garment worn by the subject is first segmented out, and then retextured by calculating the texture-domain coordinates for each pixel in the area of interest, followed by applying shading to the colour information. The pseudo-code is provided in Algorithm 1.

3.3.1 Segmentation

Segmentation part is not the main focus of the discussed method, therefore the segmentation method should be chosen based on the context in which IRT is used, for example a white fitting room where people are asked to wear green shirts could be easily handled with colour thresholds. In our case people were wearing different garments. To show the best possible results for human subjects, a semi-automatic method GrabCut [33] was used, where the user has to leave marks that could help the software to learn the background and region of interest. The aim of the segmentation is to obtain a binary image where *ones* represent the area of interest and *zeros* the background.

3.3.2 Texture Mapping

Mapping is performed by determining the location of the pixel on the reference texture based on the normalized depth value. The Kinect API enables the translation of screen coordinates of an image into real world coordinates by using following map:

$$\omega : (x, y) \to (X, Y, Z)\,, \tag{3.1}$$

where $x, y \in \mathbb{N}$ denote the screen coordinates of the image, and $X, Y, Z \in \mathbb{R}$ stand for the real-world coordinates corresponding to them. The screen coordinates are mapped to texture coordinates using the functions f_u and f_v, as follows:

$$u = f_u\,(x, y) = W\,\frac{\omega_x\,(x, y) - X_{min}}{X_{max} - X_{min}}, \tag{3.2}$$

$$v = f_v\,(x, y) = H\,\frac{\omega_y\,(x, y) - Y_{min}}{Y_{max} - Y_{min}}, \tag{3.3}$$

Input: ColourImage, IrImage, TextureImage, CoordinateMapper, IrLow, IrHigh
Output: RetexturedImage

$SegmIm \leftarrow segmentImage(ColourImage)$
$W \leftarrow ColourImage.width$
$H \leftarrow ColourImage.height$

$IR_{min} \leftarrow \min\{IrImage(x,y) \ : \ SegmIm(x,y) = 1\}$
$IR_{max} \leftarrow \max\{IrImage(x,y) \ : \ SegmIm(x,y) = 1\}$
foreach $(x,y) \in IrImage$ **do**
    $IrImage(x,y) \leftarrow IrLow + \frac{IrHigh - IrLow}{IR_{max} - IR_{min}} \cdot (IrImage(x,y) - IR_{min})$
end

foreach $(x,y) \in ColourImage$ **do**
    $i \leftarrow 0$
    if $SegmIm(x,y) = 1$ **then**
        $(X,Y,Z) \leftarrow CoordinateMapper(x,y)$
        $XCoords[i] \leftarrow X$
        $YCoords[i] \leftarrow Y$
        $i \leftarrow i + 1$
    end
end

foreach $(x,y) \in ColourImage$ **do**
    if $SegmIm(x,y) = 1$ **then**
        $(X,Y,Z) \leftarrow CoordinateMapper(x,y)$
        $u \leftarrow W\frac{X - \min(XCoords)}{\max(XCoords) - \min(XCoords)}$
        $v \leftarrow H\frac{Y - \min(YCoords)}{\max(YCoords) - \min(YCoords)}$
        $ColourImage(x,y) \leftarrow \frac{1}{255} IrImage(x,y) \cdot TextureImage(u,v)$
    end
end

Algorithm 1: A pseudo-code representing the IRT [2].

where u and v are the new texture domain coordinates, $\omega_x(x,y)$ and $\omega_y(x,y)$ denote the X and Y coordinates obtained from the map $\omega(x,y)$, and $X_{min},\ Y_{min},\ X_{max}$ and Y_{max} stand for the smallest and largest X and Y coordinates of the pixels from the area of interest in the segmented image obtained in the last step. W and H are the width and height of the texture image.

3.3.3 Shading

The IR image from Kinect 2 shows the amount of IR light reflected from the surface of an object. The shading effects are calculated using a linear map that transforms the IR image pixels to their counterparts on the RGB image.

Assuming that a colour $c = (r, g, b)$, with its red, green and blue components, is obtained from the texture image, and the colour of the corresponding pixel on the RGB image is $c' = (r', g', b')$. The colour c' depends on the scene lighting condition and can be expressed as a linear multiplier of the original colour of the pixel, $c^* = (r^*, b^*, g^*)$, as follows:

$$c' = l\,(x, y)\, c^* = (lr^*, lb^*, lg^*)\,, \tag{3.4}$$

where l denotes the light intensity at that point. Then the colour of the texture image can be treated as the new true colour in maximum lighting intensity, and l is used as a coefficient to transform it through the same map, and find the shaded colour, i.e. $lc = (lr, lg, lb)$.

Afterwards histogram stretching is used to bound its values within the range $\left[\frac{\alpha}{255}, \frac{\beta}{255}\right]$, where the values of α and β have to be determined in a case-by-case manner, depending on the experimental setup. As the last step the lighting intensity approximation is calculated as follows:

$$l\,(x, y) = \alpha + \frac{\beta - \alpha}{Ir_{max} - Ir_{min}} \cdot (Ir\,(x, y) - Ir_{min})\,, \tag{3.5}$$

where Ir_{max} and Ir_{min} are the maximum and the minimum IR intensities in the segmented image area.

3.4 Evaluation

The results ware compared with the method proposed by Shen et al. [3] and with the method of Khan et al. [4]. These methods were selected for comparison because they operate under conditions similar to IRT method: Both methods use static images, and do not depend on any prior knowledge about the surface pattern or markers. The methods were compared using mean opinion score (MOS), with IRT having 566 out of 800 votes [2], meaning that the IRT retexturing method produces more realistic retextured images than the other two methods.

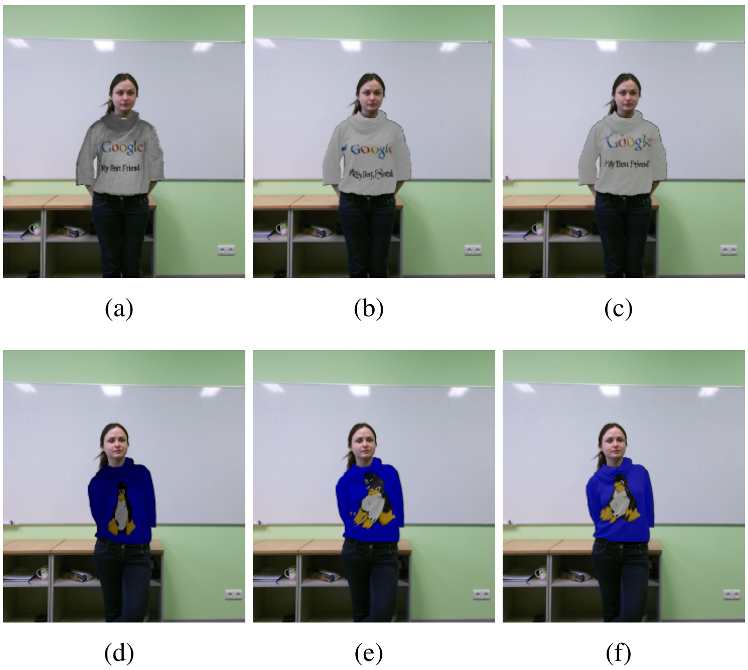


Figure 3.1: A subset of the images that were used in the MOS comparison [2]. From left to right, the images are created with the IRT, the method of Shen, Sun [3] and the method of Khan, Reinhard [4].

4 2D to 3D garment matching

4.1 Introduction

There exist several standard methods for projecting textured surfaces on screen. The simplest shading methods work only by using surface normals independently without considering the overall surface, attempting to estimate the brightness of the surface given some known viewer and light source direction. Examples of this kind of method are the Gouraud shading, Phong shading, and Blinn-Phong shading [34].

The proposed automatic retexturing method, after the segmentation stage uses the point set registration method [35] to find correspondence between the outer 2D contours of the person and the target garment. After the contour matching, the surface topology of the flat 2D garment is approximated using geodesic distance in a global closed form solution using thin plate spline (TPS) [36] and the final result is superimposed onto the segmented area.

4.2 Literature review

Due to the fact that trying on clothes is time-consuming, a virtual alternative has always been desired, and many researchers have been engaged in developing novel strategies and systems to perform such a task [37–41]. It requires scanning, classification of the body based on gender and size, 3D modelling [42–44] and visualisation. Constrained texture mapping and parametrisation of triangular mesh are some popular examples, although they suffer from some deficiencies such as finding the parameter values and manual adjustments [45, 46]. Many researchers have also suggested methodologies for visually fitting garments onto the human body based on dense point clouds [47, 48]. However, garment retexturing in a virtual fitting room is still an open problem [16, 49, 50].

The matching problem stage can be defined as a correspondence problem, which incorporates pair-wise constraints. Hence, it is often solved with a graph matching approach [51–53], which is especially suitable for deformable object matching. Furthermore, additional constraints can be added to the framework in order to reduce the computation time, or in order to take problem-specific aspects into account.

There exist various techniques for conducting a mapping from 2D image texture space to a 3D surface. Some examples are intermediate 3D shape [54], direct drawing onto the object [55], or using an exponential fast marching method by applying geodesic distance [14, 56]. Many researchers have devoted special attention trying to attempts to enhance the realism of virtual garment representation during the last decade [57]. One of the most frequently used texture fitting methods was proposed by Turquin et al. [58], which allows the users to sketch garment contours directly onto a 2D view of a mannequin.

Another popular way of mapping a 2D texture onto a 3D surface is by using a single image [59]. As proposed by [52], an estimation of a 3D pose and shape of the mannequin is followed by constructing an oriented facet for each bone of a mannequin according to angles of the pose, and projecting the 2D garment outlines into corresponding facets. Eckstein et al. [60] proposed a constrained texture mapping algorithm, which can be used for 2D and 3D modelling, and multi-resolution texture mapping and texture deformation, but it may produce a Steiner vertex effect when a simple solution does not exist. Kraevoy et al. [61] introduced a method based on iterative optimisation of a constrained texture mapping method. In their method, it is a requirement to specify the corresponding constraint points on the grid model and texture image, the parametrised mesh. Later, Yanwen et al. [62] reported a constrained texture mapping method based on harmonic mapping, with interactive constraint selection by the user; the method produces high efficiency, real-time optimisation, and adjustment of mapping results. The block based constrained texture mapping methods are also used in order to bring higher speed and lower computational costs [63].

4.3 Retexturing approach

As preprocessing, we use RGB, depth and infrared images of the Kinect and segment out the garment from the background. The segmented depth image is used to compute retexturing from a source 2D flat garment image. We reduce the problem of surface point matching to an interpolating problem by using garment contour matching. The interpolation process takes surface topology into account using geodesic distance in a global closed form solution using thin plate spline (TPS) [36]. Thus, 2D garment contours are matched beforehand applying point registration based on Gaussian mixture models [35]. Finally the resulting mapped source image is sampled, and the segmented area can be superimposed using these colours. The proposed retexturing method is visualised in Fig. 4.1.

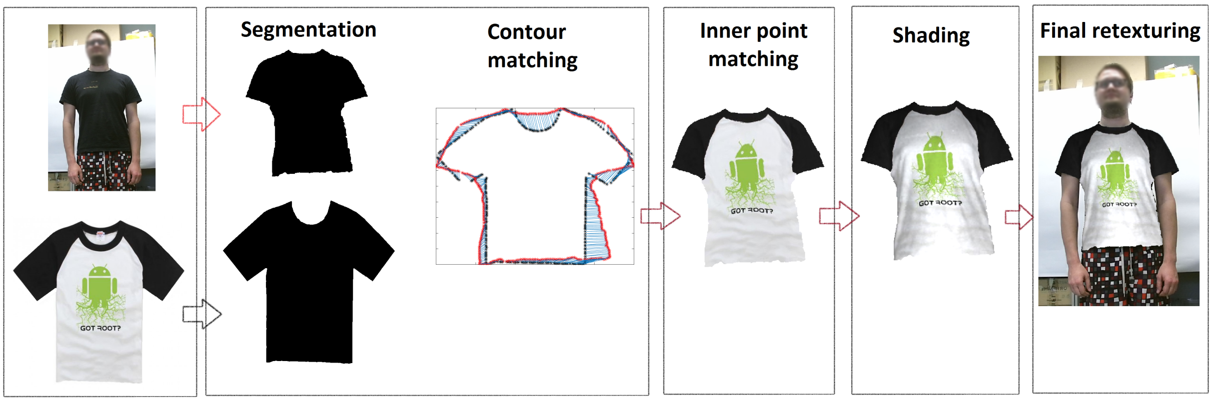


Figure 4.1: Overview of the proposed retexturing method.

4.3.1 Segmentation

The first step of the proposed retexturing method is segmentation of garments from the background. It is necessary to extract a set of points from the image corresponding to the area being retextured. The proposed method works under the following assumptions: the area to be retextured is a shirt (or some other initially known garment) worn by a manikin or a person.

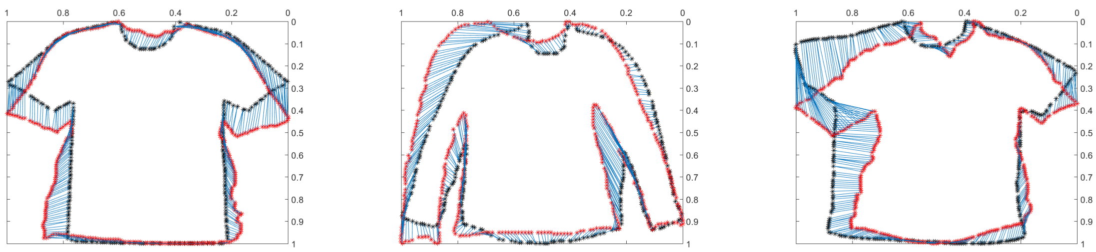
4.3.2 Outer contour matching

Contour matching can be viewed as a point set registration problem, where a correspondence must be found between a scene and a model. A few of the most well known methods for point set registration are Iterative Closest Point [64], Robust Point Matching [65, 66], and Coherent Point Drift [67] algorithms. For our purposes, a correspondence must be found between highly deformed shapes. Out of available algorithms, we have chosen to use non-rigid point set registration using Gaussian mixture models [35] because of its accurate fitting under different conditions and fast execution time. Additionally, Gaussian mixtures provide robust results even if the shapes have different features, such as different neck lines, hand positions and folds.

Let's define the contour of a garment on a real person as C_R and the contour of the flat garment as C_F. The aim is to create a correspondence between contour models C_R and C_F. In the point matching algorithm, the point sets are represented by Gaussian mixture models. The interpretation is such that a statistical sample is drawn from a continuous probability distribution of random point locations. Afterwards the point set registration problem is viewed as an optimisation problem, meaning that a certain dissimilarity measure between the Gaussian mixtures constructed from the transformed model set and the fixed scene set is minimised based on L2 distance between the mixtures.

All compared results were produced with the same set of parameters that were determined empirically. The setup parameters for matching the contours needed for the point registration algorithm [35] are set as follows (see original paper for the definition of parameters): sigma, which is the scale parameter of Gaussian mixtures, is set to 0.2 and 0.1, and the maximum number of function evaluations at each level is set to 50, 500, 100, 100 and 100. The point registration algorithm uses contours with 400 points. After the transformation and point correspondence are found, the contour is further down-sampled to 120 points and used for the inner point matching. A larger number would have resulted in a long computation time, whereas a smaller number of points resulted in some undersampled parts and produced inferior mappings. 120 points were chosen as a compromise between the execution time and the resulting mapping quality.

Before finding the corresponding points between the shapes, the contours are down-sampled and normalised. Essentially the used method provides information about how C_R has to be transformed to match C_F. After the transformation is found, nearest neighbour search is used to find the corresponding points between the two contours. Outer contour matching examples are shown in Fig. 4.2.



(a) (b) (c)

Figure 4.2: Short and long sleeve examples for contour correspondences obtained using point set registration. Red contour corresponds to C_R and blue contour corresponds to C_F.

4.3.3 Inner contour matching

Inner contour matching refers to the process of finding correspondence points between the body surface and the 2D flat garment in order to assign to each body point a colour from the garment. To do so, a triangulated 3D mesh is generated based on the depth image of the segmented area. A solution can be obtained by finding an affine deformation matrix for each face triangle to bring both source and target surfaces into alignment according to the matched points of the outer contours. By using thin plate splines (TPS) [36] as a solution in closed-form based on a radial basis kernel. Let $X = \{x_1, ..., x_N\} \in \mathbb{R}^3$ be the set of all points belonging to the segmented and discretized body surface Ω. Then, a mapping from x_i to the source image is computed through

$$W(x_i) = \sum_{j=1}^{n} \omega_j \kappa(\|x_i - C_{R_j}\|), \tag{4.1}$$

where ω is a set of trained coefficients based on C_R and C_F, $\kappa(d) = d^2 \log d$ is a radial basis kernel and n is the number of contour points. This basic formulation is based on Euclidean distance among the points which is not applicable for the problem since contour points do not cover all the surface, instead a geodesic-based distance is used to include surface topology.

The fast marching algorithm of [68] is used to compute a fast and accurate approximation of geodesic distance. The fast marching algorithm is closely related to the Dijkstra algorithm with the difference that it satisfies the Eikonal equation $\|\nabla U(x)\| = 1/s(x), x \in \Omega$ to update the graph where $\nabla U(x)$ is the gradient of the action map U and $s(x)$ is a positive outwards speed function at point x. $U(x)$ is a function of time at point x that describes the evolution of the surface with respect to $s(x)$ and surface gradient. The surface is assumed to be differentiable at all points. Starting from x_i, at each iteration, the algorithm sweeps outwards one grid point with respect to $s(x)$ to locate the proper grid point to update. Then geodesic distance can be computed for two vertices v_i and v_j from the shortest path $L = \{L_1, ..., L_m\}$ by

$$\Gamma(v_i, v_j) = \sum_{l=1}^{m-1} \|L_l - L_{l+1}\| \tag{4.2}$$

Afterwards the TPS formulation is used to compute the coefficient matrix ω as

$$\omega = \begin{bmatrix} \dot{K}_{n\times n} + \lambda I & [1|C_{R_{n\times 3}}] \\ [1|C_{R_{n\times 3}}]^\top & 0 \end{bmatrix}^{-1}_{(n+4)^2} \begin{bmatrix} C_{F_{n\times 3}} \\ 0 \end{bmatrix}_{(n+4)\times 3}, \tag{4.3}$$

where $\dot{K}_{ij} = \Gamma(C_{R_i}, C_{R_j})^2 \log \Gamma(C_{R_i}, C_{R_j}) \forall i, j \in \{1, ..., n\}, i \neq j$. λI is a regularization term and is added to the kernel $\dot{K}$ where I is the identity matrix and $\lambda \in \mathbb{R}$. The following solution can be achieved by applying trained coefficients as

$$W = [\ddot{K}_{N\times n}|1|X_{N\times 3}]\omega \tag{4.4}$$

where $\ddot{K}_{ij} = \Gamma(X_i, C_{R_j})^2 \log \Gamma(X_i, C_{R_j})$. Matrix W includes warped points to the 2D shirt image.

4.3.4 Shading

The shading effect of the garment is achieved using an adaptation of IRT method. The general procedure for obtaining the final visualisation is as follows. The point cloud corresponding to the area of interest provided by the Kinect 2 camera is triangulated and rendered as described in the previous section. The image created as a result of mapping in the previous steps is used as a texture image, such that each vertex corresponds to a point on the image. Afterwards, the rendered image is modified by the corresponding infrared values for each pixel. Finally, the segmented area in the Kinect frame is replaced by the colour information from the previous step.

4.4 Evaluation

Each image in the database contains a person facing the camera in a pose that does not significantly occlude the worn garment. The garments segmented from the original database were retextured using another database consisting of images of flat shirts. The first data set contained 91 retextured images with 14 people (11 males and 3 females). This data set used 13 flat garments (4 long sleeve garments and 9 t-shirts). The second data set contained 39 retextured images with 5 people (4 males and 1 female). This data set used 8 flat garments (4 long sleeve garments and 4 t-shirts). The garments in second data set had 16 physically attached landmarks. This was done in order to determine the retexturing precision by retexturing the same garment onto itself. Fig. 4.3(a) shows a sample of a real put-on image and the landmarked garment itself.

The method was evaluated with two metrics: qualitative comparison using the mean opinion score (MOS), and quantitative comparison using the mean square error (MSE). The MOS score was measured by showing 91 sets of images from the first dataset to 41 people. Each person was asked for an opinion about which one of the images in each set looks visually more realistic.

The MSE was measured on the second dataset by retexturing the flat version of the shirt and computing the average distance from retextured landmarks to ground truth landmarks. Fig. 4.3(b) shows the landmark displacement. The method is compared to nonrigid Iterative Closest Point (NRICP) [69] and Coherent Point Drift (CPD) [67] method, using introduced evaluation metrics.

Long and short sleeve are analysed separately in Table 4.1 and in Fig. 4.5. It can be seen that the final retextured image still has a realistic appearance even with small misalignment in outer correspondences.

Table 4.1: Mean Opinion Score (MOS) comparison

Method	T-shirt Votes	T-shirt Percentage	Long sleeve Votes	Long sleeve Percentage
NRICP [69]	77	2.68%	32	3.69%
CPD [67]	485	16.88%	245	28.23%
Ours	2311	80.44%	591	68.09%

The MSE values shown in Table 4.2 indicate that in most cases our method is more accurate than state-of-the-art methods regarding marker distances to ground truth. Often our method

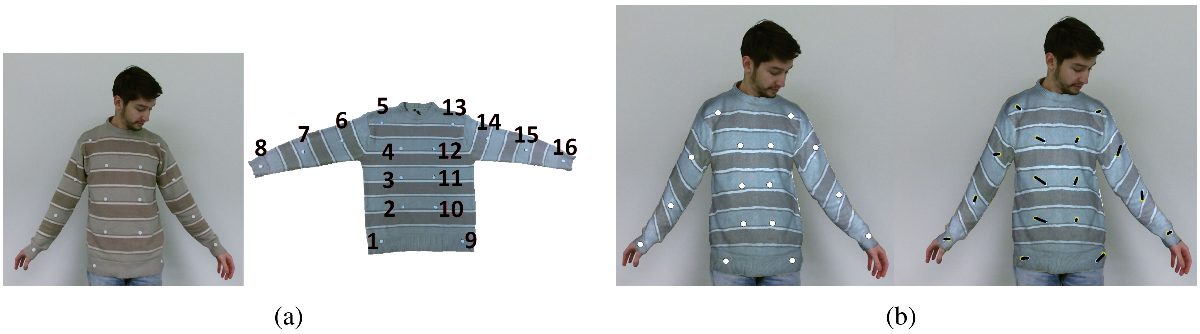


Figure 4.3: a) Landmarks used in the second dataset on the flat garment (right) and landmark locations after putting the garment on as ground truth. Landmarks are shown by indices for comparison purposes. b) Retextured garment and estimated landmarks (left) and displacement to ground truth landmarks (right) for computing error.

performs better than other methods for almost each marker in Fig. 4.4 where landmarks are the white circles that are placed on the garment, as is shown in Fig. 4.3. Landmarks numbers 8 and 16, which were placed at the end of the sleeves, have the highest error in both long and short sleeve garments, because of slight point misalignment in outer contour matching.

Table 4.2: Marker mapping error

Method	MSE for T-shirts	MSE for Long sleeves
NRICP [69]	115.400 px	215.349 px
CPD [67]	83.850 px	190.618 px
Ours	75.005 px	105.884 px

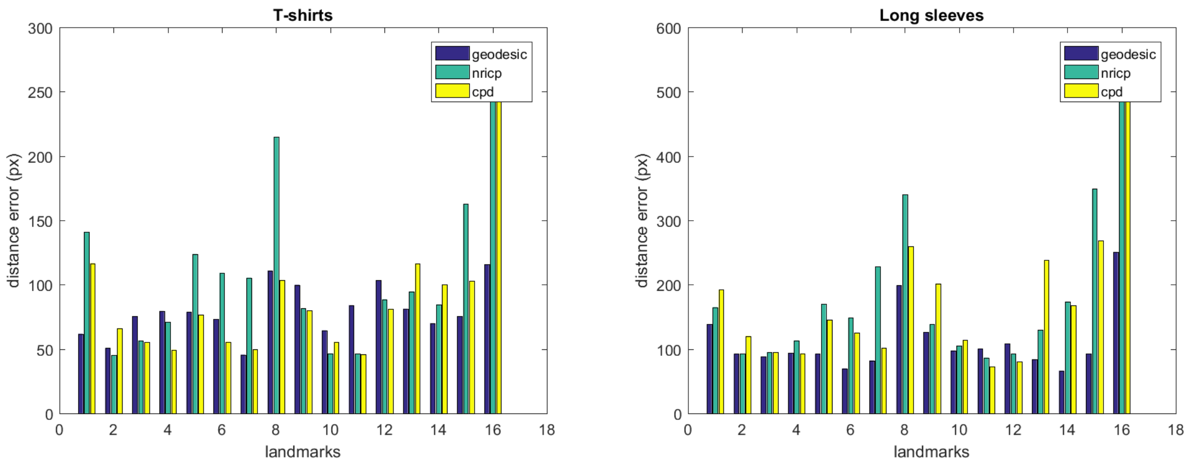


Figure 4.4: Landmark pixel distance error for t-shirts (left) and long sleeves (right)

(a) (b) (c) (d) (e) (f)

Figure 4.5: Images created by the proposed retexturing method, (a) is the original image, (b) is the image of a shirt, (c) shows the shape correspondence, (d) is the retextured image based on the geodesic mapping, (e) is mapping using CPD and (f) is mapping using the NRICP algorithm.

5 Retexturing Graphical user interface

Besides testing the IRT and 2D to 3D garment matching on human subjects, a specialised software was created (as a requirement in the development project) for robot manikins used in a foto-boot setup. By using Kinect for image capturing, it is possible to gather data for retexturing purposes. As the setup is fixed, it is possible to automate certain parts of data capturing, namely, the segmentation part. The manikins can use green or blue covers or specialised reflective material. Firstly the manikin is segmented based on depth information, afterwards the green and blue covers can be removed based on color thresholds. The reflective material is a special case, because Kinect cannot estimate depth for reflective surfaces and is show in depth image as missing data, therefore only depth segmentation is needed, see Fig. 5.1.

(a) Reflective cover (b) Infrared image for reflective cover (c) Blue cover (d) Green cover

Figure 5.1: Manikin inside a foto-boot

5.1 GUI

To setup parameters in the software shown in Fig. 5.2, firstly the user must select segmentation method and the type of cover used. Afterwards the user can either choose to perform IRT retexturing or garment shape matching. For IRT retexturing only a texture image must be provided, it is advised to use high resolution images with a recommended aspect ratio of 16:9. Shape matching relies on the provided shirt image and its mask (segmented binary image), this software does not perform automatic segmentation for the provided 2D input images, therefore the user must make them beforehand. The output from the provided options is a retextured image showing the mannequin garment with a different texture or the chosen 2D shirt image. After retexturing, image augmentation tools can be used to modify the retextured area. The provided modifications are: shading, warping and adding lines. The warping and line location is determined form depth image where high changes in depth correspond to folds in the garment or other noticeable depth changes. The user interface is separated in three main parts: input, tools and output.

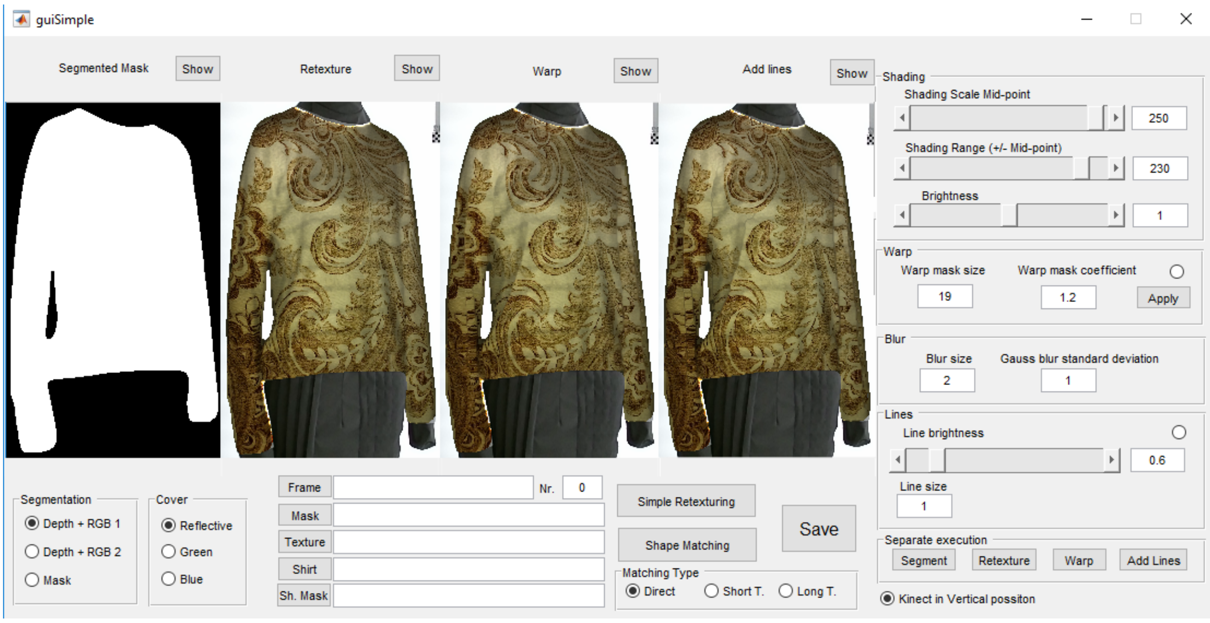


Figure 5.2: GUI for IRT and 2D to 3D garment matching

Retexturing inputs:

1. Frame - folder path with Kinect capture files and frame number
2. Mask - use provided mask, if Mask option in Segmentation is selected
3. Texture - image with the texture (RGB image, aspect ratio of 16:9 or similar)
4. Shirt - image with the shirt (RGB image)
5. Sh. Mask - image of shirt area (binary image).

Image post processing tools:

1. In Shading section, the user can change the shading effects of the retextured image. Shading range parameters control the high and low values which are defined as the α and β in IRT method. Additionally the brightens of the image can be changed by brightens slider.
2. Warp section preforms texture wrapping based on lines and folds found in the depth image. Mask size defines the size of affected texture area (must be odd number). The warp coefficient determines how much of the texture is lost. To use the warp feature the radio button must be enabled. Images are refreshed by pressing the apply button.
3. The obtained results can be smoothed out by using the blur section. Blur location is set by the fold lines. Therefore, by changing the mask size the user affects the blur size around these lines. Standard deviation determines how strong is the blur effect.

4. The wrapped areas can be made even more noticeable by enabling the line tools. Line size determines the line thickness and the brightness coefficient can be used to make the lines bright or dark.

Outputs:

1. Segmentation - shows the segmentation results
2. Retextuing - displays the retextured image with chosen shading parameters
3. Wrap - warping effects onto the detected folds
4. Add lines - added lines onto the detected folds
5. Upon pressing the ”Save” button, all displayed results are saved as images.

The GUI is a use full tool for exploring how the given parameters change the retexturing results.

5.2 Texture wrapping

The texture wrapping was introduced to create more realistic texture in garment areas with folds, which are found suing Local Phase Quantization (LPQ) [70], which is blur insensitive local texture classifier. By providing depth image, formed as a gray scale image, the LPQ can find low frequency components, which corresponds to folds in the garment. The output from this step is a mask with lines representing the fold location. Afterwards the texture is wrapped around the fold to create a visible discontinuity. The lines themselves can also be made darker and smoothed to make the results more appealing. In Fig. 5.3 it is possible to see the effects of texture wrapping, the subtle changes can be noticed in the armpit regions and larger folds. The user can choose any combination of the effects, for example only add lines and blur them.

(a) Retextured (b) Wrapped (c) Wrapped + Lines

Figure 5.3: Output example

5.3 Results

The created software was made for testing purposes and is not optimized for speed. On Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz (8 CPUs) laptop with 16GB of RAM, the retexturing process can take from 2 to 20 minutes. Nevertheless, the software can handle IRT and 2D to 3D garment matching. If long sleeve garments are used for shape matching, the hands of manikin (or person) should not occlude the garment. Fig. 5.4 show examples of production ready retextured images which can be used for creating catalogs, the same goes for shape matching results in Fig. 5.5.

Figure 5.4: IRT output examples

Figure 5.5: 2D to 3D garment matching output examples

6 3D model retexturing

6.1 Introduction

As part of the project, a method was developed for creating 3D garments with Kinect 2. The 3D models are needed for their potential in reducing the necessary positions the robot manikin needs to take. To make an image database for one garment showing how a specific size of a garments looks on different body shapes, the robot manikin needs to take around 200 shapes. If the same amount of shapes could be created by interpolating 10 (or less) 3D models, it would speed up the process and reduce the manual work.

The general process of creating a model is as follows:

1. capture a sequence with Kinect,
2. garment segmentation,
3. align the depth frames using ICP,
4. correct errors using loop closure,
5. denoise the point cloud,
6. create a mesh from the point cloud.

The sequence capturing part is done in a special foto-boot setup, where a manikin is placed on a turntable and dressed with the desired garment. Afterwards one turn of the turntable is filmed by Kinect 2 device. Usually it amounts to 150 to 300 frames. To extract only the garment region, the same depth/color threshold segmentation method as in retexturing GUI is used. All the frames are aligned with ICP, first frame in the sequence is considered as the reference frame. Loop closure performs an important part in correcting the alignment errors. After creating a dense point cloud of the garment, various techniques are used to reduce the noise. Detailed information about model creation can be found in L.Valgma bachelors thesis [6].

Additionally, to avoid recreating the 3D garment database for the same type of garments whose only difference is color or texture, a method for retexturing must be created to save time and make the 3D based size database more versatile.

6.2 Problems in 3D models creation using Kinect

The point alignment using ICP is not absolutely precise, therefore a point shifts exists within the created dense point clouds. After undergoing denoising, a random mesh is created from the remaining points. Each point also contains color information, but due to already mentioned inaccuracies in the aligned frames, the color information is also affected. Colour patterns appear blurry or shifted, affecting the visual appearance of the point cloud, which is also visible after the points are converted to a random mesh.

(a) Potential texture (b) Current texture

Figure 6.1: Example of potential and current texture quality

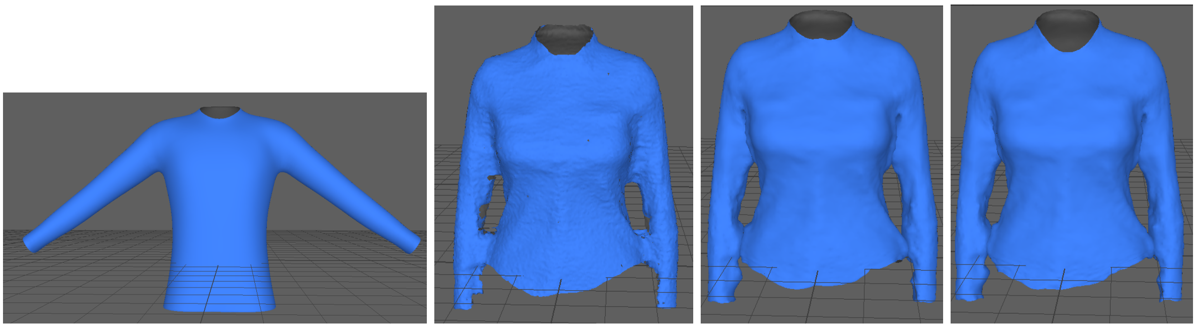
The second problem stems from the unstructured mesh of the 3D model. There are automatic methods available for generating a structured mesh, but the task of setting up the seams which are used in creating the texture map for the 3D model is a task usually done by the designer. To reduce manual work, an automatic method for texture creation also must be developed, where the desired outcome is a low resolution, structured 3D model with high quality texture.

6.3 Method

6.3.1 3D model wrapping

The Wrap3 [71] software has the ability to wrap a base model to a target model. It is done by minimizing the difference between shapes. The precision of the wrapped model can be increased by setting corresponding points in both models. The inputs for this software are .obj files, therefore the point clouds are converted to mesh using Meshlab software. The resolution of the wrapped model is determined by the polygon count in the base mesh. An example of a wrapped model can be seen in Fig. 6.2. It is possible to guide the wrapping process by setting corresponding reference points in the base and target model, but to reduce manual input, only wrapped models without reference points were used.

With the help of the base model it is possible to fill holes and create a separation between the



(a) Base model (b) Scanned garment (c) Wrapped model with reference points (d) Wrapped model without reference pts.

Figure 6.2: Base mesh wrapped to match scanned model

sleeves and the body. In addition the wrapped model preserves the seams and UV texture map from the base model.

6.3.2 Retexturing

After the base model is wrapped to match the shape of the unstructured mesh, the base mesh has the same shape as the unstructured mesh, defined mesh seams and a texture map. The next step is to recreate the original garment texture. This can be done by using the same sequence that was used to create the unstructured mesh. By aligning the frames to the correct view of the 3D model it is possible to cast the color points from the Kinect frame to the polygons in the 3D model and find the corresponding UV coordinates for the color points. To accomplish this, following method was developed:

1. load a 3D model
2. convert it to points
3. load Kincet frame of the same scan
4. find a match between the 3D model and a Kinect frame using ICP
5. transfer the texture from the RGB image to the texture map
6. repeat steps 3 to 5 till all frames are precessed

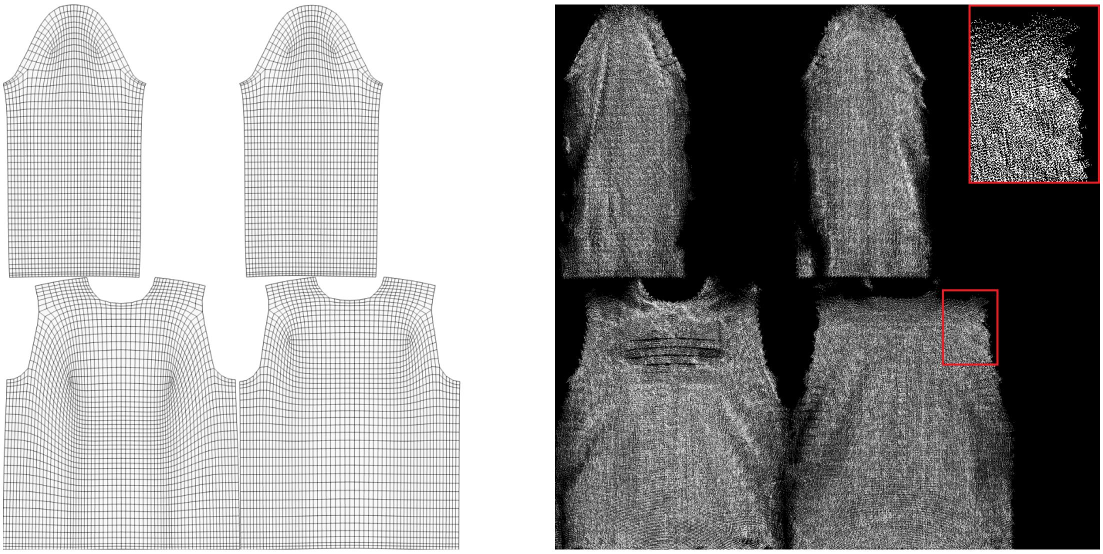
Step 5 is done as follows:

1. Cast a rays form Kinect frame points to the 3D shape
2. Find where the rays intersect the polygons in the 3D shape
3. Find the intersection point location in the texture map
4. Transfer the RGB value from points in the Kinect frame to the texture map

The output of the described method generates UV texture for each frame, in addition, a hit map and a map showing the ray-polygon surface angle is also generated. The final texture is made by updating the texture points based on the ray-polygon surface angle. If the current texture contains a point with ray reflection angle of 60 degrees and the next texture has the same point with reflection of 30 degrees, the color in the final texture will be updated with the second point. This is the process of obtaining a single texture.

6.4 Results

At the moment one frame calculation takes around 5 to 10 minutes on a laptop (Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz (8 CPUs) laptop with 16GB of RAM). The aim of the proposed method was to create higher resolution texture in comparison to the current garment model texture quality. The texture is bound by the polygon layout Fig. 6.3.a. The casted ray hit map is shown in Fig. 6.3.b, as one can see, the obtained points do not fully cover the texture area. Also texture points are missing in inner part of the sleeves, the same goes for the top part of the garment.



(a) Polygon layout

(b) Pixels found within the polygons

Figure 6.3: UV texture

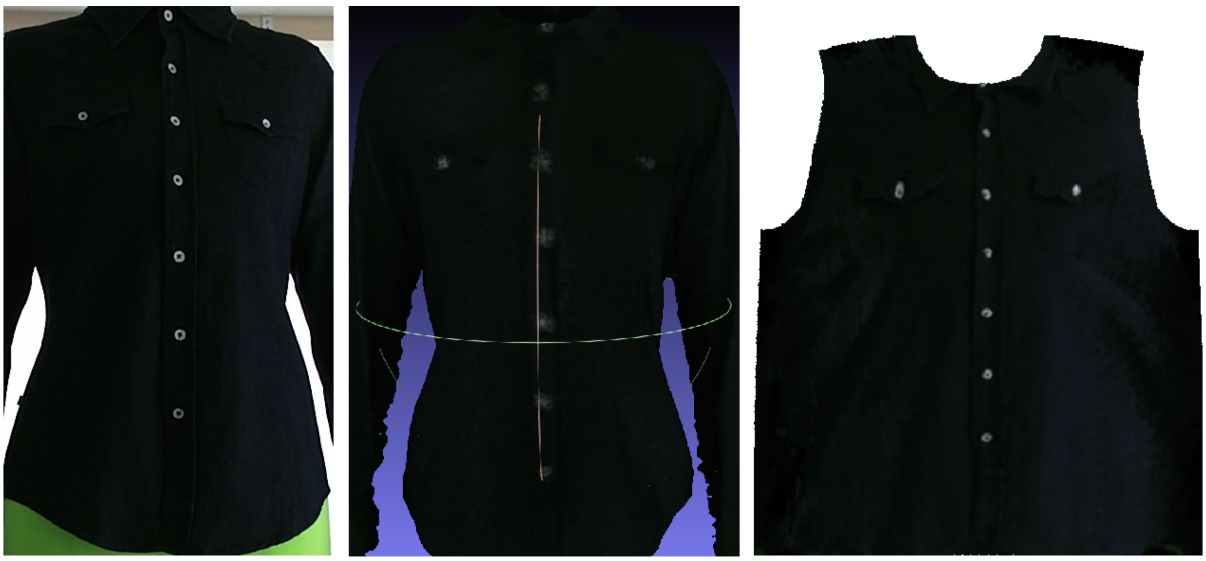
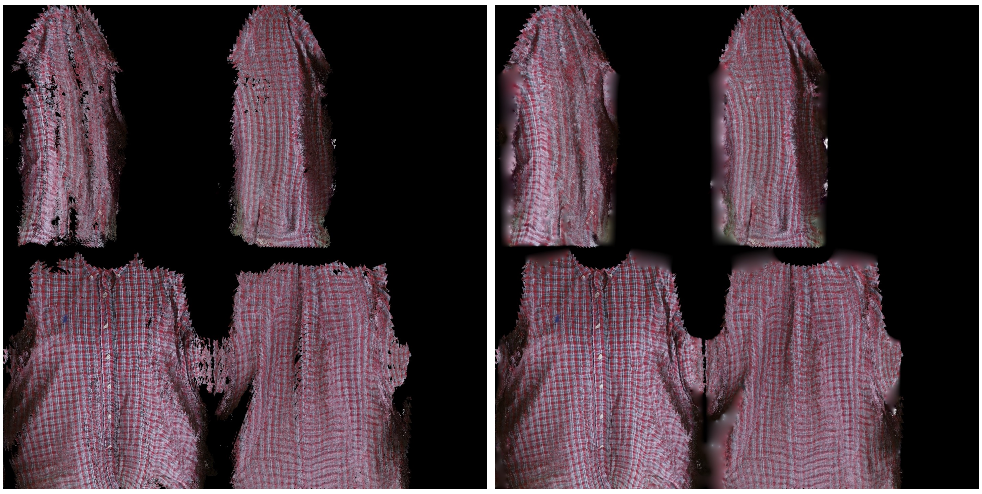


Figure 6.4: Texture quality comparison. Left: FHD image from Kinect frame, middle: current reconstructed 3D model texture quality, right: the new 3D model retexturing approach

In Fig. 6.4 the quality difference is noticeable, especially the buttons. The retexturing is based on matching the depth image of one Kinect frame with the corresponding 3D garment view, if there are errors in the alignment process or in the Kinect frame itself, they will be seen in the texture. As the method only relies on depth information, it can be considered as texture invariant.

A different garment was scanned to see how the method can handle patterns on the garment and demonstrate the current level of precision for the new model retexturing method. Fig. 6.5.a is the texture which is obtained by combining all texture images from all Kinect frames. Also it can be seen that some parts are missing in the texture, this is fixed by interpolating the color to fill the missing regions Fig. 6.5.b. The final result comparison can be seen in 6.6. The wrapped 3D model has more precise texture and separation in the arm region, some mistakes in the armpit region can be seen as well, this is because Kinect cannot see these parts while capturing the sequence.



(a) Texture (b) interpolated texture

Figure 6.5: Accumulated texture map

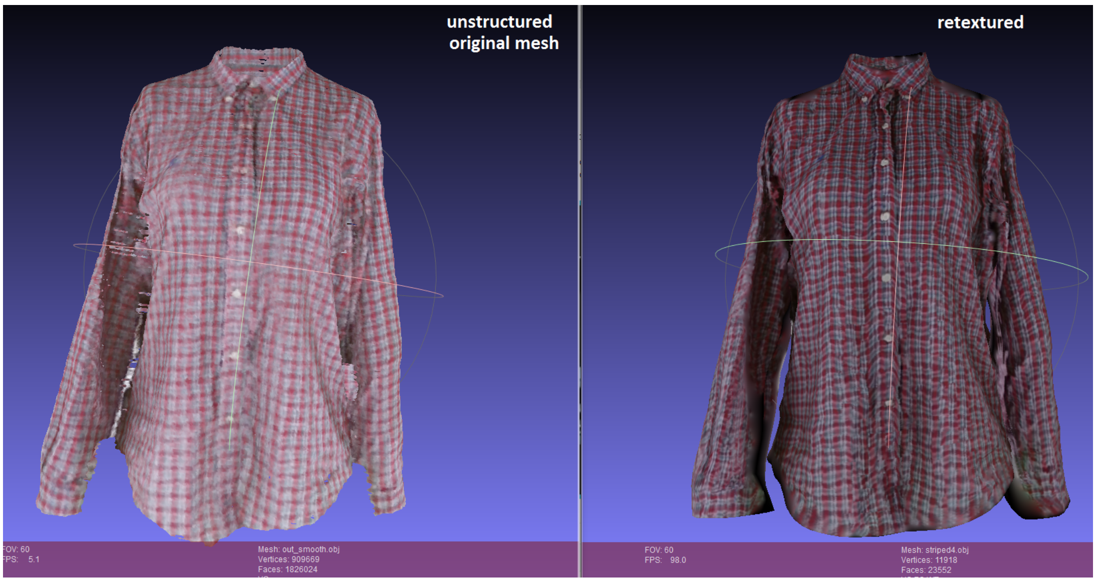


Figure 6.6: Final results

7 Conclusion

This thesis covers two retexturing methods which were implemented in a GUI. The first method, called IRT, can retexture a surface with a plane texture and shade it according to Kienct 2 IR sensor data. The details of the surface are limited by Kinect resolution, but it is possible to interpolate the surface and use higher resolution textures than the original Kinect frame resolution. The second retexturing method, which is based on 2D to 3D geodesic based garment matching, can retexture a surface by matching it to a shape within a given texture. By process of shape matching, a correspondence between the outer contours is found and the given texture is wrapped and superimposed on the surface. In the current implementation, additional effects can be added within the GUI. To create a distinction between folds on the garment, it is possible to add lines, blurred lines and texture wrapping towards the lines.

These methods serve as proof of concept and demonstrate successful results, in further stages more specific problems, especially handling self-occlusion will be addressed. The first method should have the ability to detect self-occlusion and select textures in a way, that shows clear distinction between the occluded parts. The second method suffers from a similar problem, in the next iteration, a method for detecting self-occlusion must be developed. Before shape matching stage, the system should be able to correctly detect hand position of a human or manikin and draw the contour for the occluded parts. This would allow to find correct transformation from a target image to the occluded surfaces.

The third part of the thesis discusses retexturing 3D models using the data which was used for creating it. The developed system can transfer texture from Kinect RGB-D frames to a texture map of a structured 3D model, by aligning the frames with the 3D model and using ray casting to finding the projected point location in polygons of the 3D model. This method had to be developed due to lack of texture quality in the available 3D models which use low resolution images for assigning color to aligned point cloud, which is later turned into a 3D model. The developed method can be considered as an automatic texture mapping method. For the method to work, the same Kinect frames, that were used in creating the 3D model, must be used for retexturing. Future work will focus on increasing the alignment precision during the creation of the 3D models and reducing misalignment in the transferred texture.

Acknowledgements

I would like to thank my supervisors Gholamreza Anbarjafari and Sergio Escalera for guiding and supporting me throughout the thesis process.

Special thanks to Meysam Madadi from University of Barcelona for helping with method development and members from University of Tartu Intelligent Computer Vision reaserch group.

References

[1] Image of kinect v2.0. http://assets2.ignimgs.com/2014/08/01/xbox-one-kinectjpg-8ddb5e_1280w.jpg, 12.05.2017, 10:10 (UTC).

[2] Egils Avots, Morteza Daneshmand, Andres Traumann, Sergio Escalera, and Gholamreza Anbarjafari. Automatic garment retexturing based on infrared information. *Computers & Graphics*, 59:28–38, 2016.

[3] Jianbing Shen, Hanqiu Sun, Xiaoyang Mao, Yanwen Guo, and Xiaogang Jin. Color-mood-aware clothing re-texturing. In *Computer-Aided Design and Computer Graphics, 2011 12th International Conference on*, pages 151–154. IEEE, 2011.

[4] Erum Arif Khan, Erik Reinhard, Roland W Fleming, and Heinrich H Bülthoff. Image-based material editing. *ACM Transactions on Graphics (TOG)*, 25(3):654–663, 2006.

[5] Oliver Wasenmuller and Didier Stricker. Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In *Chen CS., Lu J., Ma KK. (eds) Computer Vision ACCV 2016 Workshops. Lecture Notes in Computer Science*, volume 10117. Springer, 2016.

[6] Lembit Valgma. 3d reconstruction using kinect v2 camera. In *Bachelors thesis*. University of Tartu, 2016.

[7] Zhang L. Dong H. Alelaiwi A. El Saddik A. Yang, L. Evaluating and improving the depth accuracy of kinect for windows v2. In *IEEE Sensors Journal (Volume: 15, Issue: 8)*, pages 4275–4285. IEEE, 2015.

[8] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[9] Pedro V Sander, John Snyder, Steven J Gortler, and Hugues Hoppe. Texture mapping progressive meshes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 409–416. ACM, 2001.

[10] Eric A Bier and Ken R Sloan. Two-part texture mappings. *Computer Graphics and Applications, IEEE*, 6(9):40–53, 1986.

[11] Pat Hanrahan and Paul Haeberli. Direct WYSIWYG painting and texturing on 3D shapes. In *ACM SIGGRAPH Computer Graphics*, volume 24, pages 215–223. ACM, 1990.

[12] Yongjin Kim, Holger Winnemoller, and Seungyong Lee. Wysiwyg stereo paintingwith usability enhancements. *Visualization and Computer Graphics, IEEE Transactions on*, 20(7):957–969, 2014.

[13] Songgang Xu and John Keyser. Texture mapping for 3d painting using geodesic distance. In *Proceedings of the 18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 163–163. ACM, 2014.

[14] A Traumann, M Daneshmand, S Escalera, and G Anbarjafari. Accurate 3d measurement using optical depth information. *Electronics Letters*, 51(18):1420–1422, 2015.

[15] Kai Lawonn, Rocco Gasteiger, Christian Rössl, and Bernhard Preim. Adaptive and robust curve smoothing on surface meshes. *Computers & Graphics*, 40:22–35, 2014.

[16] Andres Traumann, Gholamreza Anbarjafari, and Sergio Escalera. A new retexturing method for virtual fitting room using kinect 2 camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 75–79, 2015.

[17] Donghyun Kim, Younsam Chae, Jonghun Shin, Uyeol Baek, and Seoksoo Kim. Interactive mirror system based on personal purchase information. *International Journal of Software Engineering & Its Applications*, 7(3), 2013.

[18] Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani. Virtual garments: A fully geometric approach for clothing design. In *Computer Graphics Forum*, volume 25, pages 625–634. Wiley Online Library, 2006.

[19] Remi Brouet, Alla Sheffer, Laurence Boissieux, and Marie-Paule Cani. Design Preserving Garment Transfer. *ACM Transactions on Graphics (TOG)*, 31(4):36, 2012.

[20] Ryan White and David Forsyth. Retexturing single views using texture and shading. In *Computer Vision–ECCV 2006*, pages 70–81. Springer, 2006.

[21] Emmanuel Turquin, Marie-Paule Cani, and John F Hughes. Sketching garments for virtual characters. In *ACM SIGGRAPH 2007 courses*, page 28. ACM, 2007.

[22] Zahraa Yasseen, Ahmad Nasri, Wajih Boukaram, Pascal Volino, Nadia Magnenat-Thalmann, et al. Sketch-based garment design with quad meshes. *Computer-Aided Design*, 2013.

[23] Mingmin Zhang, Ling Lin, Zhigeng Pan, and Nan Xiang. Topology-independent 3d garment fitting for virtual clothing. *Multimedia Tools and Applications*, pages 1–17, 2013.

[24] Julien Pilet, Vincent Lepetit, and Pascal Fua. Retexturing in the presence of complex illumination and occlusions. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–8. IEEE Computer Society, 2007.

[25] Yanwen Guo, Jin Wang, Xiang Zeng, Zhongyi Xie, Hanqiu Sun, and Qunsheng Peng. Image and video retexturing. *Computer Animation and Virtual Worlds*, 16(3-4):451–461, 2005.

[26] Jianbing Shen, Xiaogang Jin, and Hanqiu Sun. High-Dynamic Range Image Tone Mapping and Retexturing Using Fast Trilateral Filtering. *The Visual Computer*, 23(9-11):641–650, 2007.

[27] Steve Zelinka, Hui Fang, Michael Garland, and John C Hart. Interactive material replacement in photographs. In *Proceedings of Graphics Interface 2005*, pages 227–232. Canadian Human-Computer Communications Society, 2005.

[28] Anna Hilsmann and Peter Eisert. Tracking and retexturing cloth for real-time virtual clothing applications. In *Computer Vision/Computer Graphics Collaboration Techniques*, pages 94–105. Springer, 2009.

[29] Adrien Bartoli, Andrew Zisserman, et al. Direct estimation of non-rigid registrations. In *British machine vision conference*, pages 899–908, 2004.

[30] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, 2003.

[31] Daniel Pizarro and Adrien Bartoli. Feature-based Deformable Surface Detection with Self-Occlusion Reasoning. *International Journal of Computer Vision*, 97(1):54–70, 2012.

[32] Christian Kerl, Mohamed Souiai, Jurgen Sturm, and Daniel Cremers. Towards Illumination-Invariant 3D Reconstruction Using ToF RGB-D Cameras. In *Second International Conference on 3D Vision (3DV)*, volume 1, pages 39–46. IEEE, 2014.

[33] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

[34] Peter Shirley, Michael Ashikhmin, and Steve Marschner. *Fundamentals of computer graphics*. CRC Press, 2009.

[35] Bing Jian and Baba C. Vemuri. Robust point set registration using gaussian mixture models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1633–1645. IEEE, 2010.

[36] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[37] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan. Scanning 3D Full Human Bodies Using Kinects. *Transactions on Visualization and Computer Graphics*, 18(4):643–650, 2012.

[38] Zhenglong Zhou, Bo Shu, Shaojie Zhuo, Xiaoming Deng, Ping Tan, and Stephen Lin. Image-based Clothes Animation for Virtual Fitting. In *SIGGRAPH Asia 2012 Technical Briefs*, page 33. ACM, 2012.

[39] Stefan Hauswiesner, Matthias Straka, and Gerhard Reitmayr. Free Viewpoint Virtual Try-on with Commodity Depth Cameras. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pages 23–30. ACM, 2011.

[40] Morteza Daneshmand, Alvo Aabloo, Cagri Ozcinar, and Gholamreza Anbarjafari. Real-time, automatic shape-changing robot adjustment and gender classification. *Signal, Image and Video Processing*, pages 1–8, 2015.

[41] Sarbartha Sengupta and Parag Chaudhuri. Virtual Garment Simulation. In *Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pages 1–4. IEEE, 2013.

[42] Yaoye Zhang, Zhengxing Sun, Kai Liu, and Yan Zhang. A Method of 3D Garment Model Generation Using Sketchy Contours. In *Sixth International Conference on Computer Graphics, Imaging and Visualization*, pages 205–210. IEEE, 2009.

[43] Jacques Harvent, Benjamin Coudrin, Ludovic Brèthes, Jean-José Orteu, and Michel Devy. Multi-view dense 3d modelling of untextured objects from a moving projector-cameras system. *Machine vision and applications*, 24(8):1645–1659, 2013.

[44] Sid Ahmed Fezza and Mohamed-Chaker Larabi. Color Calibration of Multi-View Video Plus Depth for Advanced 3D Video. *Signal, Image and Video Processing*, pages 1–15, 2015.

[45] Yuewen Ma, Jianmin Zheng, and Jian Xie. Foldover-Free Mesh Warping for Constrained Texture Mapping. *IEEE Transactions on Visualization and Computer Graphics*, 21(3):375–388, 2015.

[46] Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J Gortler. A Local/global Approach to Mesh Parameterization. In *Computer Graphics Forum*, volume 27, pages 1495–1504. Wiley Online Library, 2008.

[47] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Rnvironments. In *12th International Symposium on Experimental Robotics (ISER)*. Citeseer, 2010.

[48] Sandro Barone, Alessandro Paoli, and Armando Viviano Razionale. Three-dimensional point cloud alignment detecting fiducial markers by structured light stereo imaging. *Machine Vision and Applications*, 23(2):217–229, 2012.

[49] Hideo Saito, Shigeyuki Baba, and Takeo Kanade. Appearance-based virtual view generation from multicamera videos captured in the 3-d room. *Multimedia, IEEE Transactions on*, 5(3):303–316, 2003.

[50] Frédéric Cordier, W Lee, H Seo, and Nadia Magnenat-Thalmann. Virtual-try-on on the web. *Laval Virtual*, 2001.

[51] P. Henry, M. Krainin, E. Herbst, X. Ren, and D Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *International Journal of Robotics Research*, 31(5):647–663, 2012.

[52] Feng Zhou and Fernando De la Torre. Factorized graph matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 127–134. IEEE, 2012.

[53] Huijie Fan, Yang Cong, and Yandong Tang. Object detection based on scale-invariant partial shape matching. *Machine Vision and Applications*, 26(6):711–721, 2015.

[54] E.A. Bier and K.R. Sloan. Two-part texture mappings. *IEEE Computer Graphics and Applications*, 6(9):40–53, 1986.

[55] Pat Hanrahan and Paul Haeberli. Direct wysiwyg painting and texturing on 3d shapes. *ACM SIGGRAPH computer graphics*, 24(4):215–223, 1990.

[56] S. Xu and J. Keyser. Texture mapping for 3d painting using geodesic distance. In *18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2014.

[57] W.C. Chang and W.C. Chang. Real-time 3d rendering based on multiple cameras and point cloud. In *7th International Conference on Ubi-Media Computing and Workshops*, pages 121–126.

[58] E. Turquin, M. P. Cani, and J. F. Hughes. Sketching garments for virtual characters. In *ACM SIGGRAPH*, 2007.

[59] B. Zhou, X. Chen, Q. Fu, K. Guo, and P. Tan. Garment modeling from a single image. *Computer Graphics Forum*, 32(7):85–91, 2013.

[60] I. Eckstein, V. Surazhsky, and C. Gotsman. Texture mapping with hard constrains. *Computer Graphics Forum*, 20(3):95–104, 2001.

[61] V. Kraevoy, A. Sheffer, and C. Gotsman. Matchmaker: Constructing constrained texture maps. 22(3), 2003.

[62] G. Yanwen, Y. Pan, X. Cui, and Q. Peng. Harmonic maps based constrained texture mapping method. *Journal of Computer Aided Design and Computer Graphics*, 7:1457–1462, 2005.

[63] Lok Ming Lui, Ka Chun Lam, Tsz Wai Wong, and Xianfeng Gu. Texture map and video compression using beltrami representation. *SIAM Journal on Imaging Sciences*, 6(4):1880–1902, 2013.

[64] N.D. McKay Besl, Paul J. A method for registration of 3-d shapes. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 239–256. IEEE, 1992.

[65] Lu Chien-Ping Suguna Pappu Gold Steven, Rangarajan Anand and Mjolsness Eric. New algorithms for 2d and 3d point matching:: pose estimation and correspondence. In *Pattern Recognition (Volume:31)*, pages 1019–1031, 1998.

[66] Chui Haili and Rangarajan Anand. A new point matching algorithm for non-rigid registration. In *Computer Vision and Image Understanding - Special issue on nonrigid image registration (Volume:89 , Issue: 2-3)*, pages 114–141. Elsevier Science Inc., 2003.

[67] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:2262 – 2275, 2010.

[68] Thomas Deschamps and Laurent D. Cohen. Fast extraction of minimal paths in 3d images and applications to virtual endoscopy, 2001.

[69] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[70] Ville Ojansivu and Janne Heikkil. Blur insensitive texture classification using local phase quantization. In *ICISP*, volume 5099, pages 236–243. Springer, 2008.

[71] Russian3dscanner: R3ds. http://www.russian3dscanner.com/, 12.05.2017, 10:14 (UTC).

Non-exclusive licence to reproduce thesis and make thesis public

I, Egils Avots

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to
   (a) reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
   (b) make available to the public via the university's web environment, including via the DSpace digital archives, as of 01.01.2019 until expiry of the term of validity of the copyright,

   "Garment retexturing using Kinect V2.0"

   supervised by Assoc. Prof. Gholamreza Anbarjafari and Assoc. Prof. Sergio Escalera
2. I am aware of the fact that the author retains these rights.
3. This is to certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **17.05.2017**