MVA2009 IAPR Conference on Machine Vision Applications, May 20-22, 2009, Yokohama, JAPAN

Texture Overlay onto Flexible Object with PCA of Silhouettes and K-means Method for Search into Database

Hiroshi Tanaka   Hideo Saito

Graduate School of Science and Technology
Keio University, Japan

Abstract

In this paper, we propose a method to overlay an arbitrary texture onto a T-shirt worn by a user. To realize such a system, two phases, an offline phase and an online phase are required. During the offline phase, we use a T-shirt with several markers for training images and establishing a correspondence between the shape of the T-shirt area and the array of markers. During the online phase, we use a T-shirt without any markers for the input image and search for a training image which has the most similar shape to the input image. By using markers from the selected training image, we overlay a texture onto the input image. In this method, we represent the shape of a T-shirt based on its contour so that we can handle occlusion by the hands when the hands cross over the T-shirt. In addition, the computation time is reduced by using PCA and K-means method, which enables our system to work in real time.

1 Introduction

AR(Augmented Reality) enables the addition of virtual objects into a video-captured real space. AR is used for augmenting the user's sight by showing the information which cannot be observed in a real space. Examples of AR include, a car navigation system[1], municipal engineering[2] and entertainment[3].

As an application of AR technique, there are some systems in which the virtual patterns are overlaid onto the surface[4, 5]. We intend to apply AR technique to a virtual clothing system in which the texture is overlaid onto the surface of the clothing and a virtual clothing system enables us to check our appearance. However, it is more difficult to overlay a texture onto the surface of the clothing than a rigid object because the surface of the clothing is complicated when the clothing stretches, wrinkles and self-occludes. Therefore it is a challenging problem to estimate the shape on the surface of the clothing and several methods have been proposed so far to realize a virtual clothing system. For example, Tzvetomir *et al.* overlay a virtual clothing by measuring the 3D shape of a human body[6]. In this approach, however, this approach cannot be applied to the movement of a human body.

Hoshino *et al.* overlay a virtual clothing by estimating the movement of a human body[7]. In this approach, the movement of a human body is estimated by transforming a 3D human model so that the model fits the human in the image. However, the error of the motion estimation might be accumulated as time passes because the motion estimation is based on previously computed model.

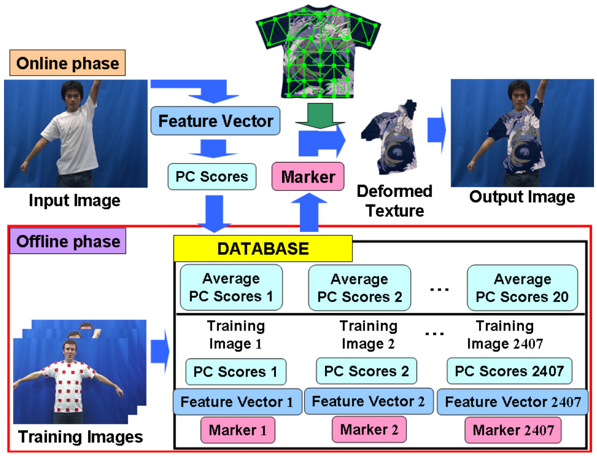


Figure 1: Overview of our method

The common problems among these approaches are that they require special devices such as a 3D range scanner or a motion capture system. In addition, it's difficult to make these methods work in real time because they are based on 3D information and require a lot of computation time.

In order to solve these problems, Ehara *et al.* proposed a method to overlay an arbitrary texture onto the surface of the T-shirt by considering only 2D information with a single web camera[8]. They assume that the deformation on the surface of the clothing depends on the 2D contour of the clothing. However, the occlusion by the hands is not taken into consideration. Therefore, the shape estimation might be incorrect when the hands occlude the clothing.

In this paper, we extend the work[8] by properly describing the shape of the contour of the clothing. Therefore, the shape on the surface of the clothing is estimated correctly even when the hands occlude the clothing. In addition, the computation time is reduced by performing PCA and K-means method, which enables our system to work in real time.

2 Proposed method

2.1 Method

The overview of our method is shown in Fig.1.

Our method is composed of two phases, offline phase and online phase.

In the offline phase, the database is made from training images with several red markers. From each train-

Figure 2: Example of Training Images

ing image, two sets of data, which are a feature vector and marker coordinates, are obtained. A feature vector represents the shape of the contour of the T-shirt and marker coordinates represents how the T-shirt is deformed or observed in the image.

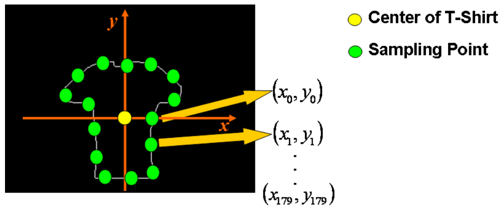
In the online phase, a texture will be overlaid onto the surface of the T-shirt of an input image. From the white T-shirt of an input image, a feature vector is obtained at first. Next, the most similar of the contour is selected from the database. Then, its corresponding marker coordinates are used for those of an input image and a texture image is deformed by performing affine transform between the markers of an input image and those of a texture image. Finally, the output image is obtained by overlaying the deformed texture onto an input image.

2.2 Offline phase

In the offline phase, the marker coordinates of the texture image is specified at first. After that, a person wears the T-shirt with several red markers and take various poses in front of a web camera for the training images Fig.2. Then, two sets of data, which are a 360-dimensional feature vector and marker coordinates, are obtained from each training image for the training data. After making the database, PCA is performed to a set of feature vectors in the database and principal component scores are obtained from each feature vector. With PCA, each 360 dimensional feature vector in the database is transformed to 21 dimensional principal component scores. Here, the dimension of principal component scores, 21, is set when the sum of the contribution rate reaches 99%. In addition, K-means method is performed to a set of principal component scores in the database to cluster the similar principal component scores. With K-means method, each principal component scores are categorized into one of 20 clusters. Here, the number of cluster, 20, is set by considering the number of the poses in the database. On the other hand, the marker coordinates are obtained semi-automatically which means that the marker coordinates are obtained completely manually at first frame and we correct the marker coordinates manually from second frame if marker-tracking fails.

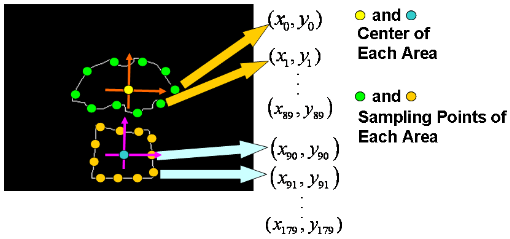
2.2.1 Feature vector

There are various ways to describe the shape of a 2D object, such as a moment feature, roundness, fractal dimension. In our method, we assume that the shape on the surface of the T-shirt is depending on the shape of the contour of the T-shirt. Therefore, a feature vector is needed to represent the shape of the contour of the T-shirt clearly. However, it is impossible to define



$$\mathbf{V} = (x_0, y_0, x_1, y_1, \cdots, x_{179}, y_{179})$$

Figure 3: Feature vector



$$\mathbf{V} = (x_0, y_0, x_1, y_1, \cdots, x_{179}, y_{179})$$

Figure 4: Feature vector in case of occlusion

a feature vector with the coordinates of all the points on the contour because the number of the point on the contour differs according to the shape of the contour and the dimension differs among feature vectors. In order to equalize the dimension of a feature vector, we sampled the same number of points on the contour. Then a feature vector is defined by arraying their x-y coordinates in sequence Fig.3. The starting point is the point where the line of the positive x direction crosses the contour of the T-shirt.

How to get a feature vector is shown in Fig.3. First, the T-shirt area is extracted by detecting white color and red color in RGB space. Next, the contour of the T-shirt is obtained from the image of the T-shirt area. Finally, a feature vector is obtained by sampling the points on the contour and arraying their x-y coordinates in sequence. The number of the sampling points is 180, therefore, the dimension of a feature vector is 360.

In case of the occlusion by the hands, the T-shirt area will be divided into more than one. In such cases, the same number of sampling points are obtained from each area as shown in Fig.4. For example, 90 sampling points are obtained from each area if the T-shirt area is divided into two areas. Therefore, the dimension of all the feature vectors is the same regardless of the number of the T-shirt area.

2.3 Online phase

In online phase, a user wears a T-shirt without markers for the input images. A feature vector of an input image is obtained the same way as the offline phase. After that, the feature vector of the input image is projected into eigenspace previously obtained at the offline phase. Then, the principal component scores of the input image are compared to all the average principal component scores on each cluster and the most similar cluster is selected. Next, all the principal component

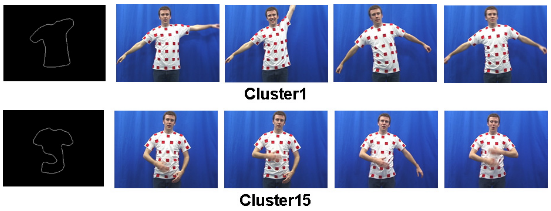


Figure 5: Example of clusters

Figure 6: Overlay and interpolate texture

scores belonging to that selected cluster are compared to the principal component scores of the input image and the most similar principal component scores are selected in the database. Finally, the marker coordinates corresponding to those selected principal component scores are used for the input image. In this way, the computation time is reduced with respect to both the dimension of a feature vector and the number of comparison.

In order to normalize the marker coordinates between the training image and the input image, we consider the width and height of the rectangular of the T-shirt area. Therefore, the marker coordinates of the input image is estimated as follows(1).

$$\begin{aligned} x_{input}(i) &= x_{training}(i)\frac{W_{input}}{W_{training}} \\ y_{input}(i) &= y_{training}(i)\frac{H_{input}}{H_{training}} \end{aligned} \qquad (1)$$

After the marker coordinates of the input image are obtained, the correspondence between the input markers and the texture markers is established and affine transform is performed for all the patches. Here, the texture might be overlaid onto the area out of the T-shirt when the markers for the input image are not properly estimated. In order to overlay the texture onto only T-shirt area, the binary image of T-shirt area is used as the mask image. In addition, the pixel where the texture is not overlaid will be interpolated by the neighboring pixel where the texture is already overlaid Fig.6.

3 Experiment

In this experiment, a user took various poses which are similar to those in the database. The input images are as same as the training images. As illustrated in Fig.7, the texture is deformed according to the shape on the surface of the T-shirt, in other words, the appropriate markers in the database are used for the input image. Even when a user partially occludes the T-shirt, the deformation of the texture is properly estimated Fig.8. Moreover, the texture deformation is properly estimated even when a user crosses the T-shirt by the hands because the feature vectors are properly described Fig.9. However, when the pose in the input image is not similar to any pose in the database, the

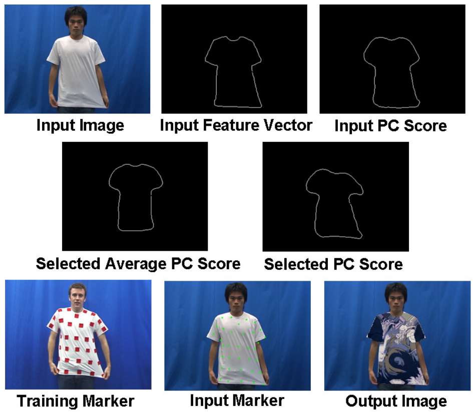


Figure 7: Result of non-occlusion

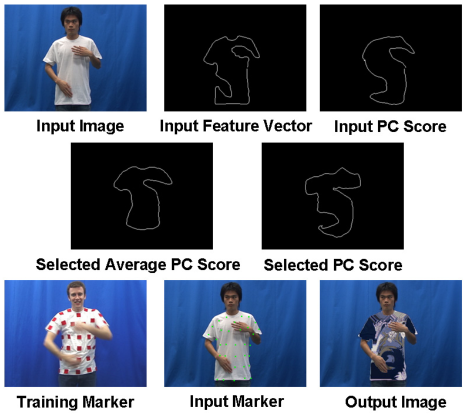


Figure 8: Result of partial occlusion

marker estimation might be done wrongly Fig.10. In such failures, the output image might seem unreal, for example the reality might be lost by the interpolation of a large area as illustrated in the red circle of Fig.10.

To examine how fast our system works, we compared processing time of both using and not using PCA and K-means method. Our method was implemented on a 2.19 GHz PC with 2048MB memory. The resolution of the input images is 320x240 pixels. The number of comparisons without PCA and K-means method is computed by the dimension of a feature vector (360) times the number of the feature vectors (2,407) in the database. The number of comparisons with PCA and K-means method is computed by the dimension of principal component scores (21) times the number of clusters (20) plus the dimension of principal component scores (21) times the average number of the element on a cluster (120). The frame rate of both cases is obtained by measuring the average processing time from 1,000 input images.

Next, we examined the error of the estimated markers with and without PCA and K-means method. For the input images of this experiment, we used a T-shirt with red markers to obtain the ground truth of the marker coordinates and Fig.11 illustrates 0, 50th, 100th, 150th, 200th, 250th frame respectively. We chose these images because the transition of the deformation of the T-shirt is large in the image therefore they are suited for examining the relationship between

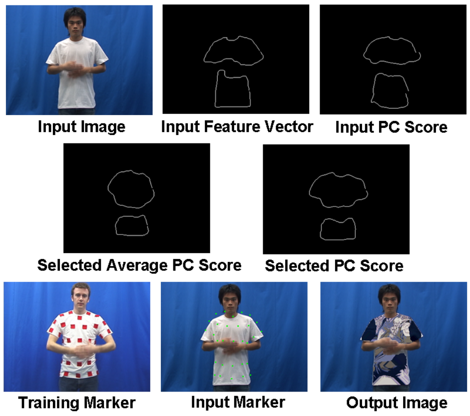


Figure 9: Result of all-occlusion

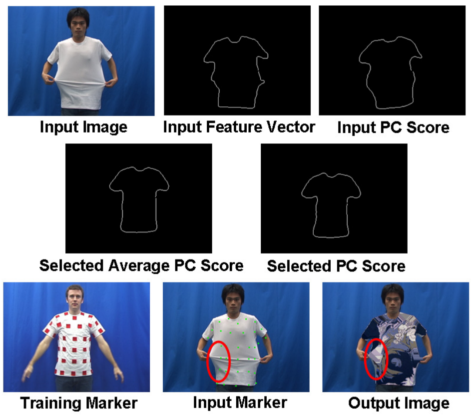


Figure 10: Failure example

	number of comparison	fps
without PCA and K-means method	886,520	3.26
with PCA and K-means method	2,940	22.01

Table 1: Comparison of processing speed

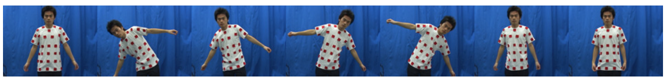


Figure 11: Input images for evaluation

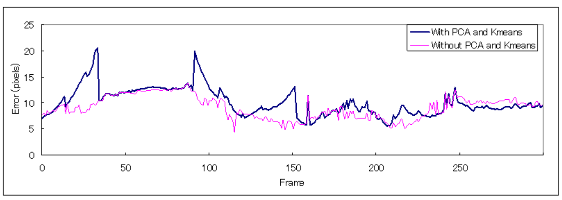


Figure 12: Error of marker estimation

the pose of a user and the error of the marker estimation. Here, we defined the error between the markers of the ground truth and the estimated markers as follows(2).

$$E = \frac{1}{n} \sum_{i \in X(i)} \|\mathbf{m}'_i - \mathbf{m}_i\| \qquad (2)$$

where n is the number of the non-occluded markers, $X(i)$ represents the non-occluded markers, $\mathbf{m}'_i$ is the i th marker being estimated, $\mathbf{m}_i$ is the i th marker of the ground truth. As shown in Fig.12, the error with PCA and K-means method is high in the frames around 30th, 100th, 150th, where the user is in the middle of tilting the body. Otherwise, the error is almost the same with and without PCA and K-means method.

4 Conclusion

We proposed a method to overlay an arbitrary texture onto the surface of the T-shirt for a virtual clothing system and showed that the shape on the surface of the T-shirt was correctly estimated even when the hands occlude the T-shirt. Our method requires only one web camera for a device which means a practical system. Also, we only use 2D information in the image, such as a contour of a T-shirt and marker coordinates, which simplifies the computation process and makes our system work fast. In the future, we would like to overlay arbitrary clothing onto the T-shirt by making a correspondence between the shapes of the T-shirt and arbitrary clothing in the database.

References

[1] Zhencheng Hu, Keiichi Uchimura, and Hanqing Lu, "Fusion of realities for vehicle navigation," International Symposium on Computer Vision, Object Trackin and Recognition, 2004.

[2] Mark Billinghurst, Daniel Belcher, Arnab Gupta, and Kiyoshi Kiyokawa, "Communication behaviors in colocated collaborative AR interfaces," International Journal on Human Computer Interaction, 16(3):395.423, 2003.

[3] Yasuhiro Okuno, Hiroyuki Kakuta, Tomohiko Takayama, and Kazuhiro Asai, "Jellyfish Party: Blowing soap bubbles in mixed reality space," Proc. of the ISMAR, pages 358.359, 2003.

[4] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev, "The MagicBook: a transitional AR interface," Computers & Graphics 25, pp.745-753, 2001.

[5] Mototsugu Emori and Hideo Saito, "Texture Overlay onto Deformable Surface Using Geometric Transformation," ICAT2003, Dec.3-5, 2003.

[6] Tzvetomir I Vassilev, "Dressing Virtual People," SCI'2000 conference, July 23-26, 2000.

[7] Jun'ichi Hoshino, Hirofumi Saito, and Masanobu Yamamoto, "A Match Moving Technique for Merging CG Cloth and Human Video," Journal of Visualization and Computer Animation, Vol.12 No.1, pp.23-29,2001.

[8] Jun Ehara and Hideo Saito, "Texture Overlay for Virtual Clothing Based on PCA of Silhouettes," ISMAR 2006, pp.139 - 142