3D Textile Reconstruction based on KinectFusion and Synthesized Texture

<table>
<thead>
<tr>
<th>Journal:</th>
<th>International Journal of Clothing Science and Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript ID</td>
<td>IJCST-01-2017-0007.R1</td>
</tr>
<tr>
<td>Manuscript Type:</td>
<td>Research Paper</td>
</tr>
<tr>
<td>Keywords:</td>
<td>Textile texture, 3D scanning, Convolutional neural networks, KinectFusion</td>
</tr>
</tbody>
</table>
3D Textile Reconstruction based on KinectFusion and Synthesized Texture

Abstract

The 3D textile model plays an important role in textile engineering. The 3D textile model mainly consists of the 3D geometric shape and the texture. Depth cameras such as Microsoft Kinect, is much cheaper than conventional 3D scanning devices. However, not much work about 3D textile reconstructed based on depth cameras and the texture is also limited by photography methods in 3D scanning. This paper presents a novel framework of reconstructing the 3D textile model with synthesized texture. Firstly, a pipeline of 3D textile reconstruction based on KinectFusion is proposed to obtain a better 3D model. Secondly, convolutional neural networks (CNN) is used to synthesize the textile texture. Experimental results show that our method can conveniently obtain 3D textile models with synthesized texture.

Keywords: Textile texture; 3D scanning; Convolutional neural networks; KinectFusion

1 Introduction

3D scan technology has been widely used in textile industry to capture the 3D geometric shapes of clothes and the human body. Texture, as one of the most important features for textiles, is still a challenging task in today’s textile industry. One important parameter of textile is the geometric shape, which can give human more intuitive information than that from two dimension images. With the rapid development of commercial depth cameras, e.g. Kinect, commercial systems for human body scanning based on depth cameras have been proposed [1,2]. However, not much previous work focus on high-quality 3D textile reconstruction, and the texture usually is captured by the RGB camera, which is limited by the photography method. To visualize different textures for 3D textile is very interesting and useful in textile engineering.

In today’s textile industry, the texture is mainly designed by the artist which is expensive and time-consuming. An increasing number of people, especially young people, like having their own pattern on the T-shirt which is called cultural T-shirt. Hence, there is a permanent interest in the development of rapid and automatic texture generation method for the textile. However, the human is extremely sensitive to the texture of textiles. It is not trivial that a realistic 3D textured textile model should be obtained.
This paper presents a novel framework for high-quality 3D textile reconstruction with synthesized texture. Our framework is shown in Fig. 1. An efficient pipeline is designed for 3D textile reconstruction based on KinectFusion. The convolution neural networks (CNN) is used to build a texture model to synthesize a new realistic textile texture.
Figure 1. The framework of 3D high-quality reconstruction with synthesized texture. A better 3D
textile model is obtained based KinectFusion using our method (left). The texture is generated using the convolutional neural network (right). An initial noise image $x$ is passed through the CNN and a loss function $E_i$ is computed on each layer of the texture model. $L$ is a weighted sum of the contributions $E_i$ from each layer. A new textile texture is found by producing the same Gram matrices $G'$ as the original texture.

2. Related Work

2.1 Kinect V2 and KinectFusion

As the recent improvement in and the declining costs of scanning technology, commercial depth cameras are becoming widely amiable. Many researchers have presented 3D scanning systems based on depth cameras \cite{3,4}. The Kinect camera, originally designed for natural interaction in computer gaming environments \cite{1}, is one of commercial depth cameras. Even though Kinect v2 relies on a different technology than Kinect v1, it also allows the acquisition of three different output streams. It is composed of two cameras, namely an RGB and an infrared (IR) cameras, which is shown in Fig.2.

![Figure 2 Kinect v2 camera](image)

The RGB camera captures color information with a resolution of 1920×1080 pixels, while the IR camera captures depth information with a 512×424 pixels resolution. The whole acquisitions can be carried out with a framerate up to 30 Hz. The last feature to be mentioned is the field of view for depth sensing of 60 degrees vertically and 70.6 degrees horizontally.

T Butkiewicz tested the depth accuracy for Kinect v2, from 0.5m to ~1.6m, the standard deviations of the depth measurements was consistently small (<1.5mm) and intriguingly randomly changing with respect to distance [°]. After ~1.6m, the standard deviation increases linearly with the maximum working distance of 4.5m.

KinectFusion is a single-Kinect 3D scanning system. It enables a user holding and moving a Kinect camera to rapidly create a detailed 3D model. Only the depth data from Kinect is used to track the 3D pose of the sensor and reconstruct 3D model in real-time [°]. Macedo et al. utilized KinectFusion to
track the real-time face[8].

![Figure 3. Standard deviations of depth measurements of an optimal flat surface across the operating range of the sensor (From T Butkiewicz)](image)

2.2 3D Mesh Processing

3D mesh processing is a hot area, which has been researched for many years. 3D mesh processing includes various kinds of sub-areas, e.g. Denoising, surface smooth and so on. Many excellent libraries are presented [9, 10] to deal with 3D mesh processing. However, these methods are designed for generic 3D meshes, which do not work very well for 3D textiles.

3D scanning extracts surface samples in the form of range maps or point clouds. The common way to convert them into discrete surfaces is to compute a volumetric distances field [11] and extract the zero-isosurface triangle mesh using Marching Cubes [12]. The quality of the resulting triangulation is typically low, but the algorithm is fast. A pipeline is presented in this paper to obtain high-quality 3D textile models.

2.3 Texture Synthesis

The aim of texture synthesis is to define a generic process that, from a given example texture, can generate arbitrarily many new samples of the same texture. The evaluation criterion for the quality of the synthesized texture is usually human inspection, which means that textures are perfectly synthesized if a human observer cannot tell the original texture from a synthesized one.

In general, existing texture synthesis algorithms can be broadly categorized into two main methods, parametric and non-parametric methods. Parametric approaches are to define a parametric texture model which usually consists of a set of statistical measurements. In the model, a texture is uniquely defined by the outcome of those measurements. Therefore new texture can be generated by finding an image that has the same measurement outcomes as the original texture. The best parametric texture model is proposed by Portilla and Simoncelli [13]. Non-parametric approaches are to resample either pixels or whole patches of the original texture without defining model [14]. This kind of methods can only randomize the original texture but not change its perceptual properties. Gatys et al. demonstrated that deep networks can generate beautiful textures and stylized images from a single texture reference[15]. Ulyanov et al. train compact feed-forward convolutional networks to generate multiple samples of the same texture of the arbitrary size and to transfer artistic style from a given texture to any other texture[16]. Lin et al. conducts a systematic evaluation of recent CNN-based texture descriptors for recognition and attempts to understand the nature of invariances captured by these representations[17].
However, these work only focus on generating a new image from a given image. In our work, we extended its application to the 3D textile reconstruction via applying a parametric method to synthesize the textile texture.

2.4 Convolutional Neural Networks (CNN)

Convolution neural networks date back decades [18] and deep CNNs have recently shown an explosive popularity partially due to its success in image classification [19]. They have also been successfully applied to other computer vision fields, such as object detection [20], face recognition [21], pedestrian detection [22], and image super resolution [23]. As modern powerful GPUs highly speed up the training process and the propose of the rectified linear unit (ReLU) [24] makes convergence much faster while still presents good quality, CNN makes great progress. In this paper, we also utilize the property of feature extracting of CNN to implement the fabric texture synthesis.

3. Overview of the Framework

Our goal is to develop a framework of reconstructing high-quality 3D textile models with synthesized textures, as shown in Fig. 1. We start by capturing the 3D textile model using KinectFusion with the help of a turntable (Section 4.1). Then a pipeline is designed for 3D textile mesh processing to increase the quality of textile shape (Section 4.2). We use the feature space provided by CNN to build the parametric texture model (Section 4.3 and Section 4.4), and then a new textile texture will be synthesized from the original texture (Section 4.5). The rest of this paper is structured as follows: Section 5 shows the results; Section 6 is the discussion and future work.

4. Methodology

4.1 3D Scan

KinectFusion is a popular single-Kinect 3D scanning system, which can be used for Kinect V1 and Kinect V2 [25]. Kinect V2 is chosen in our work as it has a higher depth resolution. For KinectFusion, the position and orientation of the moving Kinect are estimated by consistently align 2 adjacent frames of depth data captured by the Kinect via ICP algorithm. So when the Kinect is moved fast, insufficient overlaps between 2 adjacent frames of depth data will result in collapsing during scanning large objects, like the human body. In practice, we find that moving the Kinect at a constantly slow speed will obtain a better reconstructed 3D model. Therefore, a turntable is used as the platform, which the object is located on, to increase the accuracy and robustness of KinectFusion. It costs 30 seconds to scan the whole object while the turntable rotates a round. Fig. 4 shows our 3D scanning system, the Kinect is set about 1m away from the scanning object and the height of the Kinect can be adjusted freely to meet various scanning requirements.
4.2 3D Mesh Processing

With the help of the turntable, 3D textiles can be easily obtained. Without loss of generality, we take reconstructing trousers model as an example to further explain our method. It is useless to scan a flat garment laid on the ground. Therefore, a human body dressing trousers are scanned and the trousers model is extracted next. The 3D model from KinectFusion is shown in Fig.5 (a). It can be seen that the 3D model is not perfect. It needs to be further processed, which is called 3D mesh processing. In our pipeline, as shown in Fig.1 (a), the noises and outliers should be removed firstly. Bilateral Mesh Denoising \[26\] is used in this step. As KinectFusion merges too many frames of depth images, redundant data cannot be avoided. Too large data is also not conveniently used in practice. Secondly, 3D model is subsampled. Poisson-disk sample \[27\] is used in this step. The occlusion is a generic issue in almost all 3D scanning work, especially when we scan garments and bags in our work. Thirdly, a hole-filling algorithm \[28\] is applied to get complete 3D textiles. Next, the surface of 3D textile is smoothed \[29\]. Universally, a pretty good 3D model can be obtained through these four steps, as shown in Fig.5 (b). However, due to features of textiles, these processes are not sufficient. Here, we just list part of textile specificities:

- **Diversity**: Containing various kinds of objects, e.g. garments, bags, cloths and so on. So textiles have many different topologies;
- **Non-rigidity**: Textiles are non-rigid, and once the geometry of the textile is changed by extern force it is impossible to recover its original shape;
- **Utility**: Textile products are designed for meeting kinds of needs, like keeping warm, protecting and so on. So the 3D model of textile in usage is needed.

Majorities of textile products are scanned together with other accessories due to above features, e.g. the garment is dressed on a body for 3D scanning; a piece of cloth is laid on a platform to drape for 3D scanning. Therefore, differing from generic 3D mesh processing, the trousers extraction from the human body model is very important. As the data structure of the 3D model is triangle soup, zigzag edges along the cutting line will happen. As shown in Fig.6 (a), the waist of trousers and leg openings have bad shape. Our solution to this issue is to do re-topology for the trousers model.

Meshing algorithms can be classified into local and global methods. The former are usually simple,
robust and scalable. Global algorithms solve optimization problems whose size depends on the entire dataset. In our work, a re-topology method called Instant Field-aligned Meshing [30] is chosen. It computes a mesh that is globally aligned with a direction field using local orientation and position-field smooth operators. The mesh is then extracted from the fields and optionally post-processed. Comparing to other re-topology methods, Instant Field-aligned Meshing has following advantages:

- This method is simple to implement and parallelize, and it can process a variety of input surface representations, such as point clouds, range scans and triangle meshes;
- This method can process extremely large meshes and point clouds with sizes exceeding several hundred million elements, as it avoids any global optimization;
- This method is interactive.

Due to these features, Instant Field-aligned Meshing algorithm is used to implement re-topology in our work. The final trousers model using our method is shown in Fig.6 (b). It can be seen that the waist of trousers and leg openings are perfect.

4.3 Convolution Neural Network

The convolutional neural network has proven to be excellent in feature extracting. In our work, we apply the VGG-19 network, which is trained on objection recognition [31]. Here we give only a brief summary of its architecture.

There are 16 convolutional and 5 pooling layers in the VGG-19 network. We just use parts of the full layers. The network’s architecture is based on two fundamental computations:

1. The convolution is linearly rectified with filters of the size $3 \times 3 \times k$ where $k$ is the number of output per layer. The strider and padding are both set to one so that the output image has the same spatial dimensions as the input image, which satisfied the equation 1.

   Equation 1 describes the relationship among the hyper-parameters, where $W$ denotes the dimension of the input image, $F$ is the dimension of filter, $P$ represents the padding value, $S$ means the stride, and $n$ denotes the dimension of the output image.

2. Pooling layer is maximum pooling in non-overlapping $2 \times 2$ regions, which down-samples the feature maps by a factor of two. This is used for reducing the information to make the training more efficient.

   $$n = \frac{W - F + 2P}{S} + 1$$ (1)

The overall architecture of VGG-19 is shown in Fig.1 (b). Convolutional layers and max-pooling layers are connected in an alternating manner. As we use only the convolution layers, the input images can be arbitrarily large without considering the input image dimension change. The first convolution layer has the same size as the image and for the following layers, the ratio between the feature map sizes remains fixed. We usually believe that each convolutional layer defines a non-linear filter, and that is why this kind of neural network is called convolutional neural network.

The trained convolutional network is publicly available and supported by the Caffe-framework [32]. For texture generation, L Gatys et al. reports that using average pooling will improve the gradient flow and slightly cleaner results can be obtained [33]. Hence, the images shown blow were generated with average pooling. Finally, for practical reasons, the weights in the network were rescaled such that the mean activation of each filter over images and positions is equal to one.

4.4 Textile Texture Model

Similar to the texture model proposed by Portilla and Simoncelli [13], we define the texture model. The
main difference to their work is that instead of using linear filter bank and a set of carefully chosen summary statistics, we use feature space provided by the convolutional neural network and only one spatial summary statistic: the correlations between feature responses in each layer of the network. We first extract features of different sizes homogeneously from the original image. Then we compute a spatial summary statistic on the feature responses to get a stationary representation of the original image, shown in Fig.1 (b). Finally we synthesize a new image with the same stationary representation by performing gradient descent on a random image which has been initialized with white noise, as shown in Fig.1 (b).

The original texture is denoted as \( x \) in our model, we first pass \( x \) through the convolutional neural network and compute the activations for each layer \( l \) in the network. A layer with \( N_l \) distinct filters has \( N_l \) feature maps each of size \( M_l \) when vectorized. We reorganize these features in a matrix \( F^l \in \mathbb{R}^{N_l \times M_l} \), where \( F^l_{jk} \) is the activation of the \( j^{th} \) filter at position \( k \) in layer \( l \).

Textures are per definition stationary, so a texture model needs to be agnostic to spatial information. A summary statistic that discards the spatial information in the feature maps is given by the correlations between the responses of different features. These feature correlations are given by the Gram matrix \( G^l \in \mathbb{R}^{N_l \times N_l} \), where \( G^l_{ij} \) is the inner product between feature map \( i \) and \( j \) in layer \( l \):

\[
G^l_{ij} = \sum_k F^l_{jk} F^l_{ik} \tag{2}
\]

A set of Gram matrices \( \{G^1, G^2, ..., G^L\} \) from the \( L \) layers in the network responding a given texture provides a stationary representation of the texture, which fully specifies a texture in this model.

4.5 Textile Texture Generation

As we described above, we use gradient descent from an initializing white noise image to find a synthesized image that matches the Gram-matrix representation of the original image. This optimization is implemented by minimizing the mean-squared distance between the entries of the Gram matrix of the original image and the Gram matrix of the image being generated, as shown in Fig.1 (b).

Let \( x \) and \( x' \) be the original image and the generated image, then \( G^l \) and \( G^l' \) denote their respective Gram-matrix representations in layer \( l \) (Eq.2). The contribution of layer \( l \) to the total loss is then

\[
E_l = \frac{1}{4N_l^2M_l^2} \sum_{i,j}(G^l_{ij} - G'^l_{ij})^2 \tag{3}
\]

and the total loss is

\[
L(x, x') = \sum_{l=0}^{L} \omega_l E_l \tag{4}
\]
Where \( \omega_l \) are weights of the contribution of each layer to the total loss. The derivative of \( E_l \) with respect to the activations in layer \( l \) can be computed analytically:

\[
\frac{\partial E_l}{\partial F_{ij}^l} = f'(x) = \begin{cases} 
0 & \text{if } F_{ij}^l < 0 \\
1 & \text{if } F_{ij}^l > 0 \\
\frac{1}{N^2 M^2} \left( F_{ij}^l (G_{ij} - G_{ij}^l) \right) & \text{if } F_{ij}^l = 0
\end{cases}
\]  \( (5) \)

The standard error back-propagation \([34]\) can be used to compute the gradients of \( E_l \) with respect to the pixels \( x_i \). In our practice, we use L-BFGS \([35]\), which can work well for the high-dimensional optimization problem. The whole procedure relies mainly on the standard forward-backward pass used to train the convolutional network. With the help of GPUs and performance-optimized toolboxes for training convolutional neural network, textile texture generation can be done in reasonable time.

5. Results

To evaluate the proposed method, we first scan some textile objects to obtain their 3D models. A 3D textile reconstruction pipeline based on KinectFusion is proposed to obtain a better 3D textile model with the smooth edge. Fig.5 demonstrates the results of KinectFusion. We compared our method to KinectFusion. As shown in Fig.6, the extracted trousers using KinectFusion has a bad zigzag edge along the cutting line while our method can obtain a smooth edge. As shown in Table.1, the robustness and scanning speed of KinectFusion is improved with the help of a turntable when scanning large objects such as the bag and the garment.

![Figure 5 (a) 3D model from KinectFusion](image_url)  ![Figure 5 (b) 3D model after generic processing](image_url)  ![Figure 5. Scanning trousers](image_url)
Figure 6 (a). Trousers using KinectFusion

Figure 6 (b). Trousers using our method

Figure 6. Comparing our method to KinectFusion

Table 1. Comparing our method to KinectFusion

<table>
<thead>
<tr>
<th></th>
<th>Fabric</th>
<th>Bag</th>
<th>Garment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanning time</td>
<td>Kinect Fusion</td>
<td>Our method</td>
<td>Kinect Fusion</td>
</tr>
<tr>
<td></td>
<td>20s</td>
<td>30s</td>
<td>50s</td>
</tr>
<tr>
<td></td>
<td>Our method</td>
<td>30s</td>
<td>30s</td>
</tr>
<tr>
<td>Robustness</td>
<td>Kinect Fusion</td>
<td>Our method</td>
<td>Kinect Fusion</td>
</tr>
<tr>
<td></td>
<td>good</td>
<td>medium</td>
<td>bad</td>
</tr>
<tr>
<td></td>
<td>Our method</td>
<td>good</td>
<td>good</td>
</tr>
</tbody>
</table>

The texture from KinectFusion is shown in Fig.7, it is seen that some areas of the texture are blurry and noisy. In addition, the conventional 3D scanning only captures the real texture of the textile. However,
these textures have to be designed previously by the artist. In this paper, the texture is synthesized in order to automatically generate the new texture. In our experiments, the original textures come from Texture website \cite{36}. These images consist of 9 classes of textile texture, e.g. camouflage and leather textures. Fig.8 illustrates the textured 3D model using our method.

![Textured 3D model from KinectFusion](image)

We further show the textile textures generated by our method using a different number of layers that is used to constrain the gradient descent, as shown in Fig.9. It means that the images in the first row were generated only from the texture representation of the first layer (‘conv1_1’) of the VGG-19 network. The images in the second row were generated by jointly representations of the layers of ‘conv1_1’, ‘conv1_2’ and ‘pool1’. From Fig.9, we can see that the textile texture generated by constraining all layers to layer ‘pool4’ are nearly indistinguishable from the original texture (Fig.9, the last row). More results are shown in Fig.10.

![wool lace trim camouflage](image)
Figure 8. Images of the first row (a) are the original textile textures; Images of the second row (b) are the synthesized textile textures; (c) shows the 3D textured trousers model.
Figure 9. Each row corresponds to a different processing stage in the network

Figure 10. (a) 3D textured T-shirt
Figure 10. (b) 3D textured backpack

Figure 10. (c) 3D textured pillow
Finally, we compared our method to conventional scanning method. As shown in Table 2, our method can obtain a good 3D textile model with synthesized texture, which overcomes the restriction of photography texture in the area of 3D scanning.

<table>
<thead>
<tr>
<th>Devices</th>
<th>Laser scanning</th>
<th>Stereovision</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D model quality</td>
<td>perfect</td>
<td>good</td>
<td>good</td>
</tr>
<tr>
<td>Texture</td>
<td>photography</td>
<td>Synthesized texture</td>
<td></td>
</tr>
</tbody>
</table>

6. Discussion and Conclusion

We introduced a novel framework of reconstructing high-quality 3D textile models with synthesized textures. Our main contributions are as follows:

- A novel pipeline is designed to obtain 3D high-quality textile models based on KinectFusion. The accuracy and robustness of KinectFusion are improved via a turntable. The edge of extracted 3D model is smooth and complete.
- To our best knowledge, this is the first paper to explore the synthesized textile texture for the 3D textile model. This is not only simply mapping the texture onto the 3D model, but also exploring the application of artificial intelligence in the field of textile. We hope to inspire more work to
automatically generate 3D textile model and texture, rather than manually design them.

A 3D textile model with better geometry can be obtained using our 3D scanning system. And the texture synthesis method can generate new texture for the 3D textile model, which is overcoming the limitation of texture mapping from photography methods in 3D scanning. As the texture is manually designed in conventional methods. The final textured 3D textile model from our method is high-quality and realistic, as shown in Fig. 10.

There are still some limitations on our method. The 3D textile model cannot be extracted automatically due to its complicated geometry, so the quality of 3D textile models highly depend on manual precision. As to the textile texture synthesis, an original texture should be given in our work, the synthesized texture has a similar style to the original textile texture.

In the future, we hope to generate the arbitrary textile texture without any input images. Also, it is interesting to synthesize the 3D textile geometry rather than using 3D scanning. A similar concept has been proven in the field of human body modeling called Body Talk \[17\]. Only a brief word description, e.g. “medium thin” and “tall”, can generate a good human body shape.


wwwtextures.com